POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informazione e Bioingegneria

A Power Gating Methodology to Aggressively Reduce Leakage Power in Networks-on-Chip Buffers

Relatore: Prof. William Fornaciari
Correlatore: Dr. Davide Zoni

Tesi di Laurea di:
Andrea Canidio, 804496

Anno Accademico 2014-2015

Abstract

Buffers are the major leakage component in multi-cores which rely on NoC and also they affect heavily the performance of the whole interconnect and the system. Traditionally they are dimensioned for the worst traffic scenario, thus they remain idle or underutilized for the majority of the time.

The under use of these resources is magnified by the burst traffic behavior in real applications, where high traffic periods are alternated with periods where there is no traffic. The problem to mitigate leakage power is considered in many methodologies by the exploitation of the power gating actuator both at router or buffer level.

However, the power gating actuation at router level is a suitable technique against low and medium traffic while it is useless at high traffic. Moreover, they impose strong constraints to the router function and the network topology.

This thesis presents a power gating methodology focusing on the router's buffers in NoC based multi-cores to reduce leakage power while keeping almost the same performance of the considered baseline NoC. The methodology is aseptic from the routing scheme or the topology of the network and it focuses also on the minimization of power on and off oscillations, introducing traffic reshaping. Furthermore, it does not impact in any way on the critical path of the NoC router.

Results obtained with synthetic traffic and real applications with up to 64 cores systems show a limited performance overhead of at most the 2% on average introduced by the methodology, with a per router energy saving of up to 74%.

Acknowledgments

To my family, for the support of all my decisions during these past six years, always believing in me and what I was doing.

To my friends, because they always were there for me to rejoice for my successes and to make me smile in the though moments.

To my university mates, because the quality of an experience is measured by the people you live it with.

To prof Fornaciari, Davide and the HIPEAC Research Group, because they made me realize how interesting is to do research.

A todos mis amigos y compañeros de Granada, porque han sido mi familia durante seis meses inolvidables.

Contents

List of Figures

List of Tables

Acronyms

NoC = Network-on-Chip
BW = Buffer Write
RC = Route Computation
VA = Virtual-Channel Allocation
SA = Switch Allocation
ST = Switch Traversal
LT = Link Traversal
LA = Link Allocation
CMP = Chip Multi-Processor
NIC = Network Interface Controller
VC = Virtual Channel
VNET = Virtual Network
NAVCA = Non Atomic Virtual Channel Allocation
SaF = Store and Forward
VCT = Virtual Cut-Through
WH = Wormhole
VCW = Virtual Channel Wormhole
PG = Power Gating
WU = Wakeup
IC = Incoming
RP = Router Parking
FM = Centralized Fabric Manager
aRP = aggressive Router Parking
cRP = conservative Router Parking
PE = Processing Element
APNEA = Adaptive Power gating for NoC buffers Power Aware
BET = Break-Even Time
APNEA R2R = APNEA Router-to-Router

APNEA N2R = APNEA NIC-to-Router

Chapter 1

Introduction

"Your work is going to fill a large part of your life, and the only way to be truly satisfied is to do what you believe is great work. And the only way to do great work is to love what you do."

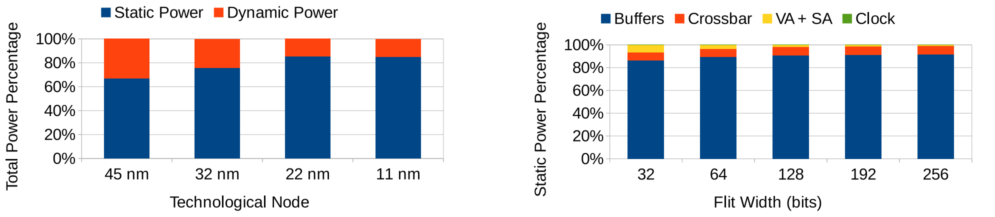
Steve Jobs

The continuous requirements for additional computational power makes the multi-cores the standard solution in current architectures. In particular, multi-core processors are used both in embedded and high performance systems due to their versatility, high performance and fine grain control over the chip power envelop. In this scenario, the increase in the number of cores in a single chip has encountered limitations in the use of a bus-based solutions, making the Networks-on-Chips (NoCs) an attractive interconnection paradigm [5].

However, its power consumption is not negligible compared to the overall chip [22], pointing out the need of a mechanism to save power. Furthermore, the continuous technology scaling highlights the leakage power as the major power source, while the dynamic power remains roughly constant [23]. All in all the NoCs are designed against the worst case scenario, thus spending a great portion of the time facing low traffic.

This requires the development of a methodology to save leakage power, without affecting the performance of the NoC. Power gating is a promising technique to achieve this goal, since it allows to switch an idle block off to almost avoid its leakage power.

The rest of the chapter is organized as follows. Section 1.1 gives an overview of the problem faced in this work. Section 1.2 states its goals and



(a) Total power breakdown vs technology node in NoC's router with flit width of 32 bits and injection rate of 0.3.

(b) Static power breakdown vs flit width in NoC's router at 45nm.

Figure 1.1: Power analysis in NoC's router done with DSENT. Figure 1.1(a) shows the different total power breakdown of a NoC router scaling down the technology node. Figure 1.1(b) presents how the static power breakdown changes varying the flit width.

contributions. Section 1.3 provides more background on Networks-on-Chips and power gating. Finally, the structure of the thesis is devoted in Section 1.4.

1.1 Problem Overview

Differently from the off-chip case, the power consumption is a key aspect in the design and the NoC total power has become a relevant part of the overall chip power, ranging from 10% to 28% [22]. This fact introduces the need for a mechanism to reduce the power in the NoC. Considering a NoC router, Figure 1.1(a) shows at 45nm the leakage power is about the 60% of the overall power consumption and scaling down to lower technology nodes this trend will be emphasized reaching up to the 85%. Data are extracted using the DSENT power model [32] targeting a baseline router as will be described in Section 1.3.

Leakage power needs to be the main target of power saving because of its great impact. However, it is important to decide the methodology granularity, namely a whole router or a single buffer. Again a breakdown of the leakage power in a NoC router is pointed out in Figure 1.1(b) considering 4 main components, respectively the buffers, the crossbar, the VA stage plus the SA stage and the clock which will be better discussed in Section 1.3. The major contribution is given by the buffers and the crossbar. Furthermore increasing the flit width both the Virtual-Channel Allocation (VA) and the

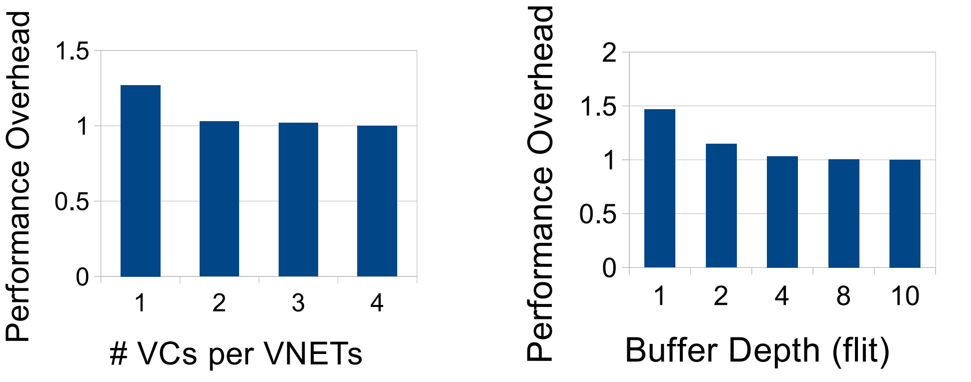
Module	Count	Static Power Percentage
4-flit VC buffer	36	78%
1-flit output latch	6	4.6%
6-to-1 CBMUX	6	3.1%
5-to-1 VCMUX	6	3.7%
Others	1	10.6%

Table 1.1: Static power breakdown from Ultra Fine-Grained [28]. The table presents the static power breakdown of the complete RTL router model synthesized at 65nm provided by [28].

Switch Allocation (SA) stages become more and more irrelevant, thus highlighting as buffers are a great target for saving static power. Data extracted from DSENT are consistent with the ones in Table 1.1 of the complete RTL model of the router at 65 nm provided by Matsutani et al. [28], thus from now on only DSENT will be used.

The first thought could be to remove some of the buffers from the NoC design or to shrink them to remove both leakage and area. Unfortunately, Figure 1.2(a) shows how performance degrade by removing too much channels. For example, the performance remain almost constant up to when 2 VCs are implemented for each VNET, but when the architecture has only 1 VC per VNET the performance degradation reaches up to the 25%. Figure 1.2(b) instead shows that even reducing the buffer depth affects the performance. The performance are quite constant until the buffer depth is 4 flits, but reducing it more the performance degradation can rise up to 45%. These analysis clearly state that there is a performance-power trade off on the buffers. They need to be implemented but leakage power can be saved using some techniques like power gating which will be explained in detail in Section 1.3.

The methodology granularity represents the second design choice. One option could be to save the whole static power given by the router targeting it as a whole, but problems arise when analyzing the opportunities of both router and input port level. Figure 1.3 and Figure 1.4(b) show how routers on average are in idle for less than the 40% of the time while, input ports on average are in idle for more or less 85% of the time, assuming in the finer level of buffers this percentage increases. Furthermore, Figure 1.4(a) demonstrate that on average even some ports in a router are used more than others, but



(a) Performance degradation vs number of virtual channels.

(b) Performance degradation vs buffer depth.

Figure 1.2: Performance analysis in NoC's router done using *qsort* [20]. Figure 1.2(a) shows how the performance varies with the number of virtual channels in the input port. Figure 1.2(b) instead shows how the performance is affected changing the buffers depth in the input port.

still the usage percentage remains limited.

It is easy to understand that at a finer level of granularity the number of opportunities to power off increases. Moreover, if a methodology that acts at router level can take advantage of idle periods of more than 10 cycles [11], the finer level granularity can take advantage of shorter idle times. Figure 1.5 for example shows that for a single flit packet traversing two consequent routers the buffer B_0 in the router R_0 can be put in sleep after the Switch Traversal (ST) stage, while the whole router R_0 can be put to sleep only after 6 cycles due to the credit back from the router R_1. Furthermore, Figure 1.6 shows the relative number of router power off opportunities as a small part compared to the input port ones. The important concept to note is that even if the power saving given by each buffer is low respect to the one given by the entire router, applying power gating at this finer level of granularity is the best decision, because of the much greater number of power off opportunities.

1.2 Goals and Contributions

The thesis aims to develop a power gating methodology to reduce the NoC leakage power by switching off the idle buffers in the router with minimal

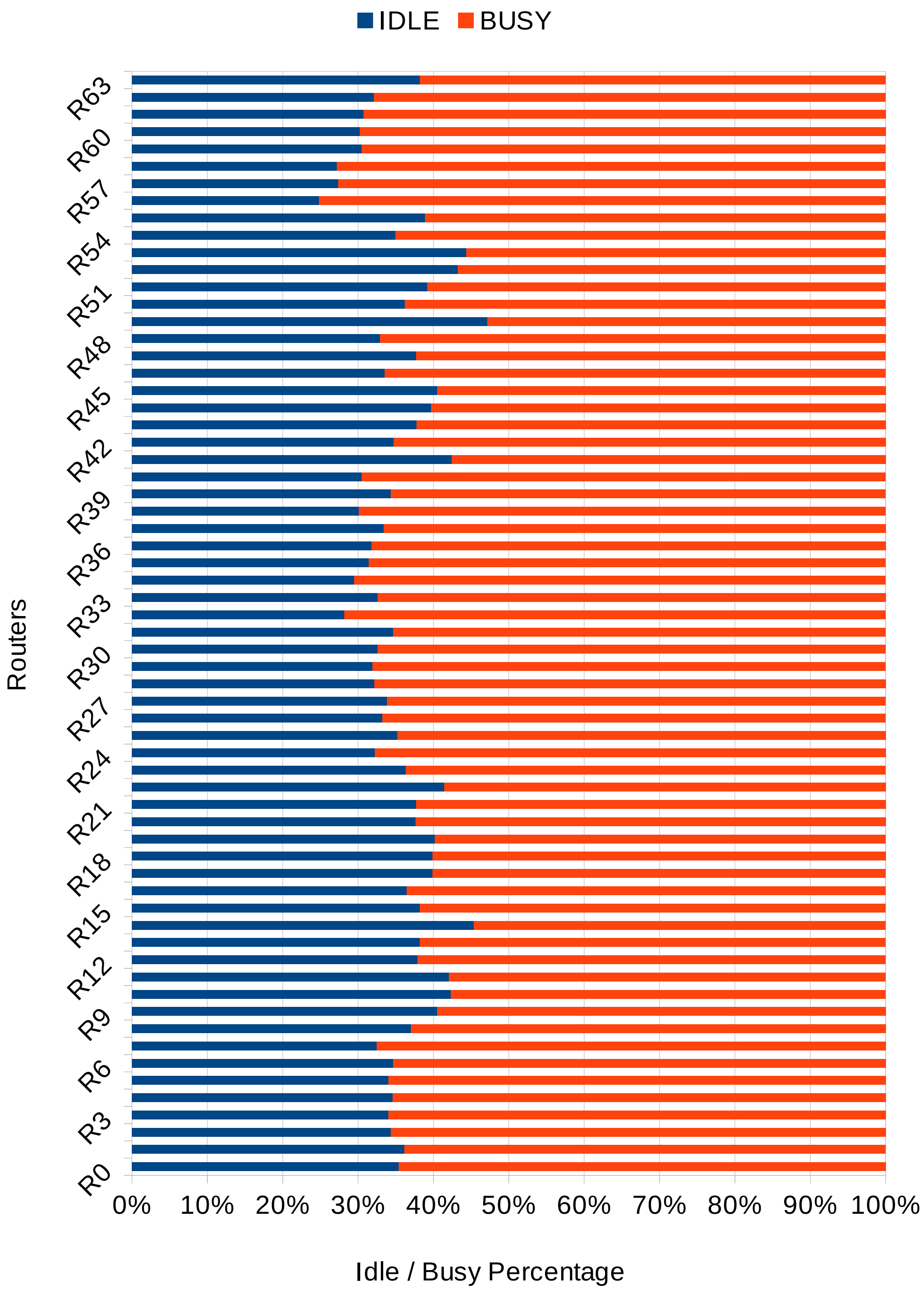
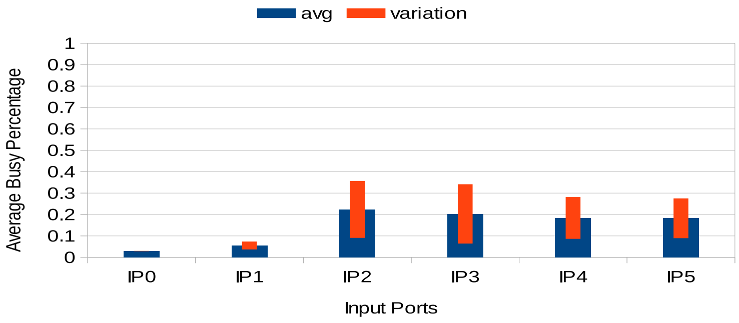
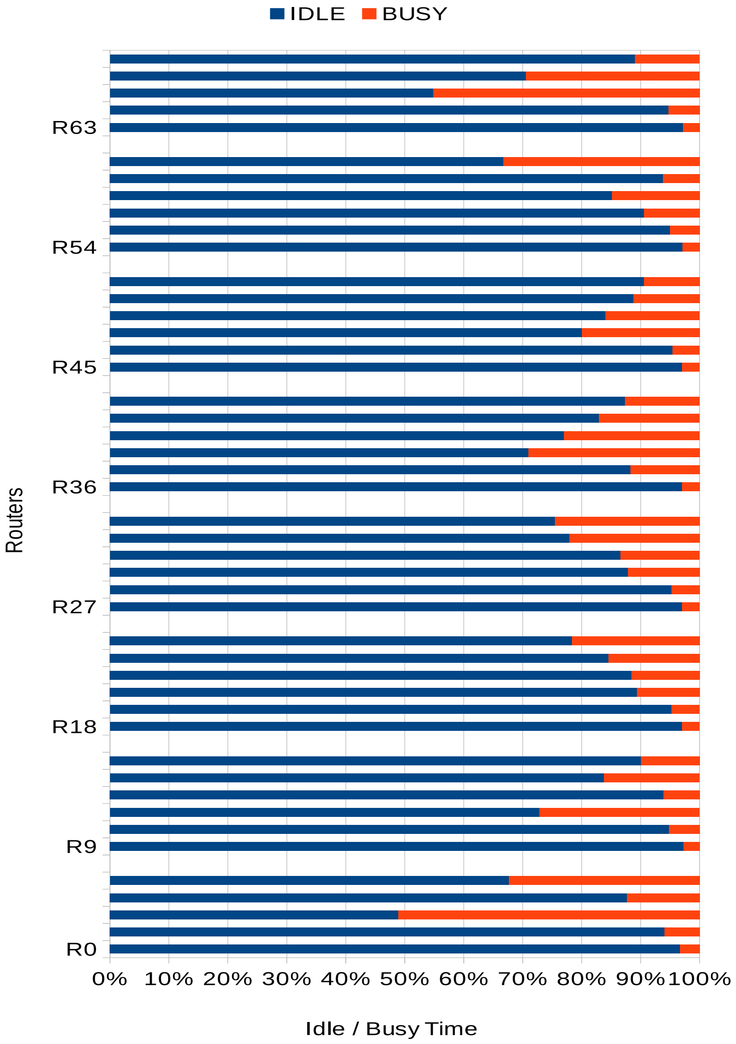


Figure 1.3: Idle/busy analysis in routers done with *ocean* [34]. The figure shows the percentage of time each router spent busy or in idle, namely with the possibility to power off.



