2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems

A Dynamic Bypass Approach to Realize Power Efficient Network-on-Chip

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 the powered-off routers in a NoC to transfer packets. Therefore, in this paper, we propose a dynamic bypass (D-bypass) approach, which is based on a reservation mechanism to allow different upstream routers to forward packets through the same powered-off router at different times. With this feature, our D-bypass power gating approach overcomes the drawbacks in related power gating approaches. Compared with a conventional NoC without power gating, our D-bypass approach causes only 2.55% performance penalty, which is less than 28.67%, 19.26%, 7.24%, and 6.69% penalties in related approaches. With small hardware overhead, our approach just consumes on average 22.23% of total power consumption in a NoC, which is slightly better compared to 27.06%, 23.89%, 26.45%, and 24.70% 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, this 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 routers ahead of packet transmission. Part of or the whole wakeup delay can be hidden, but these approaches have to power on the powered-off router every time when there is a packet going through the 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 router components. 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] 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.

In [10], Chen proposes one feasible and applicable bypass-based NoC power gating approach called Node-Router Decoupling (NoRD). By using a bypass latch (in the network interface (NI)) in a downstream router as a transfer station, a packet can be ejected from the NoC to the network interface without the need of writing the packet into a powered-off router buffer. Then the packet can be re-injected (forwarded) to the next router without the need of going through the crossbar in the powered-off router. By repeatedly forwarding packets, the NoRD approach allows packets to go through the powered-off routers in any hop count. Meanwhile, as packets still go through powered-off routers, the conventional credit-based flow control is available to guarantee that there is no buffer overflow. Compared with other bypass-based NoCs [11], this feature greatly simplifies the flow control. However, NoRD does not support bypass in all directions, i.e., in a powered-off router, the bypass latch in a network interface can accept packets from only one specific upstream router and forward packets to only one specific downstream router.

978-1-7281-2058-4/19/$31.00 ©2019 IEEE
DOI 10.1109/HPCC/SmartCity/DSS.2019.00279

As a consequence, when packets try to bypass the powered-off routers, there is only one available transmission direction and packets are forced to follow detour routing paths, not the shortest routing paths, which results in an inefficient packet transmission and poor scalability.

In order to overcome this drawback, in this paper, we propose a dynamic bypass (D-bypass) approach. Based on a reservation mechanism to dynamically reserve a bypass latch in a powered-off router, the same bypass latch can be used by different upstream routers to dynamically build the bypass path. Thus, packets can bypass a powered-off router in any direction, which makes it possible for packets to always follow their shortest routing paths. Furthermore, as the reservation process is executed in parallel (overlaps) with the router pipeline, the timing overhead caused by the reservation process is minimized. The specific novel contributions of this paper are the following:

- We extend the router structure to allow a bypass latch in a powered-off router to accept packets from any upstream router. Then, we propose a reservation mechanism to allow different upstream routes to share the same bypass latch at different times. In this way, the bypass path can be dynamically built based on the routing information of packets. Thus, when packets bypass the powered-off router, they can always follow the shortest routing paths.
- By experiments, we show that our D-bypass 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 D-bypass approach causes only 2.55% performance penalty, which is less than the 28.67% penalty in [6], 19.27% in [10], 7.24% in [9], and 5.69 in [12]. With small hardware overhead, our D-bypass just consumes on average 22.23% of total power consumption in a NoC, which is slightly less than 27.06%, 23.89%, 26.45% and 24.70% total power consumption in [6], [10], [9], and [12] respectively.

The remainder of the paper is organized as follows: Section II gives some background information on the conventional power gating approach and the NoRD power gating approach. Section III provides an overview of the related work. Section IV elaborates our D-bypass 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 contributions of this paper, in this section, we give some background information about NoC power gating and briefly introduce the NoRD approach.

A. Conventional NoC power gating

In this section, we discuss the power gating in a NoC. An implementation example of applying 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 allocator (VA), a switch allocator (SA), 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, an additional handshaking control signals WU (wakeup) and PG (power gating) are added between routers.

