Managing Mismatches in Voltage Stacking with CoreUnfolding

Ehsan K. Ardestani, UC Santa Cruz
Rafael Trapani Possignolo, UC Santa Cruz
Jose Luis Briz, Universidad Zaragoza
Jose Renau, UC Santa Cruz

Categories and Subject Descriptors: [ ]:

Additional Key Words and Phrases: Voltage Stacking, Power Delivery Network, Microarchitecture

ACM Reference Format:
Limiting Mismatches in Voltage Stacking with CoreUnfolding *ACM Trans. Architec. Code Optim.* 1, 1, Article 1 (March 2015), 25 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

The increase in current draw poses growing challenges to the design and implementation of power delivery system in modern computing devices. Among these challenges are the need for more pins and pads in the package dedicated to power delivery, elevated I^2R power loss, higher IR and Ldi/dt voltage noise, and bigger, less efficient Voltage Regulators (VR).

On-package and integrated on-die voltage regulators have been proposed [Kim et al. 2012; Ramadass et al. 2010; Chang et al. 2010; Burton et al. 2014] to offer a finer temporal granularity and better response to di/dt which boosts the benefit of Dynamic Frequency and Voltage Scaling (DVFS). Nonetheless, on-chip VRs, suffer from less efficiency compared to off-chip VRs [Kim et al. 2008; Kim et al. 2012; Burton et al. 2014].

A possible remedy to the power delivery challenges is voltage stacking [Gu and Kim 2005; Rajapandian et al. 2006; Shenoy et al. 2011; Lee et al. 2012; Candan et al. 2014], in which multiple components are "stacked" in a series configuration on top of each other. The current passing through one component is recycled by the next component, reducing the total current demand. To maintain the same power delivered to each component, the stack is supplied with a higher voltage. In other words, voltage stacking delivers the same power to the logic in a different form; higher voltage but lower current. This alleviates the aforementioned challenges, *e.g.*, VRs can be designed for better efficiency delivering the same power at higher voltage and lower current [Shenoy et al. 2010; Lee et al. 2012; Kim et al. 2011].

A main challenge in voltage stacking, which we address in this paper, is load mismatch between the stack levels. Consider a configuration with two stacked cores: running different applications on each core can result in cores demanding different power. Because the same current pass through all the components in the stack, one core can restrict power delivered to the other, potentially inducing timing failure. The load mismatch can be managed by adding extra VRs [Shenoy et al. 2011]. However, this requires VRs with equal size to VR in conventional power delivery solutions to guarantee handling of worst case mismatches. Also, such configuration can potentially result in increased system wide power consumption. These complications limits the applicability of voltage stacking because the only real benefit seems to be reducing the number of power delivery pads in the package [Zhan and Sapatnekar 2008], and voltage noise [Gu and Kim 2005].

This paper proposes *CoreUnfolding*, a novel method to use voltage stacking *within* each core. Not only *CoreUnfolding* has the same advantages as voltage stacking (reduced number of power delivery pads, improved di/dt, voltage drop and noise) but also improves the system wide energy efficiency and reduces area dedicated to voltage regulators. The novelty resides in guaranteed a maximum load mismatch between the

stacks, achieved by stacking two groups of functional units within a core, clustered based on the correlation and magnitude of their power consumption. This allows an efficient system that uses a much smaller VR. The proposed microarchitecture easily scales to multicore homogeneous and heterogeneous configuration, and works with single threaded, multithreaded applications. It is also compatible with power gating techniques, since the stacking occurs within a core, while cores are not stacked and therefore can be independently turned on/off.

CoreUnfolding partitions the functional units in a core into two groups: *Header* and *Footer*. Thus, the Data Cache could be in the *Header* group, and the Reorder Buffer in the *Footer* group. The partition aims to balance the load or power consumption in the two groups under different workloads. We study the correlation of power consumption between components among functional units. We observe that the activity and power consumption among some functional units tightly correlates, which makes them good candidates for stacking.

In some infrequent cases, the mismatch between the levels of the stack could create a high load imbalance between the *Header* and *Footer* groups, which requires some sort of safeguard. We resort to dynamic load balancing techniques to match up the power consumption of *Header* and *Footer* for this uncommon case. In this paper we cap these mismatches by using two mechanisms: "Dummy Activity" and DVFS. Consequently, the additional voltage regulator (*sVR*) is only required to match as little as 20% of the total power consumption. As a result, the proposed solution allows for a much smaller total voltage regulator size down to 70% of the size in the baseline. The benefit would be higher considering the thermal implications.

Equally important is the power savings of the system. *CoreUnfolding* works with both on-chip and off-chip voltage regulators. In our experiments the VR needs to support up to 60% of the current that the baseline VR provides, which results in up to 10% overall power savings.

The main contributions of this paper are:

— The idea of applying voltage stacking within a core;
— Using power average and transient power correlation to decide which units to stack;
— Capping the maximum mismatch to allow smaller and more efficient VRs

2. RELATED WORK

Voltage stacking has been proposed as a paradigm shift to tackle the problem of increased challenges of power delivery as the technology scales further. Voltage stacking in the logic allows for operation at higher voltage and lower current for the same power budget. This does not reduce the energy consumption of the logic. However, it allows for the power delivery to operate in a more efficient way [Shenoy et al. 2010; Kim et al. 2011].

Voltage stacking can improve power delivery systems [Rajapandian et al. 2006; Gu and Kim 2005; Zhan and Sapatnekar 2008], but managing the load mismatch between the stacked devices remains a challenge. Lee *et al.* [Lee et al. 2012] observe this issue through real measurement, but do not propose a solution. Just stacking the components like [Lee et al. 2012] would result in timing failure. Zhan *et al.* [Zhan and Sapatnekar 2008] use voltage stacking to reduce the number of pins in 3D ICs. To avoid increased power consumption due to the load mismatch between stacked levels, they partition the modules in the floorplan and assign them to appropriate level, and use extra voltage regulators to manage the load mismatch. However, their method does not guarantee bounded mismatch, and the extra voltage regulators increase the area overhead. Gu *et al.* [Gu and Kim 2005] use voltage stacking for voltage noise reduction. To avoid using an extra voltage regulator, they use "Digital Voltage Regulator", which introduces extra activity to balance the load between levels. We call this "Dummy Activity". Only relying on dummy activity for the whole processor, if even possible, would incur cycle time penalty.

Voltage stacking has been proposed in the server level, for data centers [Candan et al. 2014]. To regulate the voltage for each of the stack levels, differential power converters were employed. Instead of tryint to regulate the voltages to the nominal value, this proposal uses acceptable bands of voltages. The stacking was made at the mother board level, *i.e.*, at the 12V range. With very regular workloads, they are able to improve the voltage conversion efficiency considerably.

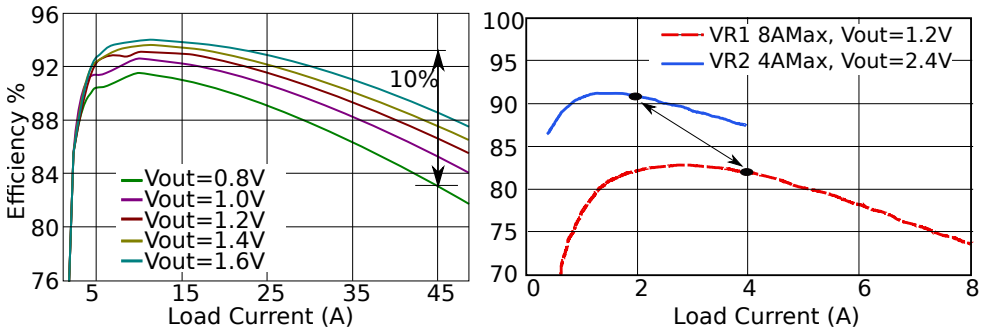
Multi-level ladder voltage converters have been proposed to be used in voltage stacked circuits [Kesarwani et al. 2013; Schaef and Stauth 2015; Lee et al. 2015]. Such topologies have the advantage of delivering multiple output voltages from a single converter, with high efficiency. In this paper, we leverage the results of a fully-integrated switched capacitor ladder DC-DC converter [Lee et al. 2015] that yields great efficiency when regulating multiple levels of voltage.

Through voltage stacking, this work proposes a framework to manage and bound the load mismatch between the stacked levels in a processor. Similar to [Zhan and Sapatnekar 2008], we use extra VRs to regulate the mismatches, but cap the demand by using dummy activity for rather uncommon high mismatches. As a safe guard, we use DVFS to scale down the power in case of extreme mismatch, so the scaled mismatch can be regulated by the small voltage regulator. Consequently, the whole system can benefit from lower current draw. This leads to system-wide power savings, better voltage margin, as well as reduced complexity and area of the voltage regulators in the system.

3. BACKGROUND

3.1. Voltage Regulators

An ideal voltage regulator (VR) delivers power from a power source to the load without any losses. Unfortunately, the regulator itself consumes power with some *conversion efficiency*, defined as the ratio of power delivered to the load by the regulator to the total power into the regulator. Regulator losses are dominated by switching power and resistive losses, which depend on the size of the switching power transistors, switching frequency, and load conditions (e.g. load current and voltage levels) [Kim et al. 2008].



(a) Dynamic knob (b) Static knob

Fig. 1: (a) For the same power output, reducing the current output of a given VR allows more efficient VRs. (b) In general, VRs are more efficient when optimized (at design time) to lower current and higher output voltage (*i.e.*, smaller step down). This is illustrated by VR1 [Inc. b] and VR2 [Inc. a] (from the same company), that are designed for the same power output, but with different efficiencies.

The benefit of higher voltage and lower current output can be considered through two knobs: static and dynamic. At runtime, drawing less current from the VR could reduce the resistive loss and lead to better efficiency, in particular for high loads. In very

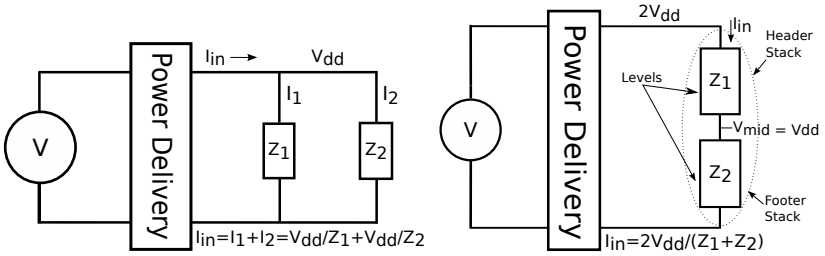
light loads the switching power loss can become dominant. Figure 1a shows the efficiency against the output current of an Intel compliant off-chip VR for desktop/server applications. Increasing the voltage from 0.8V to 1.6V, and reducing output current by half, increases the efficiency in that particular operating point by 10%.

