darkNoC: Designing Energy-Efficient Network-on-Chip with Multi-Vt Cells for Dark Silicon

Haseeb Bokhari[†] Haris Javaid[†*] Muhammad Shafique[‡] Jörg Henkel[‡]
Sri Parameswaran[†]

[†]School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

[‡]Chair for Embedded Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany

{hbokhari, harisj, sridevan}@cse.unsw.edu.au, {muhammad.shafique, henkel}@kit.edu

ABSTRACT

In this paper, we propose a novel NoC architecture, called darkNoC, where multiple layers of architecturally identical, but physically different routers are integrated, leveraging the extra transistors available due to dark silicon . Each layer is separately optimized for a particular voltage-frequency range by the adroit use of multi-Vt circuit optimization. At a given time, only one of the network layers is illuminated while all the other network layers are dark. We provide architectural support for seamless integration of multiple network layers, and a fast inter-layer switching mechanism without dropping in-network packets. Our experiments on a 4×4 mesh with multi-programmed real application workloads show that darkNoC improves energy-delay product by up to 56% compared to a traditional single layer NoC with state-of-the-art DVFS. This illustrates darkNoC can be used as an energy-efficient communication fabric in future dark silicon chips.

1. INTRODUCTION

Network-on-Chips (NoCs) provide a scalable communication fabric for connecting large number of resources within a chip. NoC can contribute significantly to the total chip power, e.g., up to 18% and 33% in Intel SCC [1] and RAW [2] architectures, respectively. To address this issue, various power saving techniques for NoC at system-, architecture- and circuit-level have emerged, among which Dynamic Voltage and Frequency Scaling (DVFS) is very prominent [3, 4]. System-level DVFS managers apply reduced voltage and frequency (VF) levels in a NoC to save power. According to several recent studies [5, 6], the energy saving potential of DVFS has been diminishing due to a number of reasons:

- With shrinking node size, the difference between nominal voltage and the threshold voltage is decreasing. As the nominal V approaches threshold voltage, the transistor delay increases exponentially, resulting in huge performance penalties [5]. This limits the factor by which V can be scaled down. Transistors with lower threshold voltages can be used in the circuit to increase the gap between the nominal and threshold voltages, but that results in an increased leakage power.
- DVFS does not influence the intrinsic physical properties of the circuit such as gate sizes, capacitances, etc. which are fixed at design time.

[*]The author is now affiliated with Google Inc.

DAC 2014, June 01 - 05 2014, San Francisco, CA, USA

http://dx.doi.org/10.1145/2593069.2593117

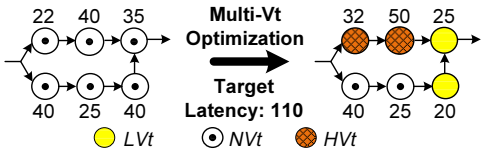

Figure 1: An example of multi-Vt circuit optimization.

With the help of the following NoC synthesis case study, we show how the problem of diminishing returns of DVFS can be alleviated.

Motivational Example: Most fabrication foundries characterize cell libraries for various gate threshold voltage (Vt) values such as normal Vt (NVt), Low Vt (LVt), and High Vt (HVt). LVt cells can switch at a much faster speed than HVt cells. However, LVt cells can be up to $5\times$ leakier than their HVt counterparts. Modern CAD tools exploit the power-delay characteristics of multi-Vt cell libraries and slacks in path delays to synthesize power efficient circuits [7]. A typical CAD tool optimizes the circuit by using LVt cells on critical paths to meet the target latency, while inserting HVt cells on the non-critical paths to reduce leakage power. For example, in Figure 1, a circle and its annotation represents the type of cell and its delay. With multi-Vt circuit optimization, a mix of LVt, NVt and HVt cells is used to meet the target latency rather than just the NVt cells. Thus, the circuit on the right will have different intrinsic properties such as gate sizes, capacitances and leakage power than the circuit on the left.

We exploited the multi-Vt circuit optimization available in CAD tools to synthesize architecturally identical NoC routers for a set of target VF levels: [1GHz, 0.9V], [750 MHz, 0.81V], [500 MHz, 0.81V] and [250 MHz, 0.72V]. Figure 2 reports the network power for operation at [500 MHz, 0.81V] and [250 MHz, 0.72V] (details of our experiments are presented in Section 4). We can observe that for operation at [500 MHz, 0.81V], the NoC designed particularly for [500 MHz, 0.81V] VF level is on average 35% and 16% more power efficient than applying DVFS on a NoC designed for [1GHz, 0.9V] and [750 MHz, 0.81V], respectively. Similarly, for operation at [250 MHz, 0.72V], the NoC designed particularly for [250 MHz, 0.72V] VF level is on average 42% and 22.7% more power efficient

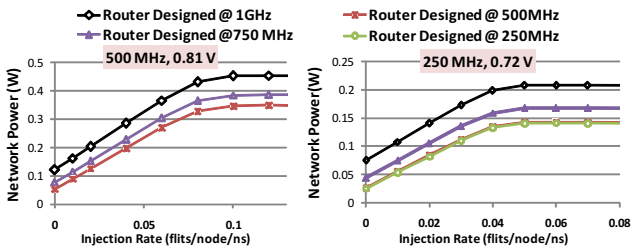

Figure 2: NoC Power for Transpose Traffic for a)(left)[500 MHz, 0.81V] VF level, b)(right) [250 MHz, 0.72V] VF level

than DVFS applied to NoC designed for [1GHz, 0.9V] and [750 MHz, 0.81V], respectively. This observation shows that, unlike traditional NoC with a single layer of routers, it may be beneficial in terms of power to have multiple layers of routers in a NoC such that each layer is optimized for a particular VF level.

