Improving Energy Consumption of NoC based Architectures through Approximate Communication

Giuseppe Ascia, Vincenzo Catania,
Salvatore Monteleone, Maurizio Palesi, and Davide Patti
Dept. of Electrical, Electronic and Computer Engineering
University of Catania
Catania, Italy
first.last@dieei.unict.it

John Jose
Dept. of Computer Science and Engineering
Indian Institute of Technology Guwahati
Guwahati, Assam, India
johnjose@iitg.ernet.in

***Abstract*—This paper presents an approximate communication technique aimed at improving the energy efficiency of Network-on-Chip (NoC) based architectures. The basic idea is a selective adjustment, at run-time, of the voltage swing of the links of the NoC to obtain a trade-off between the communication energy consumption and the accuracy of the running application. The proposed technique is applied to a case study, namely, a JPEG encoder resulting in an energy saving up to 70% with a negligible impact on the quality of the compressed image.**

***Keywords*—Approximate computing; Approximate communication; Network-on-Chip; Energy efficiency.**

I. Introduction

Approximate computing [1] can be viewed as a viable solution for improving the efficiency of systems for which accuracy is not crucial. Many modern applications are fundamentally approximate: precise answers are unnecessary or even impossible in domains such as computer vision, machine learning, speech recognition, and physical simulation [2], [3]. Such imprecision is expressed by means of a variation in the output of the application as compared to the case in which the application is executed in a reference reliable system.

Several works in the literature investigate the use of approximate computing techniques in applications that are tolerant to a reduction of the quality. Unfortunately, the majority of such works are focused on the computing subsystem while just few of them have explored the communication subsystem. The evolution from shared-memory based multicore architectures to Network-on-Chip (NoC) based manycore architectures has emphasized the role played by the on-chip communication system since it has a relevant impact on several important design optimization metrics, including, performance and energy consumption. In particular, the links of the NoC account for a significant fraction of the overall communication energy budget [4]–[8].

In this paper, we introduce the concept of *approximate communication* as a viable solution for improving the energy efficiency of a NoC based architecture. In fact, the reliability of a link (usually measured in terms of its bit error rate — BER) depends on the voltage swing used for separating the logical zero and one. If, from one hand, reducing the voltage swing of a bit-line reduces its energy consumption (there is a direct quadratic proportionality between power dissipation and voltage swing), on the other hand, its BER increases. Based on this, we propose a mechanism aimed at selectively tuning the voltage swing of a link based on the specific forgiving nature of the particular communication flow. Specifically, communications carrying error tolerant information are routed through links operating at a reduced voltage swing thus saving energy.

We augment the general NoC router architecture with a hardware module that allows the tuning at run-time of the voltage swing of the bit-lines of the outputs links. We show its negligible impacts on both critical path, and power consumption of the router. We also propose a pragma based programming environment through which the application developer can annotate those data inducing communications (*i.e.*, among processing cores and memory controller cores), that are are error tolerant and thus can travel on low energy/low reliable links. The proposed technique is applied to a reference architecture implementing a JPEG encoder showing that up to 70% of energy saving can be obtained with a degradation of the image quality (RMSE metric) that is less than 10^{-5}.

II. Approximate Communication

A. Basic Idea

The energy consumption of the links of a NoC accounts for a significant fraction of the overall communication energy budget [4]. Their energy consumption is quadratically related to the voltage swing used for encoding the one and zero bits transmitted by the link. Thus, the reduction of the voltage swing would result to a quadratically reduction of the energy consumption of the link. The price to pay for such energy reduction is a degradation of the nominal reliability of the link. One common metric for measuring such a reliability is the bit error rate (BER).

978-1-5386-5683-9/18/$31.00

During the execution of an application, data are loaded and stored from and to the main memory. In a NoC based manycore architecture, such data are packetized and transmitted from memory controllers to cores and vice versa by means of the on-chip communication fabric. Now, let us consider the application has a forgiving nature, that means, it is tolerant (to a certain extent) to errors affecting data involved in load/store induced communications. We assume that the developer is aware of the subset of data structures in the application that, if affected by errors (*e.g.*, data corruption and approximation), do not cause a failure (*e.g.*, a crash) but only impact the quality of results. Let us now suppose the availability of a programming environment through which the developer can make aware the underlying communication system that the load/store induced communications related to the access to that specific data structures are error tolerant. At this point, the system can selectively reconfigure at run-time the links of the routing path through which that particular communication will be mapped, by reducing their voltage swing. As a result, communication energy saving is obtained at the expenses of a BER increase of the links involved in that communication. However, the consequent increase of the BER will be tolerated by the application thanks to the error tolerance characteristic of the data involved in that communication.

