POLITECNICO DI TORINO
Repository ISTITUZIONALE

Beyond Ideal DVFS Through Ultra-Fine Grain Vdd-Hopping

Original
Beyond Ideal DVFS Through Ultra-Fine Grain Vdd-Hopping / Peluso, V., Rizzo, R.G., Calimera, A., Macii, E., Alioto, M. - In: VLSI-SoC: System-on-Chip in the Nanoscale Era – Design, Verification and ReliabilitySTAMPA. - [s.l] : Springer, 2017. - ISBN 978-3-319-67103-1. - pp. 152-172 [10.1007/978-3-319-67104-8_8]

Availability:
This version is available at: 11583/2681678 since: 2020-02-25T13:23:17Z

Publisher:
Springer

Published
DOI:10.1007/978-3-319-67104-8_8

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

26 November 2026

Beyond Ideal DVFS through Ultra-Fine Grain Vdd-Hopping

Valentino Peluso[1], Roberto G. Rizzo[1], Andrea Calimera[1], Enrico Macii[1], and Massimo Alioto[2]

[1] Department of Control and Computer Engineering, Politecnico di Torino, 10129, Torino
[2] Department of Electrical and Computer Engineering, National University of Singapore, 117583, Singapore
andrea.calimera@polito.it

Abstract. DVFS is the de-facto standard for low energy Multi-Processor SoCs. It is based on the simple, yet efficient principle of lowering the supply voltage (Vdd) to the minimum threshold that satisfies the frequency constraint (f_{clk}) required by the actual workload. An ideal-DVFS deals with the availability of on-chip high resolution voltage regulators that can deliver the supply voltage with a fine step resolution, a design option that is too costly.
While previous research focused on alternative solutions that can achieve, or at least get close to, the efficiency of ideal-DVFS while using a discrete set of supply voltages, this work introduces *Ultra-Fine Grain Vdd-Hopping* (FINE-VH), a practical methodology that brings DVFS beyond its theoretical limit.
FINE-VH leverages the working principle of Vdd-Hopping applied within-the-core by means of a layout-assisted, level-shifter free, dynamic dual-Vdd control strategy in which leakage currents are minimized through an optimal timing-driven poly-bias assignment procedure. We propose a dedicated back-end flow that guarantees design convergence with minimum area/delay overhead for a cutting-edge industrial Fully-Depleted SOI (FDSOI) CMOS technology at 28 nm.
Experimental results demonstrate FINE-VH allows substantial power savings w.r.t. coarse-grain (i) ideal-DVFS, (ii) Vdd-Hopping, (iii) Vdd-Dithering, when applied on the design of a RISC-V architecture. A quantitative analysis provides an accurate assessment of both savings and overheads while exploring different design options and different voltage settings.

1 Introduction

It is well known that power consumption is the most stringent constraint for the growth of digital System-on-Chips (SoCs) [1]. As an answer to this issue, the multi-core/many-core design paradigm was introduced as the only possible way out. Indeed, when high performance need to meet a low energy budget, the availability of multiple processing units that can be turned-ON/OFF, or just slowed

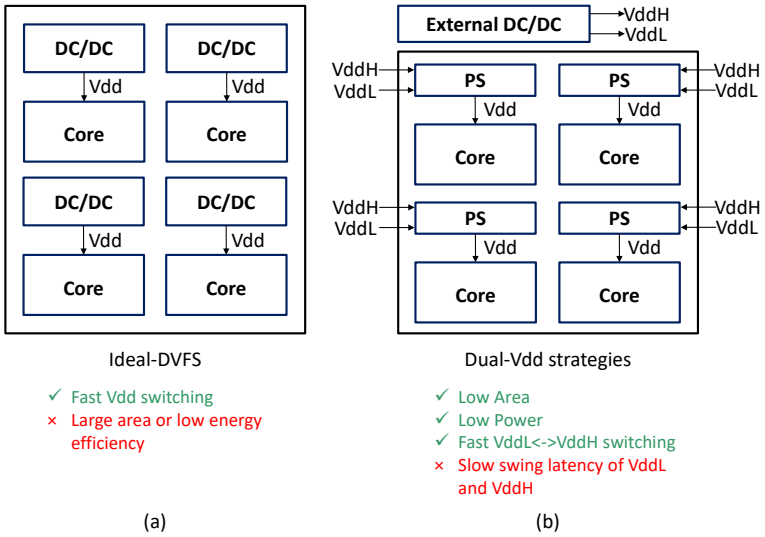

Fig. 1. Comparison between ideal-DVFS and dual-Vdd strategies

down depending on the actual workload, represents an efficient solution. Within this context, Dynamic Voltage Frequency Scaling (DVFS) has been proven to be the most effective technique to get close to minimum energy consumption. DVFS is based on a straightforward working principle, that is, reduce the supply voltage (Vdd) down to the minimum threshold that satisfies the frequency constraint (f_{clk}) imposed by the workload.

Originally applied to "monolithic" SoCs [2], the degree of freedom made available by multi-processor SoCs (MP-SoCs) architectures enabled a more efficient core-based, i.e., fine-grained, DVFS implementation [3]. With a fine-grain DVFS, each core can be set working at a different operating point in the [f_{clk}, Vdd] space; this allows to run multiple tasks asynchronously and bring down the minimum-energy point of the whole SoC. An example of fine-grain DVFS on massively parallel platforms is given in [4], where 167 processors are orchestrated over a wide frequency range achieving minimum power consumptions, from 1.07GHz - 47.5mW at 1.2V to 66MHz - 608μW at 0.675V. As an add-on feature, fine-grain DVFS is a perfect knob to compensate and/or mitigate variations due to Process, Voltage and Temperature (PVT) fluctuations that affect different cores after fabrication and during the lifetime of the circuit [5].

A practical use of DVFS on MP-SoCs deals with the availability of programmable on-chip Vdd regulators that can deliver the supply voltage with fine resolution step and fast swing (Fig. 1-a). Unfortunately, the use of integrated DC/DC converters is made impractical due to high implementation costs. Indeed, on-chip DC/DC converters fabricated with today's technologies may occupy a huge silicon area due to the low integration density of the components they contain, e.g., capacitors and inductors [4]. The picture gets even more com-

plicated if one considers that each single core should be equipped with a dedicated converter.

The challenge faced by previous works is to achieve, or at least get close to, the efficiency of high-resolution DVFS, ideal-DVFS hereafter, with a discrete set of supply voltages. In their more general embodiment, discrete DVFS strategies use two Vdd levels (VddL and VddH) generated off-chip through external voltage regulators and evenly distributed across the die (Fig. 1-b). The absolute values of VddL and VddH are shifted up/down depending on the workload, while each core is fed with the proper Vdd by means of dedicated power switches (PS). Even though this design option offers a practical solution with low impact on area and power, it comes with a speed penalty due to high voltage swing latency of the external voltage regulators [6]. Nevertheless, this is an acceptable cost as the voltage scaling process typically applies at low rate.

The two most representative cases of discrete DVFS are the Vdd-Hopping [7] and the Vdd-Dithering [8]. The Vdd-Hopping is a basic scheme in which the supply voltage range is split into a discrete set of values, two ore more depending on the external voltage regulator; the proper Vdd is selected among the available values such that the frequency constraint is met. The Vdd-Dithering scheme is a more elaborated, yet precise scheme that implements a Vdd time-sharing strategy. Differently from Vdd-Hopping, the Vdd is made switching from low (VddL) to high (VddH), leading the core to an average frequency equals to the frequency constraint. As it will be shown later in the text, the power-frequency tradeoff obtained through Vdd-Dithering can be seen as a linear approximation of ideal-DVFS.

