2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA) | 979-8-3503-2658-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISCA59077.2024.00063

BlitzCoin: Fully Decentralized Hardware Power Management for Accelerator-Rich SoCs

Martin Cochet[1], Karthik Swaminathan[1], Erik Loscalzo[2], Joseph Zuckerman[2], Maico Cassel dos Santos[2], Davide Giri[2],
Alper Buyuktosunoglu[1], Tianyu Jia[3], David Brooks[3], Gu-Yeon Wei[3], Kenneth Shepard[2], Luca P. Carloni[2], and Pradip Bose[1]
[1]IBM Research, Yorktown Heights, NY [2]Columbia University, New York, NY [3]Harvard University, Cambridge, MA

***Abstract*—On-chip power-management techniques have evolved over several processor generations. However, response time and scalability constraints have made it difficult to translate existing power-management strategies to current or next-generation System-on-Chip (SoC) architectures, which are expected to comprise tens to hundreds of cores and accelerators. In this work we present *BlitzCoin*, a fully decentralized hardware power-management strategy for large, accelerator-rich SoCs, coupled with optimized unified voltage and frequency regulation. We evaluated BlitzCoin through RTL simulations of multiple SoCs targeted toward different application domains. The results are further validated through silicon measurements of a fabricated 12 nm many-accelerator SoC that includes BlitzCoin. Our evaluations show that BlitzCoin is markedly faster, with 8× to 12× lower response times, which provides 25%-34% throughput improvement and allows for scaling to 7× to 13× larger SoCs compared to state-of-the-art centralized power-management strategies, all with an area overhead of <1%.**

I. Introduction

Chip power management is a mature field with over two decades of research and development work, which has resulted in on-chip deployments included in several generations of commercial processors in the market today [1]–[8]. In particular, workload-driven multi-core power management has been the focus of several industrial products and research papers, from Intel [1], [9]–[13], IBM [14]–[21], AMD [22]–[25] and Qualcomm [26], as well as other works from academia [27]–[40]. However, contrary to popular belief that there remains no more room for innovation in this area, recent paradigm shifts necessitate rethinking and reinventing power management strategies moving forward.

SoC architectures have rapidly evolved to use an increased number of specialized hardware units and fixed-function accelerators on a single chip. This evolution in SoC architectures has come with increasingly complex applications not only in terms of the number and type of processing elements (PEs) utilized but also because of high dynamic variation in the utilization of the PEs. Modern-day accelerator-rich SoCs, such as the Apple Mobile A-series, devote over 60% of the chip area to such specialized IP blocks in addition to traditional CPU/GPU cores [41]. These blocks include accelerators for pixel processing, depth and motion estimation, deep learning, and more. Largely driven by the slowdown of technology scaling, this trend does not appear to be ending anytime soon.

Due to the diversity of power profiles of processing elements in heterogeneous SoCs, power-management strategies for such systems must provide a large range of fine-grained power states. Furthermore, different power-allocation strategies should be supported to accommodate varying applications and operating modes. As further detailed in Section VII, the prior art targeting homogeneous SoCs does not fully address these requirements.

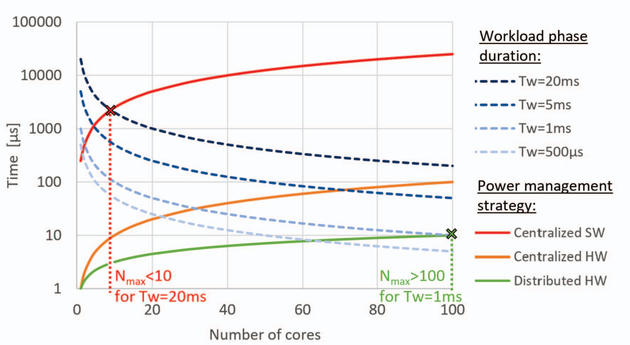


Fig. 1. Scalability of different power-management strategies in terms of their response time (solid lines) and the average interval between SoC-level activity change (dashed lines) for different accelerator-level workload phase durations T_w. The intersection of a solid line and a dashed line represents the maximum number of accelerators N_{max} that can be supported for a given workload phase duration and power-management strategy.

In order for an SoC to effectively utilize the available energy, its power management must react faster than the typical interval at which a workload changes its phase (T_w). Fig. 1 presents scaling trends for the rate of workload changes in a multi-accelerator SoC architecture (dashed lines) and response time trends (solid lines) for different power-management strategies. Here, response time is defined as the time for the power-management unit to react to a change in activity and adjust the power allocation of all accelerators accordingly. For example, if for a given accelerator a workload starts or ends on average once per $T_w = 5ms$, then for an SoC with $N = 20$ accelerators, a new power-management decision will be required, on average, once every $T_w/N = \frac{1}{20}.5ms = 250\mu s$.

Conventional software-based, centralized power-management schemes [18], [39] have response times on the order of 1 ms for a small number of accelerators and scale linearly with the number of accelerators. As the response time required to adapt to changes in workload phase (T_w, dashed curves) decreases with increasing number of accelerators, a centralized software-management implementation (red curve, response time proportional to N) cannot even scale up to 10 accelerators for a workload with $T_w \leq 20ms$.

Most modern processors have on-chip controllers containing specialized hardware/firmware units embedded within the chip. They are responsible for monitoring power and temperature sensors and adjusting the voltage and frequency of each PE to maximize performance within power constraints. They have significantly reduced re-

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

979-8-3503-2658-1/24/$31.00 ©2024 IEEE
DOI 10.1109/ISCA59077.2024.00063

sponse times compared to software controllers. However, due to the reliance on a single centralized hardware power-management unit, the response time still scales linearly with the number of accelerators (orange curve). Consequently, a centralized on-chip controller strategy with firmware-controlled power management would be rendered ineffective, since it would not be capable of rapid actuation every few μs for large values of N. Furthermore, it would limit the scalability of the system to a few tens of accelerators, as response-time constraints and network contention would be exacerbated in larger systems.

In contrast, in a decentralized approach, pairs of accelerators can exchange information in parallel across a 2-D mesh, providing a faster response than centralized schemes. A decentralized hardware approach, then, combines the fast μs-scale response time of a centralized hardware unit with sub-linear response time scaling in the number of accelerators (green curve). As seen on the graph, decentralized power management can handle workloads with timescales on the order of 1 ms for large SoCs (N≥100).

In this paper, we propose *BlitzCoin, a fully-decentralized power-management architecture for accelerator-rich SoCs*. Section II provides an overview of centralized and decentralized power management in heterogeneous SoCs, while in Section III, we present BlitzCoin at an algorithmic level and conduct a thorough design-space exploration through a variety of simulation-based experiments, showing a response time that scales with $\sqrt{N}$.

Section IV describes the hardware implementation of BlitzCoin. A decentralized approach necessitates compact and fast-responding voltage and frequency regulation. For this work, we further require a regulation approach that is easy to replicate across the heterogeneous tiles of a many-accelerator SoC. These requirements informed the design of a *Unified Voltage and Frequency Regulation* (UVFR) scheme. We implemented the control algorithm in RTL and also designed the UVFR scheme targeting a 12 nm technology node. We then integrated BlitzCoin with the whole per-tile DVFS implementation into an open-source many-accelerator SoC platform to allow for full-system experiments using BlitzCoin.

Section V outlines our experimental methodology. We leveraged full-SoC RTL simulation to conduct an in-depth analysis of our implementation. However, BlitzCoin relies on complex interactions between software workloads, digital logic, and analog components. Such complexity cannot be fully accurately characterized with existing circuit and architecture simulation tools. Only a silicon implementation can fully validate the approach. Hence, the capstone of this work is the integration of BlitzCoin into a complex heterogeneous SoC prototype fabricated in a 12 nm technology. Section VI shows the simulated results and their validation with the fully-functional silicon. *To the best of our knowledge, this is the first hardware implementation of a decentralized power-management strategy in a heterogeneous SoC.*

In summary, we make the following contributions:

• BlitzCoin, a decentralized power-management strategy for heterogeneous SoCs that is based on a novel coin-exchange algorithm.

• A design-space exploration of BlitzCoin in order to determine the configuration parameters that yield optimized results in terms of convergence time and power allocation.

• A demonstration of tile-level DVFS, in conjunction with BlitzCoin, consisting of the combination of a LDO regulator and a free-running oscillator for unified voltage and frequency regulation.

• A complete implementation of BlitzCoin, integrated into many-accelerator SoCs using an open-source hardware platform, validated by silicon measurements carried out on a 64 mm^2 SoC fabricated in 12 nm technology.

• An evaluation of BlitzCoin using different workloads and power-allocation strategies; the results demonstrate superior response time (up to 12× faster) and throughput (up to 34% better), compared to state-of-the-art centralized power-management strategies [42], [43], and scalability to SoCs with hundreds of accelerators.

II. Background

A. Power-Management Challenges in Heterogeneous SoCs

Most academic and commercial work has been focused on addressing workload-driven multi-core power management for homogeneous CPU-centric SoCs [34], [40], [44]–[46]. However, heterogeneous SoC power management presents significant additional challenges. First, a wide range of power consumption across heterogeneous accelerators, up to 10× as observed in [47], requires the power-management unit to actuate at a fine granularity, as well as span a wide range of values. Second, accelerators with different power profiles cannot efficiently be grouped in a single power domain, unlike in homogeneous chip multiprocessors (CMPs) where multiple (2 to 4) cores are often grouped into a single domain [43], [48]. In contrast, heterogeneous SoCs require one DVFS domain per accelerator [42], [47]. This finer spatial granularity further increases the complexity of power management. Third, unlike in homogeneous CMPs where one can define an aggregated throughput metric, such as MIPS, to measure the instantaneous efficiency of a given power allocation, there is no equivalent metric to aggregate the throughput across heterogeneous accelerators with widely varying workloads, making an optimal power-allocation scheme across accelerators infeasible. Instead, current state-of-the art designs propose heuristic-based greedy [42], fair [49], or equal [50] power allocations. Fourth, fixed-function accelerators might not be programmable (i.e. execute instructions) and therefore may not be able to run an OS or other software for power management. OS kernel approaches, such as those demonstrated in [43], cannot be distributed across these kinds of accelerators, and would need to rely on software running on the general-purpose cores. Hence, for SoCs that feature many fixed-function accelerators, a hardware implementation is required to perform fully-distributed power management.

B. Centralized vs. Decentralized Control

On-chip power management can rely on a centralized or decentralized control scheme. In either case, the SoC is partitioned into *tiles*. Each tile contains a single PE and logic to communicate with other tiles. In the centralized case, a single *On-chip Controller* (OCC) receives inputs, such as utilization statistics, offline characterization data in the case of custom accelerators, and other sensor data, from every tile and determines the voltage/frequency operating points and the available power budget based on these inputs. Prior power-management designs for heterogeneous SoCs, such as [42], use an OCC consisting of a centralized on-chip DVFS controller and a software daemon that implements the desired DVFS policy. The target voltage and frequency of each tile are then set in a round-robin manner. However, the increased response times of such a centralized design can result in delays that lead to periods of suboptimal operation, particularly as the degree of heterogeneity increases. With per-tile voltage regulation becoming prevalent, the actuation of the voltage and frequency can be decentralized. Sill, a centralized OCC is needed to sequentially poll each tile, compute its V/F state and communicate the state information to all tiles, which can incur substantial overheads. In contrast, a fully decentralized power-management system allows each tile to locally govern its own voltage/frequency state by adjusting its power budget based on information exchanged with neighboring tiles. The main advantage of this technique is that each tile can adapt its power very quickly to changes in applications or application phases, without the need for global re-computation of the entire system state.

C. Overview of Voltage/Frequency Actuators

Once a power allocation is decided, actuators are needed to perform the voltage scaling from a fixed input V_{in} to the logic supply V_{logic} and scale the clock frequency to match the logic's F_{max} at V_{logic}. Two main classes of voltage regulators are available: switched regulators, such as Buck-Boost or switched capacitor, and linear regulators, such as Low Drop-Out Regulators (LDOs). Switched capacitor regulators can provide very high power-conversion efficiency ($\eta_P = 80-90\%$ [51], [52]) even with a large difference between V_{in} and V_{logic}. However, they rely on bulky inductors or capacitors, which occupy large on-chip area and/or require external components. In contrast, LDOs provide a smaller, though less efficient regulator design by introducing a resistance in series with the logic, whose value is set to give a desired IR drop between V_{in} and V_{logic}. LDOs hence require a method to sense V_{logic} to provide regulation feedback. This requires further analog or digital circuitry [53], [54], which adds to the design complexity.

Frequency generation is conventionally performed with a Phase-Locked Loop (PLL), which consists of an adjustable oscillator that tunes the logic frequency (F_{logic}), by locking F_{logic} scaled by a division factor N to a reference F_{ref} and varying N. However, PLLs are complex to design and integrate, and in the case of a Globally Asynchronous Locally Synchronous (GALS) clocking approach, commonly used in SoC architectures, the phase-locking of the clock to an absolute reference is not required. Hence, many designs can use a much simpler free-running oscillator. Some rely on internal adjustment of the oscillator to a fixed reference [55], while others use the oscillator as a critical path replica to track changes in operating voltage [51], [56]. The former guarantees a known operating frequency, while the latter helps reduce guard-bands due to voltage uncertainties.

III. Decentralized Management

In this section, we describe the BlitzCoin algorithm for decentralized power management in SoCs. We then conduct a simulation-based evaluation for different schemes and configurations of BlitzCoin, which results in various optimizations that help improve the overall convergence time, scalability, and power efficiency.

A. Coin Exchange Algorithm

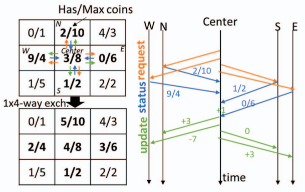
The BlitzCoin power-management strategy allows each tile to locally govern its own voltage/frequency state by adjusting its power budget in small units of power called '*coins*'. Each tile periodically exchanges coins with its neighbors based on their relative allocation compared to a target. This continues until the distribution of coins converges to an equilibrium state corresponding to the target allocation. Whenever a tile's activity changes, the target allocation changes, and coins are exchanged until the new equilibrium is reached. The total number of coins shared between the tiles at the SoC-level is fixed during the exchanges, enforcing a constant total power budget, which can be set to match the power source output, thermal design constraints, or the limit of the power delivery network.

