DozzNoC: Reducing Static and Dynamic Energy in NoCs with Low-latency Voltage Regulators using Machine Learning

Mark Clark[1], Yingping Chen[2], Avinash Karanth[1], Brian Ma[2], and Ahmed Louri[3]

[1]School of Electrical Engineering and Computer Science, Ohio University, Athens, OH 45701.
[2]Department of Electrical and Computer Engineering, University of Texas at Dallas, Dallas, TX
[3]Department of Electrical and Computer Engineering, George Washington University, Washington DC.

***Abstract*—Network-on-chips (NoCs) continues to be the choice of communication fabric in multicore architectures because the NoC effectively combines the resource efficiency of the bus with the parallelizability of the crossbar. As NoC suffers from both high static and dynamic energy consumption, power-gating and dynamic voltage and frequency scaling (DVFS) have been proposed in the literature to improve energy-efficiency. In this work, we propose DozzNoC, an adaptable power management technique that effectively combines power-gating and DVFS techniques to target both static power and dynamic energy reduction with a single inductor multiple output (SIMO) voltage regulator. The proposed power management design is further enhanced by machine learning techniques that predict future traffic load for proactive DVFS mode selection. DozzNoC utilizes a SIMO voltage regulator scheme that allows for fast, low-powered, and independently power-gated or voltage scaled routers such that each router and its outgoing links share the same voltage/frequency domain. Our simulation results using PARSEC and Splash-2 benchmarks on an 8 × 8 mesh network show that for a decrease of 7% in throughput, we can achieve an average dynamic energy savings of 25% and an average static power reduction of 53%.**

I. Introduction

The combined impact of technology scaling (14 nm and beyond) and the insertion of new transistor designs (tri-gate) have enabled a rapid increase in the number of both central processing units (CPUs) and graphical processing units (GPUs). As Network-on-Chips (NoCs) are the glue that connects heterogeneous multicores, memory hierarchies and I/O, the design and implementation of the NoC can significantly impact the power consumption and performance of multicores. Aggressive transistor scaling has resulted in unique power challenges for NoC, particularly the increase in static power due to leakage current and dynamic energy due to switching, storing and routing of packets. Therefore, there is a need for adaptable power management where the NoC consumes energy which is proportional to the multicore bandwidth demands.

Dynamic Voltage and Frequency Scaling (DVFS) is a well-known technique to scale voltage and frequency of the NoC components (routers, links) in proportion to the network load without degrading the throughput of the application [1], [2], [3], [4]. The supply voltage is decreased at low network load and any marginal loss in performance is tolerated in order to save dynamic energy. At medium to high network load, a loss in performance would lead to saturation, dropped packets, and increased network contention, and therefore, the supply voltage is proportionally increased. Recent work has also shown that machine learning techniques can be applied to select the optimal voltage level through proactive predictions of future network parameters which more accurately addresses future network needs than reactive techniques that rely on stale network parameters. [5], [6], [7], [8], [9].

On the other hand, static power may be targeted through power-gating, a technique that switches off the supply voltage to various NoCs components (routers, links) [10], [11], [12], [13], [14], [15], [16], [17], [18], [19] to reduce leakage current. Power-gating is used to maximize static power savings by completely powering off unused or lightly used network components without causing a significant impact on performance. This can be challenging to achieve since there is a large wake-up delay (*T-Wakeup*) and a minimum break-even time (*T-Breakeven*) to power back on components that were switched off[1]. A smart power-gating model will ensure that (i) only unused or lightly used components will be switched off, (ii) switched off components are woken before they cause blocking in the network, and (iii) powered-off components meet or exceed their break-even times in order to ensure that static power savings are maximized.

In this paper, we propose **DozzNoC**, an adaptable power management technique that uses single-input multiple-output (SIMO) voltage regulators to target both static and dynamic energy savings. Our scheme effectively combines power-gating (to target low-network activity) and DVFS (to target variability in network load) with supervised machine learning algorithms in order to create a more energy proportional NoC. Each router in DozzNoC has three operational states - active,

[1]T-Wakeup is the wake-up delay in cycles that a router needs to fully charge local voltage levels up to Vdd. This differs from T-Breakeven which refers to the minimum time that the router, link, or network component must be switched off before powering it back on in order to ensure a net savings in static power.

1530-2075/20/$31.00 ©2020 IEEE
DOI 10.1109/IPDPS47924.2020.00011

inactive and wakeup states; while in an active state, the router selects the appropriate DVFS voltage mode. While in the inactive state the router is power-gated. While in the wakeup state the routers' local voltage level is charged up to Vdd. DozzNoC implements DVFS by capturing several router/NoC features locally (without any global coordination) and predicts the future buffer utilization to proactively select the router's optimal voltage mode. Machine learning (ML) models have been shown to improve prediction accuracy while minimizing model error [20], [21], [22], therefore we use an offline trained linear regression-based ML algorithm to calculate the label (future buffer utilization) that will be used to predict the voltage level of the router. DozzNoC utilizes a SIMO voltage regulator scheme that allows for fast, low-powered, and independently power-gated or voltage scaled routers such that each router and its outgoing links share the same voltage/frequency domain. When applied to an 8×8 mesh network, DozzNoC achieves an average reduction of 25% in dynamic energy and 53% in static power for a loss of 7% in throughput. The major contributions of this work are as follow:

- **Power-Gating+DVFS:**DozzNoC simultaneously combines partially non-blocking power-gating technique with DVFS. This allows power-gating of NoC routers during periods of low network activity to save static power and dynamic voltage scaling during periods of medium to high network activity to reduce dynamic energy consumption.
- **SIMO/LDO Voltage Regulator:** The novelty behind the voltage regulator scheme used in DozzNoC is the combined use of SIMO and low-dropout (LDO) regulator for voltage scaling and power-gating. This allows DozzNoC to not only switch between different voltage levels with low latency, but also to improve the power and area-efficiency.
- **Machine Learning:** DozzNoC applies linear regression-based ML techniques that enable proactive DVFS using fewer router features so as to maximize energy savings with minimal impact on throughput. Offline training and local router features ensure minimal overhead and design scalability.

II. Related Work

DVFS: DVFS has been applied at different levels of granularity (fine-grain versus coarse-grain) to various NoC components (input ports, routers, buffers, crossbars). The design trade-off usually involves balancing the performance loss (throughput, latency) with improved energy savings. Prior works have used various parameters to measure network traffic to decide when to switch voltage modes such as round-trip time (RTT) [4], VFI utilization [23], network slack [24], buffer utilization [2], cache-coherence properties [25] or greedy/proportional-integral models [23]. Recent work has begun to incorporate machine learning algorithms that can predict future network parameters to select the optimum voltage mode [5], [6], [26]. By training the model offline, the overhead of ML can be restricted to only runtime overhead.

Power-Gating: Power-gating maximizes static power savings by switching off individual NoC components. One of the critical challenges with power-gating is maintaining network connectivity when individual routers are powered off. Catnap [14] breaks the NoC into multiple sub-networks and individually powers down different sub-networks, thereby allowing one sub-network to maintain full connectivity at all times alleviating deadlock and live-lock complications. Another work seeks to leverage the amount of dark-silicon on a chip in order to create multiple NoCs that allow for the selection of the most energy efficient version that meets the performance demands [11]. Others have focused on maximizing the time that a router is switched off by re-routing around powered-off routers [27], while others seek to minimize router blocking by sending wake up signals to power-up downstream routers before packets are ready to hop across them [10]. The key goal for all of these papers is to ensure that static power savings is achieved without a significant loss in performance by meeting the break-even time requirement.

