Ultra-Fine Grain Vdd-Hopping for Energy-Efficient Multi-Processor SoCs

Valentino Peluso, Andrea Calimera, Enrico Macii and Massimo Alioto[†]
Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Italy
{valentino.peluso,andrea.calimera,enrico.macii}@polito.it
[†]National University of Singapore, 21 Lower Kent Ridge Rd, 119077 Singapore
[†]massimo.alioto@nus.edu.sg

***Abstract*—This paper introduces *Ultra-Fine Grain Vdd-Hopping* (FINE-VH), an extension of Dynamic Voltage-Frequency Scaling (DVFS) for energy efficient Multi-Processor SoCs (MPSoCs). The proposed technique leverages the working principle of Vdd-Hopping applied at ultra-fine granularity, i.e., within the core, by means of a layout-assisted, level-shifter free, dynamic dual-Vdd control strategy where leakage currents are minimized through an optimal timing-driven poly-bias assignment procedure.**

A dedicated back-end flow implementing FINE-VH has been devised such to guarantee design convergence with minimum area/delay overhead; such a tool is centered for an industrial Fully-Depleted SOI (FDSOI) CMOS technology at 28nm.

Simulation results, conducted on an open-source RISC-V core (the RI5CY core) belonging to an MPSoC platform (the PULP platform), demonstrate FINE-VH allows substantial power savings w.r.t. coarse-grain ideal-DVFS (best-case 32.6%, average 22.9%) and state-of-the-art Vdd-Hopping (best-case 60.1%, average 42.5%) and Vdd-Dithering (best-case 38.3%, average 26.8%).

I. Introduction

The "power-wall" CMOS technology hit when entered the deep sub-micron scale has been recognized as the most limiting factor for the growth of digital System-on-Chips (SoCs) [1]. With the introduction of the multi-core/many-core paradigm, new design strategies were introduced to keep the Moore's law alive; Dynamic Voltage Frequency Scaling (DVFS) is one of them.

DVFS 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. Originally applied to "monolithic" SoCs [2], the degree of freedom made available by multi-processor architectures enabled a more efficient core-based, i.e., fine-grained, DVFS implementation [3]. The activity of each core is not strictly correlated to that of parallel neighbors and each of them can be set working at a different operating point in the $[f_{clk}, Vdd]$ space. This allows to track the minimum-energy point without scarifying the overall throughput. An example of a heavy-parallel computational platform using fine-grain DVFS is given in [4], where 167 processors are orchestrated over a wide frequency range achieving minimum power consumptions. Also, the dynamic nature of DVFS makes it a perfect knob to compensate and/or mitigate, at run-time, variations due to Process, Voltage and Temperature (PVT) fluctuations [5].

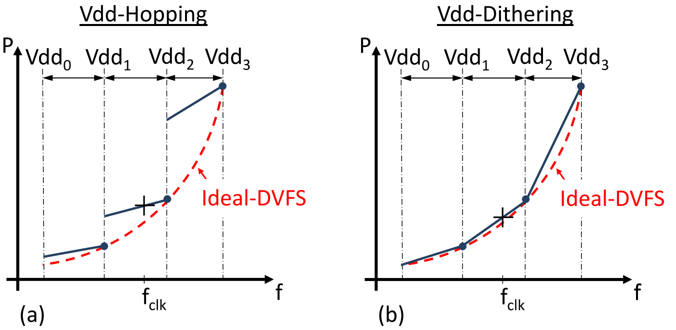

Fig. 1. Frequency-Power tradeoff of existing DVFS strategies.

Despite the many advantages, a practical use of DVFS deals with the availability of high-resolution Vdd regulators, a design option made impractical due to high implementation costs of on-chip DC/DC converters [4]. Hence, the challenge faced by previous works: achieve, or at least get close to, the efficiency of high-resolution DVFS (ideal-DVFS hereafter) using a discrete set of supply voltages.

Among the available options, Vdd-Hopping [6] and Vdd-Dithering [7] are the most representative. Differently from ideal-DVFS (dashed line in Figure 1), in a Vdd-Hopping scheme (Figure 1-a) the whole supply voltage range is split into a discrete set of intervals (three in the plots of Figure 1); once the target frequency (f_{clk}) is chosen at the application level, the proper Vdd is fed to the core, i.e., the Vdd at the right edge of the interval in which f_{clk} falls (Vdd_2 in Figure 1-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.