The provision of multiple layers of routers in a NoC may cost a significant amount of silicon; however, we can leverage *dark silicon* [8][9] to alleviate this overhead. Several researchers have proposed to leverage dark silicon to provide extra application-specific accelerators [10], extra low-power processors [11], etc. for better energy efficiency. In fact, the authors of [12] state that *"in eight years, we will be faced with designs that are 93.75% dark!"*, and, *"we will spend exponentially increasing amounts of silicon area to buy energy efficiency"*. Therefore, even with the addition of extra accelerators [10] and processors [11], *we believe that there will still be dark silicon available which could be exploited for energy-efficient design of other system components such as NoC.* Given the availability of dark silicon, three research problems need to be addressed for realization of a NoC with multiple network layers:

- What kind of architectural support is required to integrate multiple VF-optimized network layers in a NoC?
- How many network layers (and their VF levels) should be used for a set of applications under a dark silicon budget?
- How to enhance system-level DVFS managers to exploit the provision of multiple VF-optimized network layers?

In this paper, *we focus on the first research problem. We propose a novel NoC architecture, where architecturally homogenous routers that have been optimized specifically for particular VF ranges, are seamlessly integrated at the architecture-level to complement system-level DVFS managers, and exchange dark silicon for energy efficiency in NoC.* For the rest of the two problems, we provide our initial insights from extensive experiments, and leave their comprehensive solutions as future work. In particular, **our key contributions are:**

- We propose a novel NoC architecture, named *darkNoC*, where multiple network layers consisting of architecturally identical routers, but optimized to operate within different voltage and frequency ranges during synthesis are used. Only one network layer is illuminated (active) at a given time while the rest of the network layers are dark (deactivated).
- We propose an efficient hardware-based mechanism to transition from one network layer to another. Our network layer switch-over mechanism preserves the lossless communication property of a packet-switched buffered NoC, and is transparent to the software. Further, we investigate and report the factors that can potentially affect the time and energy overhead of switch-over mechanism.
- Finally, to illustrate the potential of our darkNoC architecture, we show how a state-of-the-art DVFS manager [13] is deployed. Further, we compare our darkNoC architecture having different number of network layers (due to dark silicon budget) with a traditional single network layer NoC, where both the architectures have the same DVFS manager (our results show savings of up to 56% in energy-delay product).

2. RELATED WORK

Various system-level DVFS techniques have been proposed for NoC power management to strike a balance between performance and power, such as at the granularity of communication links [14], routers [3, 4] and regions [13]. Run-time power gating has also been proposed to reduce leakage power of NoCs. Matsutani et. al. [15] proposed an ultra fine-grained power gating technique for routers. Similar techniques at other granularities have been explored in [16, 17, 18]. All of these techniques are orthogonal to our work and can be tweaked to exploit the provision of multiple network layers in a NoC. In this paper, we use the DVFS technique of [13] as the state-of-the-art for comparing our darkNoC architecture with the traditional single network layer NoC.

Pullini et.al. [19] investigated the advantages of using multi-Vt circuit optimization in terms of performance and power for a single layer NoC architecture. In contrast, we exploit the multi-Vt circuit optimization in the context of multiple network layers and DVFS. Various NoCs with multiple network layers have been proposed in [20, 21, 22]. However, all of these works customized the architecture of the network layers or the routers, in contrast to our work where we integrate architecturally homogenous yet VF-optimized network layers. Our work is inspired by [23, 24], where the authors exploited multi-Vt circuit optimization for designing processors in the context of dark silicon. However, one cannot infer from their work that multi-Vt circuit optimization will be beneficial for NoCs, and what should be the architecture of a NoC with multiple VF-optimized network layers.

In summary, to the best of our knowledge, this is the first work that proposes a NoC architecture which exploits multi-Vt circuit optimization for multiple network layers in the context of system-level DVFS and dark silicon.

3. darkNoC ARCHITECTURE

We consider NoCs with predefined topologies as shown in Figure 4(a). The NoC consists of n routers, which are divided into regions. Each region has m routers, and separate voltage and frequency (VF) rails with their own VF regulators. The value of m can vary from 1 to n, depending on the granularity of the regions. For communication between VF regions, bi-synchronous links [25] have to be used. However, in this work we assume that VF regions can only communicate if they are operating at the same VF level. A designer can opt to use the darkNoC architecture in any of the VF regions based upon his/her requirements. The following paragraphs explain the darkNoC architecture from the perspective of a single VF region.

3.1 darkNoC Layers

A region contains l number of network layers, where a network layer consists of m routers as shown in Figure 4(b). At a given time, only one of the network layers is *illuminated* (activated), while the rest of the network layers remain *dark* (deactivated). When a network layer is illuminated, all of its routers are active, and thus, at a given time, only m routers are active. Figure 3 shows an example of transitioning from network layer 0 to 3 in a region with 4 network layers.

