EVC-based Power Gating Approach to Achieve Low-power and High Performance NoC

Peng Wang*, Sobhan Niknam*, Sheng Ma†, Zhiying Wang†, Todor Stefanov*
*Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
†State Key Laboratory of High Performance Computing, National University of Defense Technology, China
*{p.wang, s.niknam, t.p.stefanov}@liacs.leidenuniv.nl, †{mashnudt, zywang}@nudt.edu.cn

***Abstract*—High power consumption becomes the major bottleneck that prevents applying Network-on-Chips (NoCs) on future many-core systems. Power gating is an effective way to reduce the power consumption of a NoC. However, conventional power gating approaches cause significant packet latency increase as well as additional power consumption overhead due to the power gating mechanism. One comprehensive way to reduce these negative impacts is to bypass powered-off routers in a NoC when transferring packets. Therefore, in this paper, we propose an express virtual channel based (EVC-based) power gating approach. In our approach, packets can take pre-defined virtual bypass paths to bypass intermediate routers that can be powered-on or powered-off. Furthermore, based on our extended router structure, a certain transmission ability of the powered-off routers is kept to transfer packets going through the normal paths. Thus, even though some packets do not take a virtual bypass path, they still have less probability to be blocked by the powered-off routers. Compared with a conventional NoC without power gating, our EVC-based power gating approach causes only 2.67% performance penalty, which is less than 28.67%, 7.24%, and 5.69% penalties in related approaches. With small hardware overhead, our approach reduces on average 68.29% of the total power consumption in a NoC, which is comparable with the 72.94%, 73.56%, and 75.3% reduction of the total power consumption in related approaches.**

I. Introduction

A Network-on-Chip (NoC) with low latency, high bandwidth, and good scalability is a promising communication infrastructure for large size many-core systems. However, NoCs consume too much power in many-core systems [1]. For example, the NoC contributes up to 28% and 19% of the total system power consumption in the Teraflop [2] and Scorpio [3] chips, respectively. In fact, such high percentage of power consumption of a NoC has become the major bottleneck that prevents applying NoCs on high performance many-core systems [4].

On the other hand, NoCs have the characteristics of a distributed structure, naturally unbalanced traffic workload, and low average injection traffic rate, which make power gating being an applicable and effective way of powering off idle NoC routers to reduce the power consumption. However, conventional power gating approaches cause two negative impacts on the NoC performance: 1) Wakeup delay, there is a notable wakeup delay (6-12 clock cycles) [5] before the powered-off routers are fully recharged to the active state. This wakeup delay blocks the packet transmission between routers and causes the packet latency to significantly increase; 2) Breakeven time (BET), the power gating process causes additional power consumption. Normally, we use breakeven time (BET) to measure the idle time required to compensate the power overhead due to power gating. This implies that frequent power gating or power gating in a short time may cause more power consumption or inefficient power reduction.

Many approaches try to overcome the aforementioned drawbacks of power gating in different aspects. In order to reduce the negative impact of the wakeup delay, [6] and [5] switch on the powered-off routers ahead of packet transmission. Part of or the whole wakeup delay can be hidden, but these approaches have to power on the whole powered-off router every time when there is a packet going through a powered-off router, which may cause frequent power gating and results in more power consumption due to the frequent power gating. On the other hand, in order to avoid non-beneficial power gating caused by BET, many works [7], [8], [9] adopt fine-grained power gating on the components in a router. Instead of waking up the whole router, these approaches individually wake up part of the router components that are required to transfer packets and keep the rest of the router components powered off. In this way, some of the router components can have longer time to stay powered off. However, these approaches are at the expense of increasing the packet latency, as packets may experience more power gating processes over a routing path. In addition to the above mentioned approaches, bypass-based approaches such as in [10], [11], [12], [13] are more attractive and comprehensive to realize power efficient NoCs. This is because, by bypassing the powered-off routes along a routing path, packets do not need to be blocked and wait for the powered-off routers to be fully charged. Thus, the packet latency increase caused by the power gating is reduced. Furthermore, without frequent interruption of the sleeping state of the powered-off routers, routers have more idle time to stay powered-off and have less power consumption overhead caused by the power gating.

However, in the aforementioned bypass-based approaches, there are only a few bypass latches to temporarily store packets on a bypass path. Before bypassing powered-off routers, packets have to be blocked until there are available bypass latches, which significantly undermines the efficiency of the bypass paths. As a result, in most of the bypass-based approaches, the bypass path is not very efficient to transfer packets. For example, the bypass path in [10], [11], [13] cannot continuously transmit packets via bypass powered-off routers. Even though the approach in [12] can continuously transmit packets via bypass powered-off routers, it has significant timing overhead and hardware overhead to recover the routing information that is lost in the powered-off routers. As a consequence, all aforementioned bypass-based approaches still have significant packet latency increase caused by power gating.

In order to overcome this drawback, we propose an express virtual channel based (EVC-based) power gating approach. In our approach, multiple virtual bypass paths are pre-defined at design time. Packets can take these virtual bypass paths to bypass intermediate routers that can be powered-on or powered-off. When a packet takes a virtual bypass path,

978-1-7281-2862-7/19/$31.00 ©2019 IEEE
DOI 10.1109/DSD.2019.00027

the sink router of the virtual bypass path is powered-on. There is sufficient amount of buffers in sink routers to hold packets. Thus, packets can continuously go through a virtual bypass path. Furthermore, compared with other bypass-based approaches [10], [11], [12], [13] in which the packets can only bypass powered-off routers, in our EVC-based approach, packets can bypass powered-on routers as well. Thus, even at a high workload traffic, our approach also can reduce the power consumption by reducing the dynamic power. The specific novel contributions of this paper are the following:

- We propose a specific distribution of virtual bypass paths on a NoC, which allows more packets to take the virtual bypass paths compared to the conventional EVC scheme [14]. More importantly, we extend the router structure to guarantee that a virtual bypass path cannot be blocked by powered-off routers. Thus, by allowing packets going through the virtual bypass paths without blocking, these packets can avoid suffering the negative impact of the power gating process at the intermediate routers. Furthermore, based on our extended router structure, a certain transmission ability of the powered-off/being charged routers is kept to transfer packets going through the normal paths. In this way, the negative impact of power gating is further reduced. We also propose an effective power gating scheme to control the power switching of routers. Finally, we propose an approach to freeze virtual bypass paths in order to resolve starvation, which is a common issue in EVC-based NoCs [14].