While the aforementioned techniques aim at pushing power consumption close to ideal-DVFS, this work proposes a solution that goes beyond that theoretical limit. Recalling the practical implementation described in [9], in this work we give a comprehensive parametric analysis of the main advantages brought by a Vdd-Hopping scheme applied at a ultra-fine granularity, i.e., within-the-core. Such a solution, called *FINE-VH*, represents a viable solution to achieve power consumption below ideal-DVFS by using a dual-Vdd scheme.

The implementation of a FINE-VH solution is not trivial nor even straightforward, especially when the goal is to devise a computer-aided design methodology and not a handcrafted design. Working with multiple voltages within the same functional block raises several concerns during the place&route stages, e.g., area overhead and timing closure due to layout fragmentation and standard cell displacement. Moreover, static power consumption increases due to leakage currents of logic gates driven by portions of the circuit powered at different Vdd. Considering a simple chain of two inverters, when the driven inverter is supplied at nominal Vdd, its static power increases up to 5.2x if the driver is powered at 90%Vdd, 22.1x at 80%Vdd. Notice that the use of voltage-level shifters is strictly forbidden at this level of granularity, as it would imply huge design overheads.

As an answer to these needs, we introduce a fully-integrated design flow that guarantees timing/power convergence through incremental re-synthesis stages.

In particular, an optimal poly-bias assignment strategy is used to reduce intra-domain leakage power at zero area/delay penalties.

The core used as benchmark is the *RI5CY*, a RISC-V instruction set architecture embedded in the ultra-low power multi-processor platform *PULP* [10]. The RI5CY core has been mapped onto a cutting-edge Fully-Depleted SOI (FDSOI) technology at 28 nm. We give an accurate design space exploration which quantifies different figures of merit, like power, area and delay, at different granularity, i.e., using a layout partitioned into 9, 25 and 49 tiles, and different voltage set, i.e., multiple values of ΔVdd=VddH-VddL . As will be shown later in the experimental section, FINE-VH gives substantial power savings w.r.t. state-of-the-art DVFS solutions: ideal-DVFS, Vdd-Hopping and Vdd-Dithering.

2 Previous Works

2.1 Approaching Ideal-DVFS

The need to get close to an ideal-DVFS implementation may suggest the use of programmable on-chip DC/DC converters that guarantee a fine Vdd regulation. Unfortunately this design option would imply large overheads, both in terms of area and power. As alternative solutions, Vdd-Hopping [7] and Vdd-Dithering [8] are a preferred option as they enable a good approximation of ideal-DVFS at reasonable implementation costs.

- *Vdd-Hopping:* differently from ideal-DVFS (dashed line in Fig. 2) where the supply voltage can be adjusted with a very high resolution, this method employs a discrete set of supply voltages to control the local Vdd of a functional core. Fig. 2-a gives a pictorial description of this principle. The whole supply voltage range is split into a specific set of intervals, three in the plots of Fig. 2; at a run-time, the proper supply voltage is selected in order to meet the target frequency (f_{clk}) imposed at the application level. Once f_{clk} is identified, the core is supplied by the Vdd at the right edge of the interval in which f_{clk} falls (Vdd_2 in Fig. 2-a). Within each interval, the Vdd is kept constant and the power consumptions decrease linearly with f_{clk}. When f_{clk} crosses a new interval, power scales accordingly with the new Vdd. Obviously, the power consumption obtained with Vdd-Hopping drifts from ideal-DVFS as f_{clk} approaches the left side of each interval.
- *Vdd-Dithering*: this scheme, graphically described in Fig. 2-b, implements a Vdd time-sharing scheme. Differently from Vdd-Hopping, the Vdd switches from low to high, i.e., from the left edge to the right edge of the reference interval (Vdd_1 and Vdd_2 in the example of Fig. 2-b). Given T_{high} as the time spent at high Vdd, hence high frequency f_{high}, and T_{low} as the time spent at low Vdd, hence low frequency f_{low}, the core operates at an average frequency (f_{avg}) which is proportional to the "switching ratio":

$$f_{\text{avg}} \propto \frac{(f_{\text{low}} \cdot T_{\text{low}}) + (f_{\text{high}} \cdot T_{\text{high}})}{T_{\text{low}} + T_{\text{high}}} \quad . \tag{1}$$

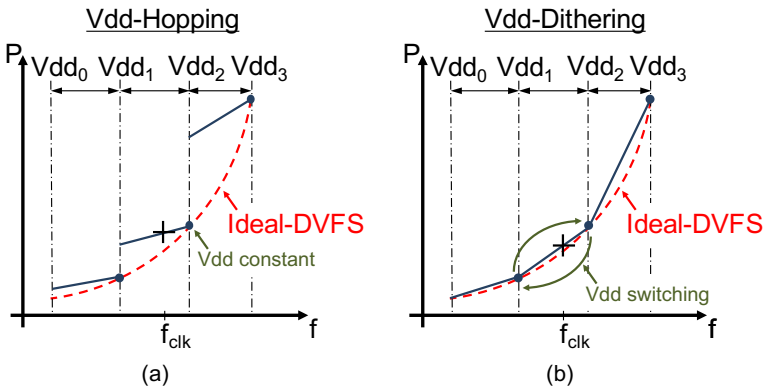

Fig. 2. Frequency-Power tradeoff of existing DVFS strategies

Fixing the proper ratio between T_{high} and T_{low} allows the core to operate at an average frequency centered on the target frequency f_{clk}. Apart from some overhead introduced by the non-ideality of power switches, experimental results in [8] have shown physical implementations of Vdd-Dithering well fit the trend line depicted in Fig. 2-b.

2.2 Within the core power management

The target of this work is to bring the Vdd-Hopping concept at a lower level of granularity, i.e., within the core. The basic concept behind this idea is not new as it follows the natural scaling other power-management strategies experienced in the last years, e.g., Multi-Vdd, Body-Biasing, Power-Gating [11–13]. The taxonomy tree shown in Fig. 3 gives a compact classification of the many granularity options.

The low-power techniques were originally applied at the architectural level where the grain was a single *functional block* (Fig. 3-a). Examples of this functional-based strategy are given in [14, 15], where the power domains are defined as the boundaries of the micro-architectural units of a microprocessor. In [14] authors introduce a voltage interpolation scheme inspired by Vdd-Dithering where the pipeline stages are dynamically fed by VddH and VddL in a time-shared manner; combined with a variable latency mechanism, this technique offers a viable solution to compensate process-variations while achieving minimum energy consumption. As an extension, the authors of [15] improve the voltage interpolation mechanism by means of a dynamic local error detection&correction scheme for protection against timing violations.

While playing with functional blocks is an intuitive solution, the amount of power savings is limited by the coarse granularity at which the low-power knobs operate. It is well known that all the low-power techniques exploit a natural characteristic of digital circuits, that is, idleness; when a functional block is not used, or it can operate at a lower speed, its power can be reduced without affecting performance. This implies the identification of idle functional blocks