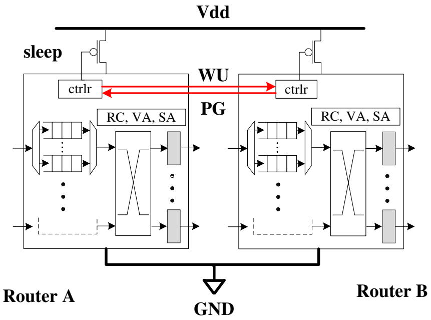


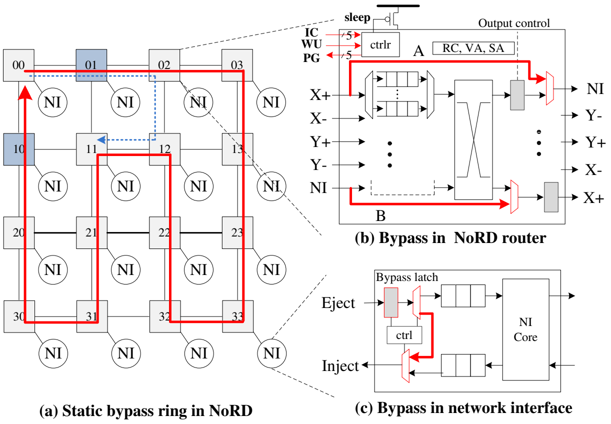
Fig. 1: Conventional NoC power gating.

We use Figure 1 to explain the power gating process 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 of PG, $RouterA$ marks the output port to $RouterB$ as being powered-off and sets the credits in the output port (to $RouterB$) to 0. When there are packets going to $RouterB$, as there is no credit in the corresponding output port, packets are blocked at input ports in $RouterA$. In this situation, $RouterA$ asserts the WU signal to wake up $RouterB$. Once the WU signal is received, the ctrlr unit in $RouterB$ clears the sleep signal to charge $RouterB$. After experiencing T_{wakeup} (wakeup delay) clock cycles, $RouterB$ is fully charged and the PG signal is cleared. $RouterA$ sets the credits in the corresponding output port to be full and consumes these credits to transfer packets to $RouterB$.

It should be noted that the output ports of a router are never powered off. This is because the output ports contain the number of credits that are used to indicate the number of free buffers in the downstream routers. If output ports are powered off, all the credit information would be lost and when the router is powered on, it is difficult and expensive [11] to recover this information. So, it is better to keep the output ports powered on to guarantee the correctness of the conventional credit-based flow control.

B. Node-Router Decoupling

Node-Router Decoupling (NoRD) [10] is a feasible way to bypass the powered-off routers to transfer packets. As shown in Figure 2(b), two bypass paths are added in a router. When the router is powered-off, packets directly go through bypass path A in Figure 2(b) and are stored in the bypass latch in Figure 2(c). Then, packets go through bypass path B in Figure 2(b) to be forwarded to the next router. In this way, packets can go through the powered-off router and be forwarded to the next router. Furthermore, as the packets still go through the powered-off router, the conventional credit-based flow control still works to guarantee that there is no buffer overflow. However, constrained by the router structure, NoRD does not support bypassing of the powered-off router in all directions, i.e., in a powered-off router, each network interface

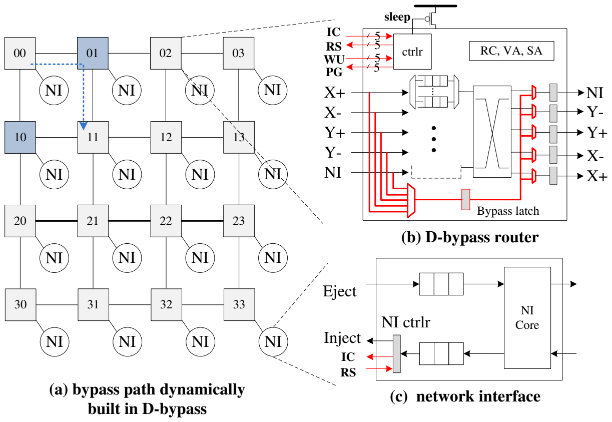


(a) Static bypass ring in NoRD

(b) Bypass in NoRD router

(c) Bypass in network interface

Fig. 2: Node-Router Decoupling.



(a) bypass path dynamically built in D-bypass

(b) D-bypass router

(c) network interface

Fig. 3: Extended router structure in D-bypass.

can accept packets from only one specific upstream router and forward packets to only one specific downstream router. As shown in Figure 2(a), in NoRD, a bypass ring is statically constructed to achieve full connectivity among routers. To bypass a powered-off router, packets have to go along the static bypass ring path. For example, as shown in Figure 2(a), $Router00$ tries to send packets to $Router11$, and its two downstream routers $Router01$ and $Router10$ are powered-off. $Router00$ only can send packets to bypass $Router01$. However, as $Router01$ only can forward packets along the bypass ring, packets are transferred to $Router02$ in spite of the fact that there is only one hop form $Router01$ to $Router11$. Then, after going through $Router02$ and $Router12$, packets reach the destination $Router11$. In this example, as NoRD only can forward packet to a special direction, packets have to be transferred in a detour/longer routing path, which undermines the transmission effectiveness. Furthermore, for a large size NoC, this static bypass ring is quite long, which extremely limits the scalability of NoRD.