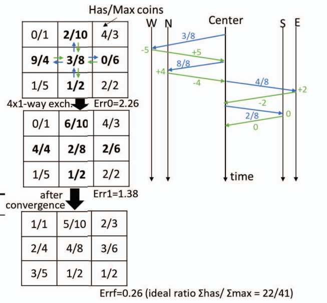
In SoCs with centralized power management, a global controller keeps track of each tile's current power setting (i.e. coin count). Upon changing its activity, a tile sends an update to the controller, which in turn sends a coin update to all the tiles. In an SoC consisting of tiles interconnected by a network-on-chip (NoC), the coin update time scales linearly with the number of tiles N, as the centralized controller must issue an update to each tile sequentially. In contrast, we propose a more scalable, decentralized approach by limiting coin exchanges to occur between neighboring tiles.

Each tile is assigned a target coin count, which we define as *max*, proportional to a maximum power value that is set by a pre-determined power-management strategy. This strategy could entail an equal *absolute* power allocation, involving setting the same *max* coin value for each tile. An alternative strategy would involve setting *max* proportional to each tile's power running at F_{max}, i.e an equal *relative* power allocation, corresponding to a workload-aware power-management strategy. The performance of these allocation strategies will be further discussed in Section VI-A. To exchange coins, each tile periodically requests the current coin count (*has*) and maximum power (*max*) from its 4 neighbors in the north (N), south (S), east (E) and west (W) directions. The *max* value of a tile is set when the execution begins and is set to 0 once it ends. Thus, the start/end of a tile's execution triggers a change in the coin distribution across the SoC due to the tile requesting/relinquishing coins. In turn, every tile updates its coin count to have the same $\frac{has}{max}$ ratio as its neighbors, while keeping the total number of coins constant. As the coin exchange happens in parallel across tiles, the fixed coin count of the SoC is distributed such that $\frac{has}{max}$ converges to the same value for every active tile. Each exchange of information between tiles is performed by messages called *packets* sent through the NoC. A packet can be a coin transfer or other information exchanged between tiles.

A variant of this approach consists of performing the coin exchange with one neighbor at a time. Fig. 2 presents the algorithm for coin exchanges between neighboring tiles in both cases. In the 4-way exchange (Algorithm 1), a tile sends a request command to its 4 neighbors. Each neighbor replies with its status in terms of its *has* and *max* coins, as shown in lines 6 and 7. Then, the center tile computes the fair coin allocation in this group of 5 tiles, such that each tile has the same $\frac{has}{max}$ ratio, within rounding error (lines 10). Then, the center tile updates its coin count accordingly and sends the update to its neighbors (line 11). This process is repeated at an interval of *refreshCount* cycles (line 16).

In the 1-way exchange (Algorithm 2), each tile exchanges coins with one neighbor at a time, as indicated on lines 10-11. Here, at each refresh interval (*refreshCount*), a pair of neighboring tiles exchange coins, with each tile rotating amongst its neighbors in a round-robin





Algorithm 1 : 4-way exchange

```
neighbors= [(x-1,y),(x+1,y),(x,y-1),(x,y+1)]
if $update.received$ then
    self.has ← self.has + update.value
else if $status.received$ then
    while $numStatusReceived < 4$ wait
    tot.has ← self.has + $\Sigma_{i=0}^{3} status.has[i]$
    tot.max ← self.max + $\Sigma_{i=0}^{3} status.max[i]$
    self.has ← self.max*totHas/totMax
    for $i$ in 0 to 3 do
        coinNew ← status.max[i]*totHas/totMax
        sendUpdate(status.addr[i],
                    coinNew-status[i].has)
    end for
else if $request.received$ then
    sendStatus(request.addr,self.has,self.max)
else if refreshCount == refreshPeriod then
    for sides in 0 to 3 do
        sendRequest(neighbors[sides].self.has.self.max)
    end for
    refreshCount ← 0
else
    refreshCount ← refreshCount + 1
end if
```

Algorithm 2 : 1-way exchange

```
neighbors= [(x-1,y),(x+1,y),(x,y-1),(x,y+1)]
if $update.received$ then
    self.has ← self.has + update.value
else if $status.received$ then
    coinNew ← self.max*(self.has+status.has)
                /(self.max+status.max)
    sendUpdate(status.addr,self.has-coinNew)
    self.has ← coinNew
else if refreshCount == refreshPeriod then
    sendStatus(neighbors[sides],self.has,self.max)
    sides ← (sides+1)% 4
else
    refreshCount ← refreshCount + 1
end if
```

Fig. 2. Top: Illustration of one step of the coin exchange between a tile and its neighbors for the 4-way (left) and 1-way (right) techniques, final converged coin values, and the error across steps (Err_0, Err_1, Err_f). Bottom: Corresponding pseudocodes.

fashion. Fig. 2 numerically illustrates one pass of such an exchange from a center tile with $\frac{has}{max}$ ratio of 3:8. In both cases, the set of exchanges moves coins from tiles with a larger ratio to tiles with a smaller ratio. The final state is also shown, which has a residual error of $Err_f = 0.26$ coins, owing to the discrete nature of the coins.

BlitzCoin can also be adapted to address thermal limitations. Global thermal caps can be enforced by the initial configuration of the coin pool, so as to not exceed the thermal budget of the SoC. Hotspot issues are local in nature and can be addressed by augmenting the algorithm to reject coins from an exchange if the total allocations to a tile and its neighbors exceed a certain threshold.

B. 4-way vs. 1-way Exchange Techniques

In order to validate the convergence and scalability of the coin exchange process for large SoCs compared to a centralized approach, we run behavioral simulations using an in-house simulator. We compare the *1-way* and *4-way* techniques on SoCs comprising a square-mesh NoC with up to 400 tiles. We assume the same interval between coin exchanges in the *4-way* and *1-way* cases. Fig. 3 illustrates the time, expressed as NoC cycles, and packets required to reach convergence for both methods as a function of the size of one dimension d of a square SoC, where $d = \sqrt{N}$ for an N-tile SoC. The error Err is defined as the average over the N tiles of the difference between the effective number of coins has_i and the number of coins expected from a fair distribution $max_i \times \sum_{k=0}^{N-1} has_k / \sum_{k=0}^{N-1} max_k$. Convergence is defined when Err becomes smaller than a defined threshold (e.g. $Err < 1$); arbitrarily small thresholds can not be reached due to quantization effects. Fig. 3's results are the average of over 100 runs with random initializations. The convergence times of both the *4-way* and *1-way* schemes follow comparable trends. These behavioral simulations show response time scaling with $\sqrt{N}$. This demonstrates the superior scalability of the decentralized approach for many-tile SoCs, since a centralized approach would scale with N.

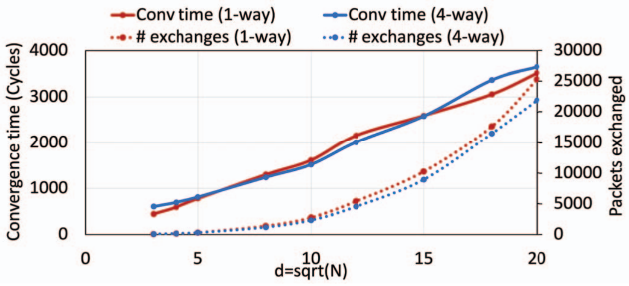


Fig. 3. Number of packets exchanged and time (NoC cycles) to reach convergence ($Err < 1.5$) for different SoC sizes (N tiles) for the *1-way* and *4-way* coin exchange methods as a function of SoC dimension $d = \sqrt{N}$.

Each *4-way* exchange contains more information than a single *1-way* one; hence, the convergence requires fewer exchanges. However, as shown in Fig. 2, a single *4-way* exchange is longer and more complex (requiring 12 messages vs 8). Moreover, the arithmetic needed for a *4-way* approach is more complicated than the *1-way* approach, requiring pipelining and synchronization primitives, which would increase the delay of each individual operation. This longer arithmetic would also increase the risk of conflict between tiles (e.g. a tile C sends a request to tile B, while tile A is still exchanging with B).

In summary, the 1-way method presents the following implementation advantages over the 4-way method:

• *1-way* exchange only requires 8 messages (status and update to each neighbor) instead of 12 in *4-way* exchange (request, status and update to each neighbor).

• *1-way* exchange requires simpler arithmetic for the coin update computation and incurs lower overheads in terms of hardware complexity and energy.

• Unlike the *4-way* exchange, which involves several many-to-one/one-to-many coin transfer operations, the *1-way* exchange only involves pairwise transfers of coins between tiles. This minimizes the likelihood of collisions, and eliminates the need for synchronization barriers between transfers.

Based on the comparable convergence performance of *1-way* and *4-way* exchanges, and the significantly reduced hardware complexity of a *1-way* implementation, *we adopt the* 1-way *coin exchange method as the preferred embodiment in BlitzCoin's implementation.*

Along with maintaining the global power cap, BlitzCoin can also be configured to address potential thermal emergencies due to localized hotspots. This is done by augmenting the coin exchange algorithm with a hard cap on the number of coins allocated to one tile or to sub-groups of tiles.

C. Comparison with Ring-Based Token Exchange Schemes

We compared our proposed exchange technique with a previous decentralized power-management design *TokenSmart (TS)* [43], that uses a ring-based tokenized approach. Unlike BlitzCoin, where tile exchanges occur with neighbors in parallel, TS sequentially passes the pool of available tokens between tiles. In the default *greedy* mode, each tile takes enough tokens from the pool to satisfy its target token count. When a tile is starved from tokens for a specified

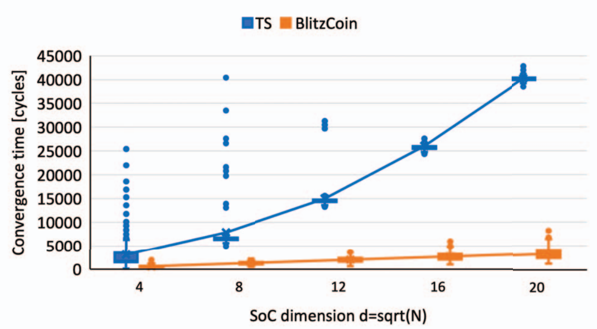


Fig. 4. Comparison of convergence time (NoC cycles) of BlitzCoin and TS as a function of the SoC dimension $d=\sqrt{N}$ across 1000 trials.

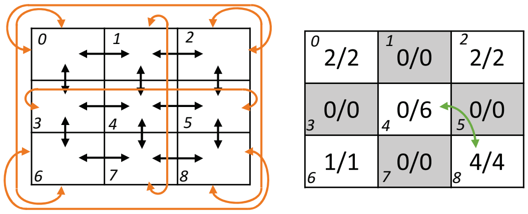


Fig. 5. Illustration of wrap-around (left) and random pairing (right).

duration, the global policy switches to a *fair* mode, which targets an equal token count in each tile. As seen in Fig. 4, BlitzCoin's convergence time scales with $d=\sqrt{N}$, while TS scales with N, resulting in $\sim$11× faster convergence for BlitzCoin versus TS for $N=400$, $d=20$. This is because even though TS is decentralized, the token exchange process is sequential. Moreover, due to the oscillations between greedy and fair modes, TS has some outliers with even longer convergence times, unlike BlitzCoin.

D. Convergence Time and Scalability

We augment the algorithm described in Fig. 2 with several optimizations that improve the performance and convergence time of BlitzCoin. These include a) *dynamic timing*, b) *wrap-around* and c) *randomized pairings*. We dynamically scale the update time between requests by using an exponential back-off algorithm; when a status update results in zero coin exchanges, the time to the next status update is scaled up by a factor λ, else it is decreased by a constant k. This provides faster convergence during sudden activity changes without causing unnecessary NoC traffic in the steady state. As shown in Fig. 5, we also expand the definition of neighboring tiles to *wrap around* to the opposite edges, so that edge/corner tiles (e.g. tile 0 in the figure) still have access to the same number of neighbors (1, 2, 3, and 6). Lastly, to avoid cases where a tile (e.g. 4) is surrounded by 4 inactive tiles (1, 3, 5, and 7), which might lead to a deadlock, we introduce a *randomized pairing feature*, wherein a tile will intermittently perform a random pairing with a tile other than one of its neighbors (e.g. 8). The random pairing frequency is configurable and our studies showed that a small value, such as once every 16 exchanges, is enough to prevent deadlock effectively. We have provided further analysis on convergence and elimination of deadlocks in Section III E.

Fig. 6 illustrates the benefits of *dynamic timing*. We observe that not only does it reduce the refresh interval but it can also reduce the total number of packet exchanges, resulting in an overall speedup.

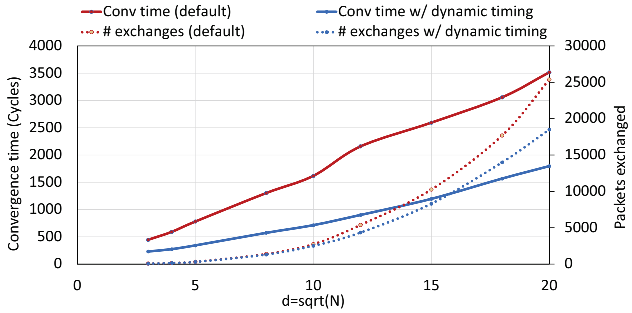


Fig. 6. Number of packets exchanged and time (NoC cycles) to reach convergence ($Err < 1.0$) a function of the SoC dimension $d = \sqrt{N}$ when comparing conventional *1-way* exchange with *1-way* with *dynamic timing*.

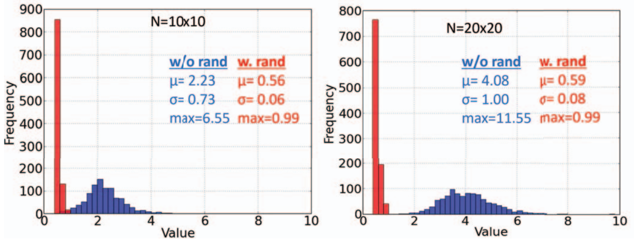


Fig. 7. Histograms of worst-case absolute error across all N tiles over 1000 runs for $N=100$ (left) and $N=400$ (right). The Blue histogram is obtained without *random pairing* and the red histogram with *random pairing* enabled.