Nonetheless, designing VR modules involves thorough optimization between a number of parameters including the switching and resistive lost. VRs designed to deliver lower current, with higher output voltage, are, in general, more efficient [Wei 2004]. We refer to this as the static knob. Figure 1b shows the efficiency for two VRs from the same manufacturer with different maximum output currents (8A and 4A) [Inc. b; a]. For the same power, it is possible to increase the efficiency by over 10% when the VR with smaller maximum output is used. Lower heat dissipation due to the higher efficiency can have secondary benefit on the package footage and cost as well.

Note that, those are not fundamental reason for this gap in efficiency. Those are empirical observations from current commercial voltage regulators. Design time optimizations, and new VR designs could change that scenario, but there is no indication of such in the near horizon. Nevertheless, from the observations made in this section, it seems plausible to consider that, by doubling the voltage and reducing the current by half, it would be possible to improve in 10% to 15% the efficiency of VRs in a real design. To be conservative, we consider that 10% improvement in efficiency is possible in the use cases found, actual gains may differ.

3.2. Voltage Stacking

Voltage stacking configures components in a series manner instead of the conventional parallel way, as shown in Figure 2. To deliver the same power, the stack is supplied with a higher voltage. In general, a stack of n components, each supplied with V_{dd} in the conventional configuration, is supplied with $n \times V_{dd}$. Charge passing through one component is recycled by the next component. Hence, the total current demand is reduced, ideally to $\frac{1}{n}$.



(a) Conventional (b) Stacked

Fig. 2: Different power delivery schemes. Both scheme deliver the same amount of power, though in different forms. For the same loads a) delivers the power in form of $V \times I$, and b) in form of $2V \times \frac{I}{2}$. The latter one could be more efficient.

Voltage stacking can increase the efficiency of power delivery subsystem, as 1) it requires the power delivery to operate at a higher voltage and lower current; 2) it requires lower step-down ratio and 3) it results in less IR drop across the power delivery. Note that the logic in the best case would consume the same amount of power as it does in conventional power delivery configuration.

While voltage stacking increases the efficiency of power delivery network, it suffers from load mismatch between stacked levels. This is because current demand (Z in Figure 2) for each component could change at runtime, for example due to changes in the workload behavior. However, the same current flows through all the levels in the stack. This results in different voltage drops across each level, where the delivery should supply each level with the same voltage. The load mismatch, and deviation of

V_{mid} from the supposed value ($\frac{2V_{dd}}{2} = V_{dd} = V_{mid}$) can be seen as voltage noise for each component, and could even result in timing failure. Managing the load mismatch and maintaining the overall benefit of the voltage stacking has remained an open problem. This paper addresses this challenge. In our experiments we focus on 2-level stacking.

Body biasing of the transistors connected to V_{mid} can introduce issues as those transistors are not connected to V_{dd} or Gnd rails. To support independent body bias for each stack level, triple well or Silicon-On-Isolator (SOI) technology must be used. Those processes provide insulation between different wells, and allow the differential body biasing for the transistors connected to V_{mid}. They are commonly offered by all the major fabs.

4. *COREUNFOLDING*

We apply voltage stacking within cores. This is based on the observation that current draw, *i.e.*, power consumption, of different functional units (FU) in a core correlates with each other. We exploit this correlation to guide the stacking. FUs with well correlating power consumption are good candidates for stacking at design time, minimizing the chance of a mismatch between the two stacked levels. We partition the FUs according to their power magnitude and correlation, and assign each partition to a level in the stack. To ensure correct functionality at runtime, we use additional dynamic techniques with minimal overhead.

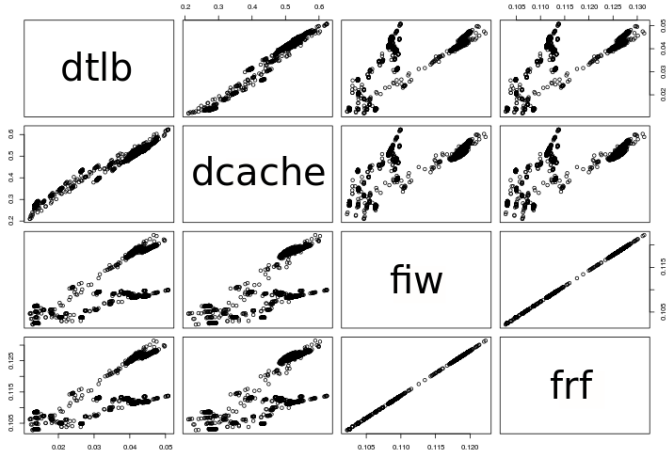


Fig. 3: Power consumption correlation matrix of functional units can guide stacking among them. In this example, fiw and frf can be stacked.

4.1. Design-Time Load Balancing Across Stacked Levels

Figure 3 shows the correlation between power consumption of four functional units in an out-of-order processor running the *applu* benchmark. Each line and column represents a FU. The intersection of FUs shows the correlation of the two FUs, points distributed closer to the diagonal indicate better correlation. It shows the correlations for data TLB ($dtlb$), data cache ($dcache$), floating point instruction window (fiw), and floating point register file (frf). In this set, $dtlb$ and $dcache$ correlate well with each other (cor=0.981). fiw and frf also correlate with each other very well (cor=0.999). However, there is no good correlation between fiw and $dcache$ (cor=0.352) or fiw and $dltb$ (cor= 0.362). This correlation among the units, in this case, suggest stacking of $dtlb$ with $dcache$, and fiw with frf. As it will become clear later in the paper, this one-on-one correlation should be seen as a presentation of concept, *CoreUnfolding* will

aggregate the power consumption of different FUs instead of comparing pairs of FUs isolated.

The magnitude of power consumption of the units should be considered in the partitioning as well. For example, even though the power consumption of $dtlb$ and $dcache$ correlates well, a stack comprising only these two units would still have load mismatch, as the average power consumption of $dcache$ is 3-4 times power consumption of $dtlb$.

For the remainder of this paper, we refer to the top level in the stack as *Header*, and the lower level as *Footer*. Figure 4 shows an example of a 2-level stacked configuration. The two partitions will be stacked one on the top of another, while the middle layers between them (V_{mid}) is shared across the partitions. This simplifies stacking of the units, as more units can contribute in matching the average and transient power consumption of the two levels. It is also simple to implement on the chip, as the middle layer between the stacked units will be treated in a similar way as other power delivery nets (*e.g.*, V_{dd}) in terms of decoupling capacitance (decap) specification and implementation.

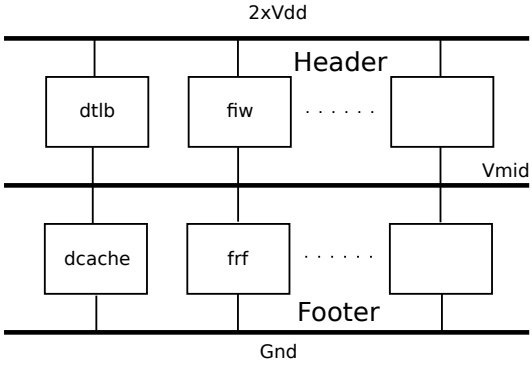


Fig. 4: The functional units will be stacked based on their power consumption and the correlation of power among units. The middle layer for all the units will form a V_{mid} power net.

4.2. Inter-Level Communication

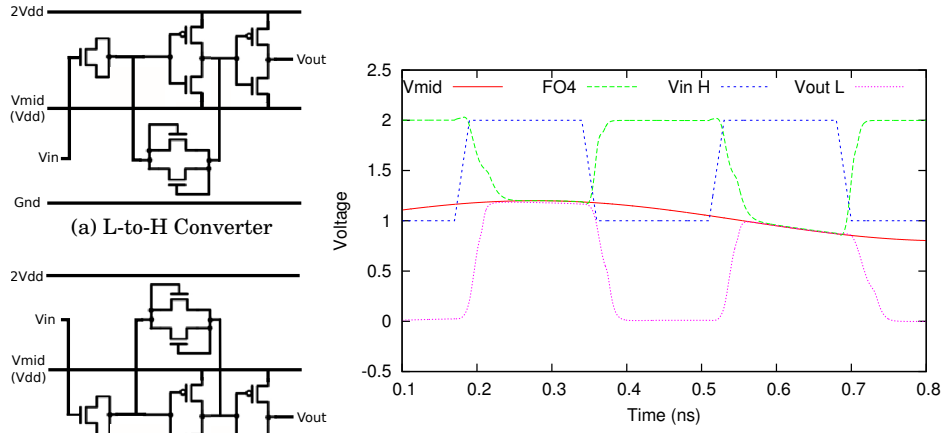
Communication between functional units located at different partitions, hence different levels, requires a level converter to adjust the logic's electrical level to that of the destination. Level converters have been studied in several works (for example see [Ishihara et al. 2004]). We use a level converter similar to the one presented in [Gu and Kim 2005] (see Figure 5).

Level converters should be utilized judiciously, as they consume power, add area and delay overhead. This actually presents the following trade-off: Finer FU granularity would provide more FUs or components per design, allowing a better matching partitions. However, increasing the number of FUs would result in the increase in the number of level converters.

To obtain the characteristics of the level converters, we implement them in SPICE using a $45nm$ technology [Zhao and Cao 2006]. To fairly consider leakage, we assume a clock width of 23 Fan-Out4 (FO4) gates [Choudhary et al. 2011][1], and scale the leakage of each gate accordingly. Figure 5c shows the transient simulation result for the *Header* to *Footer* converter using SPICE. The simulation also adds 10% voltage noise on the V_{mid} voltage to ensure robust functionality. The proposed level converter consumes $54uW$ of power at 1V, which is about 10 times power consumption of a *not* gate at the same technology. While the power consumption overhead is negligible (less than

[1]The actual range in the reference is from 23 to 33 FO4. We consider the lowest width to account for the worse case impact.

0.1% of the total power in our experiments), the level converter adds a delay as big as one FO4 *not* gate. It might not be possible to absorb such a delay in the critical path of performance. For simplicity, we assume a worst case situation with a full cycle overhead when talking between voltage domains. Section 6 will discuss the exact cost.