In order to overcome this limitation, Vdd-Dithering (Figure 1-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 Figure 1-b), leading the core to an average frequency which is proportional to the "switching ratio" (ratio between the time spent at low Vdd, Vdd_1, and high Vdd, Vdd_2); the average frequency is centered on f_{clk} using the right switching ratio. Apart from some overhead introduced by the non-ideality of Vdd regulators, experimental results in [7] have shown physical implementations of Vdd-dithering well fit the trend line depicted in Figure 1-b.

While the aforementioned techniques aim at pushing power consumption close to ideal-DVFS, this work goes beyond such a theoretical limit. More specifically, it gives a practical proof that applying the Vdd-Hopping principle at a ultra-fine granularity, i.e., within-the-core, is a viable solution to achieve power consumptions below ideal-DVFS.

This 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 consumptions increase due to leakage currents of logic gates driven by parts 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 propose 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.

It is worth emphasizing that the contribution of this work is not on the ultra-fine grain multi-Vdd concept itself, already presented in some previous work for PVT compensation [8]. Instead, we introduce a design strategy (FINE-VH) that brings the Vdd-Hopping principle a ultra-fine granularity.

The core used as benchmark is the *RI5CY*, a RISC-V instruction set architecture embedded in the ultra-low power multi-processor platform *PULP* [9]. The RI5CY core has been mapped onto a cutting-edge Fully-Depleted SOI (FDSOI) technology at 28nm. We provide a 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. As will be shown later in the experimental section, FINE-VH gives substantial power savings w.r.t. state-of-the-art DVFS solutions.

II. Previous Works on Ultra-Fine Grain Schemes

Most of the power-management techniques originally applied at the architectural level underwent a development process that brought them to work at a finer granularity: Multi-Vdd, Body-Biasing and Power-Gating are just a few examples. All of them exploit a natural characteristic of digital circuits, that is, large portions of the die are taken by non-critical components over which low-power knobs can effectively operate without incurring performance penalties. However, 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 does impose a limit to the geometrical size of the minimum grain at which the knobs can operate.

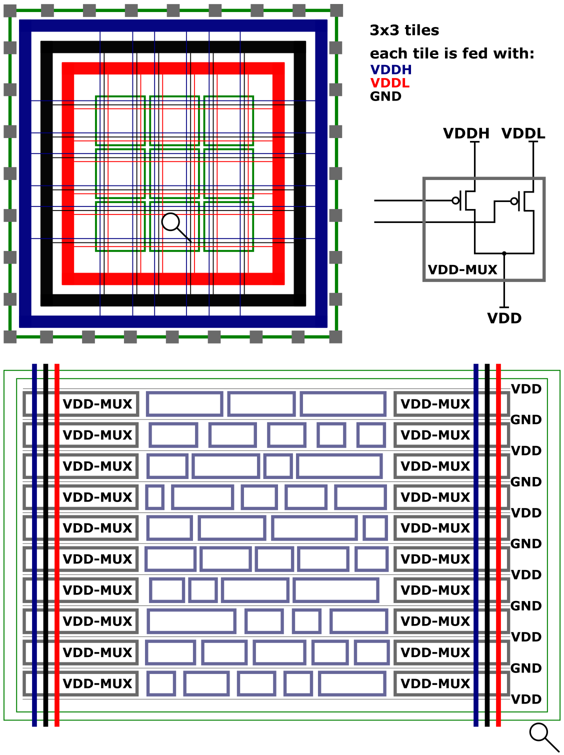

Fig. 2. Layout partitioning and tile organization.

The options proposed in the recent literature are two: *row-based* and *tile-based*.

In row-based strategies the atomic element is a single layout row. In [10], authors describe 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 is that Vdd assignment is done at design time, i.e., statically. Within the same category, [11] introduces a row-based scheme for ultra-fine grain body-biasing. Differently from [10], 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 PV compensation.

In tile-based strategies, the atomic element consists of a regular section of the layout. The most representative example is given in [8] where a PV-aware adaptive dual-Vdd strategy is applied on a DES core partitioned into 42 square tiles. Silicon measurements 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 [12], yet with a different goal, assign 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 circuit is powered at Vdd<Vdd_{min}.