Voltage Regulator: In order to maintain low latency switching in the nanosecond range [28], [29] for NoC, each router is powered on by single low dropout linear regulator (LDO). The main drawback with using LDO is that power efficiency deteriorates drastically when the output of the LDO has large voltage variations as is the case with most DVFS designs. When an LDO is scaled down from 1.1 V to 0.8 V we see a power efficiency decrease from 92% to 67%, thereby negating the gains achieved by DVFS. To mitigate this drop in power efficiency, a switching regulator can be employed that bridges the power supply and the LDO. However, this is unfeasible for the NoC as it would increase the latency to the micro second range. In [30], a hierarchical power delivery system is reported that optimizes system performance with reinforcement learning. Multiple switching regulators form an array of LDOs where the voltage drop at each LDO is kept low enough to avoid a large drop in power efficiency. The downside to this approach is the increased area overhead caused by the addition of switching components.

DozzNoC: In this paper we propose DozzNoC, wherein we implement both power-gating (with a different approach for securing downstream routers) and DVFS using offline trained regression model *simultaneously* with low-latency and high power-efficiency SIMO voltage regulator. Each router and its' outgoing links are supplied with an LDO for lower switching latencies while the inputs to each LDO are provided by a single-inductor multi-output (SIMO) voltage regulator, thereby enabling a scalable and power-efficient design. Our approach applies an offline trained Ridge regression algorithm in order to save run-time overhead while still enabling proactive mode selection.

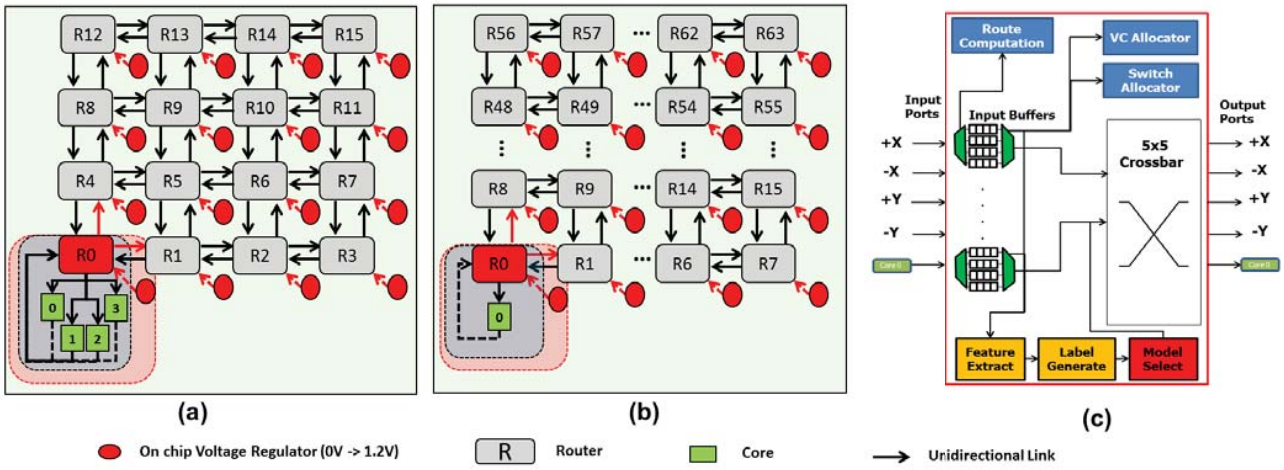

Fig. 1. **Topology:** We apply DozzNoC to both (a) concentrated mesh with 16 routers and 64 cores and (b) mesh with 64 routers and 64 cores. (c) The microarchitecture with the addition of three extra components that enable ML-based feature extraction and label generation.

III. DozzNoC Architecture

A. DozzNoC *Topology and Microachitecture*

Network Topology : DozzNoC is built with enough versatility to be applicable to multiple network topologies; we specifically apply DozzNoC to a concentrated mesh (cmesh) and a mesh network topology as shown in Figure 1(a,b). As the proposed approach does not require global coordination to select voltage level, we can scale to large number of routers and apply to different topologies. Each router and its outgoing links operate at the same frequency/voltage in DozzNoC. Varying router frequencies causes different packet latencies per network hop, and only affect the sending router (upstream) and not the receiving router (downstream). If the upstream router is faster, then the hop latency is lower and packets will traverse that router faster. If the upstream router is slower, then the hop latency is larger and packets will take longer to traverse that router. Thus, if there is a difference in router frequencies, it will simply lead to the slower routers input buffer utilization rising faster as more packets will be arriving into the router than departing the router. Moreover, our proposed SIMO voltage regulator is well suited since we can apply different voltage levels on a per-router basis to switch on/off a router and its' outgoing links with low latency and high power-efficiency (explained later). We use XY dimension order routing (DOR) to select the output ports. We also use this information to ensure that downstream routers are not allowed to be powered-off, and if they are off, to wake them up for a partially non-blocking power-gated scheme. While it would be difficult to design a partially non-blocking power-gated scheme without XY routing, it would still be possible if the downstream router can be determined in advance and woken up. Our proposed router microarchitecture is shown in Figure 1(c).

Router Microarchitecture: We implement proactive DVFS with predictive machine learning models by adding three key components to the router microarchitecture as shown in Figure 1(c). The first additional unit is called Feature Extract which gathers local and global router parameters. This data is then supplied to the next unit called Label Generate. This unit multiplies each gathered feature by its' corresponding weight and sums the results in order to generate the label. This weight vector is trained offline and is imported before the simulation begins. The last unit is called Model Select and it selects the optimal voltage mode based on the value of the predicted label. In our design, routers and links operate in any of the *three states of operation* as shown in Figure 2. These three states include an inactive state, an active state, and a wakeup state.

Inactive State: In this state, the power supply to an individual router and its' outgoing links is reduced to 0 V and the router cannot operate. While in an inactive state, the router may not send/receive packets and cannot be used to hop packets across

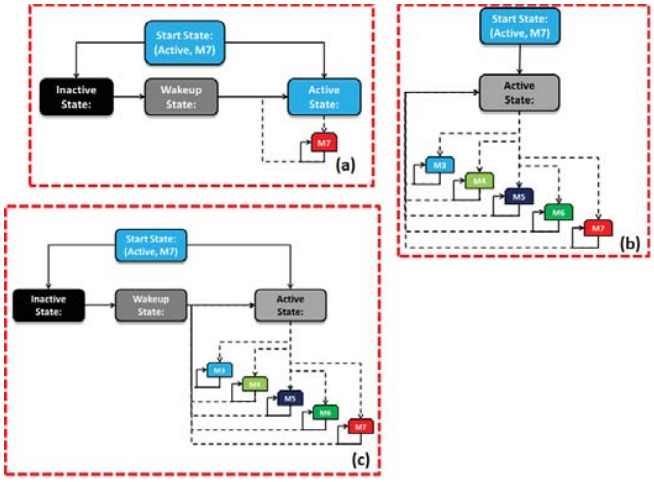

Fig. 2. **(a) Power Punch States [10], (b) LEAD-τ States [26], (c) DozzNoC States.**

it. The router can transition from an active state to an inactive state in a single cycle, but it must satisfy specific conditions before it is allowed to switch off. For this work we ensure that the routers' buffers have been empty for several consecutive cycles and that it is not a downstream router before we allow the router to be switched off.

