Post-Placement Voltage Island Generation for Timing-Speculative Circuits

Rong Ye[†], Feng Yuan[†], Zelong Sun[†], Wen-Ben Jone[§] and Qiang Xu[†‡]

[†]CUhk REliable Computing Laboratory (CURE)
Department of Computer Science & Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
[‡]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
Email: {rye, fyuan, zlsun, qxu}@cse.cuhk.edu.hk

[§]School of Electronic and Computing Systems
University of Cincinnati, USA
Email: {jonewb}@ucmail.uc.edu

ABSTRACT

Region-based multi-supply voltage (MSV) design, by which circuits are partitioned into multiple "voltage islands" and each island operates at a supply voltage that meets its own performance requirement, is an effective technique to tradeoff power and performance. Different from conventional voltage island generation techniques that work in a conservative manner to guarantee "always correct" computation, in this work, we investigate the MSV design problem for timing-speculative circuits, which achieves high energy-efficiency by allowing the occurrence of infrequent timing errors and correcting them online. A novel algorithm based on dynamic programming is developed to tackle this problem. Experimental results on various benchmark circuits demonstrate the effectiveness of the proposed methodology.

1. INTRODUCTION

Motivated by the fact that individual blocks of a circuit can have timing/power characteristics unique from the rest of the design, the concept of multi-supply voltage (MSV) design was introduced to trade off power consumption and performance, and has attracted lots of interests from both academia and industry [1–9]. In MSV designs, circuits are partitioned into multiple "voltage islands" and each island operates at a specified supply voltage that satisfies its performance requirement.

In conventional MSV designs, to meet the timing requirement of each voltage island, the corresponding supply voltage has to be high enough to drive the most timing-critical cell, even though the rest of cells may have much more relaxed timing requirements. Moreover, with the ever-increasing variation effects (e.g., process variation effects due to manufacturing imperfection and dynamic variation effects caused by voltage and temperature fluctuations) in nanometer technology, a large design guard band needs to be reserved to tolerate timing uncertainty. Due to the above, we have to be rather conservative when assigning voltages for each island, reducing the possible power savings that can be achieved with MSV designs.

Recently, *timing speculation* (TS) techniques that allow the occurrence of infrequent timing errors and employ error detection and correction techniques to recover from them have emerged as a promising solution to achieve *error-resilient computing* [10–14]. Such "better than worst-case" designs allow the tradeoff between reliability and performance/power, thereby being much more energy-efficient when compared with conventional "worst-case-oriented" designs. Intel [15] has recently demonstrated in their test chip that a timing-speculative microprocessor is able to achieve more than 30% throughput gain when compared to a conventional microprocessor design.

Introducing timing speculation capability into circuits can naturally extend the flexibility of MSV designs to a new horizon, since we do not need to guarantee "always correct" operations any longer and the voltage assignment of islands can avoid being dominated by certain sparse timing-critical cells. How to conduct MSV design for timing-speculative circuits is hence an interesting problem, which, to the best of our knowledge, has not been explored in the literature yet.

Motivated by the above, in this work, we formulate the MSV problem for timing-speculative circuits and develop a novel algorithm based on dynamic programming to solve it. The proposed technique naturally supports "recovery island" design methodology described in [20], wherein each island can recover independent of the rest of the circuit. Experimental results on various benchmark circuits demonstrate that the proposed technique is able to achieve significant power reduction when compared to exiting MSV design techniques.

The remainder of this paper is organized as follows. In Section 2, we present the preliminaries and motivation of this work. The problem formulation and the corresponding algorithms are then detailed in Section 3 and Section 4, respectively. Next, Section 5 presents our experimental results based on various benchmark circuits. Finally, Section 6 concludes this paper.

2. PRELIMINARIES AND RELATED WORK

2.1 MSV Design

A large amount of work has been devoted to MSV designs in the literature and they are applied in various design stages, e.g., floorplanning stage [1, 2], post-floorplanning stage [3], placement stage [4, 5], and post-placement stage [6–9].

DAC'13, May 29 - June 07 2013, Austin, TX, USA.

As pointed out in [7], conducting region-based MSV design before placement based on their logic boundaries, while "natural", is usually far from optimal. Instead, by using placement proximity (instead of logical) information for MSV design, the acquired solution can achieve much better power savings. Motivated by this observation, the authors proposed to utilize dynamic programming (DP) to generate voltage islands considering placement proximity. While DP provides optimal results, the computational complexity and memory requirement to conduct it at fine-grained granularity is not acceptable for a reasonable-sized circuit. Consequently, a heuristic algorithm is used to partition the circuit into $p \times q$ coarse grids first and DP is conducted at the coarse-grained level. While being more efficient, the effectiveness of this technique is inevitably constrained by the heuristic partitioning algorithm. In [8], the authors investigated how to generate an initial voltage assignment considering the physical proximity of high voltage cells as the input of [7]. After that, to tackle the problem that the freedom of voltage assignment is limited by the amount of available slacks on timing-critical paths, [9] performed incremental placement to improve timing on these paths. All the above works try to generate voltage islands with the guarantee that the timing requirements of all cells are satisfied.

