2018 51st Annual IEEE/ACM International Symposium on Microarchitecture

Voltage-Stacked GPUs: A Control Theory Driven Cross-Layer Solution for Practical Voltage Stacking in GPUs

An Zou* Jingwen Leng† Xin He* Yazhou Zu‡ Christopher D. Gill* Vijay Janapa Reddi‡§ Xuan Zhang*
*Washington University in St. Louis, †Shanghai Jiao Tong University
‡The University of Texas at Austin, §Harvard University

***Abstract*—More than 20% of the available energy is lost in *"the last centimeter"* from the PCB board to the microprocessor chip due to inherent inefficiencies of power delivery subsystems (PDSs) in today's computing systems. By series-stacking multiple voltage domains to eliminate explicit voltage conversion and reduce loss along the power delivery path, voltage stacking (VS) is a novel configuration that can improve power delivery efficiency (PDE). However, VS suffers from aggravated levels of supply noise caused by current imbalance between the stacking layers, preventing its practical adoption in mainstream computing systems. Throughput-centric manycore architectures such as GPUs intrinsically exhibit more balanced workloads, yet suffer from lower PDE, making them ideal platforms to implement voltage stacking. In this paper, we present a cross-layer approach to practical voltage stacking implementation in GPUs. It combines circuit-level voltage regulation using distributed charge-recycling integrated voltage regulators (CR-IVRs) with architecture-level voltage smoothing guided by control theory. Our proposed voltage-stacked GPUs can eliminate 61.5% of total PDS energy loss and achieve 92.3% system-level power delivery efficiency, a 12.3% improvement over the conventional single-layer based PDS. Compared to the circuit-only solution, the cross-layer approach significantly reduces the implementation cost of voltage stacking (88% reduction in area overhead) without compromising supply reliability under worst-case scenarios and across a wide range of real-world benchmarks. In addition, we demonstrate that the cross-layer solution not only complements on-chip CR-IVRs to transparently manage current imbalance and restore stable layer voltages, but also serves as a seamless interface to accommodate higher-level power optimization techniques, traditionally thought to be incompatible with a VS configuration.**

I. Introduction

Computers consume a non-trivial proportion of the total electrical energy both in the U.S. and globally [1], [2]. Due to the demand of data-intensive services, throughput-centric manycore processors such as graphic processing units (GPUs) are increasingly deployed in modern computer systems. Generally speaking, these systems exhibit high power ratings (in the range of a few hundred watts) and suffer poor power delivery efficiency (PDE) due to their high load currents [3].

An examination of PDE behavior reveals a provocative finding: transmitting and distributing electricity across tens or hundreds of miles in the grid to reach a power outlet incurs only 6% power loss [4], whereas delivering power across *"the last centimeter"* from the PCB board to the GPU chip can waste more than 20% of the power [5], [6], [7]. This indicates that improving GPU PDE yields tremendous economic savings and environmental benefits from a smaller carbon footprint.

However, energy loss in the power delivery subsystem (PDS) is difficult to eliminate. Two main inefficiencies are directly associated with power delivery: *voltage conversion loss* for converting the higher supply voltage at the board level to the lower supply voltage required by the microprocessor [8]; and *power delivery network (PDN) loss* for transferring electron charges from the off-chip power source to the distributed on-chip computing units [9], [10]. Both inefficiencies worsen with lower supply voltages, increased power density, and higher power ratings. Although various techniques have been proposed in prior work to reduce PDN loss by moving voltage regulation closer to the point-of-load [11], [12], they fail to address both inefficiencies simultaneously, and are thus unable to fundamentally close the efficiency gap.

Voltage stacking, also known as charge recycling [13] or multi-story power delivery [14], is a novel PDS configuration that allows efficient power delivery through a single high voltage source to multiple voltage domains stacked in series. Due to the inherent voltage division among the voltage domains, voltage stacking (VS) obviates the need for step-down voltage conversion and reduces the currents flowing through the PDN. Ideally, if the current loads from all the voltage domains are perfectly balanced, the input voltage is then evenly divided with no supply noise fluctuation, resulting in close to 100% theoretical PDE. Unfortunately, under realistic workloads, VS faces severe limitations due to exacerbated supply noise caused by current imbalance between stacked voltage domains [15], preventing wide-spread adoption in practical computing systems that require consistent and reliable operations.

In this paper, we propose a cross-layer approach to enabling *practical* voltage stacking in GPUs that can deliver improved PDE with guaranteed reliability against worst-case supply noise. In addition to supply reliability, two other major hurdles to overcome in a voltage-stacked system are high implementation cost due to area overhead and incompatibility with high-level power optimization techniques such as dynamic frequency scaling (DFS) and power gating (PG). Our research presented in this paper demonstrates that all three hurdles can be effectively addressed in a GPU platform, where thanks to its single-program multiple-data (SPMD) execution model, each stream multiprocessor naturally exhibits more balanced power consumption. Our proposed cross-layer approach curtails the steep area overhead associated with the circuit-only solution for VS by complementing charge-recycling integrated voltage regulators (CR-IVR) with control-theory-driven architectural support. Our analysis shows that voltage smoothing techniques such as dynamic issue width scaling and fake instruction injection are effective in suppressing low to middle frequency supply noise and thus alleviate the required regulating capacity of CR-IVR to save precious silicon area. This newly-added architectural support layer could also serve as an intermediate interface to accommodate higher-level power optimization techniques such as DFS and PG, which were previously thought to be incompatible with a voltage-stacked configuration.

In summary, this paper makes the following contributions:

- Our system-level evaluation demonstrates that power delivery efficiency as high as 92.3% can be achieved in a voltage-stacked GPU system across various real-world GPU benchmarks, eliminating 61.5% of total PDS energy loss and improving the conventional PDE by 12.3%.
- We present a cross-layer solution for practical VS imple-

978-1-5386-6240-3/18/$31.00 ©2018 IEEE
DOI 10.1109/MICRO.2018.00039

mentation in GPUs that leverages control-theory-driven voltage smoothing at the architecture level to suppress low-to-middle frequency supply noise, which complements CR-IVRs' regulating effect at high frequency. This cross-layer approach can effectively stabilize layer voltages and guarantee supply reliability under worst-case conditions with 88% lower area overhead.

- Our cross-layer solution incorporates a VS-aware hypervisor to seamlessly interface with higher-level power optimization and enable compatible collaborative operation between VS and other power management techniques such as DFS and PG with higher energy savings.

II. Background

In this section, we provide a brief overview of power delivery subsystems in modern computing systems and the state-of-the-art in their implementation. We then introduce the basic concept of voltage stacking and survey recent developments in voltage-stacked systems. Finally, due to the critical importance of supply reliability in VS, we review past research that studies supply reliability in conventional power delivery settings.

A. Power Delivery Subsystem

Due to technology scaling, supply voltages of modern processor chips are typically below $1V$, whereas the standard supply on the board is $3 \sim 5V$. A standard power delivery subsystem (PDS) consists of the step-down voltage regulation module (VRM) on the motherboard, as well as sockets, packages, off-chip decoupling capacitors and electrical connections at the board, package, and chip levels in the form of PCB traces, socket bumps and C4 bumps.

In the conventional PDS, voltage conversion using a step-down VRM happens on the board, and the lower supply required by the digital microprocessor is directly delivered to the chip. Two primary types of energy losses occur in that process. First, due to the inherent inefficiency of step-down VRMs, energy is lost during the voltage conversion. Next, energy is lost as the current consumed by the processor experiences a voltage drop against the resistive parasitics along the PDN path, resulting in energy loss in the form of resistive heating. Here, the processor current is also known as a *load current* and the voltage drop across PDN parasitic resistance is referred to as an *IR-drop*. Studies show that these two major inefficiencies can account for more than 20% combined loss of the total available energy that is supplied by the board level VRM. Because both inefficiencies worsen with higher load currents, processors with higher power ratings or under peak power operations suffer more degradation in their system-level PDE. Besides affecting the energy efficiency of the system, a PDS also determines system reliability due to its impact on supply noise. Most notably, the non-ideal dynamic fluctuations of on-chip supply voltage are largely induced by the parasitic resistance, inductance, and capacitance (RLC) in the PDN, which is discussed in more detail later in Section II-C.

One emerging PDS solution, single-layer IVR PDS, that can improve efficiency and enhance supply integrity is to move the voltage conversion closer to the point-of-load by employing integrated voltage regulators (IVR) [11], [16], [17]. Compared to conventional PDS, IVR enables lower IR-drop and more responsive and flexible voltage scaling [18], [19], [20], at a cost of extra die area of the IVRs.

B. Voltage Stacking

Voltage stacking (VS), also known as charge recycling [13] or multi-story power delivery [14] is an alternative power delivery approach that has the potential to fundamentally address both voltage conversion loss and PDN loss simultaneously. The basic concept is to series-stack many computational cores, in contrast with single-layer conventional PDS and single layer IVR PDS. VS can be intuitively understood as allowing electron charges to recycle through the stacking layers in series. The benefits of VS are two-fold: it eliminates the step-down conversion entirely and thus avoids conversion loss; at the same time, VS reduces PDN loss because (similar to single-layer IVR PDS) power can be delivered on chip at a higher voltage level, resulting in lower load currents along the PDN path [21], [22].

Under ideal conditions when all the cores (or other computing elements) have exactly balanced activities, and hence the same transient current demands, close to 100% efficiency can be achieved in a voltage-stacked system. Although such high PDE has been demonstrated in hardware prototypes with microcontroller cores running synthetic benchmarks [21], [22], practical VS in real computing systems with spatial and temporal activity mismatches remains extremely challenging. Aggravated supply noise caused by activity mismatches across voltage-stacked layers presents one of the most obstinate obstacles preventing VS adoption in real-world mainstream computing systems.

C. Supply Reliability

Supply noise refers to the fluctuation of the on-chip supply voltage. Fundamentally, it stems from the fact that electrons cannot be delivered instantaneously from the voltage source at the board to immediately satisfy the fast-changing load currents of various on-chip components. Both static IR-drop and dynamic Ldi/dt noise (and resonance noise in particular) contribute to the total supply fluctuation. Past studies have investigated supply noise's impact on system reliability [23], and characterized the contributions from different root causes. Generally speaking, the static IR-drop contribution can be effectively tamed by circuit techniques such as load line regulation [24], whereas the dynamic Ldi/dt contribution proves to be more dominant and intractable [25]. These insights have inspired various supply noise mitigation strategies [26], [27], [28]. Checkpoint-recovery methods can still allow voltage emergencies to happen, and can detect an error and re-execute, but they are not suitable to deal with frequent supply noise events or multi-core systems where re-execution may involve complex interactions; detection-throttle methods monitor supply noise signatures either through direct sensor measurements or predictions based on microarchitectural events, and then throttle the processor activities to mitigate large voltage droops; finally, compiler and runtime methods [29], [30] have been explored to eliminate voltage emergencies by optimizing static and dynamic code streams.