(a) L-to-H Converter

(b) H-to-L Converter

(c) Transient Simulation

Fig. 5: Simulation results of the level converter model in SPICE for 40nm CMOS technology. The converter delay is comparable with a not gate with fan out of 4 (FO4).

To minimize the overhead of inter-level communication, particularly for the critical path of performance, we consider the communication cost among blocks for any partitioning solution in the partitioning process. Our experiments show that splitting the core into 20-30 FUs would result in a partitioning allowing good correlation and small power overhead due to level converter.

4.3. Systematic Partitioning

We define the stacking problem as partitioning of the functional units (FU) in a core into two *Header* and *Footer* partitions. The partitioning has to satisfy three goals: 1) Both partitions have matching average power consumption; 2) Total power consumption of both partitions correlates well with each other; 3) The communication of blocks between partitions is minimized. Note that goals 1 and 2 refer to power, but goal 1 is related to the average power (*i.e.*, average over the execution time), while the goal 2 is related to instantaneous power (*i.e.*, average power over a delta time, we use 100 cycles).

To estimate the cost of the communication between blocks (Σcut), we have verified which FUs communicate with each other, and what is the weigh of such communication (*i.e.*, how many wires are used), considering a 64-bit architecture, namely $w_{i,j}$ for FUs i and j. For each solution, if i and j are in the same cluster, we add $w_{i,j}$ to the total cost, if they are in the same cluster, we add 0.

Any classical partitioning algorithm can be adopted to partition the functional units into two *Header* and *Footer* groups. We choose Genetic Algorithm (GA). Each individual, representing a valid solution, comprises a string of binary genes. “0” in location i in an individual means that the FU_i belongs to the *Footer*, while “1” means it belongs to the *Header*.

The cost function to be minimized is a superposition of the mismatch in power consumption, correlation between the two partitions, as well as the cost of inter-level communication:

$$CostFunc = \alpha \times \Delta Pow + \beta \times \Delta Cor + \gamma \times Comm_{critical} \quad (1)$$

$$\Delta Pow = (Mean(Power_{Header}(t)) - Mean(Power_{Footer}(t)))/Mean(Power_{total}(t)) \quad (2)$$

$$\Delta Cor = 1 - Cor(Power_{Header}(t), Power_{Footer}(t)) \quad (3)$$

$$Comm_{critical} = \Sigma cut/(\#FUs)^2 \quad (4)$$

4.4. Run-Time Load Balancing

Ideally, partitioning the FUs, based on the magnitude (average power) and correlation of their instantaneous power consumptions, into *Header* and *Footer* groups and stacking them would result in no load mismatch between the *Header* and *Footer* during the execution. However, in reality, the utilization of the FUs differs across workloads and varies over time. This results in transient mismatches (6% on average in our experiments) that have to be managed.

We categorize mismatches by their power magnitude. A mismatch is said to occur when there is a difference in power of more than 10%, that is considered to be an acceptable fluctuation during the operation of a chip:

Small Magnitude: Mismatches up to 20% of the power consumption are considered small.

Moderate Magnitude: Mismatches in the 20% to 50% range are considered moderate.

High Magnitude: Mismatches of more than 50% of the power consumption are considered high.

To manage these transient mismatches, we propose the use of circuit and microarchitectural mechanisms. Mismatches that are very-short (*i.e.*, $<\approx 100$ clock cycles) are too fast for microarchitectural mechanisms, therefore we rely on decaps to filter out those high frequency mismatches. Note that the total available on-die decoupling capacitance now has to be split between V_{dd}, Gnd and V_{mid}, thus we propose the use of on-package decaps as well.

The **Small Magnitude** mismatches are dealt with by a secondary VR (*sVR*) for regulation (details on Section 4.8). *sVR* is a small and fast, voltage regulator, that can efficiently balance the V_{mid} voltage. The reason to use the *sVR* only for small magnitude mismatches is to avoid the need of a full sized VR (as occurred in previous Voltage Stacking proposals).

To limit the maximum current output that *sVR* has to supply, we use "Dummy Activity" (DA) for **Moderate Magnitude** mismatches. DA introduces extra activity in few selected FUs in the level that has less power to match up power consumption of the group.

Relying too much on DA might require extra logic that can sacrifice the cycle time. In case of **High Magnitude** mismatch, DVFS brings down the overall power consumption and the magnitude of mismatch within the range that *sVR* can regulate, since it is now relatively stronger. This means that managing high magnitude mismatches implies performance degradation as well. This is an additional use of DVFS, as an addition to current uses of DVFS that include thermal and power management.

High Magnitude mismatches did not occur in our simulations, and **Moderate Magnitude** occurred a mere 0.5% of time in the execution of all the applications. In any case, voltage scaling is proposed as a safety backup to guarantee correctness in hypothetical extreme cases like a malicious program running. The *sVR* is present and operating at all times.

To estimate the mismatch, a controller based on the voltage of V_{mid} is used to trigger DA and DVFS. If the voltage is above/below the set thresholds, custom circuitry in

the core activates DA on specific FUs, or DVFS. The on-package decaps are sized to filter very short transients, giving time for DA and DVFS to actuate. The choice of on-package decaps provides large enough capacitors for the task, while avoiding the need of using pins for V_{mid} and increase in area.

4.5. Multicore

CoreUnfolding applies the stacking within a core and the interface of the core to the outside remains the same. The cores in a multicore configuration will be connected in a parallel way, similar to the traditional power delivery system, but with higher voltage and lower current. Therefore multithreaded applications would run on a multicore *CoreUnfolding* system with no extra consideration. To implement multicore *CoreUnfolding*, each *type* of core in the system goes through the partitioning framework once, and cores with the same type replicate the same partitioning. Note that we distinguish between the core (including private caches), and the shared memory levels and structures such as memory controller.

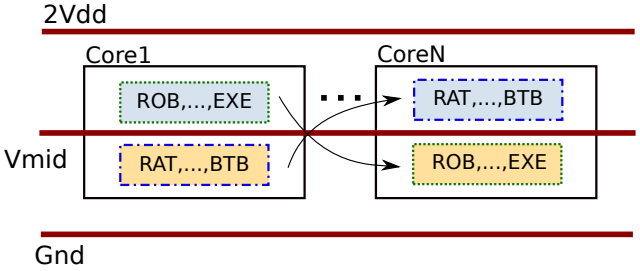


Fig. 6: In a multicore configuration, cores are connected in parallel. The stacking is applied within each core, and the two partitions are flipped across cores so that the minimal possible bias toward one partition in a core can be canceled out by the next core.

We notice that swapping *Header* and *Footer* across the cores would provide better results, as the minor bias toward one partition in one core can be canceled out by the next core of the same type (Figure 6). As we observe in our experiments, this yields to a very good balance between *Header* and *Footer*.

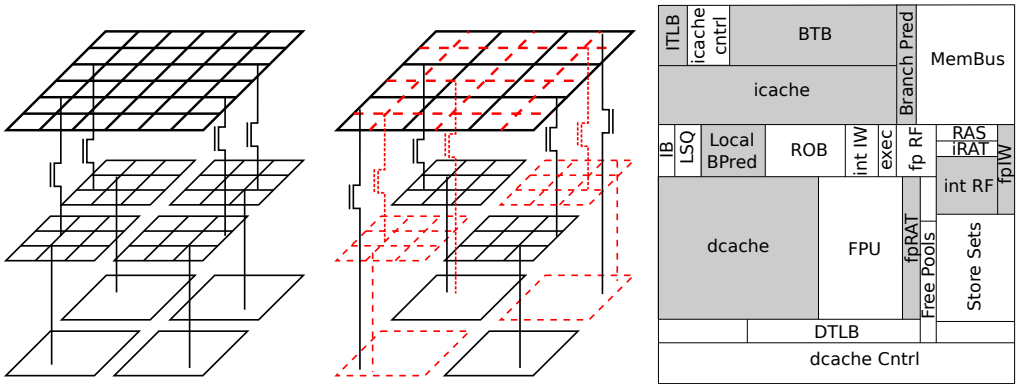
Another possibility, is to design each core with its own V_{mid}, and thus each core act as an isolated core. The small bias will remain, but will not accumulate. This option is required when per-core DVFS is needed.

4.6. Other Voltage Domains

The LLC and memory controller in many processors reside on a different voltage domain than the core. These structures consume a small fraction of the processor power. Nonetheless, the structures within each of these units can be stacked. IOs and PLLs also are among the structures with different voltage domain. However, we do not evaluate stacking for these voltage domains.

4.7. Floorplanning

To avoid adding wire delays between FUs, *CoreUnfolding* does not affect floorplanning. Each FU will either tap V_{dd} and V_{mid} or V_{mid} and Gnd, though a global vs local approach. A good analogy is to a scheme with power gating, where a block of logic has its own local power rail. The connection scheme is depicted in Figure 7b, the dashed lines represent V_{mid}. The analogy with power-gating technique is shown in Figure 7a. We keep the power-gating transistors, although that is not strict necessary for *CoreUnfolding* alone. Also, level converters require both domains, so we also pull wires directly from the global level power rails when needed. These scheme does not incur extra overhead to add a new local power rail. Finally, the addition of the global



(a) Power gating architecture (b) *CoreUnfolding* architecture (c) Floorplan

Fig. 7: Instead of changing the floorplan, *CoreUnfolding* uses an approach inspired in power-gating to avoid overheads.

V_{mid} can be mitigated by the reduction of metal dedicated to V_{dd} and Gnd, since the current through those rails is decreased.

CoreUnfolding requires either an silicon-on-insulator or triple-well technology to isolate wells across the chip [Lee et al. 2012], since silicon in each cluster will be biased with different voltage levels. Figure 7c shows the floorplan that was used by ESESC in our simulations. The shaded FUs are the FUs in the header group.

The choice of multiple power rails avoids the need of routing the three crossing domain wires of the level converter (Figure 5), avoiding overheads in routability, and reducing the costs of wires. The drawback is that it adds cost in area for the extra power rail. Our approach adds one extra rail per cell row.