In all these works, the idea of an ultra-fine grain dual-Vdd control strategy that can push DVFS beyond the theoretical limit has not been explored yet.

III. Design & Optimization

A. Layout Organization and Physical Design

The proposed FINE-VH strategy resorts to a tile-based structured layout, abstract view given in Figure 2. The core is regularly partitioned into NxN square tiles (N=3 in Figure 2), 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 Figure 1, 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 bit-stream into a dedicated flip-flop chain. The Vdd-MUXes are uniformly distributed within each tile following a row-based insertion scheme [13]. 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; 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 result into functional blocks split across multiple tiles. On the other hand, it 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, each of them implemented through dedicated TCL commands fully integrated into a commercial design platform by *Synopsys*®.

1. Synthesis: logic synthesis using technology libraries characterized at the maximum Vdd, i.e., 1.0V for our 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.

5. Post-Placement leakage opt.: 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.

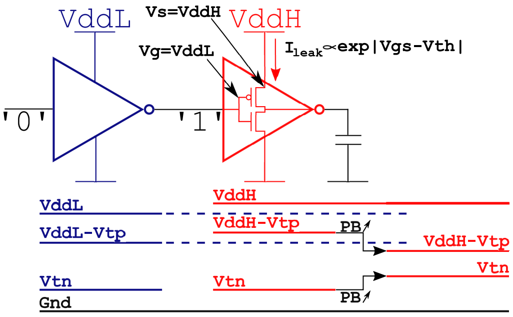

Fig. 3. Intra-tile leakage and its reduction via poly-biasing.

B. 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 Figure 3. 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 [14] or using high-V_{th} transistors [15]. 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). For each logic gate, four different versions are available: PB0 (the standard V_{th}), PB4, PB10 and PB16 (the highest V_{th}).

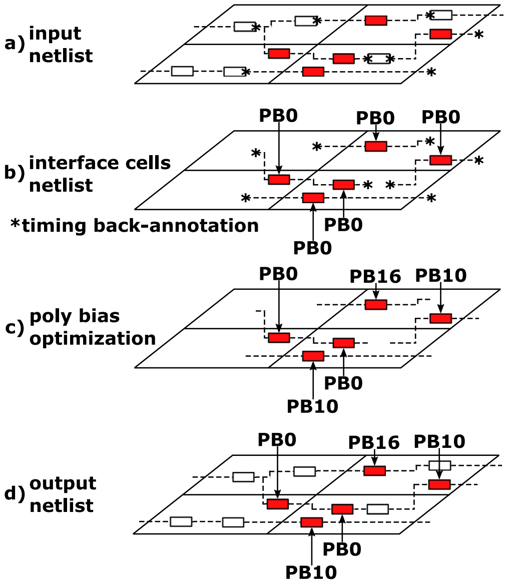

Fig. 4. PB assignment through local re-synthesis

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 Figure 4. 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, the timing-driven multi-PB assignment is run (c). The returned netlist has the minimum leakage configuration, i.e., the largest set of high-V_{th} cells, that satisfies the delay constraints. Finally, the resulting PB assignment is annotated into the main netlist (d).

IV. Simulation Results

A. The RI5CY Benchmark

We applied the proposed FINE-VH flow on the RI5CY core, an open-source RISC-V instruction set architecture [16] used in the low-power parallel-processing platform PULP [9]. 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 5 shows a layout of the die after tile partitioning.

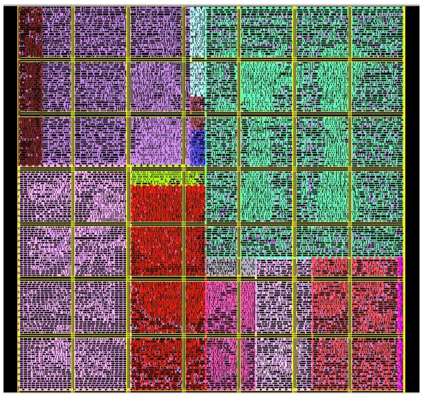

Fig. 5. A 49 tile RI5CY layout after standard-cell placement.

B. FINE-VH Simulation/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-unit needs to be emulated at design-time.

For what concerns the first point, the key issue is to estimate the intra-tile leakage power avoiding heavy SPICE simulations of the whole core. We opted for a static strategy that uses off-line characterizations. For each logic gate we compiled a derating 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:

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

where n is the number of input pins, 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.