2.2 Timing Speculation

Circuit-level timing speculation technique, being able to detect timing errors at online stage, react to the error quickly and recover from it by rolling back to a known-good pre-error state, has become one of the most promising solutions for variation-aware designs. Without loss of generality, let us discuss one of the most representative timing speculation techniques, Razor [10], to illustrate how resilient computation can be achieved with timing speculation. To detect timing errors on a critical path, the receiving end of the critical path, referred to as suspicious flip-flop, is replaced with Razor flip-flop (Razor-FF), which includes a main flip-flop (FF), an additional shadow latch and some control logic. The main flip-flop latches the output signal of the critical path at the clock edge with a possible timing error, while the shadow latch (controlled by a delayed clock signal) latches the signal a fraction of a cycle later, which guarantees to receive the correct value. Consequently, when the shadow latch and the main FF values do not agree, indicated by the comparator, timing error is detected. Then, by replaying instructions at lower frequency, the processor is able to recover from the timing error with a small re-execution cost.

Recently, Intel has demonstrated a timing-speculative microprocessor test chip in [15]. Their measurement results show that the resilient design enables 25% throughput gain over a conventional design by eliminating the guardband from circuit dynamic variations and an additional 7% throughput increase from exploiting the path-activation probabilities for timing error rate reduction. The above benefits have motivated a large amount of recent research efforts on design and optimization techniques for timing-speculative circuits (e.g., [16–19]).

3. MSV DESIGN FOR TIMING-SPECULATIVE CIRCUITS

The MSV design problem for timing-speculative circuits investigated in this work can be formulated as follows:

Problem: Given

- A timing-speculative circuit $\mathcal{C}$, equipped with timing speculators, such as Razor [10];
- A circuit placement $\mathcal{P}$ with $m \times n$ grids , where each grid g_{ij} is placed at position (i, j);

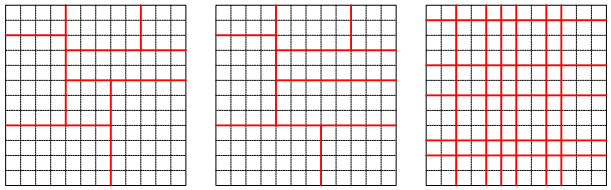

Figure 1: Three types of rectangular partitioning.

- The probability function $F_{ij}(V_{dd})$[1] for timing errors to occur in grid g_{ij} with respect to V_{dd}, where V_{dd} is the supply voltage;
- The number of voltage islands N_{VI};
- The performance degradation constraint caused by re-execution, represented by throughput degradation ratio $\eta\%$;

to determine a circuit partitioning $\mathbf{P}$ and a voltage assignment $\mathbf{V}$ for voltage island generation, such that the power consumption P_{total} of targeted circuit $\mathcal{C}$ is minimized under the performance constraint.

As it is essential to conduct re-computation when timing errors occur, the power consumption of timing-speculative circuits is:

$$P_{total}(\mathbf{P},\mathbf{V}) = P(\mathbf{P},\mathbf{V}) \cdot (1 + error(\mathbf{P},\mathbf{V}) \cdot penalty) \quad , \qquad (1)$$

where $P(\cdot)$ is the power function (including dynamic power P_d and static power P_s) of circuit $\mathcal{C}$ in one clock cycle after circuit partitioning $\mathbf{P}$ and voltage assignment $\mathbf{V}$ are given, $error(\cdot)$ is the error probability function, $penalty$ is the cost including both the cycles of wasted execution that must be discarded and the time spent on checkpointing and re-execution. Meanwhile, we need to ensure the performance constraint:

$$Th(\mathbf{P},\mathbf{V}) = \frac{1}{(1 + error(\mathbf{P},\mathbf{V}) \cdot penalty)} > 1 - \eta\% \quad , \qquad (2)$$

where $Th(\cdot)$ is the equivalent circuit throughput considering performance penalty for timing error correction.

Similar to prior works (e.g., [7–9]), we assume that only rectangular voltage islands are allowed, because voltage islands with arbitrary shapes generally lead to difficulty in power-supply network design. Note that, the hardware cost of MSV design (e.g., the overhead of voltage level shifters [1, 25]) is strongly related to the number of voltage islands, which is also considered in this work.

4. VOLTAGE ISLAND GENERATION

4.1 Partitioning Model

How to partition a circuit into rectangular voltage islands has been well studied in [7, 22]. As described in Fig. 1, arbitrary partitioning allows any partitioning with rectangular tiles, slicing partitioning performs slicing through recursive cuts, and $p \times q$ partitioning cuts the circuit into $p \times q$ coarse grids. [7] proved that the optimal slicing partitioning result is a 2-approximation for the optimal arbitrary partitioning. As a special type of slicing partitioning, $p \times q$ partitioning is used in [7] to provide the initial grids that are merged later to form voltage islands, which is also used in our work.

