



<b>Title</b>	Application-Specific Express Virtual Channels Insertion for Low-Power NoCs
<b>Author(s)</b>	Zhang, Min; Choy, Chiu-Sing
<b>Citation</b>	Proceedings : APSIPA ASC 2009 : Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference, 738-746
<b>Issue Date</b>	2009-10-04
<b>Doc URL</b>	<a href="http://hdl.handle.net/2115/39799">http://hdl.handle.net/2115/39799</a>
<b>Type</b>	proceedings
<b>Note</b>	APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference. 4-7 October 2009. Sapporo, Japan. Oral session: Intelligent Circuits and Systems (7 October 2009).
<b>File Information</b>	WA-L1-4.pdf



[Instructions for use](#)

# Application-Specific Express Virtual Channels Insertion for Low-Power NoCs

Min Zhang and Chiu-Sing Choy

The Chinese University of Hong Kong, Shatin, Hong Kong, China

E-mail: {mzhang, cschoy}@ee.cuhk.edu.hk Tel: +852-2609-8272

**Abstract**—A flow control mechanism, express virtual channels (EVCs), was recently presented to close the gap between packet-switched networks and the ideal interconnection through virtually bypass intermediate routers. Currently, EVCs paths are inserted regularly. This paper proposes a novel methodology to insert EVCs in an application-specific way (AS-EVC) by exploiting communication characteristics of applications, with the main objective to reduce more NoCs power. Additionally, accurate evaluations were performed based on low-level, synthesizable models. The results on a range of synthetic and real workloads show that up to 23.49% router power is saved by AS-EVC, compared to the baseline design. Also, AS-EVC outperforms static EVCs under all traffics, with 57.14% more power is saved on average for tested Minne-SPEC benchmarks.

## I. INTRODUCTION

Integrating multi process elements (PE) in a Systems-on-Chip (SoCs) becomes inevitable in order to utilize the exploding number of transistors in a single piece of silicon. The number of PEs in a SoCs is predicted to be around 80 in 2010, 270 in 2015, and 880 in 2020 [1]. The shared bus and dedicated point-to-point wires can't meet communication requirements of such SoCs. Therefore, networks-on-chip (NoCs) is proposed to replace them as the communication infrastructure.

As the design focus shifts from computation-centric to communication-centric, the power of communication has become comparable to the computation power, and is expected to eventually surpass them [2]. For instance, in the MIT Raw microprocessor, the on-chip networks consumes 36% of the total chip power, only 9% lower than the main processor [3]. This shows the significance to reduce the power demand of NoCs.

Packet switching technique is generally applied in NoCs to provide high-bandwidth by multiplexing of network physical channels<sup>1</sup>. However, the benefit comes at a high power cost due to the hardware complexity in designing a packet-switched router. Firstly, routers consume a certain amount of power even when no traffic is transferred, which is referred as standby power [4]. Secondly, as a packet travels from the source to the sink, it dissipates some additional streaming power for buffering, arbitration, and switch in case the contention happens. The router-dominated nature in terms of NoCs power is claimed in many researches [3, 4]. The

implementation results show that the power ratio of routers to physical links is from two to three in both 90nm and 65nm technologies [5].

**Importance of bypass.** The streaming power for a flit is denoted as  $E_{flit} = E_{router} \times DM + E_{link} \times DM$ , where DM is the Manhattan distance. The first component is the total power of routers. The second component is the total power of links. To alleviate the router-dominated problem, a bypass technique is proposed to reduce the total power of routers by completely avoiding traversal for some routers, thus effectively reducing the power of skipped routers to zero [3]. Inspired by this idea, express cubes [3] and application-specific long links [6] are inserted to reduce power respectively. However, they have to add ports at the both ends of express cubes or long links. These routers dissipate much more power than baseline routers, which partially offsets the power saving caused by low hop counts. The key, therefore, is to reduce  $H$  while keeping  $E_{router}$  overhead small. A novel flow control technique, named *express virtual channels* (EVCs), is proposed to overcome this problem [7]. EVCs remove the need of high-radix routers at the sources and sinks of express physical channels by using express virtual paths. Experiments through high-level models show that EVCs improve latency, throughput and energy-efficiency with small cost.

**Significance of application awareness.** On the other hand, application awareness places an important role within the realm of NoCs optimization, especially for SoCs domain where an accurate estimation of the communication patterns is often possible [8]. Furthermore, due to EVCs doesn't change the regular network topology, application characteristics can be well employed in it to improve the effectiveness of EVCs without changing the structured wiring.

However, the authors in [7] don't account for the application characteristics. Thus, the insertion of EVCs is carried out in a regular way. Different from this, we aim to reduce more NoCs power in an application specific insertion way (AS-EVC). We believe that emphasizing the role of communication characteristics increases the optimization room for EVCs insertion. This idea is based on two observations. Firstly, as the aggregate traffic load of a router pair<sup>2</sup> is generally different from that of another pair,

<sup>1</sup> Channels, links, and paths are used interchangeably throughout the paper

<sup>2</sup> Note that a router pair is directed. Thus, the pair from  $r_i$  to  $r_j$  is different from the pair from  $r_j$  to  $r_i$ .