III. Related Work

Many approaches try to optimize the conventional power gating approach briefly explained in Section II-A. Inspired by the look-ahead routing, Matsutani [6] proposes a run-time power gating approach. By sending the WU single ahead of one hop before packet transmission, part of the wakeup delay can be hidden, but this approach cannot hide the whole wakeup delay. In [5], Chen proposes a low cost approach to send the WU signal ahead of multiple hops before packet transmission. In this way, the whole wakeup delay can be hidden under deterministic routing algorithms. However, both approaches [6] and [5] have to power on the powered-off routers when there is a packet going through the powered-off routers. As a result, these power gating approaches are inefficient to reduce the power consumption. By contrast, in our approach, packets can be transferred through the powered-off routers without the need of powering them on. In this way, our approach is more efficient in reducing the power consumption.

As most of the components in a router are individually used by different packets, it is unnecessary to wake up the whole router, but just wake up the components that are required. In this way, the rest of the components can be powered off for a longer time to reduce the non-beneficial power gating caused by BET. Matsutani [7] proposes an ultra fine-grained power gating approach, i.e., each component of a NoC router can be individually powered-off. In this way, the idle time of each component can be fully used to reduce power consumption. Considering that virtual channels (VCs) consume most of the static power consumption in a NoC, [8] and [9] apply power gating on VCs. [8] uses the drowsy SRAM [13] to build the VCs. As the drowsy SRAM has less wakeup delay, the powered-off VCs can be waked up faster. [9] adds one buffer queue (called Duty Buffer) at each input port, which is used to temporarily replace any powered-off VC. In this way, even if all of the VCs are powered-off, the router can still keep a certain packet transmission ability. However, as fine-grained power gating approaches have larger number of power gating processes, power gating still has a serious negative impact on the NoC performance. In contrast, our D-bypass is a coarse-grained power gating approach. By transferring packets through the powered-off routers, our D-bypass approach reduces the number of power gating processes to decrease the negative impact on the NoC performance caused by power gating, Thus, our D-bypass approach is more efficient in reducing power consumption and performance penalty caused by power gating.

A few approaches explore a bypass-based power gating NoC. Fly-over [11] switches off the power of an entire router (including output ports) and allows packets to bypass the powered-off routers, but Fly-over supports bypass in only four directions. When a packet needs a router to change its transmission direction, this router must be waked up. Furthermore, as the output ports are powered off and all the credit information is lost, Fly-over has to employ a complex flow control to recover credit information when a powered-off router is powered on, which causes significant hardware overhead (a router needs 48 extra links to support this special flow control). Compared with Fly-over, Node-Router Decoupling (NoRD) [10] just uses the conventional credit-based flow to control the packet transmission. However, as we have introduced in Section II-B, NoRD supports only one direction bypass in each powered-off router, which results in an inefficient packet transmission and poor scalability. Our D-bypass approach also adopts the conventional credit-based flow that is similar to NoRD. However, in contrast to Fly-over [11] and NoRD [10], our D-bypass approach is based on a reservation mechanism to dynamically build the bypass path, thus packets can bypass the powered-off routers in any direction and in any hop count. Furthermore, the reservation mechanism needs just 10 extra links for each router, which is much less than the 48 extra links in Fly-over [11]. With these aforementioned differences, our D-bypass approach has better scalability than Fly-over [11] and has lower packet latency and less power consumption than NoRD [10].

EZ-bypass [12] has a similar bypass structure with our D-bypass and allows packets to bypass the powered-off router in any direction. In EZ-bypass, each input port of a router needs one bypass latch to temporarily hold a packet. When a packet bypasses a powered-off router, this packet has to experience the multiple pipeline stages of the router, because there may be contention with packets in other input ports. However, in our D-bypass approach, as there is only one bypass latch in a router and only one packet can reserve this bypass latch to bypass the powered-off router in a time, there is no contention when the packet is going through the powered-off router. Thus, the router pipeline can be minimized to one stage and some packet transmissions are accelerated. Furthermore, based on the number of reservation signals from the upstream routers, the powered-off router can detect the contention earlier. Thus, our D-bypass can switch on the power of the powered-off router earlier than EZ-bypass.