(a) Average busy time for each input port in the 8x8 NoC.



(b) Idle/busy time in main diagonal router's input ports.

Figure 1.4: Idle/busy analysis in input ports done with *ocean* [34].

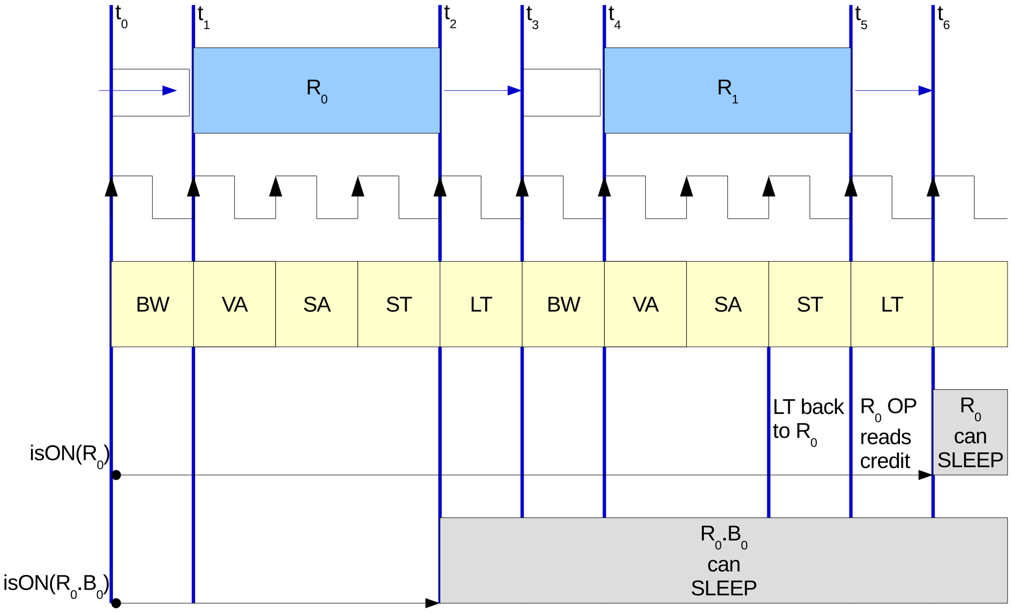


Figure 1.5: Router vs buffer sleep opportunities. There is represented a single flit packet traversing two consequent routers. Through the pipelines it is possible to understand when the upstream buffer and the whole upstream router can be turned off. B_0 can be switched off at t_2, while R_0 can be switched off at t_6.

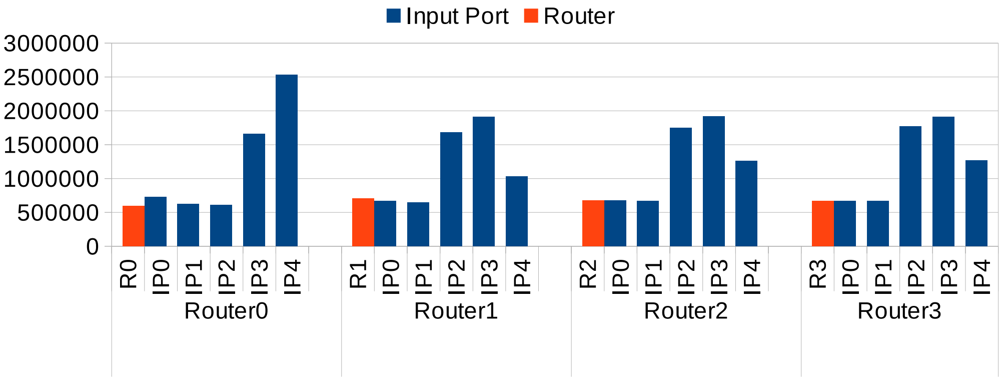


Figure 1.6: Power off opportunities comparison in routers and input ports done with *ocean* [34]. There are shown the different number of power off opportunities for the whole router and each of its input ports.

impact on the performance. This specific goal is reached proposing and developing some conceptual novel contributions that will discussed in Section 3.1.

A methodology has been developed and tested exploring part of the NoC design space to prove its effectiveness. The results show the real effectiveness and scalability of the methodology where 80% of the input buffers usage is saved on average. A total energy saving of up to 74% on average for the NoC router is obtained introducing negligible performance degradation.

1.3 Backgrounds

The Network-on-Chip - A NoC is a scalable and reliable interconnect that allows nodes to exchange data. A node can be both a CPU or a part of the memory subsystem. The NoC is composed by routers, links and Network Interface Controllers (NICs). Routers routes data into the network. NICs allow the communication between a node and a router. Links connect two routers or eventually a router and a NIC.

Figure 1.7 depicts a Chip Multi-Processor (CMP) which relies on a 4x4 2D-mesh NoC as interconnect. Routers take care of the communication between the various CPUs and memories. Each router is connected to some other through a link. For example R_{15} is connected to router R_{11} and R_{14} through two different links. R_{15} is connected also to a L1 and a L2 cache memories; L1 is also connected to the CPU. Communication between them is supported by the NICs.

From the communication view point the various nodes exchange messages, which typically are requests and responses of a coherence protocol. Traditionally, the NoC splits each message in multiple packets. Then each packet is eventually split in multiple flits to better utilize the NoC resources. These different levels of granularity are shown in Figure 1.8.

The NoC itself is characterized by some features: the layout topology, the routing scheme, the switching mechanism, the flow control mechanism and the router architecture. The NoC topology defines the way routers are interconnected to each others and how memory and CPU blocks are attached to the NoC. The most common topologies used in NoC are mesh [14], concentrated mesh [4], hybrid bus based [15] [33] and high radix [21]. In Figure 1.7 it is shown a 2D-mesh network where the routers are connected in a matrix form.

Given a specific topology the routing algorithm defines each source-destination

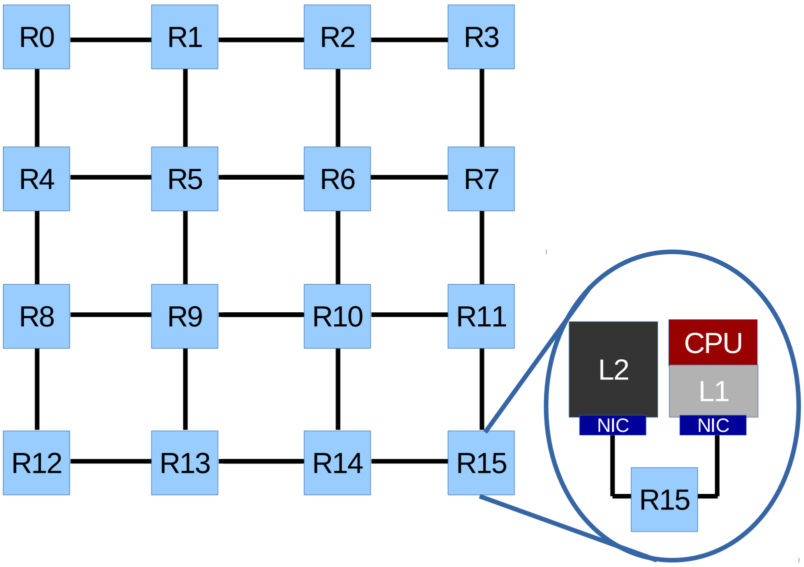


Figure 1.7: A multi-core based on NoC interconnect. It is pictured a 4x4 2D-mesh NoC. Furthermore, the focus shows how the core and caches are connected to the relative router.

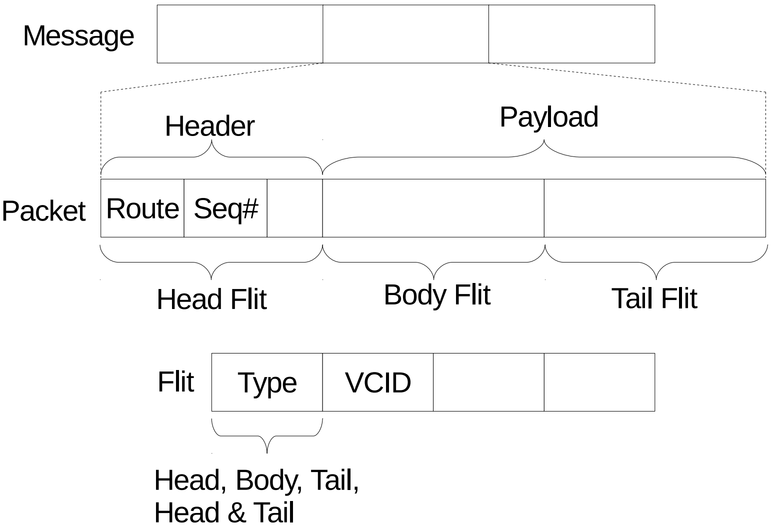


Figure 1.8: Message structure in NoC. There are presented the message, its division in packets and, furthermore, in flits highlighting the headers essential information.

path inside the NoC. Routing algorithms can be deterministic [4] or adaptive [19][17] [18], based on their capacity to alter the path taken for each packet. The most used deterministic routing scheme in 2D-meshes is XY routing where a packet first goes in the X direction and then in the Y one.

The switching mechanism is in charge of the transmission of the information between the input and output ports inside a router. Some switching mechanisms are Store-and-Forward (SaF) [26], Virtual cut-through (VCT) [25], Wormhole switching (WH) [29] and Virtual-Channel Wormhole switching (VCW) [13]. The most commonly used switching technique is VCW, which associates several virtual channels with a single physical channel, overcoming blocking problems of WH.

The flow control is a mechanism responsible of managing the advance of the flits between the routers. All the switching techniques that use buffers need a mechanism to communicate the availability of buffers at the downstream router. Three mechanism are used: Credit-based, Stop & Go and Ack/Nack [14]. The most commonly used in NoCs is the Credit-based one and with this mechanism every output port knows the exact number of free buffers and the number of slots available for every buffer in the downstream router.

A message provides the node-to-node communication, but when it trespass the NIC it is divided into NoC's basic data structure: the packet. A packet is considered split in multiple atomic transmission units called flits. The first flit of each packet is the head flit. A body flit represents an intermediate flit of the original packet while the tail flit is unique for each packet and closes the packet. There is another particular case of packets called single flit packets and they are composed by a single head/tail flit.

After this general explanation of the NoC it is now presented in Figure 1.9 presents the baseline architecture of a wormhole NoC's router that is considered in the thesis.

A wormhole router supporting VCs and VNETs is considered. It is a standard 4-stage pipeline, i.e. Buffer Write/Route Computation (BW/RC), Virtual Channel Allocation (VA), Switch Allocation (SA), and Switch Traversal (ST). A single cycle for the Link Traversal (LT) is assumed as shown in Figure 1.10. The router implements non atomic VC allocation (NAVCA). A channel can be allocated in NAVCA mode if it is already used by another packet but such packet has its tail flit already stored in the buffer, thus a new packet of the same type, i.e. same VNET, can be stored in the same

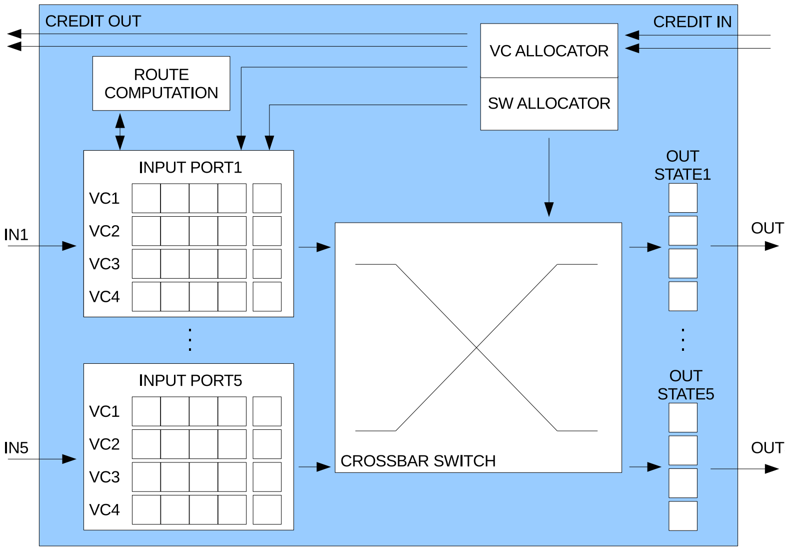


Figure 1.9: The baseline router architecture. It is presented a virtual-channel, wormhole, credit-based router highlighting its main components and how they are connected.

buffer without introducing flit mixing.

The considered NoC implements the VNET mechanism to support the coherence protocol, preventing the traffic from a VNET to be routed on a different one. When a head flit arrives to the input port of the router it has to pass through the 4 pipeline stages before traversing the link. First, it is stored in the VC buffer (BW) which has been reserved by the upstream router, and the output port is computed (RC). Then, it competes in the VA stage to reserve an idle virtual channel in the selected output port. Notice that assigned VCs belong to the set of VCs associated to the VNET of the

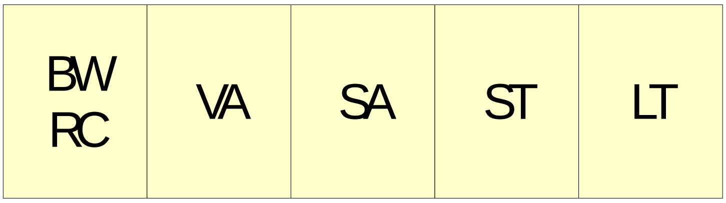


Figure 1.10: The baseline router pipeline. The figure shows the baseline NoC router 4-stages pipeline with the addition of the link traversal.

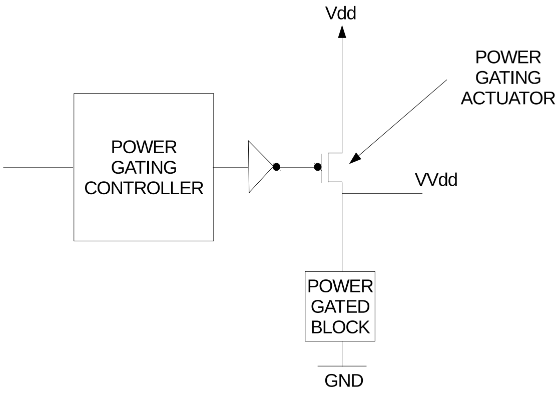


Figure 1.11: Power gating from a block diagram view point. The figure shows the main components of a power gated component from the block view point: the component itself, the controller and the actuator.

packet. Once the VA succeeds, head, body and tail, competes in packet order for a crossbar switch path (SA). Finally, each winning flit in the SA has to pass the ST and LT before reaching the next router. Note that tail and body flits pass through fewer stages since they reuse resources and information reserved by the head flit (i.e., VC and RC).

The NIC is another NoC component that provides communication between the nodes and the network. The NIC wraps up the requests from the cores as messages suitable for the NoC and rebuilds them at destination. When a message is taken from the NIC queue it is split into packets and each one of them is divided into flits. Then the whole flitisized packet is allocated in a VC. Here all the flits will compete for the link allocation and then they will be sent to the next router.

The power gating support - Power gating is a technique to reduce leakage power consumption by shutting off a block of the circuit that is not in use. A block is in active state when it is functioning, while it is in sleep state when it is shut down. Figure 1.11 shows that power gating technique is composed of two principal parts: the power gating controller and the power gating actuator, or sleep network. The first decides the logic to turn off and on the power gated block of the system, while the second is in charge of the effective actuation.

Power gating is an effective mechanism because it reduces the static power when the block is turned off. Figure 1.12 shows an example activity for a system with power gating implemented. It is worth to notice that when the

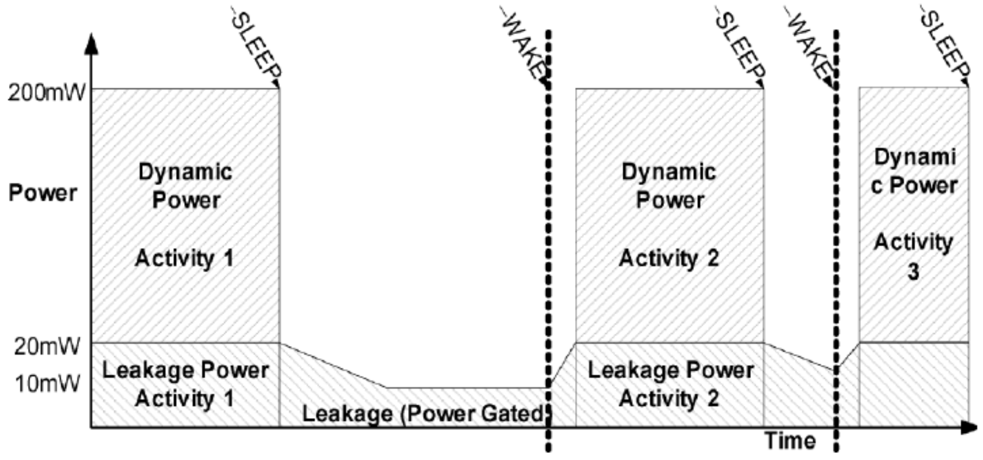


Figure 1.12: Power profile of a power gated component from Low Power Methodology Manual [24]. The figure shows how power gating a component affects on both the dynamic and the static power consumption during the time.

system is active it has both dynamic and static power consumption, while when power gated only a reduced static power is present that it the leakage of the sleep network. Moreover the wakeup of the power gated system is not instantaneous but it has a delay defined as T_{ON}. Even the sleep of the power gating system has a delay defined as T_{OFF} which usually is negligible.

