Energy Efficient Transceiver in Wireless Network on Chip Architectures

Vincenzo Catania*, Andrea Mineo*, Salvatore Monteleone*, Maurizio Palesi† and Davide Patti*
* University of Catania, Italy. *first.last@dieei.unict.it*
† Kore University, Enna, Italy. *maurizio.palesi@unikore.it*

***Abstract*—The emergent wireless Network-on-Chip (WiNoC) design paradigm has been proposed as a viable solution for addressing the scalability issues affecting the on-chip communication system in future manycores architectures. Within this scenario, the energy contribution of the buffers (both of the routers and radio-hubs) and the transceivers of the radio-hubs, account for a significant fraction of the total communication energy budget. In this paper, we propose a novel energy management scheme aimed at improving the energy efficiency of a WiNoC architecture based on the selective disabling of the power hungry modules that are predicted being not used during the forthcoming clock cycles. The proposed scheme, applied on different WiNoC topologies with different configurations and under different traffic scenarios, has shown interesting energy saving without any impact on the performance metrics and with a negligible impact on silicon area.**

***Keywords*—On-chip communication, Wireless NoC, Energy saving, analysis.**

I. Introduction

The role played by the on-chip communication system in current and future manycore architectures is central. In this context, the Network-on-Chip (NoC) design paradigm is considered as the most viable solution for addressing the communication issues which characterize such manycore architectures. Unfortunately, due to their multi-hop nature, as the network size increases, conventional NoCs which use electric point-to-point links, start to suffer from scalability problems, both in terms of communication latency and energy. To face with them, several emerging interconnect paradigms such as those relying on Optical, 3D and RF solutions have been proposed [1]. In particular, a specific class of RF interconnect introduces a wireless backbone upon the traditional wire-based NoC substrate [2].

The use of the radio medium for on-chip communication is enabled by means of antennas and transceivers which form the core of a, so called, *radio-hub*. A radio-hub augments the communication capabilities of a conventional NoC switch/router by allowing it to wirelessly communicate with other radio-hubs in a single hop. The reduction of the average communication hop count has a positive impact on both performance and power metrics but there is a price to pay in terms of silicon area due for transceivers and antennas. Such limitation is overcome by limiting the number of radio-hubs into the chip and by optimizing their topological mapping [3].

Another aspect regards the attenuation introduced by the wireless channel: since electromagnetic waves are propagated in lossy silicon, the power due to the wireless signaling represents an important contribution of the entire communication energy budget. In fact, in [4] it has been shown that the transmitter is responsible for about 65% of the overall transceiver power consumption, while in [5] such contribution is more than 74%. Thus, wireless communication results more energy efficient than wired communication when the communicating nodes are far away each other.

The most power hungry elements in a WiNoC are represented by the input buffers of routers and radio-hubs, and the transceiver into the radio-hubs. In fact, they represent the critical actors to be optimized for the sake of energy efficiency. In [6] it has been reported that the buffers to/from tiles and the wireless receiving module account for a significant fraction of the total energy budget of the radio-hub (75% and 10%, respectively). It should be pointed out that, the receiving module is continuously listening to the radio medium. In fact, it has to detect whether there is an ongoing wireless communication, and, if so, whether its radio-hub is a recipient of the communication. Thus, the receiver is left in its active mode (dissipating both dynamic and static power) even if it will not be involved in any communication for a certain number of clock cycles.

In this paper, we propose a mechanism for selectively switching-off the receiver module in the transceiver of radio-hubs when they are not involved in any communication for a number of clock cycles. The question on how to know whether, in the next amount of clock cycles, the current radio-hub will not be a recipient of any wireless communication, can be easily answered as follows. As soon as the receiver module of the radio-hub detects the starting of a wireless communication, it decodes the first few bits of the header flit of the packet corresponding to the destination address, and the bits encoding the packet size, PS. Assuming the wormhole switching, the number of clock cycles needed for completing the transmission of the packet can be computed as PS/DR, where DR is the data rate of the radio channel. Thus, during such period, the current receiver can be turned into a sleep mode since the radio channel will be used for delivering information involving other radio-hubs.