Algorithm 1: Voltage Assignment Procedure

```
Input: $VddL$, $VddH$, $f_{clk}$
Output: Vdd assignment
1 set_VddL([All_tiles]);
2 Cell_List $\leftarrow$ Cells $\in$ tiles@VddL with slack $\leq 0$
3 while $|Cell\_List| > 0$ do
4     Cell_List $\leftarrow$ sort(Cell_List, slack, increasing);
5     Critical_Tile $\leftarrow$ tile hosting Cell_List[0];
6     set_VddH(Critical_Tile);
7     Cell_List $\leftarrow$ Cells $\in$ tiles@VddL with slack $\leq 0$
8 end
```

Regarding the second issue, we implemented a simple, yet effective Vdd-assignment heuristic. The pseudo-code is given in Algorithm 1. All the tiles are initially assigned to VddL (line 1) and those cells having a negative slack are stored in a dedicated list (line 2). The same list is then sorted in terms of timing criticality (line 4). The tile hosting the most timing critical cell is assigned to VddH (line 5, 6). The loop iterates till the cell list is empty (line 3).

C. 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; those libraries are available for Vdd ranging from 0.60V - 1.00V (step of 50mV) 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 working frequency. The Vdd ranges from 0.6V up to 1.0V with a step of 25mV; 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, from 0.6V to 1.0V, is split into a finite set of intervals having a fixed width ΔVdd=200mV; the resulting Vdd values are therefore 0.60V, 0.80V, 1.00V. The Vdd is chosen depending on the target frequency (please refer to Figure 1).

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 Figure 1).

FINE-VH - the technique proposed in this work. As for the previous schemes, the Vdd ranges from 0.6V to 1.0V; two ΔVdd options are explored, i.e., 200mV (Vdd range split into 2 intervals) and 100mV (Vdd range split into 4 intervals).

For all the cases, the average power is extracted considering realistic switching activities of the primary inputs, i.e., annotating static probabilities and toggle rates obtained through functional simulations.

TABLE I
PHYSICAL CHARACTERISTICS OF THE RI5CY WITH FINE-VH

# Tiles	**1**	**9**	**25**	**49**
Core Area µm^2	40797	42252 (+3.6%)	43589 (+6.8%)	44722 (+9.6%)
# Rows	168	171 (+1.8%)	174 (3.6%)	176 (+4.8%)
Cell Area µm^2	30565	30823 (+0.8%)	30108 (-1.5%)	30689 (+0.4%)
Delay@1.0V ns	3.77	3.83 (+1.6%)	3.67 (-2.7%)	3.88 (+2.9%)
Interface Cells	0.0%	27.4%	39.0%	45.5%

TABLE II
POLY-BIAS DISTRIBUTION ACROSS THE INTERFACE-CELLS

# Tiles	**1**	**9**	**25**	**49**
PB0	100%	12.99%	15.12%	14.60%
PB4	0%	0.79%	5.49%	4.71%
PB10	0%	15.00%	12.66%	9.25%
PB16	0%	71.23%	66.73%	71.45%

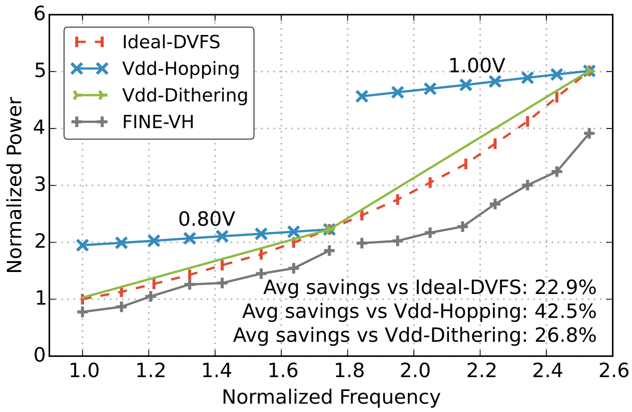

Fig. 6. Comparison of four DVFS techniques: i) ideal-DVFS, ii) Vdd-Hopping, iii) Vdd-Dithering, iv) FINE-VH (49 tiles, ΔVdd = 200mV).