This is because in large SoCs, the areas that have already converged will have fewer unnecessary messages and lower NoC traffic. Fig. 7 illustrates the distribution of residual error (maximum error across all tiles) after convergence. Without *random pairing* we see that some tiles do not converge to the target value and that the deviation grows with the SoC size. On the other hand, when *random pairing* is activated, all tiles converge to the target value – within the limits of 1-coin quantization – for both $N=100$ and $N=400$.

Our experiments also show that, while the optimizations described above reduce convergence time, error and design complexity, they do not significantly affect the convergence-time variability across runs with different coin initializations, which remains comparable to that shown in Fig. 4.

E. Analytical Insights on Convergence and Deadlock Prevention

Complementing our simulation study, we derive analytical insights on convergence mechanism of BlitzCoin. We first define the global convergence ratio $\alpha = \sum_{i=0}^{N-1} has_i / \sum_{i=0}^{N-1} max_i$, the error for a given tile $E_i = |has_i - \alpha \times max_i|$, and the global error $E = \frac{1}{N}\sum_{n=0}^{N-1} E_i$. We then define the initial coin ratios for tiles i and j as $\beta_i = has_i/max_i$ and $\beta_j = has_j/max_j$, respectively, and the final ratio as $\beta' = has'_i/max_i = has'_j/max_j$ (after the exchange the ratios are identical). Without loss of generality, we can set $\beta_i \geq \beta_j$, i.e. the coins flow from i to j. The value of the final ratio β' is between that of the two initial ratios $\beta_i \geq \beta' \geq \beta_j$. From this, as we compare β with the target ratio α we can consider four cases:

- $\beta_i \geq \beta' \geq \beta_j \geq \alpha$: Both before and after the coin exchanged, tiles i and j have too many coins compared to the target allocation. Noting c as the number of coins exchange, we get $E'_i = E_i - c$ and $E'_j = E_j + c$, and the total error E is constant.

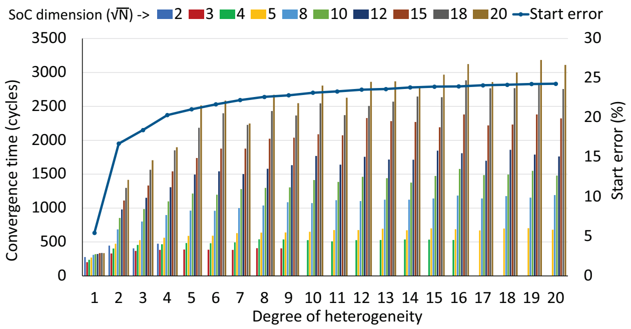


Fig. 8. Variation in convergence time for SoCs with different sizes (denoted by the dimension $d = \sqrt{N}$ for an N-tile SoC) and different degrees of heterogeneity (denoted by the number of distinct accelerator types). The figure also shows the error due to the initial assignment of *has* coins for the different configurations.

- $\beta_i \geq \beta' \geq \alpha \geq \beta_j$: Initially tile i has too many coins while tile j has too few, and after the exchange they both have too many. E_i is reduced by c, while E_j changes by at most +/- c, resulting in a reduction in the total error $E_i + E_j$.
- $\beta_i \geq \alpha \geq \beta' \geq \beta_j$: Initially tile j has too few coins, while i has too many, and after the exchange they both have too few. E_j is reduced by c, while E_i changes by at most +/- c, reducing the total error.
- $\alpha \geq \beta_i \geq \beta' \geq \beta_j$: Both before and after the coin exchange, tiles i and j have too few coins compared to the target allocation. In that case, $E'_i = E_i + c$ and $E'_j = E_j - c$, and the total error E is also constant.

With each exchange, E remains constant or decreases, and the token distribution can either converge to the desired global minimum $E_{final} = 0$ or to a local minimum $E_{final} > 0$, which could result in a deadlock. In the latter case, at least one pair of non-neighboring tiles (a, b) would exist where a has a strictly positive error and b a strictly negative one, i.e. $\beta_a > \alpha > \beta_b$. By forcing exchanges between pairs of non-neighboring tiles, *random pairing* overcomes potential deadlocks by eventually arriving at the desired pair of tiles, namely a and b. Since our hardware implements this functionality as a shift-register that eventually pairs all non-neighboring tiles, this a-b pairing is guaranteed to happen within a fixed time.

F. Convergence Time and Heterogeneity

Based on our experiments, we observe that the degree of heterogeneity in the SoC can impact the overall convergence time. Fig. 8 shows the convergence times for SoCs while varying the SoC size (denoted by dimension $d = \sqrt{N}$ for an N-tile SoC) and the degree of heterogeneity ($accType$) as two independent parameters. An $accType$ value of 1 corresponds to a completely homogeneous system, while larger values of $accType$ represent increasingly heterogeneous systems with more types of accelerators.

In our evaluations, we assume that all accelerators belonging to the same type have the same max coins, and we run the coin exchange algorithm from a random initial coin allocation until the optimized distribution of has coins has been achieved across the SoC. Fig. 8 also shows the error at the beginning of the simulation (*start_error*) due to the initial *has* coin distribution. With higher degree of heterogeneity, the *start_error* is larger, which also results in a longer convergence time.

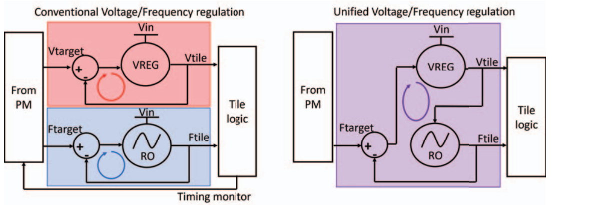


Fig. 9. Comparison of conventional and UVFR actuators.

IV. Hardware Implementation

This section describes various aspects of the hardware implementation of BlitzCoin, including details of the DVFS controller, the unified voltage and frequency regulation scheme, and integration into a heterogeneous SoC platform. The design of BlitzCoin is focused on 2D-mesh NoC architectures, as their inherent scalability matches well with the goals of BlitzCoin. A RTL design of BlitzCoin was created, integrated into a many-accelerator SoC, and evaluated in a 12 nm process technology.

A. Per-Tile DVFS Implementation

BlitzCoin requires separate voltage and frequency actuation for each accelerator in the SoC. For large SoCs, this necessitates the design of a compact regulator system. Furthermore, replicating the regulator system in each heterogeneous tile can be costly from a design time perspective. For these reasons, we target a simple, area-efficient, and digital-flow-compatible regulator. Simpler regulators may have longer actuation times, but in BlitzCoin the actuations across different tiles occur in parallel. In large SoCs, this constant overhead will be negligible in comparison to the response time of the coin exchange algorithm that scales as $O(\sqrt{N})$.

These requirements first guide the choice of a linear LDO rather than a switched regulator, which occupies more area and requires custom analog design. Then, rather than conventional dual voltage/frequency control loops, we implement a Unified Voltage and Frequency Regulator (UVFR), as shown in Fig. 9. The UVFR guarantees that the tile operates at an optimal voltage for each given frequency target, without the need for large guardbands for transient IR drop or complex circuit instrumentation like canary flip-flops [57]. Further, the use of a frequency input rather than voltage input makes the feedback comparator simple to implement as a counter-based Time-to-Digital Converter (TDC), rather than a complex, fully-analog voltage comparator.

In the UVFR scheme, the clock frequency naturally tracks voltage changes. As documented in [58]–[60], when a voltage droop occurs, the oscillator propagation time increases and delays the next clock edge sent to the accelerator. This mitigates the effects of voltage changes either caused by sudden changes of activity in the accelerator or in other accelerators sharing the same LDO input rail.

Based on these choices, we designed a BlitzCoin-enabled tile shown in Fig. 10. The SoC tiles are composed using two clock domain levels: the NoC domain runs at a single frequency F_{NoC} and voltage V_{NoC}, while the tile domain runs on its own locally generated frequency and voltage (F_i,V_i). F_i is generated by a local free-running Ring Oscillator (RO) supplied by V_i and tuned to act as a Critical Path Replica (CPR), i.e. for any value of V_i it will generate a frequency F_i close to the tile's maximum frequency at V_i. In turn, V_i is generated by an LDO regulator from a fixed voltage.

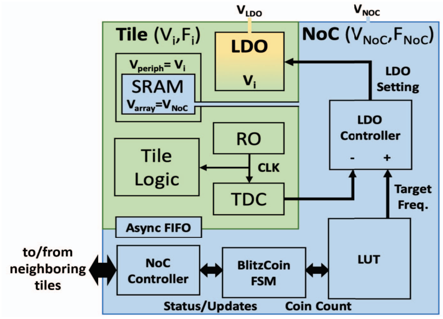


Fig. 10. Architecture of a tile with BlitzCoin integrated into its NoC domain.

Previous work that relies on switched capacitors with [51], [56] or without UVFR [61], require large area overheads (36%, 16% and 17%, respectively) and custom analog macros. Some fully-synthesizable, digital LDOs have been published with small area overheads (1.4% for a 1 mm^2 tile in [54]) or with UVFR support at the cost of larger area (4.5% for a 1 mm^2 tile in [62]). In contrast, our design includes a fully-synthesizable UVFR with under 1% area overhead (0.49% for the TDC and coin exchange logic, 0.04% for the RO and 0.01-0.03% for the LDO) in a 1 mm^2 tile.

According to prior state-of-the-art tile-based ASICs [47], [63], [64], as well as our own estimations, the NoC accounts for roughly 5-10% of the power of such SoCs. Scaling the voltage/frequency of each NoC domain in accordance with the corresponding tile would result in multiple asynchronous boundaries, thus increasing latency variability of NoC communication. Although our paradigm supports globally scaling the NoC voltage, this would affect the full SoC's performance. Hence, we restrict the NoC to operate at a single voltage level, independent of the tile-domain logic. The tile SRAMs use a split supply with the array operating at the fixed V_{NoC} and the periphery at the logic supply V_i. This prevents low-voltage failures in SRAM cells while adding negligible power overhead.

The full power-management logic, shown in Fig. 10, is placed in the NoC power domain because this domain is common across tiles, and such a placement avoids possible deadlock when scaling the tile voltage and frequency. The power management is implemented in the following steps: **(1)** The *NoC controller* receives coin-exchange-related packets from neighboring tiles and transfers them to the *BlitzCoin Finite State Machine (FSM)*, which implements the algorithm described in Section III and performs the coin update. **(2)** A lookup table (LUT) converts the coin count into a target frequency F_{target} for the tile, based on a pre-characterization of the power profile of each tile. **(3)** A Time to Digital Converter (TDC), contained in the tile domain, generates a digital representation of the current clock frequency F_i that is output by the RO. **(4)** The LDO controller compares F_{target} with F_i to adjust the LDO setting using a PID controller.

The implementation of the BlitzCoin FSM closely matches the algorithm in Section III, with a few extra practical considerations. First, we set the coin counter's precision to 6 bits, which yields 64 power levels per tile. This granularity is much finer than previous solutions, which implement between 2 and 5 power levels [42], [46], [50], [51], [65]. At the same time, it still allows the BlitzCoin FSM to finish its computation in one clock cycle. Note that, since coin exchange messages may have to compete with other message types on the NoC, a coin request can be delayed and arrive at a time where the tile has already given its coins to another neighbor, temporarily causing a negative coin count. We address this by expanding the coin register with a sign bit to address underflow. Note that negative coins only appear as transient convergence artifacts, while the steady-state coin counts are always positive.

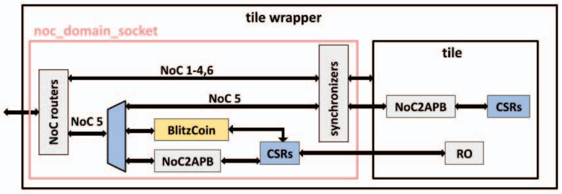


Fig. 11. The new NoC domain socket, containing the BlitzCoin unit, integrated inside the ESP tile wrapper.

B. SoC Integration through the NoC

To evaluate BlitzCoin in the context of an entire SoC, we leverage ESP, an open-source research platform for heterogeneous SoC design [66], [67]. The integration of BlitzCoin in ESP constitutes one possible implementation of the proposed power-management strategy. The scalability of ESP allows us to experiment with large SoCs, while its modular, agile design flow allows for rapid prototyping of different SoCs.

The ESP architecture is organized as a grid of tiles interconnected by a 2D-mesh, multi-plane NoC. The architecture consists of four types of tiles: 1) each processor tile hosts a CPU, in this case, a RISC-V CVA6 core [68], [69] to boot Linux and execute software applications; 2) each accelerator tile contains a loosely-coupled accelerator [70], which can be designed using one of ESP's several design flows or independently if it complies with a standard bus protocol, such as AXI [71]; 3) each memory tile contains a slice of the last-level cache [72], [73] and a channel to off-chip DRAM that services the corresponding partition of the global address space; 4) finally, the auxiliary tile supports all other I/O interfaces (i.e. Ethernet, UART) and miscellaneous components, such as the boot ROM and interrupt controller. Each accelerator tile operates in its own voltage/frequency domain, while the NoC operates on a fixed global voltage/frequency domain. Hence, each NoC message between any pair of tiles requires only two boundary crossings, when entering and exiting the NoC [47], [74].

Each tile in ESP contains a socket that provides various services (e.g. coherence, DMA, interrupts) to the tile; this socket operates in the clock and power domains of the tile. However, as previously stated, the power-management logic must be placed in the NoC domain to keep the service available even as the tile voltage and frequency change, particularly when a tile becomes fully clock gated.

Since the NoC available in the ESP release only contains logic related to its core functionality, we developed a new, *NoC domain socket* in each tile that sits in the NoC clock and power domains. As shown in Fig. 11, the NoC domain socket only interacts with NoC Plane 5, which handles accesses to memory-mapped registers throughout the SoC and interrupts; the remaining five planes – three for coherence and two for accelerator DMA – merely pass through voltage/frequency boundary-crossing synchronizers to the tile. We also added a new message type to NoC Plane 5 for coin-based power management. The NoC domain socket sits before the synchronizers into the tile domain and instantiates the BlitzCoin FSM, the LDO controller, and the LUT. It also further instantiates a set of Control and Status Registers (CSRs), including configuration registers for the BlitzCoin unit and the ring oscillator. Finally, we added a round-robin arbiter to control access to NoC Plane 5, since messages can come from the BlitzCoin unit, the NoC domain CSRs, or the register interface in the tile itself at any time.