4.2 DP-Based Voltage Island Generation

To solve the proposed voltage island generation problem for timing-speculative circuits, we resort to a DP-based algorithm that enumerates all combinations of the horizontal and vertical cuts.

[1]The error probability function $F_{ij}(V_{dd})$ of each grid g_{ij} can be acquired by timing simulation of the targeted circuit with representative workloads.

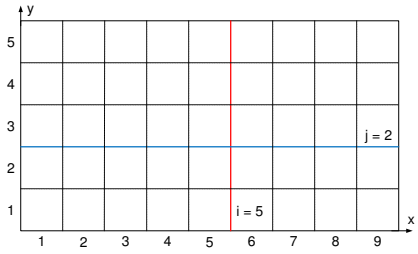

Figure 2: An example to show the enumeration process.

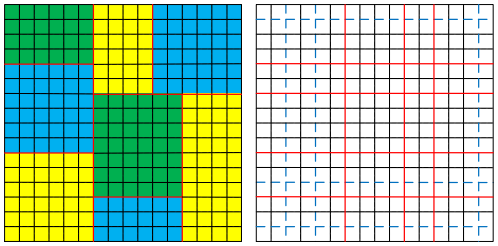

Figure 3: An example to show the coarse grid reconstruction process.

Given the error probability function $F_{ij}(V_{dd})$, we can have the power consumption of each grid g_{ij} considering power penalties,

$$P_{ij}(V_{dd}) = (P_d(V_{dd}) + P_s(V_{dd})) \cdot (1 + F_{ij}(V_{dd}) \cdot penalty) \quad , \qquad (3)$$

where $P_d(\cdot)$ is dynamic power function and $P_s(\cdot)$ is static power function. By solving Eq. 3, we can easily obtain the optimal supply voltage V^*_{dd} for which the power consumption of g_{ij} has the optimal value P^*_{ij}.

Let an $m \times n$ array $\mathbf{A}$ with $A_{ij} = P^*_{ij}$ represent the optimal power consumptions of all the grids, and $R(x_1, y_1; x_2, y_2)$ represent a rectangular region covering the grids $\{g_{ij} | x_1 \le i \le x_2, y_1 \le j \le y_2\}$. For a region $R(x_1, y_1; x_2, y_2)$, we can just replace the power and error probability functions in Eq. 3 with the corresponding terms of this region, and then use such an equation to describe the relationship between power and supply voltage. Similar to the case of a grid g_{ij}, we can also find out the optimal supply voltage V_{opt} for such a region. By denoting the optimal total power consumption of this region with all the grids in it driven by V_{opt} is P^*_R, we define the power wastage of a region $R(x_1, y_1; x_2, y_2)$ as,

$$W(R) = P^*_R - \sum_{g_{ij} \in R} P^*_{ij} \quad . \qquad (5)$$

Therefore, we can have the power wastage of a partitioning $\mathbf{P} = \{R_i\}$ as follows,

$$W(\mathbf{P}) = \sum_{1 \le i \le N_{VI}} W(R_i) \quad , \qquad (6)$$

where N_{VI} is the specified voltage island number.

With the above definitions, we can have the recursion under slicing partitioning as shown in Eq. 4. A simple example is described in Fig. 2 to show the enumeration procedure. In the 9×5 grids with s islands allowed, we can choose an either vertical (e.g., $i = 5$) or horizontal (e.g., $j = 2$) cut to partition it, and allow t and $(s - t)$ voltage islands in the newly-cut rectangular regions, respectively. This enumeration ensures DP to find the optimal partitioning.

Note that, since the error probability functions of grids $\{g_{ij}\}$ and regions $\{R_i\}$ are fed into the DP solver as inputs to calculate the optimal power consumptions, we assume the error occurrences in different grids are independen[2]. This allows us to calculate the error probability function $F_R(V_{dd})$ of a region R, given the error probabilities of the grids $\{g_{ij} | g_{ij} \in R\}$. For example, we can calculate the error probability of a region R consisting of two regions R_1 and R_2 according to Eq. 7 as follows,

$$F_R = F_{R_1} + F_{R_2} - F_{R_1} \cdot F_{R_2} \quad . \qquad (7)$$

4.3 Coarse Grid Reconstruction

The circuit partitioning problem under slicing partitioning can be solved by DP optimally [7]. However, the placement size $m \times n$ at the cell-level is usually too large to employ DP directly in practical applications. To avoid the huge time and memory costs, one intuitive and viable method is to partition the $m \times n$ grids into $p \times q$ coarse grids as shown in Fig. 1(c), and then apply DP to the coarse grids. Clearly, the effectiveness of the MSV design is limited by the heuristic coarse grid construction algorithm due to search space reduction. In [7], a heuristic-based partitioning algorithm according to [22] is used to construct the $p \times q$ coarse grids before voltage island generation. With such fixed coarse grids, only a constrained MSV design solution space can be explored. Different from their solution, we propose a novel coarse grid reconstruction algorithm to explore more solution space by reconstructing coarse grids and applying DP iteratively.

