

A Voltage-Frequency Island Aware Energy Optimization Framework for Networks-on-Chip

Wooyoung Jang, Duo Ding, David Z. Pan
Department of Electrical and Computer Engineering
University of Texas at Austin
Austin, TX 78712, USA
{wyjang, ding}@cerc.utexas.edu, dpan@ece.utexas.edu

Abstract— In this paper, we present a partitioning, mapping, and routing optimization framework for energy-efficient VFI (Voltage-Frequency Island) based Network-on-Chip. Unlike the recent work [10] which only performs partitioning together with voltage-frequency assignment for a given mesh network layout, our framework consists of three key VFI-aware components, i.e., VFI-aware partitioning, VFI-aware mapping, and VFI-aware routing. Thus our technique effectively reduces VFI overheads such as mixed clock FIFOs and voltage level converters by over 82% and energy consumption by over 9% compared with the previous state-of-art works [10].

I. INTRODUCTION

According to the International Technology Roadmap for Semiconductors [1], silicon and system complexity is rocketing exponentially due to increasing transistor counts fueled by smaller feature sizes and increasing demands for complex functionality, higher performances with lower cost and shorter time-to-market. As SoC (System-on-Chip) designs target high-performance system level integrations of existing intellectual properties such as microprocessors, digital signal processors, controllers, memories and I/Os, previously dominant point-to-point SoC interconnections and classic bus-based mechanisms such as AMBA, STBus and Sonics MicroNetwork [2] are becoming performance bottlenecks due to the increase of system complexity. NoC (Network-on-Chip) has been recently introduced as an effective solution for scalable on-chip communication for future SoC, where the network replace the traditional shared bus structures [3][4]. As a better SoC platform for scalable system integration, on-chip network provides more competitive features than the previous ad-hoc global wiring mechanism.

Recently, VFI (Voltage-Frequency Island) and GALS (Globally Asynchronous Locally Synchronous) paradigm was introduced to NoC methodology [10], where tiles are partitioned into islands and each island is optimized with its own supply voltage, threshold voltage and operating frequency to minimize the overall energy consumption. In spite of its powerful energy efficiency, there are several

limitations. First, the partitioning process is only combined with VF (Voltage and Frequency) assignment process. Such approach limits the flexibility of VFI optimization. Next, its search for optimal energy consumption is carried out on a hard mesh network, where both communication and computation components are pre-designed. Since its network mapping is not optimized by a VFI-aware manner, the solution space of [10] is inevitably constrained. Finally, the VFI based NoC needs a good routing strategy to bring down the energy consumption by minimizing the number of a MCFIFO (Mixed Clock FIFO) and a VLC (Voltage Level Converter).

In this paper, we propose a systematic VFI-aware energy optimization framework that considers partitioning, mapping and routing together to tackle the aforementioned problems and further improve energy efficiency of VFI-based NoC designs. In the proposed approach, VFI-aware partitioning is carried out with VFI assignment, followed by VFI-aware mapping and VFI-aware routing path allocation. The proposed framework provides much more flexible VFI-aware NoC optimizations in terms of energy consumption.

The rest of this paper is organized as follows: In section II, we present a motivational example and summarize our major contributions. Section III reviews related works. Section IV introduces our VFI-aware optimization framework problem formulation. Section V presents a detailed description of the VFI-aware partitioning/mapping/routing algorithms. Section VI shows experiment results in comparison with the most recent work [10]. Finally, Section VII concludes the paper.

II. MOTIVATION AND CONTRIBUTIONS

A. Motivational Example

For global energy optimization, performing a core partitioning and a VF assignment is highly desirable before mapping cores onto NoC tiles. Fig. 1, for instance, shows two NoC designs with 16 tiles. Each tile operates at either voltage A or voltage B. Let us apply [10] into these NoC designs. [10] can improve the energy consumption by running two VFIs in Fig. 1(a). In this case, its redundancy is only four routers with

This work is sponsored in part by NSF, Texas ARP, and Samsung Electronics.

a MCFIFO and a VLC. If energy saving by operating two VFIs is more efficient than the redundancy by four complex routers, it is regarded as a good solution. However, in the case of Fig. 1(b), any tile cannot operate together at the same voltage. Operating each tile as one VFI needs a complex wiring of power, ground and clock and 24 complex routers which may be much more expensive than energy saving by VFI separation. As a result, higher voltage between two voltages will be used in overall NoC such that [10] fails to save the energy. This shows that a partitioning with a VF assignment alone may be misleading during NoC energy optimization. Our solution is to combine core partitioning, VF assignment, mapping and routing path allocation together, which consider VFI-aware manner sequentially.