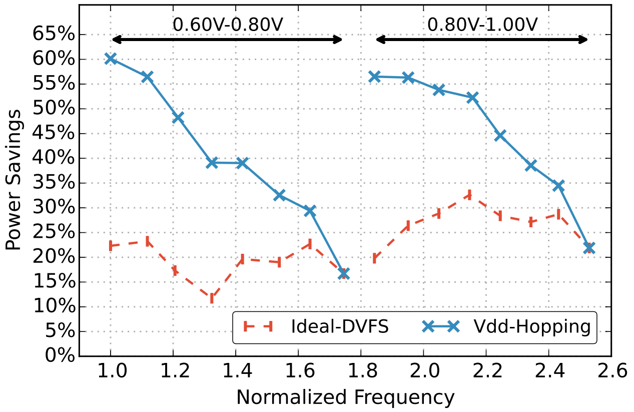

Fig. 7. Power savings of the proposed FINE-VH (49 tiles, ΔVdd = 200mV) with respect to ideal-DVFS and Vdd-Hopping.

D. Results

Tables I and II collect some key figure 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 most interesting note is that the active cell area and the nominal delay@1.0V (i.e., delay on the longest path when all the tiles are supplied at 1.0V) keep almost constant. This proves the convergence of the design flow, even at 49 tiles. Notice some negative number is the result of the heuristic nature of the optimization loops embedded into commercial tools. 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. Concerning the V_{th} distribution, the PB assignment strategy makes extensive use of cells at the highest V_{th} (PB16), more than 70% of the interface cells for both the 9 and the 49 tiles configurations. This highlights how the leakage optimization engine embedded into commercial tools well fits FINE-VH purposes.

Figure 6 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. Average power reductions of 42.5% and 26.8% are obtained with respect to Vdd-Hopping and Vdd-Dithering respectively. Moreover, FINE-VH goes quite below ideal-DVFS achieving an average power reduction of 22.9%. The power savings for each operating point are detailed through Figure 7; they range from 16.7% to 32.6% w.r.t. ideal-DVFS and from 16.7% to 60.1% w.r.t. Vdd-Hopping (the plot does not show savings w.r.t. Vdd-Dithering as they are close to ideal-DVFS). 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.

Figure 8 shows the percentage of layout supplied at low Vdd when FINE-VH is applied at different granularity, i.e., 9, 25 and 49 tiles. From the plot it is clear that 49 tiles give the best savings. The most critical functional blocks are the arithmetic units, for which at least one tile is always supplied at high Vdd. In both the two voltage intervals, i.e., [0.6 - 0.8]V and [0.8 - 1.0]V, working at higher frequencies decreases the amount of silicon area powered at low Vdd. For instance, with 9 tiles and f=1.75, the number of tiles at low Vdd falls to zero. This is due to the fact that tiles are physically pretty large, hence, all of them contain at least one timing critical logic path. In other words, the grain size is not small enough to isolate the critical portion of the circuit. The problem is progressively mitigated as the granularity gets finer; the percentage of area at low Vdd grows to 3.8% and 10.4% at 25 and 49 tiles respectively. Those results confirm once again the rule of thumb “the finer, the better”. However, there might be some exception. For instance,

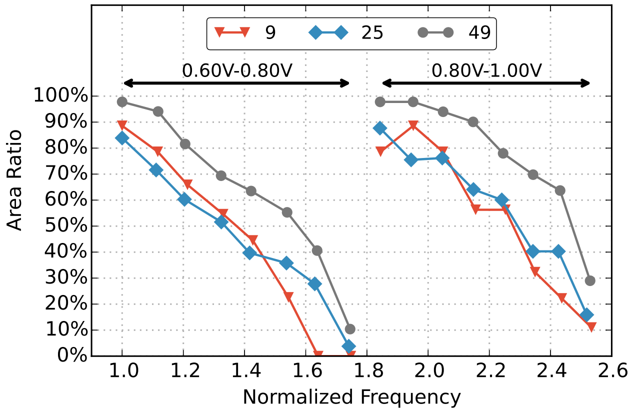

Fig. 8. Percentage of standard cell area @VddL for different number of tiles.

let's consider 9 and 25 tiles; both of them get almost the same power savings (48.7% and 52.2% on average), hence, using 9 tiles is better as it shows less design overheads. This scenario may appear depending on the type of circuit, the topological structure of the netlist and the timing-path distribution. In this context, the proposed cad flow enables a fast exploration of the design space which may help designers to weight savings against overheads. A final remarks on Figure 8 concerns the non monotonicity of the curves; this is mainly due to the heuristic nature of the Vdd selection routine which might fail to find the global optimum assignment.