As discussed in Section 4.2, given an array $\mathbf{A}$ consisting of many grids, DP can achieve an optimal solution for this array $\mathbf{A}$ if enough runtime is allowed. With this property, if we ensure the optimal voltage island design of the last $p \times q$ partitioning is still kept as a solution point in a newly-constructed coarse grids, it is guaranteed to achieve a solution not worse than the last one. Let us explain it using the following example.

Suppose we would like to generate 8 voltage islands based on a 16×16 placement and we decide to use 7×8 coarse grids to save runtime, we can perform any partitioning to divide this 16×16 placement into coarse grids and then use the DP algorithm in Section 4.2 to generate voltage islands. By doing so, we can achieve an optimal voltage island design with the current 7×8 coarse grids. Without loss of generality, we assume the generated voltage island design[3] is the one depicted in Fig. 3(a). To construct a new 7×8 coarse girds for further exploration, it is obvious that we need to determine how to partition the 16×16 placement using 6 vertical lines and 7 horizontal. It is worth noting that, if we keep all the grid lines going through the boundaries of voltage islands as the new coarse grid lines (see the solid lines in Fig. 3(b)), we can make sure the current generated voltage islands (see Fig. 3(a)) is still achievable with newly-constructed coarse grids. In other words, given the 3 vertical lines and 4 horizontal lines that go through the boundaries of voltage islands, no matter how we assign the other 3 vertical lines and 3 horizontal lines (see the dashed lines in Fig. 3(b)) to partition the 16×16 placement, the voltage island design in Fig. 3(a) is one possible solution with the reconstructed coarse grids. As DP can always find out an optimal solution with given coarse grids, we should at least find a solution as good as the previous one and hence it is guaranteed to get a solution not worse than the design in Fig. 3(a) under the new 7×8 partitioning.

The above optimization process can be clarified using Fig. 4. The rectangle represents the entire solution space for DP to explore based on the original $m \times n$ fine-grained grids, and the ellipses represent the sub-spaces after partitioning into $p \times q$ coarse grids. Once the $p \times q$ coarse grids are obtained, we can use DP to achieve the optimal solution in the corresponding sub-space. Therefore, by reconstructing the sub-space and applying DP iteratively, we can get the optimal solution in each sub-space one by one: Point A, Point B, Point C, etc.

[2]This is a simple approximation to reduce computational complexity, and its impact is reflected in our experimental results.

[3]The voltage islands are represented by rectangular blocks and plotted out using solid lines.

$$W_s^*(R(x_1,y_1;x_2,y_2)) = \min_{1\le t<s}\left\{ \min_{x_1\le i<x_2, y_1\le j<y_2}\left\{ \begin{array}{l} W_t^*(R(x_1,y_1;i,y_2)) + W_{s-t}^*(R(i+1,y_1;x_2,y_2), \\ W_t^*(R(x_1,y_1;x_2,j)) + W_{s-t}^*(R(x_1,j+1;x_2,y_2)) \end{array} \right\} \right\} \quad . \tag{4}$$

$$metric(L) = \sum_i (sd(R_i) - sd(R_{i1}) \cdot \frac{A(R_{i1})}{A(R_i)} - sd(R_{i2}) \cdot \frac{A(R_{i2})}{A(R_i)}) \cdot \frac{A(R_i)}{\sum_i A(R_i)} \quad . \tag{8}$$

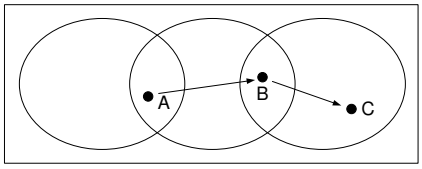

Figure 4: Solution space changes with iterative coarse grid reconstruction.

Bench.	TG #	TFF #	T_{cp} (ns)	(m,n,p,q)	Island #	Cost(%)
38584	21021	1426	6.96	(20, 20, 10, 10)	5	5.14
s38417	23949	1636	6.12	(20, 20, 10, 10)	5	6.76
des_perf	155746	9105	13.7	(30, 30, 15, 15)	10	6.63
ethernet	164912	10752	11.28	(30, 30, 15, 15)	10	7.46
AVERAGE						6.50

TG #, total gate count; TFF #, total FF count; T_{cp}, the operating clock cycle period; Island #, the specified voltage island number.

Table 1: Experimental setup.

4.4 Reconstruction Algorithm

To keep the previous partitioning inside the reconstructed solution space, we would like to use those lines going through the boundaries of voltage islands as coarse grid lines. However, in most cases, there are still some vertical and horizontal lines (see the dashed lines in Fig. 3(b)) left to obtain a different $p \times q$ partitioning, which can be used to explore new solution space. We propose a heuristic-based algorithm to obtain new $p \times q$ partitionings, which selects $(p-1-p_0)$ vertical coarse grid lines out of $(m-1-p_0)$ candidate lines and $(q-1-q_0)$ horizontal coarse grid lines out of $(n-1-q_0)$ candidate lines. Here, p_0 and q_0 are the number of vertical and horizontal lines determined by the boundaries of voltage islands.