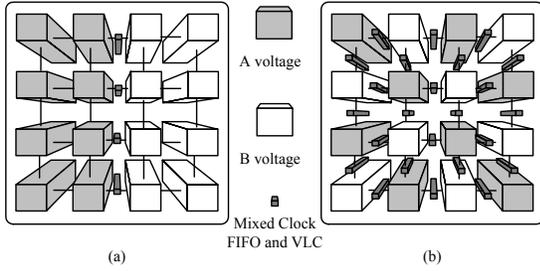


Figure 1. Motivational NoC example.

B. Major Novelty

The main novelty and contribution of the proposed VFI-aware optimization framework include:

- Earlier partitioning and VF assignment than mapping and routing path allocation provide more opportunities to build the unified VFIs.
- VFI-aware mapping is performed based on effective region growing method. Such VFI-aware mapping fits the VFI-based NoC methodology well.
- VFI-aware routing seeks to further reduce the VFI overhead through minimum traffic routing with congestion avoidance.

III. RELATED WORKS

The thriving of NoC paradigm has triggered a burst of on-chip mapping, routing and partitioning techniques in the last decade. In [5], an energy aware mapping was proposed for regular tile-based NoC structures, which was further improved in [6] by considering the packet routing flexibility during the mapping process. With on-chip communication bandwidth constraints applied, [7] developed a fast shortest path algorithm for mesh-based core mapping. With respect to routing, [8] presented a deadlock-free technique called turn model for designing partially adaptive wormhole routing algorithms without virtual channels. This approach was further improved by the odd-even turn model in [9] for fault-tolerant routing algorithms. For minimum energy consumption, voltage islands concept was applied into NoC design [10]. In addition, VFI concept combines with GALS paradigm for global on-chip asynchronous communication. In [11], the problem of energy optimal local speed and voltage selection in

VFI based system was studied under given performance constraints. [12] considered the voltage island partitioning, assignment and floorplanning in SoC Designs.

IV. PROBLEM FORMULATIONS

We start to solve VFI-applied NoC issues from a core graph consisting of cores and their communication relation since a core can be one-to-one mapped onto a tile of NoC. We use EDF (Earliest Deadline First), a heuristic called EAS (Energy Aware Scheduling) [18] and arbitrary schedulers to generate a core graph from a task graph.

A. Partitioning and VF Assignment Problem

In this stage, the object is to decide how cores should be partitioned to minimize the energy consumption except the communication energy. We assume that the maximum number of VFIs denoted by $\max\{n(VFI)\}$, a core graph G with a set of n cores where the supply and threshold voltages are $(V_1, V_{t1}), (V_2, V_{t2}), \dots, (V_n, V_{tn})$, NoC topology are given. Clock period (τ_i) for each core c_i , which can trade off with supply and threshold voltage, is defined by [10] as:

$$\tau_i(V_i, V_u) = \frac{K_i V_i}{(V_i - V_u)^\alpha} \quad (1)$$

where α is a technology parameter and K_i is a design specific constant [15][16]. Operating frequency (f_i) of the VFI j is determined by a core including the longest path as:

$$f_i \leq \min_{i \in S_j} \left\{ \frac{1}{\tau_i(V_i, V_j)} \right\} \quad (2)$$

where S_j is a set of tiles that belong to VFI j . Each core can be performed with different supply and threshold voltages and a voltage level is regarded as a legal one as long as the performance constraints can be satisfied. Based on these constraints, we partition n cores into the maximum number of VFIs given and assigned supply and threshold voltage to each core such that total power cost is minimized as follows,

$$\min \left\{ \sum_{v_i \in G} \left(R_i C_i V_i^2 + T_i k_i V_i \exp \left(-\frac{V_i}{S_i} \right) \right) \right\} \quad (3)$$

where G is a set of n cores, R_i is a number of active cycles, C_i is total switched capacitance per cycle, T_i is a number of idle cycles, k_i is a design parameter, and S_i is a technology parameter [17].