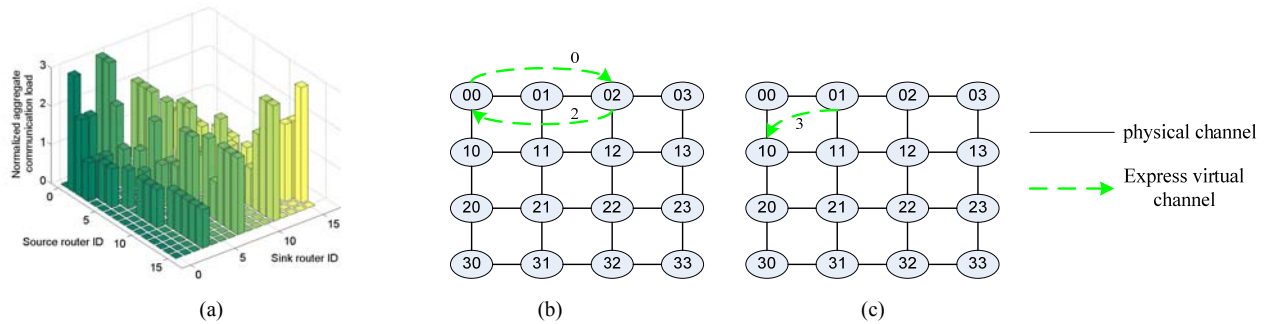


Fig. 1 Illustrations of static EVC insertion and AS-EVC insertion. (a). Aggregate communication loads of router pairs. (b). An example of static EVC insertion. (c). An example of AS-EVC insertion

they should be distinguished. Secondly, more power is saved if more traffic loads pass through it after an EVC path is inserted. Let's illustrate it by a  $4 \times 4$  mesh with XY routing for transpose traffic (Fig. 1)<sup>3</sup>. The normalized aggregate communication loads of router pairs demonstrate large variances, from 0 to 3. In the static EVC insertion, two EVC paths are inserted. One is from router 00 to router 02, and another is from 02 to 00. However, the EVC path from 00 to 02 has definitely no power saving since its traffic load is 0. This bad EVC is added due to the insertion is done in a blind way. The EVC path from 02 to 00 has power saving of 2 units. On the contrary, in our AS-EVC insertion scheme, the router pair with the largest aggregate load (from 01 to 10) is found and an EVC path is inserted there, thereby leading to power reduction of 3 units. Clearly, a 1.5x power saving is obtained by only one smart EVC path compared to that by two static EVC paths.

In order to further improve the efficiency of EVCs insertion, we remove some limitations of the static EVCs. 1). EVC paths are not limited to be straight along X or Y dimension. Switch-dimension EVC paths can be inserted. 2). Two paths between  $r_i$  and  $r_j$  are considered separately. In this way, inserting an EVC path from  $r_i$  to  $r_j$  doesn't mean an EVC path will be inserted reversely from  $r_j$  to  $r_i$ . 3). A maximum interval instead of a fixed interval is set. The length of an EVC path can be any value smaller than the pre-set maximum interval. 4). EVC source and sink routers are allowed to be bypassed.

Although high-level models developed in [9] may provide valuable power estimates and help the designers to rapidly evaluate network architectures early in the design cycle, the results are limited in accuracy. Also, some insights such as the impact of clock gating are unattainable in high-level evaluations. Therefore, our other contribution is the design of low-level models supporting the EVCs technique and the detailed, accurate low-level power evaluations. An important feature of our models is that they provide the non-uniform buffer architecture<sup>4</sup> for each input port, which is never supplied by state-of-the-art low-level NoCs simulators.

<sup>3</sup> For simplicity, the cost caused by the aggregate communication volume traveling an EVC source router is not considered in the illustration.

<sup>4</sup> Buffer architecture means the number and the depth of VC lanes.

The rest of the paper is organized as follows. In section II, a summary of related work is presented. In section III, we propose an application specific EVC insertion methodology. Section IV discusses the experiment results. Section V concludes the work and indicates possible future work.

## II. RELATED WORK

### A. Bypass Techniques

Express cubes was firstly proposed to improve network performances for k-ary n-cube off-chip networks which is node-limited [10]. Unfortunately, on-chip networks has router-dominated nature as well. Hence, express cubes was applied for power and energy efficient NoCs design [3]. Different from express cubes where long physical links were added to the network in a regular way, Ogras et al. [6] inserted long physical paths in an application-specific fashion. But, bypassing by adding express physical channels changes the original network topology and generates high-radix routers and hence large overhead. Thus, a bypass technique using express virtual channels was proposed where a long path is virtually built through flow control, avoiding fatter routers [7]. However, the express virtual channels were inserted without considering the communication features of applications, which misses a big optimization opportunity. Also, energy was evaluated by high-level models which have limited accuracy. Token flow control [11] is the latest bypass technique. The express virtual paths are formed and released dynamically, according to tokens that are indications of resource availability in the network. This *on-line* method utilizes instant run-time network states to improve effectiveness of EVCs. But, we propose a different strategy to optimize EVCs insertion *off-line* by digging the communication profile obtained from long time simulations or realistic system runnings.

### B. Low-Power Microarchitecture techniques

Low-power router components like write-through input buffers and segmented crossbar were presented in [3]. A self-corrected green coding scheme was proposed to provide low-power interconnects [12]. While these techniques targeted to save per-hop power dissipation, we aim to reduce the number of routers which a packet traverses. Thus, our proposed

technique is orthogonal to low-power microarchitecture techniques. Combining them together can achieve even larger power saving.

### III. AS-EVC INSERTION ALGORITHM

#### A. Basic Assumptions

Given an application communication graph  $CG$ , a topology graph  $TG$  and a mapping function  $M$  [13], the communication volumes between network routers can be calculated, where is the start point of our work. A 2-D mesh topology with  $m \times n$  tiles is studied. However, the proposed algorithm can be applied to other topologies with small modifications.

We use wormhole switching (WH) to build fast, inexpensive routers. Furthermore, virtual channel (VC) technique is applied to increase throughput. Deterministic XY routing algorithm is selected for the mesh network for two reasons. Firstly, traffic flows between routers can be predicted well, which is the basis to find the most beneficial place for the EVC insertion. Secondly, XY routing is simple because no deadlock recovery and flit reorder are needed.

Input buffers are partitioned by virtual channel. This means that a separate memory is provided for each VC. It is easy to implement whereas it is inefficient in sharing buffers. But, memory utilization can be increased largely through virtual channel planning [14]. The FIFO buffers for a VC can be implemented by either SRAM or registers, depending on the FIFO depth, not FIFO width [15]. We use registers to implement a FIFO since the depth is only four.

