Process Variation Tolerant Pipeline Design Through a Placement-Aware Multiple Voltage Island Design Style

Bonesi Stefano†, Davide Bertozzi†, Luca Benini‡, Enrico Macii⋆,
† ENDIF, University of Ferrara, 44100 Ferrara, Italy.
⋆ DAUIN, Politecnico di Torino, 10129 Torino, Italy.
‡ DEIS, University of Bologna, 40136 Bologna, Italy.

Abstract

A common technique to compensate process variation induced performance deviations during post-silicon testing consists of the dynamic adaptation of processor voltage. This however comes at a significant power cost. We envision multi supply voltage design (MSV) as a promising technique to mitigate such power overhead. Voltage islands are widely recognized as the state-of-the-art in MSV design. In this paper, we develop a novel design methodology that leverages voltage islands to compensate process variations through a commercial synthesis flow. Possible violation scenarios of performance requirements in fabricated chips are pre-characterized at design time through statistical static timing analysis. Then, during post-silicon testing the supply voltage of a proper number of voltage islands is raised depending on the actual violation scenario, thus bringing performance back within nominal values. Voltage islands are generated by exploiting cell proximity for minimal perturbation of performance pre-optimized placements.

1. Introduction

The basic uncertainty and variance of process parameters in sub-90 nm CMOS technologies are causing a major rethinking of design techniques and tools across the design stack, from high-level microarchitecture to physical design and layout. Advanced techniques to meet parametric yield and/or performance requirements detect process parameter deviations after manufacturing, and compensate or correct them by adapting supply voltage, clock frequency and body bias or tuning the clock skew [5]. While clock frequency adaptation trades-off performance with process variation tolerance, pipeline retiming through clock skew adjustment for each pipeline stage is effective only when variations slow down some stages while they speed up others [1]. This assumption might not hold in presence of large systematic variations featuring high spatial correlation. It is showed in [13] that adaptive voltage scaling (AVS) [18] requires a much smaller change (percentage-wise) in supply voltage than adaptive body biasing (ABB) [17] requires in threshold voltage to achieve a target frequency boost of a processor core. As a result, AVS has a much milder impact on leakage and is a more power-efficient and thermally compatible solution than ABB.

For this reason, this paper assumes supply voltage adaptation as the reference technique to boost performance of fabricated chips affected by process parameter variations. Nonetheless, the power overhead induced by this technique is significant, and this paper explores the use of dual supply voltage design techniques to achieve a better trade-off between power consumption and process variation tolerance. Although dual-Vdd design is traditionally used to reduce dynamic and leakage power by assigning high-Vdd to cells on timing critical paths and low-Vdd to cells on non-critical paths, we extended it to cope with process variations. The basic intuition behind our work is that an increasing subset of selected cells in the design could be operated at a higher supply voltage than the nominal one depending on the amount of process variation-induced performance deviations to compensate. We target voltage island (VI) technology [29] to partition the design into subsets of cells featuring the same supply voltage, since it represents the state-of-the-art in multi-Vdd design. This paper illustrates a novel design methodology for process variation tolerant microarchitecture design based on the above strategy and relying on the available multi-voltage design support in commercial synthesis tools [31, 32].

The proposed design methodology assumes that a given RTL design has already undergone placement-aware logic synthesis driven by performance optimization directives. Once the global placement is determined and timing can be estimated with enough precision, the resulting netlist is fed to an additional design step where process variations are addressed. The objective at this stage is to minimally impact performance results achieved by the previous physical synthesis run, while avoiding worst-case timing margins for process parameter variations.

The methodology involves two steps. At design time, variation-aware statistical static timing analysis (SSTA) allows to pre-characterize multiple scenarios of performance constraint violations in fabricated chips. For each scenario, placement-aware cell grouping algorithms are then applied to identify the minimum subset of adjacent cells to be operated at high-Vdd to tackle the timing violation through the correspondent performance boost. Voltage islands are generated in such a way that moving from a violation scenario to a more severe one, only the supply voltage of 1 additional voltage island needs to be raised. After chip fabrication, timing sensing circuits integrated into the design assess with a high level of correctness whether a given process variation scenario took place, and the most appropriate number of voltage islands is powered at high-Vdd as determined at design time.

We validated the methodology on a 4-way VLIW architecture and on a 65nm technology library, and demonstrated the power savings it can achieve with respect to traditional full chip supply voltage adaptation techniques in spite of the overhead associated with level shifter insertion.

This paper is structured as follows. Related work is address in Section 2. Our VI-based design methodology is illustrated in Section 3, while its intermediate steps are detailed and validated for a real design in Section 4. Power savings achievable by VI-based design styles with respect to chip-wide supply voltage adaptation techniques are proved in Section 5.

2. Related Work

Following a common practice, designers often add on worst-case guard bands to critical paths to account for variability. Statistical static timing analysis has been proposed as a way to re-

978-3-9810801-3-1/DATE08 © 2008 EDAA

duce design timing margins [15, 16], and some commercial design tools support it to some extent [24]. Characterization of process variations is of upmost importance for any attempt to analyze and optimize designs statistically [9, 20, 21].