***Wakeup State*:** A router that is in the process of transitioning from an inactive to an active state goes into a wakeup state (intermediate state). While in the wakeup state, the router consumes the same amount of power as if it were in active state. However, it may not be used to send/receive packets and it may not be used to hop packets across it until the wake-up delay has been satisfied [14]. A router can transition from an inactive state to a wakeup state in a single cycle, but the router must wait in the wakeup state for a set amount of cycles before it can be considered fully on and functional. In a power delivery system this is called the wake-up time (T-wakeup), and it is defined as the interval from the instant of a voltage change until the local voltage level settles to meet the supply voltage level. We have already accounted for the overshoot and undershoot of the power supply during this period and have determined our T-Wakeup to be 8.80 $nsec$ when using our SIMO/LDO voltage regulator design.

***Active State*:** A router that is in an active state can operate in one of five different voltage levels. These different V/F pairs are referred to as various modes of operation in which the supply voltage and clock frequency are proportionally increased/decreased. The V/F pairs our model uses in this work are $\{0.8\ V/1\ GHz$, $0.9\ V/1.5\ GHz$, $1.0\ V/1.8\ GHz$, $1.1\ V/2\ GHz$ and $1.2\ V/2.25\ GHz\}$ which correspond to being in the active state in modes 3-7. We start the numbering at mode 3 because we consider mode 1 to be the inactive state and mode 2 to be the wakeup state. These V/F pairs are similar to those used in other works [25], [26] and we have maintained the same for fair comparison. A key difference in our work is that we use real valued switching delays obtained from our SIMO voltage regulator design.

B. DozzNoC *Models*

In this subsection, we describe the various combinations of DVFS and PG models considered with and without machine learning (ML). We consider 5 models, baseline (with neither any power management nor ML implemented), PG (power-gating model with neither DVFS nor ML implemented), DVFS+ML (DVFS and ML implemented with no power-gating), DozzNoC (DVFS+PG+ML) and DozzNoC (ML-TURBO). ML+TURBO was added to see the impact on static power and dynamic energy when the highest mode is chosen instead of a lower predicted mode. All three machine learning models use the same threshold based DVFS mode selection logic. This logic looks at the current input buffer utilization and compares it to a theoretical maximum to determine what mode should be selected for the next epoch. The state transition logic for all three ML models is shown in Figure 3, where DozzNoC and ML+TURBO use the state selection logic from 3(a), and when the router is in the active state, all three comparative ML models use the logic in 3(b) to select the optimal voltage mode.

Baseline: The baseline model starts with all routers operating in the active state at the highest voltage level, mode 7. The Baseline does not allow the transition of a router into any other state. This model will always have the highest throughput and the lowest latency as it incurs no router wake-up delay and no voltage level switching delay. However the baseline offers neither static power savings nor dynamic energy savings.

Power-Gated (PG): We selected Power Punch model [10] for our power-gated design as shown in 2(a). It must be noted that the model is not an exact implementation of Power Punch, however it behaves similarly with look-ahead routing to wakeup downstream routers. This model operates routers in one of three states - inactive, waking up or active - as explained in section 3.1. If a router is active, then it will operate at the highest mode of operation, mode 7. In order for a router to transition from an active state to an inactive state, it must be idle for at least T-Idle consecutive cycles. A router is considered idle only if its' input buffers are empty and it is not a downstream router. The second condition was developed in order to make this model non-blocking in nature so that a fairer comparison to Power Punch can be made. We use XY DOR routing with a look-ahead routing algorithm which allows us to easily know the next router in a packets' path so that downstream routers can be $secured$. When a router is in a $secured$ state, it can not be switched off. If it is currently off, it will immediately transition into a wakeup state where it will stay until the wake-up delay has been met. The main purpose of this model is to compare the static power savings of a state-of-the-art power-gating technique against a design that combines power-gating and DVFS.

DozzNoC *(ML+PG+DVFS):* The proposed DozzNoC design uses the same underlying partially non-blocking power-gated design proposed earlier wherein all routers may be in one of three states as shown in Figure 2(c). The algorithm utilized in DozzNoC to decide how to transition from different state is shown in Figure 3(a). DozzNoC measures router idleness (R-Idle) every cycle. If a router has been idle for more a certain number of consecutive cycles (T-Idle) and it is not a downstream router and input buffer utilization (IBU) = 0, it will transition to the inactive state. T-Idle was based on previous work which found that T-Idle = 4 had the best performance [14]. While [14] is a multi-NoC architecture, DozzNoC is a single-NoC architecture and we use similar T-Idle value. It must be noted that a small T-idle will cause congestion since traffic will be blocked due to router being switched-off and less power savings due to T-breakeven. If T-Idle is too large, then we will not save enough power. Since our lowest voltage level has a T-wakeup of 9 cycles and T-breakeven of 8 cycles (see next subsection), our conservative estimate of T-Idle of 4 cycles will provide the correct balance. From the inactive state it will transition to the wakeup state where it must wait the full duration of the wake up delay (T-Wakeup). This delay will vary with the voltage level of the active state. When the router has been switched on it will

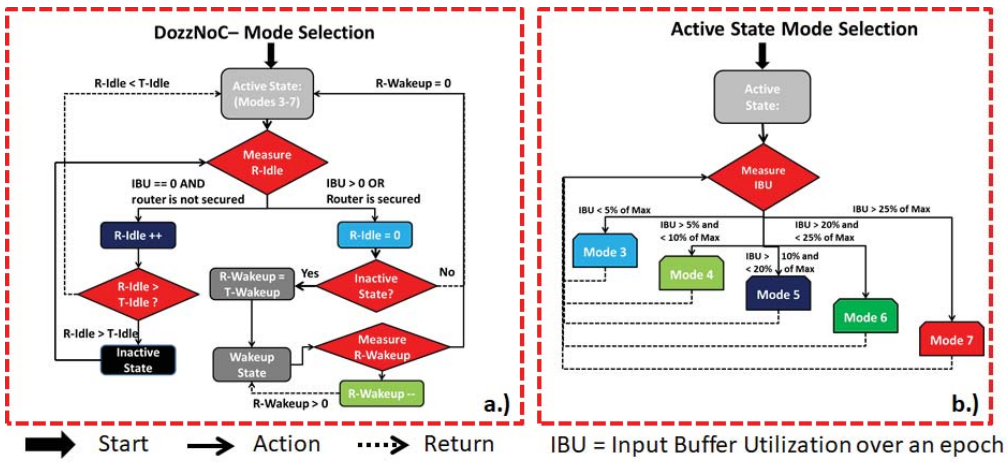

Fig. 3. (a) DozzNoC Mode Selection: Proposed DozzNoC mode selection algorithm that transitions between inactive, wake-up and active states. (b) DozzNoC Active Mode Selection: Algorithm that switches between different voltage modes under active state.