4.8. Power Delivery Architecture

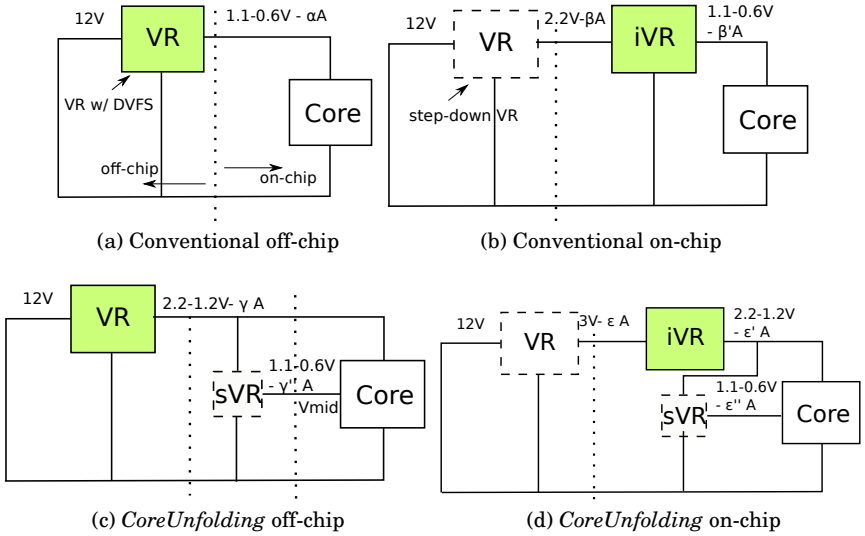
Figure 8 shows the *CoreUnfolding* configurations against conventional ones. We categorize the voltage regulators regarding their functionality. Step-down regulators perform a fixed step-down conversion from the input to output; they are specified in Figure 8 by dash lines. Voltage regulator modules with DVFS controller provide multiple voltage output, in addition to step-down. They are specified by a solid fill in Figure 8. Besides, either of the voltage regulator types can be implemented off the chip or integrated on the chip. Next we explain each power delivery architecture. The supply voltage to the system is usually 12V, or 3.7V for battery operated devices.

Conventional off-chip: As shown in Figure 8a, the off-chip VR (located on the motherboard) performs step-down, usually 12-to-Vcore, or 3.7V-to-Vcore for battery operated devices. As example, we assume 12V input voltage and maximum Vcore=1.1V. The VR can output different voltages as requested by the processor. Most processors use a similar configuration.

Conventional on-chip: In this configuration, part of the voltage conversion functionality is moved into the chip. The motherboard's VR only performs step-down to 2-3V [Burton et al. 2014]. The integrated voltage regulator (iVR) provides the final step down to the range requested by the processor. Figure 8b shows this architecture, an example of which is Intel's Haswell [Burton et al. 2014], despite being called integrated, depending on the actual implementation, they may use an on-package inductor.

4.8.1. CoreUnfolding. By increasing the supply voltage, *CoreUnfolding* improves power delivery for both off-chip and on-chip voltage regulation configurations. The benefit could be gained through static (design time) and dynamic (run-time) knobs. The lower maximum current demand allows for design time optimization that leads to **smaller**

area and **higher efficiency**, and the lower average current demand at runtime allows for less resistive loss in the delivery network. Note that *CoreUnfolding* is not dependent on a particular implementation of VRs, but, naturally, can benefit from better VR implementations. The regulation of the V_{mid} is performed by *sVR* and can also be into the package or fully integrate into the die, since it only performs a low power output step-down from the main input voltage to the core.



(a) Conventional off-chip (b) Conventional on-chip

(c) *CoreUnfolding* off-chip (d) *CoreUnfolding* on-chip

Fig. 8: Power delivery in *CoreUnfolding* (c and d) versus conventional architecture (a and b). The lower maximum current demand in the *CoreUnfolding* configurations, which is around 60% of the conventional configurations, results in smaller and more efficient voltage regulators, reduced di/dt, and better power delivery. The colored voltage regulators in the figures specify the VRs with DVFS functionality.

CoreUnfolding off-chip: In the case of an off-chip VR, *CoreUnfolding* enables the VR to benefit from a lower step down ratio [Piqué and Bergveld 2012]. This is shown in Figure 8c. Note that the VR output voltage is now twice.

CoreUnfolding on-chip: Figure 8d shows *CoreUnfolding* with integrated voltage regulator. In this configuration, the VR on the motherboard only performs the step-down, still a smaller step down can be used, since the final supply voltage is higher. In this case, it makes sense to merge iVR and *sVR* into a ladder voltage converter (*i.e.*, a converter capable of supplying more than one output voltages) [Lee et al. 2015].

VR has to be designed for worst case, which is observed when there is 20% load imbalance between the stack levels (our cap), in particular, the primary VR (either the iVR in the on-chip configuration or VR in the off-chip configuration) will be most loaded when the *Header* group consumes 20% more power. The following formulates the ratio of **maximum** output current that the voltage regulators has to provide. Note that for simplicity we ignore the efficiency of the voltage regulators in computation of the current ratios. The nomenclature is in accordance with Figure 8a and Figure 8c.

In a perfectly balanced condition, the current going through the logic in conventional and *CoreUnfolding* is:

$$\gamma = I_H = I_F = 0.5\alpha \tag{5}$$

I_H and I_F stand for current passing through the *Header* and *Footer* levels respectively. Perfect balance is not always the case, the resulting current in the case of mismatch is:

$$\gamma = max(I_H, I_F) \tag{6}$$

We cap the demand from *sVR* to 20% using DA technique.

$$\gamma'' = |I_H - I_F| = 0.2 \times 0.5\alpha. \tag{7}$$

Therefore, the maximum current ratio between the *CoreUnfolding* and conventional configuration is as follows.

$$\gamma = max(0.5\alpha + \gamma'', 0.5\alpha - \gamma'') = 0.5\alpha + \gamma'' \tag{8}$$

$$\frac{\gamma}{\alpha} = \frac{0.5\alpha + \gamma''}{\alpha} = \frac{0.5\alpha + 0.2 \times 0.5\alpha}{\alpha} = 0.6 \tag{9}$$

For the same power, the primary VR in *CoreUnfolding* is designed for 60% current output of VR in the conventional architecture. The average ratio is closer to 50% as will be discussed in Section 6. As a result, it is smaller in area and can be optimized for more efficiency in the regulation.

The current demand from *sVR*, γ'', would be much less than the VR in the conventional configuration. In fact, most of the time there would be no demand from *sVR*, and if there is a demand, it is capped to 20% of the power consumption of the group, or 10% of the power consumption of the chip (which represents $10\times$ less than previous voltage stacking approaches). As a result, *sVR* would be much simpler, smaller and more efficient than the primary VR.

Recently, industry solutions, such as IBM Power8 [**?**] and Intel Haswell [Burton et al. 2014], are opting for integrated regulators, with power switches distributed across the die, which are more suited for quick responses to load demands and allow more fine grained control of the voltage in different parts of the chip. For our experiments, we assume a primary VR such as Intel's Haswell regulator. For the secondary VR we proposed the use of a Switched-Capacitor DC-DC stacked converter such as the one presented by Lee et al. [Lee et al. 2015], that allows for very efficient stacked conversion.

5. SETUP AND EVALUATION METHODOLOGY

5.1. Architectural Simulation Setup

We use ESESC [K. Ardestani and Renau 2013] for our architectural evaluation, including performance, power and temperature. The temperature dependency of leakage power is modeled as well. Table I lists the architectural parameter of the core and processor we model.

Table I: Architectural parameters

Parameter	Value
#Cores	1 and 4
Freq	3.0 GHz single core, 2.5 GHz multicore
I$	32KB 2w (2c hit) private
D$	32KB 8w (3c hit) private
L2	256KB 16w (12c hit) private
L3	2MB×#cores 16w (12c hit) shared
Coherence	MESI
Mem.	180 cyc best case from LLC
BPred.	10 tab. ogehl 76Kb
Issue/ROB/IWin	4/256/32
Load/StoreQ	48/32

5.2. Power Delivery Network Setup

ESESC power consumption trace is used to generate a time-varying impedance model for each FU, which is then fed to a transient SPICE simulation that models the power delivery network, this methodology was proposed in [Leng et al. 2014]. The off-chip power delivery network is similar to the model presented [Leng et al. 2014], the on-chip power network is more detailed using the IBM Power Grid Benchmark [Nassif 2008] and includes on-package decoupling capacitors. Figure 9 shows the complete power delivery network used, with the simulation parameters. The on-chip *sVR* is also represented. We model our *sVR* as a fully integrated, 2-level symmetric ladder switched-capacitor DC-DC converter [Lee et al. 2015], with efficiency of around 95% (depending on the load mismatch).

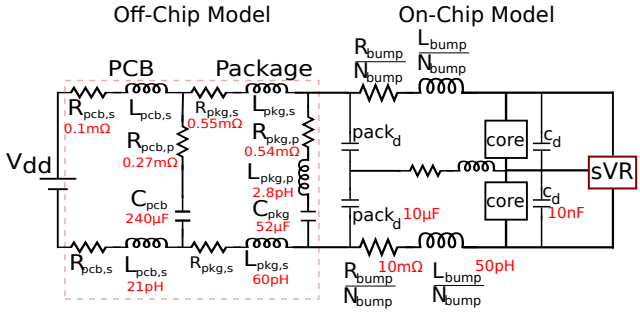


Fig. 9: A detailed power distribution model is simulated to evaluate the voltage fluctuation in the middle rail.

ESESC dumps the power trace every 100 cycles, we verified that this is an enough resolution for the modeled power delivery network. Even using a maximum power density of $1W/mm^2$ is enough to sustain the transient mismatches that appear on the order of 30ns (100 cycles) because most of the capacitance needed comes from the on-package decoupling capacitors [Popovich et al. 2007]. On-package decaps are cheaper then dedicating more on-chip area for on-chip decaps, but due to pads parasitics, present smaller decoupling power. To size these decaps, some preliminary transient SPICE simulations were used. The decap size was chosen so that V_{mid} fluctuation in 100 cycles interval was smaller than 10%. Those simulations show that just one tantalum SMD capacitor [Vishay Sprague 2013] is enough to keep V_{mid} with less than 10% drop for 300 cycles for the whole multicore simulations.

5.3. Benchmarks

We use applications from SPEC CPU 2000 [J.L. Henning 2000], CPU 2006 [Henning 2006], PARSEC [Bienia et al. 2008] and SPLASH-2 [Woo et al. 1995] benchmarks suites. Our training set consists of applications from the SPEC benchmarks (Table II), and the evaluation is done with all the three benchmarks (Table III). This allows us to evaluate the impacts of partitioning in unknown workloads, and show the robustness of the method.