#### B. Problem Formulation

Simply stated, assuming a reasonable mapping has been done from an application to a network topology, our objective is to decide which router pair should a EVC is inserted to, such that the maximum power saving is achieved. We firstly make some definitions to formulate the problem.

**Definition 1:** A router communication graph,  $RCG = G(R, C)$ , is a directed graph, where  $R$  is the set of routers and  $C$  is the set of communications. For a communication  $c_{i,j} \in C$ ,  $c_{i,j}$  represents the communication volume from a source router  $r_i$  to a sink router  $r_j$ . In other words,  $c_{i,j}$  only includes the traffic generated from  $r_i$  and consumed by  $r_j$ .

**Definition 2:** A router aggregate communication graph,  $RAG = G(R, A, B)$ , is a directed graph, where  $R$  is the set of routers,  $A$  is the set of aggregate communications between router pairs, and  $B$  is the set of aggregate communications traveling routers. For an aggregate communication  $a_{i,j} \in A$ ,  $a_{i,j}$  means the aggregate communication load from  $r_i$  to  $r_j$ . Note that  $a_{i,j}$  includes all the traffics flowing from  $r_i$  to  $r_j$ . For an aggregate communication  $b_i \in B$ ,  $b_i$  denotes the aggregate communication traveling  $r_i$ . The calculations for  $a_{i,j}$  and  $b_i$  are explained in section III.C.

In addition, many parameters are used in this paper. They are listed in Table 1 for reference.

TABLE 1  
PARAMETER LIST

Parameter	Description
$DM$	Manhattan distance travelled by a message. $DM_{i,j} =  i_x - j_x  +  i_y - j_y $
$DV$	Virtual distance travelled by a message. The computation is illustrated in Fig. 2.
$E$	Energy consumption of a component.
$f$	The normalized inter-router communication volume. $f_{i,j} = \frac{c_{i,j}}{\sum_i \sum_{j \neq i} c_{i,j}}$
$g$	The routing algorithm related coefficient.
$\alpha$	Aggressive express pipeline: $\alpha=0$ . Non-aggressive express pipeline: $\alpha=1$ .
$\beta$	The energy ratio of a crossbar to a router.
$\lambda$	The energy ratio of an EVC source router to a normal router.
$\mu$	The average inter-node distance.

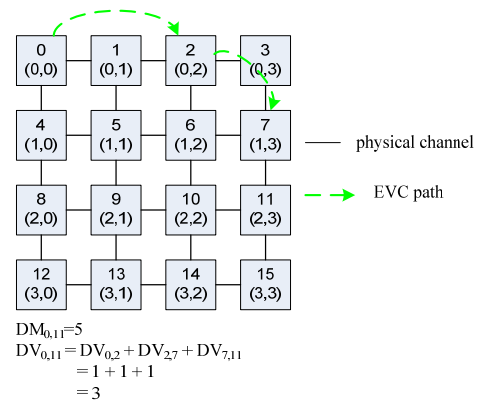


Fig. 2 Illustration of  $DV_{i,j}$  computation

Using these notations, the problem to insert EVCs in an application-specific way can be formulated as follows.

#### Given

- The router communication graph  $RCG$
- The deterministic routing algorithm
- The EVC insertion rules

#### Determine

- The set of EVCs to be added

#### Such that

- The power saving is maximized, subject to the EVC insertion rules.

Our algorithm inserts the most beneficial EVC at every iteration and it stops as soon as a pre-set threshold is checked. For low-power NoCs, the pre-set threshold is the minimum energy saving by an EVC path. It is set as zero in our experiments to achieve the maximum energy saving. Also, it can be defined as a non-zero value to prevent the EVC paths with low energy savings from inserting. Besides, other objectives, such as minimizing the average packet latency, can be set to replace the goal of maximizing power saving.

### C. Determination of the Most Beneficial EVC

**Computations of  $a_{i,j}$  and  $b_i$ .** The  $a_{i,j}$  and  $b_i$  are denoted as

$$a_{i,j} = \sum_{c_{p,q} \in C} c_{p,q} \times g(i,j,p,q) \quad (1)$$

$$b_i = \sum_{c_{p,q} \in C} c_{p,q} \times g(i,p,q) \quad (2)$$

where  $g(i,j,p,q)$  is 1 when the routing path from  $r_i$  to  $r_j$  is covered by the routing path from  $r_p$  to  $r_q$ . Likewise,  $g(i,p,q)$  is 1 if  $r_i$  is covered by the routing path from  $r_p$  to  $r_q$ . Otherwise, they are zeros. The computations for them are illustrated in Fig. 3.

**Energy saving model.** The total energy that a flit consumes at a router and its outgoing link is given as [9]:

$$\begin{aligned} E_{flit} &= E_{router} + E_{link} \\ E_{router} &= E_{wrt} + E_{read} + E_{arb} + E_{xb} \end{aligned} \quad (3)$$

where  $E_{wrt}$  and  $E_{read}$  are the energy dissipated by buffer write and read,  $E_{arb}$  is the energy consumed by control logic, including routing computation, VC allocation, switch allocation etc.,  $E_{xb}$  is the energy to traverse the crossbar switch, and  $E_{link}$  is the energy to propagate along the link. Note that energy consumed by links is the same whether or not the traffic uses the EVC path because the total distance of links traversed remains the same. Therefore, only the energy reduction and overhead of the routers is calculated.

Fig. 4 presents microarchitectures for EVC source, sink, and bypass routers [7]. At an EVC source router, control logic is the main overhead, including EVC allocator and EVC table for the remote EVC lanes, and routing logic for EVC packets. We expect energy cost of the added control logic is not big since router energy is datapath-dominated [3, 4]. At an EVC sink router, there is no buffer overhead because the total buffer lanes in the sink input port remain unchanged. They are divided into EVC lanes and normal virtual channel (NVC) lanes. Meanwhile, there is no control logic cost because the packets stored in EVC lanes are processed in the same manner as those stored in NVC lanes. Therefore, there is no energy cost. At a bypass router, there is little additional bypass setup logic and wires when aggressive express pipeline is applied. To simplify the following analysis, we ignore the little energy cost at a bypass router and only take into account the energy cost at an EVC source router.