The proposed heuristic algorithm is based on the intuition that, for MSV design, it tends to group those grids with similar voltage requirement together, in order to achieve more power savings. In previous works (e.g., [6, 7]), voltages that guarantee no timing violations are chosen. However, for timing-speculative circuits, it is preferable to use the "optimal" voltage values obtained by trading off reliability with power (see Section 4.2). In this work, to support the proposed heuristic algorithm, we use an evaluation metric to reflect the similarity of the grids that are partitioned into the same islands and we tend to select those grid lines with larger metric values during the coarse grid line selection process.

Given a circuit partitioning **P**, if the grid line L intersects n original islands $\{R_i | 1 \le i \le n\}$ to cut them into $2n$ new islands $\{R_{ij} | 1 \le i \le n, 1 \le j \le 2\}$, $metric(L_k)$ is defined as in Eq. 8, wherein $R_i = R_{i1} \cup R_{i2}$, $sd(R_i)$ is the standard deviation of all the optimal voltage values of the grids in region R_i, and $A(R_i)$ is the number of grids in it.

Note that, to avoid being trapped in local optimal points, we use ε-greedy to select the coarse grid lines for $p \times q$ partitioning. That is, we set up a probability parameter ε (e.g., $\varepsilon = 10\%$), and hence we have the probability of ε to select a grid line randomly, instead of the one with largest metric defined in Eq. 8.

5. EXPERIMENTAL RESULTS

5.1 Experimental Setup

To evaluate the effectiveness of the proposed voltage island generation methodology, we conduct experiments on several large ISCAS'89 and IWLS'05 benchmark circuits. We synthesize these circuits on a 90nm technology, conduct physical design, and obtain timing information using commercial EDA tools. To take process variation effects into consideration, we perform Monte Carlo simulations to inject gate-level delay variations following Gaussian distribution.We conduct simulations with random inputs and each simulation is performed with 100,000 cycles. By performing simulation for representative workloads and recording error rates occurring in the grids under various operational clock periods, we achieve error probability function $F_{ij}(V_{dd})$ for each grid. We employ the power and delay models used in [2, **?**, 21] in our experiments. All the experiments are conducted on a *2.8GHz* PC with *4GB* RAM.

We perform offline timing analysis with false paths excluded according to [23] and use the reported maximum path delay as the operating clock cycle period during timing simulation. For reasonable comparison, a widely-accepted voltage island generation algorithm proposed in [7] is used as the baseline solution and denoted as $MSV_{baseline}$. Because our proposed reconstruction-based $p \times q$ partitioning algorithm is also applicable for the non-TS voltage island generation problem in [7], we replace the corresponding $p \times q$ partitioning algorithm in [7] with ours and keep the rest of algorithm unchanged. This MSV design scheme is denoted as $MSV_{reconstruciton}$. We apply timing speculation directly to the MSV design of $MSV_{baseline}$, and denote this solution as MSV_{ts}. That means, in MSV_{ts} we keep the MSV design of $MSV_{baseline}$ and then perform timing simulation with different voltage assignments to obtain the error probability functions, so that we can achieve the "optimal" voltage assignment and power consumption considering timing speculation. Our proposed solution is denoted as $MSV_{proposed}$. The range of supply voltages allowed for voltage islands to operate is 0.7V to 1.0V in our experiments.

In timing-speculative circuits, we need to add timing error detectors to the receiving end of critical paths. A simple scheme is to transform all the FFs, whose maximum path delays are larger than β of the clock period (e.g., β = 80%), as Razor-FFs. Then, to avoid hold time violation on the shadow latch of Razor-FFs, we need to conduct short path padding and this is achieved by constraining paths that drive Razor-FFs with at least γ of the clock period (e.g., γ = 50%) during synthesis. In this work, once a voltage island design is generated, we perform timing analysis using timing information with voltage scaling considered and then set up Razor-FFs and conduct short path padding using the obtained path delays. Both of these hardware costs are accounted for in our experiments and β and γ are set to be 80% and 50%, respectively. The hardware cost for equipping each Razor-FF is assumed to be 10 gates. The *penalty* in Eq. 1 is assumed to be 10 clock cycles similar to prior works (e.g., [14]).