However, all the previous work focuses on single-layer conventional PDS and may not directly apply to the multi-layer PDS in a voltage-stacked system. Applying VS in GPUs presents additional supply reliability challenges. It has been shown that intricate voltage noise behaviors emerge in a GPU both spatially (local vs global) and temporally due to the interactions between streaming multiprocessors (SMs) [27]. In a voltage-stacked GPU, these correlated interactions will have even more exaggerated effects on supply voltage fluctuations because of the series connection between the vertically-stacked SMs as is evident from the empirical noise characterization using a voltage-stacked microcontroller core array [31], [32]. Until now, existing research resorts to circuit-only solutions by incorporating multi-output charge-recycling integrated voltage regulators (CR-IVR) to stabilize the layer voltages [33], which consumes significant die area and has yet to be rigorously evaluated under worst-case current imbalance scenarios.

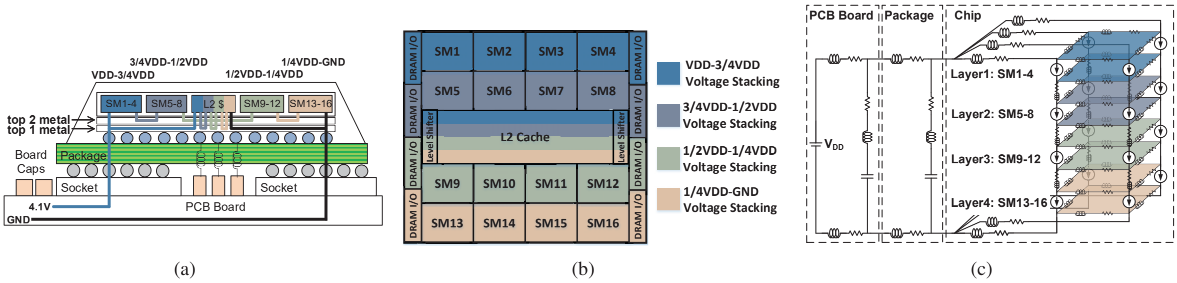


Fig. 1. Implementation of the voltage-stacked GPU system: (a) at the board level; (b) physical layout of the voltage-stacked domains on the die; (c) the corresponding electrical circuit model.

III. System Configuration and Circuit Solution

In this section, we describe the voltage stacked GPU system configuration and perform a rigorous examination of the circuit-only solution to implementing reliable voltage stacking in a GPU system. We start by presenting the implementation details of an example voltage stacked GPU based on realistic system configurations to fully account for potential performance degradation and design overhead. We then extend the effective impedance analysis as a rigorous method to characterize supply reliability in a VS setting. Finally, based on the impedance analysis, we derive the design parameters required to guarantee circuit-only reliable GPU VS operation and reveal the practical limitation of the circuit-only solution.

A. Voltage Stacking Implementation

One important motivation to explore VS in a GPU system is its single-program multiple-data (SPMD) execution model and homogeneous architecture, which naturally exhibits balanced and synchronous workload activities, as compared to the highly heterogeneous architecture and asynchronous workload in a CPU. In a SIMD system, all the cores execute the same code and experience very similar microarchitectural events, resulting in more balanced power traces across the cores. This makes voltage stacking a more appealing solution for power delivery, because most of the time the higher supply voltage is evenly divided among the layers, and given the balanced layer activities/currents, high (close to 100%) PDE can be achieved without frequent intervention from on-chip voltage regulators. To quantitatively evaluate this conjecture, we construct an example voltage-stacked GPU system using realistic system configurations and incorporating detailed VS implementation. This GPU VS system will be used throughout this paper as the baseline for GPU voltage stacking validation and evaluation.

System Configuration: The example GPU is modeled after NVIDIA Fermi [34], [35] to represent a typical manycore GPU architecture. It has 16 streaming multiprocessors (SMs) that share one L2 cache and off-chip DRAM [36]. Each SM has 32 shader cores, 16 load/store units, 4 special function units, and 64KB shared memory and L1 cache. These architecture details are summarized in Table I. To implement voltage stacking, a single high-voltage source ($4.1V$) is supplied at the board level and the on-chip components are partitioned into different voltage domains—SM1 to SM4 are in the V_{DD}-$3/4V_{DD}$ domain; SM5 to SM8 are in the $3/4V_{DD}$-$2/4V_{DD}$ domain; SM9 to SM12 are in the $2/4V_{DD}$-$1/4V_{DD}$ domain; SM13 to SM16 are in the $1/4V_{DD}$-GND domain; and the L2 cache and its interfaces with the SMs are also partitioned to four layers, based on a similar strategy from previous work on SRAM voltage stacking [37]. The power grid of SMs voltage stacking is separated from L2 cache voltage stacking. Our study focuses on the SM grid since its peak and average power account for 80% and 93% of the whole GPU.

TABLE I
Voltage stacked GPU System Configurations

Configuration	Value	Configuration	Value
PCB voltage	4.1V	SM voltage	1V
Number of SMs	16	SM clock freq.	700MHz
Threads per SM	1536	Threads per warp	32
Registers per SM	128 KB	Mem controller	FR-FCFS
Shared memory	48KB	Mem bandwidth	179.2GB/s
Memory channels	6	Warp scheduler	GTO [38]
VDD-$\frac{3}{4}$VDD	SM1-4	$\frac{3}{4}$VDD-$\frac{1}{2}$VDD	SM5-8
$\frac{1}{2}$VDD-$\frac{1}{4}$VDD	SM9-12	$\frac{1}{4}$VDD-GND	SM13-16
Process technology	40nm	PDN parameters	GPUvolt [39]

PDS Configuration: This novel VS PDS configuration is illustrated at the board level in Fig. 1(a), with the physical layout on the processor die in Fig. 1(b), and the corresponding electrical circuit model to simulate the supply fluctuation behaviors in Fig. 1(c). Here, we adopt the previous convention that models each SM as time-varying ideal current source, and use the same circuit parameters to model package, C4, and on-chip decoupling capacitances and parasitic resistance and inductance [5], [39]. Although 3D-IC implementation of VS has been studied in the past [40], our investigation focuses on its 2D implementation in a planar technology for a fair comparison with the conventional power delivery subsystem, as illustrated in Fig. 1(b). On a 2D planar chip, different voltage domains can be vertically stacked with minimal modifications to the topology of the on-chip power and ground routing. Rerouting to modify the power/ground grid from the single-layer conventional topology to the multi-layer VS topology is only required between the top metal layers and their connections to the C4 bumps, leaving the local power/ground grids in the lower metals and the physical floorplans of the underlying blocks largely intact. Assuming this minimally-invasive routing method, we can derive the corresponding electrical circuit model for the VS PDN [41] based on the typical RLC circuit equivalents and parameters used previously to study manycore systems [5], [39]. Fig. 1(c) depicts the electrical circuit model of the power delivery network for the proposed 4×4 voltage-stacked GPU system.

On-chip Regulation: The basic VS implementation illustrated in Fig. 1 does not include any regulation mechanism to stabilize the layer voltages. To remedy this problem, we employ charge-recycling integrated voltage regulators (CR-IVRs) on chip. The CR-IVR circuits are modeled using the symmetric ladder topology introduced in earlier VS prototypes [22], [33], and their basic operation can be intuitively understood as shuffling extra electrical charges from higher-voltage layers

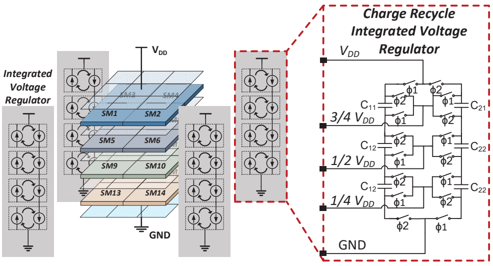


Fig. 2. Distributed charge-recycling IVR (CR-IVR) with four sub-IVRs whose outputs connect directly to each SM on each VS layer.

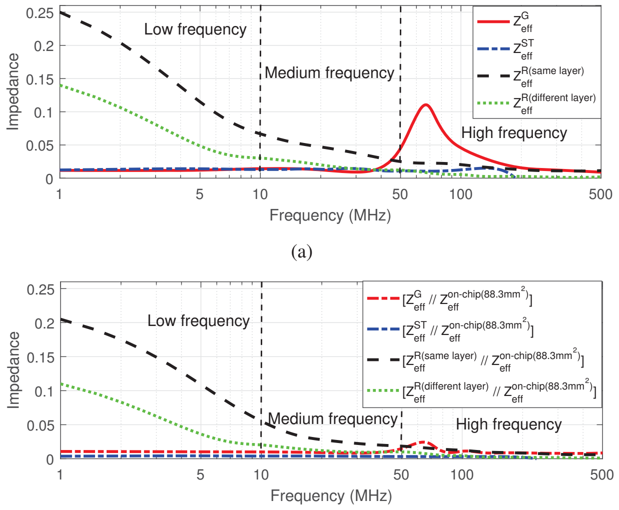
to lower-voltage layers by connecting flying capacitors to two consecutive layers in alternating switching phases. Prior study also finds that distributing the IVRs delivers better regulation [11]. Therefore, we implement a distributed CR-IVR with four sub-IVRs to maximize its supply noise suppression as illustrated in Fig. 2.

Level-shifted Interfaces: One important implementation detail to capture in a voltage-stacked system is the interfaces between the distinctive voltage domains. Since each voltage domain resides in a different voltage range, conventional level-shifter circuits do not readily apply at the interface. We account for the voltage-domain crossing in the example VS GPU system. It is worth noting that SMs do not directly communicate with each other in a GPU. Instead, messages are passed between the SMs through shared access to the L2 cache and/or memory interface to the off-chip DRAM. Therefore, the level-shifted interfaces reside at the input/output ports of L2 and memory controllers. Previous characterization estimates the level shifting overheads to be less than 6% of the total number of transistors in memory and cache [37], and evaluates several suitable level shifter circuits for a stacked architecture [42]. We choose to implement the switched-capacitor topology [33] as it has been shown to work at $1GHz$ signal transition speeds with the best energy-delay tradeoff [42].

B. Supply Reliability Characterization

Supply reliability is of uttermost importance in VS implementation. Any hardware-based solution should demonstrate reliable operation and guarantee well constrained supply noise under worst case scenarios. Although empirical results from selected benchmarks or synthetic microbenchmarks are often used to study and simulate transient supply waveforms in the context of conventional PDS [5], [39], they are insufficient to prove supply reliability given arbitrary load circumstances. To address worst case reliability in a rigorous manner, we use the analytical framework based on effective impedance analysis [5], [39] to characterize supply noise and obtain the necessary and sufficient conditions that guarantee supply reliability.