B. VFI-Aware Mapping Problem

In this section, we determine which tile each core should be mapped to in order to minimize the communication energy consumption under stringent performance constraints.

Definition 1: The partitioned core graph $G'(V, E)$ generated by section IV-A is a directed graph, where each vertex $v_i \in V$

represents a core, and each directed edge $e_{ij} \in E$ represents the communication from v_i to v_j . $vol(e_{ij})$ represents the communication volume between v_i to v_j .

Definition 2: The NoC topology graph $N(T, C)$ is a directed graph, where each vertex $t_i \in T$ represents a tile, and each directed edge $c_{ij} \in C$ represents candidate minimum paths from t_i to t_j . $vol(c_{ij})$ represents the communication volume between t_i to t_j , while $bw(c_{ij})$ represents the minimum bandwidth requirement from t_i to t_j . The one-to-one mapping function $M()$ of the partitioned core graph $G'(V, E)$ onto the NoC topology graph $N(T, C)$ is defined as follows:

$$M : V \rightarrow T, \text{ s.t. } M(v_i) = t_j, \forall v_i \in V, \exists t_j \in T \quad (4)$$

The mapping is only defined when $n(V) \leq n(T)$ where $n(X)$ is the number of $x_i \in X$. It has also two constraints, i.e., v_i should be mapped to any t_j minimizing the overall amount of communication and to any t_j operating at the same voltage with v_i if any core of VFI including v_i is mapped before.

C. VFI-Aware Routing Problem

$E_{bit}(c_{i,j})$ is the energy consumption of sending one bit of data from t_i to t_j . Assuming the bit energy values are observed at V_{DD} , this is defined by [10] as:

$$E_{bit}(c_{i,j}) = \sum_{p \in L(c_{i,j})} (E_{bit}^L(p) + E_{bit}^B(p) + E_{bit}^S(p)) \frac{V_i^2}{V_{DD}^2} \quad (5)$$

where $L(c_{i,j})$ is a set of links passed from t_i to t_j and E_{Lbit} , E_{Bbit} and E_{Sbit} is the energy consumed by the link, buffer and switch fabric, respectively. Therefore, finding a routing path from t_i to t_j is formulated to minimize follows:

$$\min \left\{ \sum_{\forall e_{i,j}} [vol(e_{i,j}) E_{bit}(c_{i,j})] + \sum_m (E_{Vconv}(m) + E_{MixClkFifo}(m)) \right\} \quad (6)$$

subject to performance constraint including processing delay and communication delay. E_{Vconv} and $E_{MixClkFifo}$ is the energy overhead of a VLC and a MCFIFO respectively.

V. VFI OPTIMIZATION FRAMEWORK

In this section, we present the proposed VFI-aware NoC methodology and detailed algorithms. Fig. 2 shows the overall flow chart of the proposed VFI-aware NoC optimization framework. We first partition n cores but not tiles into m VFIs given. Based on the partitioning of cores, a novel VFI-aware mapping algorithm and routing path allocation are applied to minimize communication energy consumption. We establish unique interconnection for key traffic paths between islands to remove the overhead of VFI. After routing path allocation is carried out, we compute the energy consumption and performance. If they are satisfied, we can get energy-efficient NoC platform with VFIs. Otherwise, we repeat these procedures decreasing the maximum number m of VFIs.

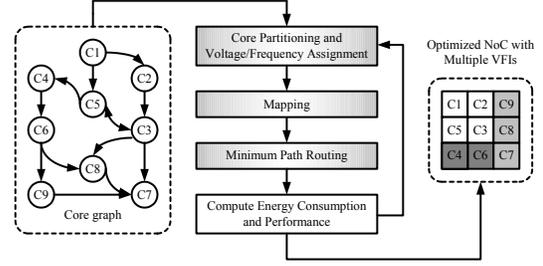


Figure 2. The proposed VFI-Aware NoC Methodology.