1.4 Thesis Structure

The rest of the thesis is organized in 5 chapters. Chapter 2 describes the state of the art in power gating applied to NoCs. Chapter 3 provides a detailed description of APNEA methodology and its novel contributions. Chapter 4 details the validation the methodology providing the results of synthetic traffic and real applications. Chapter 5 points out some future works on APNEA methodology.

Chapter 2

State of the Art

> *"A people without the knowledge of their past history, origin and culture is like a tree without roots."*
>
> Marcus Garvey

The power gating mechanism has been extensively exploited in the NoCs to reduce the leakage power [31, 11] or to improve the reliability of some part of the architecture [37, 38] or both [35]. This chapter summarizes the state of the art considering the power gating actuator to reduce leakage power consumption in current computer architectures. In particular, some methodologies are focused on power gating in the processing unit of the CPU and other are focused on the NoC. In the latter case, two different granularity levels are investigated: the router level and the buffer one.

Lungu et al. [27] show that a simple power gating policy on execution units with predictive control is not effective because the mispredicíton can lead to large increase in the energy consumption, so it proposes a guard mechanism to prevent power overruns.

Annavaram [2] highlights the problem of applying power gating to CMPs: performance degradation and negative power saving. Zhigang [23] studies power gating opportunities in execution units and finally different techniques to detect opportunities for entering sleep mode are developed. Results on the floating-point units show they are sleeping the 28% of the time with only the 2% of performance loss, while fixed-point units sleep the 40% of the time with the same performance loss.

All of these studies show that power gating is an effective technique to save leakage power in microprocessors at different levels of granularity but its control must be carefully designed to be effective.

The rest of the chapter discusses power gating at router level in Section 2.1 and power gating at buffer level in Section 2.2.

2.1 Power Gating at Router Granularity

The possibility to completely switch-off a router allows great power saving. The observation comes from the NoC design that considers the worst traffic scenario, leaving the NoC underutilized for the majority of the time.

NoRD [9] improves previous power gating methodologies focusing on the node-router decoupling. A bypass inport and outport are connected to the NIC to allow the communication even when the router is powered off so the communication does not suffer delays. To improve the topology all the bypass lines are connected to each others to create an unidirectional ring providing communication even if the routers are powered off. Thus if a flit arrives in a turned off router, it is directed from the bypass inport to the NIC or the bypass outport. Power on and off mechanisms use PG (power gating) and WU (wakeup) and IC (incoming) signals. The latter prevents power off when new flits are directed to the router. A router can power itself off when it is empty and both WU and IC signals are deasserted, while the WU signal is managed with a threshold-based mechanism monitoring the number of VC requests on the local NIC. Moreover, the methodology states that some routers affect performance more than others, thus to gain advantage a distinction between performance-centric and power-centric routers is introduced. The distinction is based on the bypass placement and the topology and their control differs on the wakeup threshold used. With this last asymmetric wakeup mechanism routers which affects more the performance has a lower threshold. NoRD depends on the placing of the bypass ring (bypass lines are expensive).

Router Parking (RP) [31] is a methodology to power save with minimum performance loss by parking some routers associated with sleeping cores and proactively aggregate traffic to the active routers. A new component, the centralized Fabric Manager (FM), is added to the architecture and it collects information about the traffic in the NoC, then choosing a subset of routers to park among the ones in which the core is powered off and update routing tables for all the routers. RP is a epoch based technique because the data are collected from the FM every X cycles, so the choice of this parameter affects a lot performance overhead and power saving. An aggressive and a conservative policy to select the set of routers to power off are explored. The aggressive one (aRP) increases the latency of the network allowing more routers to be parked while the conservative one (cRP) parks less routers in order to achieve a lower latency. The final addition to the methodology is a mechanism to switch among the two policies based on the traffic in the network to obtain the best features from both the policies. The limits in the methodology are the missing opportunities of power saving when the core is not powered down and the complex design which needs to modify the traffic to gain power off opportunities.

MP3 [10] is a methodology to power gate routers in Clos networks reaching minimal performance overhead. Clos networks are indirect networks where there is an input layer of router connected to the PEs (which are sources), an output layer of routers connected to the PEs (which are destinations) and some layers between them to provide a robust communication between the input and output layers. Thanks to this structure, Clos networks have far more paths from a PE to another compared to Mesh networks and traffic can be diverted decreasing utilization of each router and allowing to power gate routers in an effective way. MP3 maintains a set of routers powered on such that all the PEs in the network remain connected. Moreover MP3 selects some components, like input/output ports, in active routers which are not needed and shuts them down to save more power. By monitoring the traffic it can be decided which routers need to be powered on and off. This control is done using an handshake protocol where a routers asserts the power gating (PG signal) and then the upstream asserts the wakeup (WU signal) when needed. To prevent performance losses a rapid wakeup mechanism is used, chaining signals to let the downstream router to be awaken when needed. Even if this work does not introduce overheads and saves more than 47% of static power, it is designed for a particular case of networks (Clos) and uses a threshold which needs to be determined empirically.

Power Punch [11] methodology promises non blocking power gating in NoC routers. It tries to completely hide router wakeup latency of 8 cycles, sending wakeup signals up to 3 hops in advance, adding the capability of hiding a variable number of cycles depending on the router's pipeline. The huge quantity of signals that need to be received by a router (one for each router at a maximum of 3 hops distance) is compacted sharing wire channels for wakeup signals and using routing and topology information. With a 3-hop wakeup signal it can be shown that 5 bits are required on X direction and 2 bits are required on Y direction considering XY routing. Moreover as the wakeup latency is completely hidden the overhead in terms of performance is minimum. However, the methodology is tightly coupled with the mesh topologies and the deterministic routing algorithms.

Catnap [16] splits a single Network-on-Chip into four smaller physical on-chip networks (sNoCs), that combined are constrained to have the same bisection bandwidth of the original one. Each sNoC fully support the coherence protocol allowing to drive all its message types, thus implementing all the required Virtual Networks (VNETs). The methodology applies power gating to each sNoCs at the router granularity. The rationale is the impossibility to completely switch-off a router even at low traffic if a single NoC is used without affecting the implemented routing algorithm. However, providing multiple physical networks Catnap can safely switch-off the router in the unused sNoCs. Each sNoC has an ID and Catnap steers the traffic to the lower ID sNoCs first. Two different aspects complete the methodology: which sNoC to use and when a router is switched on and off. The policy is regulated by the Network Interface Controllers (NIC) that decides the sNoC where each packet is injected. Once injected a packet cannot change the sNoC. The NIC decides the sNoC to be used based on two different metrics, i.e. local and regional congestion metrics. Local congestion metric signals to a NIC a congestion in a specific sNoC by monitoring the input buffers of the NIC directly attached router. If an upper threshold is exceeded the NIC inject the next packet in the sNoC with the immediately higher ID than the one that is in use (waking up the new sNoC router before). The regional congestion metric is built considering the 8x8 2D-mesh used for Catnap assessment split in 4 4x4 quadrants. Each quadrant implements a single bit H-TREE OR-based hardware structure per each sNoC. Considering a specific sNoC, each router in a quadrant can set the bit depending on its local congestion metric. All the NICs in the quadrant can read the bit and if it is set deliver the new packets to a sNoC with higher ID. The congestion detection is completed with a threshold-based mechanism that decives when reset or set both regional and local congestion bit to steer the selection of the sNoC to be used in the near future by the NICs. A router can be switched-off if the regional congestion bit of the sNoC with immediately lower ID that the sNoC that owns the considered router is not set. Moreover, a switched-off router is waked up if the same regional congestion bit is set. A lookahead mechanism completes the wake up policy allowing routers in the same sNoC to send wkae up signals to downstream ones to shadow the switch-on latency.

DarkNoC [7] proposes a NoC architecture to take advantage of the dark silicon phenomena in its architecture. DarkNoC organizes the interconnect in multiple physical networks. Each NoC is optimized to work at a specific frequency. This kind of architecture can improve drastically the power saving compared to a single physical NoC. The first novel contribution of this methodology is the new architecture proposed where the NoC is divided into regions, each one working at a specific Voltage-Frequency (VF). Each region is connected to the others by bisynchronous links. Moreover each node contains a number of network layers, all optimized at design time to operate in a certain VF. At a given time only one layer is active (illuminated) while the other are disabled (dark). There is a darkNoC Layer Manager (dLM) which decides in each region which is the layer that needs to be illuminated and which need to be dark. These layers share the same set of links to reduce costs. To manage the router stack there is a darkNoC Router Manager (dRM) which provides the switch functionality directed by the dLM. The switch mechanism between two VFs in a determined region is a critical mechanism coordinated by the dLM and the dRMs. The dLM controls it through a NIC injecting control flits for the dRM. The phases of the switch are:

- disable the traffic injection;
- power on the dark router and flush the traffic on the active one;
- enable the ports of the dark router while disabling the ports of the active one;
- power off the active router;
- enable the traffic injection again.

Several works exploits power gating at router level to reduce the leakage power consumption. However, considering moderate and high traffic or even low traffic coupled with a uniform traffic distribution, greatly impact the effective idle time for each router. This makes the use of power gating impractical at router level, as discussed in Section 1.2. Moreover actuating at a finer level increases power off opportunities, so a different approach to power gating has been studied and will be explained in the next section.

2.2 Power Gating at Finer Granularity

Power gating at finer granularity has been proposed to overcome the limitations at router granularity. Particularly interesting is the case when such a technique is applied to input buffers, allowing to shut down a portion of the input port. A strong motivation is the greatest portion of the leakage power, which comes from the input buffers.

Ultra-fine-grained run-time power gating [28] discusses a power gating methodology considering a single VC-based Network-on-Chip, where different parts of the same router can be selectively and dynamically power gated to reduce leakage power. It organizes the router in micro power domains that can be independently power gated: each buffer, the crossbar, the VA and the SA. A critical contribution of the work is the complete RTL design at 65nm. In particular, the wakeup time for a 128bit 4 slots buffer is estimated to be 2.8ns. This result is aligned with [39] where a 4-slots 32bit buffer is waked up in 1ns. The wake-up policy relies on look-ahead signals between routers to shadow the VC wake-up latency. To avoid performance penalties, a new FIFO buffer model has been developed. Part of each buffer is always on while the second part of each buffer can be waked on demand.

FlexiBuffer proposes a power gating solution for NoC router buffers considering ultra power domain at single flit granularity. The core of the proposal relies on a novel FIFO buffer design to efficiently switch-on and off single buffer slots without imposing a circular buffer utilization as in traditional FIFO designs. However, due to the implementation overhead the final solution is to split each buffer in two parts. An always on part with few slots, able to face small traffic, eventually linked to a bigger second-half buffer that is switched on when the number of flits stored in the first half of the buffer exceeds a certain threshold. A similar solution with double threshold control has been proposed by Casu et al. in [8] achieving zero performance penalties.

Centralized Buffer Router [21] with elastic links and bubble flow control discusses a mixed approach between bufferless and buffered router depending on the traffic. It uses a bufferless scheme at low traffic, while a CB buffer is introduced when the traffic intensity increases. A simple policy is exploited to save energy in the CB buffer: after 10 cycles where it is not used, the CB turns off and after a number of cycles where a flit stalls it turns it on. This number is subject to exploration and it is a important parameter for the performance/saving trade off.

Panthre [30] exploits routing reconfiguration to optimally use power gating and selective datapath structures in the NoC router. A single VC-based NoC is considered using XY for the baseline and Up/Down routing function for Panthre implementing LBDR. The critical innovation is the possibility to reconfigure the routes without stop network operations, selecting a new configuration which complies some all-powered-ON turn restrictions. Turn restrictions are limitations added in the topology to construct a spanning tree on the network. The methodology relies on the possibility to use alternative paths between the same source-destination pair without inducing network-level deadlock. In particular, congestion and low utilization metrics are used to decide if a datapath segment in the router can be switched on or off, respectively. Panthre is an epoch based proposal using thresholds A_{th} which represent the threshold activity of a single datapath. At each epoch three decisions are taken: switchoff/switchon router parts, update the switchoff threshold and complete router shutdown. Usage counters are implemented for each power domain in the router, i.e. 4 input and 4 output domains. If at the beginning of an epoch the A_{th} threshold is exceeded the power domain is not switched off. If at the beginning a congestion bit is set the A_{th} is decreased (by 128) to prevent too aggressive power gating. The congestion bit is set based on the deroute information and the local congestion metric, i.e. the utilization of the routers' input buffers (it is global). If no congestion is set for an X number of epochs the A_{th} is increased by a Y value (128). After Z (10) increases the policy is reset and rerun since the NoC load has changed too much. Panthre is an interesting methodology but its design is really complicated and it seems to have too many thresholds in its design to tune.

Chapter 3

APNEA Methodology

"We can only see a short distance ahead, but we can see plenty there that needs to be done."

Alan Turing

The chapter presents APNEA[1], a control inspired methodology which powers gate input buffers in the NoC routers to maximize the static power saving, while ensuring almost the same performance of the reference NoC.

The methodology is able to power off all the input buffers in a router, whether the input port is connected to a router or a Network Interface Controller (NIC). In particular, the methodology leaves a physical buffer powered on to overcome the performance degradation introduced by the power gating.

The proposal relies on two high level concepts, namely the upstream and the downstream modules. The upstream module is the one which generates the traffic to the downstream one. The downstream module receives the traffic and physically manages the resources under control, i.e. the ones that can be power gated. Considering the Network-on-Chip, a router can always be considered as a downstream module, where its input buffers are the resources to be controlled. On the other hand, a router input port can be connected either to a NIC or to another router; both of them are identified as the upstream modules for the considered router.

It is worth noticing a router can be both a downstream and an upstream module depending on the observer viewpoint. For example, Figure 3.1 highlights router R_0 as the downstream module for NIC_0 and router R_1, since

[1] Adaptive Power gating for NoC buffers Energy Aware

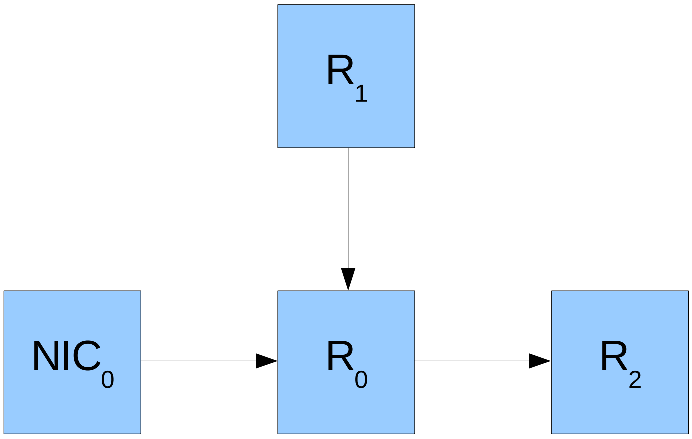


Figure 3.1: General overview of upstream and downstream modules. The figure shows how different modules are connected and the component role from different view points.

NIC_0 and R_0 are both connected to one of its input ports. At the same time, R_0 is the upstream module for router R_2, since one of its output ports is connected to one of the input ports in R_2.

The APNEA methodology defines a power gating policy split into two parts. The *physical actuation* is done in the downstream router, since the actuation is on its buffers, while the upstream module is in charge of the *control and command dispatch*, namely signaling to the downstream module to eventually switch-off or switch-on a buffer.

The router model has been modified to support the methodology, always aiming to keep the changes minimal. Another key aspect to note is that APNEA is an additional completely transparent control layer placed on top of the flow control scheme. Moreover, it can not steer the system towards a not feasible or a deadlock configuration.

It is worth noticing that the upstream and downstream modules are defined as a router or a NIC. Moreover, as explained in the rest of the chapter the APNEA policy is implemented between an output-input port pair. In particular the upstream module implements the *control and command dispatch* part of the policy in its output ports, while the downstream router implements the *physical actuation* part per each input port. To this extent a router implements one instance of the upstream module policy per each output port and one instance of the downstream module policy per each input port. Moreover, the NIC has only an implementation of the upstream module policy in its output port to the local router.

The rest of the chapter is organized as follows. Section 3.1 presents the key concepts on which the methodology relies. Section 3.2 discusses the APNEA *router-to-router* model and Section 3.3 presents the APNEA *NI-to-router* model. Finally in Section 3.4 some implementation details and notable scenarios are provided.

3.1 APNEA Methodology: Key Concepts

This section describes the main contributions of the APNEA methodology, that also represents the key concepts under whom the methodology has been developed. They allow to describe the proposal in a more abstract way.

Flow balance assumption - Informally a state for the router's output port is identified as how many channels are currently used. SA requests are used to identify the number of channels currently used, while VA requests are used to identify the number of channels that will be used in the future. The SA can be seen as the present state and the VA requests as the future one. In this perspective, the number of active buffer in an input port can be adapted considering these two quantities. For example, when there are more VA requests than SA requests it means that the traffic flow is increasing and there are less channels than needed. Thus, not all of the VA requests can be granted on time unless additional buffers are switched on for that output port. On the contrary, when the VA requests are less than the SA requests, all of them will be granted hinting that the traffic flow is decreasing and there are more channels than needed.

Late binding - In the NoC, the virtual and physical allocation of a buffer to a packet happens at arbitrarily distant moments, since the VA in the upstream router happens several cycles before the BW in the downstream one. The late binding is a mechanism to decouple virtual allocation of the channel from physical binding, eventually remapping a virtual channel into a different physical buffer (see Section 3.2.2 for a more detailed discussion of the buffer remapping idea). This is a key concept in our methodology because it allows to optimize the physical allocation in the buffers and thus to minimize the number of power on events.