Effective impedance analysis: The PDS can be modeled as a pure passive RLC network. The electrical properties of these network circuit elements determine the impact of load current variations at different frequencies on supply voltages. For a given power delivery network, we can characterize its supply reliability by examining its impedance profile, which describes the PDS' frequency sensitivity. At the circuit level, the most effective approach to guarantee reliability is to suppress the peak effective impedance over the entire frequency range, so that even if all the load currents are concentrated at the peak frequency, the resulting peak voltage fluctuation is still confined within the allocated voltage guardband. We leverage the same effective impedance analysis to characterize the voltage-stacked GPU system. The impedance plot is obtained by applying a load current at a fixed frequency and observing the magnitude of the resulting voltage noise using the electrical circuit model of the VS PDS in Fig. 1(c). However, unlike conventional PDS, there exist multiple effective impedances in a VS PDS, depending on where the load current stimulus is placed and where the voltage noise is measured due to the multi-layer topology. As illustrated in Fig. 3(a), Z^G_{eff} refers to the effective complex impedance when the load current is evenly distributed across all the SMs, which we define as the global effective impedance; $Z^{ST}_{eff,i}$ refers to the impedance when the load current is evenly distributed across the i^{th} stack, which we call stack effective impedance; and $Z^R_{effi,j}$, defined as residual effective impedance, refers to the impedance when the only load current reside in a single SM (the j^{th} SM in the i^{th} stack), after subtracting its global and stack components. The rigorous derivations of these impedances based on circuit theory are studied in our previous work in [41] and the impedance plot is presented in Fig. 3. It reveals some important unique characteristics in voltage-stacked manycore systems. Without any on-chip regulator circuits, the VS GPU exhibit two impedance peaks shown in Fig. 3(a): one centers around $70MHz$ and represents the peak Z^G_{eff}, similar to the peak effective impedance of the single-layer conventional PDS that contributes to resonance noise; the other peak happens at DC on the Z^R_{eff} curve and is associated with the residual current that represents the current imbalance between the SMs in the same stack. Using the effective impedances of different current components, we can determine the combination of current stimuli that can generate the worst case supply noise [41]. Compared to the Z^G_{eff} peak, the Z^R_{eff} peak has much higher magnitude, and hence contributes the dominant component of the worst case noise behavior [41]. Our finding corroborates with earlier empirical work that qualitatively discusses the special role of current mismatch in disturbing layer voltages in VS [31], and suggests that the key to guarantee reliable VS operation is to tackle the imbalanced layer currents.



(a)

(b)

Fig. 3. Effective impedance plot of the example voltage-stacked GPU (a) without CR-IVR; (b) with CR-IVR that reduces impedance peaks.

Impact of on-chip regulation: The regulating effect of the on-chip CR-IVR can be intuitively explained by its impact on the effective impedance. By moving charges from the higher-voltage layers to the lower-voltage layers, the CR-IVR behaves as an additional parallel impedance (Z^{CR-IVR}) connected with the previous effective impedances (Z^{G}_{eff}, Z^{ST}_{eff}, and Z^{R}_{eff}). As a parallel impedance, it reduces the combined impedance and thus suppresses the resulting supply noise. This effect can be clearly observed in Fig. 3(b). The impedance peaks are suppressed by the CR-IVR and the larger the CR-IVR is, the lower Z^{CR-IVR} is and the lower the peaks are.

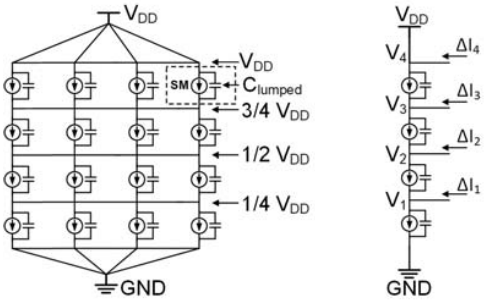
C. Limitation of the Circuit-only Solution

The effective impedance analysis allows us to derive the necessary and sufficient impedance condition to rigorously bound the magnitude of the worst case supply noise. In our example VS GPU, if the voltage guardband is set to $0.2V$[1], $912mm^2$ on-die area is required by the CR-IVR to suppress all the impedance peaks below 0.1Ω. This serious drawback of the circuit-only solution to the supply-noise challenges in voltage stacking becomes even more conspicuous when compared to other PDS configurations. To deliver power to the same GPU system with guaranteed supply reliability, VS with on-chip CR-IVR requires the largest die area ($912mm^2$), which is 1.7x the area of the GPU itself ($529mm^2$). Despite enjoying the highest power delivery efficiency (PDE), circuit-only VS implementation consumes prohibitively large area overheads and is not practical. We present a more quantitative and rigorous trade-off discussion in the evaluation section (Section VI), as summarized in Table III.

IV. Architectural Support for VS

The analysis in Section III suggests that although a circuit-only solution is able to achieve reliable operation, it is impractical for real-world VS implementation in GPUs due to area overhead. At the same time, some important insights are revealed—the highest impedance peak in a VS system happens in the low frequency range and contributes the largest supply fluctuations in the worst case scenarios. This finding opens the possibility for architecture-level techniques to suppress low frequency supply noise. In this section, we explore such opportunities by proposing control-theory-driven architectural support for VS.

The motivation to leverage control theory in the architecture-level technique is to provide strong guarantees of worst case behavior and control stability, which (unlike its conventional PDS counterpart) are necessary in a voltage stacked system. In a conventional system with single-layer PDS, the worst case supply noise is often induced by repetitive execution sequences or sudden trigger events near its peak resonance impedance. These execution activities over a short period of time (tens or hundreds of clock cycles) can be predicted and rearranged either at compile time or runtime. Such predictability does not readily apply to voltage stacking, however, because its impedance profile may exhibit a high plateau over a wide low frequency range due to the imbalanced residual current components that could span from hundreds to tens of thousands of clock cycles. In light of the intractability of the root causes of current imbalance/misalignment in the GPU, we resort to a control theory based approach to stabilizing the layer voltages in voltage stacking. We first present the control theoretic formulation of our proposed architectural support for VS, modeling the layer voltages as a four dimensional linear dynamic system. We then discuss the available voltage smoothing techniques and identify dynamic issue width scaling (DIWS), fake instruction injection (FII), and dynamic current compensation (DCC) as suitable actuation mechanisms. Finally, the detailed implementation is considered, to account for potential performance impacts and power/area overheads of the proposed techniques.



(a) (b)

Fig. 4. Simplified circuit of (a) the 4×4 VS GPU, (b) a single VS stack

A. Control Theoretic Formulation

In order to apply control theory to mitigate severe voltage droops caused by current imbalance, and to stabilize layer voltages, we model the on-chip power grid of the voltage-stacked GPU as a linear dynamic system and then formally derive the control strategy in response to the measured state of the system. Fig. 4(a) illustrates the simplified on-chip power grid of the example GPU system with 4×4 VS configuration. Here, we simplify the PDS by neglecting the parasitics impedance and assume an ideal $4V$ supply voltage (V_{DD}). We further simplify the model by only looking at the voltages and corresponding current terms in a single stack (or column) of the 4×4 array and ignoring the small parasitic on-chip inductance, as shown in Fig. 4(b). Assuming that the system reaches equilibrium when all the layer voltages are evenly divided and using that equilibrium point as the initial condition, we can write down the differential equation for each layer voltage at time t as:

$$V_i(t) = V_{i-1}(t) + \frac{1}{4}VDD + \frac{1}{C}\int_0^t (I_{i+1} - I_i + \Delta I_i)d\tau \qquad (1)$$

in which $V_i(t)$ represents the absolute voltage level at layer i. Assuming V_{DD} is an ideal voltage source, $V_4(t) = V_{DD}$ and is a constant value. The systems of equations depicted by (1) can be expressed in matrix form as:

$$\begin{bmatrix}\dot{V_1}\\ \dot{V_2}\\ \dot{V_3}\\ \dot{V_4}\end{bmatrix} = \begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}V_1\\V_2\\V_3\\V_{DD}\end{bmatrix} + \begin{bmatrix}\frac{-1}{C}&\frac{1}{C}&0&0\\ \frac{-1}{C}&0&\frac{1}{C}&0\\ \frac{-1}{C}&0&0&\frac{1}{C}\\ 0&0&0&0\end{bmatrix}\begin{bmatrix}I_1\\I_2\\I_3\\I_4\end{bmatrix} + \begin{bmatrix}\frac{\Delta I_1}{C}\\ \frac{\Delta I_2}{C}\\ \frac{\Delta I_3}{C}\\ 0\end{bmatrix} \qquad (2)$$

where I_i represents the current of the SM in the ith layer. Replacing I_i as the SM power (P_i) divided by the layer voltage across the SM, i.e., $I_i = \dfrac{P_i}{V_i - V_{i-1}}$, we have the dynamic system describing the relation between voltage and power as:

$$\begin{bmatrix}\dot{V_1}\\ \dot{V_2}\\ \dot{V_3}\\ \dot{V_4}\end{bmatrix} = \begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}V_1\\V_2\\V_3\\V_{DD}\end{bmatrix} + \begin{bmatrix}\frac{-1}{C}&\frac{1}{C}&0&0\\ \frac{-1}{C}&0&\frac{1}{C}&0\\ \frac{-1}{C}&0&0&\frac{1}{C}\\ 0&0&0&0\end{bmatrix}\begin{bmatrix}\frac{P_1}{V_1 - V_{GND}}\\ \frac{P_2}{V_2 - V_1}\\ \frac{P_3}{V_3 - V_2}\\ \frac{P_4}{V_4 - V_3}\end{bmatrix} + \begin{bmatrix}\frac{\Delta I_1}{C}\\ \frac{\Delta I_2}{C}\\ \frac{\Delta I_3}{C}\\ 0\end{bmatrix} \qquad (3)$$

Assuming small voltage disturbance, we can linearize the above system around its equilibrium point where $[V_1\ V_2\ V_3\ V_4]' =$

[1] $0.2V$ is the voltage margin used in commercial GPU systems to tolerate supply noise [43].

$[1\ 2\ 3\ 4]'$, resulting in the final linear dynamic system equation (4) which has the classic form (5).

$$\begin{bmatrix} \dot{V_1} \\ \dot{V_2} \\ \dot{V_3} \\ \dot{V_4} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_{DD} \end{bmatrix} + \begin{bmatrix} \frac{-1}{C} & \frac{1}{C} & 0 & 0 \\ \frac{-1}{C} & 0 & \frac{1}{C} & 0 \\ \frac{-1}{C} & 0 & 0 & \frac{1}{C} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} + \begin{bmatrix} \frac{\Delta I_1}{C} \\ \frac{\Delta I_2}{C} \\ \frac{\Delta I_3}{C} \\ 0 \end{bmatrix} \tag{4}$$

$$\dot{X} = AX + BU + \Delta F \tag{5}$$

where $X = [V_1 V_2 V_3 V_4]'$ is the state of the above linear dynamic system; A is the state matrix; B is the control input matrix; and $U = [P_1 P_2 P_3 P_4]'$ gives the SM power levels, which are the control inputs of the system. ΔF captures the current disturbance that incurs supply noise. We consider a classic proportional state feedback controller $U = KX$ as an illustrative example, as it is considered an effective stabilization technique with both computational advantages and satisfactory regulation results. In a proportional state feedback controller the SM power is a function of SM voltage:

$$P_i = kV_i \tag{6}$$

where k is the proportional feedback coefficient. Hence, the system with feedback control can be represented as:

$$\dot{X} = AX + BKX + \Delta F = (A + BK)X + \Delta F \tag{7}$$

B. Control Stability and Performance

The control delay plays an important role in determining the system stability and control performance in real applications. We express the total delay as T, which includes the sensor/actuator delay, communication and computation latencies along the feedback loop. We discretize the system with a sampling period of T:

$$X(n+1) = Z(A + BK)X(n) + \Delta F \tag{8}$$

where $Z(A + BK)$ is the discretization of matrix $A + BK$ with sampling rate T. The system model depicted by (8) suggests that $V_1V_2V_3$ is controllable and V_4 is equal to V_{DD}. We use MATLAB(R2018a) SIMULINK to examine the system dynamic response and select the proper coefficient k. It can be shown that the largest voltage deviations are caused by the worst case disturbance ΔF. When the disturbance frequency falls within half of the discrete system sampling frequency $\frac{1}{2T}$, the voltage deviations are guaranteed to be suppressed within a fixed range (i.e.$0.2V$), and a formal proof can be obtained by analyzing the Bode plot of the discrete system described by (8). In this way, we can rigorously prove that our control scheme is not only stable but also guaranteed to constrain the supply noise within the bound of the predetermined voltage margin. In addition to the theoretical proof, we are able to experimentally verify the systems stability and control performance under both worst case disturbances and representative benchmark workloads in Section VI-B.