IV. Dynamic bypass approach

Flyover [11] and NoRD [10] does not support bypassing in all directions. This limitation is mainly caused by the fact that the bypass latch cannot be shared by all the upstream routers to forward packets. Therefore, in our dynamic bypass approach, we first add several bypasses in a router, which allow a bypass latch to accept packets from any of its upstream routers. Then, we propose a reservation mechanism to allow different upstream routers to use the same bypass latch at different times. By reserving the bypass latch at different times, the same bypass latch can be used to dynamically build the bypass paths from any upstream router to any downstream router. Consider the same example as described in Section II-B, where a packet has to be sent from $Router00$ to $Router11$ and where $Router01$ and $Router10$ are powered off. Before packets are sent to the bypass latch in $Router01$, $Router00$ reserves the bypass latch in $Router01$. Next the head flit of a packet is sent to the bypass latch in $Router01$ and based on the routing information in the head flit, the bypass path is dynamically built from $Router01$ to $Router11$, see Figure 3(a). Then, $Router01$ can forward the packet to $Router11$. In this way, when packets go through the powered-off routers, they can always follow the shortest routing paths to their destinations.

A. Extended router structure

In this section, we introduce the extended router structure to support our D-bypass power gating approach. As shown in Figure 3(b)(c), and in contrast to NoRD [10], we remove the bypass latch from the NI and place it in the router, and put a NI controller (NI ctrlr) in the NI, which is used to reserve the bypass latch. In order to allow packets from all directions to skip the process of writing into input buffers, and directly write to the bypass latch, we add five bypasses to connect the input ports ($X+$, $X-$, $Y+$, $Y-$, and output Inject of the NI) with the input multiplexer. We also add five multiplexers, one in each output port, and connect the bypass latch to these output multiplexers. Based on the above mentioned extension, without the need of the crossbar, the bypass latch can accept packets from all input directions and forward packets to any of the output directions. All multiplexers are controlled by the ctrlr unit.

When multiple upstream routers need a bypass latch to forward packets, there is only one bypass latch, as shown in Figure 3(b), so the bypass latch cannot simultaneously forward packets coming from multiple upstream routers. However, it is possible for multiple upstream routers to share the same bypass latch by using it at different points in time. To achieve this sharing, we have devised a reservation mechanism and its hardware support. As shown in Figure 3(b), the handshaking signals, i.e., incoming signals (ICs) and reservation success signals (RSs), are added between routers. The IC signals are also used in NoRD. In an upstream router, the IC signal is asserted to inform a downstream router that a packet is coming.

Besides the aforementioned IC signal functionality in NoRD, the important role of the IC signal in our D-bypass approach is to reserve the bypass latch in the powered-off router. When an upstream router tries to send packets to a powered-off router, instead of asserting the WU signal, it asserts the IC signal to reserve the bypass latch in the powered-off router. When the ctrlr unit in the powered-off router detects this IC signal, the ctrlr unit marks the bypass latch as reserved and does not allow other upstream routers to use it. Meanwhile, the corresponding RS signal is asserted to inform the upstream router that it gets the right to use this bypass latch to forward packets. Once the upstream router receives this RS signal, it can send packets to that powered-off router. As our D-bypass router can forward packets to any output direction, when the packet is stored in the bypass latch, the ctrlr unit can, based on the routing information in the packet, forward the packet along its shortest routing path. In this way, according to the requirement of the packet transmission, the bypass path in a powered-off router can be dynamically built. When the upstream router finishes the packet transmission, it clears the IC signal. Then, the powered-off router releases the reservation of the bypass latch and allows other upstream routers to reserve it.

Based on the aforementioned reservation mechanism, at different times, the bypass latch in a powered-off router can be used by different upstream routers and the bypass path can be dynamically built to forward packets along their shortest routing path.

B. An example of the reservation process

In order to show the details of our reservation mechanism, we use the example in Figure 4 to illustrate the reservation process in our D-bypass approach. We assume a four-stage pipeline router, which consists of route computation (RC), virtual channel allocation (VA), switch allocation (SA), and switch traversal (ST). The link traversal (LT) takes one more clock cycle. $RouterA$ tries to send packets to $RouterB$, but $RouterB$ is powered-off. The reservation process is shown in Figure 4.

In Cycle 0, $RouterA$ executes the RC stage for a packet and is aware that the packet should go to $RouterB$. So, the IC signal is asserted to reserve the bypass latch in $routerB$.

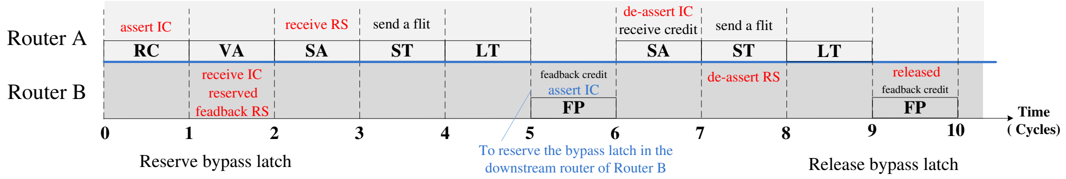


Fig. 4: Example of the reservation process.