Sliding windows - with sliding window it is introduced a mechanism,

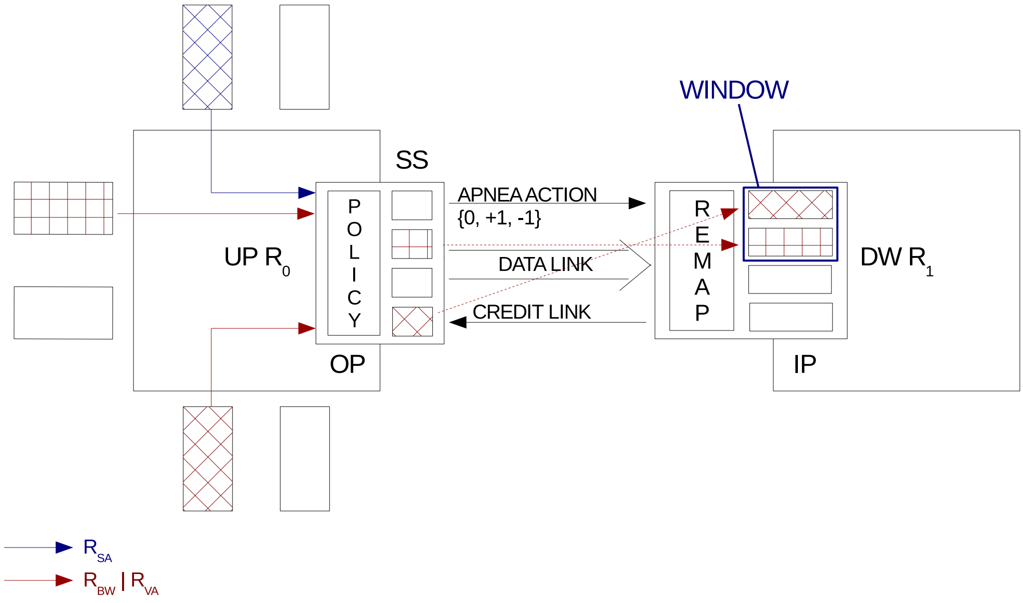


Figure 3.2: General overview of APNEA router to router architecture. An upstream and a downstream modules are pictured highlighting the differences introduced by the APNEA methodology.

adaptive by construction, in which the port can adapt itself to the actual traffic, namely allocating or deallocating channels like a sliding window or a shrinking pipe, without caring of where the traffic will be physically allocated.

3.2 APNEA Router to Router Model

This section describes the APNEA policy between a pair of routers, where one is considered as the upstream and the other one as the downstream. In the downstream one the focus is on the buffer, while in the upstream one the focus is on the traffic injector and control roles.

Figure 3.2 highlights a modified router architecture considering an upstream and a downstream router pair. R_0 has 3 packets in its input ports directed to R_1. The red ones are already in SA stage, while the blue one may be in BW or VA stages. R_0's output channels are marked when there is a packet allocated on them. The red channels are the already allocated ones, while the others are not. Given that, the R_0's policy has to decide whether to request for a new channel or wait for an already ON channel to become free. This decision is signaled to R_1 through the action signal. The packets allocated in R_0 on a certain channel may be physically allocated on a different input buffer in R_1 thanks to the remapper, which is placed before R_1's input buffers to guarantee the late binding. Moreover, the remapper module makes the late binding mechanism transparent to the upstream router. Figure 3.2 also provides a view of the **sliding window**, clearly visible in R_1's input port, as the set of turned on input buffers.

3.2.1 Upstream Router

The upstream router is the *controller and command dispatcher* of the APNEA and works per each output port of the router. Considering a single output port, it senses its traffic and it forces to enlarge or reduce the window, i.e. the number of buffers in active state.

The upstream router works on the quantities (defined below) to resize the window of a specific OP. All the required information are reported per each VNET, since each VNET experiences a specific traffic pattern and a buffer can only store packets from a single VNET.

$R_{BW_{j,i,t}}$ defines the requests in BW stage directed to an OP j for VNET i at time t. $R_{VA_{j,i,t}}$ defines the requests in VA stage directed to an OP j for VNET i at time t. $R_{SA_{j,i,t}}$ defines the requests in SA stage directed to an the OP j for VNET i at time t. They represent a characterization of the traffic the output port is experiencing. The requests in BW and VA can be viewed as incoming traffic, while those in SA as active traffic.
Moreover, $W_{OP_{j,i,t,used}}$ defines the number of VCs used in OP j for VNET i at time t, $W_{OP_{j,i,t,idle}}$ defines the number of VCs in idle in OP j for VNET i at time t and $W_{OP_{j,i,t,NAVCA}}$ defines the number of VCs allocated in Non-Atomic Virtual Channel Allocation (NAVCA) mode in an OP j for VNET i at time t. Last

$$W_{OP_{j,i,t,usable}} := W_{OP_{j,i,t,idle}} + W_{OP_{j,i,t,NAVCA}} \tag{3.1}$$

defines the number of VCs usable in an OP j for VNET i at time t, which are the channels in idle or used channels where it can be allocated in NAVCA mode.

The policy developed is a two stages decision policy. First, a local decision is taken concerning a single VNET. Then, a global policy takes all the local policy results to obtain a global decision for the OP.

Algorithm 1 shows that a local decision to shut down a channel is taken ($R_{j,i,t+1} \leftarrow -1$) when the number of usable VCs in the OP j for the VNET i

Algorithm 1 Local decision for outport j and VNET i at time t+1

```
$R_{j,i,t}$ := Local decision in OP $j$ for VNET $i$ at time $t$

if $W_{OP_{j,i,usable}} > 0$ then
    if $R_{BW_{j,i,t}} + R_{VA_{j,i,t}} \leq R_{SA_{j,i,t}}$ then
        $R_{j,i,t+1} \leftarrow -1$
    else
        $R_{j,i,t+1} \leftarrow 0$
    end if
else
    if $R_{BW_{j,i,t}} + R_{VA_{j,i,t}} > R_{SA_{j,i,t}}$ then
        $R_{j,i,t+1} \leftarrow +1$
    else
        $R_{j,i,t+1} \leftarrow 0$
    end if
end if
```

is greater than 0 and the incoming requests are at most equals to the active ones. This means the traffic is decreasing or it is constant and some active channels are unused. Conversely, a local decision to turn on a channel is taken ($R_{j,i,t+1} \leftarrow +1$) when the number of usable VCs is at most 0 and the incoming requests are greater than the active ones. This means the traffic is increasing and there are not enough channels to steer it. Otherwise, a local decision to maintain the same number of channels is taken ($R_{j,i,t+1} \leftarrow 0$).

The global decision policy for a specific output port is the mechanism to merge the local decisions from all the VNETs into a single one as reported in Algorithm 2.

The global decision depends on the local decisions and prioritizes turn on local decisions to maintain the system performance. Hence, whenever a local decision requests a turn on, the global decision will request a buffer switch on too. Conversely, the global decision requires to turn off a buffer if no turn on is coming from all the local decisions, at least one local decision requires a turn off and it has an idle buffer.

The upstream router is in charge to select the target VC, based on the global decision that has been taken. This solves the scenario where multiple local decisions require to switch on a new buffer. For example, Algorithm 3 describes how to choose a target channel to be powered on. The idea is to

Algorithm 2 Global decision for OP j at time t+1

```
$W_{total_{j,t}}$ := Window size in OP $j$ at time $t$
$R_{j,t}$ := Global decision in OP $j$ at time $t$

$W_{total_{j,t+1}} = W_{total_{j,t}} + R_{j,t+1}$

if |( $R_{j,i,t+1} == +1$ ) then
    $R_{j,t+1} = +1$
else
    if |[ $(R_{j,i,t+1} == -1) \wedge (W_{OP_{j,i,idle}} > 0)$ ] then
        $R_{j,t+1} = -1$
    else
        $R_{j,t+1} = 0$
    end if
end if
```

Algorithm 3 Channel selection for power on in OP

```
for all $vnet$ in $outport$ do
    if $R_{outport,vnet,t+1} == +1$ then
        for all $vc$ in $vnet$ do
            if $vc ==$ OFF then
                return $vc$
            end if
        end for
    end if
end for
return $-1$
```

select a powered off channel from one of the VNETs which requested a turn on.

Algorithm 4 Channel selection for power off in OP

```
$targetVC \leftarrow -1$
for all $vnet$ in $outport$ do
    if $R_{outport,vnet,t+1} == -1$ then
        for all $vc$ in $vnet$ do
            if $vc ==$ ON $\vee$ $vc ==$ IDLE then
                $targetVC = vc$
            end if
        end for
    end if
end for
return $targetVc$
```

Similarly, Algorithm 4 shows the upstream router that needs to select a target channel to shut down, because a global decision to turn off has been taken. Then, a powered on, idle channel is selected from one of the VNETs which requested a turn off.

The upstream sends the decision to the downstream router (see for a detailed discussion Section 3.2.2).

The methodology is optimized to always keep a buffer in ON state per output port. It has been noted this greatly reduces the performance penalties that a power gating methodology can introduce. A symmetrical change is needed when waking up the first channel in upstream, since there is no need to communicate the wakeup to the downstream router.

3.2.2 Downstream Router

The downstream router is the physical actuator of the methodology. This means it has to perform the power on and off of the physical buffers based on the signals received from the upstream router. It also provides the late binding mechanism to optimize the buffer switching activity.

A remapper able to allocate an upstream virtual channel on a different physical buffer is introduced to provide late binding, as shown in Figure 3.2. A remapping table where the virtual channel is associated to the physical buffer is added as support. The binding starts when the first flit of the packet arrives at the downstream router for the BW and it ends up only when the physical buffer is idle and a power off request is received. A remapping policy is defined and explained in Algorithm 5.

The upstream router targets directly a virtual channel to power off and needs to communicate the targeted channel to the downstream to remove the correct remapping from the remap table.

Algorithm 5 Channel selection in remapper

```
for all $vc$ in $inport$ do
    if $vc$ not remapped $\lor$ $vc$ == ON then
        return $vc$
    end if
end for
```

When a flit directed to a not remapped channel arrives, the remapping policy finds a not remapped powered on channel. Then, this channel is selected to remap the one the packet should use. The remapping policy compacts the traffic in the lower id buffers leaving the others almost idle. The physical buffer is now independent from the VC assigned to each VNET due to the remapper. The remapper can always find a physical buffer to remap a new channel, by construction.

It is worth noticing this strategy is designed to reduce the leakage power consumption, while different remapping policies can be used to achieve different objectives, i.e. fault tolerance and reliability. The downstream router is the power gating actuator of the buffers. This functionality relies on the action and target channel signals sent by the upstream module.

Whenever a request for a new buffer is received, the policy needs to target a new physical buffer to power on. The targeted physical buffer is decided by Algorithm 6. It is always assured that whenever a new channel is requested a physical buffer will be available because, as said in Section 3.2, the decision policy works always in feasible states of the router. Furthermore, the buffer to be turned on is chosen as the lowest id, powered off buffer.

The same reasoning is applied when the downstream router receives a request to turn off a channel. The power gating actuator needs to target a specific physical buffer among the ones that are already on or that are turning on. Algorithm 7 details the implemented policy, which prioritizes

Algorithm 6 Channel selection for power on in input port

```
for all vc in inport do
    if vc == OFF then
        return vc
    end if
end for
```

the buffers that are turning on as this can improve the saving because the buffer will not completely power on.

Algorithm 7 Channel selection for power off in input port

```
for all vc in inport do
    if vc == OFF_TO_ON then
        return vc
    end if
end for
for all vc in inport do
    if vc == ON then
        return vc
    end if
end for
return $-1$
```

The target buffer is chosen as the lower id turning on buffer and, if there is no such buffer the lower id powered on buffer is chosen. It is not possible that there is no channel to target, since the policy never reaches an unfeasible state.

3.3 APNEA Network Interface Controller to Router Model

This section describes the APNEA model between a NIC and a router, where the NIC is considered the upstream and the router the downstream.

The NIC is a traffic injector to the downstream router in which the buffers will be power gated. The same APNEA policy is used to steer decisions from

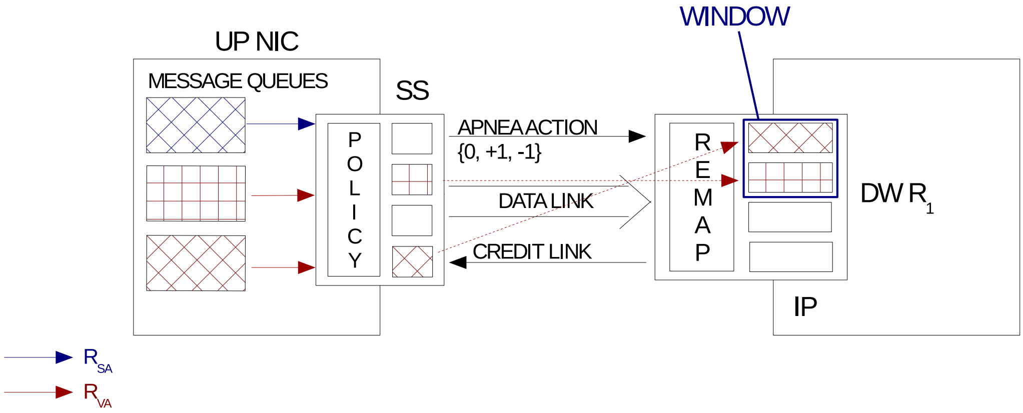


Figure 3.3: General overview of APNEA NIC to router architecture. An upstream NIC and a downstream router are shown pointing out the differences introduced by the APNEA methodology.

the NIC side, while different quantities are defined for the policy due to the different structure the NIC has with respect to the router.

Figure 3.3 highlights the modified architecture for the NIC and the router considering a NIC-to-router model. There are an upstream NIC_0 and a downstream R_1. Since the downstream module is identical to the one presented in Section 3.2, the focus will be on the upstream NIC. Note than the minimum number of queues is imposed by the exploited coherence protocols.

NIC_0 has three messages in its message queues: the red ones are in LA stage, while the blue one is in VA stage. The NIC_0's output channels are marked to know whether it is possible to allocate on them. The red ones are available, while the others are not. The APNEA policy running on the NIC has to decide to request for a new channel or to wait for an already ON channel to become free. The decision is communicated to R_1 through the action signal.

3.3.1 Upstream

The NIC-to-router upstream policy is equal to the router-to-router upstream one. However, the architectural differences between the router and the NIC requires the definition of new quantities to be exploited in the policy evaluation.

The NIC gets the messages from the message queues, it flitisizes them and allocates the resulting flits in a chosen output VC. Then it arbitrates the output requests competing for the link and send the chosen flit.

$R_{VA_{i,t}}$ is defined as the number of requests in message queue for VNET i at time t. $R_{SA_{i,t}}$ represents the number of requests ready for LA for VNET i at time t. As for the upstream routers these quantities represent the traffic the NIC is experiencing. The messages in the queue are the incoming part of the traffic while the requests in LA are the active part.

Moreover, $W_{i,t,used}$ is defined as the number of channels used for VNET i at time t, while $W_{i,idle}$ represents the number of channels in idle for VNET i at time t. Finally

$$W_{i,t,usable} := W_{i,t,idle} \tag{3.2}$$

represents the number of channels usable in the NIC for VNET i at time t.

The local decision policy is described in Algorithm 8.

Algorithm 8 Local decision in NIC for VNET i at time t+1

```
$R_{i,t}$ := Local decision in NIC for VNET $i$ at time $t$

if $W_{i,usable} > 0$ then
    if $(R_{VA_{i,t}} < R_{SA_{i,t}}) \lor (R_{VA_{i,t}} == 0 \land R_{SA_{i,t}} == 0)$ then
        $R_{i,t+1} \leftarrow -1$
    else
        $R_{i,t+1} \leftarrow 0$
    end if
else
    if $(R_{VA_{i,t}} \geq R_{SA_{i,t}}) \land \neg(R_{VA_{i,t}} == 0 \land R_{SA_{i,t}} == 0)$ then
        $R_{i,t+1} \leftarrow +1$
    else
        $R_{i,t+1} \leftarrow 0$
    end if
end if
```

When there are some usable channels and the requests in the message queue are less than the ones in LA the traffic is decreasing, thus taking the local decision of turn off a not needed channel. Note the difference with the router local decision policy where a turn off choice was made even if the traffic was constant. As the new condition is weaker we need to strengthen it by deciding to turn off a channel even when there are no requests in the NIC, meaning there is no traffic in it. On the other hand, the local decision to turn on is taken if there are not enough usable channels and the number of the incoming request is greater or equal than the number of the active ones, meaning the traffic is going to increase. $\neg(R_{VA_{i,t}} == 0 \wedge R_{SA_{i,t}} == 0)$ boolean expression is added in the turn on condition to prevent policy oscillations in absence of traffic. Again if none of the above conditions is met the local decision to keep the same number of channels is taken, because no more or less of them are needed.