The proposed energy management scheme has been designed, implemented and applied on two 256-core WiNoC configurations with four and sixteen radio-hubs under different traffic scenarios, traffic loads, and packet sizes. We found that interesting energy savings, up to 30%, can be obtained without any impact on the performance metrics and with a negligible impact on the control logic while implementing the proposed scheme.

978-3-9815370-6-2/DATE16/ ©2016 EDAA

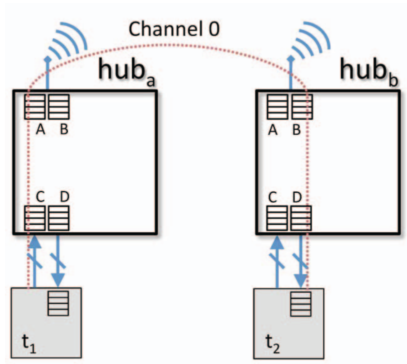


Fig. 1. Wireless communication between two tiles using radio-hubs. Four classes of radio-hub buffers are involved: (A) *antenna_buffer_TX*$_{Ch_x}$, (B) *antenna_buffer_RX*$_{Ch_x}$, (C) *buffer_from_tile*$_i$, and (D) *buffer_to_tile*$_i$

II. Related Work

Wireless Network-on-Chip (WiNoC) paradigm has been recently proposed as a CMOS compatible solution for addressing the scalability problems affecting the on-chip communication system for future manycore architectures. The WiNoC communication paradigm is constituted by a wireless backbone [2] that operates in conjunction with traditional wire-based NoC architectures [7]. This new capability requires new hardware resources such as radio frequency (RF) transceivers and antennas. Unfortunately, such devices consume a relevant fraction of the overall chip silicon area. For these reasons, several architectures have been introduced to obtain a good compromise between area overhead and expected communication bandwidth [3], [8]. The power consumption of radio-hubs which are responsible to ensure wireless links among processing elements is still an open issue: several works present techniques to reduce communication energy by mean of a tunable transmitting power mechanism to reduce the transmitter power contribution [9], [10]. Unfortunately, according with the WiNoC power model introduced in [11], a big portion of the radio-hubs' energy is due to active receivers which consume a relevant portion of the overall communication infrastructure energy.

Due to the aggressive leakage power consumption which affects modern CMOS devices, the receiver consumes power also when it is not involved in any communication. For these reasons, in this paper we introduce a power gating mechanism which safely switch-off a portion of the receivers' logic when a specific receiver is not a recipient of the incoming stream of data. The proposed mechanism will be exhaustively described in the next sections of this paper.

III. WIRXSleep Power Reduction Strategy

A. Reference Radio-Hub Architecture

To understand the proposed WIRXSleep power reduction approach is essential to get a picture of the main elements constituting the radio-hub wireless communication architecture and the involved data flows.

There are two fundamental types of communications handled by a radio-hub component: wired Tile-Hub connections between radio-hub and tiles, and wireless Hub-Hub connections between different radio-hubs. Fig. 1 shows the four classes of buffers involved in a simple scenario with two tiles t_1 and t_2, connected to hub_a and hub_b, respectively. Further, we assume that hub_a and hub_b support transmissions on a common channel Ch_0. In wired Tile-Hub communications, two kind of radio-hub buffers are involved, depending on the direction of the data flow: The input buffer (C) of the radio-hub, that receives data from a tile i, referred to as *buffer_from_tile*$_i$, and on the opposite direction, the output buffer (D) of the radio-hub, that stores data to be sent to tile i, referred to as *buffer_to_tile*$_i$. Discussions about the low-level mechanisms involved in the communication protocol are beyond the scope of this paper. It is sufficient to know that these wired Tile-Hub connections use identical communication phases as in wired Tile-Tile connections. With regard to wireless Hub-Hub communications, we will use the notation *antenna_buffer_RX*$_{Ch_x}$ and *antenna_buffer_TX*$_{Ch_x}$ to denote the buffers used for receiving and transmitting data, respectively, using the radio channel Ch_x (see buffers A and B in Fig. 1). Of course, usually more than a single tile node is connected to each radio-hub and also many different radio-hubs can share a common channel, however, we will refer to the trivial example in Fig. 1 for the sake of clarity.