C. *Extending BlitzCoin to Non-Accelerator Tiles*

As BlitzCoin is decoupled from the design of SoC components, it can theoretically be integrated with any type of component. However, there are several practical constraints that led us to limit the scope of BlitzCoin to only accelerators in our design. As the IO and memory tiles in ESP communicate with external components, it is impractical to support DVFS on them. Using BlitzCoin with CPU tiles would require the power-to-frequency LUT to be dynamically adjusted to support the wide variation in workloads run on CPUs. Although previous work [18], [75] have demonstrated the use of activity counters and other power proxies for this purpose, the increase in complexity and the fact that CPUs represent only a small part of the total power budget in accelerator-rich SoCs led us to exclude BlitzCoin from CPUs in our implementation. Similarly, scaling the NoC voltage and frequency presents challenges related to tile-to-tile communication and closing timing across multiple V/F points. Considering that the NoC represents only a small fraction of the total power when the accelerators are active (e.g. 7% reported in [49]), we implemented the NoC with fixed voltage and frequency, guaranteeing a one-cycle-per hop throughput. To maintain the global power budget with BlitzCoin only in the accelerator tiles, we allocate a fixed number of coins to all non-accelerator tiles and the NoC.

V. SoC-level Evaluation Methodology

In this section, we present the methodology used to evaluate the hardware implementation of BlitzCoin (Section IV) in the context of full SoCs. First, we describe RTL simulations of two distinct ASIC SoC designs, the applications that run on these SoCs, and the state-of-the-art centralized power-management schemes that we use as baselines for BlitzCoin. Next, we describe the 12 nm heterogeneous SoC prototype we designed, which features BlitzCoin implemented inside a *10-accelerator PM cluster*. Finally, we derive the scaling trends of BlitzCoin and the baselines, which we use to scale the results to larger SoCs comprising up to several hundreds of tiles.

A. *Evaluated SoC configurations*

We implemented BlitzCoin on 2 different SoCs, whose block diagrams are shown in Fig. 12. These consist of a 3x3-tile SoC targeting an application for connected autonomous vehicles developed in industry [76] and a 4x4-tile SoC targeting multiple computer vision applications, including night vision, image denoising, and digit classification [77]. The 3x3-tile SoC includes 3 tiles with Fast Fourier Transform accelerators (*FFT*) used for depth estimation, 2 tiles with Viterbi decoding accelerators (*Viterbi*) for vehicle-to-vehicle communication, and 1 tile with the NVIDIA Deep Learning Accelerator (*NVDLA*) for object detection and other deep learning tasks. The 4x4-tile SoC includes accelerators for 3 different kernels: computer vision (*Vision*), dense matrix multiplication (*GEMM*) and 2D convolution (*Conv2D*). The Vision accelerator includes internal engines for noise filtering, histogram equalization, and discrete wavelet transform, while the GEMM and Conv2D accelerators support CNN-based inference.

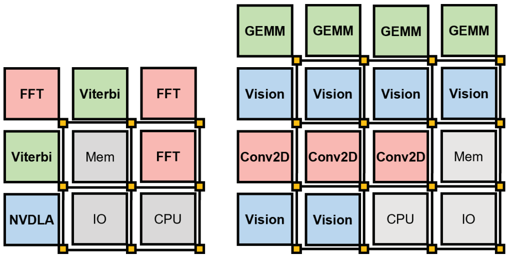


Fig. 12. 3x3-tile and 4x4-tile SoCs implemented and evaluated.

Fig. 13 shows the power-frequency curves for the accelerators in the 3x3-tile (left) and 4x4-tile (right) SoCs. The FFT, Viterbi, and NVDLA data is obtained from ASIC measurements, while the GEMM, CONV2D, and Vision data is obtained from post-synthesis simulations with Cadence Joules [78] across multiple voltage library corners. At minimum voltage, frequency can be further reduced, which produces large additional power savings when a tile is idle (7.5× from our measurements). Hence, per-tile power gating would provide negligible additional SoC-level power reductions, while adding significant implementation overheads.

To measure the DVFS performance of BlitzCoin, we carry out cycle-accurate RTL simulations of the complete SoC. Given the SoC configuration, ESP generates synthesizable RTL for the whole SoC. Each simulation runs a bare-metal C program on the CPU core of the SoC, which invokes the different accelerators based on the workloads further described in Section V-B. The SoC's memories are generated using the Arm 12 nm memory compiler [79]. The Ring Oscillator (RO), which provides the local clock to each tile, is simulated as a time-annotated block that dynamically adjusts its frequency based on the local LDO setting. The CPU and NoC run at 800MHz - the maximum NoC frequency in our fabricated SoC described in Section V-D. At the end of the simulation, we extract each tile's instant frequency at each time step, based on its LDO setting, and use it to reconstruct its power trace based on the data from Fig. 13.

Note that we only scale the voltage and frequency of the accelerator tiles, while the other tiles (i.e. CPU, Memory, I/O) and NoC are maintained at a constant frequency, as discussed in Section IV-C .

B. *Evaluated Workload Scenarios*

As noted before, BlitzCoin guarantees a system-wide power allocation such that all tiles' power are in the same proportion to their target power, denoted as *max*. However, these *max* values can be programmed to represent different power-allocation strategies. We compare SoC performance under two given strategies: *Absolute Proportional (AP)*, where all tiles are assigned the same power target, and *Relative Proportional (RP)*, where each tile receives a power target proportional to its power at F_{max}.

We examine two different dataflow scenarios for the workloads we evaluate on the 3x3 and 4x4-tile SoCs, as illustrated in Fig. 14. In

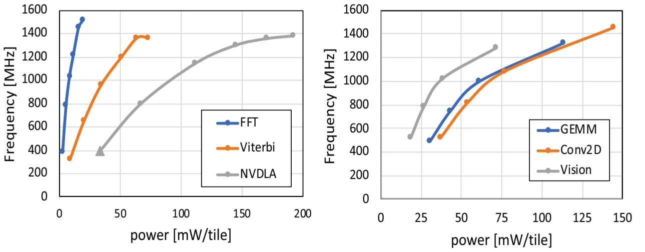


Fig. 13. Individual accelerator power/frequency characterization. Left: ASIC measurement of FFT, Viterbi (0.5V-1V) and NVDLA tiles (0.6V-1V) power across DVFS operating points. NVDLA curve is extended by frequency scaling at 0.6V (triangle marker). Right: Power characterization of GEMM, Conv2D and Vision accelerators (0.6V-0.9V) using Cadence Joules.

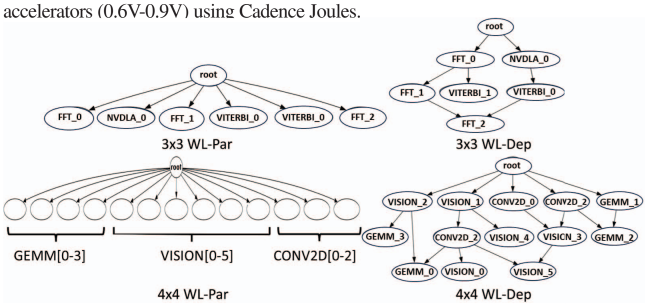
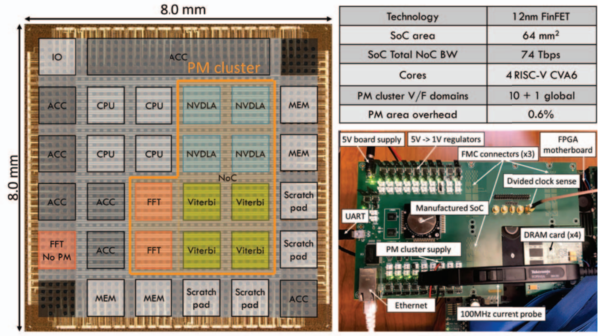


Fig. 14. Types of workloads evaluated on the 3x3 and 4x4-tile SoCs.

the Workload-Parallel (*WL-Par*) scenario (left), all accelerators run concurrently with no data dependencies between their corresponding tasks. In the Workload-Dependent (*WL-Dep*) scenario (right), tasks may be dependent on one or more tasks running on other accelerators, as would be the case for a complex, realistic application running on the SoC. We represent these dependencies across accelerators in the form of directed acyclic graphs (DAGs).

C. Evaluated Baseline

In order to compare our proposed implementation against a relevant baseline, we implemented a simplified version of the centralized controller proposed in [42]. Here, the centralized controller, which we denote as *Centralized-Round-Robin (C-RR)*, monitors the status of the different tiles and uses a round-robin scheme to decide which tiles are allowed to run at maximum (V, F) based on a global power cap. In this scheme, tiles are allocated to run alternately at maximum or minimum (V, F), and this allocation is rotated periodically to guarantee fairness. In addition to the C-RR baseline, we also compare BlitzCoin against a design that directly implements a power-allocation scheme similar to BlitzCoin, but with a centralized DVFS controller, which we refer to as *BlitzCoin-Centralized (BC-C)*. This allows us to separately evaluate the benefits of our power-allocation strategy itself and the benefits of its decentralized hardware implementation. In *BC-C*, the frequency of each tile is set in proportion to the ratio of the tile's target power to the whole SoC's power. In both *C-RR* and *BC-C*, each tile is equipped with its own oscillator that enables decentralized frequency actuation, but the control and determination of frequency states occur at the centralized controller. Since the computation and communication are centralized, the overall latency and response time will be larger than those of BlitzCoin's fully decentralized approach.



Technology	12nm FinFET
SoC area	64 mm^2
SoC Total NoC BW	74 Tbps
Cores	4 RISC-V CVA6
PM cluster V/F domains	10 + 1 global
PM area overhead	0.6%

Fig. 15. Annotated die micrograph, chip specifications and measurement setup.

D. Fabricated SoC Prototype

To validate the real silicon operation of the proposed strategy, we integrated BlitzCoin into the design of a 64 mm^2 heterogeneous SoC in a 12 nm technology [80]. Shown in Fig. 15, the SoC is a 6x6 tile grid that comprises 31 distinct tiles, 10 of which form a *PM cluster* in which BlitzCoin is implemented. The tiles in the PM cluster were selected to demonstrate the use of BlitzCoin in the context of a complete autonomous vehicles application, which relies on FFT, Viterbi and NVDLA accelerators [76]. The SoC also features 4 RISC-V CVA6 cores; one IO tile; 4 memory (MEM) tiles, which host an LLC partition and a channel to external memory; 4 1-MB scratchpad tiles; and 8 other accelerator tiles (ACC), including an additional FFT accelerator tile implemented without BlitzCoin (*FFT No-PM*) to serve as a baseline for comparison.

The chip was manufactured, packaged and experimentally measured using the laboratory setup shown in Fig. 15 (bottom right), which uses an FPGA test harness. A workload using 7 accelerators (NVDLA, 2 FFT, 4 Viterbi) in the PM cluster runs on one CVA6 core. We measured the total input current, tile frequencies, and internal power-management states during the workload's execution.

E. Extension to Larger SoCs

We leverage the data gathered from the various SoCs to extrapolate the performance of BlitzCoin (BC), BC-C, and the baseline C-RR for SoCs with a much larger number of accelerators ($N >> 16$). First, as explained in Section I, for a given accelerator-level workload phase duration T_w, the average SoC-level workload change duration is T_w/N. Hence, we must ensure that the power-management response time $T(N)$ meets $T(N) < T_w/N$. We define N_{max} as the maximum number of accelerators that can be supported by a power-management scheme for a given T_w as $T(N_{max}) = T_w/N_{max}$. Then, as described in Section III, the response times of the centralized approaches scale linearly with N while T_{BC} scales with $\sqrt{N}$. From this, we obtain the value of N_{max} as a function of T_w, where $\mathcal{T}_{C-RR}$, $\mathcal{T}_{BC-C}$, and $\mathcal{T}_{BC}$ are scaling constants that are obtained by fitting the model with the data gathered from the simulated and measured SoCs. The corresponding definitions of T and N_{max} for each of the evaluated schemes are shown in Equations 5.1-5.3.

$$T_{C-RR}(N) = N \cdot \mathcal{T}_{C-RR} \qquad N_{max,C-RR} = (T_w/\mathcal{T}_{C-RR})^{1/2} \tag{5.1}$$

$$T_{BC-C}(N) = N \cdot \mathcal{T}_{BC-C} \qquad N_{max,BC-C} = (T_w/\mathcal{T}_{BC-C})^{1/2} \tag{5.2}$$

$$T_{BC}(N) = \sqrt{N} \cdot \mathcal{T}_{BC} \qquad N_{max,BC} = (T_w/\mathcal{T}_{BC})^{2/3} \tag{5.3}$$

VI. SoC-level Evaluation Results

In this section, we present results of our SoC-level RTL simulations for various power-allocation strategies, power budgets and workload dataflows. We then show measured results from our fabricated 12 nm chip to validate the silicon embodiment of BlitzCoin.

A. Evaluation of the 3x3-Tile SoC

We compare workload execution times with BlitzCoin under the *AP* and *RP* allocations described in Section V for the 3x3 SoC shown in Fig. 12. Evaluations show that *RP* offers a 3.0% to 4.1% throughput increase compared to *AP* for power budgets from 60 to 120 mW. A fixed power budget per tile forces low-power tiles to operate at a less energy-efficient, high-voltage point to meet the power allocation. In contrast, *RP* guarantees that tiles run at more efficient (V, F) points. *Hence, the rest of the evaluation will report performance for the* RP *allocation.*

Fig. 16 presents the measured power traces of the autonomous driving workload [76] running on the 3x3-tile SoC in *WL-Par* and *WL-Dep* modes (described in Section V), assigned 120 mW and 60 mW, respectively. Power transitions near activity changes are shown magnified to the left of each power trace. In *WL-Dep*, the task dependence among accelerators results in only a subset of tiles running concurrently, thus making a lower (60 mW) power budget feasible. Note that subsequent evaluations also explore the sensitivity of both *WL-Par* and *WL-Dep* to different power budgets.