Figure 9 finally shows the power savings (w.r.t. ideal-DVFS) achieved using the proposed PB optimization. As part of a parametric analysis, we run simulations for two values of ΔVdd, i.e., 200mV and 100mV. The first option (also used in the previous plots) is a good compromise between area/cost of DC/DC converters and noise margins; 100mV represents a more aggressive solution that might help to achieve better power results at the cost of more complex DC/DC converters. At ΔVdd=200mV, do not using poly-biasing would nullify all the benefits of FINE-VH (negative savings); PB assignment helps recovering this overhead as cells at high V_{th} (i) substantially reduce the intra-tile leakage and (ii) are intrinsically less leaky. At ΔVdd = 100mV, the contribution of the intra-tile leakage to the total power is smaller. Nevertheless, the PB optimization allows to double the power efficiency, moving the average savings from 12.3% to 25.4%, still at zero performance overhead.

V. Conclusions and Final Remarks

Ultra-Fine Grain Vdd-Hopping improves the efficiency of DVFS schemes in MPSoCs. A fully automated layout-assisted flow with incremental re-synthesis for Poly-Biasing was implemented which enables ultra-fine granularity at minimum design overheads. The flow was experimented on a RISC-V core for MPSoC applications mapped onto a commercial 28nm FD-SOI technology. Simulation results demonstrate average power savings of 22.9% (w.r.t. ideal-DVFS) at no performance costs. Future works will show the proposed FINE-VH is also well suited for near-/sub-threshold ICs.

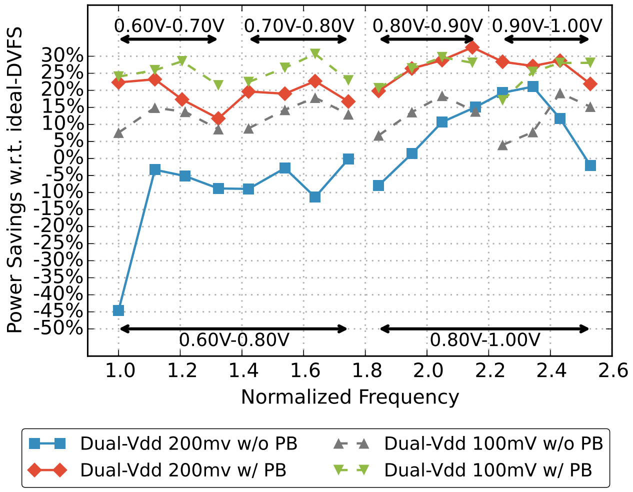

Fig. 9. Power savings with respect to ideal-DVFS before and after PB optimization for ΔVdd = 200mV and ΔVdd = 100mV (49 tiles).

References

[1] V. Venkatachalam and M. Franz, "Power reduction techniques for microprocessor systems," *ACM Computing Surveys (CSUR)*, vol. 37, no. 3, pp. 195–237, September 2005.

[2] K. J. Nowka *et al.*, "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 *et al.*, "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 *et al.*, "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] 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.

[7] E. Beigné *et al.*, "An asynchronous power aware and adaptive noc based circuit," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1167–1177, April 2009.

[8] A. Muramatsu *et al.*, "12% power reduction by within-functional-block fine-grained adaptive dual supply voltage control in logic circuits with 42 voltage domains," in *ESSCIRC'11: Proceedings of the 37th European Solid-State Circuits Conference*. IEEE, September 2011, pp. 191–194.

[9] D. Rossi *et al.*, "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.

[10] 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.

[11] Y. Nakamura *et al.*, "1/5 power reduction by global optimization based on fine-grained body biasing," in *Proceedings of the Custom Integrated Circuits Conference*. IEEE, September 2008, pp. 547 – 550.

[12] T. Yasufuku *et al.*, "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: Thirteenth International Symposium on Quality Electronic Design*. IEEE, March 2012, pp. 586–591.

[13] 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.

[14] 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.

[15] 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.

[16] "Pulp: An open parallel ultra-low-power processing-platform," http://www.pulp-platform.org/.