Each network layer is optimized at design-time to operate in a certain VF range. That is, multi-Vt circuit optimization of CAD tools is used to optimize all the routers of a network layer for a particular VF range. All the layers in a region are managed by a hardware-based darkNoC Layer Manager (*dLM*) as shown in Figure 4(b). The function of the *dLM* is to switch between network layers when directed by the system-level DVFS manager.

3.2 darkNoC Routers Stack

Each node in Figure 4(b) represents a stack of l routers to realize l network layers of the region. For example, for the 4 network layers in Figure 3, each routers stack has 4 routers and all the routers marked 2 belong to the network layer 2. As shown in Figure 4(b), all the routers in a stack share the same set of link wires with the neighboring stacks of routers, *which reduces verification costs.*

Each router in a stack has separate controls for power-gating and enabling input/output ports. For example, a stack of two routers in shown in Figure 4(c), where $R_0\_PWR\_en$ and R_0_en controls are used to power-on/off and enable/disable input/output ports of router

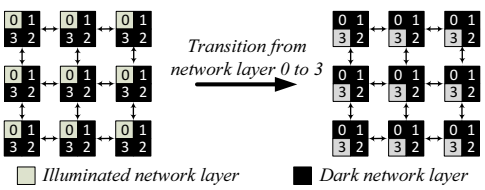

Figure 3: An example darkNoC architecture with four network layers.

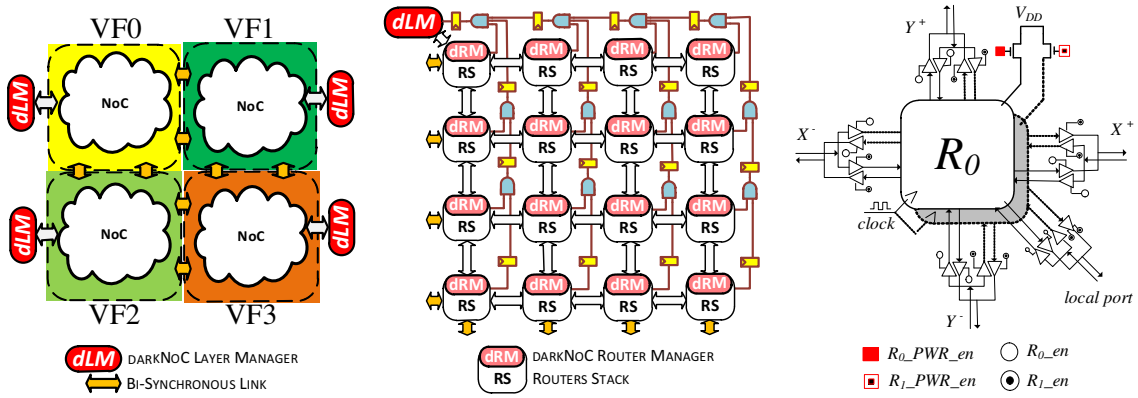

Figure 4: darkNoC architecture: **(a) (left)** NoC divided into VF regions **(b) (middle)** a single VF region **(c) (right)** a stack of (two) VF optimized routers.

Algorithm 1: dLM Function

```
1  nodes = m; // number of routers in a region
2  WaitForSwitchOverCommand();
3  for i=nodes to 0 do
4      SendDisableInjControlFlitTodRM(i);
5  WaitForNetworkFlush();
6  for i=nodes to 0 do
7      SendSwitchRouterControlFlitTodRM(i);
8  WaitForNetworkFlush();
9  for i=nodes to 0 do
10     SendEnableInjControlFlitTodRM(i);
```

Algorithm 2: dRM Function

```
   // Phase 1
1  WaitForDisableInjControlFlitFromdLM();
2  SwitchOnTargetRouter();
3  DisableInjectionFromNI();
4  if RouterLocatedAtRegionBorder then
5      SetStopFlagForNeighbourRouters();
6  WaitForRouterFlush();
7  SetEmptyFlag();
   // Phase 2
8  WaitForSwitchRouterControlFlitFromdLM();
9  DisableActiveRouterPorts();
10 EnableTargetRouterPorts();
11 PowerOffActiveRouter();
12 SetEmptyFlag();
   // Phase 3
13 WaitForEnableInjControlFlitFromdLM();
14 EnableInjectionFromNI();
15 if RouterLocatedAtRegionBorder then
16     ResetStopFlagForNeighbourRouters();
```

0, respectively. Thus, a router in a stack is illuminated in two steps: powering it on and enabling its input/output ports. The local port of a routers stack is connected to a tile (which may consist of a processor, memory, etc.) through a network interface (NI). The NI has the capability to stop injection of packets from its tile, which we exploit during the switching of network layers. For managing a routers stack, we use a hardware-based darkRouter Manager (*dRM*) as shown in Figure 4(b). The function of *dRM* is to switch between routers when directed by the *dLM*.

3.3 darkNoC Layer Switch-Over Mechanism

The switch-over between network layers is an important design requirement for our darkNoC architecture. The main challenges are: a) the lossless data communication property of packet-switched buffered NoC should be preserved, b) the switch-over mechanism should be transparent to software, and c) the switch-over mechanism should be efficient in terms of time and energy overhead. In our solution, the darkNoC Layer Manager (*dLM*) and the darkNoC Router Managers (*dRMs*) autonomously coordinate with each other to realize a switch-over mechanism with the aforementioned requirements.