The need to deal with process variations at a higher level than circuit techniques has also motivated more abstract modeling strategies [8], allowing to capture the impact of parameter variations on different microarchitectural units and their performance [23, 25]. A model allowing microarchitects to reason about how within-die variations may affect a multi-core environment is reported in [7, 13].

As summarized by the survey in [5], three kinds of techniques have been proposed to enhance yield under variations while minimizing the design overhead. In the statistical design approach, circuit parameters are modeled as statistical distributions and the circuit is designed to meet a constraint on yield, typically acting on gate sizing or dual-Vth assignment [15, 22]. Alternatively, a variation avoidance approach aims at synthesizing circuits in such a way that delay failures due to variations can be identified at runtime and avoided by adaptively switching to two-cycle operations [10]. Finally, a number of techniques take the approach of post-silicon compensation and correction. In this case, parameter shifts are detected and compensated after manufacturing by changing supply voltage [18], frequency, body bias [17] or clock skew [1]. Under certain variation scenarios, adaptive pipelining techniques [1, 11] have been proposed to avoid unacceptable operating frequency penalties.

This paper relies on supply voltage adaptation during post-manufacture testing as a means of reducing performance degradation induced by process variations. [18] is a milestone work in this domain. In [14, 19] a global voltage controller adjusts the supply voltage by monitoring the error rate of the entire pipeline. Only in [12] each pipeline stage is provided with its own voltage controller and can thus be operated at its own optimized voltage. From an implementation viewpoint, the concept of operating design sub-blocks or even single cells at a separate supply voltage is today common practice for low-power designs through a voltage island design style [29]. Early approaches applied this technique on a core-by-core basis [33]. Other researchers advocate for a finer granularity VI generation. On one hand, this prevents from raising the voltage of an entire core or unit only because of the presence of timing critical cells in it. On the other hand, a logic-based generation of voltage islands constrains the physical placement [26]. For this reason, fine-grain voltage-islands derived through a placement driven synthesis framework were proposed in [27, 28].

Our work takes an even more radical approach to placement-driven VI generation, which aims at the minimum modification of performance pre-optimized placements through the grouping of physically adjacent cells into the same voltage island. Moreover, we precisely quantify the level shifter overhead and its impact on total system power and area for an industrial 65nm technology library. Nonetheless, we prove the viability of the voltage island approach to deal with process variations by assessing the power savings with respect to chip-wide supply voltage adaptation. Commercial synthesis tools and their support for multi-voltage design are used for the experiments.

3. Methodology for Process Variation Tolerant Design

The design methodology we propose (see Fig.1) adds a few additional steps to the traditional synthesis flow of an ASIC design. In this work we assume that logic synthesis and coarse placement were optimized for high performance. This is in agreement with the choice of adaptive supply voltage as the reference compensation technique for variation-induced delays. The input to our methodology is a placed gate-level netlist, which is output by a physical compiler tool. The reason for this is that we want to avoid placement-agnostic voltage assignment to cells, since it might incur a few major problems. First, interconnect delay can be hardly estimated before placement with reasonable accuracy. This might cause too many Low-Vdd cells, with consequent timing closure troubles, or too few Low-Vdd cells, wasting useful slack

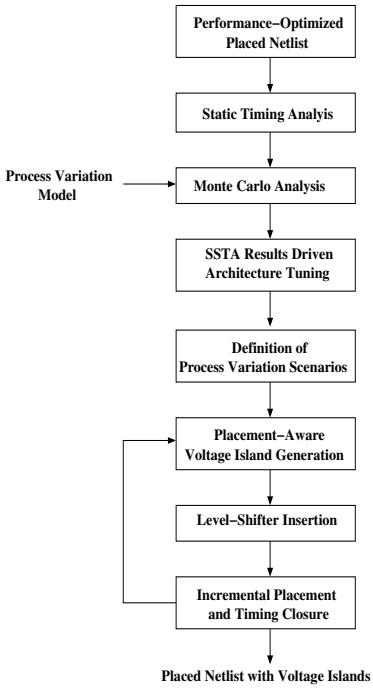

Figure 1. Voltage island based design methodology for process variation tolerance.

for power optimization. Second, logic-based voltage assignment heavily constrains the placement, and hence might jeopardize design predictability by giving rise to unexpected large wirelengths and delay penalties. As a consequence, we intend to rely on physical information for cell voltage assignment. This approach is further pushed by the fact that placement-aware logic synthesis tools are becoming mainstream. After a quick initial logic synthesis based on wireload models, such tools internally attempt a coarse placement of the current netlist, and also keep optimizing it based on the expected placement and the wire loads it implies. The resulting netlist already considers placement-related effects, and we found it an ideal candidate for feeding our voltage island based design methodology.

Then, static timing analysis is carried out to characterize design performance in typical conditions. Parasitics as extracted from the physical synthesis tool should be accounted for in delay calculations. At this stage, standard file formats do exist to transfer delay information between tools, and to report delay values for each cell/path in the design.