In Fig. 16, we see that all three methods enforce the power cap. Compared to *BC-C* and *C-RR*, BlitzCoin has the fastest response time, as seen in the zoomed-in trace showing power reallocation after NVDLA completes. This results in an overall better utilization of the available power and shorter runtime. The use of a centralized controller for power management in BC-C and C-RR slows down scheduling, further impacting run time.

Fig. 17 reports the total application runtime and power-management response time for the *WL-Par* and *WL-Dep* workloads with two power budgets of 30% and 15% of the maximum combined power of all accelerators, i.e. 120 mW and 60 mW. We observed similar trends when running other dependent workloads. From the figure, we observe that even with a centralized controller implementation, *BC-C* provides, on average, a 24% speedup versus the *C-RR* baseline due to the more efficient proposed power allocation. In addition, the proposed decentralized hardware implementation, *BC*, provides 10.1× and 12.1× average response time improvement compared to *BC-C* and *C-RR*, respectively. *BC* also provides additional throughput improvements (9% vs. *BC-C* and 34% vs. *C-RR*), which further enable better scalability to larger SoCs. As discussed previously in Section V-E, the response time of the power-management scheme becomes more critical as the size of the SoC increases.

C-RR throughput is limited by discrete power levels, while BlitzCoin and *BC-C* leverage fine-grained DVFS to utilize the full power budget. BlitzCoin's improvement versus *BC-C* is larger for higher power budgets because the workload duration T_W is reduced. Also, BlitzCoin has a higher improvement for *WL-Par* than for *WL-Dep*, because the larger number of concurrent executions maximizes the benefits of decentralized power management.

B. Evaluation of the 4x4-Tile SoC

Using the same method as with the 3x3-tile SoC, we captured power traces on the 4x4-tile SoC running either a parallel or dependent computer-vision workload. The parallel workload is simulated with power budgets of 450 mW and 900 mW, representing 33% and 66% of the maximum combined power of the accelerators, respectively. The dependent workload is simulated at 450 mW, as the combined power of concurrently executing accelerators is always below 900 mW. The results in Fig. 18 confirm the trends seen on the 3x3-tile SoC. Compared to the C-RR baseline, BC-C provides 20% throughput improvement on average, while our fully decentralized hardware implementation (BC) improves C-RR's response time by 8.3× and throughput by 25%.

C. Silicon SoC Measurements

Measurements on the 12nm SoC prototype presented in Section V-D provide real silicon data and can be used to validate the prior simulation-based results. First, we observe that when the FFT accelerator in the PM-cluster is run with BlitzCoin disabled, it operates with nearly identical performance as the baseline *FFT No-PM* tile (<2% frequency and energy efficiency difference). This shows the negligible overhead of BlitzCoin. We then validate the coin-exchange behavior at workload startup in Fig. 19 (bottom left). After a random initialization, the coins are redistributed across the seven active tiles according to their targets, with a residual error (due to quantization) of less than one coin, which matches the simulated results in Fig. 7. We further demonstrate the UVFR operation within BlitzCoin in Fig. 19 (bottom right), where we show a captured clock transition in one of the accelerator tiles when the LDO setting is updated and the corresponding measured change in the TDC readout.

By measuring the total current drawn during workload execution, including the redistribution of coins and per-tile V/F actuation, we show that the measured input power stays within the allocated budget with high power utilization $P_{avg}/P_{budget} = 97\%$, as shown in Fig. 19 (top). This verifies that BlitzCoin's operation enforces the desired power budget when real workloads are run on silicon. The plot further shows the execution-time improvement compared to a baseline where power is allocated statically. BlitzCoin achieves a 27% throughput improvement compared to this baseline. Similar experiments for 5, 4 and 3-accelerator workloads show similar 26%, 26% and 19% throughput improvements, respectively.

Finally, Fig. 20 shows the measured response time of the coin exchange by recording the number of coins held by each tile during an activity transition, namely following the end of the NVDLA task in the 7-accelerator workload. BlitzCoin's response time is 0.68 μs, while the measured response time of the BC-C and C-RR baselines for the same activity transition are 1.4 μs and 15.3 μs respectively, i.e. 2.1× and 22.5× larger than that of BlitzCoin.

D. Evaluation of Larger SoCs

We use the measured response times from Figures 17 (N=6), 18 (N=13) and 20 (N=7) to estimate the fitting values in Equations 5.1-5.3. We obtain $\mathcal{T}_{BC} = 0.20\mu s$, $\mathcal{T}_{BC-C} = 0.66\mu s$, and $\mathcal{T}_{C-RR} = 0.96\mu s$. We similarly obtained $\mathcal{T}_{TS} = 0.22\mu s$ for our hardware implementation of TokenSmart (TS), evaluated in Section III. Fig. 21 (left) presents scaled results for SoCs with a large number of

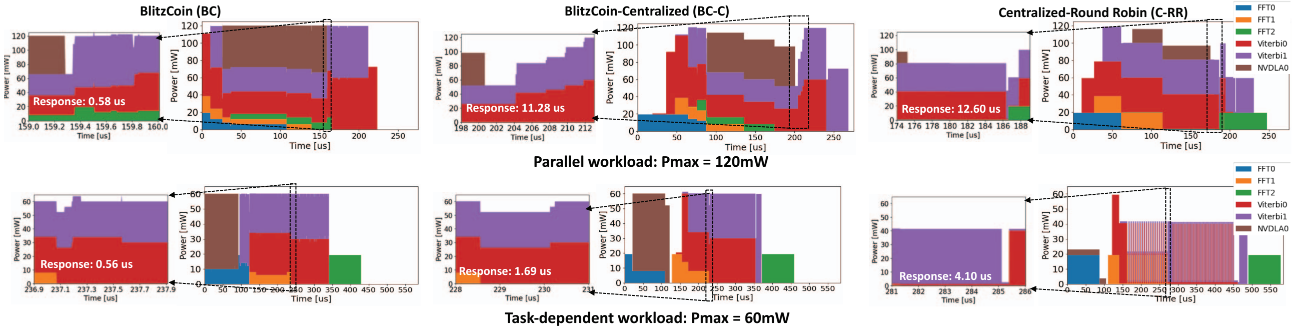


Fig. 16. Power trace for a connected autonomous vehicle workload with the individual tiles running on the 3x3 SoC in parallel (*WL-Par*) on the top and in a task-dependent manner *WL-Dep* on the bottom. In *WL-Par*, the NVDLA, FFT and Viterbi accelerators run concurrently with a total power budget $P_{max} = 120mW$ while in *WL-Dep* case they execute in a task dependent manner as shown in Fig. 14 with a total power budget of $P_{max} = 60mW$.

TABLE I
BLITZCOIN COMPARED TO OTHER IMPLEMENTED STATE-OF-THE ART DESIGNS

Strategy	Reference	Control	Power cap	DVFS scope	Allocation	DVFS domains (N)	DVFS levels	Response time	Scaling
BlitzCoin	BC	Decentralized	Yes	Heterogeneous	Equal or proportional	4-400	64	0.39-0.77us@N=13	$O(\sqrt{N})$
	BC-C	Centralized				6-13	64	3.8-8.0us@N=13	O(N)
Round robin	C-RR	Centralized	Yes	Heterogenous	Greedy	6-13	64	3.7-6.4us@N=13	O(N)
	[42]					10-12	4	1ms@N=12 **	
Fair-greedy	TS	Decentralized	Yes	Heterogenous	Greedy/Equal	4-400	64	2.9us@N=13	O(N)
	[43]			CPU		12	4	4ms@N=12	
Price theory	[81]	Hierarchical	Yes	CPU (big/small)	Bidding	4-256	8	6.62-11.4ms@N=256	sub-linear
Voting	[49]	Decentralized	No	NoC	N/A	16	3	8.19us@N=16	O(1)
Fast DVFS	[65]	N/A	No	CPU	N/A	2*	3	<2us	N/A
	[51]		Yes	CPU		1	3	<1us	
Token	[50]	Centralized	Yes	CPU	Equal	2-16	2 or 5	12.4ns@N=16 **	O(N)

One core acts as PM controller for the other **Response time normalized to 800MHz frequency*

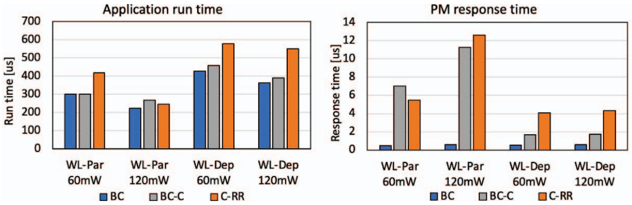


Fig. 17. Comparisons of execution time (left) and response time (right) of BlitzCoin compared to BC-C and C-RR on a 3x3 SoC for different power budgets and workload types.

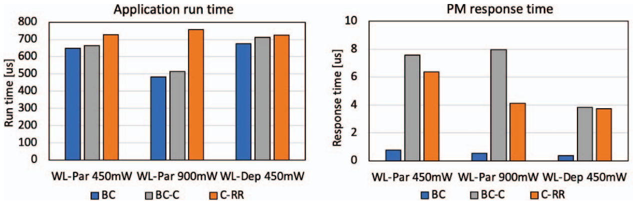


Fig. 18. Comparisons of execution time (left) and response time (right) of BlitzCoin (BC), BC-C and C-RR on a 4x4 SoC for different power budgets and workload types.

accelerators. We see that BlitzCoin can support $5.7-13.3\times$ more accelerators than BC-C and C-RR and $3.2-6.2\times$ more accelerators than TS. In other words, BlitzCoin can support N∼1000 accelerators for $T_w \geq 7.0ms$ and N∼100 for $T_w \geq 0.2ms$. This superior scalability can also be interpreted in terms of the time overhead of power-management decisions. Compared to the original software results from TokenSmart [43], BlitzCoin's overhead is $129-228\times$ smaller.

We also include a comparison with [81], which uses a hierarchical

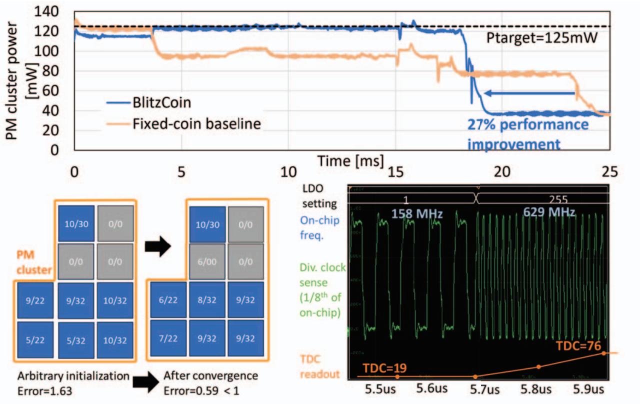


Fig. 19. Top: power and internal states traces during a 7-accelerator workload running with BlitzCoin. Bottom left: Coin allocation before and after convergence. Bottom right: Frequency transition of the tile clock and corresponding TDC readout updates during a LDO change. All results are silicon measured.

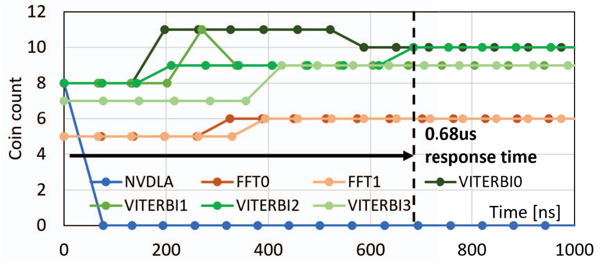


Fig. 20. Silicon measured coin exchanges after an activity change until equilibrium is reached. Captured at the end of the NVDLA task of the 7-accelerator workload.

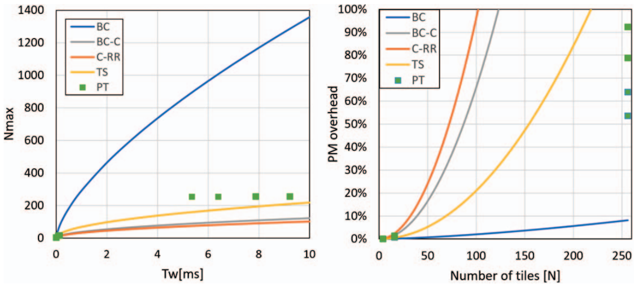


Fig. 21. Left: Maximum supported number of accelerators N_{max} as a function of workload duration T_w. Right: Time spent in PM as a function of tile count N for $T_w = 10ms$. Comparison is carried out between BC, and BC-C, C-RR, TS and PT.

power-management technique based on Price Theory (PT) with per-cluster DVFS for heterogeneous multi-cores. Implemented in software, its absolute response time is much slower than BlitzCoin; for example, it reports a 6.6-11.4 ms response time for various configurations with $N = 256$ clusters, compared to 0.003 ms for BlitzCoin. For fairer comparison, we scaled down the response time from [81] to account for HW speedup and clock speed normalization. In [43], the authors used a range of 2-3 orders of magnitude for their update period when scaling from software to hardware. Thus, we used a scaling factor of 2.5 orders of magnitude to account for this. The corresponding results are indicated by the green PT markers in Fig. 21 with various core and thread counts for a given N. BlitzCoin provides $3.2 - 5.0\times$ larger N_{max} compared to the hardware-scaled PT and $147 - 233\times$ larger N_{max} compared to the original data from [81].

Fig. 21 (right) shows the time spent in power management normalized to total execution time for different values of N across the different strategies for $T_w = 10\ ms$. For example, for $N = 100$, 10000 power-management decisions are needed per second of execution, for which C-RR, BC-C, and TS would require 960 ms, 660 ms, and 210 ms, i.e. 96%, 66%, and 21% of run time respectively. In contrast, BlitzCoin only needs 20 ms (i.e. 2.0%). Y-Values above 100% correspond to points where the power management cannot keep up with activity, i.e. $N > N_{max}$. Similarly, the PM overhead is $5.6 - 11.3\times$ larger for PT and $1775 - 3563\times$ larger for the original data from [81] compared to BlitzCoin.