- By experiments, we show that our EVC-based power gating approach can effectively reduce the power gating negative impacts on the performance and power consumption. Taking a conventional NoC without power gating as the baseline, our EVC-based power gating approach causes only 2.67% performance penalty, which is less than the 28.67% penalty in [6], 7.24% in [9], and 5.69% in [13]. With small hardware overhead, our EVC-based power gating approach reduces on average 68.29% of the total power consumption in a NoC, which is comparable with the 72.94%, 73.56%, and 75.30% reduction of the total power consumption in [6], [9], and [13], respectively. Furthermore, by allowing packets to bypass powered-on routers as well, our approach achieves lower power consumption than the related approaches [6], [9], [13] under high traffic workloads.

The remainder of the paper is organized as follows: Section II gives some background information on the conventional power gating approach and introduces the express virtual channel scheme. Section III provides an overview of the related work. Section IV elaborates our EVC-based power gating structure and power gating approach. Section V introduces the experimental setup and presents experimental results. Section VI concludes this paper.

II. Background

In order to better understand the novel contributions of this paper, in this section, we give some background information about the conventional power gating scheme on a NoC and the conventional EVC [14] scheme that allows packets to virtually bypass intermediate routers along a virtual bypass path.

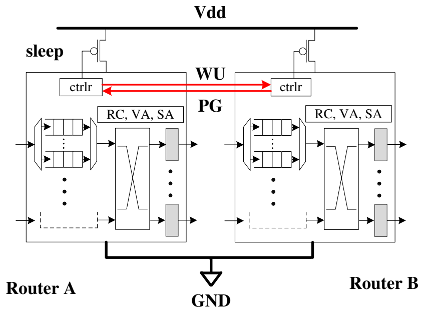


Fig. 1: Conventional NoC power gating.

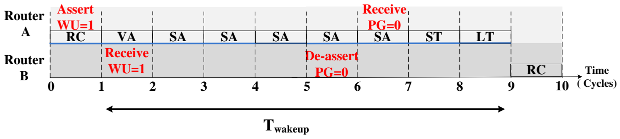


Fig. 2: Wakeup process.

A. Conventional NoC power gating

In this section, we discuss the conventional power gating in a NoC. An implementation example of applying conventional power gating on the routers is shown in Figure 1. The router is a virtual-channeled wormhole router and consists of input ports, a virtual channel (VC) allocator, a switch allocator, a crossbar, and output ports. By inserting header transistors between the voltage supply and the router, the power controller (the ctrlr unit in Figure 1) can cut off the power supply of the router to save power consumption. In order to correctly control the packet transmission, additional handshaking control signals WU (wakeup) and PG (power gating) are added between routers.

When $RouterB$ is idle (there are no flits left in input ports or the crossbar) and the WU signals are clear, the controller in $RouterB$ asserts the sleep signal to cut off the router's power supply and asserts the PG signal to notify its upstream $RouterA$. Once $RouterA$ receives the signal PG, $RouterA$ marks the output port to $RouterB$ as being powered-off and cannot send packets to go to $RouterB$.

An optimized wakeup process is shown in Figure 2. When $RouterA$ executes the routing computation (RC) stage for packets, $RouterA$ determines that there is a packet going to $RouterB$ and asserts the WU signal to wake up $RouterB$. In the following clock cycles, $RouterA$ executes the VC allocation (VA) stage and the switch allocation (SA) stage, but as $RouterB$ is powered off, the packet has to be blocked in $RouterA$. Once the WU single is received, the ctrlr unit in $RouterB$ clears the sleep signal to charge $RouterB$. After experiencing $T_{wakeup} - MARGIN$ ($MARGIN = 4$ in this example) clock cycles, $RouterB$ de-asserts the PG signal. When $RouterA$ is aware that the PG signal is de-asserted, $RouterA$ allows the packet to go to $RouterB$ and executes the switch traversal (ST) stage and the link traversal (LT) stage to transfer the packet. When the packet reaches $RouterB$, $RouterB$ is just fully charged.

B. Express virtual channel

The express virtual channel (EVC) scheme [14] is a classical virtual bypass technique. As shown in Figure 3(a), the virtual bypass paths (red dashed lines) are pre-defined on a NoC topology. These virtual bypass paths are implemented without the need for physical links, but based on the virtual channels in a router to share the existing links. The basic EVC router architecture is shown in Figure 3(b). Compared with the conventional router in Figure 1, in each input port, one EVC latch is added, and the virtual channels are partitioned into two groups, normal virtual channels (N-VCs) and express virtual channel (E-VCs). N-VCs are used to accepted packets from neighbor upstream routers. E-VCs in the sink routers of the virtual bypass paths are used to accept packets from the source routers of the virtual bypass paths.

By allocating E-VCs to packets, the source router in a virtual bypass path can determine if the packet takes the virtual bypass path. For example, in Figure 3, a packet is sent form $Router00$ to $Router04$. Based on the transmission distance, $Router00$ is aware that by taking the virtual bypass path from $Router00$ to $Router03$, the packet has lower latency. So, $Router00$ treats this packet as an E-packet (the packet going through a virtual bypass path) and allocates one E-VC in $Router03$ for this packet. When the packet reaches $Router01$ and $Router02$, this packet is temporarily held in the EVC latch with the highest priority. Then, this packet is directly sent without experiencing the pipeline stages in $Router01$ and $Router02$, and reaches $Router03$. When this packet reaches $Router03$, this packet is stored at the allocated E-VC. $Router03$ knows this packet should go to the normal path to its destination $Router04$, and treats this packet as a N-packet (the packet going through the normal path between routers) and allocates a N-VC in $Router04$ for this packet. After experiencing the pipeline stages in $Router03$, this packet is sent to its destination $Router04$.