At a high level, the *dLM* ensures correct switch-over between two network layers in the region. The *dLM* uses the injection channel of the NI of the routers stack it is attached to, and sends different types of control flits to *dRMs* through the NoC channels. Further, the *dLM* gathers the buffer occupancy information from the routers stacks using a single bit AND-gate network as shown in Figure4(b). Similar AND-gate networks have been used in NoC for congestion detection [26]. The *dRM* of a routers stack is attached to the ejection port of the NI, and receives control flits from the *dLM*. Based upon the control flits from the *dLM*, a *dRM* can: (a) power-on/off a router, (b) enable/diable input/output ports of a router, or (c) enable/disable injection of packets into the NI. Further, a *dRM* generates a single bit flag by analyzing the buffer occupancies of its router to indicate presence/absence of in-transit packets, which is propagated through the AND-gate network to the *dLM*.

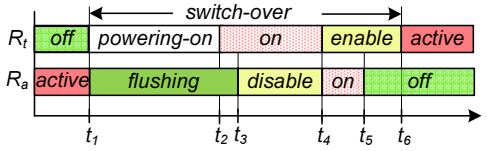

Figure 5: Timing diagram for switch-over from router R_a to R_t.

Details of dLM and dRM: Algorithms 1 and 2 report the functionalities of *dLM* and *dRM* respectively during switch-over of a network layer. Figure 5 illustrates the switch-over mechanism for a routers stack (other stacks similarly switch-over their routers in parallel). On detecting a new switch-over command (from a system-level DVFS manager), the *dLM* starts sending control flits to all the *dRMs* to initiate the switch-over mechanism (Algorithm 1, lines 2 – 4). A *dRM* enters into its first phase on receiving the control flits from the *dLM* (Algorithm 2, lines 2-5; Figure 5@t_1). The *dRM*: (1) powers-on the target router, and (2) stops NI from injecting any new packets. If the *dRM* is controlling one of the border routers, then it also sets a flag to stop neighboring routers of other regions from injecting any new packets. The reason behind stopping injection of packets is to ensure that all the in-transit packets reach their destination before the actual switch-over starts. Once the router buffers are flushed and no packets are injected by the neighboring routers, the *dRM* sets the empty flag which is propagated through the AND-gate network. It is important to note that the powering-on of target router and flushing of currently active router (Figure 5, $t_1 - t_3$) takes place simultaneously to speed up the switch-over mechanism.

Once the empty flags from all the *dRMs* have been propagated to the *dLM* through the AND-gate network, the *dLM* concludes the flushing of the currently active network layer. Then, it starts another round of control flits to all the *dRMs* to start the actual switch-over of routers (Algorithm 1, lines 5 – 7). On receiving

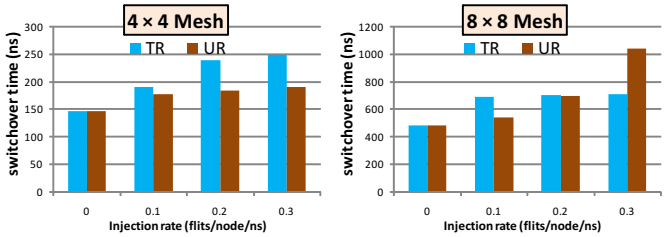

Figure 6: Time overhead of switch-over mechanism for NoC running @ 500MHz

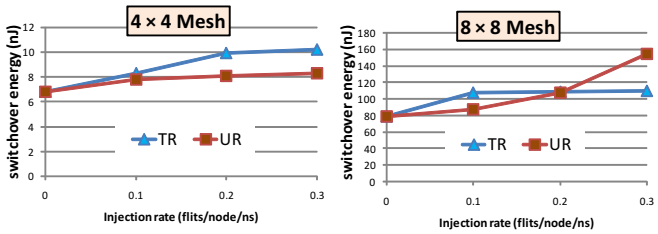

Figure 7: Energy overhead of switch-over from [500MHz,0.81V] to [750MHz,0.81V] layer

the control flit, a *dRM* enters into its second phase (Algorithm 2, lines 8 – 12), where it: (1) disables the input/output ports of the active router (Figure 5, $@t_3$), (2) enables the input/output ports of the target router (Figure 5, $@t_4$), and (3) powers-off the (previously) active router (Figure 5, $@t_5$). Note that the order of (1) and (2) is essential to avoid short-circuit conditions. At the end, the *dRM* sets the empty flag, which is propagated through the AND-gate network.

Once the *dLM* detects (from the AND-gate network) that the network is empty again, it starts the last round of control flits to all the *dRMs* (Algorithm 1, lines 8 – 10). This control flit dictates a *dRM* to enters its third phase (Algorithm 2, lines 13 – 16), where the injection of packets into NI is enabled (Figure 5, $@t_6$). Further, if the *dRM* is controlling one of the border routers, then it enables injection of packets from the neighboring routers of other regions. With this, the network layer switch-over mechanism is complete.

Time and energy overhead. Now, we present an analysis of the time and energy overhead of switch-over mechanism in dark-NoC architecture. For this analysis, we assume that the transition of VF level is done after the switch-over from an active network layer to a target network layer. We do not include the time and energy overhead of VF scaling because this overhead is present in the traditional single layer NoC as well. The time overhead in clock cycles of switch-over can be estimated as:

$$T_{so} = \textit{in-transit-packets}(m, load) + \textit{no-load}(m) + \textit{control-flits}(m, load)$$

The *in-transit-packets* term represents the time to flush the active network layer which depends on the number of routers m and the network load. The *no-load* term captures the time spent in switch-over of all the routers which only depends on m, while the term *control-flits* represents the time spent by *dLM* in sending control flits to *dRMs* and is dependent on both m and the network load. Since the whole switch-over mechanism occurs at the frequency of the active network layer, it can be used to convert the switch-over clock cycles to actual time.