In Cycle 1, $RouterA$ executes the VA stage for packets. Meanwhile, the ctrlr unit in $RouterB$ receives the IC signal, sets the input multiplexer to select the corresponding input port, marks the bypass latch as reserved, and asserts the corresponding RS signal to acknowledge that $RouterA$ can forward packets through $RouterB$. If there are multiple ICs simultaneously asserted to reserve the same bypass latch, the ctrlr unit employs a round robin arbitration to grant the bypass latch to one of them.

In Cycle 2, $RouterA$ executes the SA stage. As the RS signal has arrived at this moment, $RouterA$ gets the right to forward packets to $RouterB$. The head flit of one packet is granted to go to $RouterB$. The rest of the flits are blocked at the SA stage until that $RouterA$ receives the credit from $RouterB$ or $RouterB$ is powered on.

In Cycle 3, in the ST stage of $RouterA$, the head flit of the packet is sent to the crossbar. Then, in Cycle 4, in the LT stage of $RouterA$, the head flit is sent to $RouterB$.

In Cycle 5, $RouterB$ stores the head flit in the bypass latch. As no other packets can enter $RouterB$, there is no need to execute the VA, SA, and ST stages, so pipeline stages are simplified to one stage; Forward Packet (FP). In the FP stage, according to the routing information in the head flit, the ctrlr unit builds the bypass path for the packet, i.e., the ctrlr unit determines the output port and selects an available VC for the packet, then sets the corresponding output multiplexer to forward the head flit and the rest of flits of the packet to the downstream router of $RouterB$ (if $RouterB$ is the destination router, the packet will be directly ejected to the NI). In this way, the bypass path can be dynamically built. Furthermore, if there are multiple packets transfers through $RouterB$ at different times, different bypass paths can be dynamically built for each packet.

It should be noted that the IC signal from $RouterB$ to the downstream router of $RouterB$ is also asserted in this clock cycle. If the downstream router of $RouterB$ is also powered off, the head flit is blocked at the FP stage until $RouterB$ gets the RS signal from its downstream router. In this way, the packet can bypass multiple powered-off routers. When one flit leaves $RouterB$, one credit is feedback to $RouterA$.

In Cycle 6, $RouterA$ gets the credit to send another flit. In our example, the packet has two flits, so, the packet transmission is finished in this clock cycle and the IC signal is de-asserted.

In Cycle 7, $RouterA$ executes the ST stage for the last flit. $RouterB$ is aware that the IC signal is de-asserted and de-asserts the RS signal.

After experiencing the LT stage in Cycle 8, the last flit arrives in $RouterB$. In Cycle 9, the last flit is forwarded to the downstream router of $RouterB$. The ctrlr unit in $RouterB$ releases the reservation of the bypass latch and allows other upstream routers to reserve the bypass latch.

Based on the reservation process exemplified above, the bypass latch in the powered-off routers can be used by all upstream routers to forward packets to any direction at different times. By reserving multiple bypass latches in different routers, packets can bypass multiple powered-off routers along their routing path. Furthermore, as shown in this example, the reservation process is executed in parallel (overlaps) with the router pipeline. Thus, the timing overhead of the reservation process is minimized.

C. Power gating conditions

In this section, we introduce the conditions which drive the ctrlr unit in Figure 3(b) to control the power supply of a router.

1) Powering off a router: When there is no packet left in a router, and the ICs and WUs signals from all its upstream routers are de-asserted, the router goes into the idle state and the 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 IC or WU signals asserted during T_{idle_detect}, the ctrlr unit immediately de-asserts the 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: To keep good NoC performance, the routers should be powered on at the right moment to deal with high traffic workloads. In our D-bypass approach, we use two metrics to determine when a router should be powered on.

- N_{IC} is the number of ICs received by a powered-off router. In a powered-off router, when N_{IC} exceeds a threshold th_{IC}, the powered-off router is waked up. In this situation, the condition of powering on a router is triggered by the IC signals. As an IC signal is sent ahead of a packet transmission, part of the wakeup delay is hidden. Furthermore, during the time of charging the powered-off router, one of the upstream routers can forward packets through the powered-off router. Thus, the packet latency increase caused by the wakeup delay is reduced.
- N_{IVC} is the number of input VCs, in one upstream router, contending for the same downstream router to forward packets. N_{IVC} indicates the workload of an upstream router. As there is only one bypass latch in a router, our D-bypass approach has significant credit round-trip delay, which blocks a packet transmission to wait for credits. Powering on the downstream routers can reduce this impact. In an upstream router, when N_{IVC} to a powered-off downstream router

TABLE I: Parameters.

