

# HVCRouter: Energy Efficient Network-on-Chip Router with Heterogeneous Virtual Channels

Ji Wu, Xiangke Liao, Dezun Dong<sup>(✉)</sup>, Li Wang<sup>(✉)</sup>, and Cunlu Li

College of Computer, National University of Defense Technology,  
Changsha 410073, China  
{dong,liwang}@nudt.edu.cn

**Abstract.** The high scalability of the NoC (network-on-chip) makes it one of the best choices to meet the demand for bandwidth increasing in systems-on-chips and chip multiprocessors. However, the NoC is increasingly becoming power-constrained. A significant part of the NoC's power is consumed in the router buffer. In this paper, we propose HVCRouter, a novel NoC router design with heterogeneous virtual channels. In particular, HVCRouter incorporates a bufferless channel to respect its power efficiency at low network load. HVCRouter employs a fine-grained power gating algorithm which exploits power saving opportunities at both channel and buffer levels simultaneously, and is able to achieve high power efficiency without degrading performance at varying network utilization. Our experimental results on both synthetic and real workloads show that HVCRouter delivers similar performance with FlexiBuffer, the best in the literature. More importantly, HVCRouter consumes an average of 22.797% less power, and results in 20.698% lower EDP (energy delay product) than FlexiBuffer.

**Keywords:** Network-on-Chip router · Virtual channel · Power gating

## 1 Introduction

The NoC (network-on-chip) is widely considered as a first-order component of current and future multicore and manycore CMPs, due to its high flexibility and scalability to be able to effectively address the rapid increasing of core count. Unfortunately, NoCs are concerned about their excessive power consumption. For example, for Intel Terascale 80-core chip, its NoC consumes 28% [1] of the chip power; For MIT RAW, it is up to 36% [2,3]; In the future, NoCs in many-core processors are estimated to consume hundreds of watts of power [4] if current network implementation is naively scaled. One of the key components of an NoC router is the buffer, which is necessary to provide high performance for an NoC. However, the buffer consumes significant power (up to 30% – 40% of NoC power [5]). Recent studies have explored several optimizations to reduce the power consumption of router buffers. Bufferless routing [6] presents a new algorithm for routing without using buffers in router input/output ports. Bufferless routing is proved to be very effective in power saving at low network load,

however, because of low bandwidth and detouring, it will incur a significant performance penalty at higher network loads. Flexibuffer [7] performs power gating on buffers to reduce power consumption. Although demonstrated obvious advantages in router power saving, FlexiBuffer still incurs considerable leakage power, especially at low network load, which happens on many real-world applications for much of the time. The deficiencies of FlexiBuffer are detailed in Sect. 2.

In this paper, we propose HVCRouter, a novel NoC router design with heterogeneous virtual channels (VCs). To efficiently reduce power, HVCRouter integrates one bufferless channel to leverage its power efficiency. In addition, HVCRouter allows to power-gating not only buffers, but also channels to capture more power optimization opportunities. However, these optimizations are not achieved without a challenge. To make good use of the heterogeneous VCs, and power-gating channels as well as buffers without degrading performance at varying network loads, the channel allocation and flow control mechanisms in conventional NoC routers must be modified and carefully orchestrated as detailed in Sect. 3.

In this paper, we make the following contributions:

- We introduce a novel NoC router design with heterogeneous virtual channels. In particular, it introduces a bufferless channel to respect its power efficiency at low network utilization.
- We present a fine-grained power gating algorithm, which enables HVCRouter to adapt well to varying network loads, and achieve excellent power efficiency. To the best of our knowledge, our approach is the first to exploit power saving opportunities on NoC router at buffer-level and channel-level simultaneously.
- Our experiments show that HVCRouter achieves similar performance with FlexiBuffer, the best in the literature, consumes an average of 22.797% less power, and provides 20.698% lower EDP than FlexiBuffer.

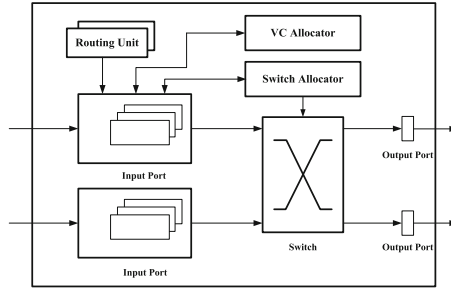
The rest of the paper is organized as follows. For background information, Sect. 2 introduces the architecture of a modern NoC router, discusses bufferless routing and Flexibuffer, and motivates our work. Section 3 describes the architecture of HVCRouter. In Sect. 4, we present our approach for power gating and VC allocation. Section 5 evaluates our work. Section 6 discusses related work. In Sect. 7, we summarize and conclude the paper.