The energy overhead of switch-over mechanism can be estimated as:

$$E_{so} = \textit{power-on-energy}(L_t, m) + \textit{control-flits}(L_t, L_a, m) + \textit{leakage-power}(L_t, L_a, m) \times T_{so}$$

where L_a and L_t refer to the active and target network layers respectively. The first term *power-on-energy* is the energy overhead of powering-on a network layer, which depends on the type of the target network layer and the number of routers m in it. During a typical switch-over, the control flits from the *dLM* use both the active and target network layers, and their energy consumption is captured in the *control-flits* term. The last term includes the leakage energy of the active and target layers, since both the layers consume leakage power for almost the whole duration of the switch-over mechanism.

Topology	4×4 mesh
Router Architecture	5 port router (4 neighbors + 1 local), 3 stage pipeline [LT+BW, RC+SA, ST]
Input Buffer	8 flit deep (no virtual channel)
Routing	Dimension Order XY
Link Flow Control	On/Off
Link Width	64 bit data, 8 bit control
Switch Arbiter	Matrix arbiter

Table 1: NoC architectural details.

Frequency Voltage	1 GHz 0.9 V	750 MHz 0.81 V	500 MHz 0.81 V	250 MHz 0.72 V
Area [μm^2]	49200	46313	45750	45225
HVt Cells [%]	44.7	60.5	90.7	92.8
NVt Cells [%]	12.5	13.6	4.3	3.9
LVt Cells [%]	42.8	25.9	5.0	3.3

Table 2: Synthesis results for a single router.

We experimented with 4×4 and 8×8 mesh sizes, transpose random (TR) and uniform random (UR) traffic patterns and varying traffic injection rates (details are in Section 4. The results for time and energy overhead are presented in Figures 6 and 7 respectively. Three trends are evident. The time and energy overhead: (1) increases with an increase in mesh size due to a higher number of control flits and hops, and longer delays in the AND-gate network, (2) increases with an increase in the traffic injection rates due to longer flushing times of in-transit packets and delays of control flits, and (3) heavily depends on the network load as trends are different for TR and UR traffic patterns. It is important to note that the time and energy overhead of our switch-over mechanism is not significant compared to the total execution time of typical applications and energy consumption of the NoC (see Section 4).

4. EVALUATION METHODOLOGY

4.1 NoC Synthesis

We use a packet-switched wormhole NoC architecture for experiments. The details of the NoC architecture are reported in Table 1. The proposed router is implemented in Verilog RTL, and synthesized using Synopsys Design Compiler for commerical TSMC 45nm libraries characterized for HVt, NVt and LVt cells. We enabled leakage and dynamic power optimization in Synopsys Design Compiler, which automatically enabled multi-Vt optimization. We explored the following four VF levels: [1GHz, 0.9V], [750MHz, 0.81V], [500 MHz, 0.81V] and [250 MHz, 0.72V], and the synthesis results are reported in the last four columns of Table 2.

We can observe that the target VF level significantly affects the type and number of cells used during optimization. For example, the router optimized for [1GHz, 0.9V] contains 42.8% LVt cells, which decreases to 5.0% and 3.3% for optimization at [500 MHz, 0.81V] and [250 MHz, 0.72V], respectively. Moreover, the cells are sized according to the target VF level, which has resulted in different silicon areas for different target VF levels (row 2). These results corroborate our motivational example that it is worthwhile to explore multi-Vt optimization in NoCs for energy efficiency.

4.2 Experimental Setup

MPSoC and Applications. We used a 16-node MPSoC laid as a 4×4 2D-mesh, with its architectural details in Table 3. Four VF levels are assumed for the NoC, whose architectural details are in Table 1. Note that the VF level of processors is not changed in our experiments. We used eight applications from mediabench suite and created diverse multi-programmed application mixes (AM) whose

Topology	4×4 mesh
Processors	Tensilica in-order LX4 @ 1GHz
L1 I/D Caches	4KB, 2-way set associative, 16 byte line size
DRAM	2 memory controllers, 40 ns latency
NoC VF Levels	[1GHz, 0.9V], [750MHz, 0.81V], [500 MHz, 0.81V], [250 MHz, 0.72V]

Table 3: MPSoC architectural details.

AM-1	mpeg2enc, sha, mpeg2dec, g721enc
AM-2	h264enc, mpeg2dec, sha, jpeg_dec
AM-3	jpeg_enc, sha, g721enc, g721dec
AM-4	g721enc, mpeg2enc, g721dec, mpeg2dec

Table 4: Details of application mixes with four copies of each application.

Name	Configuration	Relative Chip Area
baselineNoC	Traditional NoC with [1Ghz, 0.9V] layer	$1\times$
darkNoC1	[1GHz, 0.9V] + [750MHz, 0.81V]	$1.13\times$
darkNoC2	[1GHz, 0.9V] + [500MHz, 0.81V]	$1.13\times$
darkNoC3	[1GHz, 0.9V] + [750MHz, 0.81V] + [500MHz, 0.81V]	$1.26\times$

Table 5: NoC configurations used in experiments.

details are in Table 4. For an application mix, four copies of each application in that mix are used. We also used synthetic traffic injection models consisting of Uniform Random (UR), Bit Complement (BC) and Transpose (TR).