In the above discussion, we have assumed that the developer is aware about the error tolerance level of data structures of the application. Such assumption will be considered in the rest of the paper. The definition of techniques able to automatically determine whether a given data structure is tolerant to errors and, more in general, the sensitivity of the application quality results with respect to the approximation level/error rate, is left as future work. In the next section, we will present the design of the voltage swing link reconfiguration logic and the programming environment through which the developer can mark error tolerant data structures of the application.

B. Hardware Interface

Fig. 1 shows the hardware module designed to implement the reconfigurable link voltage swing. The module refers to a single bit-line and it is of course instanced for the link width. As it can be observed, the bit-line is preceded by a chain formed by a demultiplexer, two tapered buffers as line drivers and two tristate buffers based on transmission gate logic. With this solution, if the select input is high (low), the full (low) swing path is active and the low (high) swing path is disconnected by the high impedance state introduced by tristate buffers. The level restorer circuit restores the signal at full swing if the signal on the line is set to low swing, or maintain the original swing if the signal is in full swing mode.

By acting on the *Sel* input of the circuit, the link is configured to operate either at reliable mode or unreliable/low energy mode. The *Sel* input is driven by a specific flag into the header flit of the packet that dictates whether the packet has to

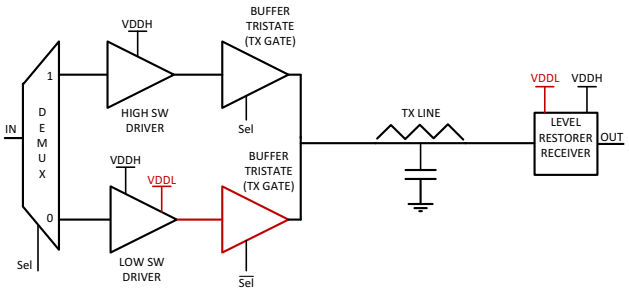

Fig. 1. Reconfigurable Link voltage swing for a bit-line.

TABLE I
HSPICE SIMULATION RESULTS FOR A BIT-LINE OF THE LINK.

	Conventional VDDH	Configurable	
		VDDH	VDDL
Technology	1.1 V	10 metal	45 nm CMOS LVT
Interconnect (Metal 7)	Width 0.4 μm, Space 0.32 μm, Length 2.8 mm		
	Rwire 225 Ω, Cwire 946 fF		
Supply	1.1 V	1.1 V	0.6 V
Worst case total delay	214 ps	410 ps	
Avg. Energy/Transition	512 fJ	527 fJ	152 fJ
BER	1.3E-17	1.3E-17	3.8E-6

be transmitted through a conventional voltage swing link or a low voltage swing link. Please notice that, even if the packet is marked to travel on low voltage swing link, the header flit will be always transmitted on nominal voltage swing link as it contains control information which are not error tolerant.

The reconfigurable voltage swing link has been designed and analyzed. The design has been targeted for working at a clock frequency of 2 GHz (which is the target clock speed of our baseline router). The analysis has been carried out with HSPICE using a 45 nm CMOS LVT library from Nangate [9] which provides 10 metal layers. The parasitics extraction from layout has been made using Cadence Virtuoso. The results are reported in Table I. The table compares the conventional link using a single VDDH voltage swing, with the proposed configurable link supporting two voltage swing levels, namely, VDDH and VDDL. We considered a conventional VDDH of 1.1 V and a VDDL of 0.6 V which determine a BER of 1.3E-17 and 3.8E-6, respectively. The worst case total delay of the proposed configurable link increases but it is still below the clock period of the baseline router. The energy per bit of the proposed configurable link increases less than 3% when it works at VDDH. This is due to the overhead introduced by the reconfiguration logic. However, when it works at VDDL, the energy saving is close to 70%.