## 2 Background and Motivation

### 2.1 Router Architecture

The architecture of a typical modern virtual-channel NoC router is shown in Fig. 1. When a flit of a packet arrives at a router from one of the input virtual channels in the router’s input port, the flit is hold in a buffer of that channel until it can be forwarded. The process on a flit is divided into several pipeline stages. First, route computation (RC) is performed by the routing unit to decide the output port. Second, virtual-channel allocation (VA) by the VC allocator

allocates an available output virtual channel in the given output port. If all output channels are occupied, the flit is kept in the buffer and waits until there is one vacant. Third, switch allocation (SA) by the switch allocator schedules a time slot on the switch and the output channel, and forwards the flit to routed output port during this time slot. Finally, switch traversal (ST) forwards the flit to depart the current router and travel to the next router in its routing path.

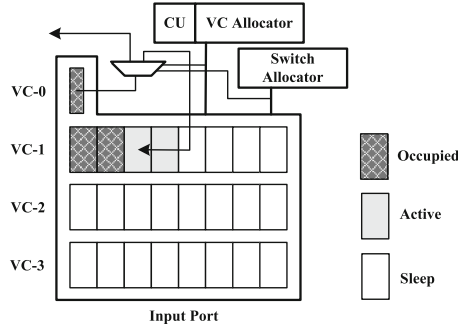


**Fig. 1.** The router architecture.

The buffer serves as one of the most important components in a modern NoC router, since it decouples the allocation of channel resources. Buffers within each router improve the bandwidth efficiency in the network because they enable a flit to wait until its allocated output channel and switch ready, otherwise the flit has to be dropped or misrouted, namely, sent to a less desirable destination port, thereby buffers reduce the number of dropped or misrouted packets. It is proved that more buffers result in significantly higher performance. However, buffers in the NoC occupy a significant portion of the power. On the other hand, Kim et al. [7] made the observation that even if the network is saturated, not all of the buffer resources are fully utilized.

## 2.2 Bufferless Routing and Flexibuffer

Recent studies have explored several optimizations to reduce the power consumption of router buffers. bufferless routing [6] presents a new algorithm for routing without using buffers in router input/output ports. When multiple flits are routed to the same output channel, an arbitration is performed as usual to choose one to get the channel. For remaining flits, bufferless routing misroutes them to other output channels and guarantees those flits will reach their destination at last. In contrast to misrouting, Mitchell et al [8] proposes another bufferless router design by dropping those losing arbitration flits, which are later retransmitted. Bufferless routing is proved to be very power efficient at low network load, however, bufferless routing suffers from poor performance and energy at higher loads because the misrouting/dropping caused by link contention leads to increased link utilization, which creates a positive feedback cycle because



**Fig. 2.** The architecture of HVCRouter.

increased link utilization further increases link contention. Consequently, bufferless networks will incur a significant performance penalty at higher network loads, and saturate at lower throughputs than buffered networks.

In contrast to eliminating buffers from routers, FlexiBuffer employs a power gating policy and adjusts the size of the active buffers adaptively. Although it demonstrates obvious advantages in terms of performance under medium-to-high network loads over bufferless routing, according to our observation, there remain optimization opportunities to be exploited. First, FlexiBuffer keeps all of the virtual channels active at any time, which will incur significant static power when network is lightly utilized. Second, FlexiBuffer includes buffers in each channel, controls them separately, and keeps some buffers active at any time for each channel, this also introduces considerable static power, we are thereby motivated to propose a router design with heterogeneous virtual channels, and eliminate buffers from one channel to reduce static power. In addition, instead of considering channels separately, our design manages channels and buffers in a global framework to allow adaptively power-gating channels as well as their buffers, if desired, to achieve higher power efficiency, without degrading performance.

### 3 HVCRouter Architecture

The architecture of HVCRouter is similar to the conventional design shown in Fig. 1, with the major differences in the input port and the VC allocator as highlighted in Fig. 2. HVCRouter employs a mixture of buffered and bufferless VCs, among the multiple VCs in an input port, the lowest-order one, i.e.,  $VC-0$  is without buffers. The output of  $VC-0$  is connected not only to the switch as usual, but also to buffers of  $VC-1$  by a demultiplexer. The output selecting of the demultiplexer could be controlled by both the VC allocator and the switch allocator. When a flit coming from bufferless VC ( $VC-0$ ) does not win the output VC arbitration during VC allocation, the VC allocator will send a signal to corresponding demultiplexer to forward the flit into the tail of buffers of  $VC-1$  to enable it to wait for next round of VC allocation. Similar process happens

on switch allocation, if a flit in bufferless VC loses the arbitration, the switch allocator will forward it into the buffer of  $VC-1$  as well. This is different with the process of bufferless routing. In bufferless routing, those flits have to be dropped or misrouted, which will significantly increase the latency. This flit-inner-forwarding design will not increase notable power consumption because  $VC-0$  only serves at low network load, in which case, competition and arbitration rarely happen. If the load increases, HVCRouter will rely on buffered channels to service packet flow instead, and power gate  $VC-0$  as detailed in Sect. 4. To solve the problem of out-of-order arriving, HVCRouter receive flits in a receiver-side buffer until all flits of a packet have arrived. We adopt a bufferless channel instead of leveraging power gating to disable all buffer entries in a channel at an input port, which is based on the observation that many real-world applications have low network utilization for much of the time [6], bufferless channel suffices in those cases, and buffers' large area and high power consumption could be saved. For a buffered channel even with power-pating, transitions between power states during power-gating will increase power consumption. Besides the introduction of a bufferless VC as well as a demultiplexer logic, another notable difference for HVCRouter is the VC allocator, which integrates a control unit (CU) to be able to dynamically and adaptively turn on/off virtual channels and buffers.