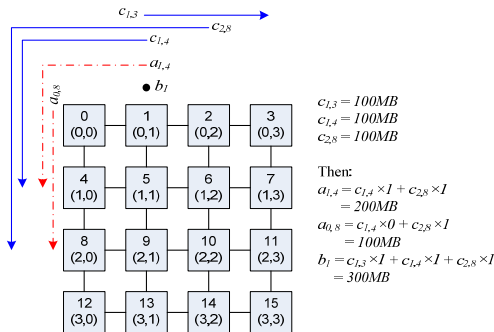


Fig. 3 Illustration of  $a_{i,j}$  and  $b_i$  computations

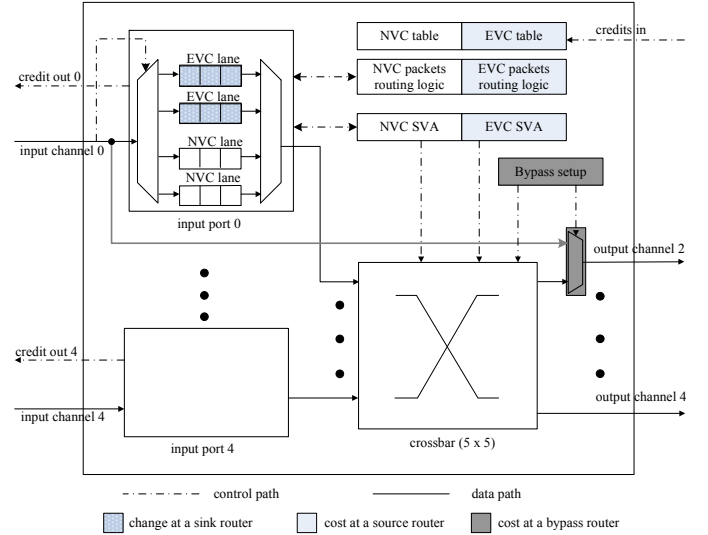


Fig. 4 EVC router microarchitecture

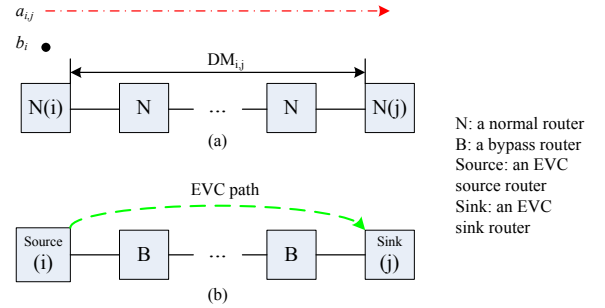


Fig. 5. EVC reduces energy consumption

Fig. 5 illustrates the application of an EVC to a linear array. A regular linear array is shown in Fig.5 (a). The Manhattan distance is  $DM_{i,j}$  and the aggregate communication volume is  $a_{i,j}$ . Router energy consumption to transmit  $a_{i,j}$  from  $r_i$  to  $r_j$  is<sup>5</sup>

$$E_{a1} = a_{i,j} \times DM_{i,j} \times E_{router} \quad (4)$$

A linear array with an EVC path is shown in Fig.5 (b). Energy consumption to traversing  $a_{i,j}$  from  $r_i$  to  $r_j$  is

$$E_{a2} = a_{i,j} \times E_{router} + a_{i,j} \times (DM_{i,j} - 1) \times \alpha \times E_{xb} \quad (5)$$

where the first component is the EVC sink router energy dissipation and the second component is the energy to bypass the  $DM_{i,j}-1$  intermediate routers. Therefore, the energy reduction is

$$\Delta E_a = a_{i,j} \times (DM_{i,j} - 1) \times (1 - \alpha\beta) \times E_{router} \quad (6)$$

On the other hand, the energy cost caused by the EVC router  $r_i$  is

$$\Delta E_b = b_i \times (\lambda - 1) \times E_{router} \quad (7)$$

Totally, the energy saving of this EVC insertion is calculated as:

<sup>5</sup> The energy dissipation to travel the EVC source router  $r_i$  is not included in  $E_a$ . Instead, it is included in  $E_b$ .

$$\Delta E = [a_{i,j} \times (DM_{i,j} - 1) \times (1 - \alpha\beta) - b_i \times (\lambda - 1)] \times E_{router} \quad (8)$$

The equation shows that energy saving is highly related to the aggregate communication volumes  $a_{i,j}$  and  $b_i$ , which highlights the significance to insert EVCs in an application-specific fashion. Also, it shows that a longer EVC path has larger energy reduction. However, the interval for EVC insertions should be carefully selected because a long EVC path occupies many intermediate physical channels.

#### D. EVC Insertion Flow

**Greedy insertion algorithm.** The flow of a greedy insertion algorithm is described in Fig. 6. When no pre-set threshold is hit, the algorithm keeps inserting the most beneficial EVC in the rest EVC paths.

The flow consists of two processes: EVC evaluation and EVC insertion. In the EVC evaluation process, a *RAG* is firstly calculated based on a *RCG* and routing algorithm inputs. Then, EVCs are inserted for all possible pairs of routers. Next, energy saving for each EVC path is computed using the models in section III.C. Meanwhile, an EVC table is generated, with the most beneficial EVC being on the top while the least one being at the bottom.