By taking virtual bypass paths, E-packets do not need to experience the pipeline stages in the intermediate routers. This implies that most of the components in the intermediate routers are unnecessary to transfer E-packets. This characteristic is attractive and promising for realizing a power gating NoC to allow packets to bypass powered-off routers. We exploit effectively this characteristic in this paper to realize our EVC-based power gating approach.

III. Related work

Several approaches propose a bypass-based power gating NoC. In Nord [10], a virtual ring is pre-defined on a NoC, which works as a backup NoC. When a packet is blocked by a powered-off router, it can go along this virtual ring to bypass the powered-off router. However, limited by the low efficiency and poor scalability of the virtual ring, packets may be detoured for a long distance to their destinations. As a consequence, Nord has significant packet latency increase and is not suitable for large NoCs. In contrast, in our approach, we pre-define multiple virtual bypass paths, which are separately distributed on the whole NoC. Packets go along their shortest routing path and separately take these virtual bypass paths to bypass the powered-off routers. Thus, our EVC-based power gating approach has lower packet latency and better scalability.

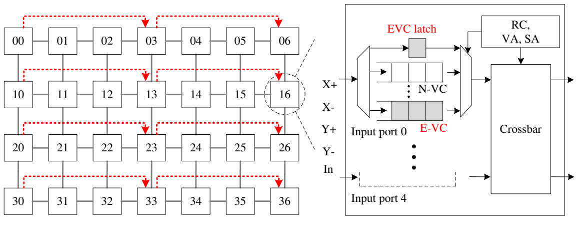
In Turn-on on Turn (TooT) [11], a bypass path is pre-defined in the horizontal $(X+/X-)$ and vertical $(Y+/Y-)$ directions. Thus, packets can bypass a powered-off router if the packets do not need the powered-off router to change the transmission direction or to eject from the NoC. So, TooT does not need to frequently power on the powered-off routers and can more efficiently reduce the static power consumption. However, limited by a few bypass latches on a bypass path, packets have to be blocked until there are available bypass latches. As a consequence, the bypass paths are inefficient to transmit packets in order to bypass the powered-off routers and TooT still has significant packet latency increase. In contrast, in our EVC-based power gating approach, when a packet goes through a virtual bypass path, the sink router is powered on. Thus, there are more buffers to be used to hold packets and packets can continuously go through the virtual bypass path. As a consequence, bypass paths in our approach are more efficient than TooT in terms of transmitting packets, therefore the packet latency increase is reduced.

Similar to TooT, Fly-over [12] also allows packets to bypass powered-off routers in the horizontal $(X+/X-)$ and vertical $(Y+/Y-)$ directions but Fly-over does not need to block packets to wait for available bypass latches between the neighbor routers. This is because Fly-over dynamically realizes the credit-based flow control [15] between the source router and the sink router on a bypass path to guarantee that there is no buffer overflow. When a source router transmits packets to bypass the intermediate powered-off routers, the sink router must be powered-on. Thus, there is sufficient amount of buffers available to be used to hold packets and Fly-over can continually transmit packets. However, Fly-over has to employ a complex mechanism to realize the credit-based flow control between the source router and the sink router, which causes significant timing and hardware overhead. In contrast, in our EVC-based power gating approach, the virtual bypass paths are (static) pre-defined. Thus, our EVC-based approach has no such extra timing overhead.

In contrast to TooT and Fly-over, the bypass path in EZ-bypass [13] is dynamically built to allow packets to bypass the powered-off routers in any direction. Thus, a packet can bypass a powered-off router, even when this router is required to change the transmission direction. As a result, EZ-bypass is more flexible and can be more efficient to reduce the power consumption. However, in EZ-bypass, when a packet bypasses powered-off routers, this packet has to stay in the powered-off routers for multiple clock cycles to experience the pipeline stages of routers. As a consequence, the bypass latch is occupied by one packet for a long time and the bypass path is frequently blocked, which undermines the efficiency of the bypass path. In contrast, in our EVC-based power gating approach, when a packet bypasses intermediate routers, this packet does not experience the router pipeline stages. Thus, our EVC-based power gating approach can achieve lower packet latency than EZ-bypass. Furthermore, compared with Nord [10], TooT [11], Fly-over [12], and EZ-bypass [13] in which the packets can bypass only powered-off routers, in our EVC-based approach, packets can bypass powered-on routers as well. Thus, even at a high workload traffic, our approach also can reduce the power consumption by reducing the dynamic power.

IV. Our EVC-based Power Gating

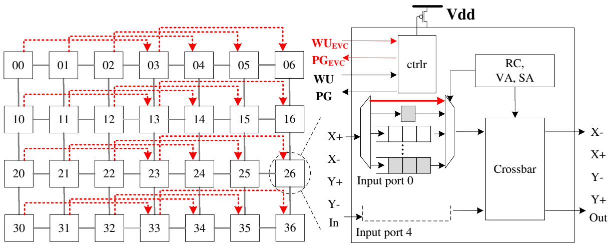
In this section, we present our novel approach to use the EVC scheme to allow packets to bypass powered-off routers. First, in **Section IV-A**, we propose a distribution of the virtual bypass paths to allow more packets to take the virtual bypass paths. Then, in **Section IV-B**, we extend the EVC router structure to guarantee that the virtual bypass paths are not blocked by the powered-off routers. Thus, packets can always take a virtual bypass path to bypass the intermediate routers that may be powered-off. Furthermore, based on our extended router structure, a powered-off router has certain transmission ability to transfer also packets that take the normal paths. So,



(a) A NoC with virutal bypass paths

(b) EVC router

Fig. 3: Express virtual channel.