## 4 Power-Gating and VC Allocation

### 4.1 Power-Gating Algorithm

The power-gating algorithm of HVCRouter introduces a state machine (as depicted in Fig. 3) with two states, characterized as follows,

- *state-0*: only  $VC-0$  and  $VC-1$  are turned on.
- *state-1*:  $VC-0$  is turned off,  $VC-1$  is turned on, other VCs may or may not be turned on.

Let us examine the power gating algorithm shown in Algorithm 1. At first, HVCRouter works at *state-0* with only  $VC-0$  and  $VC-1$  (with  $p$  active buffer entries,  $p$  is explained later) are turned on, to make use of the bufferless channel

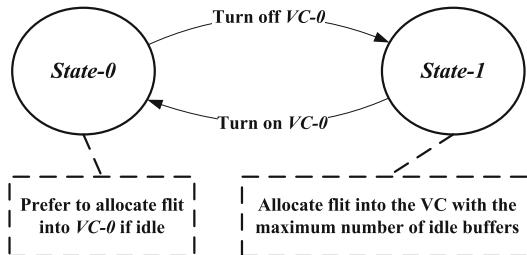


Fig. 3. The state machine of HVCRouter.

**Algorithm 1.** Power gating algorithm.

---

```

1: procedure pg_hvrouter
2: // state-0: VC-0 and VC-1 are active
3: if current_state == 0 then
4:   if vc_occupancy(1)  $\geq$  Q then
5:     turn_on_vc(2, p)
6:     power_gate_vc(0)
7:     current_state = 1
8:     return
9:   end if
10: end if
11: // state-1: VC-0 is power gated
12: if current_state == 1 then
13:   for i from 1 to num_vcs - 2 do
14:     if vc_occupancy(i)  $\geq$  Q &&vc_power_gated(i + 1) == TRUE then
15:       turn_on_vc(i + 1, P)
16:     end if
17:   end for
18:   for i from num_vcs - 1 to 2 do
19:     if vc_occupancy(i) == 0 &&vc_occupancy(i - 1) < Q then
20:       power_gate_vc(i)
21:     end if
22:   end for
23:   manage_router_buffers()
24:   if vc_occupancy(1) < Q then
25:     for i = 2; i < num_vcs; i++ do
26:       if vc_power_gated(i) == FALSE then
27:         break
28:       end if
29:     end for
30:     if i == num_vcs then
31:       turn_on_vc(0, 0)
32:       current_state = 0
33:     end if
34:   end if
35: end if

```

---

to reduce the static power to a minimum at low network utilization. In Line 4,  $vc\_occupancy(no)$  calculates the number of busy buffers (with packet flit(s) in it) in  $VC-no$ , to estimate the load. If the load of  $VC-1$  beyond a threshold, in Line 5,  $turn\_on\_vc(2, p)$  will turn on  $VC-2$  as well as its  $p$  buffer entries, here  $p$  represents the minimum number of active buffer entries needed to prevent stalls caused by the lack of available buffer entries,  $p = \max(t_w, t_{crt})$ , where  $t_w$  is the wake up delay of a buffer entry, and  $t_{crt}$  is the credit round-trip latency [7,9]. In Line 6,  $VC-0$  is power gated. As a result, the state is transited to  $state-1$  in Line 7.

---

**Algorithm 2.** Router buffer management.

---

```

1: procedure manage_router_buffers
2: for  $i$  from 1 to  $num\_vcs - 1$  do
3:   if  $vc\_power\_gated(i) == \text{TRUE}$  then
4:     continue
5:   end if
6:   if  $vc\_idle\_buffer\_num(i) \leq P - Q$  && $vc\_active\_buffer\_num(i) < \text{BUFFER\_SIZE}$ 
   then
7:      $turn\_on\_buffer(i, 1)$ 
8:   end if
9:   if  $current\_cycle() \% T == 0$  then
10:    if  $vc\_idle\_buffer\_num(i) > P - Q$  && $vc\_active\_buffer\_num(i) > P$  then
11:       $power\_gate\_buffer(i, 1)$ 
12:    end if
13:  end if
14: end for