The EVC paths in the EVC table are then inserted in the EVC insertion process in an iterative way. At each time, the top one in the EVC table is firstly selected. Then, it is checked whether or not this EVC violates any EVC insertion rule. If no violation happens, the information about this EVC is stored in the inserted EVC set. Otherwise, this EVC is removed from the top of the EVC table and the next EVC is selected. This procedure repeats until a pre-set threshold is hit. Once this takes place, output the inserted EVC set.

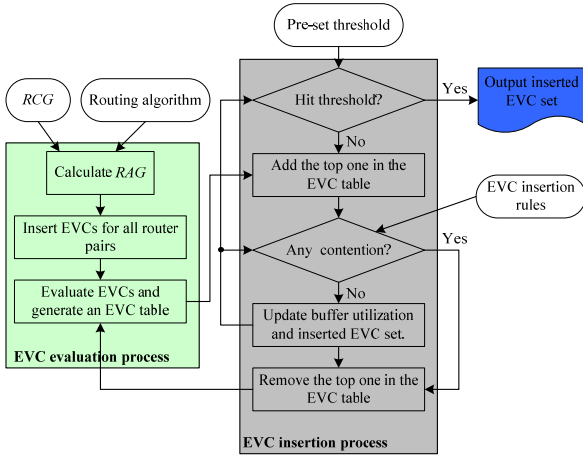


Fig. 6 Greedy insertion algorithm

**EVC insertion rules.** Each EVC insertion has to comply with several rules. Firstly, it can't contend the physical channels which have been already occupied by the previously inserted EVCs. That is to say, no EVC overlapping is allowed. Secondly, a router can have maximum four EVCs, including both EVCs sourcing from it and EVCs sinking at it. These rules reduce the EVC insertion flexibility, and thus result in some good EVCs can't be added. However, they make the

EVC control logic simple to be implemented. Thirdly, an EVC path can't exceed the maximum insertion interval because a long EVC path occupies many physical channels. Although it reduces large energy consumption, it prevents a lot of following EVCs from being inserted. Totally, it always leads to bad results.

## IV. EVALUATION

### A. Experimental Infrastructure

The AS-EVC insertion methodology was evaluated for both synthetic and real traffic loads. We compared the normal mesh network (baseline), the mesh network with static EVC insertions (static EVCs) [7], and the mesh network with AS-EVC insertions for each traffic pattern. We estimated the power savings using proposed high-level models for each topology and traffic pattern as the static EVCs interval changes in the set  $\{2, 3, 4, 5\}$  and found that the maximum power savings are all obtained at 2. Thus, the static EVCs interval is fixed as 2 for all traffics. Similarly, the maximum AS-EVC intervals for various topologies and traffic patterns are determined (Table 2).  $\lambda$  is set as 1.05 and  $\beta$  is defined as 0.25 empirically.

TABLE 2  
The MAXIMUM EVC INTERVALS FOR AS-EVC

Topology	Traffic	Max interval
4x4	uniform	2
	transpose	4
6x6	uniform	2
	transpose	4
10x4	apsi, gzip, swim, parser	4

A baseline router has five ports, four FIFOs per input port, and four-flit deep buffers for each FIFO. A flit is 69-bit wide, consisting of 64-bit payload and 5-bit flit control overhead. Some router microarchitecture optimizations such as look-ahead routing, combined VC and switch allocation were incorporated in the baseline router. If an input port is the sink port of an EVC path, the 4 FIFOs are divided into 2 EVC lanes and 2 NVC lanes. Aggressive express pipeline was used unless otherwise stated.

NoCs supporting EVC flow control was modeled using SystemVerilog. After EVC insertions, the corresponding input ports at EVC sink routers have 2 NVC lanes and 2 EVC lanes. Thereby, the numbers of NVCs at input ports are no longer uniform. Some of them have 2 NVC lanes whereas others have 4 NVC lanes. Our models handle this problem by setting the number of NVCs at each input port as a parameter. All the arbitration logic for NVCs is also controlled by the parameter to reduce control logic redundancy.

Instead of using high-level models for fast energy evaluations, a standard ASIC tool flow was used to provide detailed and accurate power evaluations on EVC insertions, including the following steps. 1) The whole NoCs was synthesized by Synopsys DC. 2) A post-synthesis simulation was run by Synopsys VCS to obtain the switching activity

information of the NoCs. 3) The post-synthesis netlist and the switching activity were fed into Synopsys PTPX to calculate the total power of all routers of the NoCs. Evaluations were performed in post-synthesis stage for two reasons. Firstly, the accuracy is acceptable because we don't need to calculate the power dissipations of the long inter-router links that consume the same power for the three compared architectures. Secondly, the time to do post-layout evaluations for a range of traffic patterns is unacceptable. UMC130nm library with 1.2V power supply voltage was applied. All the simulations run at 250MHz. For a traffic pattern, to ensure the compared three NoCs have nearly the same throughput when their power profiles were obtained, the injection rate was set before any of the compared NoCs enters saturation.

### B. Synthetic Traffic loads

We considered uniform and transpose as synthetic traffics. Uniform traffic assumes randomly distributed destinations. Transpose traffic assumes the destination node for the packets generated by a node is always the symmetric node with respect to the diagonal. Therefore, it achieves the maximum degree of temporal locality.

The average inter-node distance is an important dynamic property of networks because it represents the average number of routers traveled by the packets. It is computed as<sup>6</sup> [6]:

$$\mu = \sum_i \sum_{j \neq i} f_{i,j} (DV_{i,j} + 1) \quad (9)$$

Clearly,  $\mu$  determines the average packet delay without contention, and the power dissipation of routers. A larger reduction of  $\mu$  indicates that larger power reduction may be obtained. Also, it is easy to be computed. Hence, it is a useful metric to estimate the effect of EVCs insertion in the early stage. However, a larger  $\mu$  decrease doesn't definitely mean a higher power saving because it assumes an ideal condition where the power of a router can be entirely removed if it is bypassed and no power overhead is generated by EVCs insertion.