VII. Comparison with Related Works

Various prior works have addressed SoC power management, either leveraging hardware designed for fast response times or decentralized implementations for scalability. However, none of them address the full set of challenges heterogeneous SoCs pose, namely, the need for efficient fine-grained spatial and temporal DVFS, fast and scalable response time and flexible power allocation, even when PEs do not support local software programming.

Table I summarizes the key characteristics of the most relevant prior art and their limitations compared to our approach. Cebrian et al. were the first to propose a token-based (i.e. quantized power units) power-management scheme with a custom hardware unit to enforce a global power cap in a multi-core SoC [50]. Though the system provides a fast response time, the authors state that its centralized nature would prevent scalability past $N > 32$ and finer DVFS levels. Integrated switched-regulator-based DVFS is shown to provide sub-μs response time in [51], [65], but they do not address the scalability of multi-core management. A fully-decentralized, scalable DVFS voting mechanism is described in [49]. However it only addresses power management of the NoC itself, which is a small contributor to the total power in the systems we consider ($\sim 10\%$ in [47]); because of the lack of coordination between domains, it does not enforce a global power cap. TokenSmart presents a fully decentralized token-based power-management design [43], [46]. However, it relies on a software implementation in each CPU core, and because of its sequential token-passing strategy, its convergence time scales linearly with the number of units N, similarly to centralized schemes. Mantovani et al. propose a power-management scheme for heterogeneous SoCs [42], where power allocation decisions are done on decentralized hardware, but the power cap is enforced by a centralized daemon, resulting in the same $O(N)$ scalability. Moreover, validation on [42], [49], [50] is limited to FPGAs and simulations, as opposed to our silicon-proven work. In [81], the authors propose a hierarchical price-theory-based power allocation. However, it is restricted to a SW implementation, its DVFS granularity is limited to the cluster level, and its hierarchical topology still requires a centralized controller, as well as multiple levels of SoC partitioning. As shown in Fig. 21, a direct comparison of their reported results with BlitzCoin shows a 3$\times$ slower response times, even accounting for a HW implementation of their solution.

Additional works like AgilePkgC [82] and AgileWatts [83] demonstrate hardware-enabled fast response time to CPU activity changes using hybrid voltage-regulator designs but do not address either the area and integration overheads of complex voltage regulators or the power allocation in multi-accelerator SoCs and its related scalability challenges. Juang *et al.* [84] propose distributed power management for homogeneous multi-cores, which allocates power in priority to the task with the longest completion time estimated by the task queue depth. However, this does not translate to heterogeneous SoCs, as workload durations are highly accelerator-dependent, unlike our work which does not rely on task queues. In [85], the authors compare hierarchical, gradient-ascent, and decentralized forms of the MaxBIPS policy proposed in [86] but report super-linear response-time scaling for their methods. Vega et al. conceptually introduce decentralized token-based power management in [87] but do not provide implementation or evaluation details.

VIII. Conclusions

In this paper, we presented *BlitzCoin*, a fully decentralized hardware power-management strategy for accelerator-rich SoCs that leverages unified voltage and frequency regulation. BlitzCoin achieves highly efficient tile-level DVFS with sub-μs response times on many-accelerator SoCs. We validated BlitzCoin in a fabricated 12 nm SoC, demonstrating both a negligible area overhead ($< 1\%$) and full functionality. To further characterize its performance across multiple use cases, we conducted cycle-accurate, full-system simulations of BlitzCoin-enabled SoCs targeting different application domains. Compared to a state-of-the-art centralized power-management solution, BlitzCoin shows markedly faster response times (8-12$\times$) and throughput (25%-34%), allowing us to scale its application to 7-13$\times$ larger SoCs. We have released the design of BlitzCoin as open-source hardware in the public domain; please refer to the Artifact Evaluation Appendix for further information.

Appendix A
Artifact Evaluation

A. Abstract

To build and evaluate BlitzCoin, we performed a combination of Monte-Carlo simulation-based evaluations of the coin-exchange algorithm (Section III), RTL simulations of its integration in various SoCs using a 12nm technology (Sections VI.A VI.B), silicon measurements (Section VI.C) and analytical scaling (Section VI.D).

The SoC integration was performed leveraging the ESP framework (https://github.com/sld-columbia/esp). In a public fork of ESP, we have released the coin exchange emulator, the full RTL of BlitzCoin with an example of a technology-independent SoC, and the analytical scaling model. We also plan to merge the RTL implementation of BlitzCoin to the main ESP repository on GitHub. This document contains the instructions for reproducing the experiments presented in the paper.

Due to the proprietary 12nm technology used and practical aspects of experimental measurements on a fabricated ASIC, some results cannot be directly reproduced.

B. Artifact check-list (meta-information)

- **Algorithm:** Novel power allocation utilizing a distributed coin exchange algorithm, implemented in hardware as RTL, as well as tested in a custom Python emulator.
- **Program:** Baremetal workloads invoking different accelerators using BlitzCoin as well as two different baselines.
- **Compilation:** Provided Makefiles compile the baremetal code for the CVA6 core of the simulated SoC.
- **Run-time environment:** SoC configuration and simulation using one of the operating systems supported by ESP. Tested on Red Hat Enterprise Linux 8.
- **Hardware:** Paper results were obtained through simulation in a proprietary 12nm technology and on fabricated silicon. To enable reproducibility, we have provided a technology-independent RTL simulation environment.
- **Metrics:** For the emulator: convergence time and number of packets exchanged. For RTL simulations: execution time, respo time and power traces.
- **Output:** CSV data with post-processing scripts for figure generation.
- **Experiments:** Python-based emulation, RTL simulations and Python-based analytical scaling.
- **How much disk space required (approximately)?** Approx. 32GB
- **How much time is needed to prepare workflow (approximately)?** The emulator is already set up. The RTL simulation environment takes approximately 2-3 hours to prepare via the ESP setup guide.
- **How much time is needed to complete experiments (approximately)?** Emulator runs take several minutes up to a few hours. RTL simulations take approximately 8 hours per experiment. Analytical scaling is almost instant.
- **Publicly available?** Yes, it is currently in a public fork of ESP. We also plan to merge it into main ESP release [88].
- **Code licenses :** Apache 2.0
- **Archived (provide DOI)?** Yes, DOI: 10.5281/zenodo.10995290.

C. Description

1) How to access: We integrated BlitzCoin in ESP and we released it on a public GitHub fork (https://github.com/karthiksv/esp_isca_ae) and on Zenodo (10.5281/zenodo.10995290).

The most relevant directories and files for the integration and evaluation of BlitzCoin are the following:

- `emulator`: Python-based emulation of the coin exchange.
- `analytical_scaling`: Python-based scaling of convergence results for large SoCs.
- `rtl/sockets/dvfs`: RTL implementation of BlitzCoin and NoC integration.
- `socs/esp_asic_generic`: technology-independent 3x3 SoC design for RTL simulations.
- `soft/common/apps/baremetal`: C code running on the SoC's CVA6 core to execute BlitzCoin and the centralized baselines (BC-C and C-RR) on representative workloads.

2) Hardware dependencies: As reproducing the measured results on the actual fabricated ASIC or with the proprietary 12nm technology models is not feasible, we have instead shared a simulation environment using technology-independent RTL. Hence, all results are obtained with RTL simulations rather than FPGA emulation or ASIC measurements, and no specific HW is required.

3) Software dependencies: The software dependencies of ESP are described in the "How to: setup" guide (https://esp.cs.columbia.edu/docs/setup/setup-guide/), specifically in the sections "Software packets", "CAD tools", "Environment variables", and "Docker" for users interested in using the ESP Docker image. In terms of commercial tools, evaluating BlitzCoin requires the Cadence Xcelium Logic Simulator (version 19.03) for running RTL simulations, Xilinx Vivado (version 2023.2) for various IPs used in ESP, and Stratus HLS (version 20.24) for generating the accelerators used in our evaluation. The simulation and compilation flow has been run on RHEL 7 and RHEL 8 machines. The emulator-generated data can then be plotted using the user's software of choice.

D. Installation

Please refer to the `README.md` of the git repository for complete details regarding the installation process for the full BlitzCoin RTL simulation.

The emulation and analytical scaling modules are Python scripts and do not require specific installation.

Evaluating BlitzCoin requires cloning our fork of ESP (https://github.com/karthiksv/esp_isca_ae) and initializing the submodules:

```
git clone
https://github.com/karthiksv/esp_isca_ae.git
cd esp_isca_ae
./utils/scripts/submodule_init.sh
```

The only submodules needed for these experiments are Ariane, Stratus HLS and NVDLA; you may answer no for installing the remainder.

After this, run HLS to generate the RTL for the FFT and Viterbi accelerators. This involves the following steps:

```
cd socs/xilinx-vcu118-xcvu9p
make fft2_stratus-hls
make vitdodec_stratus-hls
sh convert_inferred.sh
```

Then, generate the NVDLA accelerator and the full SoC:

```
cd socs/esp_asic_generic
make NV_NVDLA
make esp-config
make socketgen
```

The 3x3 SoC configuration from Fig. 12 can be viewed in GUI mode with:

```
make esp-xconfig
```

Lastly, the software is compiled with the following commands to generate the BC, BC-C, and C-RR executables:

```
make token_pm_3x3-baremetal
make token_pm_3x3_BCC-baremetal
make token_pm_3x3_CRR-baremetal
```

E. Experiment workflow

After completing the steps above, the emulation and simulation experiments can be run.

For Python emulation: Follow the directions in the `README.txt` of the directories `emulation` and `analytical_scaling`.

For RTL simulation: These can be run from the corresponding SoC directory `socs/esp_asic_generic`:

Fist, generate the text files used to load the bootloader onto the SoC:

```
python3 ../../utils/scripts/
file_handling/bin2text.py ariane
```

Then, lanuch the RTL simulations in graphical mode.

```
TEST_PROGRAM=soft-build/ariane/baremetal/
<exec>.exe make xmsim-gui
```

Where `<exec>` is replaced by the desired program, either `token_pm_3x3`, `token_pm_3x3_BCC` or `token_pm_3x3_CRR`.

To parallelize simulations, you can create different copies of the `socs/esp_asic_generic` directory for the BlitzCoin, BCC and CRR runs.

F. Evaluation and expected results

For Python emulation: The generated CSV data is evaluated by plotting it using the instructions in `emulation/README.txt`. We also provided the original data referenced in the paper in Microsoft Excel format. Due to statistical variations of Monte Carlo simulations, the data may vary slightly compared to the reference, but trends and relative comparisons are expected to hold.

For RTL simulation: The Xcelium simulation is run in interactive mode and the activity waveforms of the different tiles are exported as a CSV file. When starting the simulator, load the saved Simvision waveform configurations (`socs/esp_asic_generic/restore_all.tcl.svcf`). The simulation completes after about 2500 us of runtime, once all 6 tile activity signals are back to 0. At this point, stop the simulation and export the data following the directions in `socs/esp_asic_generic/README.txt`.

We have provided Python scripts for post-processing the CSV files and generating the power traces shown in Figure 16. After exporting the CSV file, it can be run with the following command:

```
python3 socs/esp_asic_generic/post_process.py
```

The same CSV file is also used to derive the execution time and response time for the different experiments. The experiments conducted in the paper were performed with a proprietary 12nm technology, while the publicly available repository uses technology-independent RTL. This can result in small differences in performance compared to the results published in the paper.

References

[1] Efraim Rotem, Alon Naveh, Avinash Ananthakrishnan, Eliezer Weissmann, and Doron Rajwan. Power-Management Architecture of the Intel Microarchitecture Code-Named Sandy Bridge. *IEEE Micro*, 32(2):20–27, 2012.

[2] Ronak Singhal. Inside Intel® Core microarchitecture (Nehalem). In *2008 IEEE Hot Chips 20 Symposium (HCS)*, pages 1–25, 2008.

[3] Dominik Brodowski, Rafael J. Wysocki, and Viresh Kumar. Cpu frequency and voltage scaling code in the linux(tm) kernel. https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt, 2010.

[4] Venkatesh Pallipadi and Alexey Starikovski. The Ondemand Governor. In *Proceedings of the Ottawa Linux Symposium*, volume 2, pages 215–230, 2006.

[5] Hye-Young McCreary, Martha A. Broyles, Michael S. Floyd, Andrew J. Geissler, Steven P. Hartman, Freeman L. Rawson III, Todd J. Rosedahl, Juan C. Rubio, and Malcolm S. Ware. Energyscale for IBM POWER6 microprocessor-based systems. *IBM J. Res. Dev.*, 51(6):775–786, 2007.

[6] Zeynep Toprak Deniz, Michael A. Sperling, John F. Bulzacchelli, Gregory S. Still, Ryan Kruse, Seongwon Kim, David Boerstler, Tilman Gloekler, Raphael Robertazzi, Kevin Stawiasz, Tim Diemoz, George English, David Hui, Paul Muench, and Joshua Friedrich. 5.2 distributed system of digitally controlled microregulators enabling per-core DVFS for the POWER8™ microprocessor. In *2014 IEEE International Conference on Solid-State Circuits Conference, ISSCC 2014, Digest of Technical Papers, San Francisco, CA, USA, February 9-13, 2014*, pages 98–99. IEEE, 2014.

[7] Christopher J. Berry, David Wolpert, Christos Vezyrtzis, Richard F. Rizzolo, Sean M. Carey, Yaniv Maroz, Hunter F. Shi, Dureseti Chidambarrao, Christian Jacobi, Anthony Saporito, Thomas Strach, Alper Buyuktosunoglu, Preetham Lobo, Pierce Chuang, Pawel Owczarczyk, Ramon Bertran, Tobias Webel, and Phillip J. Restle. IBM z14: Processor characterization and power management for high-reliability mainframe systems. *IEEE J. Solid State Circuits*, 54(1):121–132, 2019.

[8] David Wolpert, Christopher J. Berry, Brian Bell, Adam Jatkowski, Jesse Surprise, John Isakson, Ofer Geva, Brian Deskin, Mark Cichanowski, Dina Hamid, Chris Cavitt, Gregory Fredeman, Dinesh Kannambadi, Anthony Saporito, Ashutosh Mishra, Alper Buyuktosunoglu, Tobias Webel, Preetham Lobo, Ramon Bertran, Pradeep Bhadravati Parashurama, Dureseti Chidambarrao, Brandon Bruen, Alan P. Wagstaff, Eric Lukes, Sean M. Carey, Hunter F. Shi, Michael Romain, Paul Logsdon, and Ishita Agarwal. Cores, cache, content, and characterization: IBM's second generation 14-nm product, z15. *IEEE J. Solid State Circuits*, 56(1):98–111, 2021.