5.2 Results and Discussion

In Table 1, we report the operating clock period obtained by excluding false paths according to [23], the used parameters (m,n,p,q), the specified voltage island number and the hardware cost to enable timing speculation for each benchmark circuit. To be specific, we set up the values of (m, n, p, q) as $(20,20,10,10)$ for small-scale circuits

Bench.	$MSV_{baseline}$		$MSV_{reconstruction}$			MSV_{ts}				$MSV_{proposed}$					
	power	σ	*power*	σ	$\Delta_1(\%)$	*power*	σ	$\Delta Th(\%)$	$\Delta_2(\%)$	*power*	σ	$\Delta Th(\%)$	$\Delta_3(\%)$	$\Delta_4(\%)$	Runtime (s)
s38584	0.852	0.014	0.813	0.015	-4.58	0.793	0.014	-4.28	-6.92	0.689	0.016	-3.55	-13.11	-19.13	2.75
s38417	0.857	0.013	0.835	0.016	-2.57	0.825	0.018	-3.56	-3.73	0.781	0.020	-4.09	-5.33	-8.87	1.92
des_perf	0.862	0.019	0.806	0.017	-6.50	0.674	0.014	-5.52	-21.81	0.598	0.015	-7.03	-11.28	-30.63	17.35
ethernet	0.778	0.018	0.723	0.019	-7.07	0.631	0.015	-6.32	-18.89	0.581	0.012	-5.63	-7.92	-25.32	15.04
AVERAGE					-5.18			-4.92	-12.84			-5.08	-9.41	-20.99	

σ: standard deviation of *power*; Δ_1: *power* difference ratio between $MSV_{reconstruction}$ and $MSV_{baseline}$; Δ_2: *power* difference ratio between MSV_{ts} and $MSV_{baseline}$;
Δ_3: *power* difference ratio between $MSV_{proposed}$ and MSV_{ts}; Δ_4: *power* difference ratio between $MSV_{proposed}$ and $MSV_{baseline}$; ΔTh, performance degradation ratio.

Table 2: Results on the proposed reconstruction-based $p \times q$ partitioning.

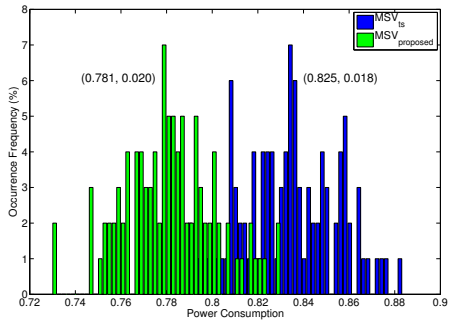

Figure 5: Monte Carlo simulation results.

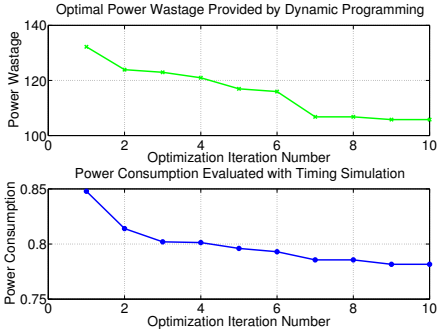

Figure 6: Power wastage and power consumption wrt. optimization iteration number.

(e.g., *s*38584 and *s*38417) and as $(30, 30, 15, 15)$ for large-scale circuits (e.g., *des_perf* and *ethernet*). The average hardware cost to equip the circuits with TS capability (including timing speculator and short path padding cost) is about 6.5%.

To verify the effectiveness of the proposed voltage island generation methodology, we, first of all, perform Monte Carlo simulation to produce 100 sample chips with different variation patterns for each benchmark circuit. In Table 2, we report the average power consumption[4] and its standard deviation σ for $MSV_{baseline}$, $MSV_{reconstruction}$, MSV_{ts} and $MSV_{proposed}$, respectively. It is important to note that, the reported results includes the power overhead of MSV design (e.g., level shifters) and power penalties to correct timing errors.

As can be seen from Table 2, compared to $MSV_{baseline}$, the proposed $MSV_{reconstruction}$ can achieve 5.18% power saving on average. This improvement comes from using our proposed $p \times q$ partitioning algorithm to replace the corresponding one in $MSV_{baseline}$ only, which demonstrates the effectiveness of our reconstruction algorithm. In other words, even for non-TS conventional circuits, our proposed solution lead to much more power-efficient MSV designs.

Besides, MSV_{ts} can achieve 12.84% power reduction on average when compared with $MSV_{baseline}$. This improvement reflects the efficacy of timing speculation itself, since in MSV_{ts} we just apply timing speculation directly to the MSV design of $MSV_{baseline}$. Compared with MSV_{ts}, our proposed methodology $MSV_{proposed}$ can further achieve 9.41% power reduction on average, which reflects the efficacy of explicitly considering timing speculation during the MSV design process. The runtime of the proposed algorithm (see Fig. **??**) is quite small.

$MSV_{proposed}$ achieve better results because (i) the proposed partitioning model and DP-based voltage island generation method facilitate to identify voltage islands with optimal supply voltages based on circuit slack distribution, which gives the first-level of power saving; (ii) once a voltage island has been formed, another level of power saving can be achieved by minimizing the timing error rates. Any voltage island with only a small number of critical paths (i.e., most circuit paths have relatively large slacks) can fully take advantage of this power saving while maintaining the performance. At the same time, we can observe that the power reduction ratios of these four benchmark circuits are quite different, and we attribute this phenomenon to the unique timing characteristic of each circuit. Generally speaking, if a circuit has gradually-decreasing path delay distribution, the benefit brought by timing speculation can be larger than that of those circuits with a sharply-declining delay distribution. This is because, in the latter case, a large number of paths may fail at the same time in the design when voltage overscaling exceeds a critical point, which causes a steep increase of timing error rate [24].