Simulation Setup. We used two step system simulation methodology where memory access trace of each application executing on a processor is collected from Xtensa instruction set simulator. These memory access traces are then simulated through a closed-loop cycle-accurate NoC and DRAM simulator. Our NoC simulator also modeled different VF levels accurately for the NoC. We also modeled the *dLM*, *dRM* and AND-gate network in the NoC simulator. For application mixes, each processor was executed for at least 2 million instructions. For synthetic traffic injection models, the NoC was warmed up for 100,000 clock cycles, followed by collection of statistics for 80,000 clock cycles. The injection rates were measured in $flits/node/ns$.

NoC Power Estimation. We used Synopsys PrimeTime tool to analyze the netlist generated by Synopsys Design Compiler to compute leakage and dynamic power of a NoC. We first apply this analysis to compute power values at the nominal VF level of the NoC. Then, scaled VF conditions are applied to compute power values at VF levels lower than the nominal VF level. These computed power values are used in the NoC simulator to compute the total NoC power/energy consumption. This methodology is also used for all the network layers in a darkNoC.

NoC Configurations. We used different NoC configurations in our experiments, which are listed in Table 5. The **baselineNoC** represents the traditional single layer NoC designed for the highest VF level, [1GHz, 0.9V]. *For fairness of comparison, we synthesized baselineNoC with multi-Vt optimization.* The **darkNoC1** and **darkNoC2** configurations have 2 VF-optimized network layers each, while **darkNoC3** configuration is a superset of darkNoC1 and darkNoC2. In Table 2, the difference in silicon area and distribution of multi-Vt cells between [500 MHz, 0.81V] and [250 MHz, 0.72V] optimized routers (columns 4 and 5) is small. This means that two network layers optimized for [500 MHz, 0.81V] and [250 MHz, 0.72V] will have nearly identical properties, and thus similar energy efficiencies. Therefore, a designer can use synthesis results for early elimination of network layers in a darkNoC. That is why, we did not include [250 MHz, 0.72V] optimized network layer in our experiments.

The last column of Table 5 reports the relative chip area of NoC configurations, measured as a *sum of the area of the processors, caches, network layers, and the dLM and dRMs*. A designer can use darkNoC1 or darkNoC2 if he/she has dark silicon budget of a single network layer. Likewise, he/she can use darkNoC3 if the dark silicon budget permits inclusion of 2 network layers. Analyzing router micro-architecture in context of multi-Vth optimization is our future work.

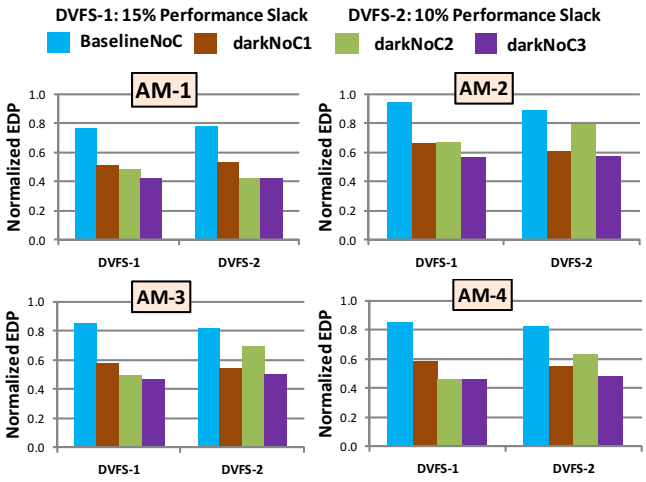

Figure 8: NoC Energy-Delay Product (EDP) normalized w.r.t traditional NoC operating at highest VF level.

System-level DVFS Manager. We used the state-of-the-art memory latency based NoC DVFS technique introduced by Chen et al. [13] for the baselineNoC. We used a control interval of $50K$ clock cycles for the DVFS. For the darkNoC configurations, we modified their technique as follows. If DVFS manager decides to switch to *i-th* VF level and there is a network layer optimized for *i-th* VF level, then the DVFS manager requests the *dLM* to switch-over to that particular network layer. On the other hand, if there is no network layer optimized for *i-th* VF level, then the DVFS manager requests the *dLM* to switch-over to the network layer optimized for the closest yet higher VF level, and will scale VF of the selected network layer. For example, in darkNoC1, if system-level DVFS manager decides to operate NoC at [250 MHz, 0.72V], then the [750 MHz, 0.81V] network layer will be scaled down to operate at [250 MHz, 0.72V] rather than the [1 GHz, 0.9V] network layer. We created two flavors of DVFS managers based upon application requirements: **DVFS-1 with a target performance loss of 15%** and **DVFS-2 with a target performance loss of 10%.** Note that our NoC simulator *included the time and energy overhead of network layer switch-over* for the darkNoC configurations.