C. Software Interface

The developer makes use of pragmas to annotate data structures involved in specific region of the code and for which he wants to exploit the approximate communication in exchange to energy saving. In the following piece of code:

```
#pragma resilient(w)
for (i=0; i<n; i++)
```

```
v[i] = f(w[i]);
```

the array `w` is read accessed. The pragma `resilient(w)` specifies that, the communications induced by the load instructions involving `w` will be implemented by packets that configure the traversed NoC links in their low voltage swing mode. It should be pointed out that, a load induced communication transaction results in two messages from the processing core to the memory controller core (request packet) and from the memory controller core to the processing core (response packet). The request packet will always travel on links configured to operate in their nominal voltage swing. This is because a request packet transports control information (*e.g.*, memory address and data size) that are not error tolerant. On the other side, the response packet transports data that, if belonging to a data structure marked as resilient, can travel on links configured to work in the low voltage swing mode. Thus, to correctly operate, the request packet is enriched with an appropriate flag used to inform the memory controller core if the response packet has to travel on links configured to operate in their nominal or low voltage swing mode.

For store induced communications, let us consider the following piece of code:

```
#pragma resilient(v)
for (i=0; i<n; i++)
  v[i] = f(w[i]);
```

where the array `v` is write accessed. The pragma `resilient(v)` here specifies that, the communications induced by the store instructions involving `v` will be implemented by packets that configure the traversed NoC links in their low voltage swing mode.

III. Experiments

The proposed technique has been applied to the exploration of the energy *vs.* quality trade-off during the design of a JPEG encoder.

We considered a reference pipelined JPEG encoder implementation imported from AxBench [10] and shown in Fig. 2(a). The application has been then mapped on a mesh based NoC as shown in Fig. 2(b). We considered two memories from which each task of the encoder fetches its data and to which writes the results that will be used by the subsequent task in the pipeline. *Level Shift*, *DCT*, and *Quantize* tasks use memory *Mem 1*, whereas *Entropy Encode* fetches its data from memory *Mem 1* and stores the encoded stream to memory *Mem 2*.

From now on, we consider two configurations of the NoC, namely, baseline and approx. The baseline NoC is a conventional NoC in which links work at their nominal voltage swing (VDDH) whereas the approx NoC implements the proposed technique in which links can be selectively configured at runtime to work at the nominal voltage swing (VDDH) or at the reduced voltage swing (VDDL).

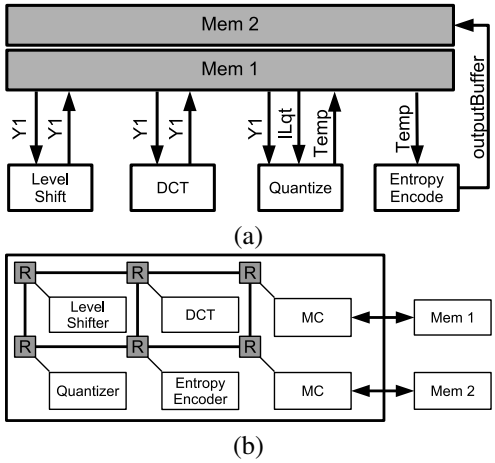

Fig. 2. Pipelined JPEG encoder (a) and its mapping on a mesh-based NoC (b).

We analyze the impact on energy consumption and quality of the encoded image by incrementally approximating the input(s) and output(s) of each task in the JPEG pipeline. That is, we consider 10 different configurations as follows: in configuration 0, no data flows are approximated; in configuration 1, only load Y1 induced communication [see Fig 2(a)] are approximated; in configuration 2, load Y1 and store Y1 induced communications are approximated, and so on.

Fig. 3 shows the energy consumption and a measure of the image quality for the different configurations. The metric used to assess the image quality, is the Root-Mean-Square Error (RMSE) of the RGB pixels of the image encoded by the baseline NoC and of the image encoded by the approx NoC. The energy consumption is normalized as respect to the energy consumption of the baseline NoC. As expected, as the approximation level increases, the energy consumption decreases and the RMSE increases. It is interesting to observe that the RMSE suddenly increases passing from configuration 6 to configuration 7. In fact, in configuration 7, the approximation in write mode [data flow (7)] to data structure generated by the quantization task, has a strong impact on the accuracy metric. To better understand the impact of the approximation of the considered data structures on the quality of the encoded image, we perform a sensitivity analysis. We apply the approximation to each data structure in read and write mode but, this time, in isolation. Then, we measure the impact of the quality of the encoded image by computing the RMSE. Fig. 4 shows the result of the sensitivity analysis. According to the previous experiment, it can be observed how the maximum sensitivity is found for `Temp` data structure both in read (in) and write mode (out). Based on this analysis, let us consider a new configuration, namely, *opt*, in which we apply the approximation technique only to the less sensitive data structures (those marked in Fig. 4 with a red bounding box). In this way, we will limit the negative impact on the