A. VFI-Aware Partitioning and VF Assignment Algorithm

The proposed VFI-aware partitioning algorithm is different from [10] partitioning only tiles placed in a neighbor on NoC grid. Since the partitioning stage is performed before the stages of mapping and routing path allocation, any core can be clustered together to the same VFI. Algorithm 1 shows the proposed partitioning algorithm for a given core graph $G(V, E)$ and the maximum number m of VFIs. Since we assume that the voltage of a core can trade off with its operating frequency, in line 1, the lowest supply and threshold voltage of each core are computed by (1), which satisfies performance of each core. If there are k VF used by cores and m VFIs are built, we can choose m VF among k VF, which are total kC_m cases. Then, the lowest VF among the chosen m VF are assigned to each core if performance of each core is satisfied by the VF level. When the chosen m VF are satisfied with performance of all cores, energy consumption is computed by (3). This procedure repeats all kC_m VF cases. After completing this procedure, we choose the best VF pair consuming the lowest energy.

Algorithm 1: VFI-Aware Partitioning and VF Assignment

Input: $G(V, E)$, $\max\{n(VFI)\}=m$

- 1: compute the lowest voltage of each core satisfied with performance using (1);
- 2: **for** all cases that choose m voltages among all voltages (k) used in each core **do**
- 3: assign the lowest operable voltage among m to all n cores;
- 4: **if** chosen m voltages are satisfied with performance of all n cores **then**
- 5: compute overall energy consumption by (3);
- 6: **end if**
- 7: **end for**
- 8: choose the best VF pair consuming minimum energy;

Output: $G'(V, E)$ partitioned into VFI

B. VFI-Aware Mapping Algorithm

We can know which cores a VFI consist of because the core partitioning is performed in previous section. Therefore, this information should be reflected in a mapping stage. In the mapping step, we use a heuristic approach based on the partitioned core graph, as shown in Algorithm 2. In line 1, cores are sorted in decreasing order by the amount of traffic and then they are mapped in the order. We define a $VF_List()$ indicating whether VF of current core being mapped is already used on NoC grid. From line 3 to 11, initial mapping algorithm starts for all sorted v_i . In line 4, the proposed mapping algorithm checks whether VF of current core being mapped is used throughout $VF_List()$. If VF of the core being mapped does not exist in $VF_List()$, the core is mapped on any empty tile of NoC grid with the maximum neighbor tiles and

the minimum traffics and VF of the core is recorded in $VF_List()$ (line 5-6). If VF of the core being mapped exists in $VF_List()$, the core is mapped on any candidate tile with the same VF (line 8). Then candidates are marked in line 10, where $NSWE(t_i)$ indicates north, south, west and east tile of the mapped t_i . They can be candidates for the next cores with the same VF. This repeats until all cores are mapped onto NoC grid. The initial mapping algorithm reduces the number of isolated tiles separating from the group of cores (VFI) running at the same VF. However, since we cannot completely remove an isolated tile around the edge of NoC grid, the isolated tile can be moved into near its main VFI if the moving costs are less than the overhead of the extra isolated tiles (line 13). The procedure is repeated until isolated tiles disappear. Finally, a pair-wise swapping of tiles within each island is executed to find the best mapping for minimum traffics (line 16).

Algorithm 2: VFI-Aware Mapping

Input: $G(V,E)$, NoC topology

- 1: $sort(vol(v_i))$ in decreasing order;
- 2: $VF_List() = empty$;
- 3: **for** all sorted v_i **do** // initial mapping
- 4: **if** VF of v_i does not exist in $VF_List()$ **then**
- 5: $M(v_i)$ on any empty tile t_j with max. neighbors and min. traffics;
- 6: add VF into $VF_List()$;
- 7: **else then**
- 8: $M(v_i)$ on any candidate tile t_j with min. traffics;
- 9: **end if**
- 10: add empty $NSWE(t_i)$ into candidate with VF of v_i ;
- 11: **end for**
- 12: **for** all isolated island t_i **do** // moving of an isolated tile
- 13: $pair-wise\ swapping(t_i, t_j)$ to be clustered to VFI using the same VF under min. traffic increase;
- 14: **end for**
- 15: **for** all t_i **do** // minimization of the overall traffics
- 16: $pair-wise\ swapping(t_i, t_j)$ within island for min. traffic;
- 17: **end for**