B. WIRXSleep *Formal Description*

Recalling again the reference architecture depicted in Fig. 1, we can see how radio-hub elements in the WiNoC consist of hardware components (buffers and logic) that can be functionally associated with four main types of data flow: (1) Wireless data transmission starting from the current radio-hub (2) Wireless data reception coming from another radio-hub (3) Flit dispatching from the current radio-hub towards a tile via wired Tile-Hub connection (4) Flit reception from a tile via wired Tile-Hub connection.

The main idea behind the WIRXSleep power management scheme is to focus on power contributions of those components that are involved in the wireless data reception. However, even if no data is being transmitted on channel Ch_x and all the involved buffers are empty, we cannot trivially disable the related radio-hub transceiver components. To better explain this concept, we briefly show two simple counterexamples. First, let us consider a given radio-hub hub_i, connected to a set of tiles T_i and supporting transmissions on a channel Ch_x. Let us also suppose that hub_i is not currently transmitting on channel Ch_x. Suppose also that the *antenna_buffer_TX*$_{Ch_x}$ and all the *buffer_from_tile* are empty. This is apparently an ideal situation for switching off transmission buffers and logic: no data is queued on the antenna for wireless transmission and no data is coming from tiles. However, we cannot safely disable these components, since new transmission events could be scheduled in any moment, depending on incoming data from tiles belonging to the set T_i. These are complex and poorly predictable events, *e.g.*, depending on the packet generation at each Processing Element of T_i, on dynamic behavior of flit routing between remote tiles, on traffic congestion, *etc.*. On the other side, let us consider the same hub_i when no data is being received on channel Ch_x. Similarly to the previous case, we cannot turn off buffers and logic related to the wireless data receiving, since we cannot safely predict when another hub will start a transmission having the current hub_i as recipient. To summarize, these two examples show that periods of wireless channel inactivity are not safe to make assumptions regarding future events in the considered radio-hub WiNoC architecture.

Conversely, the idea behind WIRXSleep is to exploit information gathered from new channel events. Assuming that a new transmission, started by a radio-hub different from hub_i, begins on channel Ch_x, only two different situations can happen: (i) the radio-hub hub_i is the recipient, (ii) the hub_i is not the recipient. The proposed approach aims to detect situations belonging to the case (ii), setting radio-hub hub_i for channel Ch_x to status `WIRXSLEEP_ON`. The meaning of such status is that hub_i will not be the target of any transmission on channel Ch_x for a certain amount of cycles. With regard to this, it should be pointed out that a `WIRXSLEEP_ON` status on hub_i does not necessarily mean that hardware blocks related to wireless data receivers can be switched off. In fact, the proposed approach implements a selective strategy, checking cycle-by-cycle components that can be safely disabled in the `WIRXSLEEP_ON` status. For example, in order to disable the *buffer_to_tile* of hub_i is not sufficient to check whether they are empty or not, since a previously filled upstream *antenna_buffer_RX* could dispatch new flits to a given *buffer_to_tile*, even if no wireless data is currently being received by hub_i.

WIRXSleep control logic implements a simple mechanism to detected safe conditions for performing such power down decisions. Fig. 2 shows the WIRXSleep algorithm performed on each radio-hub. At each cycle, the behavior of the algorithm is dictated by the value of *WIRXStatus*, a vector of booleans storing the `WIRXSLEEP_ON/OFF` status for each channel supported by the radio-hub. The value of *WIRXStatus* is initially set to `WIRXSLEEP_OFF` for all its elements, meaning that no power reduction strategy is being applied.

When a new transmission is detected on a channel Ch_x, the corresponding value of *WIRXStatus* is updated. In particular, if hub_i is not the target, we can safely assume that no other transmission on channel Ch_x will be sent to hub_i at least for a given number of cycles. Please notice that a precise and deterministic number of cycles can be computed, since we are assuming a packet transmission token policy in which the token is released only after the whole packet has been transmitted. Thus, the number of safe sleep cycles can be computed as:

$$N_{sleep} = T_{delay} \times packet\ size \times clock\ frequency \quad (1)$$

where *packet size* is expressed in number of flits and T_{delay} is the amount of time required to transmit a flit, computed as $flit\ size/DR$, where DR is the data rate of the antenna in bit/s and *flit size* is the flit size in bits. For example, assuming a clock frequency of 1 GHz, a flit size of 32 bits and a 16 Gbs mm-Wave antenna using On-Off keying (OOK) modulation, T_{delay} is 2 ns, corresponding to 2 cycles. When `WIRXSLEEP_ON` status is enabled, hub_i can start ignoring wireless flit receiving events for N_{sleep} clock cycles, since it is not feasible to complete the packet transmission in less cycles, for a given antenna data rate, flit size, packet length and clock frequency. It is important to notice that, this is a conservative assumption: we do not know *when* the transmission will exactly end: heavy traffic loads, congestion events could introduce further buffer delays, *etc.*.