MSV_{ts} and $MSV_{proposed}$ would suffer from performance degradation caused by infrequent timing errors. We report this performance degradation in Table 1 and denote it as ΔTh, compared to the case that the circuit uses the maximum path delay as its operational clock period. On average, MSV_{ts} and $MSV_{proposed}$ have 4.92% and 5.08% throughput degradation, respectively. However, it is important to note that, for $MSV_{baseline}$ without timing error correction capability, designers usually have to reserve a large timing guard band (e.g., 15% of maximum path delay) to tolerate variation-induced timing uncertainty and hence system throughput is degraded due to lower operational frequency [15]. From this perspective, if we consider the timing guardband existing in the non-TS solution $MSV_{baseline}$, the performance of MSV_{ts} and $MSV_{proposed}$ would be actually better than that of $MSV_{baseline}$.

To get more details of the proposed methodology, we take *s*38417 as an example in the following experiments. In Fig. 5, we show the results of MSV_{ts} and $MSV_{proposed}$ with process variation effects after performing Monte Carlo simulation. The corresponding mean value of power consumption and standard deviation for each case are depicted in the figure in the form of (μ, σ), which, again, demonstrates the benefits of $MSV_{proposed}$. In Fig. 6, we plot the curves to reflect the changes of both the power wastage provided by DP and the power consumption evaluated by timing simulation with error penalties taken into account. As can be seen, the power wastage is decreased all the time, which proves the effectiveness of the reconstruction algorithm to explore new solution space and guarantee the power wastage to be optimized step by step, as discussed in Section 4.3. Note that, this can be used to trade off the algorithm runtime with optimization quality during design process. Moreover, with respect to the

[4]Each power value has been normalized by using the power consumption of the case without MSV design as unit value.

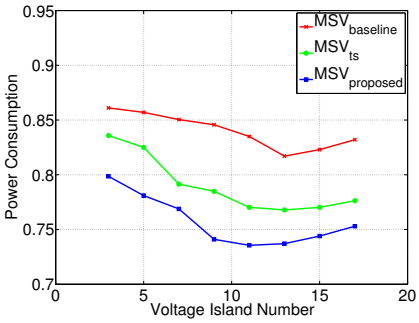

Figure 7: Power consumption wrt. voltage island number.

optimization iteration number, the two curves descends in the same manner. The similar trends of these two curves can prove the effectiveness of our proposed optimization process. To investigate the effects with different specified voltage island number, we vary the number of islands and get the power consumption curves of $MSV_{baseline}$, MSV_{ts} and $MSV_{proposed}$ as described in Fig. 7. Clearly, with different number of voltage islands, $MSV_{proposed}$ always outperforms the other solutions. It can be also observed that, with increasing number of allowed voltage islands in the MSV design, the power savings of all these solutions increase in the beginning, but decrease in the end. This is because, more voltage islands allow fine-grained voltage assignments that satisfy the performance constraint of each individual island, leading to better power savings. However, more voltage islands also incur higher cost for the supporting circuitries (e.g., level shifters). Consequently, when the number is too large, the benefit provided with fine-grained voltage assignment cannot compensate the associated power cost.

6. CONCLUSION

Region-based MSV design has been used as an effective technique to reduce power consumption and attracted lots of research interests. However, all of the previous MSV works try to guarantee "always correct" operations, which greatly limits the design flexibility. In this work, we formulate the MSV design problem for timing-speculative circuits, and propose a novel DP-based algorithm to generate voltage islands. Experimental results based on various benchmark circuits demonstrate that the proposed methodology is able to significantly reduce power consumption of timing-speculative circuits with acceptable performance degradation.

7. ACKNOWLEDGEMENT

This work was supported in part by the Hong Kong SAR Research Grants Council under General Research Fund No. CUHK418111 and No. CUHK418812.

8. REFERENCES

[1] W. K. Mak, J. W. Chen, Voltage Island Generation under Performance Requirement for SoC Designs. In *Proc. IEEE/ACM Asia South Pacific Design Automation Conference (ASP-DAC)*, pp. 798-803, 2007.

[2] Q. Ma, E. Young, Multivoltage Floorplan Design. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 4, pp. 607-617, April 2010.

[3] W. Lee, H. Liu, Y. Chang, An ILP Algorithm for Post-Floorplanning Voltage-Island Generation Considering Power-Network Planning. In *Proc. ACM/IEEE International Conference on Computer-Aided Design (ICCAD)*, pp. 650-655, 2007

[4] B. Liu, Y. Cai, Q. Zhou, X. Hong, Power Driven Placement with Layout Aware Supply Voltage Assignment for Voltage Island Generation in Dual-Vdd Designs. In *Proc. IEEE/ACM Asia South Pacific Design Automation Conference (ASP-DAC)*, pp. 582-587, 2006.