3.4 APNEA Methodology: from the Model to Reality

This section discusses some implementation and feasibility details of the presented policy. APNEA has been integrated in an event-based cycle accurate simulator. The event-based simulators models the execution of a determined component using events that can can be directly scheduled on the component itself. This means that at each clock cycle eventually each component in the simulated architecture processes its own event and can generate an event on a subsequent component to be processed after one or several cycles in the future. For example, considering the router pipeline described in Section 1.3 the BW stage writes the flit from the link to the input buffer on the router and computes the output port for the stored flit. Moreover, it triggers an event on the VA pipeline stage for the next cycle to require the computation of the output virtual channel given the already computed output port for the flit. The presented methodology is executed at each clock edge exploiting the sampled quantities in the previous cycle. This keeps the APNEA policy away from the critical path of the router, since it is logically in parallel to each pipeline stage. It uses a signal forward mechanism from the link to update the requests in BW to be used by the policy. However, this is a standard mechanism that does not impact the timing of the router pipeline

Figure 3.4 clearly states when APNEA policy is executed. The pipeline stages of a 4 flit packet traversing a router are examined. The considered T_{ON} is equal to 2 cycles. When the head flit arrives in the upstream router it is detected by the APNEA policy during the BW stage. In the same stage, since the incoming requests are greater than the active ones it is asserted the signal to power on. In upstream the VC allocation is stalled because for T_{ON} cycle so a delayed pipeline execution is reported. The signal is asserted

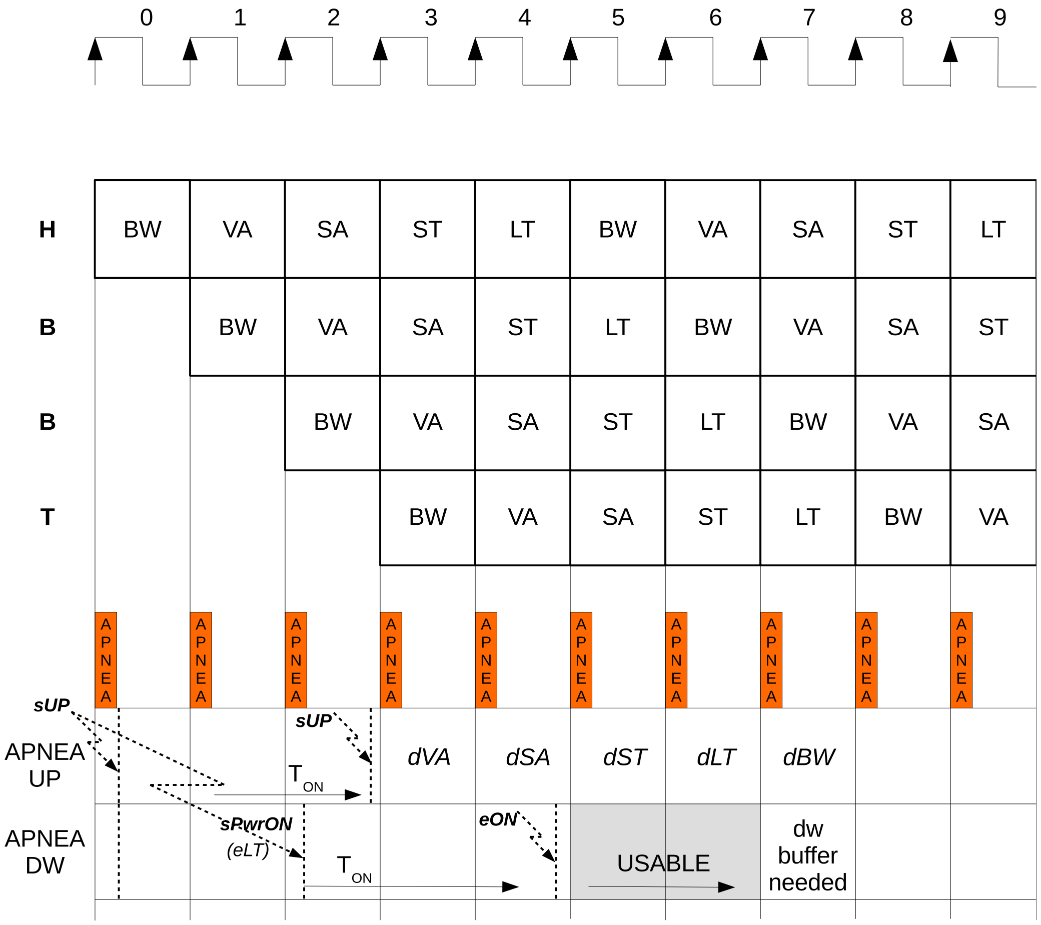


Figure 3.4: A detailed overview of APNEA execution.

on the downstream router too and it is received in two cycles because it needs to traverse the link and be received by the router itself. When the downstream process the received signal it targets a VC to power on, which will be available after T_{ON} cycles at cycle 5. An important fact to note is that the concept to notice is there are 2 cycles between the complete buffer power on and its effective need, thus the shadow cycles optimization is introduced. In upstream it is not up to 2 cycles of the T_{ON} can be hidden preventing the pipeline from stalling in this case or, removing up to 2 stall cycles for higher T_{ON}.

Considering the actual APNEA implementation a single corner case can introduce performance degradation in the presented methodology. Results presented in Chapter 4 highlight a negligible impact on the overall system performance due to this issue. However, a description of such a corner case is provided in the rest of this section. Figure 3.5 depicts the scenario where the serialization issue can affect the policy performance. In particular, when the number of VA requests and SA requests are equals, APNEA prevents an additional channel to turn on, thus incurring in a serialization problem, imposed by the flow balance assumption.

The P_3 flit in R_0 is blocked even if its path is free. This happens because P_2 flit is in R_1 blocked by another flit and it keeps the channel in R_0 active waiting for the credit back. P_3 is blocked due to the flow balance assumption in APNEA, which prevents a channel turn on if the number of incoming requests and active ones are equals. Moreover even P_2 suffer the same kind of problem since in R_1 the number of active requests and incoming ones are equal due to flit P_1 in R_2.

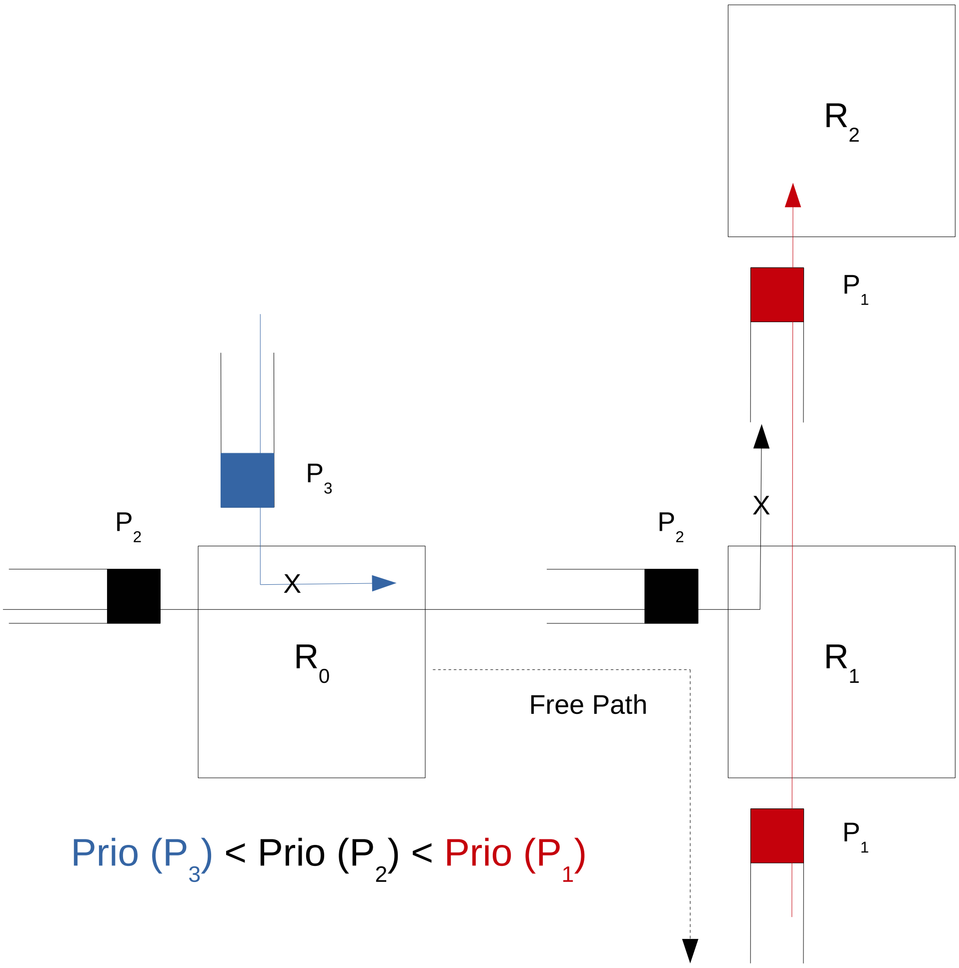


Figure 3.5: The serialization problem.

Chapter 4

Results

"Success consists of going from failure to failure without loss of enthusiasm."

Winston Churchill

This chapter aims to evaluate the APNEA methodology in terms of performance and leakage power. The analysis considers three metrics: the performance overhead, the average buffer usage and the total energy consumption.

Both synthetic traffic and real applications have been evaluated. Furthermore, an exploration of some router design parameters is done to strengthen the flexibility of the methodology.

The rest of the chapter is organized as follows. Section 4.1 describes the simulation environment setup, the target architecture and the used benchmarks. Section 4.2 presents the synthetic traffic results. Section 4.3 details the results exploiting the *mibench* applications [20].

4.1 Simulation Setup

The APNEA methodology has been integrated in the enhanced version [41, 40] of the *gem5* cycle accurate simulator [6].

The DSENT [32] NoC power tool has been used to extract power data of the simulated NoC architecture in place of the Orion 2.0 power model that was implemented in the [12, 36]. Considering the work in [11] the router wakeup energy is assumed to be equal to the router static power consumed during 10 cycles (BET[1]). This work elaborates on the results proposed in [11] with the power numbers provided from DSENT defining the router wakeup energy as:

$$E_{wakeup,router} = P_{leakage,router} * BET * FREQ \tag{4.1}$$

Furthermore, the single buffer wakeup energy has been assumed as:

$$E_{wakeup,buffer} = \frac{E_{wakeup,router}}{N_{inports} * N_{vcs,inport}} \tag{4.2}$$

The quantity found is reasonable since at most it can lead to a little overestimation of the wakeup energy of a single buffer.

From the energy view point four different quantities have been examined for a buffer: the ON time, the OFF time, the OFF to ON time and the ON to OFF time. The ON time is the portion of time the buffer is turned on and it is labeled in the results as ON. The OFF time is the portion of the time the buffer is sensed as power gated and it is labeled as OFF. The OFF to ON time is the portion of time the buffer is sensed in the transient wakeup state and it is labeled as OFF→ ON. Finally the ON to OFF time is the portion of the time the buffer is sensed in the transient turnoff state and it is labeled as ON→ OFF.

With the usage data extracted by the system simulation and the power consumption data obtained by DSENT it is possible to estimate the static energy consumption of the router, both in the baseline and APNEA cases.

Finally Table 4.1 reports the used simulation parameters, while Table 4.2 shows the tested parts of the methodology. The methodology has been tested in R2R, N2R and FULL APNEA to show how the different upstream module types affect the results.

4.1.1 Synthetic Traffic Setup

In the synthetic traffic setup 4 different traffic patters have been tested to give a more complete behavior of APNEA. They are listed in Table 4.1.

Furthermore a 3 VNET NoC is considered because this is the minimum number of virtual networks to support a coherence protocol [3]. One of the VNETs is considered the data VNET.

[1] The Break-Even Time is the minimum number of consecutive cycles that a router needs to remain in sleep before waking up to equal the wakeup energy

Processor Core	1GHz, Out-of-Order Alpha Core
L1I Cache	32kB 2-way Set Associative
L1D Cache	32kB / 64kB 2-way Set Associative
L2 cache	256kB / 512kB per bank, 8-way Set Associative
Coherence Prot.	MESI (3 VNET protocols)
Router	4-stage Wormhole Virtual Channelled
	32bit Link Width
	Buffer Depth 4 / 8 Flits
	2 Virtual Channels (VCs) for each Virtual Network
	3 Virtual Networks (Garnet Network [1])
Topology	2D-mesh 4x4 at 16 Cores / 8x8 at 64 Cores
Technology	45nm at 1.0V
Synthetic Traffic Patterns	Uniform Random, Tornado, Bit Complement and Transpose
Real Traffic	Subset of the *mibench* benchmarks.

Table 4.1: Experimental setup: processor and router micro-architectures and technology parameters.

Name	NameID	Details
Router-to-Router	**R2R**	APNEA methodology applied only to the input buffers whose ports are attached to a router.
NIC-to-Router	**N2R**	APNEA methodology applied only to the input buffers whose ports are attached to a NIC.
Full	**FULL**	APNEA methodology applied to all the input buffers in the router.

Table 4.2: Evaluated APNEA policies with a detailed description of each configuration.

mibench Applications	Launch Parameters
susan	input_small.pgm
qsort	input_small.dat
sha	input_small.asc
search	stringsearch/search_small
dijkstra	input.dat
bitcount	75000
basicmath	basicmath_small

Table 4.3: *mibench* applications used with the detailed input parameters of each application.

During the experiments different data sizes have been tried. First synthetic traffic with data packet size of 1 flit is used to stress the methodology in the worst case, i.e. requiring a buffer per each new packet. The other data packet chosen is 3 flits because it is a more realistic test case considering a cache coherent multi-core, where both long and short packets are injected in the NoC. In fact, this represents in the test configuration a link width of 192 bits.

The cache configuration is a 64KB L1 data cache, a 32KB L1 instruction cache and a 512 KB L2 cache per bank. The NoC topology is a 8x8 2D-mesh with 64 Alpha cores and NoC routers with 2 VCs per VNET, each with buffer depth of 4 flits.

4.1.2 *mibench* Benchmarks Setup

The *mibench* test suite provides single threaded applications. Among the full set of applications present in the suite a reduced set of 7 applications showed in Table 4.3 has been chosen. *mibench* are used to evaluate the methodology behavior under real traffic. They are single threaded applications, thus a number of instances equal to the number of cores in the architecture is used.

The cache configuration is a 32KB L1 caches a 256 KB L2 cache per bank. Cache coherency exploits the MESI protocol, which requires 3 VNETs. The NoC topology is a 4x4 2D-mesh with 16 cores and NoC routers with 2 VCs per VNET. Buffer depth has been explored with 4 flits and 8 flits. The decision of these two different buffer depth is done to stress the methodology in two scenarios: one where the round trip time is not covered and another where it is.

4.2 Synthetic Traffic Results

This section discusses results considering 4 different synthetic traffic types of the APNEA methodology as described in Section 4.1.1.

They are used to stress the performance of the proposed methodology considering corner case scenarios, especially from the upstream role view point, since real applications can hardly impose such high traffic.

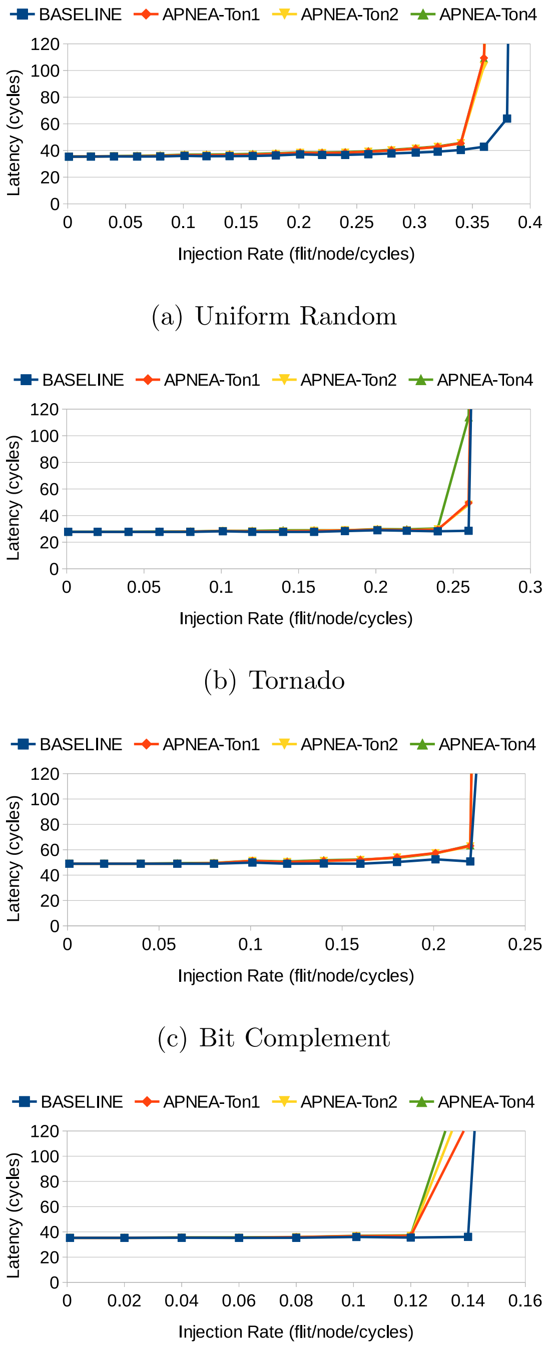
Figure 4.1 shows the results when the synthetic traffic is composed by single flit packets only, for all the 3 considered VNETs.

This scenario mimics a non coherent system. Moreover, this is the worst case scenario because a new buffer is requested for every packet, thus it is an important result to focus on.

APNEA methodology obtains the same average latency of the baseline NoC considering low, medium and high traffic conditions however, Figure 4.1(a) highlights a small performance degradation close to the saturation point for all the considered traffic patterns. This is due to the serialization problem described in Section 3.4. Flow balance assumption in APNEA methodology tries to keep low the number of active buffers. Thus, it prevents the switch on of a new buffer if the number of active ones is equal to the number of VA requests, thus forcing some packet in VA stage to wait for an active buffer to become free.