Output: $N(T,C)$ mapped onto NoC grid

Fig. 3 is a simple example for the initial mapping algorithm. In Fig. 3(a), the number is the mapping order by sorting cores depending on the amount of traffic of the cores and two clusters, i.e., grey and white groups denoted VFI 1 and VFI 2 respectively exist. The core 1 which has the maximum traffic is placed onto the center of the mesh nodes including the maximum neighbors as shown in Fig. 3(b). Four candidates, a , b , c and d are also marked as a VFI 1 for the next mapped core using the same VF. Core 2 which has the next maximum traffics is placed onto the candidates minimizing the communication cost with the cores previously mapped if VFI 2 running at the same VF of core 2 exists. Otherwise, core 2 is only placed onto any unmapped tile that minimizes the communication cost with cores previously mapped. Core 2 is mapped by latter case as shown in Fig. 3(c). Three candidates, e , f and g are also marked as a VFI 2 for the next mapped core using the same VF. Core 3 which has the next maximum traffics is placed onto the candidates of VFI 1 minimizing the communication cost with the cores previously mapped. In Fig. 3(c), there are three candidates, b , c and d and candidate b is chosen because b generates minimum traffics than c and d . Then, three candidates, e , h and i are also marked as a VFI 1, where any core running at VFI 1 and VFI 2 can be mapped to tile e in Fig. 4(d). The procedure repeats until all of

the cores are mapped as shown in Fig. 3(e)-(f). Since this method makes the region of VFI grow toward its candidates, a VFI is prevented splitting into two VFIs using the same VF.

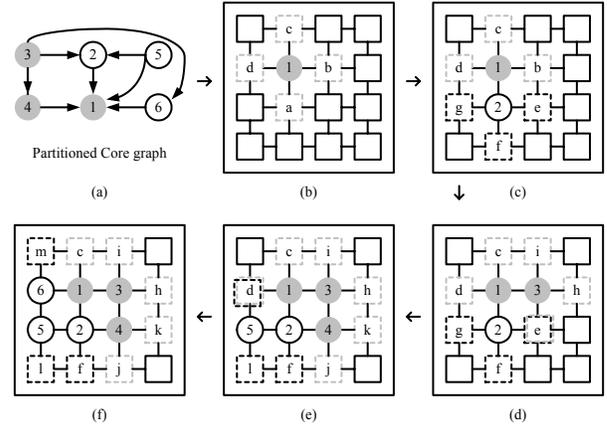


Figure 3. Incremental core mapping onto NoC grid.

C. VFI-Aware Routing Path Allocation

In this section, we present the VFI-aware routing path allocation algorithm. The concept of the proposed routing path allocation is to build minimum interconnection between VFIs. Fig. 4 shows how interconnection between tiles is built briefly. After VFI-aware mapping in section V-B, we assume that there is no interconnection between tiles as shown in Fig. 4(a). In Fig. 4(b), all of the interconnections are built within each VFI. Then, interconnections between VFIs are partially built, as shown in Fig. 4(c). Next, a routing path should be considered to minimize energy consumption and to improve performance in irregular NoC interconnection of Fig. 4(c).

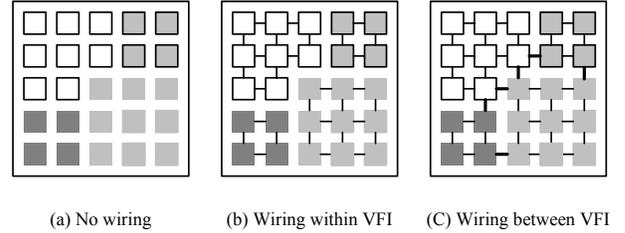


Figure 4. The procedure of connection between tiles or VFIs.

Algorithm 3 shows how an interconnection between tiles is built and how a routing path is allocated. First, we should connect all tiles within each VFI (line 1) as shown in Fig. 4(b) because a router without a MCFIFO and a VLC is cheap. The optimal number of expensive routers with a MCFIFO and a VLC for connecting two islands are computed as:

$$b = \lceil w_{i,j} vol(VFI_{i,j}) / bw(VFI_{i,j}) \rceil \quad (8)$$

where $\lceil x \rceil$ is the smallest integer larger than x , $w_{i,j}$ is the weight of link between VFIs and $vol(VFI_{i,j})$ and $bw(VFI_{i,j})$ is the communication volume and minimum bandwidth requirement between VFI i and VFI j , respectively (line 2).