```

---

In *state-1*, in Lines 13 – 17, the algorithm checks if some channel has encountered congestion, and turns on a higher order channel to mitigate the congestion. In Lines 18 – 22, on the contrary, the algorithm checks if some channel is idle, and its immediately lower order channel is also lightly utilized, in that case, the idle channel is power gated. In Line 23, the algorithm invokes `manage_router_buffers()` to perform fine-grained power gating on router buffers. As shown in Algorithm 2, for each buffered and active virtual channel, if current number of active-but-idle buffer entries is no more than  $P - Q$ , and there exists buffer(s) in sleep state, then one more buffer is waked up. On the contrary, if current number of active-but-idle buffer entries is more than  $P - Q$ , and number of active buffer entries is more than  $P$ , then one buffer is power gated. In contrast to turn on buffers, power-gating is executed less frequently (per  $T$  consecutive cycles) to avoid thrashing and reduce control overhead. Finally, in Lines 24 – 34 in Algorithm 1, if all channels except  $VC-1$  are power gated, and  $VC-1$  has a low utilization, then  $VC-0$  is turned on with the state transited back to *state-0*.

As the algorithm shown, when load increases, it tends to turn on more channels first, rather than more buffers in an already active channel, because more virtual channels inside a single physical channel allow other flits to use the channel bandwidth that would otherwise be left idle when a flit blocks [9], which will improve the performance.

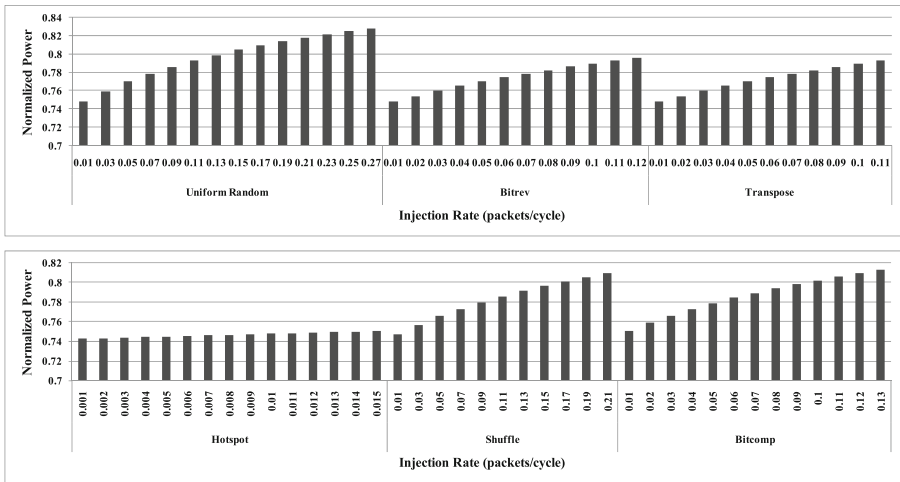
## 4.2 VC Allocation

This section discusses the VC allocation policy of HVCRouter. The VC allocator allocates output channel for a flit according to the state of the corresponding input port of the downstream router. Suppose a flit in router  $A$  is waiting for VC allocation, with its routed output port being  $P_{A\_OUT}$ , and the downstream router is  $B$  with corresponding input port being  $P_{B\_IN}$ . If the state of  $P_{B\_IN}$  is *state-0*, which implies only  $VC-0$  and  $VC-1$  are active in  $P_{B\_IN}$ , then the VC allocator

**Table 1.** Network configuration.

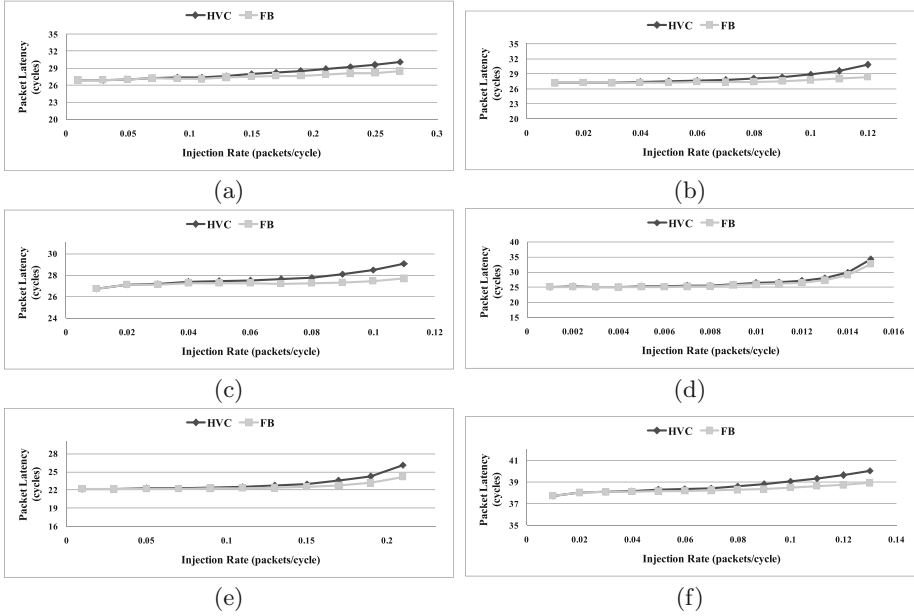
Node count	64
Topology	8*8 2D MESH
VC count	4VCs per port
Buffer depth	8 flit per VC
Flit length	16 byte
Switch allocator	islip
Routing algorithm	Dimension-order
P (parameter in Algorithm 1)	3
Q (parameter in Algorithm 1)	2
T (parameter in Algorithm 2)	3