Network topology	8 × 8 mesh
Router	4-stage pipeline
Virtual channel	2 VCs/VN, 3 VNs
Input buffer size	1-flit/ ctrl VC, 5-flit / data VC
Routing algorithm	X-Y, Adaptive
Link bandwidth	128 bits/cycle
Wakeup delay	8 clock cycles
Break even time	10 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

exceeds a threshold th_{IVC}, the corresponding WU signal is asserted to wakeup the downstream router. During the time of waiting the downstream router to fully charge, the upstream router can forward packets through the bypass of the downstream router, so the impact of the wakeup delay is also reduced.

In order to avoid performance penalties as much as possible, we aggressively set the thresholds $th_{IC} = 1$ and $th_{IVC} = 1$, which implies that when multiple packets are sent simultaneously to the same powered-off router, the powered-off router should be powered on. The low th_{IC} and th_{IVC} may tend to trigger more often the condition of powering on a router, which may cause frequent power gating on a router. However, considering the low average injection rate in real applications, there is still high probability of transferring packets through powered-off routers without frequently triggering the condition of powering on a router.

V. Experimental Results

In order to evaluate our approach in terms of performance and power consumption, we have implemented our approach using a full system simulator called Agate [14]. Agate is based on the widely used full-system simulator GEM5 [15], 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 [16] and Dsent [17], respectively. The key parameters used in our experiments are shown in Table I. We choose a four-stage pipeline router. The number of VCs and the buffer size of control VCs and data VCs are set based on the related works [5] and [10]. For simplicity, we use X-Y deterministic routing algorithm in our D-bypass approach and other related approaches, but for the NoRD approach, we have implemented the special adaptive routing algorithm required by NoRD [10] to fairly compare with the NoRD approach. The value of the wakeup delay and break even time (BET) are according to related works [5] and [10]. As there are additional components added in our D-bypass router and the routers in related approaches, in order to evaluate the power consumption of these components, we use Dsent [17] to estimate the power consumption of the major components, such as the buffers and multiplexers, to make the experimental results more accurate.

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 is hidden in our experiments;

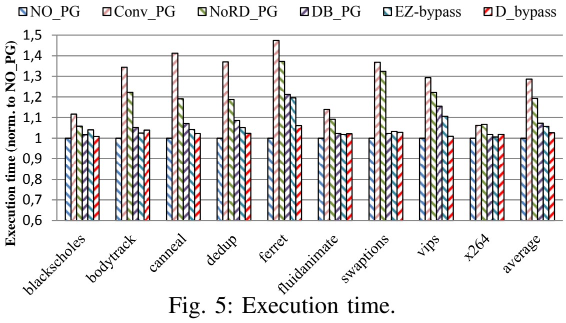


Fig. 5: Execution time.

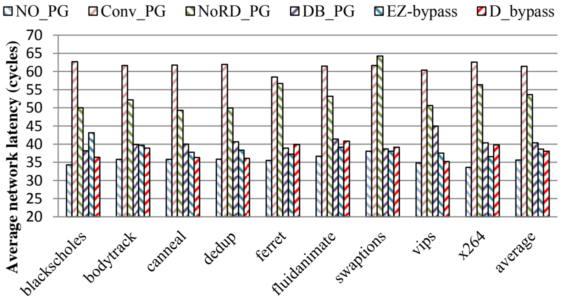


Fig. 6: Average packet latency.

(3) NoRD_PG [10]: the power gating NoC with the NoRD approach; (4) 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 that, in terms of functionality, DB_PG is similar to our D-bypass approach; (5) EZ_bypass [12]: the power gating NoC with EZ-bypass approach in which the bypass structure is similar to our approach. (6) D-bypass: the power gating NoC with our D-bypass approach introduced in Section IV.

A. Evaluation on Real Application Workloads

In this section, we use real application workloads to compare the approaches in terms of the application performance, the NoC average packet latency, and the NoC power consumption. To do so, we use nine applications from the Parsec [18] benchmark suit.