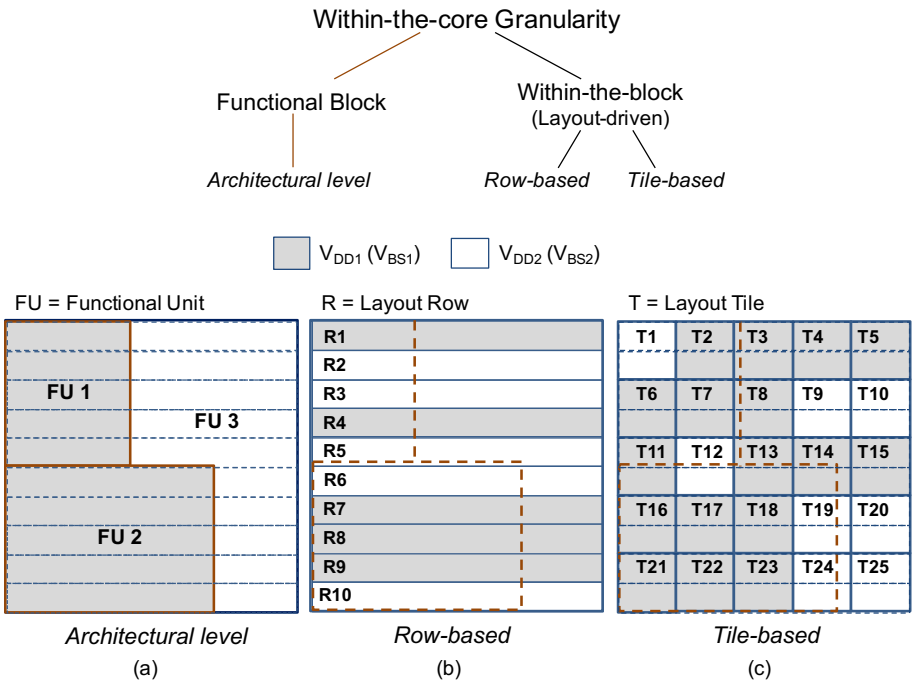

Fig. 3. Taxonomy of low-power knobs granularity. From coarse-grained architectural level (a), up to finer granularity as row-based (b) and tile-based (c) layout organization

and active ones. However, also within each single block large portion of the die are taken by non-critical components (e.g., standard cells belonging to fast logic paths) over which low-power knobs could effectively operate without inducing any performance penalty. Following this path, low-power techniques underwent a development process that brought them to work at a finer granularity, i.e., *within-the-block*.

Unfortunately, working at a finer granularity is not a free lunch as strict rules imposed by semi-custom row-based layouts might prevent the physical implementation. This imposes a higher level of awareness of the layout constraints and it forces hard constraints on the geometrical size of the minimum grain. The most common layout-driven solutions are *row-based* and *tile-based*.

As shown in Fig. 3-b, in row-based strategies the atomic element is a single layout row. In [16], authors describe a row-based dual-Vdd technique for PV compensation. The core of this technique is a timing-driven clustering procedure where timing critical rows are assigned to high Vdd and the remaining rows to low Vdd. A customized placement algorithm groups critical cells on adjacent rows such that leakage currents at the row interfaces are minimized. The main limitation of this work is that Vdd assignment is done at design time, i.e., statically. Within the same category, [17] introduces a row-based scheme for ultra-fine grain body-biasing. Differently from [16], the layout is partitioned into equally sized bunches of rows. Such a structure is more flexible as it enables dynamic body-biasing scheduling for post-silicon process variation compensation.

In tile-based strategies, the atomic element consists of a regular section of the layout; Fig. 3-c reports a scheme where the layout of a logic circuit is arranged in 5x5 square mesh. Within this category, the most representative example is given in [18] where a PV-aware adaptive dual-Vdd strategy is applied on a DES core partitioned into 42 square tiles. The measurements obtained on a silicon test-chip demonstrate power savings are limited to 12% (w.r.t. monolithic DVFS) due to static power overheads induced by intra-tile leakage currents. The same tile-based architecture is adopted in [19], yet with a different goal; it assigns a high Vdd to those tiles containing standard cells whose electrical behavior requires a minimum operating voltage (Vdd_{min}) larger than the majority of the other cells. This allows fault-free, low power operation even if the overall circuit is powered at Vdd< Vdd_{min}.

3 Implementing FINE-VH

3.1 Design and Optimization

As highlighted in the introduction, the main contribution of this work is the implementation of a novel design strategy, FINE-VH, that reshapes the Vdd-Hopping principle at a ultra-fine granularity. For this purpose, we devised a computer-aided design methodology that implements FINE-VH strategy by means of ultra-fine dual-Vdd. Working with multiple voltages within the same functional unit raises several concerns during the place&route stages, e.g., area overhead and timing closure, due to layout fragmentation and standard cell displacement. Moreover, static power consumption increases due to leakage currents of logic gates driven by parts of the circuit powered at different Vdd.

This section firstly introduces the layout organization and the physical design stages that implement the FINE-VH; secondly it describes an optimal poly-bias assignment technique that compensates the intra-tile leakage at no delay penalty; finally it proposes a Simulation/Emulation procedure for optimal Vdd selection.

Physical Design The proposed FINE-VH strategy resorts to a tile-based structured layout, abstract view given in Fig. 4. The core is regularly partitioned into NxN square tiles (N=3 in Fig. 4), each of them provided with dual-Vdd, i.e., low-Vdd (VddL) and high-Vdd (VddH), taken from around-the-core power-rings. The two power supply voltages are provided by external DC/DC converters and their value is fixed at the application level depending on the target frequency (referring to the example of Fig. 2, VddL=Vdd_1 and VddH=Vdd_2). Upper-metal horizontal/vertical stripes run over the core area forming five power-grids: VddH, VddL, Gnd, Vbn (n-bias), Vbp (p-bias). Notice that this scheme is compatible with adaptive back-biasing strategies (out of the scope of this work).

The layout rows are tied to the power-grids through p-type header power switches enclosed into dedicated cells, the *Vdd-MUX* cells. The power-management unit is in charge of driving those Vdd-MUXes by loading the Vdd configuration

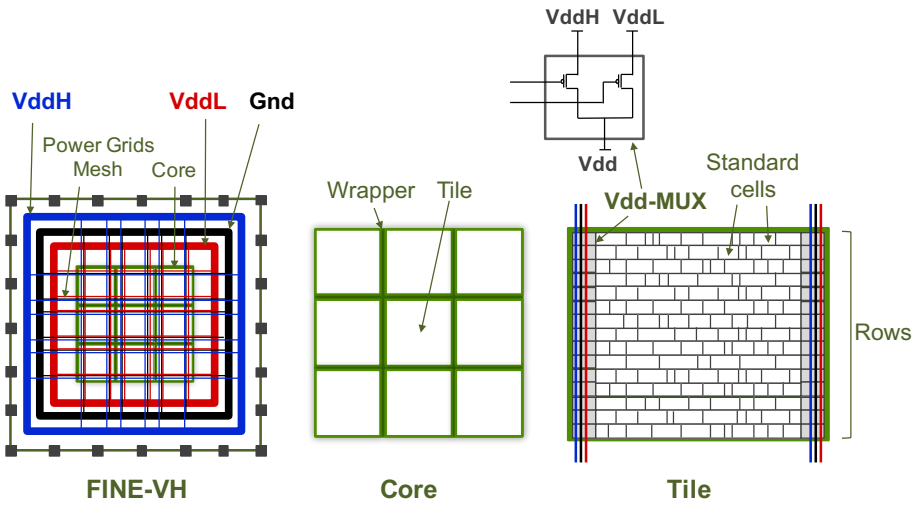

Fig. 4. Layout partitioning and tile organization

bit-stream into a dedicated flip-flop chain. The Vdd-MUXes are uniformly distributed within each tile following a row-based insertion scheme [20, 21]. The power-grids are aligned with the Vdd-MUX columns, hence, VddL and VddH area easily brought to the Vdd-MUX cells using vertical vias.

Tiles are isolated each other by a void-space wrapper that creates discontinuity in the lower-metal power rails. That's mandatory as adjacent tiles might have a different Vdd. The wrapper width is defined by the minimum metal-to-metal distance for the technology in use.

It is worth noticing that the layout partitioning follows a "no-look" style, that is, once the grain size is defined through the parameter N, the tile partitioning is done at the floorplanning stage without considering how and where the sub-functional blocks of the core will be placed. On the one hand, this might seem an overhead as functional blocks are split across multiple tiles. However, that's what allows commercial place&route tools digesting the ultra-fine granularity so as to achieve (i) a regular power planning and (ii) faster timing closure.