Table II: Training Set

SPEC	bwaves, lbm, vpr, applu, equake, gcc, mgrid, mesa, art, astar, soplex, twolf, perlbench, swim, crafty, povray, milc, sphinx, dealII, wupwise, vortex, libquantum, mcf, gap, leslied

Table III: Evaluation Set

SPEC	bwaves, lbm, vpr, applu, equake, gcc, mgrid, mesa, art, astar, soplex, twolf, perlbench, swim, crafty, povray, milc, sphinx, dealII, wupwise, vortex, libquantum, mcf, gap, leslied
PARSEC	blackscholes, bodytrack, canneal, facesim, ferret, fluidanimate, swaptions, x264
SPLASH-2	ocean, fft, fmm, radix

5.4. Evaluation Methodology

We go through two phases in our evaluation methodology. First phase, **Training and Partitioning**, gathers power consumption information across benchmarks. Then it uses that information to extract the correlations and generates a partitioning solution. The next phase, **Validation**, performs full simulation to measure the mismatches with a fine grained resolution. For all the simulations, we skip the first 2B instructions, and simulate up to 10B instruction.

5.4.1. Training and Partitioning. We use sampling to gather power consumption information. The sampling simulates 50K instructions every 1M instruction, and dumps power consumption information at the end of every sample. We only apply training for the SPEC applications. The power traces from all the benchmarks are aggregated, and use the result based on which we perform the GA partitioning.

It is also possible to perform the training per benchmark for better results. Implementation of this, however, would require support for reconfiguration of the stacking. We do not consider reconfiguration in this work, but we will report the result to show how good the global partitioning performs (evaluated as $base$ and $peak$ partitioning in Section 6). For all the applications, the GA converges within the first 100 generations. The execution time on average was 3 hours. One can perform full simulation for training as well, but we noticed that sampling based information was good enough to guide the training. Table IV shows the resulting partition, used through our evaluation. If we go back to Figure 3, we note that dtlb and dcache have very good correlation, and indeed ended up in different clusters, as one would expect. Note, however, that this is not necessary the case for all the FUs, since the clustering is done considering the combination of FUs, rather than individual FUs. This partitioning solution resulted in around 1k level converters (considering a 64-bit architecture), which we consider to be a small overhead.

Table IV: A partitioning solution.

Partition	**Functional Units**
Header	FP RATs, Branch Predictor, Branch Target Buf., itlb, icache, dcache, FP inst. window (FPIW), int Register File (iRF)
Footer	Int RATs, Free Pools, Ret. Addr. Stack (RAS), Instruction Buffer, icache cntrl, dtlb, dcache Cntrl, Load Store Queue, Store Sets, Int Inst. Win., Exe. Units, FPU, FP Reg. File, Reorder Buf.

6. EVALUATION AND RESULTS

We first compare *CoreUnfolding* against the conventional configuration for area, power, performance, and di/dt. Then we also evaluate the quality of the partitioning algorithm on which *CoreUnfolding* is based.

6.1. Area

Higher current draw requires larger inductance and capacitance components, as well as wider drivers. Multiphase VRs and shared current bus have been proposed to enable delivering higher current [Zhou et al. 2000; Huang et al. 2003]. Despite the considerable advances in increasing VR current density recently, specially with integrated VRs, such as the Intel Haswell FIVR [Burton et al. 2014] and the IBM Power8 [Toprak-Deniz et al. 2014], VR still takes a lot of area. Considering a Intel Haswell-like processor with maximum current draw of roughly 100A, and $31A/mm^2$ [Burton et al. 2014; Intel 2014], the *iVR* takes $\frac{100A}{31A/mm^2} = 4.12mm^2$.

Figure 10 shows the distribution of current in the baseline (b) and *CoreUnfolding* (c). For multicore applications, the results are reported per core. *CoreUnfolding* reduces the maximum and average current across all the applications. Based on the formulations in Section 4.8.1, the maximum current should be reduced by 40%, the smaller reduction in Figure 10 are due to the fact that these applications do not necessarily draw maximum current. Given the maximum current draw reduction, and considering 10% area overhead of *sVR* compared to the conventional VR, *CoreUnfolding* would allow for a reduction of 30% in VR die area (or $1.24mm^2$ in the Haswell-like solution). This does not take into account on-package inductors needed, that take up to $20mm \times 8mm$ area in the FIVR solution [Burton et al. 2014].

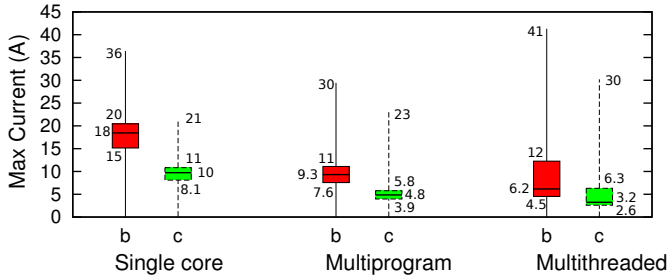
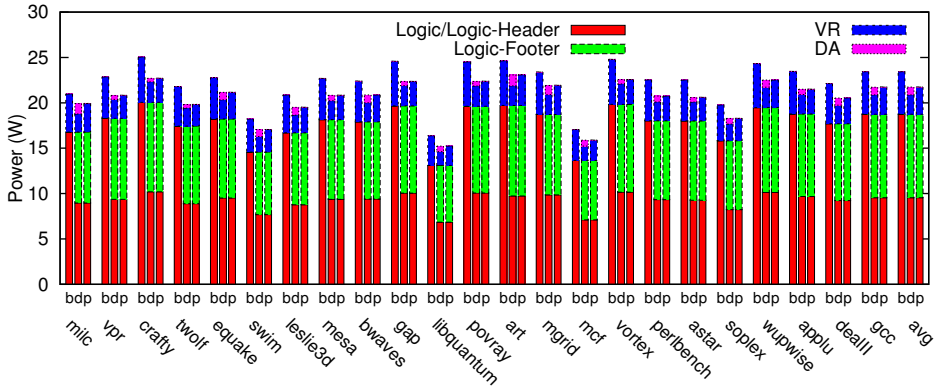


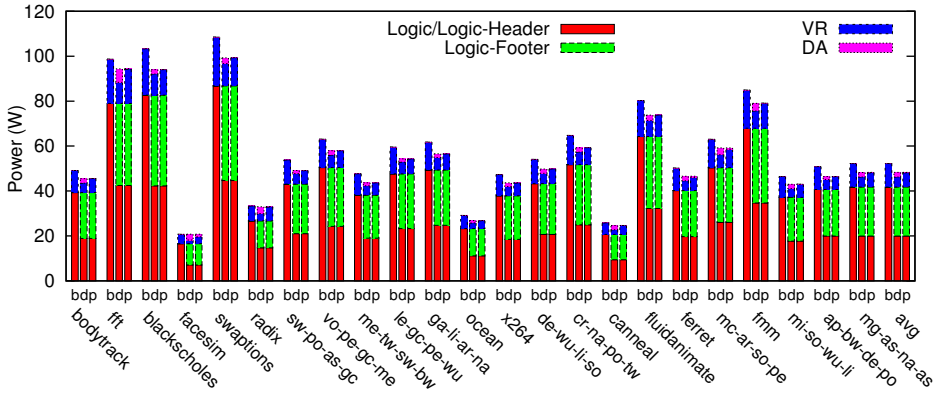
Fig. 10: *CoreUnfolding* reduces the maximum as well as the average current drawn by each core. At peak, *CoreUnfolding* (c) needs 40% less current than Baseline (b).

At high current densities, thermal is a major limiting factor. A non ideal VR with 70-80% efficiency [Burton et al. 2014] with current density of 8 A/mm^2 would have power densities around 1.6 W/mm^2 (8 A/mm^2 × 1V × (100-80)% = 1.6 W/mm^2), which is higher than the core itself. This could incur extra cooling. Another possibility is to increase the area of the voltage regulator for power density of $1W/mm^2$, to keep the power density of the VR below that of the core. Hence, for high density voltage regulators, the power density is proportional to the inefficiency as well. Better efficiency in this case could translate to denser and smaller modules. From the observations made from currently available commercial voltage regulators (Figure 1), we estimate that *CoreUnfolding* would be able to improve around 10% the in VR efficiency (by doubling the output voltage and reducing by 40% peak current). The power density in the *CoreUnfolding* configuration would be (8 W/mm^2 × (100-88)% = 0.96 W/mm^2). Since this is below 1 W/mm^2 that conventional cooling solutions can handle, there is no need to increase the area to reduce the density. Hence, at both fronts (current and power densities), *CoreUnfolding* results in a smaller voltage regulator.

6.1.1. Sizing of sVR:. The size of *sVR* is dictated by the dynamic load balancing engagement thresholds, *i.e.*, thresholds for using *sVR*, DA, and DVFS. The value for this thresholds should be decided based on the common case, *i.e.*, expected load mismatch ratio. The average load imbalance was about 6% of the total power. Less than 1% of the time there is a mismatch greater than 20% of the total power. Therefore we conservatively pick the threshold of 20% for *sVR*. As a result, the *sVR* will be very small, up to 10% of the baseline voltage regulator. Hence, *sVR* can be fabricated in a distributed way across the chip using stacked power transistors [Rajapandian et al. 2005]. On-chip implementation of *sVR* is crucial for *CoreUnfolding* as such an implementation provides a quick response to mismatches, even easing the pressure on the amount of decoupling capacitance required for very short transient mismatches. Overall, up to 30% of the area dedicated to on-chip VR can be saved.



(a) Singlecore



(b) Multicore

Fig. 11: Power consumption of different techniques: baseline (b), *CoreUnfolding* with DA (d) and *CoreUnfolding* with DA and VR (p). *CoreUnfolding* utilizing DA and VR for runtime load management saves about 7.5% of the total power.

The number of pins dedicated to power delivery is also proportional to the maximum current required by the design. The total number of pins in the package defines the minimum footprint size of the package, which could be an important factor nowadays with the emergence of mobile devices. While the maximum current per pin increases over the years, the number of power pins is proportional to the maximum current. Given that *CoreUnfolding* reduces the maximum current by 40%, the number of pins

in a processor with *CoreUnfolding* can be reduced by the same amount. Given the number of power pins in current technologies, this reduction would account for about 20% of total pin count of the package. According to [Dong et al. 2010], 20% reduction in pin count reduces the package cost by about 20%.