[9] Sam Naffziger. Foxton technology. In *2005 IEEE Hot Chips XVII Symposium (HCS)*, pages 1–21, 2005.

[10] Jack Doweck, Wen-Fu Kao, Allen Kuan-yu Lu, Julius Mandelblat, Anirudha Rahatekar, Lihu Rappoport, Efraim Rotem, Ahmad Yasin, and Adi Yoaz. Inside 6th-Generation Intel Core: New Microarchitecture Code-Named Skylake. *IEEE Micro*, 37(2):52–62, 2017.

[11] Tim Fischer, Jayen Desai, Bruce Andrew Doyle, Samuel Naffziger, and Ben Patella. A 90-nm variable frequency clock system for a power-managed itanium architecture processor. *IEEE J. Solid State Circuits*, 41(1):218–228, 2006.

[12] Eyal Fayneh, Marcelo Yuffe, Ernest Knoll, Michael Zelikson, Muhammad Abozaed, Yair Talker, Ziv Shmuely, and Saher Abu Rahme. 4.1 14nm 6th-generation Core processor SoC with low power consumption and improved performance. In *2016 IEEE International Solid-State Circuits Conference, ISSCC 2016, San Francisco, CA, USA, January 31 - February 4, 2016*, pages 72–73. IEEE, 2016.

[13] Howard David, Eugene Gorbatov, Ulf R. Hanebutte, Rahul Khanna, and Christian Le. RAPL: memory power estimation and capping. In *Proceedings of the 2010 International Symposium on Low Power Electronics and Design, 2010, Austin, Texas, USA, August 18-20, 2010*, pages 189–194. ACM, 2010.

[14] Pradip Bose, Alper Buyuktosunoglu, John A. Darringer, Meeta Sharma Gupta, Michael B. Healy, Hans M. Jacobson, Indira Nair, Jude A. Rivers, Jeonghee Shin, Augusto Vega, and Alan J. Weger. Power management of multi-core chips: Challenges and pitfalls. In *2012 Design, Automation & Test in Europe Conference & Exhibition, DATE 2012, Dresden, Germany, March 12-16, 2012*, pages 977–982. IEEE, 2012.

[15] David M. Brooks, Pradip Bose, Stanley Schuster, Hans M. Jacobson, Prabhakar Kudva, Alper Buyuktosunoglu, John-David Wellman, Victor V. Zyuban, Manish Gupta, and Peter W. Cook. Power-Aware Microarchitecture: Design and Modeling Challenges for Next-Generation Microprocessors. *IEEE Micro*, 20(6):26–44, 2000.

[16] Alper Buyuktosunoglu, Tejas Karkhanis, David H. Albonesi, and Pradip Bose. Energy Efficient Co-Adaptive Instruction Fetch and Issue. In *30th International Symposium on Computer Architecture (ISCA 2003), 9-11 June 2003, San Diego, California, USA*, pages 147–156. IEEE Computer Society, 2003.

[17] Augusto Vega, Alper Buyuktosunoglu, Heather Hanson, Pradip Bose, and Srinivasan Ramani. Crank it up or dial it down: coordinated multiprocessor frequency and folding control. In *The 46th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-46, Davis, CA, USA, December 7-11, 2013*, pages 210–221. ACM, 2013.

[18] Michael S. Floyd, Malcolm Allen-Ware, Karthick Rajamani, Tilman Gloekler, Bishop Brock, Pradip Bose, Alper Buyuktosunoglu, Juan C. Rubio, Birgit Schubert, Bruno Spruth, José A. Tierno, and Lorena Pesantez. Adaptive energy-management features of the IBM POWER7 chip. *IBM J. Res. Dev.*, 55(3):8, 2011.

[19] Pierce I-Jen Chuang, Christos Vezyrtzis, Divya Pathak, Richard F. Rizzolo, Tobias Webel, Thomas Strach, Otto A. Torreiter, Preetham Lobo, Alper Buyuktosunoglu, Ramon Bertran, Michael S. Floyd, Malcolm S. Ware, Gerard Salem, Sean M. Carey, and Phillip Restle. 26.2 power supply noise in a 22nm z13™ microprocessor. In *2017 IEEE International Solid-State Circuits Conference, ISSCC 2017, San Francisco, CA, USA, February 5-9, 2017*, pages 438–439. IEEE, 2017.

[20] Christos Vezyrtzis, Thomas Strach, Pierce I-Jen Chuang, Preetham Lobo, Richard F. Rizzolo, Tobias Webel, Pawel Owczarczyk, Alper Buyuktosunoglu, Ramon Bertran, David T. Hui, Susan M. Eickhoff, Michael S. Floyd, Gerard Salem, Sean M. Carey, Stelios G. Tsapepas, and Phillip J. Restle. Droop mitigation using critical-path sensors and an on-chip distributed power supply estimation engine in the z14™ enterprise processor. In *2018 IEEE International Solid-State Circuits Conference, ISSCC 2018, San Francisco, CA, USA, February 11-15, 2018*, pages 300–302. IEEE, 2018.

[21] Tobias Webel, Preetham M. Lobo, Thomas Strach, Pradeep Bhadravati Parashurama, Srinivas Purushotham, Ramon Bertran, and Alper Buyuktosunoglu. Proactive power management in IBM z15. *IBM J. Res. Dev.*, 64(5/6):15:1–15:12, 2020.

[22] Bo Su, Junli Gu, Li Shen, Wei Huang, Joseph L. Greathouse, and Zhiying Wang. PPEP: Online Performance, Power, and Energy Prediction Framework and DVFS Space Exploration. In *47th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2014, Cambridge, United Kingdom, December 13-17, 2014*, pages 445–457. IEEE Computer Society, 2014.

[23] William Lloyd Bircher and Sam Naffziger. AMD SOC power management: Improving performance/watt using run-time feedback. In *Proceedings of the IEEE 2014 Custom Integrated Circuits Conference, CICC 2014, San Jose, CA, USA, September 15-17, 2014*, pages 1–4. IEEE, 2014.

[24] Thomas Burd, Noah Beck, Sean White, Milam Paraschou, Nathan Kalyanasundharam, Gregg Donley, Alan Smith, Larry Hewitt, and Samuel Naffziger. ”Zeppelin”: An SoC for Multichip Architectures. *IEEE J. Solid State Circuits*, 54(1):133–143, 2019.

[25] Sriram Sundaram, Aaron Grenat, Samuel Naffziger, Tom Burd, Stephen Kosonocky, Steve Liepe, Ravinder Rachala, Miguel Rodriguez, Michael Austin, and Sriram Sambamurthy. Bristol Ridge: A 28-nm × 86 Performance-Enhanced Microprocessor Through System Power Management. *IEEE Journal of Solid-State Circuits*, 52(1):89–97, 2017.

[26] Vijay Kiran Kalyanam, Eric Mahurin, Keith A. Bowman, and Suresh Venkumahanti. 35.3 Thread-Level Power Management for a Current- and Temperature-Limiting System in a 7nm Hexagon™ Processor. In *IEEE International Solid-State Circuits Conference, ISSCC 2021, San Francisco, CA, USA, February 13-22, 2021*, pages 494–496. IEEE, 2021.

[27] Krishna K. Rangan, Gu-Yeon Wei, and David M. Brooks. Thread motion: fine-grained power management for multi-core systems. In *36th International Symposium on Computer Architecture (ISCA 2009), June 20-24, 2009, Austin, TX, USA*, pages 302–313. ACM, 2009.

[28] Harshad Kasture, Davide B. Bartolini, Nathan Beckmann, and Daniel Sánchez. Rubik: fast analytical power management for latency-critical systems. In *Proceedings of the 48th International Symposium on Microarchitecture, MICRO 2015, Waikiki, HI, USA, December 5-9, 2015*, pages 598–610. ACM, 2015.

[29] Vasileios Spiliopoulos, Stefanos Kaxiras, and Georgios Keramidas. Green governors: A framework for continuously adaptive DVFS. In *2011 International Green Computing Conference and Workshops, IGCC 2012, Orlando, FL, USA, July 25-28, 2011*, pages 1–8. IEEE Computer Society, 2011.

[30] Sebastian Herbert and Diana Marculescu. Analysis of dynamic voltage/frequency scaling in chip-multiprocessors. In *Proceedings of the 2007 International Symposium on Low Power Electronics and Design, 2007, Portland, OR, USA, August 27-29, 2007*, pages 38–43. ACM, 2007.

[31] Radu Teodorescu and Josep Torrellas. Variation-Aware Application Scheduling and Power Management for Chip Multiprocessors. In *35th International Symposium on Computer Architecture (ISCA 2008), June 21-25, 2008, Beijing, China*, pages 363–374. IEEE Computer Society, 2008.

[32] Yefu Wang, Kai Ma, and Xiaorui Wang. Temperature-constrained power control for chip multiprocessors with online model estimation. In *36th International Symposium on Computer Architecture (ISCA 2009), June 20-24, 2009, Austin, TX, USA*, pages 314–324. ACM, 2009.

[33] Abhishek Bhattacharjee and Margaret Martonosi. Thread criticality predictors for dynamic performance, power, and resource management in chip multiprocessors. In *36th International Symposium on Computer Architecture (ISCA 2009), June 20-24, 2009, Austin, TX, USA*, pages 290–301. ACM, 2009.

[34] Huazhe Zhang and Henry Hoffmann. Maximizing Performance Under a Power Cap: A Comparison of Hardware, Software, and Hybrid Techniques. In *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2016, Atlanta, GA, USA, April 2-6, 2016*, pages 545–559. ACM, 2016.

[35] Ayse Kivilcim Coskun, Tajana Simunic Rosing, and Kenny C. Gross. Utilizing Predictors for Efficient Thermal Management in Multiprocessor SoCs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 28(10):1503–1516, 2009.

[36] Ke Meng, Russ Joseph, Robert P. Dick, and Li Shang. Multi-optimization power management for chip multiprocessors. In *17th International Conference on Parallel Architectures and Compilation Techniques, PACT 2008, Toronto, Ontario, Canada, October 25-29, 2008*, pages 177–186. ACM, 2008.

[37] Raghavendra Pradyumna Pothukuchi, Joseph L. Greathouse, Karthik Rao, Christopher Erb, Leonardo Piga, Petros G. Voulgaris, and Josep Torrellas. Tangram: Integrated Control of Heterogeneous Computers. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2019, Columbus, OH, USA, October 12-16, 2019*, pages 384–398. ACM, 2019.

[38] Bryan Donyanavard, Tiago Mück, Santanu Sarma, and Nikil D. Dutt. SPARTA: runtime task allocation for energy efficient heterogeneous many-cores. In *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, CODES 2016, Pittsburgh, Pennsylvania, USA, October 1-7, 2016*, pages 27:1–27:10. ACM, 2016.

[39] Jawad Haj-Yahya, Mohammed Alser, Jeremie S. Kim, Abdullah Giray Yaglikçi, Nandita Vijaykumar, Efraim Rotem, and Onur Mutlu. SysScale: Exploiting Multi-domain Dynamic Voltage and Frequency Scaling for Energy Efficient Mobile Processors. In *47th ACM/IEEE Annual International Symposium on Computer Architecture, ISCA 2020, Valencia, Spain, May 30 - June 3, 2020*, pages 227–240. IEEE, 2020.

[40] Peter Bailis, Vijay Janapa Reddi, Sanjay Gandhi, David M. Brooks, and Margo I. Seltzer. Dimetrodon: processor-level preventive thermal management via idle cycle injection. In *Proceedings of the 48th Design Automation Conference, DAC 2011, San Diego, California, USA, June 5-10, 2011*, pages 89–94. ACM, 2011.

[41] Yakun Sophia Shao, Brandon Reagen, Gu-Yeon Wei, and David M. Brooks. The Aladdin Approach to Accelerator Design and Modeling. *IEEE Micro*, 35(3):58–70, 2015.

[42] Paolo Mantovani, Emilio G. Cota, Kevin Tien, Christian Pilato, Giuseppe Di Guglielmo, Kenneth L. Shepard, and Luca P. Carloni. An FPGA-based infrastructure for fine-grained DVFS analysis in high-performance embedded systems. In *Proceedings of the 53rd Annual Design Automation Conference, DAC 2016, Austin, TX, USA, June 5-9, 2016*, pages 157:1–157:6. ACM, 2016.

[43] Parth Shah, Ranjal Gautham Shenoy, Vaidyanathan Srinivasan, Pradip Bose, and Alper Buyuktosunoglu. Tokensmart: Distributed, scalable power management in the many-core era. *ACM Trans. Archit. Code Optim.*, 20(1), Nov. 2022.

[44] Reinaldo A. Bergamaschi, Guoling Han, Alper Buyuktosunoglu, Hiren D. Patel, Indira Nair, Gero Dittmann, Geert Janssen, Nagu R. Dhanwada, Zhigang Hu, Pradip Bose, and John A. Darringer. Exploring power management in multi-core systems. In *Proceedings of the 13th Asia South Pacific Design Automation Conference, ASP-DAC 2008, Seoul, Korea, January 21-24, 2008*, pages 708–713. IEEE, 2008.

[45] Ryan Cochran, Can Hankendi, Ayse K. Coskun, and Sherief Reda. Pack & Cap: adaptive DVFS and thread packing under power caps. In *44rd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2011, Porto Alegre, Brazil, December 3-7, 2011*, pages 175–185. ACM, 2011.

[46] Parth Shah, Ranjal Gautham Shenoy, Vaidyanathan Srinivasan, Pradip Bose, and Alper Buyuktosunoglu. TokenSmart: Distributed, Scalable Power Management in the Many-Core Era. *IEEE Comput. Archit. Lett.*, 20(1):42–45, 2021.