Annotated delays then serve as the nominal values for the SSTA. This latter requires the knowledge of the statistical characteristics of the manufacturing process, which are used to inject delay variations in the netlist. We propose to model both systematic and random process variations based on measured data and on modeling strategies reported in the literature. A typical approach consists of modeling die-to-die and within-die effective gate length and threshold voltage variations, resulting in a statistical distribution of gate delays.

The Monte Carlo analysis then allows to derive the distribution of the maximum delay in the design, or of the maximum delay per stage in pipelined designs. Due to the effect of systematic variations, the parameters of the output distributions are a function of the chip location on the exposure field and of a functional unit placement within the chip. The power of Monte Carlo analysis lies in the ability to display the process output both in range and shape. Both of them can provide valuable information on the design (e.g., distribution of path lengths, logic depth) and guidelines for its optimization. As an example, the work in [22] reports pipeline yield improvements through a modification of logic depth and imbalance between stage delays.

The knowledge of the range of parameter distributions can also be exploited to tune post-silicon self-calibrating compensation techniques. In this case, integrated timing sensing circuitry needs to detect or even to quantify violations of performance requirements. Common techniques are the shadow latches for delayed sampling implemented in Razor flip flops [14] or the signal latency detection unit presented in [6]. In both approaches, delays need to be applied to the clock or to other signals, and the value of such delays could be tuned based on the results of the Monte Carlo analysis. This is also the approach which is taken in this paper.

Our methodology then proceeds with placement-aware voltage island generation. At this stage, several cell grouping strategies can be used. Generating voltage islands at the functional unit level, as in [29], achieves differentiated power and frequency for each unit, and can be used for instance to gate the power supply of the entire macro in order to completely power it off, thus cutting down on standby power. Unfortunately, the coarse grain granularity of such partitioning is not suitable for designs with tight power or timing budgets. Alternatively, microarchitecture sub-units can be selected for voltage assignment [12]. However, this comes with heavy placement implications, since the need to keep cells belonging to the same unit close together in the layout prevents aggressive physical optimizations. To minimize the physical design overhead while still exploiting the logic structure of the design, cells in the same signal paths can be clustered in the same voltage islands. The need to minimize the number of level shifters is orthogonal to all design strategies. The approach we propose in this paper is to exploit the physical proximity of placed cells for voltage island generation in order to minimize perturbations of the previous performance-optimized placement. Although in principle all kinds of placement-aware grouping strategies can be explored [27, 28] and their implications on global timing closure assessed, this paper will present conservative voltage island generation strategies grouping cells that have already been placed close together, without any further cell clustering and placement refinement iteration. We envision incremental placement only for level shifter insertion, as showed in Fig.1. Level shifters might in fact lead to some performance degradation or power overhead or to dead spaces in the floorplan, and this might require to group adjacent cells based on different criteria.

In the following section, the above methodology will be validated on the design of a processor core. The physical synthesis and analysis toolflow from Synopsys will be used, consisting of Physical Compiler for physical synthesis, PrimeTime for static timing analysis and PrimePower for power reports.

4. Methodology validation

As a first step toward the validation of the above methodology, we need to develop a model for process variations and to come up with a representative test design.

4.1 Process Variation Model

In nanoscale CMOS technologies, the most important processing parameter affecting circuit performance is the effective gate length (L_{gate}) of the MOS transistor. ITRS roadmaps report the 3σ standard deviation of the L_{gate} distribution to determine the control limits expected in a given process. For our target 65 nm technology node, we derived from [3] a slightly conservative value of $3\sigma_{TOT}/\mu = 9\%$ for L_{gate} control. We then divided the L_{gate} variations into two components: across-field systematic variations $f(x, y)$ and random variations ϵ. In [2], a slightly larger impact of the random over the systematic component on 65nm process variability is predicted, resulting in a ratio of the respective 3σ values of about 1.2. Combining these considerations leads to a variability of $3\sigma_{Rnd}/\mu = 6.5\%$ for the random component and of $3\sigma_{Sys}/\mu = 5.5\%$ for the systematic component.

While this model captures L_{gate} variance and hence is effective in describing gate/path delay variations, we are also interested in capturing the relation between a particular position within the chip and process variation induced delay deviations. In fact, in nanoscale technologies stepper systems are forced to operate closer to their optical resolution limit, thus inducing illumination and imaging nonuniformity due to lens aberrations. As a result, printed transistors display a distinct spatial L_{gate} map, making their characteristics dependent on the location within the chip. It was showed in [7, 9] that this systematic component of intrachip L_{gate} variability can be modelled as a simple polynomial function of position within the exposure field. Assuming a 28mm x 28mm exposure field, the equation for the second order polynomial model of L_{gate} can be approximated by