4.3 Results and Discussion

Overall Trend in NoC Energy-Delay Product (EDP) Savings. Figure 8 reports the savings in NoC EDP for the four application mixes of Table 4, four NoC configurations of Table 5 and two DVFS managers. Overall, darkNoC configurations provide significant improvement in EDP over baselineNoC. For example, for AM-2, the DVFS-1 reduced EDP by only 5% in baselineNoC. On the other hand, EDP is reduced by 34% and 44% by DVFS-1 in darkNoC1 and darkNoC3, respectively. Likewise, for AM-3, DVFS-1 in baselineNoC saved 15% EDP, whereas DVFS-1 in darkNoC1 and darkNoC3 reduced EDP by 45% and 52%, respectively. In summary, darkNoC provides up to 56% EDP savings (AM-1, darkNoC3, DVFS-2) over baselineNoC. These results show that DVFS alone can be ineffective in reducing NoC energy due to increasing leakage power to dynamic power ratio. Thus, our darkNoC architecture can well-complement system-level DVFS managers.

Interplay of Applications, DVFS Manager and darkNoC Configurations. EDP savings vary significantly across DVFS managers and darkNoC configurations. For example, in AM-2, AM-3 and AM-4, use of DVFS-2 with darkNoC1 is far more beneficial than DVFS-2 with darkNoC2. Likewise, for AM-3 and AM-4,

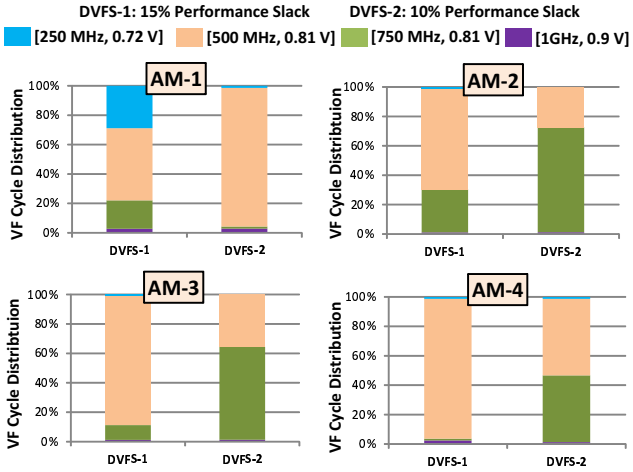

Figure 9: Distribution of NoC VF cycles.

DVFS-1 with darkNoC2 results in more EDP savings than DVFS-1 with darkNoC1.

To further explain these results, in Figure 9, we report the distribution of NoC VF cycles, which is the percentage of total execution cycles the NoC operates at a particular VF level. For AM-2, DVFS-2 operates the NoC at [750MHz, 0.81V] and [500MHz, 0.81V] levels for 71.4% and 27.4% of the time, respectively. Thus, darkNoC1 results in better EDP savings because it has a [750MHz, 0.81V] optimized network layer. In case of AM-4, DVFS-1 with darkNoC3 provided only 1% EDP improvement over darkNoC2, at the expense of an additional network layer. This is because the NoC is operated at [500MHz, 0.81V] level for 95% of the time, and thus a darkNoC with [500MHz, 0.81V] optimized network layer is more than sufficient.

Insights. From the above results, we can conclude that savings in energy-delay product depends on:

- darkNoC configuration
- System-level DVFS manager
- Applications

There is a complex interaction between these factors. For example, the energy efficiency of a darkNoC configuration (number and type of network layers) heavily depends upon the dark silicon budget and distribution of VF cycles, which in turn depends upon the target performance loss set in the system-level DVFS manager and the executing applications. Taking it one level further, the decisions taken by the system-level DVFS manager depend on the number and type of network layers in the darkNoC configuration, which results in a cyclic dependency between these factors. Thus, the problems (highlighted in Introduction section) of: (1) determining the number and type of network layers, and (2) enhancement of system-level DVFS manager to exploit multiple network layers, are heavily dependent on each other. Consequently, these problems result in a *multidimensional design space exploration problem, with no naive solution.* In future, we will attempt this multidimensional problem with either optimal algorithms or cleverly-crafted heuristics.

5. CONCLUSION

In this paper, we presented darkNoC, a novel NoC architecture consisting of multiple network layers to complement system-level DVFS managers. Each network layer is optimized at design-time to operate within a voltage-frequency range using multi-Vt circuit optimization. We utilize dark silicon that will be available in future chips to realize the extra network layers. When compared to a traditional single layer NoC, darkNoC enabled system-level DVFS to achieve up to 56% better energy-delay product. We believe that darkNoC will open new research avenues in energy-efficient NoC design for dark silicon era.

6. REFERENCES

[1] P. Salihundam, S. Jain, T. Jacob, S. Kumar, V. Erraguntla, Y. Hoskote, S. R. Vangal, G. Ruhl, and N. Borkar, "A 2 tb/s 6 x 4 mesh network for a single-chip cloud computer with dvfs in 45 nm cmos.," *J. Solid-State Circuits*, vol. 46, no. 4, pp. 757–766, 2011.

[2] J. S. Kim, M. B. Taylor, J. Miller, and D. Wentzlaff, "Energy characterization of a tiled architecture processor with on-chip networks," in *Proceedings of the 2003 international symposium on Low power electronics and design*, ISLPED '03, (New York, NY, USA), pp. 424–427, ACM, 2003.

[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 *Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium on*, pp. 35–42, 2012.

[4] A. Mishra, R. Das, S. Eachempati, R. Iyer, N. Vijaykrishnan, and C. Das, "A case for dynamic frequency tuning in on-chip networks," in *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*, pp. 292–303, 2009.