of  $A$  will check if  $VC-0$  in  $P_{B\_IN}$  is idle ( $A$  is able to know this by conventional credit based backpressured flow control [9]), and allocate the flit into  $VC-0$  in  $P_{A\_OUT}$ , otherwise, allocate it into  $VC-1$  in  $P_{A\_OUT}$ . To support power-gating router buffers, HVCRouter makes the modification to conventional credit-based flow control similar to that of FlexiBuffer. However, due to the heterogenous nature of HVCRouter, the credit count of  $VC-0$  is initialized to 1, that of  $VC-1$  is initialized to  $p - 1$  (one may be stolen by  $VC-0$ ), and the others are initialized to  $p$ . If the state of  $P_{B\_IN}$  is *state-1*, and supposes the input channel with the maximum number of idle buffers, i.e., maximum credits, is  $VC-m$ , then the VC allocator in  $A$  will allocate the flit into  $VC-m$  in  $P_{A\_OUT}$ .



**Fig. 4.** Power of HVCRouter, normalized to that of FlexiBuffer, for six traffic patterns.





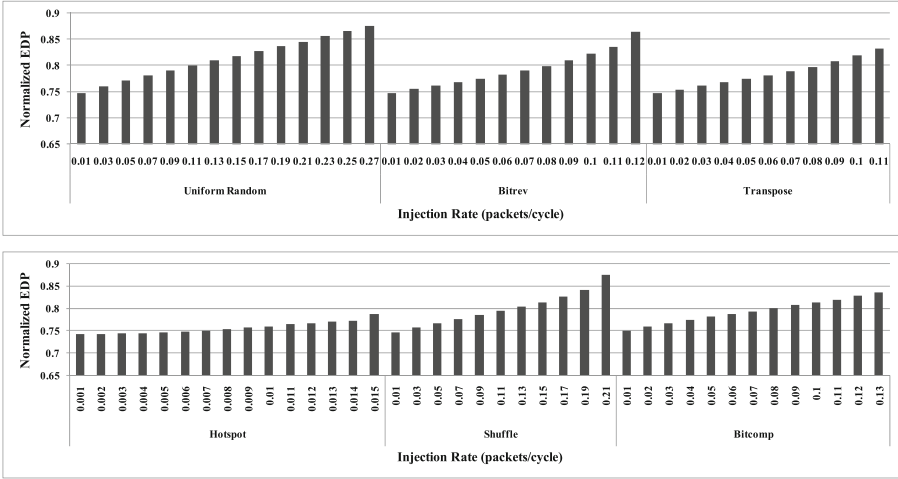
**Fig. 5.** Performance of HVCRouter and FlexiBuffer for (a) Uniform random (b) Bit reverse (c) Transpose (d) Hot-spot (e) Shuffle and (f) Bit complement traffic patterns.

## 5 Evaluation

We evaluate HVCRouter in terms of power, network latency and energy delay product by using BookSim [10] as well as DSENT [11]. Booksim is a cycle-accurate interconnection network simulator to measure network latency, and we revise it to model HVCRouter. Using the results obtained by Booksim as well as the network configuration files as input, DSENT will output the power values. The detailed network configurations used in our evaluation are listed in Table 1.

### 5.1 Synthetic Traffic Evaluation

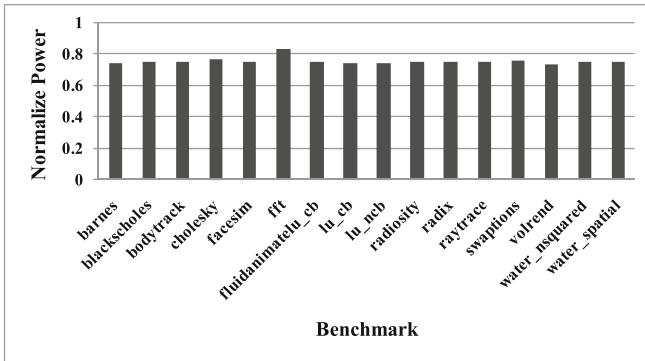
Figure 4 shows the power consumption of HVCRouter for six synthetic traffic patterns, normalized to that of FlexiBuffer. It is observed that HVCRouter consumes lower power than FlexiBuffer for all the patterns under various packet injection rates. Let us examine the results in detail. All the curves demonstrate similar trend. At low injection rates, HVCRouter consumes significantly lower power than FlexiBuffer, more specifically, a maximum of 25.75% lower for *Hot-spot*. When network is lightly utilized, HVCRouter utilizes bufferless virtual channel to reduce both static and dynamic power by scheduling packets into bufferless channel whenever possible, and power gating other channels as well as buffers. As the injection rate increases, HVCRouter adaptively turns on channels and buffers to mitigate the link contention of bufferless channel. Attributing



**Fig. 6.** The energy delay product (EDP) of HVCRouter for six traffic patterns, normalized to that of FlexiBuffer.

to the finer-grained power gating at both channel and buffer level, HVCRouter remains lower power consumption than FlexiBuffer, with an average of 22.367 % less power compared to FlexiBuffer.