In essence, formulating the on-chip VS power grid as a discrete-time linear dynamic system allows us to employ rigorous voltage smoothing mechanisms in the VS setting. The sampling rate T of the discretized system accounts for various latencies introduced by real implementations of the front-end detector, the controller, and the back-end actuator in real implementations. To effectively mitigate the dominant low frequency plateau exhibited by the effective impedance of the VS GPU, we need the total latency to be such that the low frequency peaks can safely fall within $\frac{1}{2T}$. Detailed choice of actuation mechanisms and implementation considerations of the voltage smoothing scheme are discussed next.

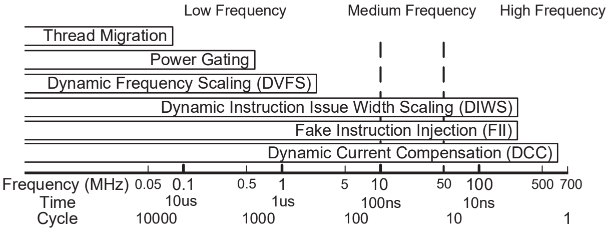

Low Frequency
Medium Frequency
High Frequency
Thread Migration
Power Gating
Dynamic Frequency Scaling (DVFS)
Dynamic Instruction Issue Width Scaling (DIWS)
Fake Instruction Injection (FII)
Dynamic Current Compensation (DCC)
Frequency (MHz) 0.05 0.1 0.5 1 5 10 50 100 500 700
Time 10us 1us 100ns 10ns
Cycle 10000 1000 100 10 1

Fig. 5. Timescales of different power actuation mechanisms.

C. Voltage Smoothing Actuation

As described by equation (4), the power consumption by the SM in each VS layer can be used as a control input, which suggests that any mechanism that actively modulates SM power can be considered a type of voltage smoothing actuator. Fig. 5 surveys several typical power management techniques in a GPU together with their respective response time scales. To achieve effective control, the actuator response time generally has to be at least an order of magnitude faster than the time scale of the relevant disturbance. In the case of a voltage-stacked GPU, we have shown earlier through impedance analysis that the part of the noise to be suppressed using architecture-level techniques is associated with the low frequency impedance caused by the residual current components. Therefore the maximum response time required of the voltage smoothing actuator is on the order of hundreds of clock cycles or around tens of *MHz*. Techniques such as *Thread Migration* [44], [45], [46] and *Power Gating* [47], [48] require content migration or state saving and operate at slower time scales (longer than 1000 clock cycles). The speed of *Dynamic Frequency Scaling* is determined by the re-locking time of the digital phase-locked loop (DPLL) and is typically on the order of *ms* [49], [50]. Our survey rules out these slower techniques and identifies three promising candidates for voltage smoothing actuation: *dynamic issue width scaling*, *fake instruction injection*, and *dynamic current compensation*.

Dynamic Issue Width Scaling (DIWS): In the Fermi architecture, each SM has a 2 warp/cycle issue width. Each warp includes 16 instructions. Either one or two warps can be dispatched in each cycle to any two of the four execution blocks within a Fermi SM—two blocks of 16 cores each, one block of four special function units (SFU), and one block of 16 load/store units (LSU), as shown in Fig. 6. To reduce the SM power, its warp issue width can be reduced, which can later be restored up to 2 warp/cycle, when voltage smoothing is no longer needed. One appealing advantage of DIWS is its low performance penalty when dynamically scaled. Before a warp is issued, the warp scheduler first checks with the scoreboard. Only when the warp is marked ready in the scoreboard, can it then be issued. Therefore, although each SM has a 2 warp/cycle issue width, the number of warps issued at each cycle varies at runtime. In our experiments with benchmarks from Rodinia and NVIDIA Cuda SDK, the average issue rate is $0.8-1.8$ warps per cycle due to data dependences, memory stalls, compute stalls and idle cycles. When DIWS is applied, even though the peak issue rate is reduced, which thus throttles the performance in certain cycles, it may result in more "ready" warps being accumulated in the warp pool. These accumulated "ready" warps can be issued opportunistically later, to fully occupy the issue width, offering a speedup that partly compensates for the performance loss from previous issues. For example in Fig.6, the warps in cycles 1 to i are issued without DIWS; during cycles i to $k-1$ DIWS sets the issue width to 1; and from

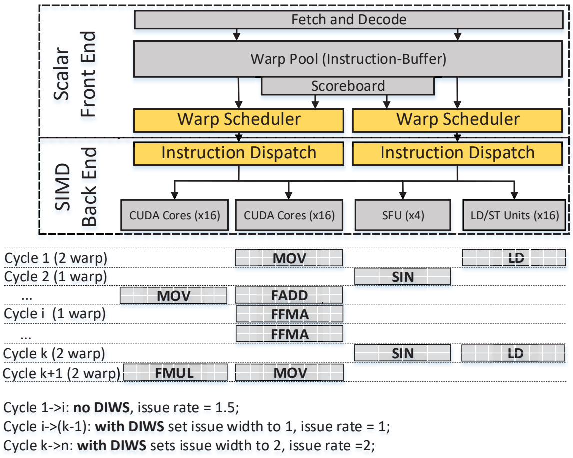


Fig. 6. SM microarchitecture and operation of dynamic issue width scaling.

cycles k to n the issue width is back up to 2.

Fake Instruction Injection (FII): Inserting fake instructions to fill up the issue width slack also can be used to introduce extra power consumption. Like DIWS, FII operates at the warp issuing speed, and thus has a fast response time. FII can leverage existing GPU architectures and does not require extra circuitry or die area to implement, but its availability is limited by the difference between the number of valid instructions and the maximum issue width at each cycle: when there are already two valid instructions in the warp pool, no extra instruction can be injected.

Dynamic Current Compensation (DCC): Finally, dummy digitally controlled current sources can be added on-chip to provide extra current/power and thus help balance the layer currents. We refer to this method as dynamic current compensation (DCC). While a similar method has been implemented using ring oscillator circuits [51], we employ binary-weighted current ladder circuits that are widely used as digital-to-analog converters (DACs). These DACs can be digitally controlled at runtime to compensate layer current imbalance at the time scale of a single clock cycle. Compared to DIWS and FII, deploying DCC requires extra die area and consumes more leakage power, and thus should be used sparingly to avoid energy and area penalties.

Weighted Control Inputs: Given that each of these three temporally suitable actuation mechanisms has its own merits and drawbacks, we consider a weighted linear combination of DIWS, FII, and DCC to exert the control inputs in equation (4). Therefore, the actual control inputs can be expressed as follows:

$$P_{SM} = w_1 P_{dyn,ins} \frac{IssueWidth}{max(IssueWidth)} + w_2 P_{dyn,ins} N_{FII} + w_3 P_{d0} N_{DCC} \tag{9}$$

where w_1, w_2, and w_3 are the respective weights for the power components of DIWS, FII, and DCC; $P_{dyn,ins}$ represents the dynamic power of the SM while executing the instruction *ins*; P_{d0} represents the unit power of the least significant bit (LSB) of the DCC current DAC; $N_{FII} \in 0, 1, 2$ is the number of fake instructions injected; and $0 \leq N_{DCC} \leq 2^{n_{DCC}}$ is the digital code that controls the n_{DCC}-bit current DAC to implement DCC. Formulating the control input as a weighted sum allows us to explore the design space of our proposed voltage smoothing method by sweeping different combinations for the same power effect, and to find optimal control strategies under different optimization objectives.

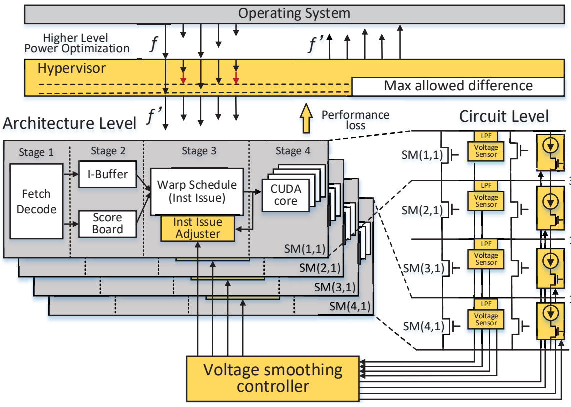


Fig. 7. Implementation of the proposed cross-layer VS GPU solution with architectural support for voltage smoothing and VS-aware PM hypervisor.

TABLE II
VOLTAGE DETECTOR OPTIONS

Sensor	Latency (cycle)	Power (mW)	Resolution (mV)	Output
ODDD	1-2	0-10	10-20	detect indicator
CPM	10-100	30-60	10-100	timing variation
ADC	1-10	10-100	$1/2^N$ V	N-bit digit signal

D. Implementation Considerations

A number of circuit-level and microarchitecture-level changes have to be made in a GPU system to accommodate the proposed control theory driven voltage smoothing technique. Fig. 7 illustrates the overall architecture to implement our scheme, which consists of the front-end detector and back-end actuator circuits, the voltage smoothing controller, and the VS-aware power management hypervisor.

1) Detector and actuator: To monitor spatial and temporal voltage fluctuations, front-end voltage detectors are placed close to each SM. A RC low pass filter is applied before the voltage detector to filter out high-frequency noise. The cutoff frequency of the filter is $\omega_c = 50MHz$ and it can be implemented with a $10K\Omega$ resistor and a $2pF$ capacitor, which together occupy $1120\mu m^2$ area. On-chip voltage detector circuits can be implemented in a number of ways using on-die droop detector (ODDD) [52], [53], [54], critical path monitor (CPM) [55], or analog digital converter (ADC) [56] approaches, as listed in Table II. All these voltage sensing/inference methods are compatible with the front-end detector requirements of our proposed scheme. The back-end actuators consist of the instruction issue adjuster embedded in the warp scheduler at each SM to support DIWS and FII, and the binary-weighted current DAC located near the load of each distributed CR-IVR to support DCC. The instruction issue adjuster arbitrates the instruction issue width and issues fake instructions to exert power actuation. Since each SM can issue up to two instructions per cycle, we can adjust the total number of instructions issued every N cycles to achieve finer-grained control resolution, from 1 to 1/N instructions per cycle on average. For instance, if the issue width is set to 1.7 instructions per cycle, it is adjusted by setting the down-counter that arbitrates the instruction issue to 17, with a reset every 10 cycles.