Returning to Fig. 2, if a `WIRXSLEEP_ON` status is present for a given channel Ch_i, the algorithm must perform a set of checks to determine which components can be disabled.

```
Require: WIRXStatus[]
switch_on_buffer_to_tile = FALSE
for all ch_i ∈ RX_Channels do
    if antenna_buffer_RX[ch_i].isEmpty() then
        if WIRXStatus[ch_i] is WIRXSLEEP_ON then
            disable(antenna_buffer_RX[ch_i])
        end if
    else
        switch_on_buffer_to_tile = TRUE
    end if
end for
if switch_on_buffer_to_tile = FALSE then
    for all port ∈ Buffer_to_tile_ports do
        if buffer_to_tile[port].isEmpty() then
            disable(buffer_to_tile[port])
        end if
    end for
else
    enable all buffer_to_tile
end if
```

Fig. 2. WIRXSleep algorithm performed at each radio-hub

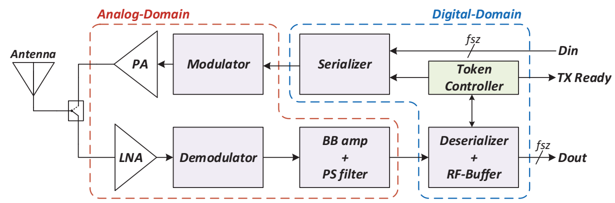


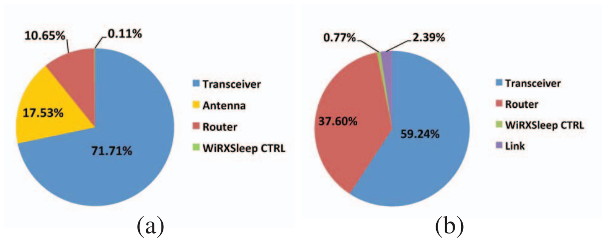
Fig. 3. A generic WiNoC transceiver.

First of all, channels having an empty antenna RX buffer can be disabled when the corresponding value in *WIRXStatus* is `WIRXSLEEP_ON`, since, as stated above, no new data can be received from that channel for N_{sleep} clock cycles. This includes disabling contribution from $antenna\_buffer\_RX$, transceiver RX biasing and leakage. It is important to note how a special *switch_on_buffer_to_tile* flag is set as `TRUE` if any of the antenna RX buffers is found not empty, regardless the value of the *WIRXStatus* vector. This flag plays a fundamental role in the second part of the algorithm, during the selection of the *buffer_to_tile* buffers that can be disabled. In particular, the whole second part should be skipped if any data is present in some antenna RX buffers. As it can be observed from Fig. 1, each *buffer_to_tile* can receive data from an antenna RX buffer that has been filled in a previous wireless reception. So, when data is present on antenna RX buffers all the subsequent *buffer_to_tile* buffers have to be re-enabled.

It is essential to point out again the particular role played by the *WIRXStatus* vector: each of its elements might be set as `WIRXSLEEP_ON` for a given amount of clock cycles, but the corresponding power saving decisions must be updated cycle-by-cycle. This is a conservative approach, chosen by design for *WIRXSleep* in order to achieve an appreciable and safe power saving result adding only a very small complexity.

C. Hardware Implementation

Fig. 3, shows a generic WiNoC single channel OOK transceiver constituted by a transmitter and a receiver which share the same antenna by means of a RF-switch. The main task of transmitters consists in adapting the data incoming from the electrical medium to the wireless medium by means of an antenna. In particular, a transmitter is constituted by a serializer which converts parallel streams of data (flits) in a serial fashion; a modulator which adapts a low frequency signal



(a) (b)

Fig. 4. Area (a) and Power (b) breakdown of the radio-hub.