The performance of HVCRouter is evaluated by measuring the packet latency. Figure 5 presents the average packet latency achieved on HVCRouter and FlexiBuffer for six traffic patterns. As the results shown, HVCRouter achieves similar performance with FlexiBuffer, with an average of 1.704 % larger packet latency than that of FlexiBuffer. The results prove that HVCRouter makes good use of the heterogeneous architecture, reacts quickly to congestion.



**Fig. 7.** Power of HVCRouter for real workload traces, normalized to that of FlexiBuffer.

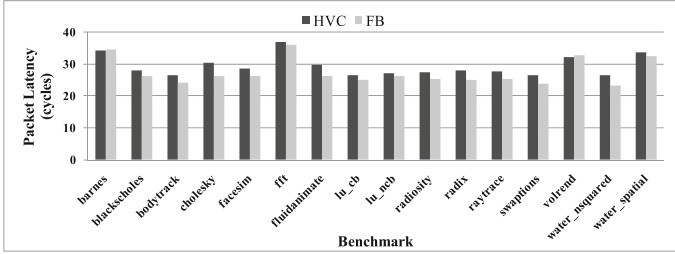


Fig. 8. Performance of HVCRouter and FlexiBuffer for real workload traces.

Figure 6 reports the energy delay product (EDP) of HVCRouter for six traffic patterns, normalized to that of FlexiBuffer. Overall, HVCRouter results in an average of 21.08% lower EDP for all the patterns than FlexiBuffer.

### 5.2 Real Workload Evaluation

We also use the traffic generated by SynFull [12] for BookSim to simulate real-world workloads. SynFull introduces a synthetic traffic generation methodology that captures both application and cache coherence behavior to evaluate NoCs. Using a real-world benchmark as the input, SynFull is able to output a traffic, which could be further fed into BookSim for simulation. Currently, the traffics made publicly available by SynFull are a set of 16 traffics produced from PARSEC [13] and SPLASH-2 [14] benchmarks with the sim-small input set for 16 cores. The power and packet latency are measured with the results shown in Figs. 7 and 8, respectively. The results demonstrate that HVCRouter consumes an average of 24.83% less power on all of the benchmarks than FlexiBuffer.

Regarding the network performance, overall, HVCRouter delivers similar performance than FlexiBuffer, with an average of 7.53% larger latency than that

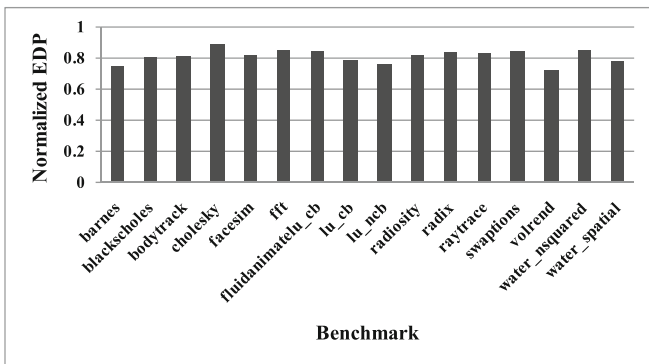


Fig. 9. The energy delay product (EDP) of HVCRouter for real workload traces, normalized to that of FlexiBuffer.

of FlexiBuffer. Figure 9 reports the energy delay product (EDP) of HVCRouter, normalized to that of FlexiBuffer. It is observed that HVCRouter provides an average of 19.176% lower EDP than FlexiBuffer on all the workloads.