Let's firstly demonstrate the impact of EVCs for a  $4 \times 4$  mesh network (Fig. 7). Compared to the baseline, static EVCs reduces total router power by 6.81% for uniform traffic. This reduction increases to 7.41% when using AS-EVC. For transpose traffic, the power reduction is 8.44% and 23.49% for static EVCs and AS-EVC respectively.

**Compare AS-EVC with static EVCs.** When the traffic changes from uniform to transpose, power decrease by static EVCs shifts a little from 6.81% to 7.41%. However, power reduction by AS-EVC increases significantly from 8.44% to 23.49%. This claims that AS-EVC effectively exploit the characteristics of the highly-specific transpose traffic whereas static EVCs loses a huge optimization room because it considers transpose traffic in the same way as uniform traffic. Normalized average inter-node distance (Fig. 9) supports this finding as well. Reduction of  $\mu$  by static EVCs grows only

<sup>6</sup> Use  $DV_{i,j} + 1$  instead of  $DV_{i,j}$  because we assume that a packet takes one hop to eject out at the sink router.

from 15.68% to 16.73% while it rises significantly from 18.96% to 39.80% when applying AS-EVC. Fig.7 (b) compares the total router power savings of the two methods for the same traffics. AS-EVC outperforms static EVCs by 23.98% for uniform traffic, and 216.99% for transpose traffic.

**Scalability analysis.** To evaluate the scalability of AS-EVC, we investigate a  $6 \times 6$  mesh network under the same two traffics (Fig. 8). As can be seen, AS-EVC continues to show a considerable power gain as compared to the baseline, with the power reduction of 11.01% under uniform traffic and 20.48% under transpose traffic. However, the gain over static EVCs decreases when network size increases. AS-EVC reduces power only 2.45% more than static EVCs for uniform traffic. It implies that static EVCs scheme is enough for large network with randomly distributed load. But, the improvement is still pronounced for transpose traffic where static EVCs saves 17.6mw while AS-EVC reduces 36.5mw, with 107.31% more power saving.

It is interesting to observe that power saved for a  $6 \times 6$  network (20.48%), under transpose traffic, is smaller than a  $4 \times 4$  network (23.49%) although the former (48.31%) obtains a bigger  $\mu$  reduction than the latter (39.80%). It indicates that although the power reduced by bypassing intermediate routers for a  $6 \times 6$  network is bigger than a  $4 \times 4$  network, the power overhead caused by EVC source routers for a  $6 \times 6$  network is bigger than a  $4 \times 4$  network, and the second effect overwhelms the first effect.

**Impact of express pipelines.** Fig. 10 shows the power for a  $4 \times 4$  mesh network when the non-aggressive express pipeline is applied. As expected, compared to the aggressive express pipeline (Fig.7 (a)), less power consumptions are saved for both uniform traffic and transpose traffic. The main reason is that packets flowing through an EVC path have to travel crossbar switches at the intermediate routers for the non-aggressive pipeline.

### C. Real Traffic Loads

The benchmarks in the Minne-SPEC suit [16] were used to evaluate the impact on realistic traffics. Firstly, a benchmark in the Minne-SPEC suit was fed into the TRIPS on-chip network (OCN) simulator to capture an OCN traffic trace. This OCN trace was then applied to a traffic decoder to generate a RCG. OCN is a wormhole routed,  $4 \times 10$  mesh network with YX routing. It serves as an infrastructure to interconnect the two TRIPS processor cores, the individual banks that form the second level cache and the I/O units [17, 18]. We equivalently mapped OCN to a  $10 \times 4$  mesh with XY routing since our AS-EVC algorithm and router models are based on XY routing.

Fig.11 (a) and Fig.11 (b) present the simulation results of Minne-SPEC benchmarks. These graphs follow the same trend as the experiments for synthetic traffics, with AS-EVC clearly outperforming both baseline and static EVCs structures. Power reduction compared to baseline architecture is above 12% for all tested benchmarks, with the most (15.82%, 31.3mw) for *gzip* and the least (12.99%, 30.1mw) for *apsi*. This is a significant improvement because the power