Figure 4.2 shows the 4 traffic patterns considering single flit packets for 2 VNETs and 3 flit packets for the third one, thus simulating a cache coherent multi-core where short and long packets are integrated in the NoC. In this second scenario, the APNEA methodology average latency is identical to the baseline one and near the saturation point the use of longer packets mitigates the effects of the serialization issue due to flow balance assumption.

Figure 4.2(a) highlights that the APNEA methodology reaches better performance compared to the baseline. This problem is due to the synthetic traffic generator, which keeps only a constant injection rate of packets. However, the packet arrival distribution is not controlled and two long packets can arrive in two consequent cycles or interleaved by some empty cycles. The first case causes an higher average latency since the second packet needs to wait more cycles. This problem is highlighted especially close to the saturation point because of the high traffic in the network.



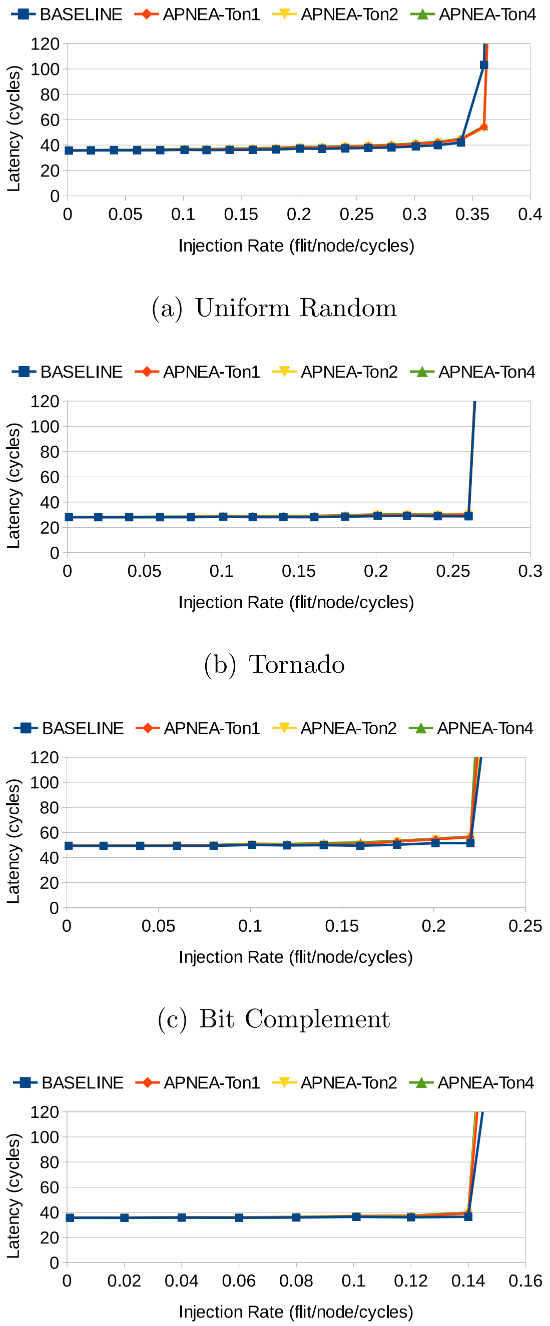
(a) Uniform Random

(b) Tornado

(c) Bit Complement

(d) Transpose 1

Figure 4.1: Average packet latency with data packet size 1. Figure 4.1(a) shows the average latency comparison between the baseline and the APNEA using uniform random traffic. Figure 4.1(b) presents the same results using tornado traffic, while Figure 4.1(c) uses bit complement traffic and Figure 4.1(d) exploits transpose traffic.



(a) Uniform Random

(b) Tornado

(c) Bit Complement

(d) Transpose 1

Figure 4.2: Average packet latency with data packet size 3. Figure 4.2(a) shows the average latency comparison between the baseline and the APNEA using uniform random traffic. Figure 4.2(b) presents the same results using tornado traffic, while Figure 4.2(c) uses bit complement traffic and Figure 4.2(d) exploits transpose traffic.

4.3 Real Traffic Results: *mibench*

This section provides the results of the APNEA methodology compared to the baseline architecture using the *mibench* applications. Results are reported considering APNEA applied to different upstream modules: routers only (R2R), NIC only (N2R) and both NICs and routers (FULL). This helps to understand the critical points from energy and performance view point and it gives the overall view of the methodology applied to the NoC.

4.3.1 APNEA Router-to-Router

This section discusses the APNEA R2R, where only the buffers attached to another router can be power gated. Thus the buffers attached to NIC are always active while the energy saving is still considered for the entire router.

Figure 4.3 shows the results for both performance overhead and energy saving due to leakage power when the methodology is applied to the router upstream module.

Figure 4.3(a) shows that the APNEA R2R methodology is able to keep the performance overhead limited (no performance degradation on average among all the tests).

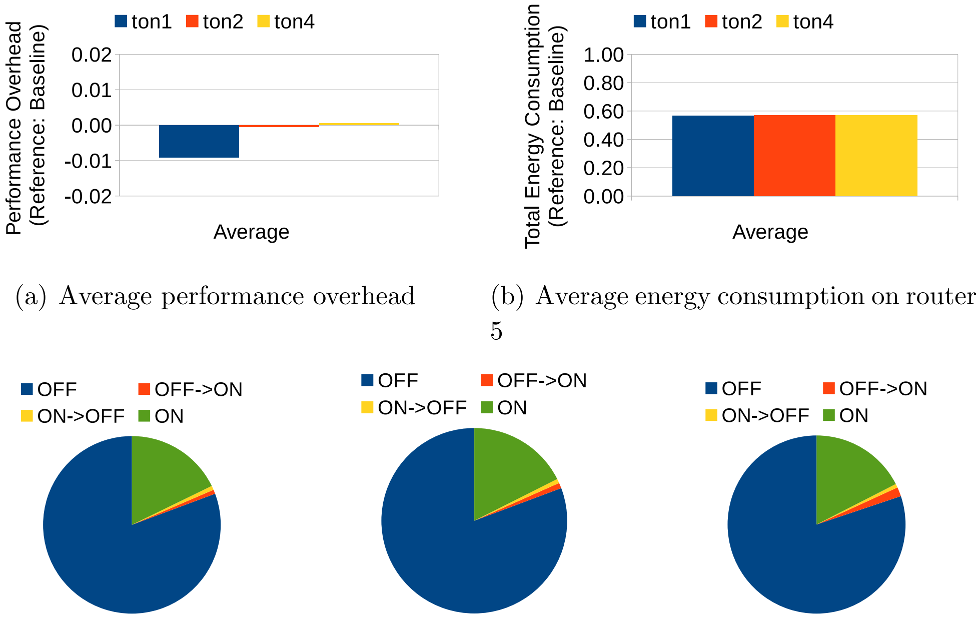
The total energy saving given by the leakage power is the 43% on average on all the benchmarks as pointed out in Figure 4.3(b). The energy saving is not optimal since APNEA R2R only actuate on a part of the input ports.

However, the effectiveness of the methodology is presented for example in Figure 4.3(c), where the APNEA R2R allows a buffer to stay OFF for 80% of the time on average.

It is worth noticing that the wakeup events provide a minor impact on both performance overhead and energy consumption. The degradation is limited to 1% on average even considering a wakeup time of 4 cycles demonstrating the good scalability of the methodology.

The time the buffer is in OFF$\rightarrow$ ON state increases with the T_{ON}, but it remains limited, because the APNEA methodology prevents oscillations.

Finally, the detailed results of performance overhead, buffer utilization and energy saving for each application in the *mibench* are reported in Figure 4.4. It is worth noticing that the variance with the average results it limited among all the benchmarks.



(a) Average performance overhead

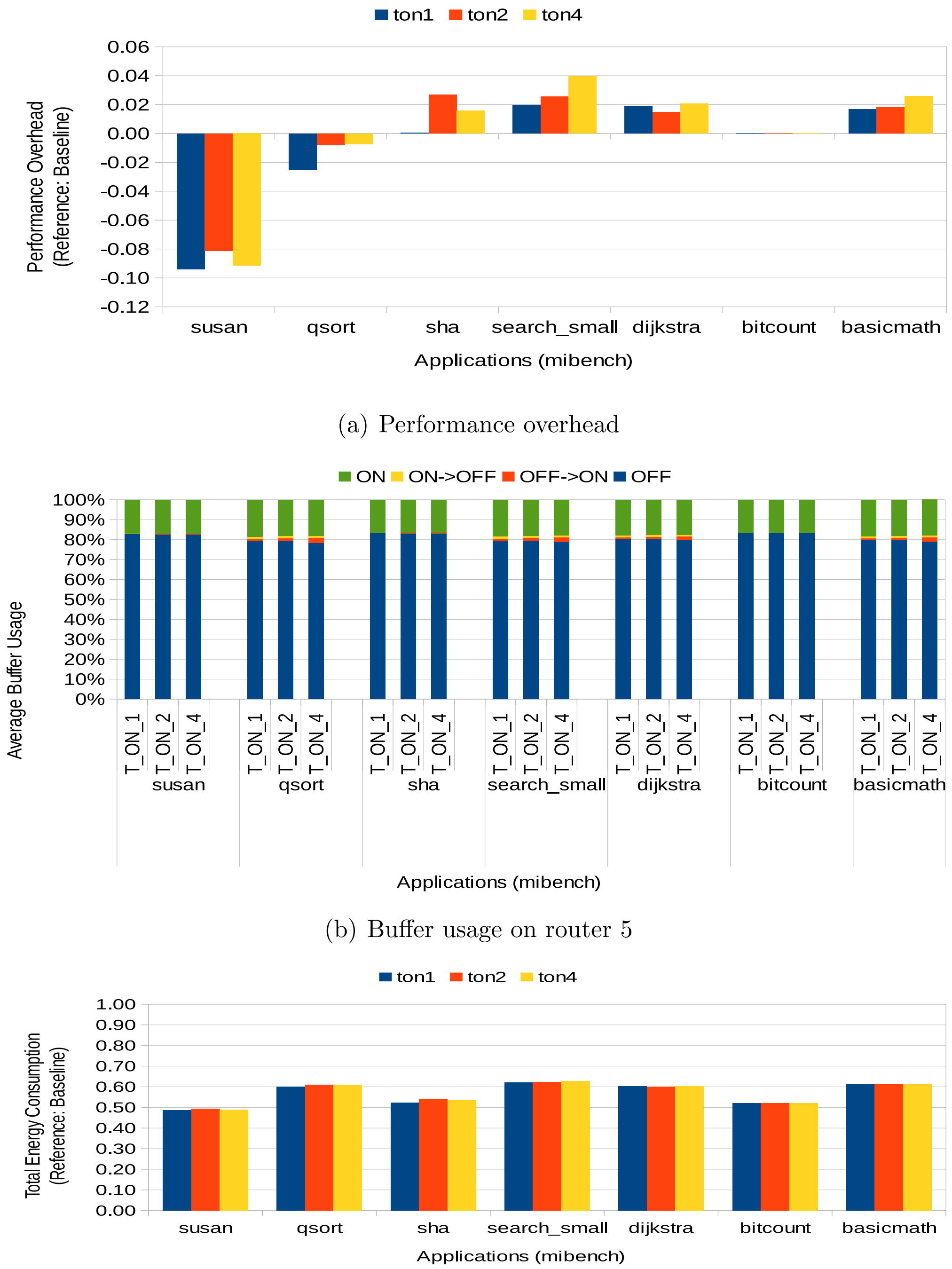
(b) Average energy consumption on router 5

(c) Average buffer usage on router 5 with T_{ON}=1

(d) Average buffer usage on router 5 with T_{ON}=2

(e) Average buffer usage on router 5 with T_{ON}=4

Figure 4.3: Average APNEA R2R results with buffer depth 4. Figure 4.3(a) shows the average performance overhead compared to the baseline. Figure 4.3(b) presents the average energy consumption respect to the baseline router. Figure 4.3(c), Figure 4.3(d) and Figure 4.3(e) exhibit the average buffer usage.

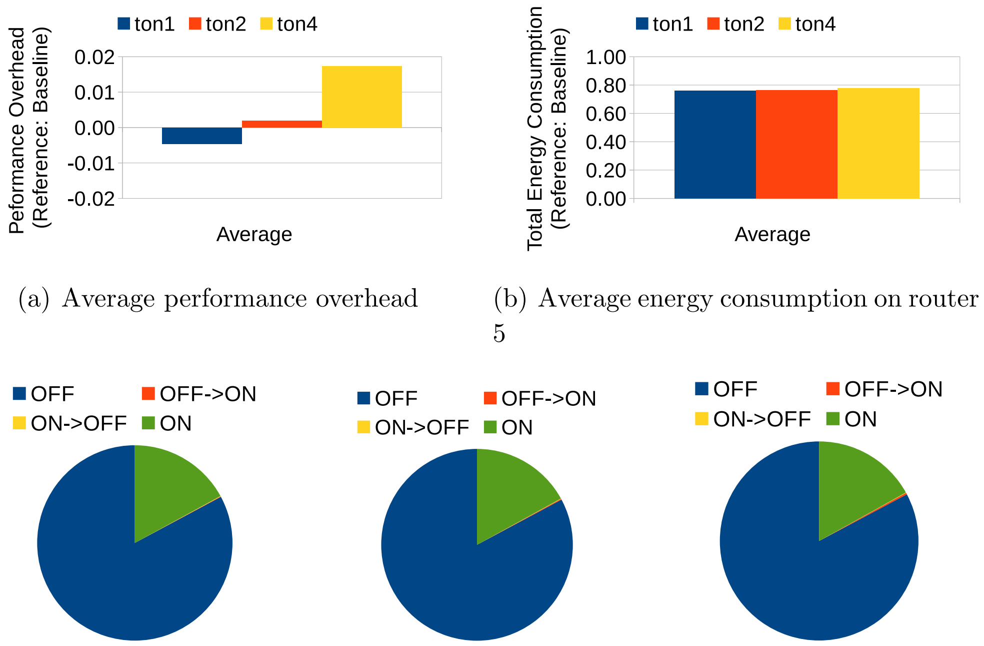


(a) Performance overhead

(b) Buffer usage on router 5

(c) Total energy consumption on router 5

Figure 4.4: Detailed APNEA R2R results with buffer depth 4. Figure 4.4(a) shows the performance overhead for each application. Figure 4.4(b) presents the detailed buffer usage, while Figure 4.4(c) exhibits the total router energy consumption for each application.



(a) Average performance overhead

(b) Average energy consumption on router 5

(c) Average buffer usage on router 5 with T_{ON}=1

(d) Average buffer usage on router 5 with T_{ON}=2

(e) Average buffer usage on router 5 with T_{ON}=4

Figure 4.5: Average APNEA N2R results with buffer depth 4. Figure 4.5(a) shows the average performance overhead compared to the baseline. Figure 4.5(b) presents the average energy consumption respect to the baseline router. Figure 4.5(c), Figure 4.5(d) and Figure 4.5(e) exhibit the average buffer usage.

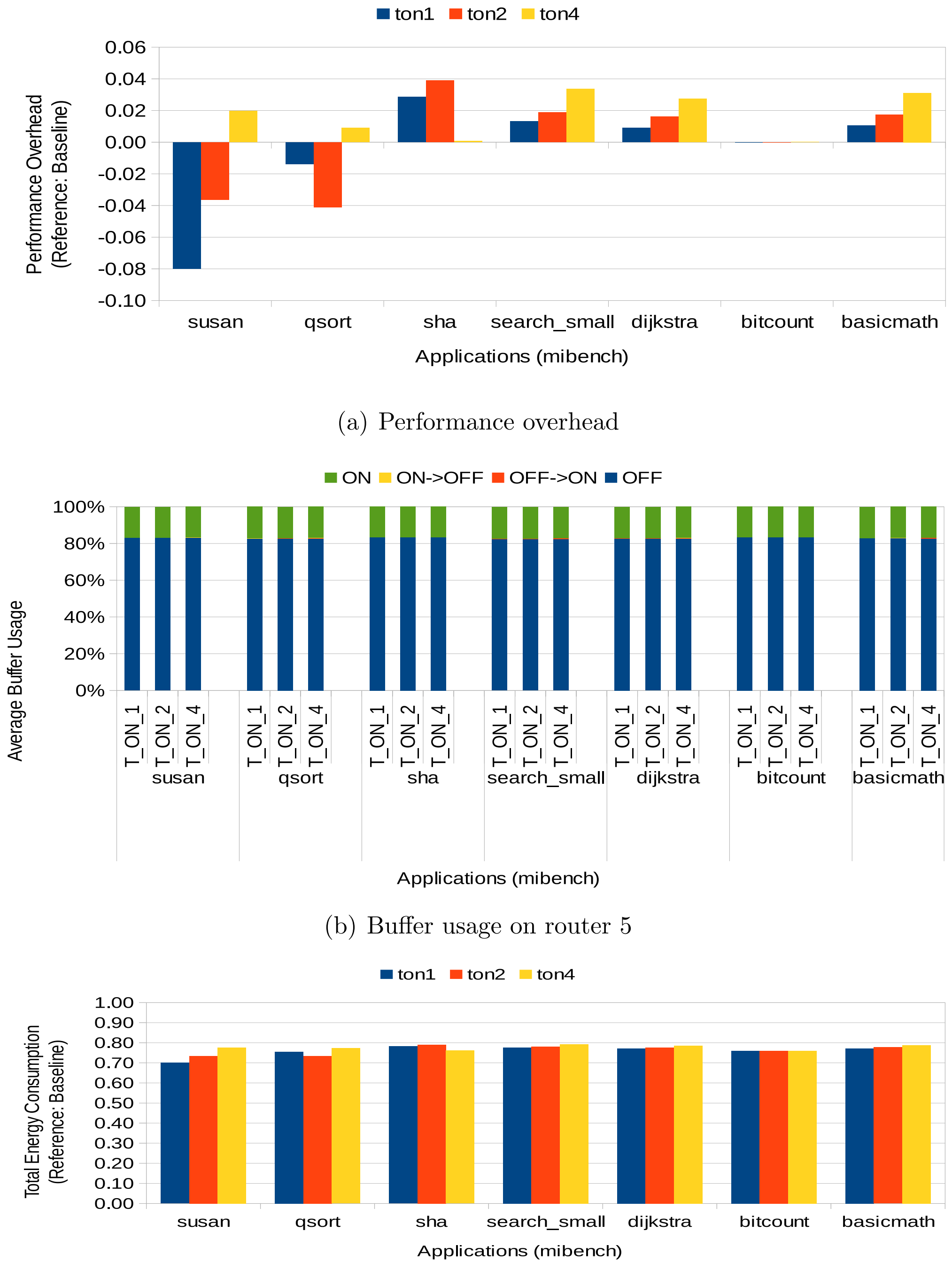
4.3.2 APNEA NIC-to-Router

The APNEA N2R can only power gate the input port's buffers connected to a NIC. Figure 4.6 shows the simulation result reposting the performance overhead, the buffer utilization and the energy saving.