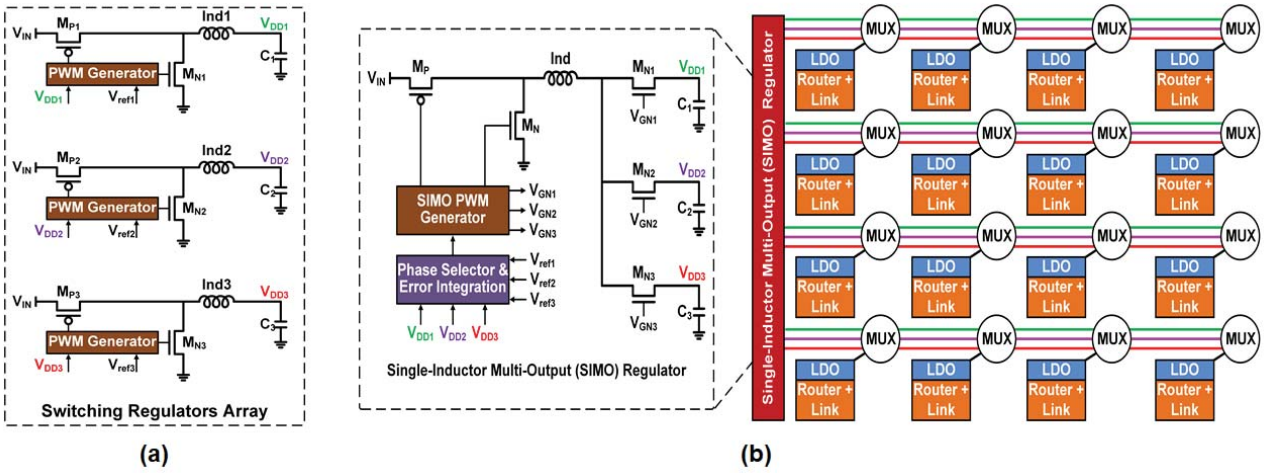

Fig. 4. (a) **LDO/Switching Regulator Array:** A conventional hierarchical power delivery system with multiple Switching Regulators and LDOs allowing for the selection of several different supply voltages. (b) **SIMO power delivery system:** Our SIMO design allows for the selection of multiple output voltages for DVFS with low switching latency and high power efficiency.

operate at one of five different voltage levels similar to the DVFS model described in [26]. This differs from the Power Punch model which may only be active in the highest mode of operation, mode 7. DozzNoC uses predictive machine learning techniques to determine the optimal voltage level for a router that is in an active state and dynamically adjusts the supply voltage to select it as shown in 3(b). In order to do this, we predict future input buffer utilization of a router and then compare this to the theoretical maximum utilization to determine the optimal voltage level that meets network performance demands while still ensuring dynamic energy savings. This DVFS design relies on aggressive voltage scaling that minimizes potential loss in throughput. For epoch size of 100 cycles, if we predict the buffers to be less than 5% of the maximum over the next epoch, we select the lowest voltage level for the active state to operate at, mode 3. If the buffers are predicted to be between 5% and 10% of the maximum we select mode 4, if the buffers are predicted to be between 10% and 20% we select mode 5, if the buffers are between 20% and 25% we select mode 6, and finally if the buffers are predicted to be more than 25% full we select mode 7. This scheme allows for switching between different voltage levels due to our proposed SIMO voltage regulator design.

(DVFS+ML): LEAD-τ [26] is used to compare against our proposed DozzNoC since LEAD-τ implements DVFS+ML in NoC architectures. In this scenario, the router can only be in an active state and use the same mode selection logic as DozzNoC where future input buffer utilization is predicted and an optimal active voltage level is calculated as shown in Figure 3(b). This model may transition from any voltage level to any other voltage level within the range of 0.8V to 1.2V. The main purpose behind including this model is to compare

how a stand-alone machine learning DVFS design performs against a machine learning design that has DVFS and power-gating.

ML+TURBO: This model seeks to apply power-gating and DVFS to the NoC in a similar fashion to DozzNoC. It uses three states of operation, the inactive state, the wakeup state, and the active state. When a router and its' links are active, a prediction of the future input buffer utilization is used to govern the voltage level. The key difference between this model and DozzNoC is that every three times we predict that a router should be at any active mode other than mode 3 or mode 7, we instead select the highest voltage level for the next epoch. The key goal of this model is to improve throughput at the cost of dynamic energy since we opt for the highest mode even if we predict a lower mode to be more optimal in the hopes of saving more static power.

C. SIMO/LDO Voltage Regulator

Prior work on designing hierarchical power delivery system to optimize system performance and latency has been reported [30] and is shown in Figure 4(a). However, the downside is extra power to switch components. Our proposed DozzNoC is built upon a unique SIMO/LDO voltage regulator design shown in Figure 4(b). Each router and its' outgoing links are supplied with an LDO for lower switching latencies while the inputs to each LDO are provided by a single-inductor multi-output (SIMO) voltage regulator [31]. It is critical that we use SIMO regulators to enable variable supply voltages because without them the input voltage is a fixed battery voltage. Our DVFS models can use this SIMO/LDO design to select different operating voltages within the $0.8V$ to $1.2V$ range. The input voltage of the LDO dynamically selects the MUX for different voltage levels of $0.9V$, $1.1V$, and $1.2V$. This design also allows for power-gating when both the input and output of the LDO are switched to ground. This allows us to design power-gated models that can save static power. Another advantage to our SIMO regulator scheme is that there is very low area overhead cost compared to conventional power delivery systems such as the switching regulator/LDO array. There is a single inductor that can provide three different output voltages simultaneously. To regulate the three voltages to the desired values respectively, the SIMO regulator adopts time-multiplexing control scheme. Our SIMO design reduces the number of power switches from 6 to 5 which leads to an overall decrease in on-chip and off-chip components for reduced area overhead. We show in Figure 6 that the overall power efficiency of the proposed system is higher than 87%. Compared to the baseline where the LDO is supplied with 1.2V, our design achieves an average power efficiency improvement of 15% at four various points of comparison with a maximum efficiency increase of almost 25% at 0.9V.

In Table I we show how the voltage dropout of the LDO can be made equivalent to a 100 mV drop leading to much higher power-efficiency than similar designs that would need much higher dropouts in order to be able to provide voltages in the 0.8V to 1.2V range. This is because the SIMO regulator supplies three V_{dd}'s at the same time. The change in latency and output voltage of the SIMO regulator are small enough that they can be ignored. Thus the overall latency is determined only by the LDO. In Figure 5 we show the waveforms from power-gating a router from 0V to 0.8V as well as switching from 0.8V to 1.2V. V_{O-LDO}, I_{LD} and V_{IN-LDO} represent the output of the LDO, the equivalent load current, and the input of the LDO. The input of the LDO changes with the output such that the maximum dropout remains between 0 and 100 mV. LDOs have high bandwidth, thus the latencies are still within the $nsec$ range. The real valued latency to perform power-gating and DVFS from any voltage within the range of 0.8V to 1.2V is listed in Table II. These costs need to be converted to cycles so that they can be simulated in our cycle accurate network simulator. The cycle cost of these real valued delays are shown in Table III. We apply the worst case power-gating/voltage level switching latency to every case. For instance, the worst case power-gating delay (T-Wakeup) is 8.8ns, thus we apply T-Wakeup cost to every router that wants to switch from 0V to any voltage level in the range of 0.8V to 1.2V (inactive state to active state). The worst case voltage switching delay (T-Switch) is 6.9ns, thus we apply that switching cost to every router that wants to switch from any active mode to any other active mode. The break-even time (T-Breakeven) is applied according to the mode that a router wants to switch on into. According to other work, the value of T-Breakeven is around 10 cycles [13]. We conservatively estimate our T-Breakeven to be 12 cycles for the highest mode and proportionally less for lower modes.