From a practical viewpoint, the FINE-VH flow encompasses six different stages fully integrated into a commercial design platform by *Synopsys*® by mean of dedicated TCL procedures:

1. **Synthesis**: logic synthesis using technology libraries characterized at the maximum Vdd, e.g., 1.0V for our 28 nm technology.
2. **Floorplanning**: estimation of the core area and creation of an empty layout; the latter is then automatically partitioned into NxN regular tiles using placement blockages.
3. **PG-Synthesis**: power-grids are synthesized following a regular mesh over the partitioned layout.
4. **Placement**: the Vdd-MUXes are placed at the boundaries of the tiles while standard cells are placed within the tiles so that timing constraints are satisfied.

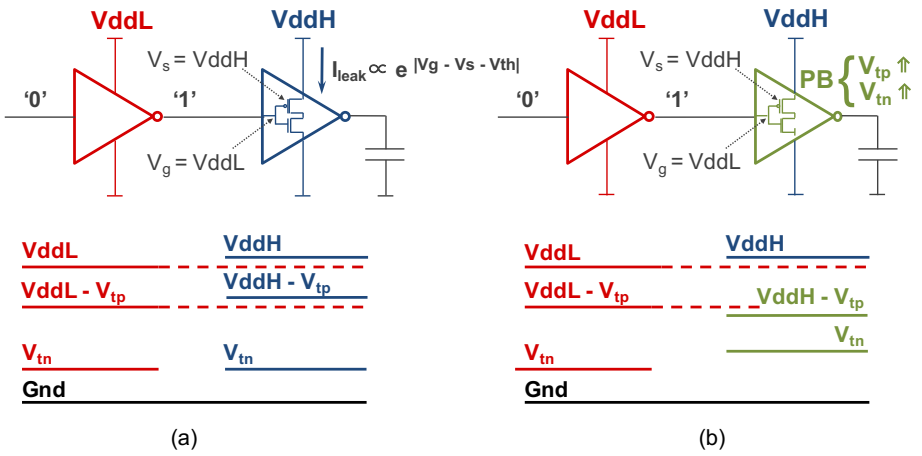

Fig. 5. Intra-tile leakage (a) and its reduction via poly-biasing (b)

5. **Post-Placement leakage optimization**: a re-synthesis stage performing optimal poly-bias assignment for those cells at the interface of the tiles (additional details in the next subsection).
6. **Routing**: a standard timing-driven routing for logic signals.

Poly-Bias optimization In a FINE-VH design, static power consumption may increase due to larger leakage currents in the "interface-cells", i.e., cells driven by signals coming from other tiles. When an interface-cell is fed with an input signal having a voltage lower than its Vdd, its internal pull-up network is partially turned-ON and leakage currents increase. This scenario is depicted in Fig. 5-a.

This over-consumption effect can be mitigated increasing the p-MOS threshold voltage (V_{th}) of the interface-cells; the latter is typically done either through gate length modulation [22] or using high-V_{th} transistors [23]. The FDSOI CMOS process, target of this work, is provided with multi-V_{th} libraries obtained through the former technique, i.e., gate length modulation, also called *poly-biasing* (PB) and represented in Fig. 5-b. For each logic gate, four different versions are available: PB0 (the standard V_{th}), PB4, PB10 and PB16 (the highest V_{th}).

Since the Vdd assignment process is done at run-time, foreseeing those cells affected by intra-tile leakage is not feasible. At the same time, a conservative approach where all the interface-cells are swapped to high-V_{th} would imply excessive delay overhead. As a compromise we introduce a timing-driven post-placement optimal poly-bias assignment which works as illustrated in Fig. 6. Starting from a placed netlist of standard V_{th} cells, i.e., PB0, the interface-cells are first identified (a) and then virtually isolated in a separated netlist with back-annotated delay information (b). Using the optimization engine embedded into the physical synthesizer, a timing-driven multi-PB assignment is run (c). The netlist returned by the multi-PB assignment step has the minimum leakage configuration, i.e., the largest set of high-V_{th} cells, that satisfies the delay

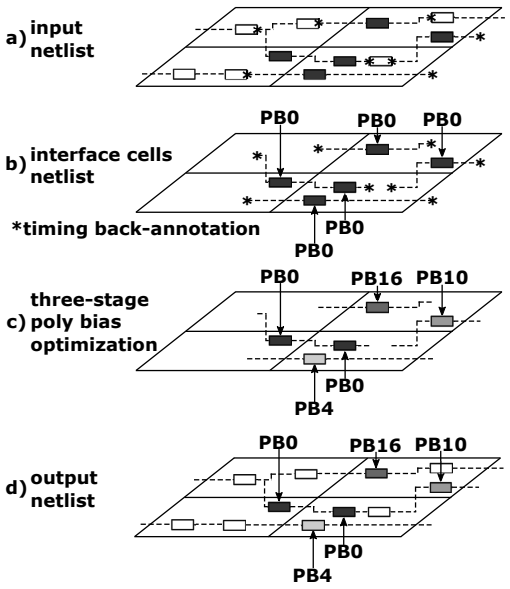

Fig. 6. PB assignment through local re-synthesis

constraints. Finally, the resulting PB assignment is annotated into the main netlist (d).

Step (c) is where the actual leakage optimization takes places. Since the goal is zero-delay penalty, only those interface cells crossed by timing paths with a timing slack greater than a certain threshold are taken into consideration. Moreover, in order to maximize the usage of cells with the highest V_{th} (PB16), we resorted to splitting the procedure into three incremental PB-assignment substages; at each substage, only a given class of PB cells is considered: at the first stage PB16, at the second stage PB10, at the third stage PB4. This procedure allows to eat the available slack first by PB16 cells (which guarantee the highest protection from intra-tile leakage), then by the PB10 cells, and finally by the PB4 cells (for which intra-tile leakage compensation is minimal).

The slack threshold used for the selection of the interface cells (i.e., those considered during PB assignment) may change depending on the actual substage, namely, depending on the class of PB cells. For instance, at the first substage (PB16 assignment), a larger slack threshold is needed as a replacement to PB16 would introduce a large delay penalty; at the third substage (PB4 assignment), the slack threshold can be relaxed. To this aim, a simple rule is followed. We first define a *slow-down factor* α_{sd} for each class of PB cells; this parameter is empirically extracted from the plot of Fig. 7 which reports the delay overhead introduced by poly-biasing: 2.82x for PB4, 8.72x for PB10, 19.88x for PB16. Then, depending on the current substage, the slack-threshold S_{th} is defined by (2):

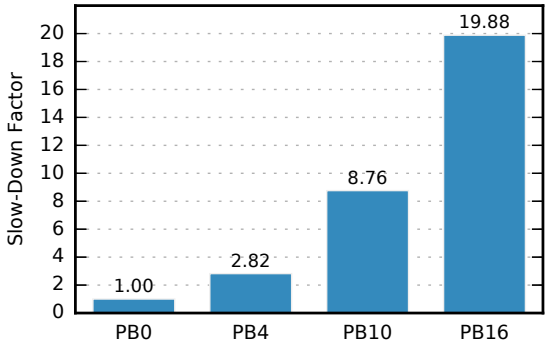

Fig. 7. Normalized average delay overhead for the different poly-bias options

$$S_{\text{th}}(PB) = d_{\text{avg}} \cdot \alpha_{\text{sd}} \ , \quad PB = [4, 10, 16] \ . \tag{2}$$

where d_{avg} represents the average delay of a cell belonging to the critical path, and α_{sd} the slow-down factor reported in Fig. 7. This enables a slack threshold that is proportional to the type of PB used at each substage.

As final remark, it is important to note that the implemented multi-stage re-synthesis is suited not only for the 28 nm FDSOI technology as it can be easily extended to all those technologies that offer multi-V_{th} libraries.

3.2 Simulation and Emulation

Commercial CAD tools lack static-analysis engines that can process level-shifter free multi-Vdd designs. Moreover, since FINE-VH does apply at run-time, the Vdd-selection policy implemented by the power-management unit needs to be emulated at design-time. For what concerns the first point, we opted for a static approach that uses off-line characterizations, while regarding the second issue, we implemented a simple, yet effective timing-driven Vdd-assignment.