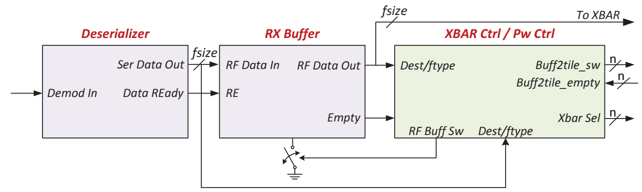


Fig. 5. Internal architecture of power gating controller, where n is the number of Buffer2Tile (number of ports) and *fsize* is the flit size.

into a high frequency one; a power amplifier directly connected to the antenna which delivers the required transmitting power. It also comprises a token flow controller to decode incoming information and manage the token flow mechanism, in order to share the wireless medium among radio-hubs.

Conversely, the receiver presents a Low Noise Amplifier (LNA) which amplifies incoming signals introducing less noise as possible; a demodulator which shifts high frequency signals into the baseband; a baseband amplifier and a Pulse Shift Filter that amplify and shape signals to obtain a digital signal which is stored in a specific buffer.

Observing Fig. 3, it should be pointed out that the transceiver can be partitioned into two different domains, namely, analog and digital. In fact, amplifiers (such as LNA, PA and baseband amplifier) and, both modulator and demodulator are analog circuitries, while serializer and deserializer, RF-buffer and the token flow controller are digital circuits.

Fig. 5 highlights the components involved in the execution of the algorithm presented in Sec. III-B. In particular, the XBAR Ctrl/Pw Ctrl block is responsible for decoding flits that come from the deserializer. In case of a header flit, the XBAR controller can decide the correct destination by driving the XBAR selector accordingly (Xbar_sel). At the same time, if a particular radio-hub is not a recipient of the flit, the power controller checks whether the *antenna_buffer_RX* is empty (if the relative RX Buffer pin is asserted). In the same way, if each *buffer_to_tile* buffer is empty (*Buff2tile_empty* pins), the Pw Controller can safely switch-off both RF and *buffer_to_tile* buffers via the *RF Buff Sw* and *Buff2tile_sw* switches. Since Pw Controller is directly connected to the deserializer, when a header flit is destined to that particular radio-hub, it can switch-on both RF and *buffer_to_tile* buffers one cycle before storing flits in such buffers. Based on this, the proposed implementation does not result in any penalty in terms of latency.

The XBar Ctrl/Pw Ctrl module has been designed HDL, synthesized, mapped on a 45 nm CMOS standard cell library from TSMC and evaluated with Synopsys Design Compiler. As expected, its overhead both in terms of silicon area and power is negligible as respect to the area and power figures of the radio-hub. Specifically, a radio-hub augmented with the WIRXSleep logic increases its power and area by 0.8% and 0.1%, respectively. Power and area breakdown of the radio-hub implementation are shown in Fig. 4.

IV. Experiments

In this section, the WIRXSleep scheme is assessed on two WiNoC configurations with four and sixteen radio-hubs, respectively. Both 4- and 16-radio-hubs configurations refer to a 256-node NoC in which 4 and 16 radio-hubs are used for implementing wireless long-range communications, respectively. The 16×16 mesh topology is partitioned into a number of regular regions depending on the number or radio-hubs. Specifically, let n be the number of considered radio-hubs. Each radio-hub is associated with one and only one regular region of size $256/n$. Intra-region communication is carried out by means of the conventional electric NoC whereas inter-region communication is performed wirelessly by means of radio-hubs. The simulation environment is based on Noxim [12], a cycle accurate NoC simulator able to assess WiNoC architectures both in terms of performance and energy. The simulation parameters used in the setup are summarized in Tab. I.

TABLE I. Simulation parameters.