2) Voltage smoothing controller: The voltage smoothing controller executes the boundary triggered control algorithm using measured voltages from the detectors, and sends the updated issue width to the instruction issue adjuster. *Algorithm*

Algorithm 1: Streaming Multiprocessor Power Controller

```
Input: Measured voltage from voltage sensor $V_{(i,j)}$
Output: Issue Width: $Issue_{SM(i,j)}$, Fake Rate: $N_{fake-SM(i,j)}$
Procedure: The Controller
Read in measured voltage: $V_{(1,1)}...V_{(N_{layer},N_{column})}$;
for $(i \leq N_{layer}, j \leq N_{column})$ do:
    Calculate $SM_{(i,j)}$ voltage: $V_{SM(i,j)}$=$V_{(i,j)}$-$V_{(i-1,j)}$;
    if $(V_{SM(i,j)} < V_{threshold})$ then:
        Power control enable:
        $SM_{(i,j)} = active$; $nSM = nSM + 1$;
        $Issue_{SM(i,j)} = Issue_{max} - k_1 \times w_1 \times (1 - V_{SM(i,j)})$;
        $N_{fake-SM(i+1,j)} = k_2 \times w_2 \times (1 - V_{SM(i,j)})$;
        $P_{current-SM(i+1,j)} = k_3 \times w_3 \times (1 - V_{SM(i,j)})$;
        where $k_1, k_2, k_3$ are proportional control factors
    end if
end for    //finish a round of calculation
return $Issue_{SM(i,j)}$, $N_{fake-SM(i,j)}$, $P_{current-SM(i,j)}$
```

Algorithm 2:: VS-aware Power Management Hypervisor

```
Input: Command from OS: $f_{SM(i,j)}, gate_{SM(i,j)}$
Output: Command to SMs: $f^{'}_{SM(i,j)}, gate^{'}_{SM(i,j)}$
Procedure: Command Mapping
Read in operation system command:
$f_{SM(1,1)} \cdots f_{SM(N_{layer},N_{column})}, gate_{SM(1,1)} \cdots gate_{SM(N_{layer},N_{column})}$
for $(i \leq N_{layer}, j \leq N_{column})$ do:
    Calculate $\Delta f_{SM(i,j)}, \Delta p_{leakage-SM(i,j)}$:
    $\Delta f_{SM(i,j)}$=$f_{SM(i,j)}$-$f_{SM(i+N_{layer},j)}$;
    $\Delta p_{leakage-SM(i,j)}$=$p_{leakage-SM(i,j)}$-$p_{leakage-SM(i+N_{layer},j)}$;
    Update $f_{threshold\_SM(i,j)}$,$p_{threshold\_SM(i,j)}$;
    if $(|\Delta f_{SM(i,j)}| > f_{threshold\_SM(i,j)})$ then:
        Increase the frequency of $SM(i+N_{layer}, j)$:
        $f^{'}_{SM(i,j)} = min(f_{SM(\neq i,j)}) + f_{threshold\_SM(i,j)}$;
    end if
    if $(|\Delta p_{leakage-SM(i,j)}| > p_{threshold\_SM(i,j)})$ then:
        $gate^{'}_{SM(i,j)} = 0$
    end if
end for
return $f^{'}_{SM(i,j)}$, $gate^{'}_{SM(i,j)}$
```

shows an implementation of the proportional control algorithm. To reduce the negative effect of voltage smoothing on system performance, the controller is triggered by real-time supply noise measurements from the voltage detectors and only intervenes when a voltage droop below a certain threshold is detected. To evaluate the performance and overhead of the voltage smoothing controller accurately, we implement the controller and the SM instruction issue adjusters using VHDL[2]. We synthesize the VHDL code in the Synopsys Design Complier with TSMC 40nm technology, which is comparable to the process used in the NVIDIA Fermi GPU. The voltage smoothing controller and the 16 SM instruction issue adjusters in total consume $1.634mW$ power and occupy $3084\mu m^2$ area when operating at the same GPU frequency of $700MHz$. Finally, we account for control latency from several components: the detector response time, the controller computation time, the actuation delay, and the round-trip communication delay between the detector/actuator and the controller. We obtain the detector response time from previous work, calculate the controller computation time and the actuation delay based on our synthesized circuit model, and estimate the communication delay using an Elmore delay model based on tapered inverter buffer chains, assuming the controller is situated in the middle voltage stacking layer near the center of the SM.

3) VS-aware power management hypervisor: Due to voltage stacking's unique topology and constraints on layer current imbalance, previous VS studies have not thoroughly explored its compatibility with higher-level power optimization techniques such as dynamic frequency scaling (DFS) [57], [58], [59], [60] and power gating (PG) [61], [62], [63]. We consider the implications of collaborative power management in a voltage stacking setting and propose a voltage-stacking-aware hypervisor layer to interface with other power techniques. This hypervisor interface is added between the operating system layer and the GPU architecture layer as illustrated in Fig. 7. Since the voltage smoothing actuation mechanisms (DIWS, FII, and DCC) used in our cross-layer solution are orthogonal to the optimization mechanisms (frequency scaling and power gating) used in other techniques, we can accommodate these higher level mechanisms, which often operate over longer time scales, in the same control framework. The most significant impact of higher-level power management via frequency scaling and power gating on voltage stacking is that they may inadvertently introduce current imbalance, due to the different scaling/gating actions at the SM as determined by the power or performance optimization strategies. In terms of reliability, since these power-management-induced imbalances do not exceed the worst case imbalance analyzed previously, system reliability is still guaranteed by our control theory driven approach. However, a large imbalance could lead to undesirable energy loss associated with the on-chip CR-IVRs and performance penalties associated with throttling actions in the voltage smoothing mechanism. Here, we propose a heuristic optimization algorithm to constrain layer current imbalance and alleviate performance penalties as shown in *Algorithm* . The VS-aware hypervisor actively maintains balanced power across each voltage stack by preventing the frequency scaling and power gating requested by the power optimization techniques from exceeding a maximum power imbalance budget. The budget is dynamically adjusted according to the SM performance loss, which gauges how many instructions have been throttled due to voltage smoothing.

V. Evaluation Methodology

To evaluate our cross-layer solution approach, we develop an *integrated hybrid simulation infrastructure* that combines SPICE 3 [64] and GPGPU-Sim 3.1.1 (with GPUWattch) [65], [66], where SPICE 3 simulates the circuit level models for the VS PDS and the distributed CR-IVR as illustrated in Fig.1(c) and in Fig.2, and GPGPU-Sim simulates the architecture level systems specified in Table I. Our integrated simulation infrastructure instruments the interfaces between SPICE 3 and GPGPU-Sim and enables them to simulate synchronously. GPGPU-Sim generates the real-time power trace of each SM every clock cycle, and SPICE 3 takes this power trace and simulates the transient voltage at each node according to the netlist that depicts the VS PDS of the GPU system. A functional model of the voltage smoothing controller is built into the simulation infrastructure to compute the control input according to the simulated transient voltages after accounting for various

[2] https://github.com/xz-group/gpuvs.git

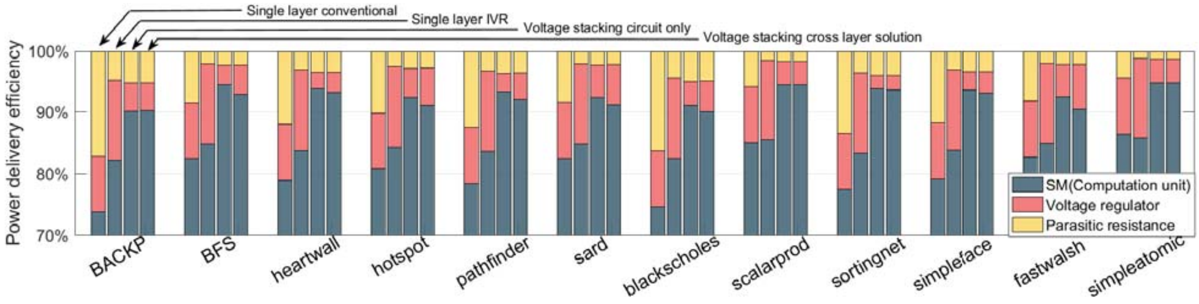


Fig. 8. Power delivery efficiency and power breakdown across benchmarks and power delivery subsystems configurations.

latencies. Based on the control input, the GPU simulator dynamically configures the instruction issue adjuster and current source to modify issue width, issue fake instructions and compensate the extra current. The integrated hybrid simulation infrastructure allows us to run real-world GPU benchmarks for our evaluation and we select twelve representative benchmarks that cover a wide range of scientific and computational domains from two benchmark suites–six from Rodinia2.0 [67] and six from NVIDIA CUDA SDK [36].

To evaluate the compatibility of our proposed cross layer VS solution with other higher-level power optimization techniques, we implement simplified versions of the control-theoretic dynamic frequency scaling strategy in GRAPE [57] and the power gating strategy in Warped Gates [62]. Similar to GRAPE, in our experiment, the frequency scaling step is set to $50MHz$ and each decision period is 4096 cycles. The dynamic frequency is implemented by masking the clock in GPGPU-Sim. In Warped gates, idle execution units inside the SMs (i.e. ALU, SFU, and LSU) are power gated to eliminate leakage power. We implement the gating-aware two-level warp scheduler (GATES) and Blackout gating scheme and evaluate power gating benefits using idle detect cycle and break-even cycle techniques in a similar manner [62].

VI. Evaluation Results

In this section, we quantitatively evaluate the efficiency, overhead, and reliability of our cross-layer voltage-stacked GPU system leveraging control theory. We first examine system-level power delivery efficiency and compare with alternative PDS configurations. The results indicate that our cross-layer voltage-stacked PDS is the only practical solution that can deliver power at 92.3% efficiency–12.3% improvement over the conventional PDS–without incurring prohibitive area overhead. Next, we evaluate the supply noise behavior of our solution against both synthetic worst-case scenarios and real-world benchmarks to verify that it can sustain the specified voltage margin with strong guarantees. We then perform a sensitivity study and design space exploration to reveal the potential performance and energy efficiency tradeoffs in the voltage-stacked GPU system. Finally, we demonstrate collaborative power management operations by combining the cross-layer VS framework with other higher-level power optimization techniques, which can can yield better overall system-level efficiency results than any of the individual methods alone.

A. *System-level Efficiency*

System-level power delivery efficiency (PDE) is evaluated by running a wide range of GPU benchmarks on our integrated hybrid simulation infrastructure. We compare our cross-layer VS solution with three alternatives: the conventional single-layer PDS with a board-level voltage regulator module (VRM), the single-layer IVR PDS with an on-chip switched-capacitor integrated voltage regulator but without voltage stacking, and the circuit-only solution to implement VS with the aid of on-chip charge-recycling IVR (CR-IVR).