TABLE I
LDO Voltage dropout range for three dynamically selected input voltages.

LDO Vin	LDO Vout Range	Dropout Range
0.9V	0.8V - 0.9V	0V - 0.1V
1.1V	1.0V - 1.1V	0V - 0.1V
1.2V	1.2V	0V

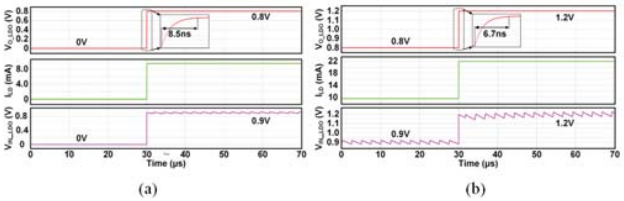

Fig. 5. **Real-Valued Delay: (a) T-Wakeup:** The real-valued wake-up delay for a router to transition from an inactive state to an active state during power-gating where the switching starts at 30 μsec. **(b) T-Switch:** The real-valued voltage switching delay for a router to switch between voltage levels when using DVFS.

D. Machine Learning-based Mode Selection

Machine Learning enables us to use proactive mode selection techniques for all three ML models (DozzNoC, DVFS+ML and ML-TURBO). Our feature set corresponds to relevant network throughput parameters such as buffer utilization, link utilization, or router idle time while our

TABLE II
MEASURED DELAY TO SWITCH BETWEEN ANY MODE IN THE VOLTAGE RANGE OF 0.8V - 1.2V.

Latency	PG	0.8V	0.9V	1.0V	1.1V	1.2V
PG	0ns	8.5ns	8.7ns	8.7ns	8.7ns	8.8ns
0.8V	8.5ns	0ns	4.2ns	5.5ns	6.2ns	6.7ns
0.9V	8.7ns	4.2ns	0ns	4.4ns	5.5ns	6.3ns
1.0V	8.7ns	5.5ns	4.4ns	0ns	4.3ns	5.5ns
1.1V	8.7ns	6.3ns	5.4ns	4.3ns	0ns	4.3s
1.2V	8.8ns	6.9ns	6.3ns	5.4ns	4.1ns	0ns

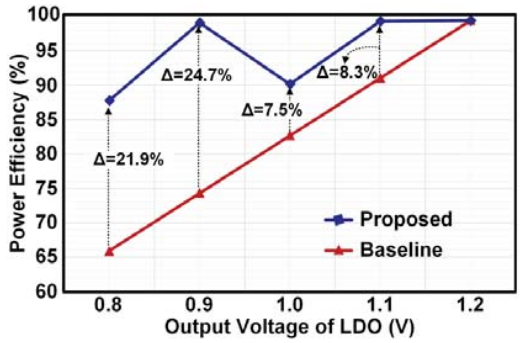

Fig. 6. **Power Efficiency:**The power efficiency of our SIMO design versus a baseline regulator switching array.

weight vector corresponds to the impact that each feature has in determining the overall label. We use Ridge Regression and perform supervised learning using the following equation:

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M} w_j^2$$

The core of the Ridge Regression equation is the minimization of the sum of square errors. This means that the error between the actual value of the label and the predicted value of the label will be made as small as possible during the training phase. Our training phase takes place outside of our network simulator as it is done offline in Matlab. The predicted label ($y(x_n, w)$) is the routers' predicted future input buffer utilization, and this is minimized with respect to the actual label (t_n). The routers' actual future input buffer utilization is supplied during training along with the features. We tune the equation $\frac{\lambda}{2}\sum_{j=1}^{M} w_j^2$ with different lambda hyper parameter values until the best-fitting solution is found. This is exported in the form of a weight array and used by the network simulator during testing. We used 14 trace files in total - 6 trace files for testing, 3 for validation, and the remaining 5 for testing the generalized performance of each trained model.

Feature Set: The feature set is carefully crafted such that prediction accuracy is maximized while overhead is kept to a minimum. This is accomplished by selecting local router features that gives the greatest insight into network performance while minimizing features that may require global coordination or communication. Each additional feature equates to more computational overhead because the number of additions and multiplications necessary to generate the label increases. The original feature set proposed in prior work [26] contained 41 features in total as well as a label, however we have reduced this to only five critical features. These five features are listed in detail in Table IV and this will be further discussed in results section.

TABLE III
MEASURED DELAY COSTS FOR T-WAKEUP, T-SWITCH, T-BREAKEVEN.

Volt.	Freq.	T-Switch	T-Wake up	T-Break even
0.8V	1 Ghz	7 cycles	9 cycles	8 cycles
0.9V	1.5 Ghz	11 cycles	12 cycles	9 cycles
1.0V	1.8 Ghz	13 cycles	15 cycles	10 cycles
1.1V	2 Ghz	14 cycles	16 cycles	11 cycles
1.2V	2.25 Ghz	16 cycles	18 cycles	12 cycles

TABLE IV
REDUCED FEATURE SET USING ONLY LOCAL ROUTER FEATURES.

Feature Set:	
Feature 1:	Array of 1's
Feature 2:	Requests Sent by 4 Cores Connected to Router
Feature 3:	Requests Received by 4 Cores Connected to Router
Feature 4:	Router Total Off Time
Feature 5:	Current Input Buffer Utilization
Label:	Future Input Buffer Utilization

Label: In order to generate the training features and their corresponding labels, we must first design reactive versions of each machine learning model that uses current or past network parameters to govern mode selection. We run the training traces with these reactive mode selection models and export the features as well as the label every epoch. The label that all models are supplied with is the future input buffer utilization of the router. This value is tacked onto the feature set at the end of the simulation since it is not actually known until the next epoch. This data must be collected separately across all training/validation benchmarks for each of the various models such that each model will use unique training/validation data. Once the models have been trained, they are exported back to the network simulator where they are used to generate labels that allow proactive mode selection, thus each ML model is trained offline and is ready to use at test time.

Machine Learning Overhead: After a model has been trained, the weight vector is exported to the network simulator where it can be used to select voltage levels when routers are active. The additional overhead incurred from machine learning can be broken down into the timing, area, and energy cost to execute a series of additions and multiplications as this is how a label is calculated. Each feature is multiplied by its equivalent weight and then the results are summed in order to generate a label. Prior work [32] has already estimated the cost to do these operations. The energy cost of a single 16 bit floating point add is 0.4 pJ and the area cost is 1360 um^2. To execute a multiply would consume an estimated 1.1 pJ with an area overhead of 1640 um^2. Prior work that used 41 features calculated the total energy overhead cost to be 61.1pJ, the total area overhead cost to be 0.122 mm^2, and

the total timing cost to be 3-4 cycles. We have shown that the feature set can be reduced down to 5 features without causing a significant impact on model performance. Therefore the overhead to generate a label can be reduced to only 5 multiplications and 4 additions. This equates to a total energy overhead cost of 7.1pJ, a total area overhead cost of 0.013 mm^2, and a total timing cost of 3-4 cycles per router. Our epoch size is 500 cycles and a label only needs to be calculated by a router once per epoch.