(a) Distribution of virtual bypass path in the X+ direction

(b) Extended EVC router

Fig. 4: Extended EVC-based power gating approach.

even though some packets do not take a virtual bypass path, they can avoid as much as possible to be blocked by powered-off routers. In **Section IV-C**, we describe our power gating scheme used in our EVC-based power gating approach, and in **Section IV-D**, we use an example to illustrate our power gating scheme. Finally, in **Section IV-E**, we propose an approach to resolve the starvation which may occur when using our EVC-based power gating approach.

A. Distribution of virtual bypass paths

In the EVC scheme, packets can bypass the intermediate routers only when they take virtual bypass paths. So, in order to allow packets to bypass the intermediate routers that may be powered-off, we have to allow more packets to take the virtual bypass paths. To achieve this goal, in each direction, we pre-define one virtual bypass path between each two routers with three hops. As shown in Figure 4(a), in the $X+$ direction, we set one virtual bypass path between $Router00$ and $Router03$, $Router01$ and $Router04$ and so on. The virtual bypass paths in the $X-$, $Y+$, and $Y-$ directions have similar settings, but are not shown in Figure 4(a) for the sake of clarity. Compared with the conventional distribution of the virtual bypass paths [14] in Figure 3(a), the packets in Figure 4(a) have higher probability to take a virtual bypass path. For example, in a 8×8 2D mesh, there are in total 4032 routing paths from one source node to a destination node. Based on the distribution of the virtual bypass paths in Figure 3(a), the average number of virtual bypass paths on a routing path is 0.56, while, based on our distribution of the virtual bypass paths in Figure 4(a), the average number of virtual bypass paths on a routing path is 1.13.

In our EVC-based power gating approach, routers always try to send packets to a virtual bypass path. Only when there is no virtual bypass path available, the packets are sent along the normal path between routers.

B. Extended router structure

We have extended the basic EVC router in Figure 3(b) to enable and support our novel power gating scheme. As shown in Figure 4(b), one power control (ctrlr) unit is added in the router. Handshaking control signals WU (wakeup) and PG (power gating) are added between routers. Compared with the conventional power gating, introduced in Section II-A, extra handshaking control signals, WU_{EVC} and PG_{EVC} are added between the source router and the sink router for a virtual bypass path. In each input port, one direct link is added (the red arrow in Input port 0, shown in Figure 4(b)). These direct links are used to build the bypasses in the direction from $X+$ to $X-$, $X-$ to $X+$, $Y+$ to $Y-$, and $Y+$ to $Y-$. **To avoid N-packets to be blocked by the powered-off routers, in our EVC based power gating approach, the EVC latch is also used to hold N-packets when the router is powered-off or being charged**. When a router is powered off and the EVC latch in an input port is used to hold a N-packet, a bypass path is setup by using the direct link in the input port and the crossbar for E-packets. For example, when a router is powered-off and the EVC latch in the $X+$ input port holds a N-packet, a bypass path from $X+$ to $X-$ is built by using the direct link in the $X+$ input port and the crossbar in this router for E-packets. Then, if an E-packet is coming, it directly goes through this router by taking this directly built bypass in the router. In this way, we guarantee that the virtual bypass path always works for E-packets even when the EVC-latch is occupied by a N-packet. Furthermore, the powered-off router has certain transmission ability to transfer N-packets through the normal paths. In this way, the N-packets have less probability to be blocked by powered-off routers.

To transfer N-packets though a powered-off router, the RC unit, the EVC latches, the VA unit, the SA unit, and the crossbar are always powered on to execute the router pipeline stages. The power control (ctrlr) unit only cuts off the power supply of VCs. In this way, even at the powered-off state, the router still keeps a certain ability to transfer packets. Thus, the packets going through the normal paths have less probability to be blocked by the powered-off routers. Furthermore, as these units consume much less power than VCs [9], [8], our EVC-based power gating approach still can efficiently reduce the static power consumption by powering off the idle VCs.

C. Power gating scheme

In this section, we introduce the conditions which drive our ctrlr unit in Figure 4(b) to control the power supply of a router.

1) Powering off a router: When there are no packets left in EVC latches, N-VCs, E-VCs, or the crossbar in a router, and the WU and WU_{EVC} signals from all its upstream routers are de-asserted, the router goes into the idle state, the PG_{EVC} and PG signals are asserted to all upstream routers, but at this moment, the power supply is not cut off yet. After waiting T_{idle_detect} clock cycles, the ctrlr unit cuts off the power supply. If there is any WU or WU_{EVC} signal asserted during T_{idle_detect}, the ctrlr unit immediately de-asserts the PG_{EVC} and PG signals. By delaying T_{idle_detect} clock cycles to cut off the power supply, we can avoid non-beneficial power gating caused by short idle time of routers, which causes frequent power gating and additional power consumption.

2) Powering on a router: If a source router determines that a packet should take the virtual bypass path to the sink router, this source router asserts the corresponding WU_{EVC} to power on the sink router. If a router determines that a packet should take the normal path to the downstream router, this router asserts WU to power on the downstream router. Once the powered-off router receives the WU_{EVC} signal or the WU signal, the powered-off router starts to charge and goes into the wakeup state. After $T_{wakeup} - MARGIN_{EVC}$ clock cycles, the router de-asserts PG_{EVC} and the source router can send packets to this router using the virtual bypass path. After $T_{wakeup} - MARGIN$ clock cycles, the router de-asserts PG and the upstream router can send packets to this router using the normal path. By setting properly $MARGIN_{EVC}$ and $MRGIN$, a router can send packets before the powered-off router is fully charged, but it is guaranteed that when a packet reaches the powered-off router, this router is just fully charged. In this way, we can hide part of the wakeup delay and optimize the power gating process. It should be noted that $MARGIN_{EVC}$ is larger than $MARGIN$. This is because by taking virtual bypass paths, E-packets have more time on the transmission via multiple hops than N-packets taking the normal path to transfer over single hop. This implies that the wakeup delay has less negative impact on the virtual bypass paths. Thus, it is more beneficial for packets to take the virtual bypass paths to avoid the negative impact of power gating.