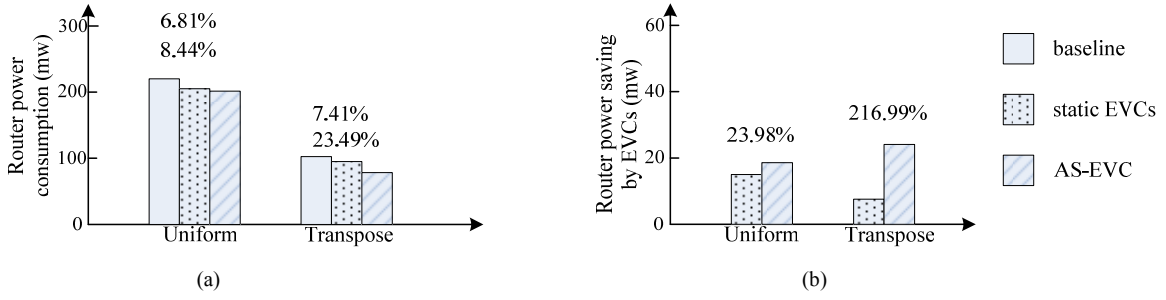


Fig. 7 The entire NoCs power for a 4x4 mesh network

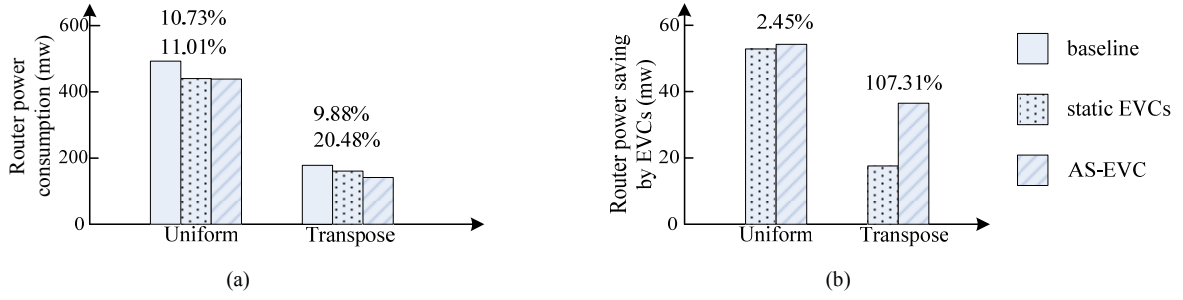


Fig. 8 The entire NoCs power for a 6x6 mesh network

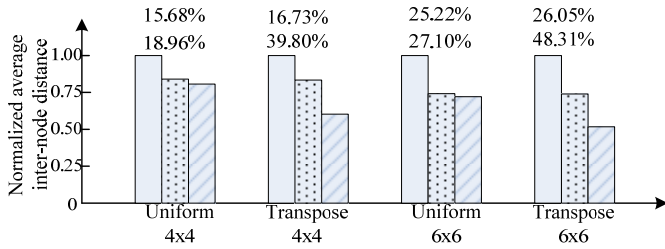


Fig. 9 Normalized  $\mu$  for synthetic traffic

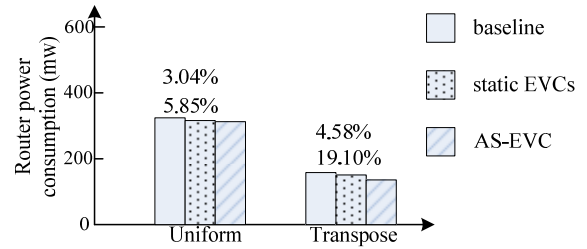


Fig. 10 Power at non-aggressive pipeline

reduction is not over a single router component, but over the total power of all the routers in a NoCs. The gain over static EVCs is bigger than 35% for all traffic, with an average value of 57.14%. The largest improvement is seen for *gzip*. While static EVCs reduces 17.5mw, AS-EVC decreases 31.3mw (78.86% more). AS-EVC obtains the improvement by inserting only 20 EVC paths for *gzip*, 32 less than static EVCs.

#### D. Detailed Power and Area Profiles

To further demonstrate how EVCs technique reduces power, we analyze total standby power, total stream power, and stream powers for different components. Since consistent results have been obtained for the realistic traffic loads, we report only the results for *swim* benchmark.

Total power of NoCs consists of standby power and stream power [4]. NoCs dissipates a lot of standby power even when it is totally idle. It is a fixed cost for a specific architecture. Stream power represents additional power when packets

stream from the source to the sink. More stream power is consumed if more packets are processed.

The principle of EVCs technique is to skip some operations in bypassing routers as packets flow along the routing path. In other words, EVCs technique can only save power for streaming packets. No power can be reduced by EVCs if no packets are routing. It increases standby power because it requires some extra logic. As seen in Fig.12, standby power overhead is 1.50% for AS-EVC. It increases to 3.22% for static EVCs where more EVC paths are inserted. But, stream power is reduced by 20.09% and 12.43% for AS-EVC and static EVCs respectively.

The number of logic gates of the entire NoCs for *swim* traffic goes up from 1683.13K gates to 1720.74K, which only increases 2.23%. A single baseline router has 47604 logic gates. The area of EVC source routers with different number of source paths and EVC bypass routers with various number of bypass paths is summarized in Table 3. It shows that adding four source paths only increases area by 7.89% and area cost of adding bypass paths is even smaller. This is



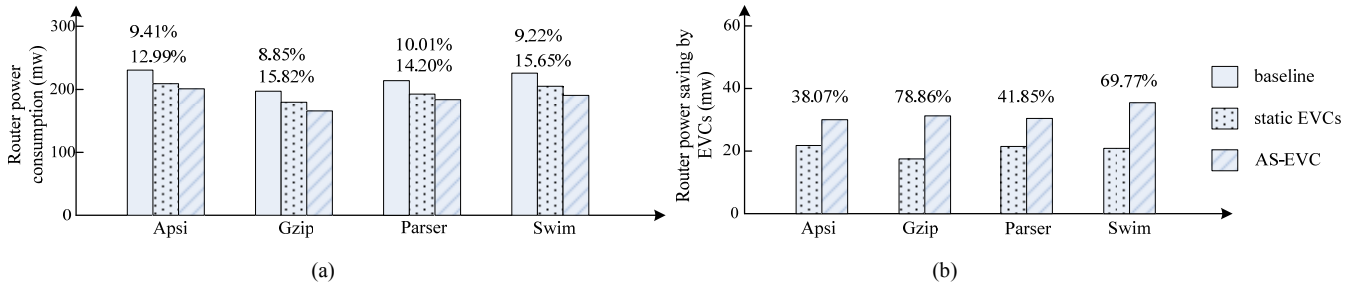


Fig. 11 The entire NoCs power for Minne-SPEC benchmarks

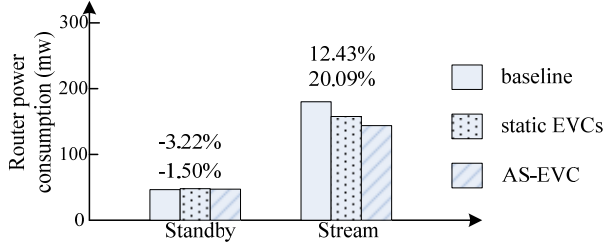


Fig. 12 Power profile for the swim traffic

significantly smaller than the technique that requires adding ports for routers. Thus, EVCs technique is highly scalable in terms of area.

## V. CONCLUSIONS

We have proposed a novel methodology to insert application-specific EVCs for low-power NoCs in this paper. A *RAG* is firstly defined to help designers to clearly know the communication characteristics of an application. Then, the simple power reduction model and power overhead model are presented to calculate power savings for all possible EVC paths. Finally, a greedy algorithm is applied to add EVC paths in an iterative way, subjecting to some insertion rules. In a word, for an application, AS-EVC is able to quickly insert appropriate EVCs early in the design stage.