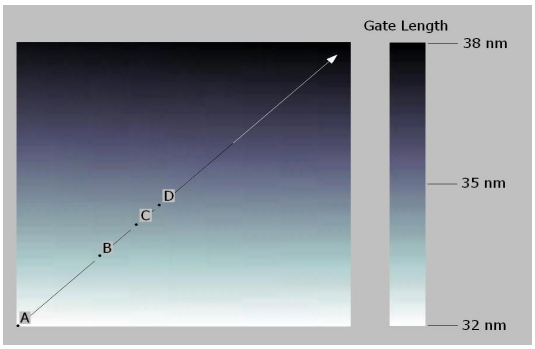

Figure 2. Systematic variation aware L_{gate} map.

$$f(x, y) = a \cdot x^2 + b \cdot y^2 + c \cdot x + d \cdot y + e \cdot xy + intercept \quad (1)$$

where x and y are the coordinates on the exposure field. Baseline values for the coefficients were provided in [9] for a 130nm industrial process from measured data. We scaled those coefficients for our target 65nm technology so to match the range of systematic variations computed above. Our model hence generates maximum systematic L_{gate} deviations by +/- 5.5% with respect to the nominal value. An example L_{gate} map on a 14mm x 14mm chip accounting for systematic variations is reported in Fig.2. A VLIW core placed in the lower-left corner (point A) would experience the slowest performance, while in the upper-right corner it would be maximally accelerated.

As a result of our modeling framework, in each coordinate of the exposure field we compute effective gate length by summing a polynomial function modeling systematic variations and a normally distributed random variable:

$$L_{gate}(x, y) = f(x, y) + \epsilon \quad (2)$$

We ignore variation in wires, in agreement with current variation models [13].

In order to model the dependency between L_{gate} and gate delay D, we used the model proposed by Orshansky et al. [20]:

$$D \sim L_{gate}^{1.5} \cdot V_{dd}/(V_{dd} - V_{th})^{\alpha} \quad (3)$$

where V_{dd} is the supply voltage, V_{th} the threshold voltage and α the velocity saturation. As channel lengths become shorter, this value will approach 1. Similarly to [7], we chose $\alpha = 1.3$.

Because of drain induced barrier lowering, gate length also affects threshold voltage. We accounted for this dependency through the model presented in [30]:

$$V_{theff} = V_{th0} - V_{dd} \cdot exp(-\alpha_{DIBL} \cdot L_{eff}) \quad (4)$$

where V_{th0} is the threshold voltage for long channel transistors (0,22), α_{DIBL} is the DIBL coefficient (0.15). From this equation, an increase of L_{gate} causes an increase of V_{th}, with further delay and leakage power implications.

4.2 Target VLIW Architecture

Based on the above model, we assessed the impact of process variations on the design of a processor core and the cost for variation tolerant design techniques. We target the VLIW VEX architecture described in [4]. It defines a 32-bit clustered VLIW

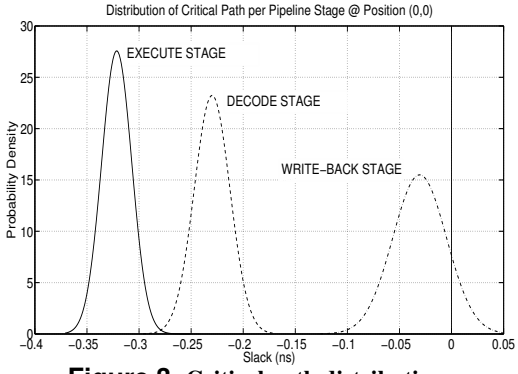

Figure 3. Critical path distribution.

ISA that is scalable and customizable to individual application domains. VEX is a clustered architecture: it provides scalability of issue width and functionality using modular execution clusters. A cluster is a collection of register files and tightly coupled set of functional units.

The target architecture was generated using the LISATek tools. It consists of 4 pipeline stages, a 32-bit data memory and a 128-bit program memory. 4 parallel slots were instantiated in the execution stage. Each slot consists of an ALU, with a shifter in series to it for shift and accumulate instructions, a compare unit checking MSB bits of ALU results, an address computation unit for loads and stores, and a multiplier in parallel with the above units. Two forwarding units were implemented to deal with read-after-write hazards. The branch unit has a static prediction on branch-not-taken and resides in the decode stage so that, in case of branch taken, only two instructions are flushed. The design was fully synthesized, even the register file, since a further optimization through full-custom design techniques was outside the scope of this paper. Finally, all memory devices (data and program memory, instruction and data caches) were modelled at behavioral level with single cycle access time. Hence, in the analysis that follows, performance of the VEX processor core will not be memory-limited. In practice, this can be achieved through pipelining of memory accesses. The VEX C compiler was available, deploying trace scheduling as its global scheduling engine.

The VEX processor was synthesized on a 65nm 1V STMicroelectronics low-power technology library. The synthesis flow included logic and physical synthesis (Synopsys Physical Compiler) and was optimized for high-performance. The maximum operating frequency was 256 MHz, the critical path being in the execute stage and going through a forwarding unit (22%) and an ALU (60%). Area was 314638 um^2 with a row utilization of about 70%. Power consumption was measured through Modelsim simulation (switching activity back-annotation) and Synopsys PrimePower reports for a FIR filtering benchmark. Total power turns out to be 30.8 mW, with only 1.1% of leakage power.