## 6 Related Work

Recent studies have explored some NoC power optimizations at various granularities. First, at channel and switch level, Michelogiannakis et al. [15] propose adaptive bandwidth networks (ABNs) to divide channels and switches into lanes to reduce power consumption of NoC. Second, at router level, NoRD [16] provides a power-gating bypass to decouple the node's ability for transferring packets from the powered-on/off status of the associated router, thereby increases the length of idle periods, and eliminates node-network disconnection problem. Panthre [17] adopts topology and routing reconfiguration to steer away the packets that would normally use power-gated components, to provide long intervals of uninterrupted sleep to selected units. Matsutani et al. [18] adopt look-ahead routing to hide the wake-up delay and reduce the short-term sleeps of channels. Chen et al. [19] propose a performance-aware, power reduction scheme that aims to achieve non blocking power-gating of on-chip network routers through looking ahead routing. Router Parking [20] selectively power gates routers attached to parked cores dynamically, and adopts adaptive routing to ensure the network performance. In addition, some topology dependent approaches are also proposed. Yue et al. [21] present Smart Butterfly, a core-state-aware NoC power-gating scheme based on flattened butterfly topology that utilizes the active/sleep state information of processing cores to improve power-gating effectiveness. Since Clos network has multiple alternative paths for every packet, Chen et al. [22] propose power-gating scheme MP3, which is able to achieve minimal performance penalty and save more static energy than conventional Clos network. Third, at the granularity of subnet, Balfour et al. [23] present a concentrated mesh topology with replicated sub-networks to improve area and energy efficiency in NoC. Das.R et al. [24] propose the Catnap architecture which performs power-gating on subnets in a multi-layer NoC. Mishra et al. [25] introduce two separate networks on chip, where one is optimized for bandwidth and the other for latency, and the steering of applications to the appropriate network. darkNoC [26] integrates multiple layers of architecturally identical, but physically different routers, leveraging the extra transistors available due to dark silicon. Each layer is separately optimized for a particular voltage-frequency range. Finally, at router buffer level, Moscibroda et al. [6] found that a bufferless router consumes a very low leakage power compared to that of a traditional buffered router, and they present a bufferless NoC design and a new routing algorithm. However, bufferless NoC is only applicable at low network load. Chris Fallin et al. [27] propose the minimally-buffered deflection (MinBD) router, which combines deflection routing in bufferless network with a small side buffer to reduce deflections, which improves bufferless routing to some extent. FlexiBuffer [7] reduces buffer leakage power by using fine-grained power gating and adjusting the size of the active buffers adaptively.

## 7 Conclusion

The router buffer makes a significant contribution to the overall NoC power. We discuss HVCRouter, a novel NoC router architecture which couples buffered and bufferless virtual channels. Employing a fine-grained power gating policy, HVCRouter consumes an average of 22.797% less power than FlexiBuffer, the state of the art power efficient NoC router design. In terms of performance, HVCRouter matches FlexiBuffer on all the benchmarks.

**Acknowledgments.** We thank the anonymous reviewers for their precious feedback. We gratefully acknowledge members of Tianhe interconnect group at NUDT for many inspiring conversations early in the project. The work was partially supported by 863 Program under Grant No. 2012AA01A301, NSFC under Grant No. 61370018, 61272482, and FANEDD under Grant No. 201450.

## References

1. Hoskote, Y., Vangal, S., Singh, A., Borkar, N., Borkar, S.: A 5-GHz mesh interconnect for a teraflops processor. *IEEE Micro* **27**(5), 51–61 (2007)
2. Taylor, M.B., Kim, J., Miller, J., Wentzlaff, D., Ghodrati, F., Greenwald, B., Hoffman, H., Johnson, P., Lee, J.-W., Lee, W., Ma, A., Saraf, A., Seneski, M., Shnidman, N., Strumpfen, V., Frank, M., Amarasinghe, S., Agarwal, A.: The raw microprocessor: a computational fabric for software circuits and general-purpose programs. *IEEE Micro* **22**(2), 25–35 (2002)
3. Kim, J.S., Taylor, M.B., Miller, J., Wentzlaff, D.: Energy characterization of a tiled architecture processor with on-chip networks. In: *Proceedings of the 2003 International Symposium on Low Power Electronics and Design*, ser. ISLPED 2003, pp. 424–427. ACM, New York, NY, USA (2003)
4. Borkar, S.: Thousand core chips: a technology perspective. In: *Proceedings of the 44th Annual Design Automation Conference*, ser. DAC 2007, pp. 746–749. ACM, New York, NY, USA (2007)
5. Jafri, S.A.R., Hong, Y.-J., Thottethodi, M., Vijaykumar, T.N.: Adaptive flow control for robust performance and energy. In: *Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '10, pp. 433–444. IEEE Computer Society, Washington, DC, USA (2010)
6. Moscibroda, T., Mutlu, O.: A case for bufferless routing in on-chip networks. In: *Proceedings of the 36th Annual International Symposium on Computer Architecture*, ser. ISCA 2009, pp. 196–207. ACM, New York, NY, USA (2009)
7. Kim, G., Kim, J., Yoo, S.: Flexibuffer: reducing leakage power in on-chip network routers. In: *Proceedings of the 48th Design Automation Conference*, ser. DAC 2011, pp. 936–941. ACM, New York, USA (2011)
8. Hayenga, M., Jeger, N.E., Lipasti, M.: Scarab: a single cycle adaptive routing and bufferless network. In: *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 42, pp. 244–254. ACM, New York, NY, USA (2009)
9. Dally, W.J., Towles, B.: *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, San Francisco (2004)