[47] Tianyu Jia, Paolo Mantovani, Maico Cassel Dos Santos, Davide Giri, Joseph Zuckerman, Erik Jens Loscalzo, Martin Cochet, Karthik Swaminathan, Gabriele Tombesi, Jeff Jun Zhang, Nandhini Chandramoorthy, John-David Wellman, Kevin Tien, Luca Carloni, Kenneth Shepard, David Brooks, Gu-Yeon Wei, and Pradip Bose. A 12nm Agile-Designed SoC for Swarm-Based Perception with Heterogeneous IP Blocks, a Reconfigurable Memory Hierarchy, and an 800MHz Multi-Plane NoC. In *ESSCIRC 2022- IEEE 48th European Solid State Circuits Conference (ESSCIRC)*, pages 269–272, 2022.

[48] Eric J. Fluhr, Rahul M. Rao, Howard Smith, Alper Buyuktosunoglu, and Ramon Bertran Monfort. IBM POWER9 circuit design and energy optimization for 14-nm technology. *IBM J. Res. Dev.*, 62(4/5):4:1–4:11, 2018.

[49] Yuan Yao and Zhonghai Lu. DVFS for NoCs in CMPs: A thread voting approach. In *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 309–320, 2016.

[50] Juan M. Cebrian, Juan L. Aragón, and Stefanos Kaxiras. Power Token Balancing: Adapting CMPs to Power Constraints for Parallel Multithreaded Workloads. In *25th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2011, Anchorage, Alaska, USA, 16-20 May, 2011 - Conference Proceedings*, pages 431–442. IEEE, 2011.

[51] Ben Keller, Martin Cochet, Brian Zimmer, Jaehwa Kwak, Alberto Puggelli, Yunsup Lee, Milovan Blagojevic, Stevo Bailey, Pi-Feng Chiu, Daniel Palmer Dabbelt, Colin Schmidt, Elad Alon, Krste Asanovic, and Borivoje Nikolic. A RISC-V Processor SoC With Integrated Power Management at Submicrosecond Timescales in 28 nm FD-SOI. *IEEE J. Solid State Circuits*, 52(7):1863–1875, 2017.

[52] Edward A. Burton, Gerhard Schrom, Fabrice Paillet, Jonathan Douglas, William J. Lambert, Kaladhar Radhakrishnan, and Michael J. Hill. FIVR — Fully integrated voltage regulators on 4th generation Intel® Core™ SoCs. In *2014 IEEE Applied Power Electronics Conference and Exposition - APEC 2014*, pages 432–439, 2014.

[53] Jun-Eun Park, Jeongho Hwang, Jonghyun Oh, and Deog-Kyoon Jeong. 32.4 A 0.4-to-1.2V 0.0057mm2 55fs-Transient-FoM Ring-Amplifier-Based Low-Dropout Regulator with Replica-Based PSR Enhancement. In *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, pages 492–494, 2020.

[54] Suyoung Bang, Wootaek Lim, Charles Augustine, Andres Malavasi, Muhammad Khellah, James Tschanz, and Vivek De. 25.1 a fully synthesizable distributed and scalable all-digital ldo in 10nm cmos. In *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, pages 380–382, 2020.

[55] Sae Kyu Lee, Paul N. Whatmough, Marco Donato, Glenn G. Ko, David Brooks, and Gu-Yeon Wei. SMIV: A 16-nm 25-mm² SoC for IoT With Arm Cortex-A53, eFPGA, and Coherent Accelerators. *IEEE Journal of Solid-State Circuits*, 57(2):639–650, 2022.

[56] Brian Zimmer, Yunsup Lee, Alberto Puggelli, Jaehwa Kwak, Ruzica Jevtić, Ben Keller, Steven Bailey, Milovan Blagojević, Pi-Feng Chiu, Hanh-Phuc Le, Po-Hung Chen, Nicholas Sutardja, Rimas Avizienis, Andrew Waterman, Brian Richards, Philippe Flatresse, Elad Alon, Krste Asanović, and Borivoje Nikolić. A RISC-V Vector Processor With Simultaneous-Switching Switched-Capacitor DC–DC Converters in 28 nm FDSOI. *IEEE Journal of Solid-State Circuits*, 51(4):930–942, 2016.

[57] S. Das, D. Roberts, Seokwoo Lee, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. A self-tuning DVS processor using delay-error detection and correction. *IEEE Journal of Solid-State Circuits*, 41(4):792–804, 2006.

[58] Jaehwa Kwak and Borivoje Nikolić. A Self-Adjustable Clock Generator With Wide Dynamic Range in 28 nm FDSOI. *IEEE Journal of Solid-State Circuits*, 51(10):2368–2379, 2016.

[59] Fahim ur Rahman, Sung Kim, Naveen John, Roshan Kumar, Xi Li, Rajesh Pamula, Keith A. Bowman, and Visvesh S. Sathe. A Unified Clock and Switched-Capacitor-Based Power Delivery Architecture for Variation Tolerance in Low-Voltage SoC Domains. *IEEE Journal of Solid-State Circuits*, 54(4):1173–1184, 2019.

[60] Xuliang Wang, Xiaosen Liu, and Wing-Hung Ki. A Self-Clocked and Variation-Tolerant Unified Voltage-and-Frequency Regulator for In-Order Executed Digital Loads. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 70(11):4627–4640, 2023.

[61] David Bol, Julien De Vos, Cédric Hocquet, François Botman, François Durvaux, Sarah Boyd, Denis Flandre, and Jean-Didier Legat. SleepWalker: A 25-MHz 0.4-V Sub- mm^2 7- μW/MHz Microcontroller in 65-nm LP/GP CMOS for Low-Carbon Wireless Sensor Nodes. *IEEE Journal of Solid-State Circuits*, 48(1):20–32, 2013.

[62] Xuliang Wang, Xiaosen Liu, and Wing-Hung Ki. A self-clocked and variation-tolerant unified voltage-and-frequency regulator for in-order executed digital loads. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 70(11):4627–4640, 2023.

[63] Brent Bohnenstiehl, Aaron Stillmaker, Jon J. Pimentel, Timothy Andreas, Bin Liu, Anh Tran, Emmanuel Adeagbo, and Bevan M. Baas. KiloCore: A 32-nm 1000-Processor Computational Array. *IEEE J. Solid State Circuits*, 52(4):891–902, 2017.

[64] Brian Zimmer, Rangharajan Venkatesan, Yakun Sophia Shao, Jason Clemons, Matthew Fojtik, Nan Jiang, Ben Keller, Alicia Klinefelter, Nathaniel Ross Pinckney, Priyanka Raina, Stephen G. Tell, Yanqing Zhang, William J. Dally, Joel S. Emer, C. Thomas Gray, Stephen W. Keckler, and Brucek Khailany. A 0.32-128 TOPS, Scalable Multi-Chip-Module-Based Deep Neural Network Inference Accelerator With Ground-Referenced Signaling in 16 nm. *IEEE J. Solid State Circuits*, 55(4):920–932, 2020.

[65] John Charles Wright, Colin Schmidt, Ben Keller, Daniel Palmer Dabbelt, Jaehwa Kwak, Vighnesh Iyer, Nandish Mehta, Pi-Feng Chiu, Stevo Bailey, Krste Asanović, and Borivoje Nikolić. A Dual-Core RISC-V Vector Processor With On-Chip Fine-Grain Power Management in 28-nm FD-SOI. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(12):2721–2725, 2020.

[66] Paolo Mantovani, Davide Giri, Giuseppe Di Guglielmo, Luca Piccolboni, Joseph Zuckerman, Emilio G. Cota, Michele Petracca, Christian Pilato, and Luca P. Carloni. Agile SoC Development with Open ESP : Invited Paper. In *IEEE/ACM International Conference On Computer Aided Design, ICCAD 2020, San Diego, CA, USA, November 2-5, 2020*, pages 96:1–96:9. IEEE, 2020.

[67] Luca P. Carloni. The Case for Embedded Scalable Platforms. In *Proceedings of the Design Automation Conference (DAC)*, pages 17:1–17:6, June 2016.

[68] Florian Zaruba and Luca Benini. The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-Ready 1.7-GHz 64-Bit RISC-V Core in 22-nm FDSOI Technology. *IEEE Trans. Very Large Scale Integr. Syst.*, 27(11):2629–2640, 2019.

[69] Joseph Zuckerman, Paolo Mantovani, Davide Giri, and Luca P. Carloni. Enabling Heterogeneous, Multicore SoC Research with RISC-V and ESP. 2022.

[70] Emilio G. Cota, Paolo Mantovani, Giuseppe Di Guglielmo, and Luca P. Carloni. An Analysis of Accelerator Coupling in Heterogeneous Architectures. In *Proceedings of the Design Automation Conference (DAC)*, DAC'15, pages 202:1–202:6, New York, NY, USA, June 2015. ACM.

[71] Davide Giri, Kuan-Lin Chiu, Guy Eichler, Paolo Mantovani, and Luca P. Carloni. Accelerator Integration for Open-Source SoC Design. *IEEE Micro (Special Issue: FPGAs in Computing)*, 41(4):8–14, 2021.

[72] Davide Giri, Paolo Mantovani, and Luca P. Carloni. NoC-Based Support of Heterogeneous Cache-Coherence Models for Accelerators. 2018.

[73] Joseph Zuckerman, Davide Giri, Jihye Kwon, Paolo Mantovani, and Luca P. Carloni. Cohmeleon: Learning-Based Orchestration of Accelerator Coherence in Heterogeneous SoCs. In *Proceedings of the IEEE/ACM Symposium on Microarchitecture (MICRO)*, 2021.

[74] Maico Cassel Dos Santos, Tianyu Jia, Martin Cochet, Karthik Swaminathan, Joseph Zuckerman, Paolo Mantovani, Davide Giri, Jeff Jun Zhang, Erik Jens Loscalzo, Gabriele Tombesi, Kevin Tien, Nandhini Chandramoorthy, John-David Wellman, David Brooks, Gu-Yeon Wei, Kenneth Shepard, Luca Carloni, and Pradip Bose. A Scalable Methodology for Agile Chip Development with Open-Source Hardware Components. In *Proceedings of the IEEE International Conference on Computer-Aided Design (ICCAD)*, 2022.

[75] Wei Huang, Charles Lefurgy, William Kuk, Alper Buyuktosunoglu, Michael Floyd, Karthick Rajamani, Malcolm Allen-Ware, and Bishop Brock. Accurate Fine-Grained Processor Power Proxies. In *2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 224–234, 2012.

[76] IBM. Mini-ERA: Simplified Version of the Main ERA Workload. https://github.com/IBM/mini-era, 2021.

[77] Davide Giri, Kuan-Lin Chiu, Giuseppe Di Guglielmo, Paolo Mantovani, and Luca P. Carloni. ESP4ML: Platform-Based Design of Systems-on-Chip for Embedded Machine Learning. In *2020 Design, Automation & Test in Europe Conference & Exhibition, DATE 2020, Grenoble, France, March 9-13, 2020*, pages 1049–1054. IEEE, 2020.

[78] Joules RTL Power Solution. https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/power-analysis/joules-rtl-power-solution.html.

[79] Arm Compiler for Embedded User Guide. https://developer.arm.com/documentation/100748/0618.

[80] Maico Cassel Dos Santos, Tianyu Jia, Joseph Zuckerman, Martin Cochet, Davide Giri, Erik Jens Loscalzo, Karthik Swaminathan, Thierry Tambe, Jeff Jun Zhang, Alper Buyuktosunoglu, Kuan-Lin Chiu, Giuseppe Di Guglielmo, Paolo Mantovani, Luca Piccolboni, Gabriele Tombesi, David Trilla, John-David Wellman, En-Yu Yang, Aporva Amarnath, Ying Jing, Bakshree Mishra, Joshua Park, Vignesh Suresh, Sarita Adve, Pradip Bose, David Brooks, Luca P. Carloni, Kenneth L. Shepard, and Gu-Yeon Wei. 14.5 A 12nm Linux-SMP-Capable RISC-V SoC with 14 Accelerator Types, Distributed Hardware Power Management and Flexible NoC-Based Data Orchestration. In *2024 IEEE International Solid-State Circuits Conference (ISSCC)*, volume 67, pages 262–264, 2024.

[81] Thannirmalai Somu Muthukaruppan, Anuj Pathania, and Tulika Mitra. Price theory based power management for heterogeneous multi-cores. *SIGARCH Comput. Archit. News*, 42(1):161–176, feb 2014.

[82] Georgia Antoniou, Haris Volos, Davide B. Bartolini, Tom Rollet, Yiannakis Sazeides, and Jawad Haj Yahya. AgilePkgC: An Agile System Idle State Architecture for Energy Proportional Datacenter Servers. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 851–867, 2022.

[83] Jawad Haj Yahya, Haris Volos, Davide B. Bartolini, Georgia Antoniou, Jeremie S. Kim, Zhe Wang, Kleovoulos Kalaitzidis, Tom Rollet, Zhirui Chen, Ye Geng, Onur Mutlu, and Yiannakis Sazeides. AgileWatts: An Energy-Efficient CPU Core Idle-State Architecture for Latency-Sensitive Server Applications. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 835–850, 2022.

[84] Philo Juang, Qiang Wu, Li-Shiuan Peh, Margaret Martonosi, and Douglas W. Clark. Coordinated, distributed, formal energy management of chip multiprocessors. In *Proceedings of the 2005 International Symposium on Low Power Electronics and Design, 2005, San Diego, California, USA, August 8-10, 2005*, pages 127–130. ACM, 2005.

[85] John Sartori and Rakesh Kumar. Distributed peak power management for many-core architectures. In *Design, Automation and Test in Europe, DATE 2009, Nice, France, April 20-24, 2009*, pages 1556–1559. IEEE, 2009.

[86] Canturk Isci, Alper Buyuktosunoglu, Chen-Yong Cher, Pradip Bose, and Margaret Martonosi. An Analysis of Efficient Multi-Core Global Power Management Policies: Maximizing Performance for a Given Power Budget. In *39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-39 2006), 9-13 December 2006, Orlando, Florida, USA*, pages 347–358. IEEE Computer Society, 2006.

[87] Augusto Vega, Alper Buyuktosunoglu, and Pradip Bose. Invited paper: Secure swarm intelligence: A new approach to many-core power management. In *2017 IEEE/ACM International Symposium on Low Power Electronics and Design, ISLPED 2017, Taipei, Taiwan, July 24-26, 2017*, pages 1–6. IEEE, 2017.

[88] Columbia SLD Group. ESP Release. www.esp.cs.columbia.edu, 2024.