Figure 4.5(a) shows that the APNEA N2R is able to keep the performance overhead limited, with a negligible performance degradation on average among all the tests.

Figure 4.5(b) points out that the energy saving given by the reduced leakage power is on average the 23% of the baseline total energy consumption. It is worth noticing that the benefit is limited due to the limited number of input ports connected to the NIC, that are on average 2 out of 6 in a router.

The average buffers sleep time is 80% as it can be reported for example in Figure 4.5(c), where a $T_{ON} = 1$ is considered. The result considers only

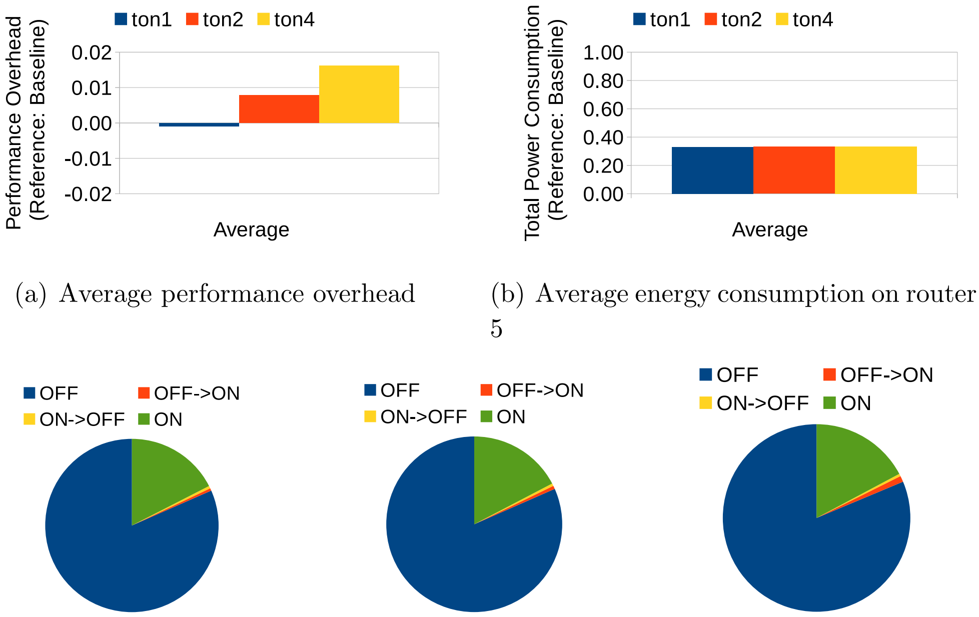


(a) Performance overhead

(b) Buffer usage on router 5

(c) Total energy consumption on router 5

Figure 4.6: Detailed APNEA N2R results with buffer depth 4. Figure 4.6(a) shows the performance overhead for each application. Figure 4.6(b) presents the detailed buffer usage, while Figure 4.6(c) exhibits the total router energy consumption for each application.



(a) Average performance overhead

(b) Average energy consumption on router 5

(c) Average buffer usage on router 5 with T_{ON}=1

(d) Average buffer usage on router 5 with T_{ON}=2

(e) Average buffer usage on router 5 with T_{ON}=4

Figure 4.7: Average APNEA FULL results with buffer depth 4. The average performance overhead is reported in Figure 4.7(a). The average router energy consumption is exhibited in Figure 4.7(b). Figure 4.7(c), Figure 4.7(d) and Figure 4.7(e) shows the average buffer usage for different T_{ON}.

the buffers managed by the APNEA N2R.

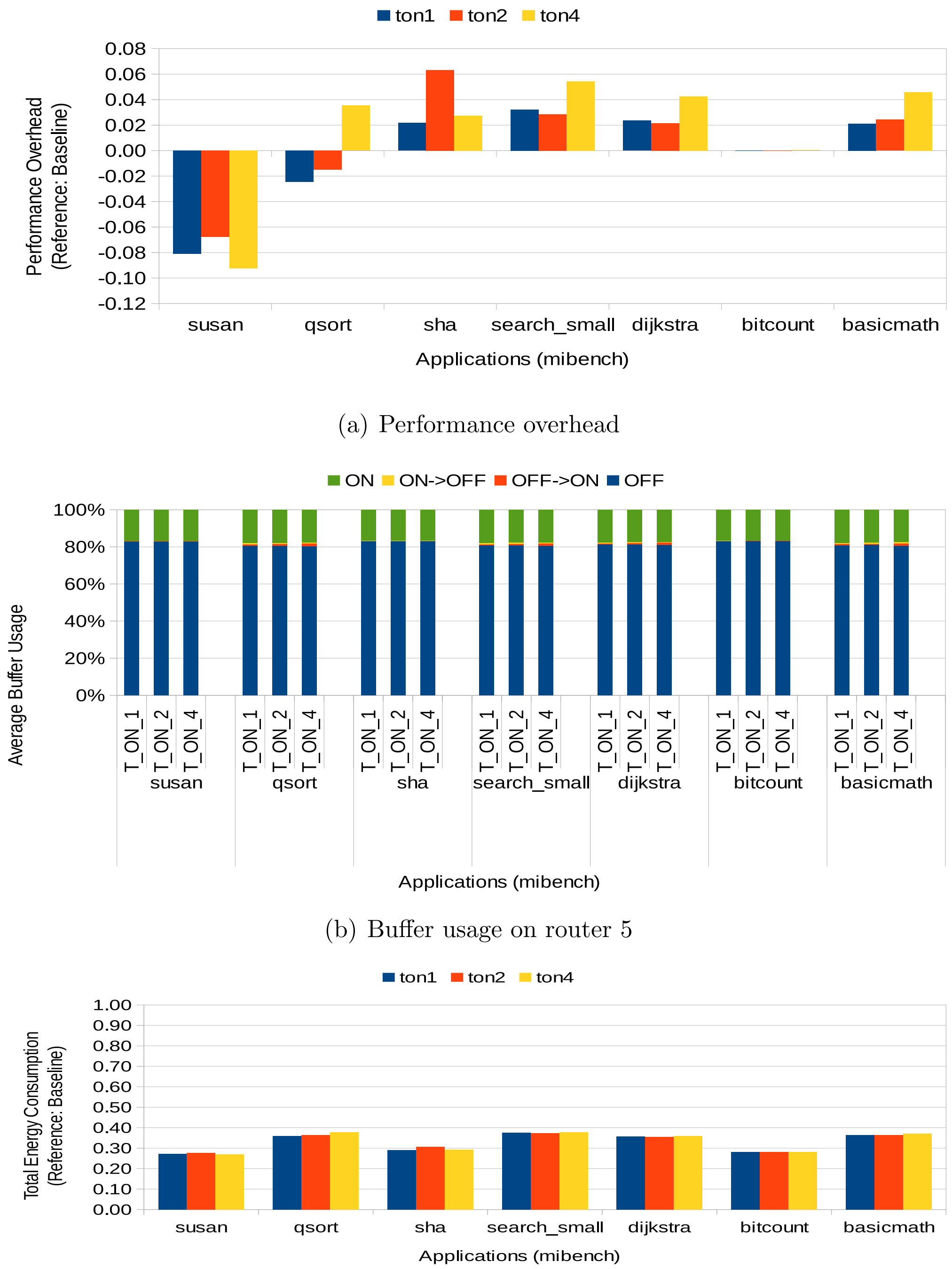
Lastly, the detailed results are reported in Figure 4.6 showing the performance overhead, the buffer usage and the total energy consumption in each considered application of *mibench*.

4.3.3 Full APNEA

This section discusses the results considering the FULL APNEA methodology, namely both APNEA R2R and N2R are used at the same time. Thus, in a single router it is possible to switch off all of its buffers.

Figure 4.7 presents the results reposting the performance overhead, the buffer utilization and the energy saving considering the *mibench* application.

Figure 4.7(a) shows that the performance overhead is not additive with respect to the R2R and the N2R, while it is still confined within 1% on average for the APNEA FULL. On the other hand, results in Figure 4.7(b)

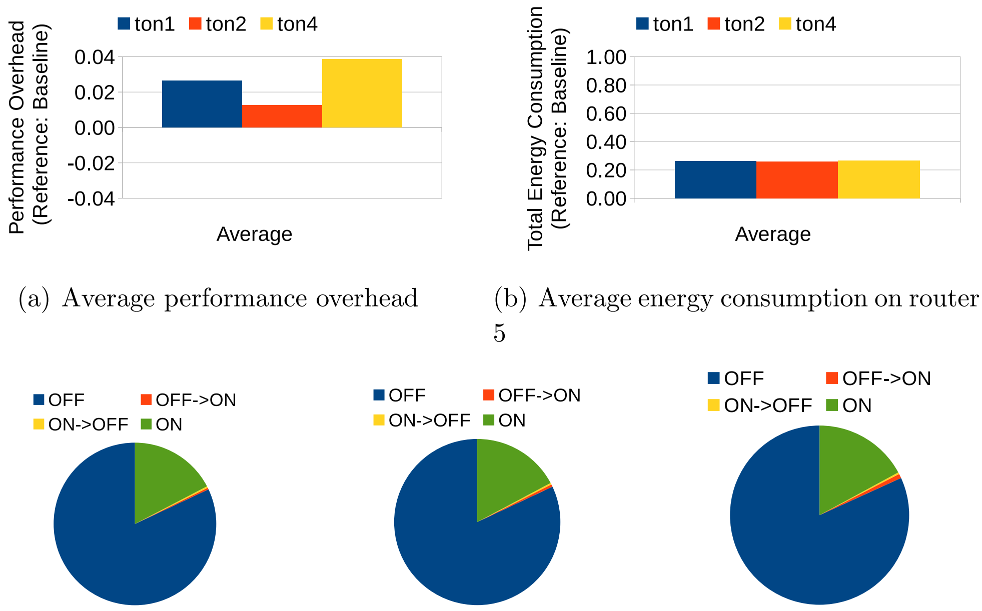


(a) Performance overhead

(b) Buffer usage on router 5

(c) Total energy consumption on router 5

Figure 4.8: Detailed FULL APNEA results with buffer depth 4. Figure 4.8(a) shows the performance overhead for each application. Figure 4.8(b) presents the detailed buffer usage, while Figure 4.8(c) exhibits the total router energy consumption for each application.



(a) Average performance overhead

(b) Average energy consumption on router 5

(c) Average buffer usage on router 5 with T_{ON}=1

(d) Average buffer usage on router 5 with T_{ON}=2

(e) Average buffer usage on router 5 with T_{ON}=4

Figure 4.9: Average APNEA FULL results with buffer depth 8. The average performance overhead is reported in Figure 4.9(a). The average router energy consumption is exhibited in Figure 4.9(b). Figure 4.9(c), Figure 4.9(d) and Figure 4.9(e) shows the average buffer usage for different T_{ON}.

point out the additivity of the energy saving with respect to APNEA R2R and APNEA N2R.

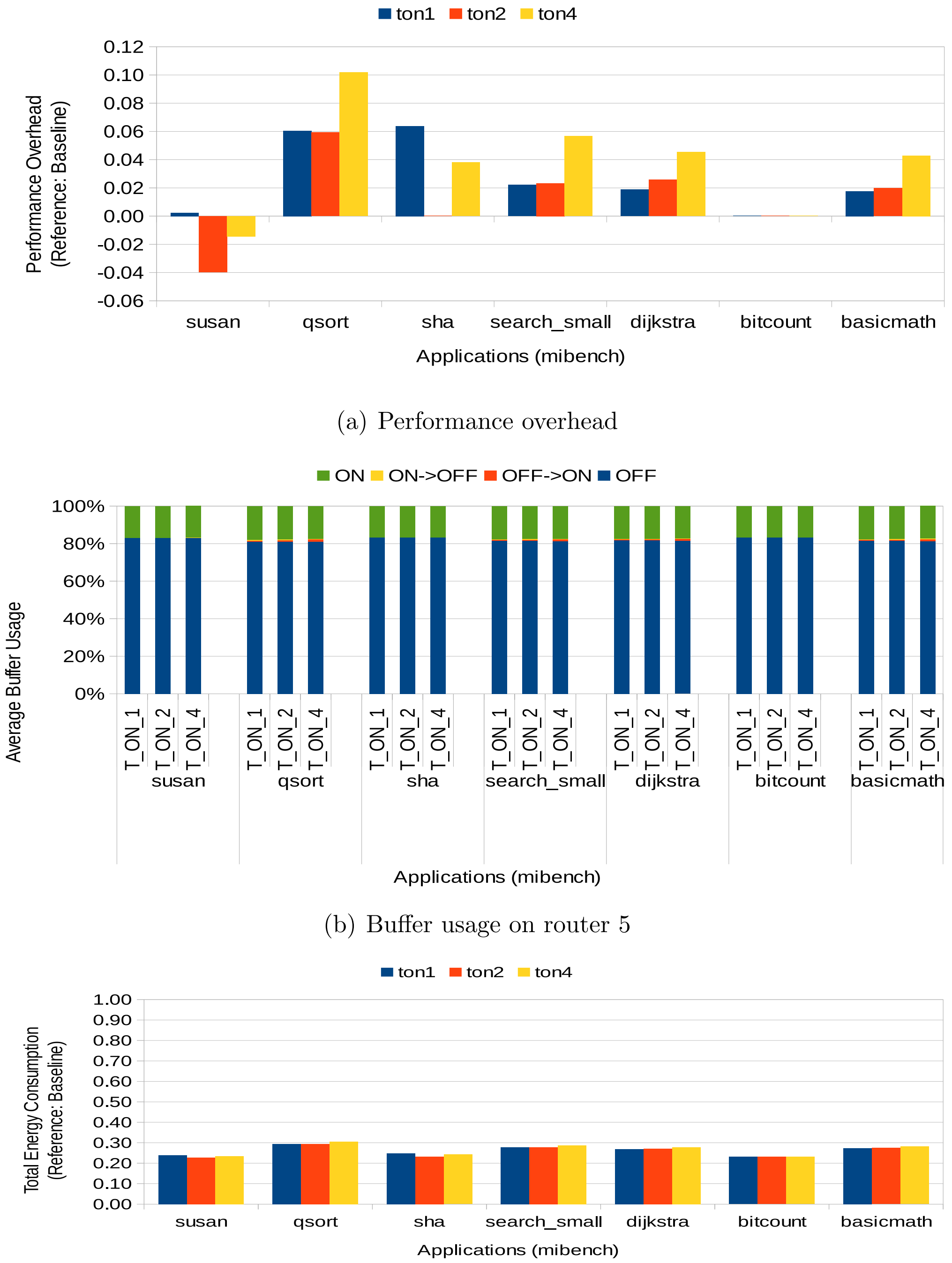
Moreover, Figure 4.7(d) for example reports a buffer OFF time of 80% considering all the buffers in the router, where $T_{ON} = 2$.

Again, the complete results of performance overhead, buffer utilization and total energy consumption for each considered application in the *mibench* is detailed in Figure 4.8.

Some exploration on the buffer depth has been done for the FULL APNEA methodology and in Figure 4.9 there are presented the average results when the buffer depth is set to 8, while in Figure 4.10 presents the detailed results of each application in *mibench*.

The performance overhead profile is similar to the ones in Figure 4.7, thus highlighting that a deeper buffer negligibly impacts on the methodology.

Conversely Figure 4.9(b) shows how a deeper buffer provides a greater power saving, which reaches the 74%, i.e. a deeper buffers has a greater static



(a) Performance overhead

(b) Buffer usage on router 5

(c) Total energy consumption on router 5

Figure 4.10: Detailed FULL APNEA results with buffer depth 8. Figure 4.10(a) shows the performance overhead for each application. Figure 4.10(b) presents the detailed buffer usage, while Figure 4.10(c) exhibits the total router energy consumption for each application.