The b number of routers with a MCFIFO and a VLC are placed between two VFIs, where minimum traffics are generated. Here is a simple example in Fig. 5, where S1, S2 and S3 communicate with D1, D2 and D3 respectively. We assume that the amount of each communication is 1Mbit/s, each link can contain 5Mbit/s and w_{ij} is 1. Therefore, $vol(VFI_{i,j})$ is 3Mbit/s and $bw(VFI_{i,j})$ is 5Mbit/s such that b is 1. Now, we can build one connection between islands. The overall amount of traffic applied the shortest path is 9Mbit/s, 11Mbit/s and 7Mbit/s in Fig. 5(a), (b) and (c) respectively. Therefore, we build one interconnection between two islands like Fig. 5(c) because it generates minimum traffic.

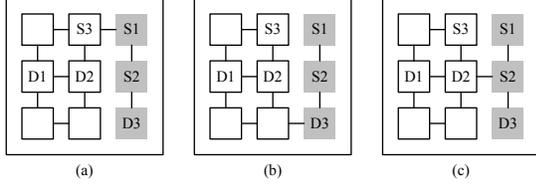


Figure 5. Interconnection between islands.

Now, we perform a routing path allocation for irregular interconnection. In line 4, c_{ij} is sorted by communication distance. In Fig. 5(c), the path from S2 to D2 is the shortest and the paths from S1 to D1 and from S3 to D3 are the same.

Rule 1: The short path c_{ij} among C is allocated earlier to relieve traffic congestion between VFIs.

In Fig. 6(a), the path from S1 to D1 has only path A as the shortest path and the path from S2 to D2 has two paths, i.e., B and C as the shortest path. If the path from S2 to D2 is allocated to path B earlier than the path from S1 to D1, the path A will overlap with the path B since the path from S1 to D1 has no choice. However, if the path A is allocated earlier than the path from S2 to D2, the path C but not the path B can be chosen as the path from S2 to D2. Therefore, routing order is important to reduce traffic congestion and balance network load. In Fig. 5(c), the path from S2 to D2 is allocated earlier and then S1 to D1 or S3 to D3 is allocated according to rule 1.

Rule 2: If the VFI of communication source is different from the VFI of its destination, the shortest path which passes through the fewest islands is allocated.

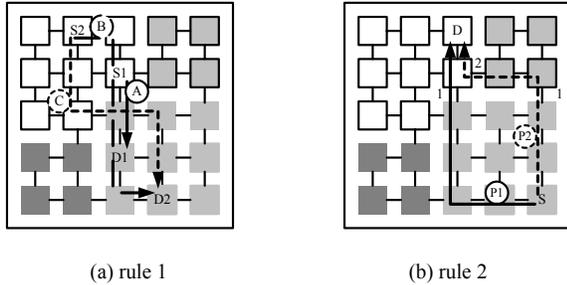


Figure 6. Routing Path Allocation.

In Fig. 6(b), the path P1 passing through one island is better than the path P2 passing two islands due to better performance and lower energy consumption from (5). As a method for rule 2, we put a cost into links located between

VFIs (line 5). For each c_{ij} , a quadrant graph is formed (line 7) and then, the path with minimum cost is obtained within the quadrant graph by Dijkstra's shortest path algorithm. If performance is not satisfied, link weight w_{ij} between VFIs should increase, go to line 2 and then repeat this procedure.