	Area	Power
Register File	53%	64.13%
Execute	26.34%	16.89%
Decode	13.63%	8,57%
Write Back	0.04%	0.1%
Fetch	0,09%	0.03%
Pipe Regs	6.9%	10.28%

Table 1. Area and Power Breakdown for the VEX architecture.

4.3 Statistical Static Timing Analysis

After the VEX processor was synthesized with Physical Compiler and static timing analysis performed with PrimeTime, coarse placement as well as timing information were available through the def and the sdf files respectively.

Therefore, we could apply the process variation model directly to the gate delays of the synthesized netlist. We developed a parser of the sdf file that checks the cell position within the chip, computes effective gate length in that location (Equation 2) and modifies its delay accordingly (Equation 3). The underlying approximation is that physical parameter and delay variations are applied on a gate-by-gate basis, neglecting transistor-level details. We also neglect other parameters impacting variability such as pattern density, orientation, and sizing. However, such approximations are implicit in our micro-architecture-level approach to process variations.

The sdf file with altered gate delays can then be re-imported in PrimeTime for static timing analysis, in order to assess how path delays have been affected by process variations or which paths are more sensitive to them. We leveraged this framework for variability injection and analysis to perform a Monte Carlo analysis, returning the statistical distribution of the design critical path. In particular, the critical path distribution for each pipeline stage was derived. Experimental data from the Monte Carlo analysis were then fitted to a normal distribution through a χ^2 goodness-of-fit test with a confidence level of 95%.

The distribution of critical path delays for a VEX processor placed in the lower-left corner of the chip (point A, worst-case systematic variation scenario) is illustrated in Fig.3. Only the execute (EX), decode (DC) and write-back (WB) stages were analyzed, since the lack of memory implementation does not allow useful insights into the fetch stage. The vertical line denotes the slack-met condition in a process variation-free scenario. As can be observed, all pipeline stages violate the timing constraint (negative slack in the plot). Similarly to the modeling assumptions in [12], the global critical path is almost always in the execute stage (neglecting distribution tails). Considering the 3σ value of the execute stage critical path delay distribution (0.0435ns), in the worst case clock frequency of the processor core is degraded by 10% with respect to the nominal value of 256 MHz.

Fig.3 also shows a lower variance for the execute stage max. delay distribution. In contrast, the write-back stage features the largest variance. We associated this result with some implementation characteristics of each stage, which confirm the modeling assumptions of previous work. First, we observed that the execute stage features a higher number of signal paths in a short time window close to the critical path delay. As a consequence, the maximum delay is likely to be determined by an outlier (see also [13]). For the same reason, the mean value of the execute stage distribution has also incurred the largest deviation from the nominal value of the execute critical path. Second, since path delays are determined by taking an aggregate sum of each gate's delay in the path, the path's ratio of variance to mean will decrease as the logic depth increases.

4.4 Process variation scenarios and SSTA-driven architecture tuning

The maximum performance degradation from Fig.3 could be used to tune additional timing margins at design time to account for the risk of process variations. This would translate into at least a 10% reduction of VEX processor frequency. However, when performance is a key design metric, a more radical approach needs to be taken.

The approach of this paper consists of identifying a few possible relevant violation scenarios of performance constraints at design time, and then of taking the proper course of action for each of them during post-silicon testing. The process variation compensation techniques we want to develop should be tailored to each scenario and not over-designed for the worst-case scenario. In addition, such scenario-driven tailoring process of compensation techniques should be performed at design time and validated through statistical static timing analysis. This avoids costly online decision logic to be integrated into the design. Our approach only involves online detection of the timing violation scenario through timing sensing circuits.

We move from the observation that as we place the VEX processor core from the lower left corner of the chip to the upper right one along the diagonal, process variation induced timing violations become progressively less severe. In point B of the floorplan (Fig.2), only the execute and the decode stages violate timing constraints, while the WB-related delay distribution is within the nominal slack-met condition. In point C, only the execute distribution

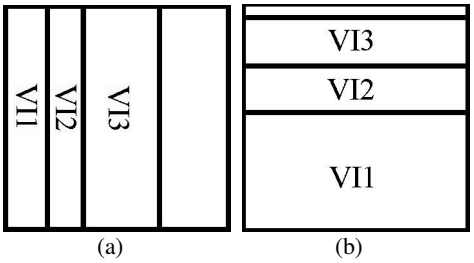

Figure 4. Voltage island generation through (a) vertical slicing and (b) horizontal slicing.

is outside nominal bounds, while from point D on nominal performance of the processor can be guaranteed. As a consequence, we identify *3 timing violation scenarios*, based on the number of pipeline stages that do not meet the nominal slack-met condition. For each scenario, we compensate the worst case timing violation.