D. Example of our power gating approach

In this section, we use the example in Figure 5 to clearly illustrate our EVC-based power gating approach.

In Figure 5(a), at time T=0, $Router0$ and $Router1$ are powered-on and $Router2$ and $Router3$ are powered-off. $Router0$ is going to send an E-packet (the red blocks in Figure 5) to $Router3$ by using the virtual bypass path, so $Router0$ asserts the WU_{EVC} signal to wakeup $Router3$. $Router1$ is going to send one packet to $Router3$, but there is no virtual bypass path available, so $Router1$ treats this packet as a N-packet (the blue blocks in Figure 5) and sends it by using the normal path to $Router2$ first. So, $Router1$ has to asserts the WU signal to wakeup $Router2$.

At time $T = 1$, $Router2$ and $Router3$ receive the WU and WU_{EVC} and begin to power on, respectively. At time $T = 0, 1, 2, 3$, $Router1$ executes the router pipeline stages for its N-packet. The head flit of the N-packet leaves $Router1$ at time $T = 3$. At time $T = 4$, this head flit is going through the link, as shown in Figure 5(b). At time $T = 2$, $Router2$ and $Router3$ de-asserts the PG_{EVC} signals, but the E-packet is still blocked for one clock cycle at $Router0$. So, at time $T = 4$ (Figure 5(b)), the E-packet has not been sent yet.

In Figure 5(c), at time $T = 5$, the head flit of the N-packet reaches $Router2$ and $Router2$ holds this head flit at its EVC latch. At the same time, in $Router2$, one bypass path is setup by using the direct link and the crossbar. The head flit of the E-packet leaves $Router0$ and is traversing the link.

In Figure 5(d), at time $T = 6$, as $Router2$ has to execute the router pipeline stages for the N-packet. The head flit of the N-packets has to occupy the EVC latch for multiple clock cycles. For the E-packet, the head flit reaches $Router1$ and is held at the EVC latch. The tail flit of the E-packet also leaves $Router0$.

In Figure 5(e), at time $T = 7$, the head flit of the E-packet leaves $Router1$ and the tail flit of the E-packet is held at the EVC latch of $Router1$.

In Figure 5(f), at time $T = 8$, the head flit of the E-packet directly goes through the directly built bypass path in $Router2$, and is traversing the link from $Router2$ to $Router3$. The tail flit of the E-packet is traversing the link from $Router1$ to $Router2$.

In Figure 5(g), at time $T = 9$, the head flit of the N-packet leaves $Router2$ and the bypass path in $Router2$ is demolished. For the E-packet, the head flit reaches its destination $Router3$. $Router3$ is just fully charged and stores this flit into the allocated E-VC. The tail flit of the E-packet is held at the EVC latch in $Router2$.

In Figure 5(h), at time $T = 10$, the head flit of the N-packet is stored in $Router3$ and the tail flit of this N-packet is stored in $Router2$. As $Router2$ and $Router3$ are already full charged. These flits are stored in the corresponding N-VCs.

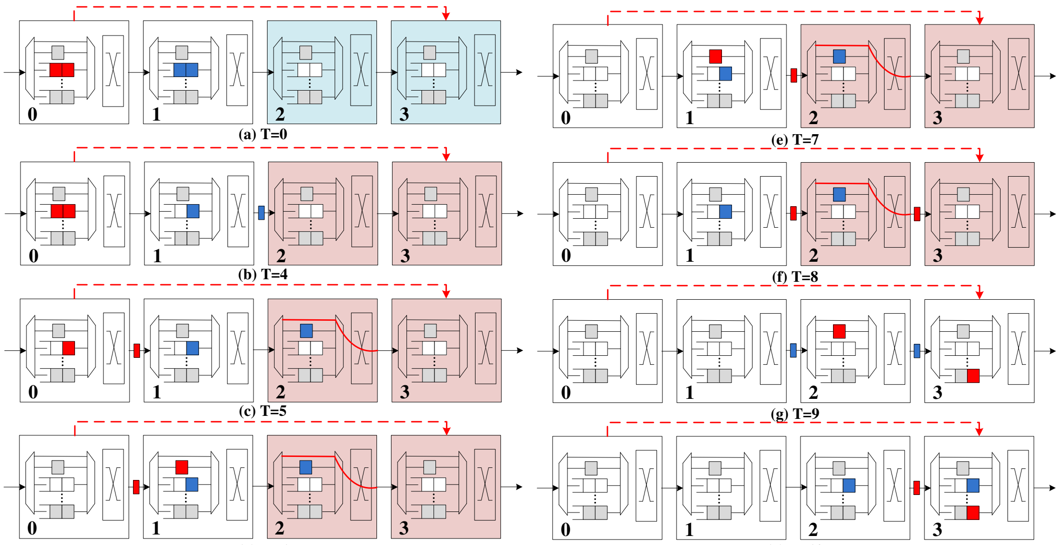
This example clearly shows that, by temporarily holding the packets in the EVC latches, the powered-off/ being charged routers can keep certain transmission ability to transfer N-packets. Thus, the N-packet can avoid as much as possible to be blocked by the powered-off/being charged routers. Furthermore, this process does not block the virtual bypass paths at all.