1) Effect on Application Performance: Figure 5 shows the execution time of the nine applications, which is normalized to the baseline NO_PG, and the tenth set of bars in Figure 5 gives the average results over these nine applications. Our D-bypass approach causes less performance penalty (execution time increase) than the related approaches. Compared with the baseline NO_PG, our D-bypass causes an average of 2.55% performance penalty, which is less than the 28.67% performance penalty in Conv_PG, 19.27% in NoRD_PG, 7.24% in DB_PG, and 5.69% in EZ_bypass. In ferret, our D-bypass has its largest performance penalty of 6.03%, and Conv_PG, NoRD_PG, DB_PG, and EZ_bypass have also their largest performance penalty of 47.39%, 37.18%, 21.22%, and 19.51%, respectively.

2) Effect on NoC network latency: Figure 6 shows the average network latency across the nine applications. Our D-bypass approach can efficiently reduce network latency increase caused by power gating. Compared with NO_PG across the applications, the average network latency in our D-bypass

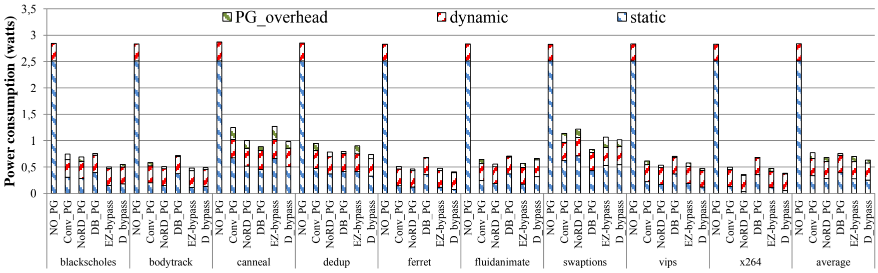


Fig. 7: Breakdown of the NoC power consumption.

approach slightly increases, but is much lower than Conv_PG and NoRD_PG. This is because our D-bypass approach can dynamically build the bypass path and allow packets to bypass the powered-off router in all directions. Thus, packets can go along the shortest routing paths to bypass the powered-off routers, and are not blocked due to the power gating processes.

In most of applications, our D-bypass approach has slightly lower average network latency then DB_PG and EZ_bypass. This is because DB_PG is a fine-grained power gating approach and causes more power gating processes. Compared with EZ_bypass, our D_bypass is based on a reservation mechanism which can power on the powered-off router earlier when multiple upstream routers need the same powered-off router to forwards packets. However, in ferret, fluidanimate, swaptions, and x264, our D-bypass approach has slightly higher average network latency than EZ_bypass, because each input port in EZ_bypass has a bypass latch to hole one flit of a packet, whereas in our D-bypass approach, all input ports in a router have to share one bypass latch to forward packets, which may result in more contention and block some packet transmissions. However, in our D-bypass, as only one packet is allowed to go through a powered-off router at a time, the router pipeline stage can be minimized to one stage when packets bypass the powered-off routers. Thus, some packet transmissions are accelerated and our D-bypass approach has lower application execution time than EZ_bypass in ferret and swaptions, in spite of the fact that our D-bypass approach has slightly higher average packet latency than EZ_bypass.

3) Effect on NoC power consumption: Figure 7 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 dynamic/static power consumption of routers (dynamic/static).

As shown in Figure 7, our D-bypass approach consumes slightly less total power consumption than the related approaches. Compared with NO_PG, our D-bypass just consumes on average 22.23% of total power consumption, which is slightly less than 27.06% total power consumption in Conv_PG, 23.89% in NoRD_PG, 26.45% in DB_PG, and 24.70% in EZ_bypass. This is because our D-bypass approach can transfer packets through the powered-off routers without waking up them. Thus, routers can be powered off for longer time and consumes less Router_static and PG_overhead. Even though NoRD_PG is also a bypass-based power gating approach, it does not support bypass in all directions and forces packets to go along the bypass ring. Packets have to go through more routers, which may cause more power gating processes. As a consequence, NoRD_PG consumes more Router_static and PG_overhead than our D_bypass.

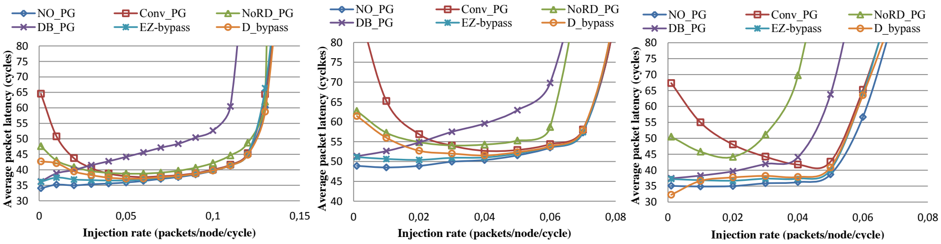
The fine-grained power gating approach DB_PG has the lowest PG_overhead, because DB_PG does not need to power on the entire router, but just powers on the VCs that are required to transfer packets. However, DB_PG requires more buffers to support the Duty Buffer scheme, which consumes more power. As a consequence, under real applications, DB_PG consumes more static power than our D-bypass approach.