Algorithm 3: VFI-Aware Routing Path Allocation

```

Input:  $N(T,C)$ 
1: connect all tiles within each VFI;
2: compute the optimal number of routers with a MCFIFO or a VLS
   between two VFIs by (8);
3: insert the  $b$  number of routers to any place between VFIs, where
   minimum traffic is generated;
4:  $sort(length(c_{ij}))$  in increasing order; // rule 1
5: put a cost to link located between VFIs; // rule 2
6: for all  $c_{ij}$  do
7:   quadrant graph is formed from source to destination;
8:   Dijkstra's shortest path algorithm;
9:   If performance is not satisfied then
10:    increase  $w_{ij}$  of (8);
11:    go to line 2;
12:   end if
13: end for

```

Output: deterministic, minimal, deadlock-free routing path

VI. EXPERIMENTAL RESULTS

In this section, we show the experimental results obtained by applying the VFI-aware NoC framework on MPEG-4 Video Object Plane Decoder [13] and E3S benchmark suites [14]. Because the first application has 16 cores, this is mapped onto 4x4 NoC grids. The second application consists of office-automation, consumer, networking, auto-industry and telecom application containing 5, 12, 13, 24 and 30 tasks respectively. They are scheduled on to 4, 9, 9, 16 and 25 processors respectively by arbitrary schedulers. They are again mapped onto 2x2, 3x3, 3x3, 4x4 and 5x5 NoC grids respectively.

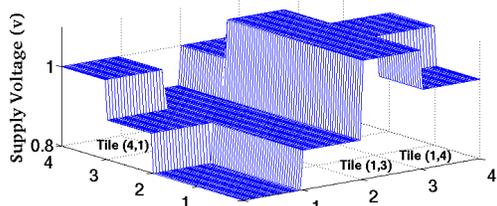
We experiment the VFI-aware NoC methodology by two versions, i.e., VFI-aware mapping combined with general minimum path algorithm and VFI-aware mapping and routing to verify the performance of mapping and routing, denoted as VFI-M and VFI-R respectively. Table I shows that the VFI-M saves more MCFIFOs and VLCs due to VFI-aware mapping based on early partitioning. In addition, VFI-R needs the least number of MCFIFOs and VLCs. The VFI-aware NoC approach commonly causes a slight increase of traffic due to VFI-aware mapping and routing path allocation. However, the maximum congestion is more relieved because routing order is considered to balance network load. The lower congestion makes a chip stable and operating at a low clock. A thorough cross-compare of E3S benchmark is listed in Table II. Its runtime ranges from a few seconds to a few minutes.

TABLE I. COMPARISON OF VIDEO OBJECT PLANE DECODER

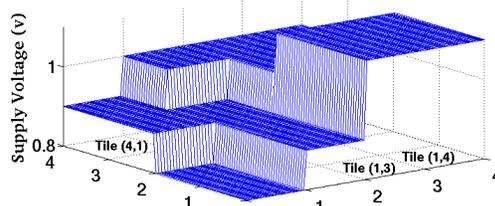
Content	Algorithm	2-VF	3-VF	4-VF
# of complex router	[10]	6	11	14
	VFI-M	5	7	10
	VFI-R	1	2	3
Total traffic (MB/s)	[10]	4309	4309	4309
	VFI-R	4353	4211	4211
Congestion (MB/s)	[10]	923	923	923
	VFI-R	516	613	613

TABLE II. CORSS COMPARISON OF E3S BENCHMARK

Content	Algorithm	Telecommunication				Auto-Industry			Networking			Consumer		
		2-VF	3-VF	4-VF	5-VF	2-VF	3-VF	4-VF	2-VF	3-VF	4-VF	2-VF	3-VF	4-VF
# of complex router	[10]	20	24	29	29	11	12	15	8	8	9	4	8	10
	VFI-M	13	14	22	20	8	10	13	6	6	7	3	5	7
	VFI-R	10	11	18	16	6	7	11	5	5	6	3	4	6
Total traffic	[10]	107	107	107	107	172	172	172	79692	79692	79692	30	30	30
	VFI-R	133	133	138	153	178	193	205	83886	83886	109051	33	35	39



(a) NoC Partitioning by [10]



(b) The Proposed VFI-Aware NoC

Figure 7. VFI partition illustration on 4x4 NoC.

Fig. 7 illustrates the comparison of [10] and the proposed VFI-aware approach performed on 4x4 NoC. Now that [10] has six VFIs (including three islands separated) and the VFI-aware NoC has four VFIs, the VFI-aware NoC framework clearly provides better partitioned VFI such that it is beneficial for low energy consumption as well as a physical design. Table III shows that the VFI-R consumes less energy than [10] since many tiles running at the same VF can be partitioned into a VFI. However, in case of Network application, the VFI-aware NoC approach is worse than [10] because the amount of traffic enormously increases in case of 4-VFI.