E. Resolving starvation

Starvation is a common issue in EVC-based NoCs [14]. When an E-packet goes through an intermediate router along one virtual bypass path, the E-packet has the highest priority and the intermediate router has to send it first. If the source router continuously transfers E-packets through the virtual bypass path, the N-packets in the intermediate router cannot get a chance to be sent and starvation occurs. In order to resolve the starvation, we use the approach provided in [14] to detect the starvation and then temporarily freeze the related virtual bypass paths. For example, in Figure 4(a), if $Router01$ continuously sends E-packets to $Router04$ or $Router02$ continuously sends E-packets to $Router05$, $Router03$ cannot send packets to its downstream $Router04$. Once such starvation occurs, $Router03$ needs to freeze both these two virtual bypass paths. To simplify the control between routers, we use two different ways to freeze these two virtual bypass paths: 1) To freeze the virtual bypass path from $Router01$ to $Router04$, $Router03$ informs the sink $Router04$ to assert PG_{EVC} in the direction $X-$. In this way, $Router01$ cannot send E-packets to $Router04$; 2) At the same time, to freeze the virtual bypass path from $Router02$ to $Router05$, $Router03$ informs the source $Router02$ to stop allocating E-VCs in the $X+$ direction to packets. In this way, $Router02$ cannot send E-packets to $Router05$ and the virtual bypass path is freezed. Thus, as all the virtual bypass paths through $Router03$ are freezed, no E-packets prevent $Router03$ to send its packets, thereby resolving the starvation. When the packets, initially affected by the starvation, leave $Router03$, then $Router03$ informs $Router04$ to de-assert the PG_{EVC} signal as well as $Router03$ allows $Router02$ to allocate E-VCs to packets. In this way, the frozen virtual bypass paths are activated and can be used again.

V. Experimental Results

In order to evaluate our EVC-based power gating approach in terms of performance and power consumption, we have implemented our approach using the full-system simulator called Agate [16]. Agate is based on the widely used full-system simulator GEM5 [17] and Agate supports the simulation of the key items in NoC power gating techniques. The NoC model and power model used in Agate are based on Garnet [18] and Dsent [19], respectively. The key parameters used in our



(a) T=0 (b) T=4 (c) T=5 (d) T=6 (e) T=7 (f) T=8 (g) T=9 (h) T=10

Fig. 5: An example of our power gating approach.

TABLE I: Parameters used in experiments.

Network topology	8 × 8 mesh
Router	4-stage pipeline
Virtual channel	(1 N-VC, 1 E-VC)/VN, 3 VNs,
Input buffer size	1-flit/ ctrl VC, 5-flit / data VC
Routing algorithm	X-Y
Link bandwidth/delay	128 bits/cycle, 1 clock cycle
Voltage, Frequency, Technology	1V, 1GHz, 45nm
Wakeup delay	8 clock cycles
Break even time	10 clock cycles
T_{idle_detect}	8 clock cycles
$MARGIN_{evc}$ / $MARGIN$	6/4 clock cycles
Private I/D L1$	32 KB
Shared L2 per bank	256 KB
Cache block size	16 Bytes
Coherence protocol	Two-level MESI
Memory controllers	4, located one at each corner

experiments are shown in Table I. We choose a four-stage pipeline router. There are three virtual networks (VNs): two data VNs and one ctrl VN. In each input port, there is one N-VC and one E-VC for each VN. The value of the wakeup delay and break even time (BET) are set according to the related works [5] and [10]. Based on the NoC configuration, we set T_{idle_detect}, $MARGIN_{evc}$, and $MARGIN$ such that we keep the correctness of the NoC.

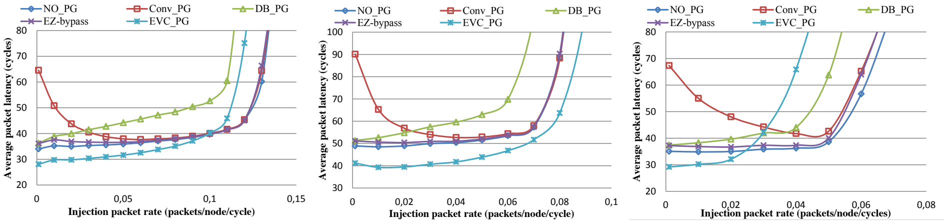
For comparison purpose, we have implemented the following power gating approaches: (1) NO_PG: the baseline NoC without power gating; (2) Conv_PG [6]: conventional power-gating NoC, which is deeply optimized by sending WU and de-asserting PG signals in advance, thus 6 clock cycles of the wakeup delay are hidden in our experiments; (3) DB_PG [9]: the power gating NoC with Duty Buffer structure. In each input port of a router, a one-flit size duty buffer is added to implement the Duty Buffer approach. The reason that we choose the DB_PG approach is because DB_PG is a fined-grant power gating approach which is effective on reducing the NoC power consumption, but also efficiently reduces the packet latency increase caused by power gating; (4) EZ_bypass [13]: the power gating NoC with the EZ_bypass scheme to reduce the negative impact of the power gating process. Compared with other bypass-based related approaches [10], [11], [12], EZ_bypass is more flexible to allow packets to bypass the powered-off routers. (5) EVC_PG: the NoC with our EVC-based power gating approach.

A. Evaluation on Synthetic Workloads

In order to explore the behaviour of our EVC_PG, in this section, we evaluate the performance and power consumption of our EVC_PG approach under synthetic traffic patterns. We select three synthetic traffic patterns: 1) uniform random: packets' destinations are randomly selected; 2) bit-complement: packets from source node (x, y) are sent to destination node (N-x, N-y), N is the number of nodes in the X and Y dimensions of a NoC; 3) transpose: packets from source node (x, y) are sent to destination node (y, x);

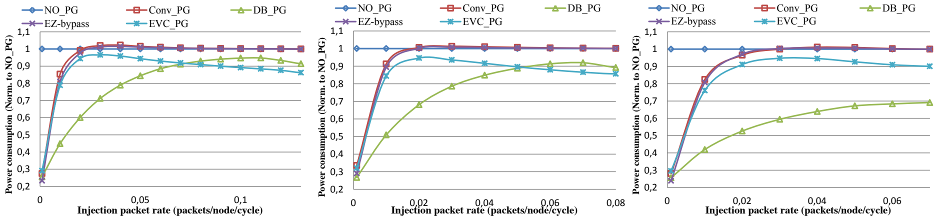
Figure 6 shows the average packet latency under different injection rates. Compared with NO_PG, Conv_PG, DB_PG, and EZ_bypass, our EVC_PG has the lowest average packet latency. These results indicate that our EVC_PG can effectively reduce the negative impact of the wakeup delay and can be used to achieve low latency communication. On the other hand, our EVC_PG has lower saturation points than NO_PG, Conv_PG, and EZ_bypass for the Uniform random and Transpose patterns, but has higher saturation point for the Bit-complement pattern. The lower saturation points indicate that our EVC_PG causes some throughput loss. This is because, in order to support the EVC scheme, the VCs in our EVC_PG are partitioned into E-VCs and N-VCs, which may undermine the flexibility and effectiveness of VCs. Since, Conv_PG and EZ_bypass are based on NO_PG, they have the same saturation points as NO_PG. However, the impact caused by the partition of E-VCs and N-VCs highly depends on the traffic pattern. Thus, for Bit-complement, our EVC_PG achieves higher saturation point.