10. Jiang, N., Becker, D.U., Michelogiannakis, G., Balfour, J., Towles, B., Kim, J., Dally, W.J.: A detailed and flexible cycle-accurate network-on-chip simulator. In: Proceedings of the 2013 IEEE International Symposium on Performance Analysis of Systems and Software (2013)
11. Sun, C., Chen, C.-H.O., Kurian, G., Wei, L., Miller, J., Agarwal, A., Peh, L.-S., Stojanovic, V.: Dsent - a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling. In: Proceedings of the 2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip, ser. NOCS 2012, pp. 201–210. IEEE Computer Society, Washington, DC, USA (2012)
12. Badr, M., Jerger, N.E.: Synfull: synthetic traffic models capturing cache coherent behaviour. In: Proceeding of the 41st Annual International Symposium on Computer Architecture, ser. ISCA 2014, pp. 109–120. IEEE Press, Piscataway, NJ, USA (2014)
13. Bienia, C.: Benchmarking modern multiprocessors. Ph.D. dissertation, aAI3445564, Princeton, NJ, USA (2011)
14. Woo, S.C., Ohara, M., Torrie, E., Singh, J.P., Gupta, A.: The splash-2 programs: characterization and methodological considerations. In: Proceedings of the 22nd Annual International Symposium on Computer Architecture, ser. ISCA 1995, pp. 24–36. ACM, New York, USA (1995)
15. Michelogiannakis, G., Shalf, J.: Variable-width datapath for on-chip network static power reduction. In: Proceedings of the 2014 IEEE/ACM Sixth International Symposium on Networks-on-Chip, ser. NOCS 2014, pp. 96–103. IEEE Computer Society, Washington, DC, USA (2014)
16. Chen, L., Pinkston, T.M.: Nord: node-router decoupling for effective power-gating of on-chip routers. In: Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture, ser. MICRO-45, pp. 270–281. IEEE Computer Society, Washington, DC, USA (2012)
17. Parikh, R., Das, R., Bertacco, V.: Power-aware nocs through routing and topology reconfiguration. In: Proceedings of the 51st Annual Design Automation Conference, ser. DAC 2014, pp. 162:1–162:6. ACM, New York, NY, USA (2014)
18. Matsutani, H., Koibuchi, M., Wang, D., Amano, H.: Run-time power gating of on-chip routers using look-ahead routing. In: Proceedings of the 2008 Asia and South Pacific Design Automation Conference, ser. ASP-DAC 2008, pp. 55–60. IEEE Computer Society Press, Los Alamitos, CA, USA (2008)
19. Chen, L., Zhu, D., Pedram, M., Pinkston, T.M.: Power punch: towards non-blocking power-gating of noc routers. In: Proceedings of the 2015 IEEE 21th International Symposium on High Performance Computer Architecture (HPCA), ser. HPCA 2015. IEEE Computer Society, Washington, DC, USA (2015)
20. Samih, A., Wang, R., Krishna, A., Maciocco, C., Tai, C., Solihin, Y.: Energy-efficient interconnect via router parking. In: Proceedings of the 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA), ser. HPCA 2013. IEEE Computer Society, Washington, DC, USA (2013)
21. Yue, D.Z.T.M.P.S., Chen, L., Pedram, M.: Smart butterfly: reducing static power dissipation of network-on-chip with core-state-awareness. In: Proceedings of the 2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED), pp. 311–314 (2014)
22. Chen, L., Zhao, L., Wang, R., Pinkston, T.M.: MP3: minimizing performance penalty for power-gating of clos network-on-chip. In: 20th IEEE International Symposium on High Performance Computer Architecture, HPCA 2014, pp. 296–307. IEEE Computer Society, Orlando, FL, USA, 15–19 Feb 2014

23. Balfour, J., Dally, W.J.: Design tradeoffs for tiled cmp on-chip networks. In: Proceedings of the 20th Annual International Conference on Supercomputing, ser. ICS 2006, pp. 187–198. ACM, New York, NY, USA (2006)
24. Das, R., Narayanasamy, S., Satpathy, S.K., Dreslinski, R.G.: Catnap: energy proportional multiple network-on-chip. In: Proceedings of the 40th Annual International Symposium on Computer Architecture, ser. ISCA 2013, pp. 320–331. ACM, New York, NY, USA (2013)
25. Mishra, A.K., Mutlu, O., Das, C.R.: A heterogeneous multiple network-on-chip design: an application-aware approach. In: The 50th Annual Design Automation Conference 2013, DAC 2013, p. 36. Austin, TX, USA, May 29–June 07 2013
26. Bokhari, H., Javaid, H., Shafique, M., Henkel, J., Parameswaran, S.: Darknoc: designing energy-efficient network-on-chip with multi-vt cells for dark silicon. In: Proceedings of the 51st Annual Design Automation Conference, ser. DAC 2014, pp. 161:1–161:6. ACM, New York, NY, USA 2014
27. Fallin, C., Nazario, G., Yu, X., Chang, K., Ausavarungnirun, R., Mutlu, O.: Minbd: minimally-buffered deflection routing for energy-efficient interconnect. In: Proceedings of the 2012 IEEE/ACM Sixth International Symposium on Networks-on-Chip, ser. NOCS 2012, pp. 1–10. IEEE Computer Society, Washington, DC, USA (2012)