TABLE III. NOC ENERGY CONSUMPTION COMPARISON

Benchmark	Algorithm	Normalized Total Energy Consumption			
		1-VFI	2-VFI	3-VFI	4-VFI
Consumer	[10]	1	0.56	0.53	0.54
	VFI-R	1	0.55	0.51	0.50
Network	[10]	1	0.8	0.79	0.79
	VFI-R	1	0.78	0.76	0.89
Auto-industry	[10]	1	0.69	0.65	0.67
	VFI-R	1	0.63	0.59	0.58
Telecom	[10]	1	0.58	0.57	0.58
	VFI-R	1	0.53	0.51	0.49

VII. CONCLUSION

We proposed a systematic energy optimization framework, including VFI-aware partitioning, VFI-aware mapping and VFI-aware routing for VFI based NoC paradigms. Compared to the recent state-of-the-art NoC design techniques with VFI [10], our VFI-aware optimization framework demonstrates an energy efficiency improvement of over 9% and the overhead reduction of over 82% under a variety of system constraints.

REFERENCES

[1] International Technology Roadmap for Semiconductors, 2006.

- [2] D. Wingard, "Micronetwork-based integration for SoCs" In Proc. Design Automation Conf., 2001.
- [3] Luca Benini and Giovanni De Micheli, "Network on chips: a new SoC paradigm," Computer, 2002, 35, pp. 70-78.
- [4] William J. Dally and Brian Towles, "Route packets, not wires: on-chip interconnection networks," In Proc. Design Automation Conf., 2001.
- [5] Jingcao Hu and Radu Marculescu, "Energy-aware mapping for tile-based NoC architectures under performance constraints," In Proc. Asian and South Pacific Design Automation Conference, 2003.
- [6] Jingcao Hu and Radu Marculescu, "Exploiting the routing flexibility for energy/performance aware mapping of regular NoC architectures," In Proc. Design, Automation & Test in Europe, 2003.
- [7] Srinivasan Murali and Giovanni De Micheli, "Bandwidth-constrained mapping of cores onto NoC architecture," In Proc. DATE'04.
- [8] C. J. Class and L. M. Ni, "The turn model for adaptive routing," J. ACM, vol. 41, no. 5, pp. 874-902, Sept. 1994.
- [9] Ge-Ming Chiu, "The odd-even turn model for adaptive routing," IEEE Trans. Parallel and Distributed Systems, vol. 11, no. 7, pp. 729-738, July 2000.
- [10] Umit Y. Ogras, Radu Marculescu, Puru Choudhary and Diana Marculescu, "Voltage-frequency island partitioning for GALS-based networks-on-chip," In Proc. Design Automation Conf., 2007.
- [11] Koushik Niyogi and Diana Marculescu, "Speed and voltage selection for GALS systems based on voltage/frequency islands," In Proc. Asian and South Pacific Design Automation Conference, 2005.
- [12] J. Hu, Y. Shin, N. Dhanwada, and R. Marculescu, "Architecting voltage islands in core-based system-on-a-chip designs. In Proc. ISLPED'04.
- [13] EB.Van Der Tol, EGT. Jaspers, "Mapping of MPEG-4 decoding on flexible architecture platform," SPIE 2002, pp. 1-13, Jan. 2002.
- [14] Robert P. Dick, "Embedded system synthesis benchmarks suites", <http://www.ece.northwestern.edu/~dickrp>.
- [15] S. Martin, K. Flautner, T. Mudge and D. Blaauw, "Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads," In Proc. ICCAD'02.
- [16] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," IEEE Journal of Solid-State Circuits, 25(2), pp. 584-594, Apr. 1990.
- [17] J. A. Butts and G.S. Sohi, "A static power model for architects," In Proc. Intl. Symp. of Microarchitecture, Dec. 2000.
- [18] J. Hu and R. Marculescu, "Communication and task scheduling of application-specific networks-on-chips," In IEE Proc. Computers & Digital Techniques, pp. 643-651, Sep. 2006.