Figure 7 shows the power consumption normalized to NO_PG under different injection rates. When the injection rate is around 0.001 packets/node/cycle, our EVC_PG has slightly higher power consumption than Conv_PG and EZ-bypass, but much lower than NO_PG. This is because, in order to avoid packets to be blocked by powered-off routers, we always keep some components powered on in the powered-off routers, which causes extra power consumption but this power consumption is rather low. When the injection rate increases,



(a) Uniform random (b) Bit-complement (c) Transpose

Fig. 6: Latency across different injection rates.



(a) Uniform random (b) Bit-complement (c) Transpose

Fig. 7: Power consumption across different injection rates.

more and more routers become busy and cannot be powered off. The power reduction in Conv_PG, DB_PG, EZ_PG, and EVC_PG becomes lower and lower, but DB_PG has much higher power reduction than the other approaches. This is because DB_PG can separately power off VCs in each input port of routers whereas Conv_PG, EZ_bypass, and EVC_PG can power off a router only when all of the input ports of the router are idle. Thus, DB_PG fully utilizes the idle time of each input port to reduce the power consumption.

When the injection rate is higher than 0.02 packets/node/cycle in Figure 7(a), and in Figure 7(b), and higher than 0.03 packets/node/cycle in Figure 7(c), Conv_PG and EZ_bypass become ineffective on reducing the power consumption, while DB_PG and EVC_PG still can effectively reduce the power consumption. The power reduction in our EVC_PG is due to the fact that packets can also bypass powered-on routers, which saves some dynamic power.

When the injection rate further increases, the dynamic power takes higher and higher portion of the total power consumption. Our EVC_PG reduces more dynamic power consumption, which causes the curves for our EVC_PG in Figure 7(a), Figure 7(b), and Figure 7(c) to decline. As a result, when the injection rates are higher than 0.07 packets/node/cycle in Figure 7(a) and 0.05 packets/node/cycle in Figure 7(b), our EVC_PG has lower power consumption than DB_PG. However, in Figure 7(c), DB_PG has always lower power consumption than our EVC_PG. This is because DB_PG and EVC_PG reach their saturation points at low packet injection rates as shown in Figure 7(c). So, the dynamic power consumption takes small portion of the total power consumption. As a consequence, the efficient reduction of the dynamic power consumption in our EVC_PG does not play a significant role in reducing the total power consumption in this case, whereas DB_PG more efficiently reduces the static power consumption by separately powering off input ports of routers, leading to better reduction of the total power consumption in this case.

B. Evaluation on Real Application Workloads

In this section, we use real application workloads to compare the approaches in terms of the application performance,

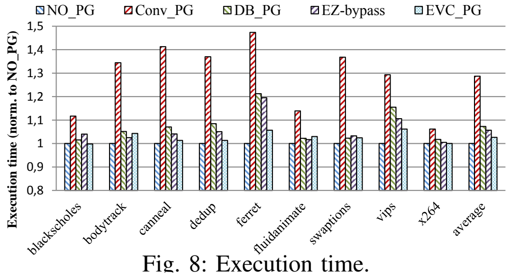


Fig. 8: Execution time.

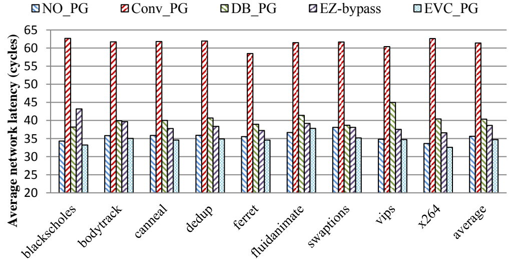


Fig. 9: Average network latency.

the average network latency, and the NoC power consumption. To do so, we use nine applications from the Parsec [20] benchmark suit.

1) Effect on the application performance: Figure 8 shows the execution time of the nine applications, which is normalized to the baseline NO_PG, and the tenth set of bars in Figure 8 gives the average results over these nine applications. Our EVC_PG approach causes less performance penalty (execution time increase) than the related approaches. Compared with the baseline NO_PG, our EVC_PG causes, on average, 2.67% performance penalty, which is less than the 28.67% performance penalty in Conv_PG, 7.24% in DB_PG, and 5.69% in EZ_bypass. For blackscholes and x264, our EVC_PG has slightly lower execution time than NO_PG. In vips, our EVC_PG has its highest performance penalty of 6.17%, which is still lower compared to Conv_PG, DB_PG, and EZ_bypass. For ferret, Conv_PG, DB_PG, and EZ-bypass have their highest performance penalty of 47.39%, 21.21%, and 19.51%, respectively.

2) Effect on the average network latency: Figure 9 shows the average network latency across the nine applications.

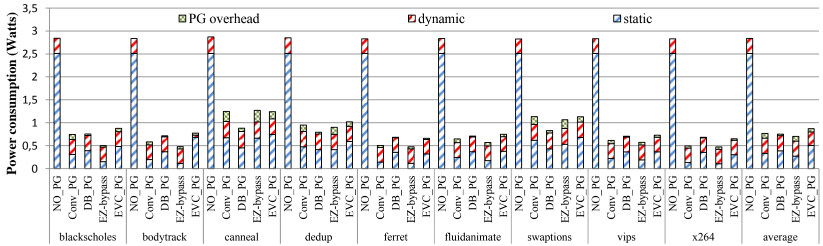


Fig. 10: Power consumption.