B. Evaluation on Synthetic Workloads

In order to further explore the behaviour of our D-bypass approach under a wider range of packet injection rates, in this section, we evaluate the performance of our D-bypass 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 (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).

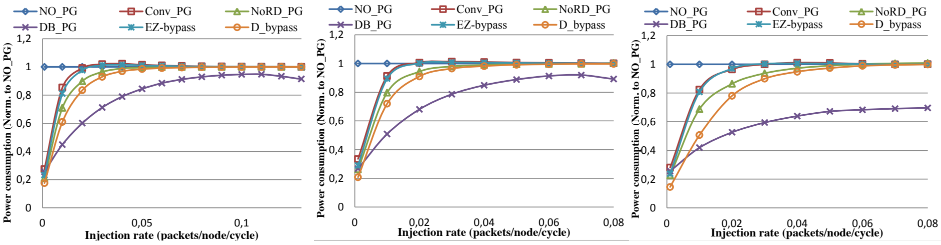
As shown in Figure 8(a) and Figure 8(b), when the injection is around 0.001 packets/node/cycle, our D-bypass approach has higher average packet latency than DB_PG and EZ_PG, but lower than Conv_PG and NoRD_PG This is because in our D-bypass approach, multiple packets cannot simultaneously bypass the same powered-off routers at the same time, and some packets are blocked due to power gating. However, compared with Conv_PG, there are significant number of packets that can bypass the powered-off routers. On the other hand, as when the packet bypasses the powered-off router, the powered-off router pipeline is minimized to one stage and some packets can be accelerated. Thus, in Figure 8(c), our D-bypass has lowest packet latency among all the approaches.

As shown in Figure 8, with the injection rate increasing before the saturation injection rate (around 0.13 in uniform random, 0.07 in bit-complement, 0.05 in transpose), the curve of average packet latency in our D-bypass approach slowly drops, and it is lower than the curve of Conv_PG and NoRD_PG, and gradually gets close to the curve of NO_PG. This indicates that our D-bypass approach can more efficiently deal with high bursty traffic workloads than Conv_PG and NoRD_PG, which meets requirements of real applications where traffic workloads are bursty. However, with the injection rate increase, as shown in Figure 9, the power consumption in our D-bypass approach increases and fast equals to the NO_PG. This is because, we apply power gating on a router. When the injection rate increases, more routers become busy and cannot be powered



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

Fig. 8: Packet latency across full range of workloads.



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

Fig. 9: Power consumption normalized to NO_PG across full range of workloads.

off. As a result, our D-bypass approach can efficiently reduce the power power consumption only under the low injection rate.

The saturation injection rate is also an improve parameter to evaluate the NoC performance. A NoC with higher saturation injection rate can achieve higher throughput. As shown in Figure 8, our D-bypass approach has the same saturation injection rate as the baseline NO_PG, but NoRD_PG and DB_PG have lower saturation injection rate. This is because, at the saturation injection rate, all routers are powered on and our D-bypass approach works the same as NO_PG. However, the routers in NoRD_PG are not as efficient as the routers in NO_PG. This is because NoRD_PG needs VCs to support its special adaptive routing along the bypass ring. As a consequence, NoRD_PG cannot fully utilize VCs to achieve the same saturation injection rate as NO_PG. Therefore, compared with the bypass-based power gating scheme NoRD_PG, our D-bypass approach can achieve higher throughput.

VI. Conclusion

In this paper, we propose a dynamic bypass approach to allow packets to bypass the powered-off router in any hop count and in any direction. Based on a reservation mechanism, all the upstream routers can share the same bypass latch to dynamically build the bypass path for different packets. In this way, packets can be transferred along their shortest routing paths. With small hardware overhead, our D-bypass approach can efficiently reduce the power consumption and has less performance penalty.

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 *et al.*, “Nord: Node-router decoupling for effective power-gating of on-chip routers,” in *MICRO*, 2012.

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

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

[13] K. Flautner *et al.*, “Drowsy caches: simple techniques for reducing leakage power,” in *ISCA*, 2002.

[14] L. Chen *et al.*, “Simulation of noc power-gating: Requirements, optimizations, and the agate simulator,” *Journal of Parallel and Distributed Computing*, 2016.

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

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

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

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