TABLE III
Comparison of Different Power Delivery Subsystems (PDS)

PDS Configuration	PDE	Die Area Overhead
Single layer VRM [68]	80%	N/A
Single layer IVR [69]	85%	172.3mm^2 (0.33×GPU die)
VS circuit only [22], [70]	93.0%	912mm^2 (1.72×GPU die)
VS cross-layer	92.3%	105.8mm^2 (0.2×GPU die)

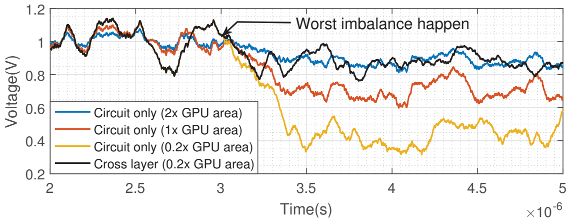


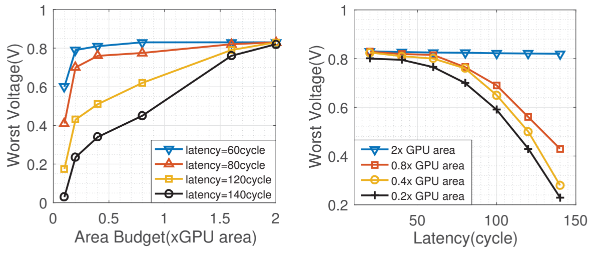
Fig. 9. Transient voltage waveforms under worst imbalance scenarios.

The normalized breakdown of the total system power across benchmarks is shown in Fig.8. On average, both voltage stacking PDS configurations (circuit-only and cross-layer) can deliver power at close to 92.3% efficiency, as compared to 80% for single-layer VRM (conventional baseline) and 85% for single-layer IVR. The reason that IVR in VS outperforms IVR in single-layer PDS is because the former only needs to shuffle the imbalanced load, which is usually less than 20%, of the layer power, whereas the latter delivers the total power.

Table III summarizes the comparison results. Besides efficiency, it also highlights different PDS configurations' die area overhead. Although both VS solutions exhibit high PDE, the circuit-only approach consumes excessive die area (1.72× the GPU die area) in order for the CR-IVR to have enough capacity to deal with the worst-case current imbalance. In contrast, our cross-layer approach that leverages architecture-level support to deal with the slow-changing part of the current imbalance appears to be the *only practical* solution known that can consistently achieve above 90% efficiency.

B. *Supply Reliability*

We first construct a synthetic worst case scenario to verify reliable operation of the proposed VS GPU. At the $3\mu s$ mark (Fig. 9), we manually turn off SMs in one layer to simulate extreme current imbalance. In the circuit-only VS systems, the voltage droop worsens as the CR-IVR area decreases and it takes about 2× the GPU area to stabilize the voltage above



(a)

(b)

Fig. 10. Worst supply noise in response to worst imbalance as a function of (a) CR-IVR area and (b) control latency.

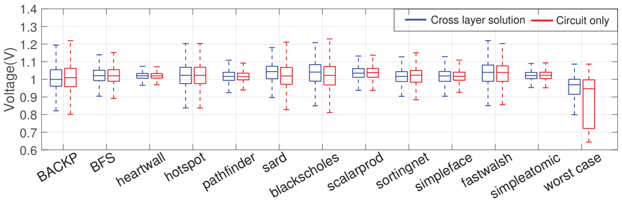


Fig. 11. Noise distribution across benchmarks and the worst-case imbalance.

0.8 V. Instead, our cross-layer solution incurs only 0.2× area overhead to achieve a similarly stable transient SM voltage, which is a nearly 90% area reduction.

We also perform a sensitivity study on the impact of CR-IVR area and control latency on the supply reliability of our cross-layer VS GPU. Fig.10 plots the worst voltage droop in response to the synthetic current imbalance event as a function of CR-IVR area (a) and control latency (b). In the left plot, when the control latency is greater than 80 cycles, the worst-case voltage droop becomes highly sensitive to the area budge. Similarly in the right plot, when the area budge is smaller than 0.8×, the worst-case voltage droop becomes highly sensitive to the control latency. Since the architecture-level voltage smoothing scheme can only deal with slow-changing supply fluctuations, a minimal-sized CR-IVR is always required to handle fast current imbalances. From the sensitivity analysis, we choose a 0.2× sized CR-IVR and a 60 cycle latency controller as the optimal parameters to implement the cross-layer VS solution, and use that default setting from now on. We also simulate the distribution of supply noise across real world benchmarks. Each box in Fig. 11 summarizes the noise distribution of all 16 SMs for a benchmark. We compare noise distribution between the cross-layer solution and the circuit-only solution, both with 0.2× sized CR-IVR. 9 out of 12 benchmarks experience modest reduction in voltage noise magnitude from the control theoretic voltage smoothing. The 3 outliers (*pathfinder, simpleatomic, fastwalsh*) are due to the choice of control parameters and boundary transitions, but their lowest voltage excursions are still bounded by $0.8V$, satisfying the specified $0.2V$ voltage margin. The rightmost box plot represents the worst-case noise distribution, which indicates that although the architecture voltage smoothing is only occasionally trigged for regular benchmarks, it is essential to provide the worst-case guarantee to ensure supply reliability.

C. Performance Tradeoffs

Due to the throttling nature of voltage smoothing mechanisms such as DIWS, our cross-layer approach inevitably incurs performance penalties. When evaluating energy efficiency of the GPU system, such performance penalties lead to longer total execution times and higher energy consumption caused by leakage power. We account for such performance penalties and their resulting increased leakage energy in our total energy savings calculation. The normalized performance penalty and net energy savings of our proposed cross-layer VS GPU is presented in Fig.14, normalized against the performance and total energy of conventional PDS with single-layer VRM. The performance penalty is distributed within $2\% - 4\%$ across benchmarks. After taking the extended execution time and increased leakage energy into account, voltage-stacked GPUs with the cross-layer solution still enjoy $10\% - 15\%$ net energy savings (improved energy efficiency) due to higher power delivery efficiency. We perform another sensitivity study by varying the voltage threshold ($V_{threshold}$) used in the voltage smoothing controller, as it determines how often DIWS throttling is triggered. The results across benchmarks are shown in Fig 12. A lower threshold leads to a smaller performance overhead, but jeopardizes supply reliability. In this paper, we set the default $V_{threshold}$ at $0.9V$, and at this level, less than 20% of the cycles are affected by voltage smoothing during benchmark execution when the layer voltage is below $0.9V$.

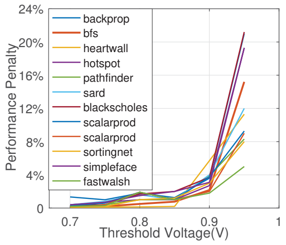


Fig. 12. Performance penalty varies with controller voltage threshold.

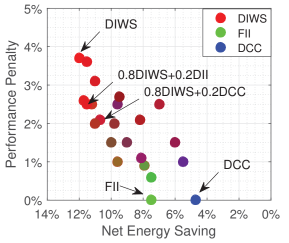


Fig. 13. Energy saving and performance penalty tradeoff space.

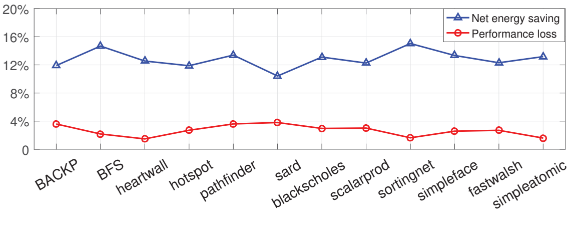


Fig. 14. Performance penalty and energy saving across benchmarks.

In the previous evaluation, we use only DIWS as the voltage smoothing mechanism, and the performance penalty is a result of its throttling effect. If an even smaller performance penalty is desired, our cross-layer approach has the flexibility to incorporate other mechanisms using the weighted control inputs as specified in (9). We explore the space of different weight combinations and the resulting performance penalty and net energy savings in Fig 13. On the Pareto frontier of the design space, we can see that when high net energy saving is desired, DIWS is generally the better voltage smoothing mechanism to choose, while FII and DCC can deliver a lower performance penalty. Due to its extra area overhead and leakage current, DCC is usually an inferior mechanism when FII can be applied to achieve similar performance.

D. Collaborative Power Management

Finally, we demonstrate the collaborative operation of voltage stacking with dynamic frequency scaling (DFS) and power gating (PG) for high-level power optimization. Previous DFS studies [57], [71] find the optimal SM operating frequencies to minimize the computational power under different performance

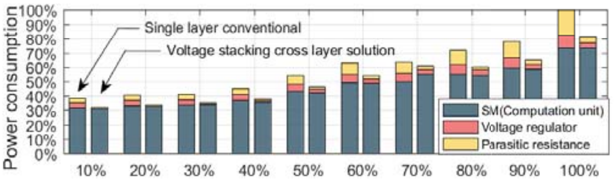


Fig. 15. Applying DFS on conventional and proposed voltage-stacked GPU.

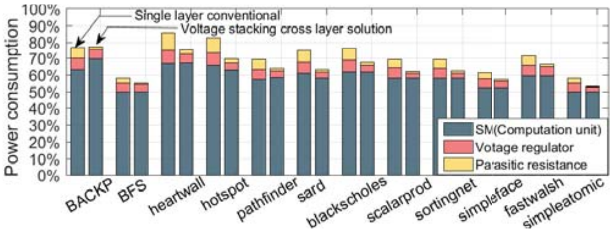


Fig. 16. Applying PG on conventional and proposed voltage-stacked GPU.

goals. We apply a similar DFS strategy and examine the total GPU energy consumption with and without VS. The energy in Fig.15 is normalized by the total GPU energy operating at its peak performance when the power delivery inefficiency is taken into account. Since our VS-aware power management hypervisor may modify the optimal frequency settings to ensure a bounded layer current imbalance, this negative effect of VS on DFS can be observed in the slight increase of computational energy (1-2%) in the second bar representing our cross-layer VS GPU solution. However, when the power delivery loss is considered, the slight energy penalty experienced by the VS GPU is more than compensated by its superior PDE, resulting in overall energy savings of 7-13% compared to applying DFS in the GPU with a conventional PDS. We observe similar results when combining PG techniques (i.e., Warped Gates [62]) with VS. As shown in Fig. 16, although the minimum current imbalance requirement in the VS GPU disrupts the optimal PG setting, it is more than compensated by improved PDE.

These favorable DFS and PG results can be better understood by carefully examining the distribution of imbalanced currents between two vertically stacked SMs across cycles. We normalize the current imbalance by the peak SM current and plot its distribution during the lifetime of the benchmark execution in Fig.17. When no power management is applied, the benchmark with the most imbalance is BACKUP (left bar) and the benchmark with the highest uniformity is heartwall (right bar). The middle bar presents the distribution averaged over all benchmarks and it shows that 50% of the time, the current imbalance is less than 10% of its peak magnitude, and 93% of the time, it is less than 40% of the peak. Similar exercises can be performed when DFS and PG are applied by evaluating the imbalance distribution for the worst, best, and average benchmarks. Fig.17 suggests that SM-level activities are overwhelmingly uniform and synchronized, resulting in well-balanced currents across the stack, and high-level power optimizations such as DFS and PG do not fundamentally disturb such balanced activities.

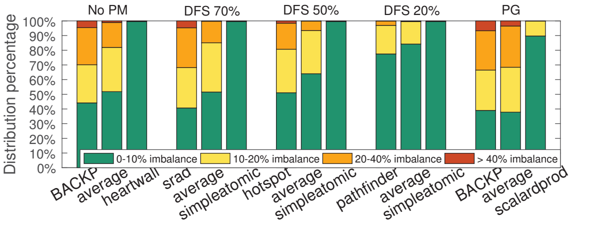


Fig. 17. Distribution of imbalanced currents by their normalized magnitudes when no power management (No PM), DFS with different performance goals, and power gating are applied in a VS GPU.