NoCs supporting EVCs technique is modeled and evaluated in a low-level fashion to provide detailed and accurate power results. Experiments on both synthetic and realistic workloads show that AS-EVC achieves great power improvements over the entire NoCs compared to both baseline and static EVCs.

We plan to extend the present work in several directions. A NoCs design equipped with EVCs is currently being placed and routed by Synopsys Astro. Our ultimate goal is a silicon implementation. Then, based on power results provided by post-layout simulations, more accurate power reduction model when a flit bypassing a router and power cost model when a flit traveling an EVC source router are expected to be built. They help to compute the power saving for each EVC path more precisely. Also, some EVC insertion rules will be removed to increase insertion flexibility. For example, EVC paths are allowed to be overlapped in some cases. With these two improvements, more optimized EVCs paths can be inserted. In addition, more metrics like energy dissipation per

TABLE 3  
AREA OF SOURCE AND BYPASS ROUTERS

Source router	Gate count	Bypass router	Gate count
1-EVC	49295 (3.55%)	1-bypass	47781 (0.37%)
2-EVC	50179 (5.41%)	2-bypass	47960 (0.75%)
3-EVC	51092 (7.33%)	3-bypass	48140 (1.13%)
4-EVC	51359 (7.89%)	4-bypass	48347 (1.56%)

bit, network throughput, and average packet latency, will be used to obtain a more complete evaluation for AS-EVC.

Another future direction is to combine application-specific EVCs with application-specific long-range physical links together. Inserting EVCs obtains latency, throughput, and energy gains with very low cost. But, as a flow-control technique, EVCs benefits global packets at the cost of increasing the contention delay of packets those are locally buffered in bypassing routers due to the shared buffers, crossbar and physical link. On the other hand, inserting long physical links reduces latency for global packets without blocking the packets traveling the intermediate routers. But, the power and area costs by fatter routers are high. Thus, it is very interesting to explore a mid-way between them to exploit the best of both techniques.

## ACKNOWLEDGMENT

The authors would like to thank Paul Gratz of the U. T. Austin TRIPS Team for supplying the *OCN* traffic traces of Minne-SPEC benchmarks and a decoder to process them.

## REFERENCES

- [1] "ITRS, International Technology Roadmap for Semiconductors," 2007.
- [2] S. Heo and K. Asanovic, "Replacing global wires with an on-chip network: a power analysis," in *International Symposium on Low Power Electronics and Design*, 2005, pp. 369-374.
- [3] H. S. Wang, L. S. Peh, and S. Malik, "Power-driven design of router microarchitectures in on-chip networks," in *International Symposium on Microarchitecture*, 2003, pp. 105-116.
- [4] A. Banerjee, R. Mullins, and S. Moore, "A power and energy exploration of network-on-chip architectures," in *International Symposium on Networks-on-Chip*, 2007, pp. 163-172.
- [5] A. Pullini et al., "NoC design and implementation in 65nm technology," in *International Symposium on Networks-on-Chip*, 2007, pp. 273-282.

- [6] U. Y. Ogras and R. Marculescu, "It's a small world after all: NoC performance optimization via long-range link insertion," *IEEE Transaction on Very Large Scale Integration Systems*, Vol. 14, 2006, pp. 693-706.
- [7] A. Kumar, L. S. Peh, P. Kundu, and N. K. Jha, "Express virtual channels: Towards the ideal interconnection fabric," in *International Symposium on Computer Architecture (and IEEE Micro Top Picks 2008)*, 2007, pp. 150-161.
- [8] J. Xu, W. Wolf, J. Henkel, and S. Chakradhar, "H.264 HDTV decoder using application-specific networks-on-chip," in *International Conference on Multimedia and Expo*, 2005, pp. 1508-1511.
- [9] H. S. Wang, X. Zhu, L. S. Peh, and S. Malik, "Orion: A power-performance simulator for interconnection networks," in *International Symposium on Microarchitecture*, 2002, pp. 294-305.
- [10] W. J. Dally, "Express cubes: Improving the performance of k-ary n-cube interconnection networks," *IEEE Transaction on Computers*, Vol. 40, No. 9, 1991, pp. 1016-1023.
- [11] A. Kumar, L. S. Peh, and N. K. Jha, "Token flow control," in *International Symposium on Microarchitecture*, 2008.
- [12] P. T. Huang, W. L. Fang, Y. L. Wang, and W. Hwang, "Low power and reliable interconnection with self-corrected green coding scheme for network-on-chip," in *International Symposium on Networks-on-Chip*, 2008, pp. 77-83.
- [13] M. Palesi et al., "Design of bandwidth aware and congestion avoiding efficient routing algorithms for networks-on-chip platforms," in *International Symposium on Networks-on-Chip*, 2008, pp. 97-106.
- [14] T. H. Huang, U. Y. Ogras, and R. Marculescu, "Virtual channels planning for networks-on-chip," in *the 8th International Symposium on Quality Electronic Design*, 2007, pp. 879-884.
- [15] H. Matsutani, M. Koibuchi, D. Wang, and H. Amano, "Adding slow-silent virtual channels for low-power on-chip networks," in *International Symposium on Networks-on-Chip*, 2008, pp. 23-32.
- [16] A. J. Kleinosowski and D. J. Lilja, "MinneSPEC: A new SPEC benchmark workload for simulation-based computer architecture research," *Computer Architecture Letters*, Vol. 1, 2002.
- [17] P. Gratz et al., "Implementation and evaluation of on-chip network architectures," in *International Conference on Computer Design*, 2006, pp. 477-484.
- [18] P. Gratz et al., "On-chip interconnection networks of the TRIPS chip," *IEEE Micro*, Vol. 27, 2007, pp. 41-50.