IV. Performance Evaluation

A. Simulation Setup

We use a cycle accurate full system simulator to gather trace files from real industry standard benchmarks[33]. This allows us to run both PARSEC 2.1 [34] and SPLASH2 [35] benchmarks in order to generate trace files which contain per core network traffic. When a packet is injected into the network, the source, destination, type (request/response) and injection time are all saved as a single entry. These traces are then supplied to our network simulator and used as input for real traffic patterns in order to gather training and validation data for our various models. We developed reactive versions of each machine learning model (DozzNoC, LEAD-τ, and ML+TURBO) which rely on current buffer utilization to select voltage levels while the router is in an active state. This allows us to run our network simulator and export features and a label every epoch. This data is used to train our various mode selection models using supervised learning with Ridge Regression.

From a total of 14 trace files, we use a total of six trace files for training purposes, three for validation, and then the final five for testing. During validation the lambda hyper parameter is tuned until the best-fitting solution is found, meaning the array of weights that produced the smallest error between the predicted label and the supplied label. After training and validation we test the trained algorithm by exporting the trained weights for use in our network simulator where they are used to generate labels (future input buffer utilization). This future input buffer utilization is then used to govern mode selection allowing proactive models based on accurate predictions of future network parameters. This is repeated for all three ML models, DozzNoC, LEAD-τ, and ML+TURBO. The test traces are not used for training or validation ensuring that the performance of each model can be measured as accurately as possible. Dsent [36] is used to model the router and the links as well as to obtain their respective static power/dynamic energy costs. The static power cost as well as the dynamic energy cost of the router and it's outgoing links for a concentrated mesh are shown in Table V. The latency and power/energy costs of a concentrated mesh are higher than a mesh because they have more components and larger crossbars, thus they are used as a worst case for any latency/power/energy costs. These delays were gathered for the five different modes of operation at a technology size of 22nm with 128-bit flit width [36].

B. Results

The results section will be divided up into two subsections. The first section will discuss trade-off studies such as comparing the mode selection accuracy of multiple individual features as well as mode prediction breakdown of each ML model. The second section will discuss throughput and dynamic and static energy savings for compressed and uncompressed traffic traces.

1) Trade-Off Studies: In Figure 7, we show the distribution of predicted DVFS modes for all three ML models. This means that when a DozzNoC router is in the active state, it will operate at M3-M7 proportionally. These active state voltage levels are updated every epoch but a router may transition between active, waking up, and inactive at any point within an epoch. Both LEAD-τ and ML+TURBO do not apply power-gating, thus these routers will always be active in the mode that was determined optimal for that epoch according to the generated label. The baseline and the Power Punch scheme are not shown as they do not use DVFS logic to select optimal active modes. From the results, we observe that the mode switching for DVFS with ML models appear to be similar.

TABLE V
Static power and dynamic energy cost to hop across the router and a link at 22nm technology [36].

Volt.	Freq.	Static Power (J/s)	Static Power (Cycle)	Dynamic Energy (pJ/hop)
0.8V	1 Ghz	.036	.667	25.1
0.9V	1.5 Ghz	.041	.750	31.8
1.0V	1.8 Ghz	.045	.833	39.2
1.1V	2 Ghz	.050	.917	47.5
1.2V	2.25 Ghz	.054	1.0	56.5

In Figure 11, we show how we determined which features had the best correlation to accurate predictions of future input buffer utilization by comparing mode selection accuracies. Mode selection accuracy is defined as the total number of accurate mode selections divided by all accurate and inaccurate mode selections. We record the labels every epoch and compare them to the real value of the buffer utilization at the next epoch. As long as both would lead to the same mode being selected, the selection was considered to be accurate.For our trade-off study in Fig. 11 we trained and validated our DozzNoC model using only a single feature plus an array of 1's for normalization. Each weight is trained/validated/tested individually so we can analyze the mode selection accuracy using that particular feature. This will help us weed out features that do not predict the label (future input buffer utilization). Weights are trained such that when a weight is multiplied with its' respective feature it generates a label. This predicted label is subtracted from a supplied label (future buffer utilization) and the error between the two is brought as close to zero as possible. Once each single feature weight has been trained it is exported for use in our network simulator where it can be used at runtime to generate labels that are subsequently used to govern mode selection.

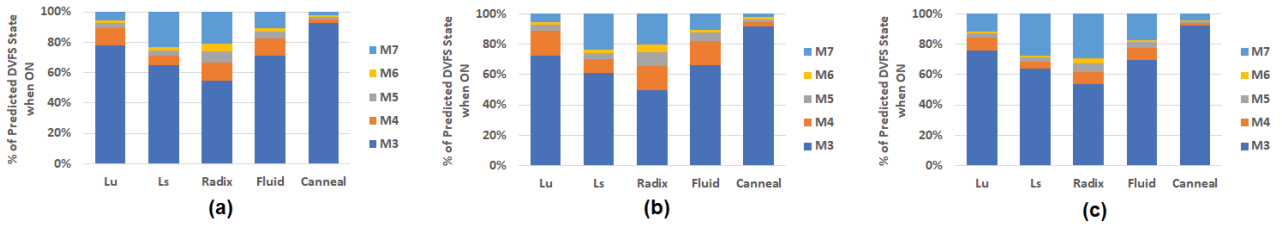

Fig. 7. Breakdown of different DVFS modes predicted for various benchmarks for 8 × 8 uncompressed for window size of 500 (a) DozzNoC, (b) LEAD-τ and (c) ML+TURBO.