VII. Related Work

The feasibility of voltage stacking has been demonstrated previously in proof-of-concept circuits [14], [37] and silicon prototypes [13], [21], [22], [70], [72] using low-power microcontrollers. Design methodologies for floorplanning and placement [73], [74] also have been developed to facilitate its implementation. To overcome supply noise, most VS prototypes resort to employing integrated voltage regulators (IVRs) to actively balance current mismatches. However, they are limited to simple assemblies of uncorrelated cores with low power density, which often masks the fundamental limitation of prohibitively high area overhead in the circuit-only approach for voltage stacking. Built upon these early prototypes, a number of novel approaches have been proposed to take advantage of VS under different scenarios, such as 3D-IC with varying TSV, on-chip decoupling capacitance, and package parameters [40], [75], optimal system partitioning to unfold CPU cores [76], [77], and GPU systems with supercapacitors [51] and operation under near-threshold voltages to compensate process variation [77], [78]. While offering many interesting ideas about voltage stacking, this previous work does not rigorously address practical implementation of voltage stacking in a realistic manycore system, such as the GPU, using mature process technology with reliability guarantees against worst-case supply noise scenarios. Our work is also the first to address the compatibility of VS with other power optimization techniques. Control theoretic approaches have been proposed for dynamic resource management [57], [71] and reactive voltage emergency mitigation [53] in computing systems. Our control-theory-driven method leverages the synergistic collaboration between faster circuit-level voltage regulation and slower architecture-level voltage smoothing which to our knowledge has not been explored previously.

VIII. Conclusion

We propose a cross-layer solution for practical voltage stacking implementation in GPUs that are able to achieve: 92.3% system-level power delivery efficiency and 61.5% reduction of total PDS loss across a wide range of real-world benchmarks; guaranteed reliability against worst-case supply noise scenarios and over benchmarks while eliminating 88% area overhead compared to the circuit-only solution; and compatible collaborative operation between VS and high-level power optimization techniques with higher energy savings.

IX. Acknowledgement

The research described in this paper was partly supported by NSF CPS grant CNS-1739643, NSF award CCF-1528045, Semiconductor Research Corporation (SRC) task 2810.003 through the University of Texas at Dallas Texas Analog Center of Excellence (TxACE), the National Basic Research 973 Program of China 2015CB352403, and the National Natural Science Foundation of China (NSFC) 61702328. We are also grateful to the reviewers for their constructive feedback.

References

[1] Laura M Platchkov and Michael G Pollitt. The economics of energy (and electricity) demand. *The Future of Electricity Demand: Customers, Citizens and Loads*, 69:17, 2011.

[2] U.S. Energy Information Administration (EIA). Annual Energy Outlook 2016 with Projections to 2040. http://www.eia.gov/forecasts/aeo/.

[3] Jingwen Leng, Tayler Hetherington, Ahmed ElTantawy, Syed Gilani, Nam Sung Kim, Tor M Aamodt, and Vijay Janapa Reddi. Gpuwattch: enabling energy optimizations in gpgpus. In *ACM SIGARCH Computer Architecture News*, volume 41, pages 487–498. ACM, 2013.

[4] U.S. Energy Information Administration (EIA). FAQ:How much electricity is lost in transmission and distribution in the United States? http://www.eia.gov/tools/faqs/faq.cfm?id=105&t=3.

[5] Renji Thomas et al. Emergpu: Understanding and mitigating resonance-induced voltage noise in gpu architectures. In *ISPASS 2016*.

[6] Jingwen Leng, Yazhou Zu, and Vijay Janapa Reddi. Energy efficiency benefits of reducing the voltage guardband on the kepler gpu architecture. *Proc. of SELSE*, 2014.

[7] Haoran Li, Jiang Xu, Zhe Wang, Rafael KV Maeda, Peng Yang, and Zhongyuan Tian. Workload-aware adaptive power delivery system management for many-core processors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017.

[8] Rinkle Jain, Bibiche M Geuskens, Stephen T Kim, Muhammad M Khellah, Jaydeep Kulkarni, James W Tschanz, and Vivek De. A 0.45–1 v fully-integrated distributed switched capacitor dc-dc converter with high density mim capacitor in 22 nm tri-gate cmos. *IEEE Journal of Solid-State Circuits*, 49(4):917–927, 2014.

[9] Meeta S Gupta, Jarod L Oatley, Russ Joseph, Gu-Yeon Wei, and David M Brooks. Understanding voltage variations in chip multiprocessors using a distributed power-delivery network. In *Design, Automation & Test in Europe Conference & Exhibition, 2007. DATE'07*. IEEE, 2007.

[10] Intel Corp. Voltage Regulator Module, Enterprise Voltage Regulator-Down 10.0. http://www.intel.com/content/www/us/en/power-management/voltage-regulator-module-enterprise-voltage-regulator-down-10-0-guidelines.html.

[11] An Zou, Jingwen Leng, Yazhou Zu, Tao Tong, Vijay Janapa Reddi, David Brooks, Gu-Yeon Wei, and Xuan Zhang. Ivory: Early-stage design space exploration tool for integrated voltage regulators. In *Proceedings of the 54th Annual Design Automation Conference 2017*, page 1. ACM, 2017.

[12] Xuan Wang, Jiang Xu, Zhe Wang, Kevin J Chen, Xiaowen Wu, Zhehui Wang, Peng Yang, and Luan HK Duong. An analytical study of power delivery systems for many-core processors using on-chip and off-chip voltage regulators. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(9):1401–1414, 2015.

[13] Saravanan Rajapandian, Zheng Xu, and Kenneth L Shepard. Implicit dc-dc downconversion through charge-recycling. *IEEE journal of solid-state circuits*, 40(4):846–852, 2005.

[14] Pulkit Jain, Tae-Hyoung Kim, John Keane, and Chris H Kim. A multi-story power delivery technique for 3d integrated circuits. In *Low Power Electronics and Design (ISLPED), 2008 ACM/IEEE International Symposium on*, pages 57–62. IEEE, 2008.

[15] Sae Kyu Lee, David Brooks, and Gu-Yeon Wei. Evaluation of voltage stacking for near-threshold multicore computing. In *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, pages 373–378. ACM, 2012.

[16] Wonyoung Kim, Meeta S Gupta, Gu-Yeon Wei, and David Brooks. 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, 2008.

[17] Tao Tong, Xuan Zhang, Wonyoung Kim, David Brooks, and Gu-Yeon Wei. A fully integrated battery-connected switched-capacitor 4: 1 voltage regulator with 70% peak efficiency using bottom-plate charge recycling. In *Custom Integrated Circuits Conference (CICC), 2013 IEEE*.

[18] Xin He, Guihai Yan, Yinhe Han, and Xiaowei Li. Superrange: wide operational range power delivery design for both stv and ntv computing. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 146. European Design and Automation Association, 2014.

[19] Xin He, Gui-Hai Yan, Yin-He Han, and Xiao-Wei Li. Wide operational range processor power delivery design for both super-threshold voltage and near-threshold voltage computing. *Journal of Computer Science and Technology*, 31(2):253–266, 2016.

[20] Yunfei Gu, Dengxue Yan, Vaibhav Verma, Mircea R Stan, and Xuan Zhang. Sram based opportunistic energy efficiency improvement in dual-supply near-threshold processors. In *Proceedings of the 55th Annual Design Automation Conference*, page 41. ACM, 2018.

[21] Kristof Blutman, Ajay Kapoor, Arjun Majumdar, Jacinto Garcia Martinez, Juan Echeverri, Leo Sevat, Arnoud P van der Wel, Hamed Fatemi, Kofi AA Makinwa, and José Pineda de Gyvez. A low-power microcontroller in a 40-nm cmos using charge recycling. *IEEE Journal of Solid-State Circuits*, 52(4):950–960, 2017.

[22] Sae Kyu Lee, Tao Tong, Xuan Zhang, David Brooks, and Gu-Yeon Wei. A 16-core voltage-stacked system with adaptive clocking and an integrated switched-capacitor dc–dc converter. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(4):1271–1284, 2017.

[23] Xuan Zhang et al. Characterizing and evaluating voltage noise in multicore near-threshold processors. In *ISLPED*, 2013.

[24] Angel V Peterchev and Seth R Sanders. Load-line regulation with estimated load-current feedforward: Application to microprocessor voltage regulators. *IEEE Transactions on Power Electronics*, 2006.

[25] Patrik Larsson. di/dt noise in cmos integrated circuits. *Analog Integrated Circuits and Signal Processing*, 14(1-2):113–129, 1997.

[26] Vijay Janapa Reddi, Svilen Kanev, Wonyoung Kim, Simone Campanoni, Michael D Smith, Gu-Yeon Wei, and David Brooks. Voltage smoothing: Characterizing and mitigating voltage noise in production processors via software-guided thread scheduling. In *Microarchitecture (MICRO), 2010 43rd Annual IEEE/ACM International Symposium on*. IEEE, 2010.

[27] Jingwen Leng, Yazhou Zu, and Vijay Janapa Reddi. Gpu voltage noise: Characterization and hierarchical smoothing of spatial and temporal voltage noise interference in gpu architectures. In *High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on*, pages 161–173. IEEE, 2015.

[28] Renji Thomas, Kristin Barber, Naser Sedaghati, Li Zhou, and Radu Teodorescu. Core tunneling: Variation-aware voltage noise mitigation in gpus. In *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on*, pages 151–162. IEEE, 2016.

[29] Vijay Janapa Reddi, Simone Campanoni, Meeta S Gupta, Michael D Smith, Gu-Yeon Wei, David Brooks, and Kim Hazelwood. Eliminating voltage emergencies via software-guided code transformations. *ACM Transactions on Architecture and Code Optimization (TACO)*, 2010.

[30] Giang Hoang, Robby Bruce Findler, and Russ Joseph. Exploring circuit timing-aware language and compilation. In *ACM SIGPLAN Notices*, volume 46, pages 345–356. ACM, 2011.

[31] Sae Kyu Lee, David Brooks, and Gu-Yeon Wei. Evaluation of voltage stacking for near-threshold multicore computing. In *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, pages 373–378. ACM, 2012.

[32] Sae Kyu Lee, Tao Tong, Xuan Zhang, David Brooks, and Gu-Yeon Wei. A 16-core voltage-stacked system with an integrated switched-capacitor dc-dc converter. In *VLSI Circuits (VLSI Circuits), 2015 Symposium on*, pages C318–C319. IEEE, 2015.

[33] Tao Tong, Sae Kyu Lee, Xuan Zhang, David Brooks, and Gu-Yeon Wei. A fully integrated reconfigurable switched-capacitor dc-dc converter with four stacked output channels for voltage stacking applications. *IEEE Journal of Solid-State Circuits*, 51(9):2142–2152, 2016.

[34] NVIDIA. Nvidia's next generation cuda tm compute architecture: Fermi. https://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf.