[5] L. Guo, Y. Cai, Q. Zhou, X. Hong, Logic and Layout Aware Voltage Island Generation for Low Power Design. In *Proc. IEEE/ACM Asia South Pacific Design Automation Conference (ASP-DAC)*, pp. 666-671, 2007.

[6] R. Ching, E. Young, K. Leung, C. Chu, Post-Placement Voltage Island Generation. In *Proc. ACM/IEEE International Conference on Computer-Aided Design (ICCAD)*, pp. 641-646, Nov. 2006.

[7] H. Wu, M, Wong, I. Liu, Y. Wang, Placement-Proximity-Based Voltage Island Grouping Under Performance Requirement. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.26, no.7, pp.1256-1269, July 2007.

[8] H. Wu, M. Wong, I. Liu, Timing-Constrained and Voltage-Island-Aware voltage assignment. In *Proc. ACM/IEEE Design Automation Conference (DAC)*, pp. 429-432, 2006.

[9] H. Wu, M. Wong, Incremental Improvement of Voltage Assignment. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.28, no.2, pp.217-230, Feb. 2009.

[10] D. Ernst, et al., Razor: a Low-Power Pipeline based on Circuit-Level Timing Speculation. In *Proc. IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 7-18, 2003.

[11] S. R. Sarangi, et al., VARIUS: a Model of Process Variation and Resulting Timing Errors for Microarchitectus. In *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, pp. 3-13, Feb. 2008.

[12] S. Sarangi, B. Greskamp, A. Tiwari, and J. Torrellas, EVAL: Utilizing Processors with Variation-Induced Timing Errors. In *Proc. IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 423-434, 2008.

[13] T. Austin, V. Bertacco, D. Blaauw and T. Mudge, Opportunities and Challenges for Better than Worst-Case Design. In *Proc. IEEE/ACM Asia South Pacific Design Automation Conference (ASP-DAC)*, pp. 2-7, 2005.

[14] M. de Kruijf, S. Nomura, and K. Sankaralingam, A Unified Model for Timing Speculation: Evaluating the Impact of Technology Scaling, CMOS Design Style, and Fault Recovery Mechanism. In *Proc. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 487-496, 2010.

[15] K. Bowman, J. Tschanz, C. Wilkerson, S. Lu, T. Karnik, V. De, S. Borkar, Energy-Efficient and Metastability-Immune Resilient Circuits for Dynamic Variation Tolerance. In *IEEE Journal of Solid-State Circuits*, 44(1): 49–62, 2009.

[16] B. Greskamp, et al. Blueshift: Designing processors for timing speculation from the ground up. *Proc. IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 213-224, 2009.

[17] A. B. Kahng, et al. Slack redistribution for graceful degradation under voltage overscaling. *Proc. IEEE/ACM Asia South Pacific Design Automation Conference (ASP-DAC)*, pp. 825-831, 2010.

[18] R. Ye, F. Yuan and Q. Xu. Online clock skew tuning for timing speculation. *Proc. ACM/IEEE International Conference on Computer-Aided Design (ICCAD)*, pp. 442–447, 2011.

[19] Y. Liu, et al. On logic synthesis for timing speculation. *Proc. ACM/IEEE International Conference on Computer-Aided Design (ICCAD)*, pp. 591–596, 2012.

[20] V. Kozhikkottu, S. Dey, A. Raghunathan, Recovery-based design for variation-tolerant SoCs. In *Proc. ACM/IEEE Design Automation Conference (DAC)*, pp. 826-833, 2012.

[21] K. Roy, S. Mukhopadhyay, H. Mahmoodi-Meimand, Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS Circuits. In *Proceedings of the IEEE*, vol.91, no.2, pp. 305- 327, Feb. 2003.

[22] S. Muthukrishnan, T. Suel, Approximation Algorithms for Array Partitioning Problems. In *Journal of Algorithms*, Volume 54, Issue 1, pp. 85-104, Jan. 2005.

[23] F. Yuan and Q. Xu, On Timing-Independent False Path Identification. In *Proc. ACM/IEEE International Conference on Computer-Aided Design (ICCAD)*, pp. 532-535, 2010.

[24] J. Sartori and R. Kumar, Architecting Processors to Allow Voltage/Reliability Tradeoffs. In *Proc. ACM International Conference on Compilers Architecture and Synthesis for Embedded Systems (CASES)*, pp. 115-124, 2011.

[25] J. Lin, W. Cheng, C. Lee and R. C.J. Hsu, Voltage island-driven floorplanning considering level shifter placement. In *Proc. IEEE/ACM Asia South Pacific Design Automation Conference (ASP-DAC)*, pp. 443-448, 2012.