Parameter	Value
Network size [core]	256 (16×16)
Number of radio-hubs	4, 16
Switching technique	Wormhole [13]
Radio Access Control Mechanism	Token based
Wireless data rate [Gbps]	16
Packet size [flit]	4, 8, 16, 32
Flit size [bit]	64
Router input buffer size [flit]	4
Radio-hub input buffer size [flit]	4
Radio-hub antenna buffer size [flit]	16

Let us, first, analyze the performance and energy differences between the two considered network configurations with 4 (WiNoC-4) and 16 (WiNoC-16) radio-hubs, assuming a fixed packet size of 16 flits. We consider a uniform traffic with different locality percentages. A uniform traffic with locality l corresponds to a uniform traffic scenario in which l percent of communications are intra-region while the remaining $100 - l$ percent of communications are inter-region. Fig. 6 shows the average communication delay as the packet

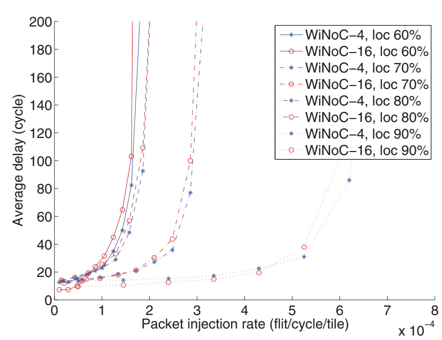


Fig. 6. Average communication delay for different packet injection rates under uniform traffic with different locality levels.

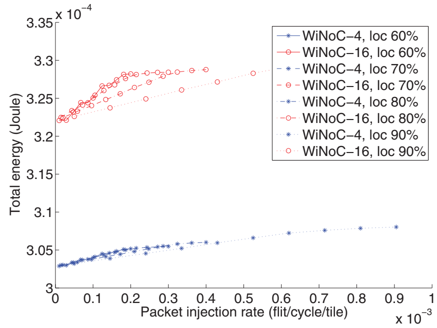


Fig. 7. Total communication energy consumption for different packet injection rates under uniform traffic with different locality levels.

injection rate (pir) has made to vary. As it can be noticed, for low pir values WiNoC-16 and WiNoC-4 performances are quite similar, whereas as the pir increases WiNoC-4 exhibits lower communication delay than WiNoC-16. To explain this behavior, it should be recalled that having more radio-hubs implies having longer token round time. So, when a radio-hub is holding the token for transmitting a packet, the higher is the pir the higher is the probability that other radio-hubs will enqueue packets for a new transmission. At this point, having a short ring of 4 radio-hubs introduces less waiting time when a lot of radio-hubs is asking for the token. For example, assuming a wireless transmission rate of 1 flit per cycle, in the worst case scenario a radio-hub of WiNoC-4 should wait for 3 other radio-hub transmissions, for a total of $16 \times 3 = 48$ cycles, while in the WiNoC-16 configuration the waiting time would be of $16 \times 15 = 690$ cycles.

Fig. 7 shows the total communication energy for the same scenarios. To better understand the results, we should recall how the total energy consists of two main contributions: a static energy, related to the physical presence of hardware components, and a dynamic energy related to the activity performed by such components. As expected, the energy consumption in terms of absolute value is bigger when more radio-hubs are present, mainly because of the static power contributions introduced. Notice how higher locality slightly mitigates energy consumption when 16 radio-hubs are used, since less wireless transmission are started. Also, the energy consumed increases for higher pir values, due dynamic energy contributions associated to the increased number of transmission events.

Let us, now, assess the energy saving obtained by the application of the proposed technique on the 4- and 16-radio-hubs WiNoC configurations. We analyze the impact on energy saving due to packet size and traffic characteristic. The percentage energy saving results along with the average fraction of the time in which the receiver module of the transceiver of a generic radio-hub is in sleep mode are shown in Fig. 8. As expected, as the packet size increases the energy saving increases due to the fact that the wireless transmission time per communication increases, making it possible for the radio-hubs, not involved in the current communication, to remain in their receiving sleep mode for more time, as can be observed by looking at the percentage sleep curve in the graphs. Such behavior is partially smoothed under higher traffic locality, due to the reduced amount of long range wireless communications. This can be more clearly observed in Fig. 9 in which the energy

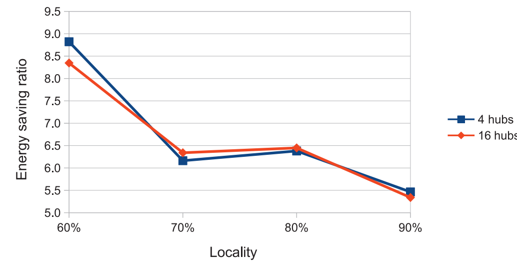


Fig. 9. Energy saving ratio under different traffic locality levels.