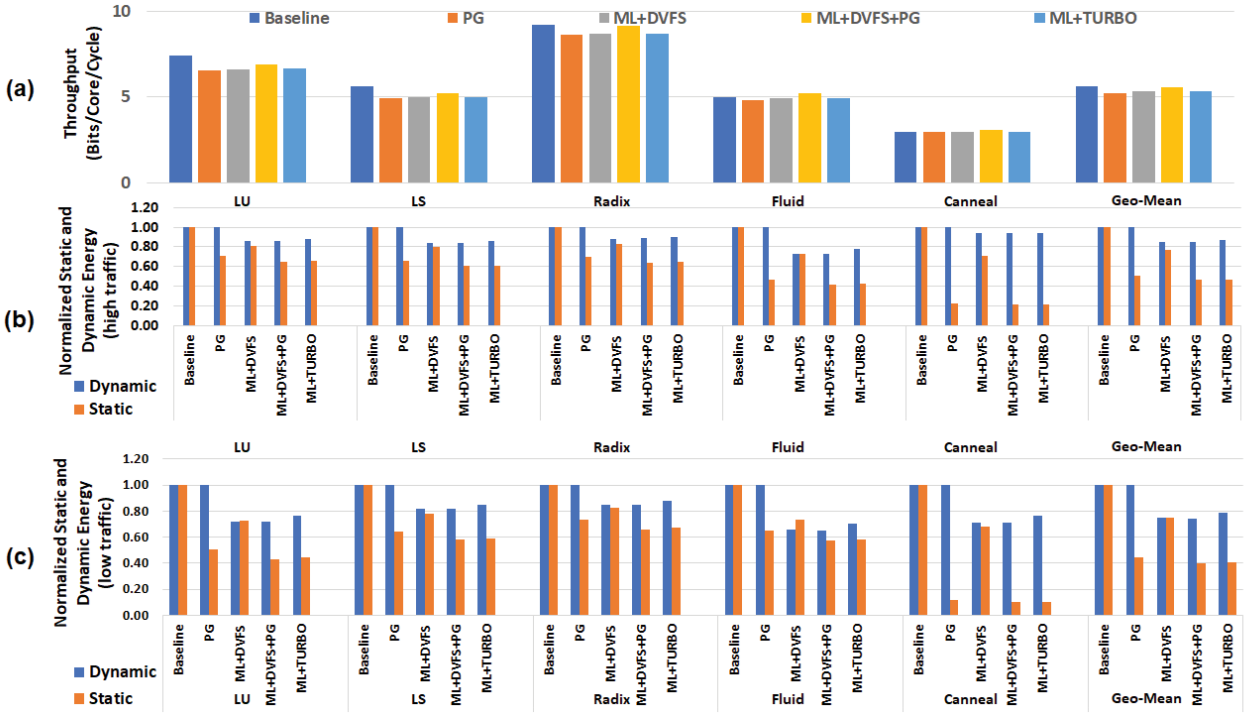

Fig. 8. (a) shows the throughput for compressed MESH architecture for Baseline, Power Punch (PG), LEAD-tau (ML+DVFS), DozzNoC (ML+DVFS+PG), and ML in TURBO mode (ML+TURBO) for a window size of 500 cycles. The static and dynamic energy normalized to the baseline for 8 × 8 MESH for a window size of 500 with (b) compressed traces, and (c) uncompressed traces is shown.

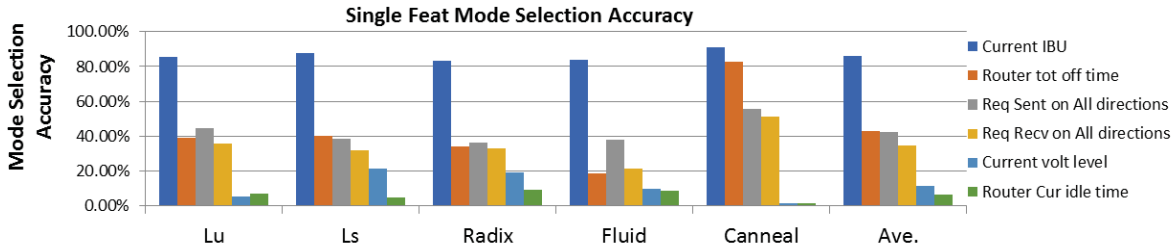

Fig. 9. Shows the mode selection accuracy of using only a single feature for DozzNoC model training and testing. The mode selection accuracy across all 5 test traces is shown with the average value given above each benchmark for each individual feature.

From the results, we observe that input buffer utilization has the most impact on mode selection accuracy (80%). Then the total router off time and traffic in all directions provides accurate mode selection 40% of the time. Using only the top 5 features, we observed that there is almost no impact on throughput, latency, dynamic energy savings, static power savings, or EDP between our DozzNoC model that was trained and validated with 41 features (DozzNoC-41) and

a model that used only the top 5 best performing features from our feature test in 11 as well as an array of 1's for normalization (DozzNoC-5). The main reason is that the current input buffer utilization predicts almost 80% in mode selection accuracy and the remaining top 4 features provide 40% in mode selection accuracy.

Therefore, by combining the top 5 features, we obtain no loss in performance. Moreover, we also tested across multiple epoch sizes (100 - 1000) and determined that 500 allowed us to maintain increased model performance while still allowing us to maintain a healthy amount of training and validation data. Since the predictive model has an impact on the future buffer utilization, we specifically train/test/validate our model on different epoch sizes so that the offline-sampled labels learn the model by considering the inter-epoch dependencies. Therefore, each epoch size has a separately trained model which retains all inter-epoch dependencies when frequency/voltage are changed.

2) Model Performance: For traditional DVFS designs the main focus is the trade-off between performance loss and dynamic energy savings, while traditional power-gated designs focus on the trade-off between performance loss and static power savings. For our work, our final results must focus on the trade-off between performance loss and both dynamic energy savings and static power savings. This is why we compare a baseline model that has neither power-gating nor DVFS implemented against four other models in order to show case the numerous trade-offs we seek to highlight. The baseline model is always active and always operates routers and links at the highest voltage level while the power-gated design operates routers and links at mode 7 when turned on.

For a mesh topology at an epoch size of 500 cycles our version of power-gated design can achieve an average of 47% static power savings for an increase of 5% in latency and a throughput loss of 9%. For a mesh topology at an epoch size of 500 cycles LEAD-tau model can achieve an average of 25% dynamic energy savings and 25% static power savings for a 1% latency increase and a 3% loss in throughput. We note that static power savings are obtainable while only using DVFS because lower voltage levels will consume proportionally less static power than the baseline which always operates at the highest voltage level. Our DozzNoC model highlights our novel design which seeks to save both static power and dynamic energy. For a mesh topology with an epoch size of 500 cycles, our DozzNoC model can save on average 53% static power and 25% dynamic energy while only increasing latency by 3% and decreasing throughput by 7%. For a cmesh network DozzNoC can save on average 39% static power and 18% dynamic energy for a latency increase of 2% and a throughput loss of 5%. Our ML+TURBO model is an experimental model designed to show the trade-off between dynamic energy savings and static power savings. Every three epochs that our ML+TURBO model determined a router should operate in a mode other than the lowest or highest mode, we instead forced that router to operate in the highest mode with the goal of losing some dynamic energy savings to see if we could obtain a greater increase in static power savings through faster simulations. For a mesh topology with an epoch size of 500 cycles, ML+TURBO saved on average 52% static power and 21% dynamic energy for a latency increase of 3% and a throughput loss of 7%. When compared to our DozzNoC model we note that not only did we have a slight loss in static power savings, but we also had a slight loss in dynamic energy savings. This is because the highest mode of operation consumes the most dynamic energy and it has the highest static power cost. Also, just because we operate in the highest mode does not necessarily mean that the simulation will end sooner because packet injection is based on real valued cycle times.

V. Conclusions

This paper discusses techniques to save both static power and dynamic energy by combining partially non-blocking power-gating and smart proactive DVFS. The power-gated portion of the design can be operated on a fine-grain timescale to ensure that break-even times, wakeup times, and idle counters are accounted for while the DVFS portion can be operated on a coarse-grain timescale to ensure switching delays can be minimized. The LEAD-τ model as well as the power-gated model were used for comparative purposes and highlighted the individual trade-offs associated with using either a modern partially non-blocking power-gated scheme or a smart proactive mode selection model for DVFS. Our novel DozzNoC model showed how we can combine the underlying ideas behind these two key models in order to save both dynamic energy and static power for minimal loss in performance with only 5 critical features for reduced computational run-time overhead. We also show that there are several key benefits of using LDO's to reduce voltage switching and wakeup up delays.