[35] NVIDIA. Nvidia's fermi: The first complete gpu computing architecture. http://www.nvidia.com/content/PDF/fermi_white_papers/P.Glaskowsky_NVIDIA%27s_Fermi-The_First_Complete_GPU_Architecture.pdf.

[36] NVIDIA. Whitepaper nvidias next generation cudatm compute architecture: Fermi.

[37] Elnaz Ebrahimi, Rafael Trapani Possignolo, and Jose Renau. Sram voltage stacking. In *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, pages 1634–1637. IEEE, 2016.

[38] Timothy G Rogers, Mike O'Connor, and Tor M Aamodt. Cache-conscious wavefront scheduling. In *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE, 2012.

[39] Jingwen Leng, Yazhou Zu, Minsoo Rhu, Meeta Gupta, and Vijay Janapa Reddi. Gpuvolt: Modeling and characterizing voltage noise in gpu architectures. In *Proceedings of the 2014 international symposium on Low power electronics and design*, pages 141–146. ACM, 2014.

[40] Runjie Zhang, Kaushik Mazumdar, Brett H Meyer, Ke Wang, Kevin Skadron, and Mircea Stan. A cross-layer design exploration of charge-

recycled power-delivery in many-layer 3d-ic. In *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*. IEEE, 2015.

[41] An Zou, Jingwen Leng, Xin He, Yazhou Zu, Vijay Janapa Reddi, and Xuan Zhang. Efficient and reliable power delivery in voltage-stacked manycore system with hybrid charge-recycling regulators. In *Proceedings of the 55th Annual Design Automation Conference*, page 43. ACM, 2018.

[42] Elnaz Ebrahimi, Rafael Trapani Possignolo, and Jose Renau. Level shifter design for voltage stacking. In *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, pages 1–4. IEEE, 2017.

[43] Jingwen Leng, Alper Buyuktosunoglu, Ramon Bertran, Pradip Bose, and Vijay Janapa Reddi. Safe limits on voltage reduction efficiency in gpus: A direct measurement approach. In *Proceedings of the 48th International Symposium on Microarchitecture*, pages 294–307. ACM, 2015.

[44] Andrew Lukefahr, Shruti Padmanabha, Reetuparna Das, Faissal M Sleiman, Ronald Dreslinski, Thomas F Wenisch, and Scott Mahlke. Composite cores: Pushing heterogeneity into a core. In *Microarchitecture (MICRO), 2012 45th Annual IEEE/ACM International Symposium on*, pages 317–328. IEEE, 2012.

[45] Krishna K Rangan, Gu-Yeon Wei, and David Brooks. Thread motion: fine-grained power management for multi-core systems. In *ACM SIGARCH Computer Architecture News*, volume 37, pages 302–313. ACM, 2009.

[46] Miguel Rodrigues, Nuno Roma, and Pedro Tomás. Fast and scalable thread migration for multi-core architectures. In *Embedded and Ubiquitous Computing (EUC), 2015 IEEE 13th International Conference on*, pages 9–16. IEEE, 2015.

[47] Zhigang Hu, Alper Buyuktosunoglu, Viji Srinivasan, Victor Zyuban, Hans Jacobson, and Pradip Bose. Microarchitectural techniques for power gating of execution units. In *Proceedings of the 2004 international symposium on Low power electronics and design*. ACM, 2004.

[48] Manish Arora, Srilatha Manne, Indrani Paul, Nuwan Jayasena, and Dean M Tullsen. Understanding idle behavior and power gating mechanisms in the context of modern benchmarks on cpu-gpu integrated systems. In *High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on*, pages 366–377. IEEE, 2015.

[49] Jaehyun Park, Donghwa Shin, Naehyuck Chang, and Massoud Pedram. Accurate modeling and calculation of delay and energy overheads of dynamic voltage scaling in modern high-performance microprocessors. In *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design*, pages 419–424. ACM, 2010.

[50] Amir Bashir, Jing Li, Kiran Ivatury, Naveed Khan, Nirav Gala, Noam Familia, and Zulfiqar Mohammed. Fast lock scheme for phase-locked loops. In *Custom Integrated Circuits Conference. CICC'09. IEEE*, 2009.

[51] Qixiang Zhang, Liangzhen Lai, Mark Gottscho, and Puneet Gupta. Multi-story power distribution networks for gpus. In *Proceedings of the 2016 Conference on Design, Automation & Test in Europe*. EDA Consortium, 2016.

[52] Ali Muhtaroglu, Greg Taylor, and Tawfik Rahal-Arabi. On-die droop detector for analog sensing of power supply noise. *IEEE Journal of solid-state circuits*, 39(4):651–660, 2004.

[53] Russ Joseph, David Brooks, and Margaret Martonosi. Control techniques to eliminate voltage emergencies in high performance processors. In *High-Performance Computer Architecture, 2003. HPCA-9 2003. Proceedings. The Ninth International Symposium on*, pages 79–90. IEEE, 2003.

[54] Jaeha Kim and Mark A Horowitz. An efficient digital sliding controller for adaptive power-supply regulation. *IEEE Journal of solid-state circuits*, 37(5):639–647, 2002.

[55] Bruce Fleischer, Christos Vezyrtzis, Karthik Balakrishnan, and Keith A Jenkins. A statistical critical path monitor in 14nm cmos. In *Computer Design (ICCD), 2016 IEEE 34th International Conference on*, 2016.

[56] maxim integrated. Max19506 data sheet. https://www.maximintegrated.com/en/products/analog/data-converters/analog-to-digital-converters/MAX19506.html/.

[57] Muhammad Husni Santriaji and Henry Hoffmann. Grape: Minimizing energy for gpu applications with performance requirements. In *Microarchitecture (MICRO), 2016 49th Annual IEEE/ACM International Symposium on*, pages 1–13. IEEE, 2016.

[58] Pietro Mercati, Raid Ayoub, Michael Kishinevsky, Eric Samson, Marc Beuchat, Francesco Paterna, and Tajana Šimunić Rosing. Multi-variable dynamic power management for the gpu subsystem. In *Design Automation Conference (DAC), 2017 54th ACM/EDAC/IEEE*, pages 1–6. IEEE, 2017.

[59] Rong Ge, Ryan Vogt, Jahangir Majumder, Arif Alam, Martin Burtscher, and Ziliang Zong. Effects of dynamic voltage and frequency scaling on a k20 gpu. In *Parallel Processing (ICPP), 2013 42nd International Conference on*, pages 826–833. IEEE, 2013.

[60] Onur Kayıran, Adwait Jog, Mahmut Taylan Kandemir, and Chita Ranjan Das. Neither more nor less: optimizing thread-level parallelism for gpgpus. In *Proceedings of the 22nd international conference on Parallel architectures and compilation techniques*. IEEE Press, 2013.

[61] Po-Han Wang, Chia-Lin Yang, Yen-Ming Chen, and Yu-Jung Cheng. Power gating strategies on gpus. *ACM Transactions on Architecture and Code Optimization (TACO)*, 8(3):13, 2011.

[62] Mohammad Abdel-Majeed, Daniel Wong, and Murali Annavaram. Warped gates: gating aware scheduling and power gating for gpgpus. In *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 111–122. ACM, 2013.

[63] Yue Wang, Soumyaroop Roy, and Nagarajan Ranganathan. Run-time power-gating in caches of gpus for leakage energy savings. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012*, pages 300–303. IEEE, 2012.

[64] Thomas Quarles, AR Newton, DO Pederson, and A Sangiovanni-Vincentelli. Spice3 version 3f3 users manual, 1994.

[65] Ali Bakhoda, George L Yuan, Wilson WL Fung, Henry Wong, and Tor M Aamodt. Analyzing cuda workloads using a detailed gpu simulator. In *Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on*, pages 163–174. IEEE, 2009.

[66] Jingwen Leng, Tayler Hetherington, Ahmed ElTantawy, Syed Gilani, Nam Sung Kim, Tor M. Aamodt, and Vijay Janapa Reddi. Gpuwattch: Enabling energy optimizations in gpgpus. In *Proceedings of the 40th Annual International Symposium on Computer Architecture*, ISCA '13, pages 487–498, New York, NY, USA, 2013. ACM.

[67] Shuai Che, Michael Boyer, Jiayuan Meng, David Tarjan, Jeremy W Sheaffer, Sang-Ha Lee, and Kevin Skadron. Rodinia: A benchmark suite for heterogeneous computing. In *International Symposium on Workload Characterization*, 2009.

[68] https://www.infineon.com/. Digital multi-phase gpu buck controller.

[69] Edward A Burton, Gerhard Schrom, Fabrice Paillet, Jonathan Douglas, William J Lambert, Kaladhar Radhakrishnan, and Michael J Hill. Fivrfully integrated voltage regulators on 4th generation intel® core socs. In *Applied Power Electronics Conference and Exposition (APEC), 2014 Twenty-Ninth Annual IEEE*, pages 432–439. IEEE, 2014.

[70] Tao Tong, Sae Kyu Lee, Xuan Zhang, David Brooks, and Gu-Yeon Wei. A fully integrated reconfigurable switched-capacitor dc-dc converter with four stacked output channels for voltage stacking applications. *IEEE Journal of Solid-State Circuits*, 51(9):2142–2152, 2016.

[71] Abhinandan Majumdar, Leonardo Piga, Indrani Paul, Joseph L Greathouse, Wei Huang, and David H Albonesi. Dynamic gpgpu power management using adaptive model predictive control. In *2017 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 613–624. IEEE, 2017.

[72] Kristof Blutman, Ajay Kapoor, Arjun Majumdar, Jacinto Garcia Martinez, Juan Echeverri, Leo Sevat, Arnoud Van Der Wel, Hamed Fatemi, José Pineda de Gyvez, and Kofi Makinwa. A microcontroller with 96% power-conversion efficiency using stacked voltage domains. In *VLSI Circuits (VLSI-Circuits), 2016 IEEE Symposium on*. IEEE, 2016.

[73] Kristof Blutman, Hamed Fatemi, Andrew B Kahng, Ajay Kapoor, Jiajia Li, and José Pineda de Gyvez. Floorplan and placement methodology for improved energy reduction in stacked power-domain design. In *Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific*, pages 444–449. IEEE, 2017.

[74] Kristof Blutman, Hamed Fatemi, Ajay Kapoor, Andrew B Kahng, Jiajia Li, and José Pineda de Gyvez. Logic design partitioning for stacked power domains. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017.

[75] Kaushik Mazumdar and Mircea Stan. Breaking the power delivery wall using voltage stacking. In *Proceedings of the great lakes symposium on VLSI*, pages 51–54. ACM, 2012.

[76] Ehsan K Ardestani, Rafael Trapani Possignolo, Jose Luis Briz, and Jose Renau. Managing mismatches in voltage stacking with coreunfolding. *ACM Transactions on Architecture and Code Optimization (TACO)*, 12(4):43, 2016.

[77] Rafael T. Possignolo. Gpu ntc process variation compensation with voltage stacking. In *Parallel Architectures and Compilation Techniques (PACT), International Conference on.*, 2015.

[78] Rafael Trapani Possignolo, Elnaz Ebrahimi, Ehsan Khish Ardestani, Alamelu Sankaranarayanan, Jose Luis Briz, and Jose Renau. Gpu ntc process variation compensation with voltage stacking. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, (99):1–14, 2018.