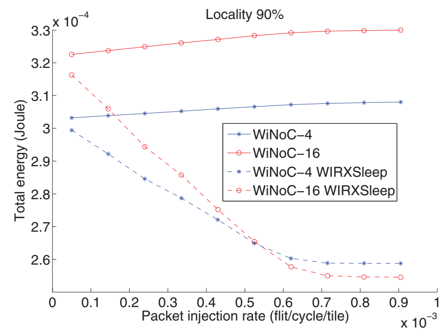


Fig. 10. Total communication energy consumption for different packet injection rates with and without WIRXSleep.

saving ratio, obtained with 32 and 4 packet sizes scenarios, shrinks from a 9x to a 5x factor with higher locality levels.

It is also worth noticing that, the effectiveness of the proposed technique improves as the number of radio-hubs increases. This behavior can be explained by the fact that the more radio-hubs share a token, the more radio-hubs will take benefit of not being the recipient of the current transmission (thus setting their internal *WIRXStatus* as `WIRXSLEEP_ON`). A trivial counterexample is a network where there are only two radio-hubs sharing a single token: being alternatively source and destination of the transmission, they could never enable a `WIRXSLEEP_ON` status.

The energy saving figures presented so far have been computed for a given packet injection rate. Let us, now, analyze the impact of the packet injection rate on the effectiveness of the proposed technique. Fig. 10 shows the total communication energy with and without WIRXSleep, for WiNoC-4 and WiNoC-16 under uniform traffic with locality 80%. As it can be observed, as the pir increases, the total communication energy of the conventional WiNoC, increases whereas it decreases when WIRXSleep is used. Such behavior can be explained by recalling that the energy optimization opportunities exposed to WIRXSleep increase as the number of wireless communications increases. Thus, as the pir increases, the number of communications (both wired and wireless) increases as well with a consequent energy saving due to the use of WIRXSleep technique.

V. Conclusion

The radio-hubs in a WiNoC architecture are one of the main contributors of the total communication energy. In particular, the receiver and the buffers into a radio-hub account