In order to detect which scenario actually occurs after manufacturing, we need timing sensing circuits in each pipeline stage. For this purpose, we could even reuse the razor flip-flops presented in [14]: they sample the input signal of a flip-flop twice, with the nominal clock and with its delayed version, thus detecting whether a delayed signal transition took place and eventually asserting an error flag. Interestingly, we do not need to replace each pipeline register with razor flip-flops. In fact, the Monte Carlo analysis and the use of Synopsys PrimeTime for it provided us with the number of signal paths that can become critical under process variations for each pipeline stage. For the execute stage and a processor core placed in point A of the chip, we had 12 signal paths becoming critical with a probability roughly proportional to their positive slack under nominal conditions. Therefore, we need to place razor-based sensing circuits only on the flip-flops fed by these signal paths, thus significantly reducing the overhead. The same considerations apply to the other pipeline stages.

4.5 Voltage island generation

The post-silicon compensation technique we use is an increase of the supply voltage for a selective subset of logic cells, so to compensate parameter variation induced delays with the performance speed-up of a proper portion of the circuit. We target voltage island (VI) technology for this purpose.

For the generation of voltage islands, we aim at minimum perturbation of the input placement. This latter should be modified at least for level-shifter insertion. We want to avoid other sources of placement modifications such as the grouping into the same VI of cells that are logically inter-related (e.g., they belong to the same functional unit or to the same signal path) but are placed far apart in the input placement. This is actually our case, since in the input netlist the performance-driven placement optimization has led to a distribution and interleaving across the floorplan of cells belonging to different pipeline stages. Even the minimum mismatch between logic-level grouping and physical proximity may cause large wirelength and delay penalties. As a consequence, we base our VI generation strategies only on physical proximity. This approach is challenging for the global speed-up achieved by a given placement-aware cell grouping in an high-Vdd area is hardly predictable since the involved logic structures are not clearly visible at this level. Probably, the best approach consists of placement-aware cell grouping driven by the knowledge of logic structure distribution across the floorplan, but this exploration is left for future work. Even in our conservative assumptions, we prove the effectiveness of our process variation tolerant design methodology.

We explored two placement-aware VI generation algorithms. One involves horizontal slicing of the floorplan, while the other one involves vertical slicing. These two approaches are the simplest ones that facilitate the synthesis of power supply networks with minimum impact on the results of the previous placement run. We use a greedy algorithm for both horizontal and vertical VIs definition.

Based on cell density considerations, we assess the most promising side of the processor core floorplan (upper, lower, left or right) to start selecting candidate cells for high-Vdd. We then progressively extend the slice till the achieved performance speed-up is enough to compensate the less severe timing violation scenario (i.e., only the EX stage does not meet nominal performance constraints). This identifies the first voltage island. Then, we build a second island incrementally from the first, covering enough further cells to compensate the next timing violation scenario in order of severity. Finally, a third voltage island will be incrementally derived, so that even a core located in point A (i.e., all pipeline stages violating the constraints) can be brought back within nominal performance. The resulting voltage islands are illustrated in Fig.4(a) and 4(b). As a consequence, 3 voltage islands need to be operated at high-Vdd in the worst case timing violation scenario, and only 1 VI in the best case.

4.6 Level-shifter insertion

We inferred slice-shaped voltage islands through the multi-voltage design support of Synopsys tools. Then, a command in Physical Compiler automatically returns the number of level shifters required for a given VI partitioning. We retain only the nets connecting low- to high-Vdd domains as candidate for level-shifter insertion, in order to avoid the static power overhead for non-fully switched-off pMOS transistors in the high-Vdd domain. The required number of level shifters is reported in the first row of Table2. We can see that the horizontal VIs incurred a higher number of level shifters. These statistics are highly design-specific and may motivate a new VI design iteration.

Finally, we had the level shifters inserted in the design. The final placed netlist incurs a performance degradation of 8% for vertically sliced voltage islands and of 15% for horizontally sliced ones.

	Horizontal Slicing	Vertical Slicing
Number of LS	8187	6353
LS area	31.51%	26.31%
LS tot. power (point A)	0.97%	4,17%
LS tot. power (point B)	1.08%	4.93%
LS tot. power (point C)	1.14%	5.23%

Table 2. Level-shifter overhead with respect to processor area/power.

5. Experimental results

During post-silicon testing, a straightforward technique to compensate process variations consists of raising the supply voltage of the entire design. In this section, we prove the power savings that our VI-based process variation tolerant design achieves over the chip-wide supply voltage adaptation in spite of its overheads. We base our results on the dual-Vdd 65nm technology library from STMicroelectronics. 1V and 1.2V supply voltages are available. For a fair power comparison, we assumed all solutions running at the same frequency (the highest one), so to focus at first on the impact of switching activity and of leakage on total power. Performance differentiation was introduced later on.

In our validation methodology, we consider one timing violation scenario at a time. For each scenario, we know the number of VIs (and hence the individual cells) to be switched to high-Vdd after manufacturing. We compare the power consumed in this case with that incurred by a design entirely powered at 1.2V. In all cases, we ensure that nominal performance constraints are met. Power characterization was performed in three steps: gate-level delay tuning in the input *sdf* file accounting for the process variation model and for the gates working at high-Vdd, HDL simulation with switching activity back-annotation and Synopsys PrimePower power reports. A FIR filtering benchmark executed on the VEX processor core was used for power assessment. Dynamic and static power values were then increased by the contribution of level-shifters.