This is the first work that caps the demand of the extra voltage regulator used by voltage stacking. Without capping the demand, stacking can reduce the total number of pins, but it can not reduce the VR area. Note that none of the related works cap the demand from the extra VR. For example, in [Zhan and Sapatnekar 2008] the secondary VRs overall need to source the same maximum current as the primary VR. So in the best case, all the area saving in primary VR will be lost in the secondary VRs. [Gu and Kim 2005] does not require *sVR*, but the solution is deemed impractical due to timing overhead of merely relying on "digital voltage regulator" or dummy activity. So we do not compare against this work.

6.2. Power

Figure 11 shows the total (dynamic plus leakage) power consumption breakdown for "logic" (split between *Header* and *Footer* for *CoreUnfolding*), voltage regulator (VR) and dummy-activity (DA). Both single and multicore configurations are reported. For each benchmark, three bars are shown:

Baseline (b), that reports the power consumed for the non-stacked baseline core. We consider 80% VR efficiency [Inc. a; Technologies ; Inc. c].

***CoreUnfolding* with DA** (d), that reports the power consumed when only dummy activity (DA) is used for mismatch management. We report the breakdown of power for *Header* and *Footer* partitions, and assume a moderate 10% increase in the efficiency due to lower current draw and voltage drop [Inc. a; Technologies ; Inc. c]. This configuration does not rely on voltage regulator for load balancing between the stacks. DA effectively manages the load mismatch, however, all the power consumed by DA is wasted power, which accounts for 3.6% of the power consumption. The DA power consumption is relatively more for benchmarks that exhibit more mismatch (See Figure 14 for detailed mismatch distribution during the execution of each benchmark). In reality, merely relying on DA for load management could have cycle time implications.

***CoreUnfolding* with *sVR* and DA** (p) reports the power consumed when both DA and *sVR* are used as mismatch management. The power wasted by *sVR* is inversely proportional to the efficiency of the VR, which we consider to be around 95%. This extra power is not distinguishable in the plots, compared to the d bars, because it adds up to 0.6% of the total power consumption. As mentioned in Section 4, to limit the size of the *sVR*, we use DA to cap the maximum mismatch, hence the maximum current demand.

In our experiments, **high magnitude** mismatches did not occur, and thus DVFS was not necessary. As one can note in Figure 11, the power consumption of the load management (VR and DA) is a small portion of the total power. For the benchmarks evaluated, the utilization of DA is rare (about 0.5% of the total execution on average).

CoreUnfolding configurations (d and p) also have a component of power consumed by the level converters. However, this components accounts for less than 0.1% of the total power, and hence are not shown. We also do not include the resistive loss in the delivery network, *i.e.*, pins and pads for off-chip solutions, and power grid on the chip, in the report of power savings. Considering about $10m\Omega$ resistance for pads and rails, extra 2-3% extra power saving would be observed.

Efficiency of voltage regulators varies over the program execution. However, to simplify the evaluation, and without loss of generality, for both base and *CoreUnfolding* configurations, we consider a flat efficiency rate to estimate the possible power savings. *CoreUnfolding* configurations (d and p) have roughly the same power consumption (p around 0.5% higher than d). When the *sVR* is used, it takes some power, but DA is less active, and thus takes less power. DA only provides a very bad management of mismatch, since DA it is considerably slow.

6.3. Performance

While power consumption and area overhead of the level converters used for communication between the stack levels is negligible, the performance overhead might not be possible to absorb. The SPICE simulation shows that delay of the level converter is about the same as a FO4 gate. This could account for about 4.3% of the clock width in a modern processor [Choudhary et al. 2011].

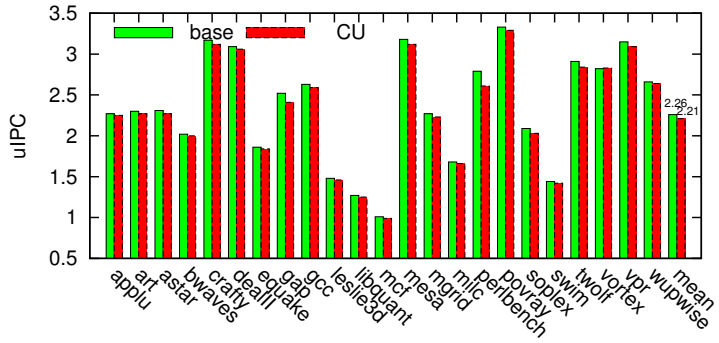


Fig. 12: *CoreUnfolding* has less than 2.2% performance ignoring improved power, noise and area provided by *CoreUnfolding*.

The partitioning algorithm tries to minimize utilization of level shifter for the critical components. However, it is hard to guarantee no additional penalty in all the critical paths. Instead of widening the clock width, we add 1 cycle to the access time of the functional units that require level shifters. In a real design, adding one cycle of delay in every boundary cross is pessimistic, since most of the crossing will not be in critical paths. Thus, for a real implementation, the latency addition should be more carefully evaluated. Figure 12 shows the results in terms of uIPC for single threaded applications, already considering the extra latency. uIPC is the retiring rate of micro-operations (result of instruction decode). The simulations show 2.21% decrease in the performance as a result.

Better voltage supply in terms of IR drop and Ldi/dt results in better performance. Our experiments, presented in more detail in Section 6.4, show 49% less di/dt and 45% less IR drop. Therefore, the peak performance loss can be compensated by the reduced voltage noise and better voltage margin as a result of lower current draw. Also, multithreaded or multiprogrammed applications performance could even increase due to the power savings, and better thermal profile in case an on-chip voltage regulator is used (similar to Haswell). In such a case, the power saving can allow for higher turbo multiplier when more than one core is active.

6.4. di/dt

Reduced current improves both resistive and inductive (di/dt) voltage drops. Both type of voltage drops reduce the voltage margin, which is already constrained in modern high performance processors. Inductance in *CoreUnfolding* should be more or less the same compared to the baseline for on-chip VR solutions. For off-chip VR solutions the number of pins is reduced in half, but the number of pads does not necessarily change. Hence, the inductance close to the load should not change much. Overall, smaller di/dt relaxes the quick response pressure on the voltage regulator and improves the noise. Figure 13 shows di/dt of the benchmark categories for the conventional and proposed architecture. di/dt was calculated using power traces from ESESC, which were used to calculate the current in each interval. For the multicore benchmarks, the metric is

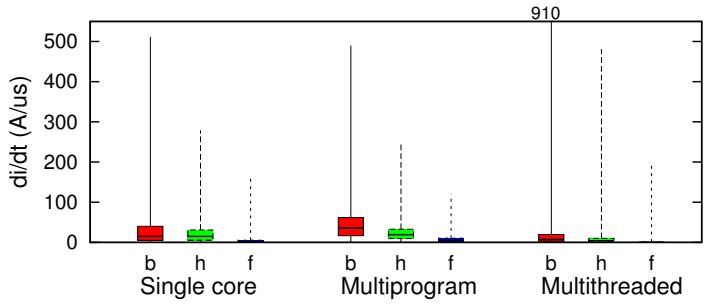


Fig. 13: *CoreUnfolding* reduces the maximum as well as the average di/dt.

reported per core. We compare the gradients for every 100 processor cycles, which is the highest temporal resolution in our setup, and scale it to the desired unit of $A/\mu s$. b shows the results for the baseline. For *CoreUnfolding*, the metric is shown for both V_{dd} and V_{mid} rails, labeled respectively h and f in Figure 13.

Fluctuations in power consumption affects di/dt of the V_{dd} rail. In *CoreUnfolding*, the possible load mismatch between the *Header* and *Footer* groups would have a minimal impact on the di/dt of the V_{dd} rail as well. This is because the *sVR* could draw current from the V_{dd} rail for regulation. The V_{mid} rail on the other hand, is only affected by the capped mismatch rather than the total current draw. Hence it demonstrates less di/dt. The experiments are performed without dynamic scaling of voltage or frequency. As Figure 13 shows, by reducing the current demand, *CoreUnfolding* reduces the maximum di/dt by 49% on average across the benchmarks.

Decoupling capacitance of the supply nets is reduced in *CoreUnfolding*. Nonetheless, with the decrease in the current draw, the impact of decreased decoupling capacitance on voltage noise is compensated.

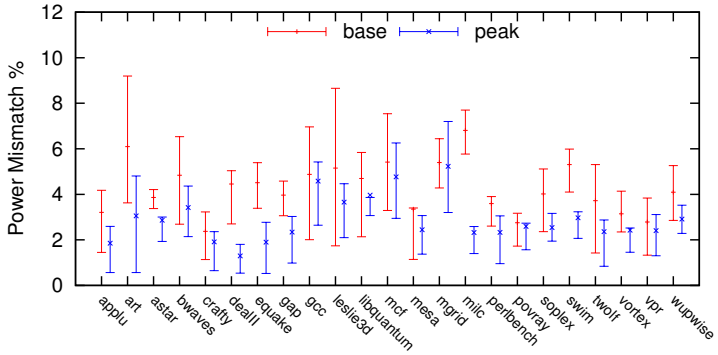
Multithreaded applications, *i.e.*, PARSEC and SPLASH benchmarks, exhibit a smaller average di/dt, but higher maximum. The reason is that threads dynamically spawn or terminate on a core during the execution, which results in a sporadic higher di/dt points. Nonetheless, these points are not frequent enough to shift the average.

6.5. Transient analysis of V_{mid} Voltage and quality of partitioning

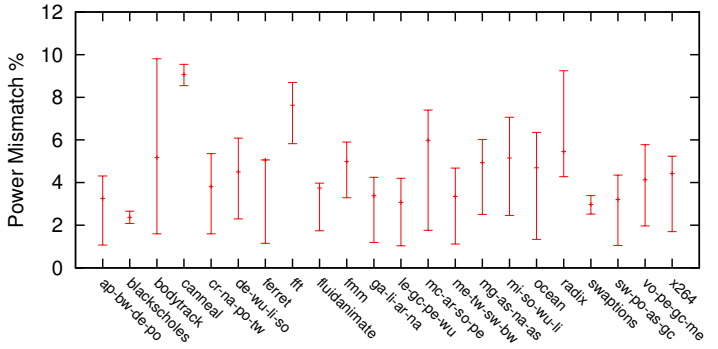
This section evaluates the quality of partitioning algorithm in terms of reducing the load mismatch between *Header* and *Footer* groups. This is before the dynamic, runtime load balancing techniques (DA and VR) get engaged to remove all the mismatches. The partitioning is critical in *CoreUnfolding* to minimize the overhead of runtime load balancing.