Intra-Tile Leakage Power Estimation. The key issue of static-analysis of multi-Vdd designs is to estimate the intra-tile leakage power avoiding heavy SPICE simulations of the whole core. We used a static methodology that adopts off-line characterizations. For each logic gate we compiled a look-up table containing the leakage power derating factors for all possible input patterns and all possible VddL/VddH voltage configurations. As for standard timing libraries, the LUTs are obtained under different operating conditions. Having those LUTs, the static power of a single cell is estimated using the same model implemented into commercial tools:

$$P_{\text{leak}} = \sum_{i=1}^{2^n} P_i \cdot L_i \cdot k_i \tag{3}$$

Algorithm 1: Voltage Assignment Procedure

```
   Input: VddL, VddH, f_clk
   Output: Vdd assignment
 1 set_VddL([All_tiles]);
 2 while Worst_Slack < 0 do
 3 |   zero(Tile_Score[All_tiles]);
 4 |   Cell_List ← Cells ∈ Critical Path
 5 |   foreach Cell ∈ Cell_List do
 6 |   |   Cell_Delay ← propagation delay of Cell;
 7 |   |   Tile_ID ← tile hosting Cell;
 8 |   |   Tile_Score[Tile_ID] += Cell_Delay;
 9 |   end
10 |   Tile_Score ← Tile_Score[Tiles@VddL];
11 |   Tiles_Score ← sort(Tile_Score, decreasing);
12 |   Critical_Tile ← Tile_Score[0];
13 |   set_VddH(Critical_Tile);
14 end
```

where n is the number of input pins (the number of pins, for sequential cells), P_i is the input pattern probability, L_i is the nominal static power extracted from standard timing libraries, and k_i is a derating factor picked from the LUT. Notice that k_i=1 if the driver cell is placed in a tile having the same Vdd of the logic cell under analysis.

Vdd assignment. In order to emulate at design-time the Vdd-selection strategy we implemented a timing-driven Vdd-assignment whose pseudo-code is given in Algorithm 1. The algorithm applies a cell-based procedure during which tiles are sorted in terms of their timing criticality, that is, the tile containing the largest number of timing critical cells are assigned to VddH first. The procedure returns when all the cells show a positive slack.

The procedure starts by assigning all the tiles to VddL (line 1). For each tile a *criticality score* is initialized to zero (line 3). Once the most critical path is extracted, all the cells belonging to that path are stored in a dedicated list called *Cell_List* (line 4). For each cell in *Cell_List*, the propagation delay is extracted and added to the criticality score of the tile hosting the current cell (lines 5 to 8). Once all the cells in *Cell_List* have been processed, those tiles that are still at VddL are sorted according to their score (line 10, 11). The tile with the highest score (line 12) is assigned to VddH (line 13). The procedure is run until the core presents a positive path slack, that is, when the required f_{clk} is met (line 2).

4 Simulation Results

4.1 The RI5CY Benchmark

With no lack of generality, we validated the proposed FINE-VH flow on the RI5CY core, an open-source RISC-V instruction set architecture [24] used in the low-power parallel-processing platform PULP [10]. The core consists of the following units: prefetch buffer, instruction decoder, a 31x32 bit register file, integer ALU, single-cycle 32x32 integer multiplier, a control status register, hardware loop unit, debug unit, load and store unit. Figure 8 shows a layout of the die after tile partitioning.

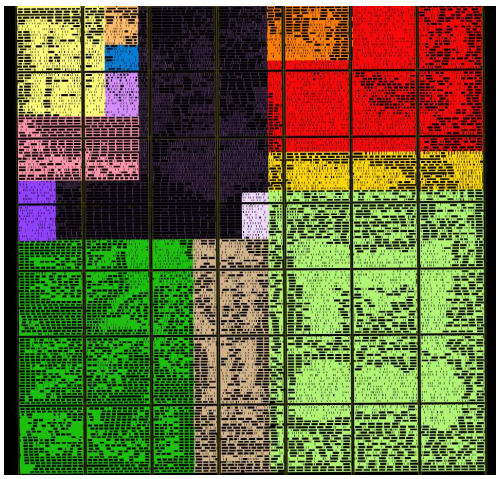

Fig. 8. A 49 tile RI5CY layout after standard-cell placement

4.2 Experimental Set-Up

Static Timing and Power analysis are performed with the STA tool by Synopsys (PrimeTime). We used technology libraries provided by the silicon vendor characterized for Vdd ranging from 0.60 V to 1.00 V (step of 50 mV) and worst-case corner (SS and 125 °C).

The four DVFS schemes used for the comparison have been set as follows.

- **Ideal-DVFS:** for each Vdd, the maximum frequency is extracted and set as the working frequency. The Vdd ranges from 0.60 V up to 1.00 V with a step of 25 mV, resulting in eight (f, Vdd) operating points; for those Vdd not available in the library set, we used a cross-library scaling feature embedded into the STA tool.

- **Vdd-Hopping:** the Vdd range [0.60 V-1.00 V] is split into a finite set of intervals having a fixed width: ΔVdd=200 mV and ΔVdd=100 mV; the resulting Vdd values are 0.60 V, 0.80 V, 1.00 V and 0.60 V, 0.70 V, 0.80 V, 0.90 V, 1.00 V respectively. The Vdd is chosen depending on the target frequency (please refer to Fig. 2).
- **Vdd-Dithering:** same Vdd ranges/intervals of the Vdd-Hopping scheme, but power consumption is a liner interpolation of points obtained through ideal-DVFS (please refer to Fig. 2).
- **FINE-VH:** the technique proposed in this work. As for the previous schemes, the Vdd ranges is [0.60 V-1.00 V]; two ΔVdd options are explored, i.e., 200 mV (Vdd range split into two intervals) and 100 mV (Vdd range split into four intervals).

For all the above design strategies the average power is extracted considering realistic switching activities of the primary inputs, i.e., annotating static probabilities and toggle rates extracted from functional simulations.

4.3 Results

Table 1 collects some key figures of the RI5CY core after FINE-VH is applied at different levels of granularity (#Tiles=1 implies no FINE-VH). Both the core area and the number of layout rows increase with granularity due to wrapper insertion around the tiles. Such a void space is used for de-cap cells insertion and intensive routing. The area overhead ranges from 2.42% for 9 tiles to 5.95% for 49 tiles. The most interesting note is that the active cell area and the nominal delay@1.00V (i.e., delay on the longest path when all the tiles are supplied at 1.00V) keep almost constant; this proves the convergence of the design flow, even at 49 tiles. As one can observe, the percentage of interface cells increases with the number of tiles, and so do the intra-tile interconnections. However, as will be shown later in the text, the intra-tile leakage overhead is controlled using the proposed poly-biasing optimization.

Table 1. Physical characteristics of the RI5CY with FINE-VH

# Tiles	**1**	**9**	**25**	**49**
Core Area µm^2	36229	37105	37626	38386
# Rows	158	160	161	163
Cell Area µm^2	25550	25397	25302	25698
Delay@1.00V ns	3.00	2.95	2.99	3.00
Interface Cells	0.0%	38.44%	56.89%	62.65%

Concerning the V_{th} distribution, Fig. 9 shows that the PB assignment strategy makes extensive use of cells at the highest V_{th} (PB16), above 57% in all the three configurations. This highlights how the leakage optimization engine

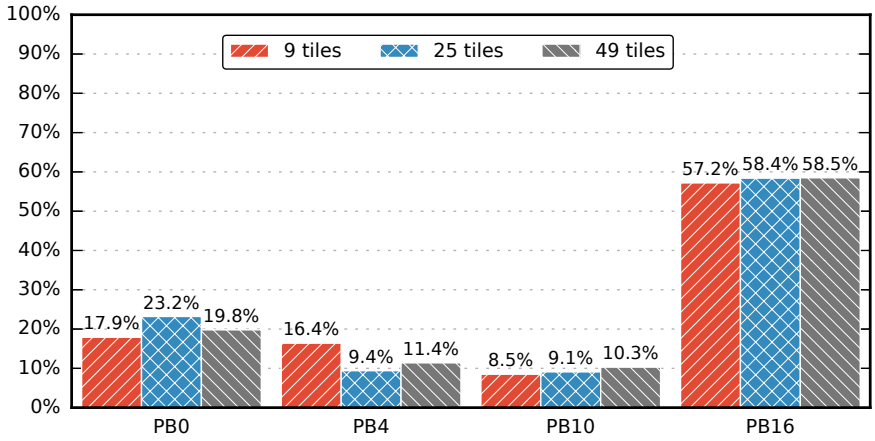

Fig. 9. Poly-bias distribution across the interface-cells

embedded into commercial tools well fits FINE-VH purposes when properly instructed.