Total power consumption of process variation tolerant schemes for each violation scenario is reported in Fig.5. First, we note that the vertical VI slicing is more power-efficient for this design in all cases. Second, we see that as the timing violation scenario becomes less severe (and hence less voltage islands need to be operated at high-Vdd), power savings of VI-based designs with

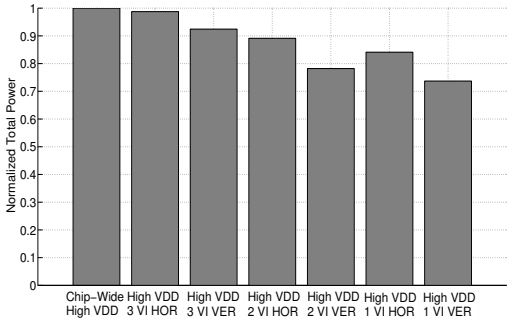

Figure 5. Total power results for different timing violation scenarios.

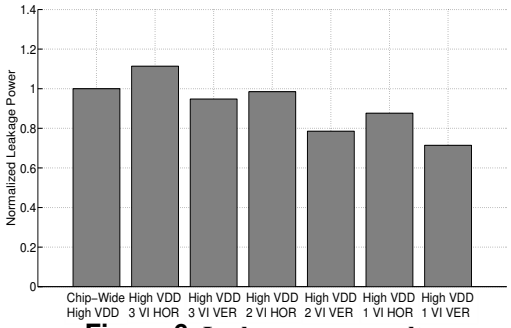

Figure 6. Leakage power results.

respect to chip-wide high-Vdd power supply become significant. Considering vertical slicing, such savings range from 8% (core placed in point A in the map of Fig.2) to 27% (core in point C).

The contribution of level-shifters to total power is illustrated in Table2. We can see that their power overhead is limited to at most 5% of total processor power.

Since our technology libraries are optimized for low power, leakage power is not a major concern. In all cases, leakage power affects total power by no more than 1.6%. However, it is interesting to check whether the high impact of level shifters on logic (not physical) area (see Table2) translates into a higher leakage power of our VI-based designs with respect to level-shifter free chip-wide high-Vdd designs. Results are illustrated in Fig.6. It can be observed that even in the worst case variation scenario (core in point A), vertical slicing incurs less leakage power than the high-Vdd chip design. This means that for the vertical slicing technique the leakage increment associated with level shifters is not as much as the increase in leakage for cells whose supply voltage is switched from 1V to 1.2V. However, the unacceptable results of the horizontal slicing scheme stress the importance of accurate validation methodologies like the one presented in this paper to assess level shifter overhead.

Finally, if we consider that VI-based designs run 8% slower than level-shifter-free designs, we would have an even lower total power of VI-based solutions. Since on the other hand we would have a correspondent increase in their execution times, the energy ratios between the different solutions would be similar to the power ratios presented above. This was confirmed via experimental results.

6. Conclusions

In this work we propose to compensate process variation induced delays during post-silicon testing by raising the supply voltage of a proper number of VIs depending on the severity of timing violations. The validation of the VI-based design methodology on a VLIW architecture for a 65 nm technology node demonstrated significant power savings with respect to chip-wide supply voltage adaptation. Future work includes the exploration of further cell grouping strategies.

7. REFERENCES

[1] A.Tiwari, S.R.Sarangi, J.Torrellas, "ReCycle: Pipeline Adaptation to Tolerate Process Variation", ISCA 2007, pp.323-334, June 2007.

[2] T.Karnik, S.Borkar, V.De, "Probabilistic and Variation-Tolerant Design: Key to Continued Moore's Law", TAU Workshop, 2004.

[3] International Technology Roadmap for Semiconductors, "Lithography", 2006.

[4] J.A.Fisher, P.Foraboschi, C.Young, "Embedded Computing. A VLIW Approach To Architecture, Compilers and Tools", Morgan Kaufmann, San Francisco, 2005.

[5] S.Bhunia, S.Mukhopadhyay, K.Roy, "Process Variations and Process-Tolerant Design", Int'l Conf. on VLSI Design, pp.699-704, 2007.

[6] J.A.Blome, S.Feng, S.Gupta, S.Mahlke, "Online Timing Analysis for Wearout Detection", 2nd Workshop on Architectural Reliability, Dec. 2006.

[7] E.Humenay, D.Tarjan, K.Skadron, "Impact of process variations on multicore performance symmetry", DATE 2007: pp.1653-1658.

[8] S.Borkar, T.Karnik, V.De, "Design and Reliability Challenges in Nanometer Technologies", Proc. of the 41st Annual Int'l Conf. on Design and Automation, 2004.

[9] J.Cain, "Characterization of Spatial Variability in Photolithography", M.S. Thesis, University of California, Berkeley EECS Dept., November 2002.