Figure 14 shows the results. The base method in this figure partitions the functional units to *Header* and *Footer* groups based on the aggregated data from running multiple benchmarks. The base approach limits the worst case scenario at the cost of loosing the best scenario. The best scenario allows for adjustment per benchmark, *i.e.*, to partition the functional units per benchmark. Figure 14a presents the results for this under $peak$ label. The $peak$ is only presented to show how well the actual $base$ partitioning performs. We do not use multicore applications to train the partitioning, hence $peak$ does not apply to those applications. Instantaneous power mismatch stays within 10% for 75% of time for all applications. Also, it is below 20% for 99% of the time for the combination of applications (not shown in the figure), the maximum observed mismatch was of around 25%.

To better illustrate how this impacts the voltage in each level of the stack, we run transient SPICE simulations with the power traces for the partitioned core using the model from Section 5.4. Figure 17 shows the trace of the voltage for the *Header* and



(a) Single Core



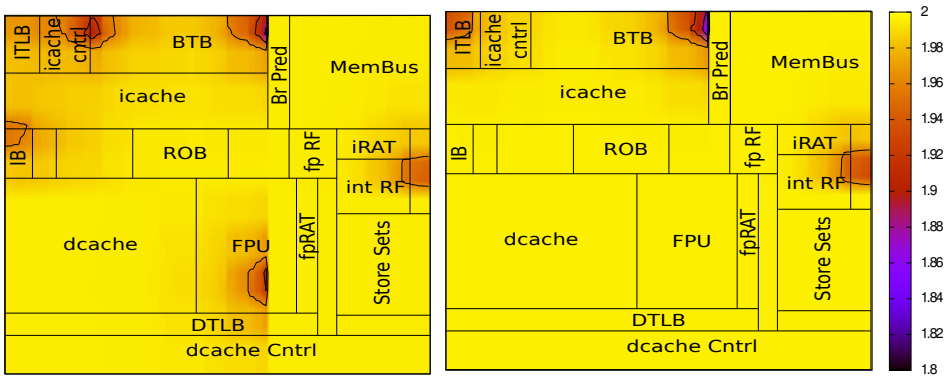
(b) Multicore

Fig. 14: Distribution (Q1/Median/Q3) of instantaneous load mismatch for the benchmarks before runtime load balancing is engaged. The $base$ method, using one partition for all the applications, performs comparably to $peak$ which adjusts the partitioning per application.

Footer groups for two applications. More than 99% of the time, decaps and *sVR* are able to keep V_{mid} stable within 10% of the nominal voltage. A full transient SPICE simulation of architectural benchmarks is unfeasible, the proposed approach is compatible with industry standards, with the difference of using a default power grid instead of extracting parasitics from GDS.

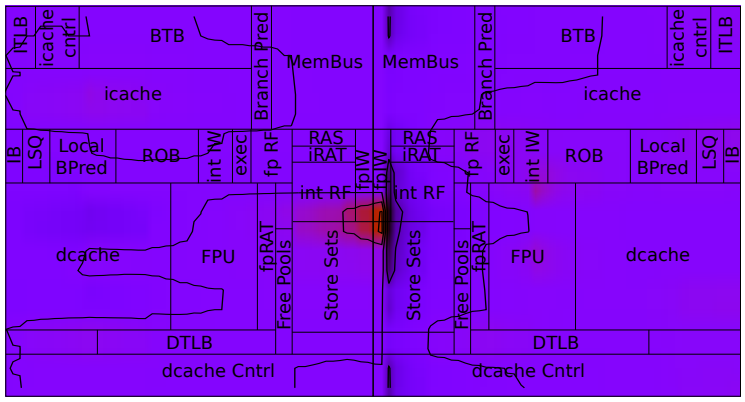
Another important issue is IR drop across the chip area. Two factors play a role in this, the first is the decreased current flowing through V_{dd}, which tend to reduce the drop. The second is, given a fixed budget for power rails, and the introduction of V_{mid}, the total amount of metal used by V_{dd} will decrease. Overall, we see a decreased IR drop, as shown in Figure 15 (the voltage was scaled for the baseline for better comparison), since the reduction in current is larger than the reduction on V_{dd} metal. The floorplan is partially shown over the figure, just as a reference.

In the case of multicore, we also evaluate the effects of swapping the cores, as proposed in Section 4.5, as opposed to not swapping them. As discussed, there is a small residual bias towards one partition due to the impossibility of exact balance. To evaluate it, we measure V_{mid} across a dual-core chip, with one core running mgrid and one running the astar benchmark. The *sVR* is disabled to allow for the analysis of the impact of the swapping only. Our experiments show that, swapping bring V_{mid} closer to the mid-point voltage across the chip, even for cores running different benchmarks, from an average of 0.7V to an average of around 0.95V. The results are shown in the heatmap in Figure 16.

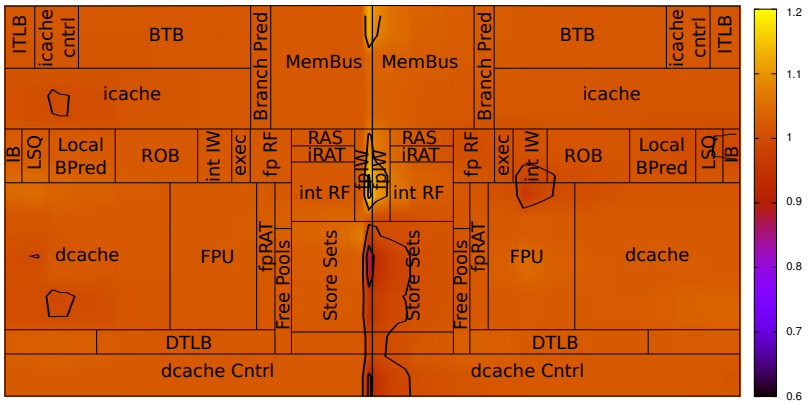


(a) Baseline (b) *CoreUnfolding*

Fig. 15: *CoreUnfolding* reduces IR drop across the chip area.



(a) No swap



(b) Swap

Fig. 16: In multicore configurations, swapping the *Header* and *Footer* clusters help to bring V_{mid} to balance across the chip area, when there is no swap, the small partitioning bias accumulates.

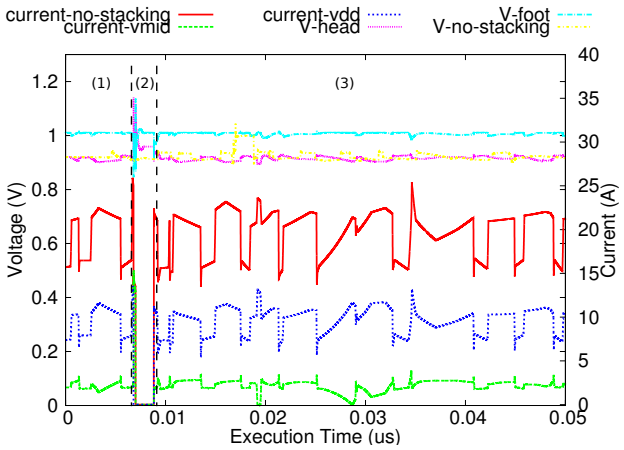


Fig. 17: *CoreUnfolding* allows for smaller inrush current. (1) mgrid execution. (2) Power gated. (3) execution continued.

6.6. Worst case and voltage noise analysis

One concern with *CoreUnfolding* is the behavior of the power delivery network during power up and down, in particular the magnitude of inrush current during this phases. To evaluate this, we use the same set-up from the previous section but consider start and stop of cores. This also simulates the case where the core is power gated and then turned back on. Figure 17 shows the transient analysis results for current and voltage for a core running mgrid benchmark, which is then powered down and back up (see marks on the figure), for both a stacked and a not-stacked configuration. The spike after re-start is more than two times bigger in the conventional not-stacked configuration than with *CoreUnfolding*, due the fact that the total current itself is smaller in *CoreUnfolding*.

There is a small difference between the V_{foot} and V_{head}, which is due to a small residual difference in the partitioning. This observation is compatible with previous finding in the context of Voltage Stacking [Lee et al. 2012]. $V_{no-stacking}$ is lower due to a higher IR drop, caused by the higher current, even considering less resources to V_{dd} and Gnd grids in the stacked configuration (due to the use of resources for V_{mid}).

Also note that counter intuition voltage noise (and droop) is smaller in the case of *CoreUnfolding*. This is, again, due to the smaller current in V_{dd}, which makes it more stable, but also due to the balancing achieved by the partitioning in *CoreUnfolding* and the presence of mechanisms to cap the maximum mismatch (DA and DVFS).

One interesting point is that the reduction in voltage noise could allow the reduction of voltage margins. Considering an initial voltage margin of 10% and 40% reduction in noise, and thus in voltage margin, additional 8% of total power could be possible. This would add to around 15-18% of total power savings. In our analysis, however, we keep the conservative analysis, not considering this extra gains.

7. CONCLUSION

This paper presents *CoreUnfolding*, a microarchitecture level technique to limit the load mismatch wanted by voltage stacking power delivery systems. To have a balanced load among stacks, *CoreUnfolding* 1) stacks the components inside a core instead of stacking the cores. Components within a core are grouped and stacked based on their instantaneous power correlation and average power magnitude. 2) The minimal mismatches that could still happen are managed by combination of DA, small secondary VR, and DVFS, depending on the magnitude of the mismatch. Even without DA, the proposed solution reduces the mismatch to about 6% of the total execution on average.

A key contribution of this paper is the capacity to reduce overall current in the power supply, and thus enabling smaller (estimated in 30%) and more efficient (estimated in 10%) VRs. This is possible by capping its usage through novel static mismatch reduction. Dynamic load balancing techniques used at runtime bring the minimal load mismatch to zero. As a result, total power can be reduced by up to 10% in our experiments due to the improved VR efficiency, and reduced I^2R power loss in the delivery network. Also both resistive (IR) and inductive (Ldi/dt) voltage noises are reduced. Reduction of the current also results in using less power dedicated pins.

The downside of *CoreUnfolding* is 2.2% decrease in the peak performance of the processor for single core applications, which could be compensated for by the reduced voltage noise. That is considering the on-chip VR power consumption does not cause thermal throttling. If such effects are taken into account, speed ups should be expected in several benchmarks. Overall, the presented *CoreUnfolding* system allows for power savings, area savings, and makes the usage of voltage stacking more feasible for future systems.