[5] K. Swaminathan, E. Kultursay, V. Saripalli, V. Narayanan, M. Kandemir, and S. Datta, "Steep slope devices: From dark to dim silicon," *IEEE Micro*, vol. 99, no. PrePrints, p. 1, 2013.

[6] S. Garg, D. Marculescu, R. Marculescu, and U. Ogras, "Technology-driven limits on dvfs controllability of multiple voltage-frequency island designs: A system-level perspective," in *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, pp. 818–821, 2009.

[7] T. Karnik, Y. Ye, J. Tschanz, L. Wei, S. Burns, V. Govindarajulu, V. De, and S. Borkar, "Total power optimization by simultaneous dual-vt allocation and device sizing in high performance microprocessors," in *Design Automation Conference, 2002. Proceedings. 39th*, pp. 486–491, 2002.

[8] H. Esmaeilzadeh, E. Blem, R. St.Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, pp. 365–376, 2011.

[9] M. Taylor, "A landscape of the new dark silicon design regime," *Micro, IEEE*, vol. PP, no. 99, pp. 1–1, 2013.

[10] N. Goulding-Hotta, J. Sampson, G. Venkatesh, S. Garcia, J. Auricchio, P. Huang, M. Arora, S. Nath, V. Bhatt, J. Babb, S. Swanson, and M. Taylor, "The greendroid mobile application processor: An architecture for silicon's dark future," *Micro, IEEE*, vol. 31, no. 2, pp. 86–95, 2011.

[11] Y. Turakhia, B. Raghunathan, S. Garg, and D. Marculescu, "Hades: architectural synthesis for heterogeneous dark silicon chip multi-processors," in *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, (New York, NY, USA), pp. 173:1–173:7, ACM, 2013.

[12] M. B. Taylor, "Is dark silicon useful?: Harnessing the four horsemen of the coming dark silicon apocalypse," in *Proceedings of the 49th Annual Design Automation Conference*, DAC '12, (New York, NY, USA), pp. 1131–1136, ACM, 2012.

[13] X. Chen, Z. Xu, H. Kim, P. V. Gratz, J. Hu, M. Kishinevsky, U. Ogras, and R. Ayoub, "Dynamic voltage and frequency scaling for shared resources in multicore processor designs," in *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, (New York, NY, USA), pp. 114:1–114:7, ACM, 2013.

[14] L. Shang, L.-S. Peh, and N. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks," in *High-Performance Computer Architecture, 2003. HPCA-9 2003. Proceedings. The Ninth International Symposium on*, pp. 91–102, 2003.

[15] H. Matsutani, M. Koibuchi, D. Ikebuchi, K. Usami, H. Nakamura, and H. Amano, "Performance, area, and power evaluations of ultrafine-grained runtime power-gating routers for cmps," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 30, no. 4, pp. 520–533, 2011.

[16] G. Kim, J. Kim, and S. Yoo, "Flexibuffer: Reducing leakage power in on-chip network routers," in *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pp. 936–941, 2011.

[17] A. Samih, R. Wang, A. Krishna, C. Maciocco, C. Tai, and Y. Solihin, "Energy-efficient interconnect via router parking," in *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, pp. 508–519, 2013.

[18] R. Das, S. Narayanasamy, S. K. Satpathy, and R. G. Dreslinski, "Catnap: energy proportional multiple network-on-chip," in *Proceedings of the 40th Annual International Symposium on Computer Architecture*, ISCA '13, (New York, NY, USA), pp. 320–331, ACM, 2013.

[19] A. Pullini, F. Angiolini, S. Murali, D. Atienza, G. De Micheli, and L. Benini, "Bringing nocs to 65 nm," *Micro, IEEE*, vol. 27, no. 5, pp. 75–85, 2007.

[20] A. K. Mishra, O. Mutlu, and C. R. Das, "A heterogeneous multiple network-on-chip design: an application-aware approach," in *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, (New York, NY, USA), pp. 36:1–36:10, ACM, 2013.

[21] K. Sankaralingam, R. Nagarajan, H. Liu, C. Kim, J. Huh, D. Burger, S. Keckler, and C. Moore, "Exploiting ilp, tlp, and dlp with the polymorphous trips architecture," *Micro, IEEE*, vol. 23, no. 6, pp. 46–51, 2003.

[22] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. F. Brown III, and A. Agarwal, "On-chip interconnection architecture of the tile processor," *IEEE Micro*, vol. 27, pp. 15–31, Sept. 2007.

[23] J. Allred, S. Roy, and K. Chakraborty, "Dark silicon aware multicore systems: Employing design automation with architectural insight," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.

[24] K. Chakraborty and S. Roy, "Architecturally homogeneous power-performance heterogeneous multicore systems," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 21, no. 4, pp. 670–679, 2013.

[25] A. Strano, D. Ludovici, and D. Bertozzi, "A library of dual-clock fifos for cost-effective and flexible mpsoc design," in *Embedded Computer Systems (SAMOS), 2010 International Conference on*, pp. 20–27, 2010.

[26] S. Ma, N. Enright Jerger, and Z. Wang, "Dbar: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip," in *Proceedings of the 38th annual international symposium on Computer architecture*, ISCA '11, (New York, NY, USA), pp. 413–424, ACM, 2011.