Figure 10 shows the power vs. frequency tradeoff curves for a 49-tile FINE-VH configuration and the three state-of-the-art DVFS schemes. Numbers are normalized w.r.t. an ideal-DVFS implementation (dashed line in the plot) supplied at minimum Vdd. As expected, Vdd-Hopping and Vdd-Dithering do approximate the behavior of ideal-DVFS, even though in a different way. Within each Vdd interval the Vdd-Hopping gets worse at lower frequencies, while Vdd-Dithering always runs close to ideal-DVFS. FINE-VH outperforms the competitors for all the operating points, both for ΔVdd=200 mV and ΔVdd=100 mV. When ΔVdd = 200 mV, average power reductions of 43.5% and 27.5% are obtained with respect to Vdd-Hopping and Vdd-Dithering respectively. Moreover, FINE-VH goes quite below ideal-DVFS achieving a 23.4% of power savings. The benefits of the proposed technique are even more empathized at ΔVdd=100 mV, where the average power consumption is reduced to 35.3% w.r.t Vdd-Dithering and to 34.6% w.r.t. ideal-DVFS.

The power savings for each operating point are detailed through Fig. 11 (the plot does not show savings w.r.t. Vdd-Dithering as they are close to ideal-DVFS). For ΔVdd=200 mV savings range from 8.5% to 30.6% w.r.t. ideal-DVFS and from 30.0% to 61.0% w.r.t. Vdd-Hopping. For ΔVdd= 100 mV power savings range from 32.0% to 38.2% w.r.t. ideal-DVFS and from 33.6% to 49.7% w.r.t. Vdd-Hopping. When considering a direct comparison to Vdd-Hopping, larger savings are achieved at lower operating frequencies (left side of each Vdd interval), where a finer granularity allows to supply more portions of the layout at the low voltage. Vdd-Hopping, instead, forces the core running at a Vdd that is quite far from the optimal one.

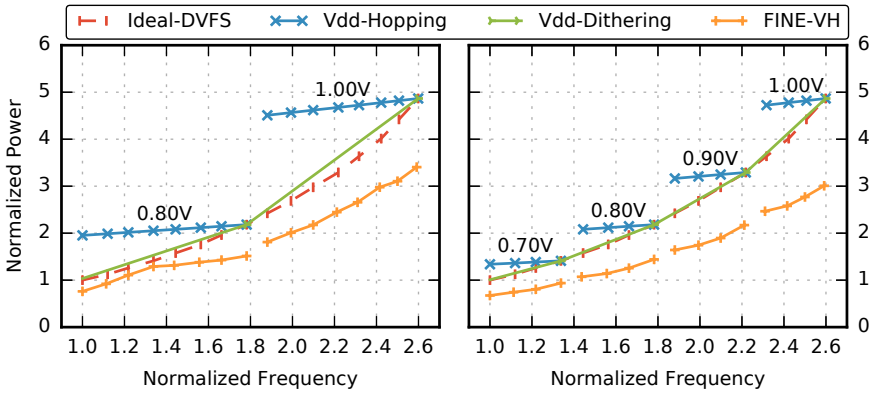

Fig. 10. Comparison of four DVFS techniques: i) ideal-DVFS, ii) Vdd-Hopping, iii) Vdd-Dithering, iv) FINE-VH (49 tiles); ΔVdd=200 mV (left), ΔVdd=100 mV (right)

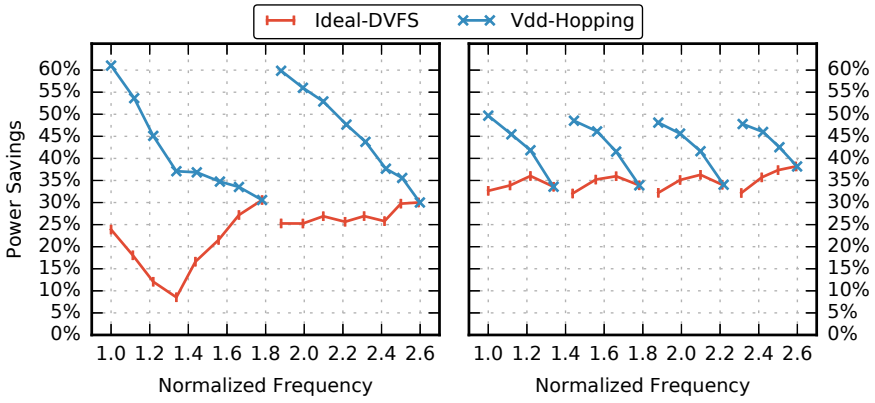

Fig. 11. Power savings of the proposed FINE-VH (49 tiles) w.r.t. ideal-DVFS and Vdd-Hopping; ΔVdd=200 mV (left), ΔVdd=100 mV (right)

Working with ΔVdd=100 mV brings larger average savings (w.r.t. ideal-DVFS) for two main reasons: *(i)* a finer voltage granularity allows a better selection of tiles that can be set at VddL, hence, it helps to get closer the global optimal; *(ii)* a smaller ΔVdd guarantees better noise margins while mitigating the effects of intra-tile leakage. This would suggest that a smaller ΔVdd is a better design option. That's true as long as power/delay overheads of external voltage regulators are neglected. Indeed, when the target frequency imposed at the application level is shifted outside the reference interval, VddL and VddH need to be set to different values (please refer to Fig. 2); this implies some extra power overheads and additional latencies due to off-chip DC/DC converters. Intuitively, a lower number of intervals, i.e., a larger ΔVdd, may allow to cover