Future work may include the generalization to a n-level stack, with further savings, but that presents more challenging in the partitioning. Another possibility for modifying the scheme is to have an uneven partitioning, and thus changing the $2V_{dd}/V_{mid}$ ratio to an arbitrary value. This opens the opportunity to save power by having different parts of the core at different voltages, but could also impact performance.

REFERENCES

Christian Bienia, Sanjeev Kumar, Jaswinder Pal Singh, and Kai Li. 2008. The PARSEC benchmark suite: characterization and architectural implications. In *PACT '08: Proceedings of the 17th international conference on Parallel architectures and compilation techniques*. ACM, New York, NY, USA, 72–81. DOI:http://dx.doi.org/10.1145/1454115.1454128

E.A. Burton, G. Schrom, F. Paillet, J. Douglas, W.J. Lambert, K. Radhakrishnan, and M.J. Hill. 2014. FIVR – Fully integrated voltage regulators on 4th generation Intel®Core™SoCs. In *Applied Power Electronics Conference and Exposition (APEC), 2014 Twenty-Ninth Annual IEEE*. 432–439. DOI:http://dx.doi.org/10.1109/APEC.2014.6803344

E. Candan, P.S. Shenoy, and R.C.N. Pilawa-Podgurski. 2014. A series-stacked power delivery architecture with isolated differential power conversion for data centers. In *Telecommunications Energy Conference (INTELEC), 2014 IEEE 36th International*. 1–8. DOI:http://dx.doi.org/10.1109/INTLEC.2014.6972231

Leland Chang, Robert K Montoye, Brian L Ji, Alan J Weger, Kevin G Stawiasz, and Robert H Dennard. 2010. A fully-integrated switched-capacitor 2 1 voltage converter with regulation capability and 90% efficiency at 2.3 A/mm 2. In *VLSI Circuits (VLSIC), 2010 IEEE Symposium on*. IEEE, 55–56.

N. K. Choudhary, S. V. Wadhavkar, Tanmay A. Shah, Hiran Mayukh, Jayneel Gandhi, Brandon H. Dwiel, Sandeep Navada, Hashem H. Najaf-abadi, and Eric Rotenberg. 2011. FabScalar: Composing Synthesizable RTL Designs of Arbitrary Cores within a Canonical Superscalar Template. In *Proceedings of the 38th annual international symposium on Computer architecture (ISCA '11)*. ACM, New York, NY, USA, 11–22.

Xiangyu Dong, Jishen Zhao, and Yuan Xie. 2010. Fabrication cost analysis and cost-aware design space exploration for 3-D ICs. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 29, 12 (2010), 1959–1972.

J. Gu and C.H. Kim. 2005. Multi-story power delivery for supply noise reduction and low voltage operation. In *Proceedings of the 2005 international symposium on Low power electronics and design*. ACM, 192–197.

J. L. Henning. 2006. SPEC CPU2006 benchmark descriptions. *SIGARCH Comput. Archit. News* 34, 4 (Sept. 2006), 1–17.

Wenkang Huang, George Schuellein, and Danny Clavette. 2003. A scalable multiphase buck converter with average current share bus. In *Applied Power Electronics Conference and Exposition, 2003. APEC'03. Eighteenth Annual IEEE*, Vol. 1. IEEE, 438–443.

Linear Technology Inc. Dual 4A per channel DC/DC uModule regulator. (????). http://www.linear.com/product/LTM4614.

Linear Technology Inc. Dual 8A per channel DC/DC uModule regulator. (????). http://www.linear.com/product/LTM4616.

Linear Technology Inc. Ultralow Vin, 15A DC/DC uModule Regulator. (????).

Inc. Intel. 2014. Desktop 4th Generation Intel®Core™Processor Family, Desktop Intel Pentium®Processor Family, and Desktop Intel®Celeron®Processor Family. (2014).

Fujio Ishihara, Farhana Sheikh, and Borivoje Nikolic. 2004. Level conversion for dual-supply systems. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* 12, 2 (2004), 185–195.

J.L. Henning. 2000. SPEC CPU2000: Measuring Performance in the New Millenium. *IEEE Computer* 33, 7 (July 2000), 28–35.

E. K. Ardestani and J. Renau. 2013. ESESC: A Fast Multicore Simulator Using Time-Based Sampling. In *International Symposium on High Performance Computer Architecture (HPCA'19)*.

K. Kesarwani, C. Schaef, C.R. Sullivan, and J.T. Stauth. 2013. A multi-level ladder converter supporting vertically-stacked digital voltage domains. In *Applied Power Electronics Conference and Exposition (APEC), 2013 Twenty-Eighth Annual IEEE*. 429–434. DOI:http://dx.doi.org/10.1109/APEC.2013.6520245

W. Kim, D.M. Brooks, and G.Y. Wei. 2011. A fully-integrated 3-level DC/DC converter for nanosecond-scale DVS with fast shunt regulation. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*. IEEE, 268–270.

Wonyoung Kim, David Brooks, and Gu-Yeon Wei. 2012. A fully-integrated 3-level DC-DC converter for nanosecond-scale DVFS. *Solid-State Circuits, IEEE Journal of* 47, 1 (2012), 206–219.

Wonyoung Kim, Meeta S Gupta, Gu-Yeon Wei, and David Brooks. 2008. System level analysis of fast, per-core DVFS using on-chip switching regulators. In *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*. IEEE, 123–134.

S.K. Lee, D. Brooks, and G.Y. Wei. 2012. Evaluation of Voltage Stacking for Near-threshold Multicore Computing. In *Low Power Electronics and Design (ISLPED), 2012 IEEE International Symposium on*. ACM, 373–378.

Sae Kyu Lee, Tao Tong, Xuang Zhang, David Brooks, and Gu-Yeon Wei. 2015. A 16-Core Voltage-Stacked System with an Integrated Switched-Capacitor DC-DC Converter. In *IEEE Symposium on VLSI Circuits (VLSIC)* (2015-06-16). http://vlsiarch.eecs.harvard.edu/wp-content/uploads/2015/06/VLSI_2015_FVS_Submission_Final.pdf

Jingwen Leng, Yazhou Zu, Minsoo Rhu, Meeta Gupta, , and Vijay Janapa Reddi. 2014. GPUVolt: Modeling and Characterizing Voltage Noise in GPU Architectures. (2014).

S.R. Nassif. 2008. Power grid analysis benchmarks. In *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*. 376–381. DOI:http://dx.doi.org/10.1109/ASPDAC.2008.4483978

Gerard Villar Piqué and Henk Jan Bergveld. 2012. State-of-the-art of integrated switching power converters. In *Analog Circuit Design*. Springer, 259–281.

M. Popovich, A.V. Mezhiba, and E.G. Friedman. 2007. *Power Distribution Networks with On-Chip Decoupling Capacitors*. Springer. http://books.google.com/books?id=d9uHqcfvaP4C

S. Rajapandian, K.L. Shepard, P. Hazucha, and T. Karnik. 2006. High-voltage power delivery through charge recycling. *Solid-State Circuits, IEEE Journal of* 41, 6 (2006), 1400–1410.

Saravanan Rajapandian, Zheng Xu, and Kenneth L Shepard. 2005. Implicit DC-DC downconversion through charge-recycling. *Solid-State Circuits, IEEE Journal of* 40, 4 (2005), 846–852.

Yogesh K Ramadass, Ayman A Fayed, and Anantha P Chandrakasan. 2010. A fully-integrated switched-capacitor step-down DC-DC converter with digital capacitance modulation in 45 nm CMOS. *Solid-State Circuits, IEEE Journal of* 45, 12 (2010), 2557–2565.

C. Schaef and J.T. Stauth. 2015. Efficient Voltage Regulation for Microprocessor Cores Stacked in Vertical Voltage Domains. *Power Electronics, IEEE Transactions on* PP, 99 (2015), 1–1. DOI:http://dx.doi.org/10.1109/TPEL.2015.2426572

PS Shenoy, VT Buyukdegirmenci, AM Bazzi, and PT Krein. 2010. System level trade-offs of microprocessor supply voltage reduction. In *Energy Aware Computing (ICEAC), 2010 International Conference on*. IEEE, 1–4.

Pradeep S Shenoy, Igor Fedorov, Tyler Neyens, and Philip T Krein. 2011. Power delivery for series connected voltage domains in digital circuits. In *Energy Aware Computing (ICEAC), 2011 International Conference on*. IEEE, 1–6.

Infineon Technologies. High Performance DrMos TDA21220. (????).

Z. Toprak-Deniz, M. Sperling, J. Bulzacchelli, G. Still, R. Kruse, Seongwon Kim, D. Boerstler, T. Gloekler, R. Robertazzi, K. Stawiasz, T. Diemoz, G. English, D. Hui, P. Muench, and J. Friedrich. 2014. 5.2 Distributed system of digitally controlled microregulators enabling per-core DVFS for the POWER8TM microprocessor. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*. 98–99. DOI:http://dx.doi.org/10.1109/ISSCC.2014.6757354

Vishay Sprague 2013. *Solid Tantalum Surface Mount Chip Capacitors*. Vishay Sprague, Santa Clara, CA.

Jia Wei. 2004. *High frequency high-efficiency voltage regulators for future microprocessors*. Ph.D. Dissertation. Virginia Polytechnic Institute and State University.

Steven Cameron Woo, Moriyoshi Ohara, Evan Torrie, Jaswinder Pal Singh, and Anoop Gupta. 1995. The SPLASH-2 programs: characterization and methodological considerations. In *ISCA '95: Proceedings of the 22nd annual international symposium on Computer architecture*. ACM, New York, NY, USA, 24–36. DOI:http://dx.doi.org/10.1145/223982.223990

Yong Zhan and Sachin S Sapatnekar. 2008. Automated module assignment in stacked-Vdd designs for high-efficiency power delivery. *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 4, 4 (2008), 18.

Wei Zhao and Yu Cao. 2006. New generation of predictive technology model for sub-45 nm early design exploration. *Electron Devices, IEEE Transactions on* 53, 11 (2006), 2816–2823.

Xunwei Zhou, Pit-Leong Wong, Peng Xu, Fred C Lee, and Alex Q Huang. 2000. Investigation of candidate VRM topologies for future microprocessors. *Power Electronics, IEEE Transactions on* 15, 6 (2000), 1172–1182.