[10] S.Ghosh, S.Bhunia, K.Roy, "A New Paradigm for Low-Power, Variation-Tolerant Circuit Synthesis Using Critical Path Isolation", Int'l Conf. on Computer-Aided Design, pp.619-624, 2006.

[11] X.Liang, D.Brooks, "Mitigating the impact of process variations on CPU register file and execution units", Int.Symposium on Microarchitecture, December 2006.

[12] S.Lee, S.Das, T.Pham, T.Austin, D.Blaauw, T.Mudge, "Reducing Pipeline Energy Demands with Local DVS and Dynamic Retiming" Int. Symposium on Low Power Electronics and Design, pp.319-324, August 2004.

[13] E.Humenay, D.Tarjan, K.Skadron, "Impact of Parameter Variations on Multi-Core Chips", Int.Workshop on Architectural Support for Gigascale Integration, 2006.

[14] D.Ernst et al., "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation", Int. Symp. on Microarchitecture, page 7, Dec. 2003.

[15] H.Chang, S.S.Sapatnekar, "Statistical Timing Analysis Considering Spatial Correlations Using a Single PERT-like Traversal", Proc. Int'l Conf. Computer-Aided Design, pp.621-625, 2003.

[16] L.Zhang, W.Chen, Y.Hu, J.A.Gubner, C.C.P.Chen, "Statistical Timing Analysis with Extended Pseudo-Canonical Timing Model", Design Automation and Test in Europe, pp.952-957, March 2005.

[17] J.Tschanz et al., "Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage", IEEE Journal of Solid-State Circuits, vol.37, no. 11, November 2002, pp.1396-1402.

[18] J.Tschanz et al., "Effectiveness of Adaptive Supply Voltage and Body Bias for Reducing Impact of Parameter Variations in Low Power and High Performance Microprocessors", IEEE Journal of Solid-State Circuits, vol.38, no.5, May 2003, pp.826-829.

[19] S.Das, D.Roberts, S.Lee, S.Pant, D.Blaauw, T.Austin, K.Flautner, T.Mudge, "A Self-Tuning DVS Processor Using Delay-Error Detection and Correction", IEEE Journal of Solid-State Circuits, vol.41, no. 4, pp.792-804, April 2006.

[20] M.Orshansky, L.Milor, P.Chen, K.Keutzer, C.Hu, "Impact of Spatial Intrachip Gate Length Variability on the Performance of High-Speed Digital Circuits", IEEE Trans. on Computer-Aided Design of ICs and Systems, vol.21, no.5, May 2002.

[21] P.Friedberg, Y.Cao, J.Cain, R.Wang, J.Rabaey, C.Spanos, "Modeling Within-Die Spatial Correlation Effects for Process-Design Co-optimization", Proc. Int'l Symp. on Quality Electronic Design, pp.516-521, March 205.

[22] A.Datta et al., "Delay Modeling and Statistical Design of Pipelined Circuit under Process Variations", IEEE TCAD, pp.2427-2436, 2006.

[23] K.A.Bowman, S.G.Duvall, J.M.Meindl, "Impact of Die-to-Die and Within-Die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration", IEEE Journal of Solid-State Circuits,vol.37, no.2, February 2002.

[24] Synopsys PrimeTime VX Application Note, Implementation Methodology with Variation-Aware Timing Analysis, Version 1.0, May 2007.

[25] D.Marculescu, E.Talpes, "Variability and Energy Awareness: a Microarchitecture-Level Perspective", DAC 2005, pp.11-16, 2005.

[26] Liangpeng Guo, Yici Cai, Qiang Zhou, Xianlong Hong, "Logic and Layout Aware Voltage Island Generation for Low Power Design", EDA Lab, Department of Computer Science and Technology, Tsinghua University, Beijing, China

[27] H.Wu, I.Liu, M.Wong, and Y.Wang, "Post-Placement Voltage Island Generation under Performance Requirements", Int'l Conf. on Computer-Aided Design, pp.309-316, 2005.

[28] H.Wu, M.D.F.Wong, "Improving Voltage Assignment by Outlier Detection and Incremental Placement", DAC 2007, pp.459-464, 2007.

[29] D.E.Lackey, P.S.Zuchowski, T.R.Bednar, D.W.Stout, S.W.Gould, J.M.Cohn, "Managing Power and Performance for System-on-Chip Designs Using Voltage Islands", Int'l Conf. on Computer Aided Design, Nov.2002, pp.195-202.

[30] Y. Cao, L. T. Clark, "Mapping Statistical Process Variations Toward Circuit Performance Variability: An Analytical Modeling Approach", Proc.42nd DAC, 2005

[31] Synopsys Inc., "Galaxy Design Platform Multi Voltage Design", available online.

[32] Cadence Inc., Cadence/TSMC Reference Flow 6.0.

[33] J.Hu, Y.Shin, N.Dhanwada, R.Marculescu, "Architecting Voltage Islands in Core-Based System-on-a-Chip Designs", ISLPED'04, pp.180-185, 2004.