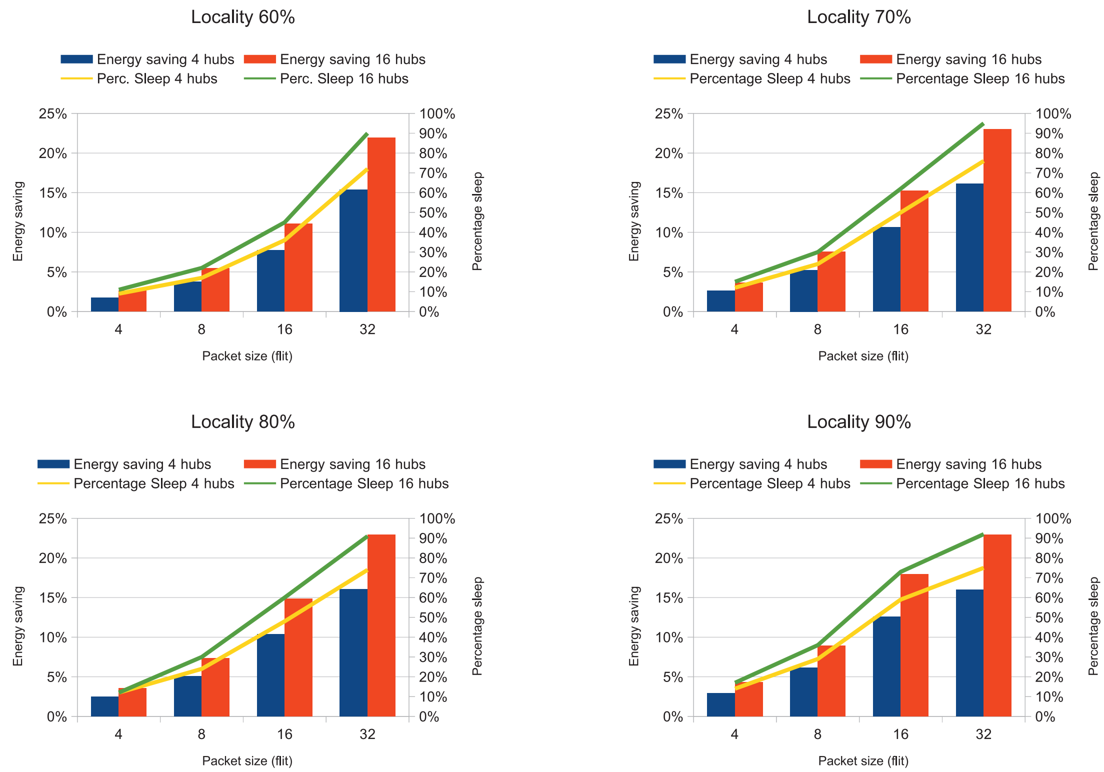


Fig. 8. Percentage energy saving for different packet sizes under different traffic locality levels.

for a significant fraction of its total energy consumption. Thus, reducing the energy consumption of such power-hungry elements is mandatory in the context of energy efficient WiNoC architectures. In this paper, we have presented a novel energy management scheme aimed at reducing the overall energy consumption of radio-hubs by selectively switching off their receivers and buffers when it is predicted that they will not be involved in any communication for the next forthcoming clock cycles. The proposed scheme, namely WIRXSleep, attacks both the dynamic and static power contributions resulting in significant energy saving. WIRXSleep has been assessed on different WiNoC topologies, with different configurations, and under different traffic scenarios. Total communication energy savings, up to 25%, have been observed without any impact on performance metrics and with a negligible impact on silicon area.

References

[1] L. Carloni, P. Pande, and Y. Xie, "Networks-on-chip in emerging interconnect paradigms: Advantages and challenges," in *ACM/IEEE International Symposium on Networks-on-Chip*, 2009, pp. 93–102.

[2] S. Deb, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Wireless NoC as interconnection backbone for multicore chips: Promises and challenges," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 228–239, 2012.

[3] S. Deb, K. Chang, X. Yu, S. P. Sah, M. Cosic, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Design of an energy-efficient CMOS-compatible NoC architecture with millimeter-wave wireless interconnects," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2382–2396, Dec. 2013.

[4] X. Yu, S. P. Sah, S. Deb, P. P. Pande, B. Belzer, and D. Heo, "A wideband body-enabled millimeter-wave transceiver for wireless network-on-chip," in *International Midwest Symposium on Circuits and Systems*, 2011, pp. 1–4.

[5] D. C. Daly and A. P. Chandrakasan, "An energy-efficient ook transceiver for wireless sensor networks," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 5, pp. 1003–1011, 2007.

[6] A. Mineo, M. Palesi, G. Ascia, and V. Catania, "Runtime tunable transmitting power technique in mm-wave WiNoC architectures," *IEEE Transactions on Very Large Scale Integration Systems*, vol. PP, no. 99, pp. 1–1, 2015.

[7] L. Benini and G. D. Micheli, "Networks on chips: a new SoC paradigm," *IEEE Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[8] X. Yu, J. Baylon, P. Wettin, D. Heo, P. P. Pande, and S. Mirabbasi, "Architecture and design of multi-channel millimeter-wave wireless network-on-chip," *IEEE Design & Test*, vol. 31, no. 6, pp. 19–28, 2014.

[9] A. Mineo, M. Palesi, G. Ascia, and V. Catania, "An adaptive transmitting power technique for energy efficient mm-wave wireless NoCs," in *Design Automation and Test in Europe*, 2014, pp. 1–6.

[10] A. Mineo, M. S. Rusli, M. Palesi, G. Ascia, V. Catania, and M. N. Marsono, "A closed loop transmitting power self-calibration scheme for energy efficient winoc architectures," in *Design, Automation Test in Europe Conference (DATE15)*, Mar. 2015.

[11] S. Abadal, M. Iannazzo, M. Nemirovsky, A. Cabellos-Aparicio, H. Lee, and E. Alarcon, "On the area and energy scalability of wireless network-on-chip: A model-based benchmarked design space exploration," Jul. 2014, p. xx.

[12] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Noxim: An open, extensible and cycle-accurate network on chip simulator," in *IEEE International Conference on Application-specific Systems, Architectures and Processors*, Toronto, Canada, Jul. 2015.

[13] P. Mohapatra, "Wormhole routing techniques for directly connected multicomputer systems," *ACM Computing Surveys*, vol. 30, no. 8, pp. 374–410, Sep. 1998.