VI. Acknowledgement

This research was partially supported by NSF grants CCF-1513606, CCF-1703013, CCF-1901192, CCF-1547034, CCF-1547035, CCF-1547036, CCF-1702980 and CCF-1702496. We sincerely thank the anonymous reviewers for their excellent feedback.

References

[1] A. K. Mishra, R. Das, S. Eachempati, R. Iyer, N. Vijaykrishnan, and C. R. Das, "A case for dynamic frequency tuning in on-chip networks," in *Proceedings of the International Symposium on Microarchitecture (MICRO)*, 2009, pp. 392–303.

[2] R. David, P. Bogdan, and R. Marculescu, "Dynamic power management for multicores: Case study using the intel scc," in *Internationa Conference on VLSI and System-on-Chip (VLSI-SoC)*, October 2012, pp. 147–152.

[3] P. Bogdan, R. Marculescu, S. Jain, and R. Gavila, "An optimal control approach to power management for multi-voltage and frequency islands multiprocessor platforms under highly variable workloads," in *International Symposium on Networks on Chip (NoCS)*, May 2012, pp. 35–42.

[4] L. Shang, L. Peh, and N. Jha, "Power-efficient interconnection networks: Dynamic voltage scaling with links," in *Computer Architecture Letters, 1(1)*, January 2002.

[5] R. Jain, P. Panda, and S. Subramoney, "Machine learned machines: Adaptive co-optimization of caches, cores, and on-chip network," in *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, April 2016.

[6] G. Dhiman and T. Rosing, "Dynamic voltage frequecy scaling for multi-tasking systems using online learning," in *ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, August 2007.

[7] R. Hay, "Machine learning based dvfs for energy efficient execution of multithreaded workloads," in *Dissertations and Theses Technical Reports-Computer Science*, November 2014.

[8] X. Chen, Z. Xu, H. Kim, P. Gratz, J. Hu, M. Kishinevsky, U. Ogras, and R. Ayoub, "Dynamic voltage and frequency scaling for shared resources in multicore processor designs," in *50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, July 2013.

[9] H. Shen, J. Lu, and Q. Qiu, "Learning based dvfs for simultaneous temperature, performance and energy management," in *13th International Symposium on Quality Electronic Design (ISQED)*, March 2012.

[10] L. Chen, D. Zhu, M. Pedram, and T. Pinkston, "Power punch: Towards non-blocking power-gating of noc routers," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, July 2015, pp. 378–389.

[11] H. Bokhari, H. Javaid, M. Shafique, J. Henkel, and S. Parameswaran, "darknoc: Designing energy-efficient network-on-chip with multi-vt cells for dark silicon," in *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC) (DAC-51)*, June 2014, pp. 1–6.

[12] L. Chen, L. Zhao, R. Wang, and T. Pinkston, "Mp3: Minimizing performance penalty for power-gating of clos network-on-chip," in *2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*, February 2014, pp. 296–307.

[13] L. Chen and T. Pinkston, "Nord: Node-router decoupling for effective power-gating of on-chip routers," in *2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, December 2012, pp. 270–281.

[14] R. Das, S. Narayanasamy, S. Satpathy, and R. Dreslinski, "Catnap: Energy proportional multiple network-on-chip," in *ISCA '13 Proceedings of the 40th Annual International Symposium on Computer Architecture*, June 2013, pp. 320–331.

[15] A. Samih, R. Wang, A. Krishna, C. Maciocco, C. Tai, and Y. Solihin, "Energy-efficient interconnnect via router parking," in *2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, February 2013, pp. 508–519.

[16] M. Casu, M. Yadav, and M. Zamboni, "Power-gating technique for network-on-chip buffers," in *Electronics Letters*, November 2013, pp. 1438–1440.

[17] H. Farrokhbakht, M. Taram, B. Khaleghi, and S. Hessabi, "Toot: an efficient and scalable power-gating method for noc routers," in *2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, August 2016, pp. 1–8.

[18] N. Nasirian, R. Soosahabi, and M. Bayoumi, "Traffic-aware power-gating scheme for network-on-chip routers," in *2016 IEEE Dallas Circuits and Systems Conference (DCAS)*, October 2016, pp. 1–4.

[19] K. Hale, B. Grot, and S. Keckler, "Segment gating for static energy reduction in networks-on-chip," in *2009 2nd International Workshop on Network on Chip Architectures*, December 2009, pp. 57–62.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," in *Journal of Machine Learning Research 15*, June 2014, pp. 1929–1958.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[23] S. Herbert and D. Marculescu, "Analysis of dynamic voltage/frequency scaling in chip-multiprocessors," in *Proceedings of the 2007 international symposium on Low power electronics and design (ISLPED)*, August 2007.

[24] S. Eyerman and L. Eeckhout, "Fine-grained dvfs using on-chip regulators," in *ACM Transactions on Architecture and Code Optimization (TACO)*, April 2011.

[25] R. Hesse and N. Jerger, "Improving dvfs in nocs with coherence prediction," in *NOCS '15 Proceedings of the 9th International Symposium on Networks-on-Chip*, September 2015.

[26] M. Clark, A. Kodi, R. Bunescu, and A. Louri, "Lead: Learning-enabled energy-aware dynamic voltage/frequency scaling in nocs," in *The 55th Annual Design Automation Conference (DAC)*, June 2018.

[27] R. Parikh, R. Das, and V. Bertacco, "Power-aware nocs through routing and topology reconfiguration," in *DAC '14 Proceedings of the 51st Annual Design Automation Conference*, June 2014, pp. 1–6.

[28] I. Vaisband and E. Friedman, "Dynamic power management with power network-on-chip," in *2014 IEEE 12th International New Circuits and Systems Conference (NEWCAS)*, October 2014.

[29] M. Manda, S. Pakala, and P. Furth, "A multi-loop low-dropout fvf voltage regulator with enhanced load regulation," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, August 2017.

[30] Y. Bai, V. Lee, and E. Ipek, "Voltage regulator efficiency aware power management," in *Proceedings of the 22nd International Conference on Architectural Support for Programming Languages and Operating System (ASPLOS)*, April 2017, pp. 825–838.

[31] Dongsheng Ma, Wing-Hung Ki, Chi-Ying Tsui, and P. K. T. Mok, "Single-inductor multiple-output switching converters with time-multiplexing control in discontinuous conduction mode," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 1, pp. 89–100, Jan 2003.

[32] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, February 2014, pp. 10–14.

[33] R. Ubal, B. Jang, P. Mistry, D. Schaa, and D. Kaeli, "Multi2sim: A simulation framework for cpu-gpu computing," in *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '12, 2012, pp. 335–344.

[34] C. Bienia and K. Li, " PARSEC 2.0: A New Benchmark Suite for Chip-Multiprocessors ," in *Proc. of the 5th Annual Workshop on Modeling, Benchmarking and Simulation*, June 2009.

[35] S. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 Programs: Characterization and Methodological Considerations," in *Proc. of the 22nd International Symposium on Computer Architecture*, June 1995.

[36] C. Sun, C.-H. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L.-S. Peh, and V. Stojanovic, "Dsent - a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling," in *Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium on*, 2012, pp. 201–210.