Compared with NO_PG across the applications, the average network latency in our EVC_PG approach is slightly lower, whereas Conv_PG, DB_PG, and EZ_bypass have higher average network latency compared to NO_PG. As DB_PG uses a fined-grain power gating scheme, packets in DB_PG suffer more power gating processes. As a consequence, DB_PG has much higher average network latency than our EVC_PG and EZ_bypass. EZ_bypass allows packets to bypass powered-off routers, but packets have to stay ib powered-off routers for a long time experiencing the router pipeline stages. In contrast, in our EVC_PG, the packets can bypass the intermediate routers without the need to experience the router pipeline stages. Thus, our EVC_PG has lower average network latency than EZ_PG.

Even though our EVC_PG has a slightly lower average network latency compared to NO_PG (see Figure 9), our EVC_PG still causes a slightly higher execution time in most of the applications compared to NO_PG (see Figure 8). This is because EVC_PG breaks the fairness of the communication between routers when E-packets take the virtual bypass paths to bypass intermediate routers and have a higher priority compared to N-packets.

3) Effect on the NoC power consumption: Figure 10 shows the breakdown of the NoC power consumption across the nine applications and the tenth set of bars shows the average over these nine applications. The NoC power is broken down into three parts: the power consumption caused by power gating (PG overhead), the static/dynamic power consumption of routers (static/dynamic).

As shown in Figure 10, our EVC_PG approach consumes slightly higher total power than the related approaches Conv_PG, DB_PG, and EZ_PG. This is because our EVC_PG needs some components in a router to be always powered on, which causes slightly more static power consumption compared to Conv_PG, DB_PG, and EZ_PG. As the traffic workloads in real applications are low, the dynamic power consumption is low. As a result, the dynamic power reduction in our EVC_PG does not play a significant role in reducing the total power consumption. Compared with NO_PG, our EVC_PG reduces on average 68.29% of the total power consumption, which is comparable with the 72.94%, 73.56%, and 75.30% reduction of the total power consumption in Conv_PG, DB_PG, and EZ_bypass, respectively.

VI. Conclusion

In this paper, we propose an EVC-based power gating approach. In our approach, packets can take pre-defined virtual bypass paths to bypass intermediate routers that may be powered-on or powered-off. Furthermore, even though some packets do not take a virtual bypass path, our approach tries to ensure that these packets avoid as much as possible blocking in the powered-off routers. As a result, our approach reduces more efficiently the packet latency increase caused by power gating. Furthermore, by allowing packets to bypass powered-on routers to reduce dynamic power consumption, our approach can achieve lower power consumption under high traffic workloads.

VII. Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61672526, Grant 61572508, in part by the Research Project of NUDT under Grant ZK17-03-06, and in part by the Science and Technology Innovation Project of Hunan Province under Grant 2018RS3083.

References

[1] S. Borkar, “Thousand core chips: a technology perspective,” in *DAC*, 2007.

[2] Y. Hoskote *et al.*, “A 5-ghz mesh interconnect for a teraflops processor,” *IEEE Micro*, 2007.

[3] B. K. Daya *et al.*, “Scorpio: a 36-core research chip demonstrating snoopy coherence on a scalable mesh noc with in-network ordering,” *ACM SIGARCH Computer Architecture News*, 2014.

[4] H. Esmaeilzadeh *et al.*, “Dark silicon and the end of multicore scaling,” in *ACM SIGARCH Computer Architecture News*, 2011.

[5] L. Chen *et al.*, “Power punch: Towards non-blocking power-gating of noc routers,” in *HPCA*, 2015.

[6] H. Matsutani *et al.*, “Run-time power gating of on-chip routers using look-ahead routing,” in *DAC*, 2008.

[7] H. Matsutani *et al.*, “Ultra fine-grained run-time power gating of on-chip routers for cmps,” in *NOCS*, 2010.

[8] J. Zhan *et al.*, “Dimnoc: A dim silicon approach towards power-efficient on-chip network,” in *DAC*, 2015.

[9] P. Wang *et al.*, “A novel approach to reduce packet latency increase caused by power gating in network-on-chip,” in *NOCS*, 2017.

[10] L. Chen and T. M. Pinkston, “Nord: Node-router decoupling for effective power-gating of on-chip routers,” in *ISCA*, 2012.

[11] H. Farrokhbakht *et al.*, “Toot: an efficient and scalable power-gating method for noc routers.,” in *NOCS*, 2016.

[12] R. Boyapati *et al.*, “Fly-over: A light-weight distributed power-gating mechanism for energy-efficient networks-on-chip,” in *IPDPS*, 2017.

[13] H. Zheng and A. Louri, “Ez-pass: An energy & performance-efficient power-gating router architecture for scalable nocs,” *IEEE Computer Architecture Letters*, 2018.

[14] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, “Express virtual channels: towards the ideal interconnection fabric,” in *ACM SIGARCH Computer Architecture News*, 2007.

[15] W. J. Dally *et al.*, *Principles and practices of interconnection networks*. Elsevier, 2004.

[16] L. Chen *et al.*, “Simulation of noc power-gating: Requirements, optimizations, and the agate simulator,” *JPDC*, 2016.

[17] N. Binkert *et al.*, “The gem5 simulator,” *ACM SIGARCH Computer Architecture News*, 2011.

[18] N. Agarwal *et al.*, “Garnet: A detailed on-chip network model inside a full-system simulator,” in *ISPASS*, 2009.

[19] C. Sun *et al.*, “Dsent-a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling,” in *NoCs*, 2012.

[20] C. Bienia *et al.*, “The parsec benchmark suite: Characterization and architectural implications,” in *PACT*, 2008.