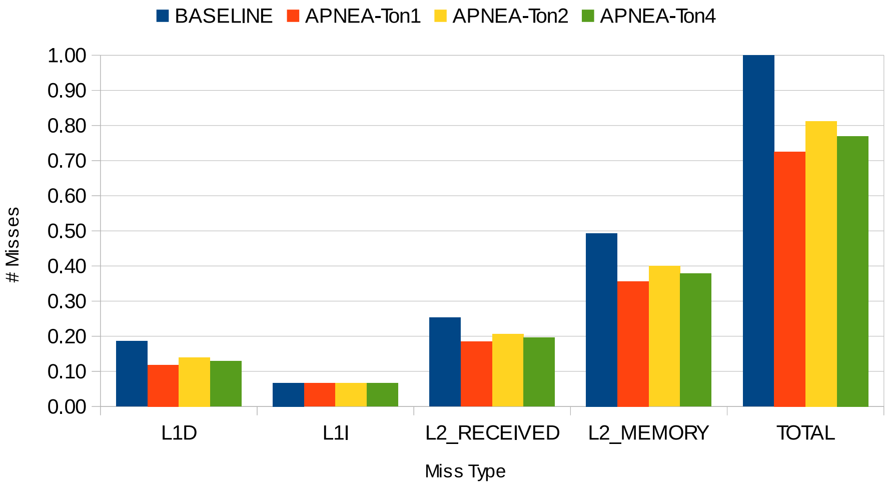


Figure 4.11: Number of misses in APNEA FULL in *susan* [20]. There are reported the absolute number of misses, divided for each tested architecture, to highlight the differences between APNEA and the baseline.

power and consequently the APNEA saving on the total energy increases.

4.3.4 A Particular Case Explanation

This section explains the results obtained using the *susan* application with APNEA R2R, N2R and FULL. As an example, it will discussed the APNEA FULL with buffer depth 4. Figure 4.8(a) shows that *susan* improves the performance by 10% compared with the baseline NoC.

This behavior is given by the packet latency overhead introduced by the APNEA methodology. When a packet is delayed in the network, it causes the coherence protocol to possibly actuate in a different way and modify the generated requests and responses. Thus, a different traffic is generated in the network.

The requests and responses in the cache coherence protocol are generated due to misses in the various memory components. Additional experiments on the emphsusan application highlight the experiences misses in the benchmark. Figure 4.11 reports that the number of misses from L1 to L2 and from L2 to memory are different for the baseline and the APNEA tests. The total number of misses in APNEA simulation is on average the 75% of the baseline ones, meaning the number of injected packets in the NoC is with APNEA.

Chapter 5

Conclusions and Future Works

Considering the NoC based multi-cores architectures, buffers are the resources with the major leakage impact and they determine the performance of the whole interconnect. Furthermore, the dimensioning of these resources is done respect to the worst case, thus they remain idle or underutilized for the majority of the time.

This trend is emphasized by the burst behavior of the applications. Typically an application interconnect usage in not constant but there are periods where the traffic is higher and others where it is really low.

Different methodologies have been proposed to mitigate the leakage power in NoC router, focusing on the complete router power gating or on the buffer power gating. However the power gating at router level is suitable against low traffic or medium traffic adding some assumptions on the routing algorithm or the topology of the network.

In this context a power gating methodology targeting the router's buffers in NoC based multi-core architectures to reduce leakage power while keeping almost the same performance of the baseline NoC is proposed.

The methodology is totally independent from the topology or the routing algorithm employed and it focuses on the ON and OFF oscillation minimization on the single buffers through the reshaping of the traffic. Furthermore, it does not impact on the critical path of the NoC router in any way.

Experimental results conducted with synthetic traffic and real applications using architectures with up to 64 cores highlighted a performance overhead of the methodology limited to 2% on average, with an average energy saving up to 74%.

5.1 Conceptual Comparison with State of the Art Methodologies

This section describes two state of the art methodologies, namely Power Punch [11] and Ultra Fine-Grained [28]. They represent the best candidates from the performance penalties and the power saving viewpoints among all the works discussed in this manuscript. However, they are still limited considering certain desirable properties, i.e. scalability, flexibility and maximum power saving. This thesis developed a new methodology to overcome the limitations found in the state of the art, while a direct comparison with other proposals is not provided. To this extent, the rest of this section discusses the such comparison using a limited but meaningful set of experimental results.

Power Punch tries to save leakage power by power gating at a router granularity. The first problem of this approach is the low traffic assumption, which is not a strong requirement because the principal target of this kind of architectures is high performance computing. Section 1.1 has been provided the average idle time of a router using an oracle approach, namely in each idle cycle the router is considered as powered off. Such percentage is no more than 40% considering medium traffic benchmarks, thus showing the limited opportunity to save static power. Moreover, Power Punch assumes a deterministic routing algorithm and it is limited to mesh topologies. Finally, the wakeup signal is exchanged through a complex network and the methodology is not scalable using higher T_{ON} values, i.e. the time required to wake up a logic block.

[28] focuses on power gating at buffer granularity, overcoming several of the issues exposed in Power Punch. It relies on the look ahead and deterministic routing. It exploits a multi-hop wake up signaling network to wake up the downstream routers. Moreover, [28] can not switch off the buffers directly connected to a NIC, otherwise the presented methodology imposes a great performance penalty. Furthermore, the number of additional lines introduced to support this forwarding is huge because the wakeup must arrive from all the 2-hop distance routers and it is directed to a single VC in the router itself. Last, there is no mechanism that minimizes the number of wake up events of a single buffer, thus imposing a non-negligible wakeup energy. Figure 5.1 and Figure 5.2 present the distribution of idleness in the channels of same input port for both a baseline router and the APNEA router. It is worth noticing that APNEA compacts the traffic in the lowest id chan-

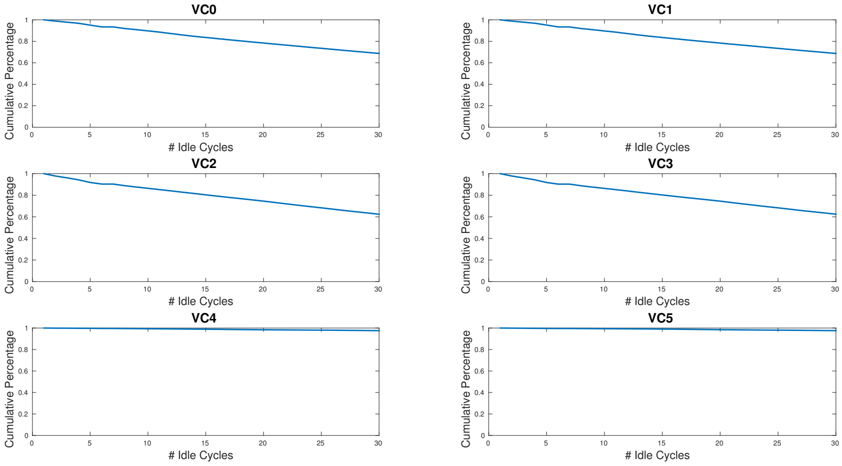


Figure 5.1: Idleness distribution in a single input port with a baseline router using *qsort* [20]. The figure shows the cumulative idleness distribution of all the VCs in a specific port. Every line corresponds to a VNET, namely VC0 and VC1 are in the VNET0.

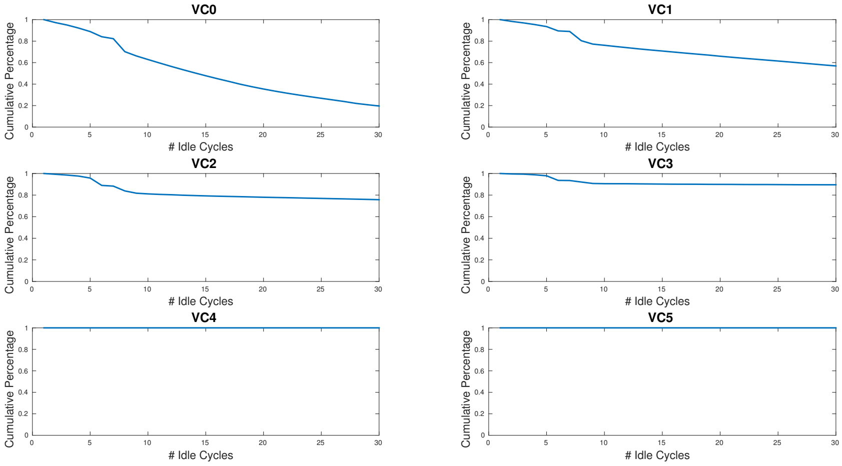


Figure 5.2: Idleness distribution in a single input port with an APNEA router using *qsort* [20]. The figure shows the cumulative idleness distribution of all the VCs in a specific port. Every line corresponds to a VNET, namely VC0 and VC1 are in the VNET0.

nels through the late binding mechanism reducing the switching activity of the higher id ones, while in the baseline router all the channels in the same VNET achieve the same idleness distribution increasing the total number of wakeups. It is worth noticing that [28] does not change the virtual channel allocation policy thus providing the very same packet allocation distribution of a baseline NoC, as depicted in Figure 5.1.

5.2 Future Works

The methodology actuates only on the buffers to limit the leakage power. However, in literature many techniques have been proposed to actuate in a combined way on the buffers, the crossbar and the control logic within the router.

Future works will focus on the extension of APNEA to support the power gating of other router components in order to maximize the static power saving.

Furthermore, the manuscript only provides a methodological comparison with both router granularity and buffer granularity state of the art methodologies. Another extension of the thesis is to provide a strong comparison against the two methodologies discussed in Section 5.1.

Bibliography

[1] N. Agarwal, T. Krishna, Li-Shiuan Peh, and N.K. Jha. Garnet: A detailed on-chip network model inside a full-system simulator. In *Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on*, pages 33–42, April 2009.

[2] M. Annavaram. A case for guarded power gating for multi-core processors. In *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, pages 291–300, Feb 2011.

[3] M. Badr and N.E. Jerger. Synfull: Synthetic traffic models capturing cache coherent behaviour. In *Computer Architecture (ISCA), 2014 ACM/IEEE 41st International Symposium on*, pages 109–120, June 2014.

[4] J. Balfour and W.J. Dally. Design tradeoffs for tiled cmp on-chip networks. In *Proceedings of the 20th annual international conference on Supercomputing*, pages 187–198. ACM, 2006.

[5] A. Banerjee, R. Mullins, and S. Moore. A power and energy exploration of network-on-chip architectures. In *Networks-on-Chip, 2007. NOCS 2007. First International Symposium on*, pages 163–172, May 2007.

[6] N. Binkert, B. Beckmann, G. Black, S.K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D.R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M.D. Hill, and D.A. Wood. The gem5 simulator. *SIGARCH Comput. Archit. News*, 39(2):1–7, August 2011.

[7] H. Bokhari, H. Javaid, M. Shafique, J. Henkel, and S. Parameswaran. darknoc: Designing energy-efficient network-on-chip with multi-vt cells for dark silicon. In *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, pages 1–6, June 2014.

[8] M.R. Casu, M.K. Yadav, and M. Zamboni. Power-gating technique for network-on-chip buffers. volume 49, pages 1438–1440, Nov 2013.

[9] L. Chen and T.M. Pinkston. Nord: Node-router decoupling for effective power-gating of on-chip routers. In *Microarchitecture (MICRO), 2012 45th Annual IEEE/ACM International Symposium on*, pages 270–281, Dec 2012.

[10] L. Chen, L. Zhao, R. Wang, and T.M. Pinkston. Mp3: Minimizing performance penalty for power-gating of clos network-on-chip. In *High Performance Computer Architecture (HPCA), 2014 IEEE 20th International Symposium on*, pages 296–307, Feb 2014.

[11] L. Chen, D. Zhu, M. Pedram, and T.M. Pinkston. Power punch: Towards non-blocking power-gating of noc routers. In *High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on*, pages 378–389, Feb 2015.

[12] S. Corbetta, D. Zoni, and W. Fornaciari. A temperature and reliability oriented simulation framework for multi-core architectures. In *VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on*, pages 51–56, Aug 2012.

[13] W.J. Dally. Virtual-channel flow control. *Parallel and Distributed Systems, IEEE Transactions on*, 3(2):194–205, Mar 1992.

[14] W.J. Dally and B.P. Towles. *Principles and practices of interconnection networks*. Elsevier, 2004.

[15] R. Das, S. Eachempati, A.K. Mishra, V. Narayanan, and C.R. Das. Design and evaluation of a hierarchical on-chip interconnect for next-generation cmps. In *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on*, pages 175–186, Feb 2009.

[16] R. Das, S. Narayanasamy, S.K. Satpathy, and R.G. Dreslinski. Catnap: Energy proportional multiple network-on-chip. In *Proceedings of the 40th Annual International Symposium on Computer Architecture*, pages 320–331, 2013.

[17] J. Duato. A new theory of deadlock-free adaptive routing in wormhole networks. *Parallel and Distributed Systems, IEEE Transactions on*, 4(12):1320–1331, Dec 1993.

[18] J. Duato. A necessary and sufficient condition for deadlock-free adaptive routing in wormhole networks. *Parallel and Distributed Systems, IEEE Transactions on*, 6(10):1055–1067, Oct 1995.

[19] J. Duato and T.M. Pinkston. A general theory for deadlock-free adaptive routing using a mixed set of resources. *Parallel and Distributed Systems, IEEE Transactions on*, 12(12):1219–1235, Dec 2001.

[20] M.R. Guthaus, J.S. Ringenberg, D. Ernst, T.M. Austin, T. Mudge, and R.B. Brown. Mibench: A free, commercially representative embedded benchmark suite. In *Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop on*, pages 3–14, Dec 2001.

[21] S.M. Hassan and S. Yalamanchili. Centralized buffer router: A low latency, low power router for high radix nocs. In *Networks on Chip (NoCS), 2013 Seventh IEEE/ACM International Symposium on*, pages 1–8, April 2013.

[22] Y. Hoskote, S. Vangal, A. Singh, N. Borkar, and S. Borkar. A 5-ghz mesh interconnect for a teraflops processor. *Micro, IEEE*, 27(5):51–61, Sept 2007.

[23] Z. Hu, A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose. Microarchitectural techniques for power gating of execution units. In *Low Power Electronics and Design, 2004. ISLPED '04. Proceedings of the 2004 International Symposium on*, pages 32–37, Aug 2004.

[24] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi. *Low Power Methodology Manual: For System-on-Chip Design.* Springer Publishing Company, Incorporated, 2007.

[25] P. Kermani and L. Kleinrock. Virtual cut-through: A new computer communication switching technique. *Computer Networks (1976)*, 3(4):267–286, 1979.

[26] A. Leroy, J. Picalausa, and D. Milojevic. Quantitative comparison of switching strategies for networks on chip. In *Programmable Logic, 2007. SPL '07. 2007 3rd Southern Conference on*, pages 57–62, Feb 2007.

[27] A. Lungu, P. Bose, A. Buyuktosunoglu, and D.J. Sorin. Dynamic power gating with quality guarantees. In *Proceedings of the 2009 ACM/IEEE International Symposium on Low Power Electronics and Design*, pages 377–382, 2009.

[28] H. Matsutani, M. Koibuchi, D. Ikebuchi, K. Usami, H. Nakamura, and H. Amano. Ultra fine-grained run-time power gating of on-chip routers for cmps. In *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*, pages 61–68, May 2010.

[29] L.M. Ni and P.K. McKinley. A survey of wormhole routing techniques in direct networks. *Computer*, 26(2):62–76, Feb 1993.

[30] R. Parikh, R. Das, and V. Bertacco. Power-aware nocs through routing and topology reconfiguration. In *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, pages 1–6, June 2014.

[31] A. Samih, R. Wang, A. Krishna, C. Maciocco, C. Tai, and Y. Solihin. Energy-efficient interconnect via router parking. In *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, pages 508–519, Feb 2013.

[32] C. Sun, C.H.O. Chen, G. Kurian, L. Wei, J. Miller, A. Agarwal, L.S. Peh, and V. Stojanovic. Dsent - a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling. In *Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium on*, pages 201–210, May 2012.

[33] A.N. Udipi, N. Muralimanohar, and R. Balasubramonian. Towards scalable, energy-efficient, bus-based on-chip networks. In *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, pages 1–12, Jan 2010.

[34] S.C. Woo, M. Ohara, E. Torrie, J.P. Singh, and A. Gupta. The splash-2 programs: characterization and methodological considerations. In *Computer Architecture, 1995. Proceedings., 22nd Annual International Symposium on*, pages 24–36, June 1995.

[35] D. Zoni, L. Borghese, G. Massari, S. Libutti, and W. Fornaciari. Test: Assessing noc policies facing aging and leakage power. In *EUROMICRO DSD/SEAA, Funchal, Madeira, PORTUGAL, Aug 26-28*, pages 1–8, 2015.

[36] D. Zoni, S. Corbetta, and W. Fornaciari. Hands: Heterogeneous architectures and networks-on-chip design and simulation. In *Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design*, ISLPED '12, pages 261–266, New York, NY, USA, 2012. ACM.

[37] D. Zoni and W. Fornaciari. A sensor-less nbti mitigation methodology for noc architectures. In *SOC Conference (SOCC), 2012 IEEE International*, pages 340–345, 2012.

[38] D. Zoni and W. Fornaciari. Nbti-aware design of noc buffers. In *Proceedings of the 2013 Interconnection Network Architecture: On-Chip, Multi-Chip*, IMA-OCMC '13, pages 25–28, New York, NY, USA, 2013. ACM.

[39] D. Zoni and W. Fornaciari. Sensor-wise methodology to face nbti stress of noc buffers. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, pages 1038–1043, March 2013.

[40] D. Zoni and W. Fornaciari. Modeling dvfs and power gating actuators for cycle accurate noc-based simulators. *Journal of Emerging Technologies in Computing Systems*, pages 1–15, 2015.

[41] D. Zoni, F. Terraneo, and W. Fornaciari. A dvfs cycle accurate simulation framework with asynchronous noc design for power-performance optimizations. *Journal of Signal Processing Systems*, pages 1–15, 2015.