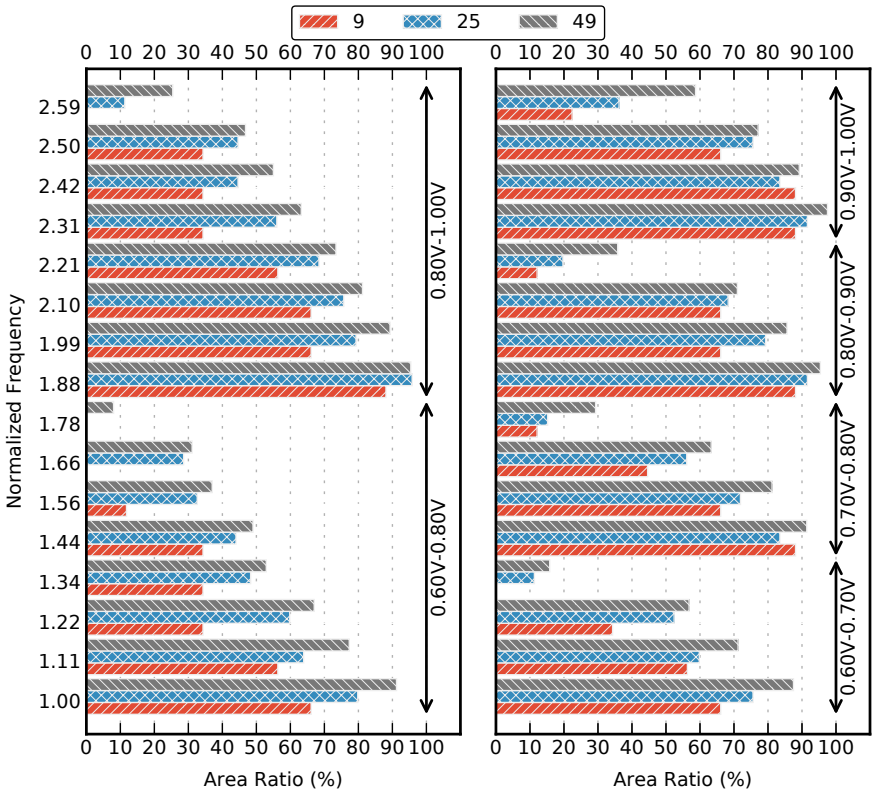

Fig. 12. Percentage of standard cell area @VddL for different number of tiles

a larger set of target frequencies with the same pair of values for VddL and VddH. In other words, a larger ΔVdd reduces the probability of an external voltage shift. The choice of an optimal ΔVdd is a tradeoff imposed by design specifications.

Figure 12 shows the percentage of the cell area supplied at low Vdd when FINE-VH is applied at different granularity, i.e., 9, 25 and 49 tiles. The plot clearly shows that 49 tiles give the best savings. For both the two ΔVdd options, working at higher frequencies decreases the amount of silicon area powered at low Vdd. For instance, at the maximum frequency, $f_{\text{clk}} = 2.59$, with 9 tiles, the percentage of area at low Vdd drastically reduces to zero for ΔVdd=200 mV, and to 2.4% for ΔVdd=100 mV. A lower granularity implies larger tiles that, most probably, will contain at least one timing critical logic path that forces the selection of a high Vdd. This issue is progressively mitigated as the granularity gets finer. As one can observe, at the maximum frequency $f_{\text{clk}} = 2.59$, with ΔVdd=200 mV, the percentage of area powered at low Vdd increases to 11.3% at 25 tiles and 25.4% at 49 tiles. Savings are even larger when ΔVdd=100 mV,

36.2% at 25 tiles and 58.7% for 49 tiles. The advantage of a finer granularity can be better appreciated considering the average over the whole frequency spectrum: for ΔVdd=200 mV, the average percentage increases to 38.5% (9 tiles), 52.0% (25 tiles), 58.9% (49 tiles); for ΔVdd=100 mV, the average percentage increases to 54.0% (9 tile), 60.7% (25 tiles), 69.2% (49 tiles). This confirms once again the rule of thumb: "the finer, the better".

Figure 13 shows the power savings (w.r.t. ideal-DVFS) achieved with the proposed PB optimization. For ΔVdd=200 mV, do not using poly-biasing nullifies all the savings brought by FINE-VH, i.e., negative savings. On the contrary, the PB assignment helps recovering the overheads due to a level-shifter free strategy as multi-V_{th} cells substantially reduce the intra-tile leakage and are intrinsically less leaky. Our PB strategy achieves average savings of 23.4%. For ΔVdd=100 mV, though the overhead imposed by intra-tile leakage currents is smaller, our PB optimization allows larger savings, 34.6% (PB) against 10.8% (no PB), without any performance penalty.

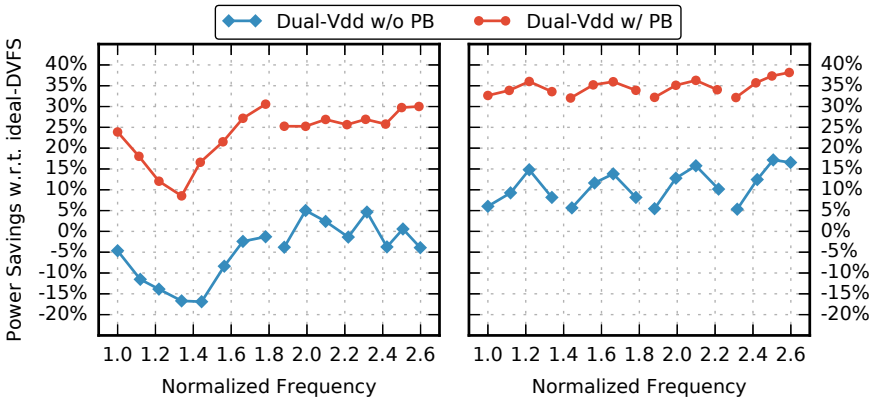

Fig. 13. Power savings with respect to ideal-DVFS before and after PB optimization for ΔVdd=100 mV and ΔVdd=200 mV (49 tiles)

As additional piece of information, Fig. 14 reports the Vdd configuration obtained by running Algorithm 1 for the case ΔVdd=100 mV. The plot shows the Vdd assignment for each tile and each operating frequency; for the sake of space we only show the 25 tiles configuration. When the frequency increases, the number of tiles supplied at low Vdd decreases. However, some tiles, like the #4 and the #5, are more critical than others as they are constantly fed by VddH; other tiles are less critical, like tile #3, #6, #16, and, even at higher frequencies, they still keep running at low Vdd. The most critical functional blocks are the arithmetic units, for which at least one tile is always supplied at high Vdd. The proof that the Vdd-assignment algorithm detects the most timing-critical tiles can be inferred by observing the regularity of the Vdd distribution within each

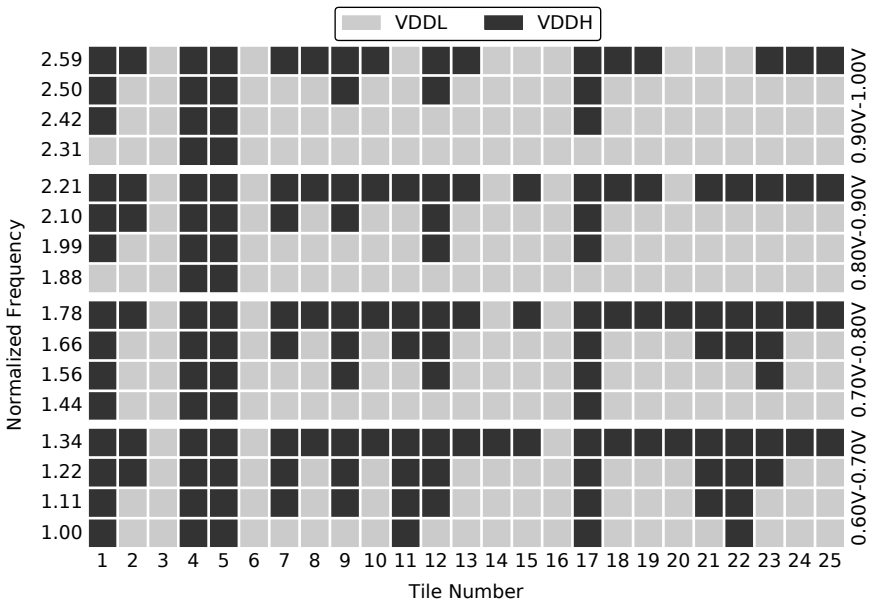

Fig. 14. Voltage Assignment (25 tiles)

individual voltage interval: once a critical tile is assigned to VddH it never swaps to VddL. This holds for tile #7, #8, #9, #10, #12.

5 Conclusions and Final Remarks

Ultra-Fine Grain Vdd-Hopping (FINE-VH) improves the efficiency of DVFS schemes in MP-SoCs. We implemented a fully automated layout-assisted, level-shifter free flow which enables tile-based Vdd-Hopping at ultra-fine granularity with minimum design overheads. The proposed technique includes a timing-driven incremental re-synthesis stage for optimal poly-biasing assignment addressing intra-tile leakage waste. The FINE-VH strategy was experimented on a RISC-V core for MP-SoC applications mapped onto a commercial 28 nm FD-SOI technology. In order to measure the benefits of the proposed technique, we devised an experimental framework capable of emulating the Vdd assignment at target clock frequency and estimating the intra-tile leakage power.

Experimental results shows that FINE-VH outperforms state-of-the-art dual-Vdd schemes and, most importantly, it goes beyond the theoretical limit imposed by ideal-DVFS. Indeed, when FINE-VH is compared with ideal-DVFS, average power savings range from 23.4% for ΔVdd=200 mV up to 34.6% for ΔVdd=100 mV.

An accurate parametric analysis clearly stated finer layout granularity enhances FINE-VH power efficiency; average layout area at low Vdd increases

from 38.5% for 9 tiles, up to 58.9% for 49 tiles (when ΔVdd =200 mV). Once again the rule of thumb "the finer, the better" is confirmed. Our quantitative analysis suggests FINE-VH can work for a wide range of ΔVdd values, from 100 mV (10%Vdd) up to 200 mV (20%Vdd). This represents an important degree of freedom as a proper selection of ΔVdd may depend on the characteristics of the off-chip voltage regulators and the actual design specifications.

References

1. V. Venkatachalam and M. Franz, "Power reduction techniques for microprocessor systems," *ACM Computing Surveys*, vol. 37, no. 3, pp. 195–237, September 2005.
2. K. J. Nowka, G. D. Carpenter, E. W. MacDonald, H. C. Ngo, B. C. Brock, K. I. Ishii, T. Y. Nguyen, and J. L. Burns, "A 32-bit powerpc system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1441–1447, November 2002.
3. T. Kolpe, A. Zhai, and S. S. Sapatnekar, "Enabling improved power management in multicore processors through clustered dvfs," in *DATE'11: Design, Automation & Test in Europe Conference & Exhibition*. IEEE, March 2011, pp. 1–6.
4. D. N. Truong, W. H. Cheng, T. Mohsenin, Z. Yu, A. T. Jacobson, G. Landge, M. J. Meeuwsen, C. Watnik, A. T. Tran, Z. Xiao, E. W. Work, J. W. Webb, P. V. Mejia, and B. M. Baas, "A 167-processor computational platform in 65 nm cmos," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1130–1144, April 2009.
5. S. Dighe, S. R. Vangal, P. A. Aseron, S. Kumar, T. Jacob, K. A. Bowman, J. Tschanz, N. Borkar, V. De, J. Howard, V. Erraguntla, and S. Borkar, "Within-die variation-aware dynamic-voltage-frequency-scaling with optimal core allocation and thread hopping for the 80-core teraflops processor," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 184–193, November 2010.
6. J. Park, D. Shin, N. Chang, and M. Pedram, "Accurate modeling and calculation of delay and energy overheads of dynamic voltage scaling in modern high-performance microprocessors," in *ISLPED'10: International Symposium on Low-Power Electronics and Design*, Aug 2010, pp. 419–424.
7. S. Miermont, P. Vivet, and M. Renaudin, "A power supply selector for energy- and area-efficient local dynamic voltage scaling," in *Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation*. Springer, 2007, vol. 4644, pp. 556–565.
8. E. Beigné, F. Clermidy, H. Lhermet, S. Miermont, Y. Thonnart, X.-T. Tran, A. Valentian, D. Varreau, P. Vivet, X. Popon, and H. Lebreton, "An asynchronous power aware and adaptive noc based circuit," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1167–1177, April 2009.
9. V. Peluso, A. Calimera, E. Macii, and M. Alioto, "Ultra-fine grain vdd-hopping for energy-efficient multi-processor socs," in *VLSI-SoC'16: International Conference on Very Large Scale Integration*. IEEE, September 2016, pp. 1–6.
10. D. Rossi, A. Pullini, I. Loi, M. Gautschi, F. K. Gürkaynak, A. Bartolini, P. Flatresse, and L. Benini, "A 60 gops/w, 1.8 v to 0.9 v body bias ulp cluster in 28 nm utbb fd-soi technology," *Solid-State Electronics*, vol. 117, pp. 170–184, March 2016.
11. L. Bolzani, A. Calimera, A. Macii, E. Macii, and M. Poncino, "Enabling concurrent clock and power gating in an industrial design flow," in *DATE'09: Design, Automation & Test in Europe Conference Exhibition*, April 2009, pp. 334–339.

12. A. Calimera, A. Macii, E. Macii, and M. Poncino, "Power-gating for leakage control and beyond," in *Circuit Design for Reliability*. Springer New York, 2015, pp. 175–205.
13. V. Tenace, S. Miryala, A. Calimera, A. Macii, E. Macii, and M. Poncino, "Row-based body-bias assignment for dynamic thermal clock-skew compensation," *Microelectronics Journal*, vol. 45, no. 5, pp. 530–538, 2014.
14. X. Liang, G. Y. Wei, and D. Brooks, "Revival: A variation-tolerant architecture using voltage interpolation and variable latency," *IEEE Micro*, vol. 29, no. 1, pp. 127–138, Jan 2009.
15. M. S. Gupta, J. A. Rivers, P. Bose, G. Y. Wei, and D. Brooks, "Tribeca: Design for pvt variations with local recovery and fine-grained adaptation," in *MICRO'09: International Symposium on Microarchitecture*, Dec 2009, pp. 435–446.
16. M. R. Kakoee and L. Benini, "Fine-grained power and body-bias control for near-threshold deep sub-micron cmos circuits," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 2, pp. 131–140, June 2011.
17. Y. Nakamura, D. Levacq, L. Xiao, T. Minakawa, T. Niiyama, M. Takamiya, and T. Sakurai, "1/5 power reduction by global optimization based on fine-grained body biasing," in *CICC'08: Custom Integrated Circuits Conference*. IEEE, September 2008, pp. 547 – 550.
18. A. Muramatsu, T. Yasufuku, M. Nomura, M. Takamiya, H. Shinohara, and T. Sakurai, "12% power reduction by within-functional-block fine-grained adaptive dual supply voltage control in logic circuits with 42 voltage domains," in *ESSCIRC'11: European Solid-State Circuits Conference*. IEEE, September 2011, pp. 191–194.
19. T. Yasufuku, K. Hirairi, Y. Pu, Y. F. Zheng, R. Takahashi, M. Sasaki, A. Muramatsu, M. Nomura, H. Shinohara, M. Takamiya, T. Sakurai, and H. Fuketa, "24% power reduction by post-fabrication dual supply voltage control of 64 voltage domains in vddmin limited ultra low voltage logic circuits," in *ISQED'12: International Symposium on Quality Electronic Design*. IEEE, March 2012, pp. 586–591.
20. P. Babighian, L. Benini, A. Macii, and E. Macii, "Post-layout leakage power minimization based on distributed sleep transistor insertion," in *ISLPED'04: International Symposium on Low Power Electronics and Design*. IEEE, August 2004, pp. 138–143.
21. A. Calimera, A. Pullini, A. V. Sathanur, L. Benini, A. Macii, E. Macii, and M. Poncino, "Design of a family of sleep transistor cells for a clustered power-gating flow in 65nm technology," in *GLSVLSI'07: Great Lakes symposium on VLSI*. ACM, 2007, pp. 501–504.
22. D. Saha, A. Chatterjee, S. Chatterjee, and C. K. Sarkar, "Row-based dual vdd assignment, for a level converter free csa design and its near-threshold operation," *Advances in Electrical Engineering*, vol. 2014, pp. 1–6, July 2014.
23. A. U. Diril, Y. S. Dhillon, A. Chatterjee, and A. D. Singh, "Level-shifter free design of low power dual supply voltage cmos circuits using dual threshold voltages," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 9, pp. 1103–1107, September 2005.
24. "Pulp: An open parallel ultra-low-power processing-platform," http://www.pulp-platform.org/.