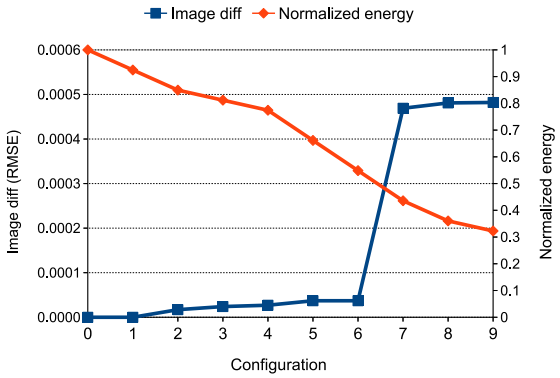

Fig. 3. Energy *vs.* image quality.

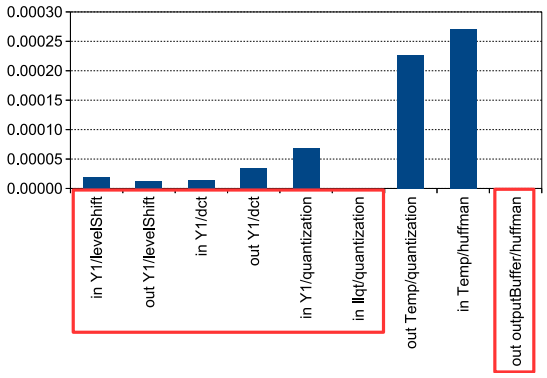

Fig. 4. Sensitivity analysis.

quality of the encoded image as we are acting only on the less sensitive data structures. Fig. 5 shows the RMSE *vs.* normalized energy for all the configurations considered before along with the *opt* configuration derived by the observation of the results of the sensitivity analysis. As it can be observed, the *opt* configuration provides an interesting trade-off between the two considered indexes. In fact, it provides almost 70% of energy reduction and a RMSE of 10^{-5} with respect to the baseline configuration.

IV. Conclusions

In this paper, we have presented an *approximate communication* technique aimed at improving the energy efficiency of NoC based architectures. This technique relies on the pragma based annotation of data structures (and their scopes) that are error tolerant. Then, the load and store induced messages related to the access on such data structures will be realized by packets that selectively configure the NoC links they traverse to operate at a reduced low voltage swing. This results in a reduction of the energy consumption in exchange to a degradation of the quality of the application output. We have also presented the design of the hardware reconfiguration logic implementing the proposed approximate communication technique. We have shown how its integration into a conventional

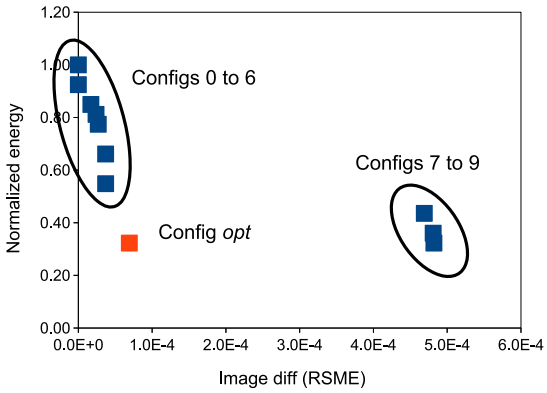

Fig. 5. RMSE and normalized energy consumption for the considered configurations and *opt* configuration derived by the sensitivity analysis.

NoC router has a negligible impact in terms of timing and power dissipation. The proposed technique has been applied to the design of an energy efficient JPEG encoder. We found that, as compared to a baseline implementation, up to 70% of energy saving is obtained with an image quality which differs less than 10^{-5} according to RMSE metric. Future works will be mainly focused on validating the proposed techniques on other relevant applications. Specifically, will be more extensively studied the trade-off between energy saving and BER increase. Further, the impact on the critical path when the link is configured to work at a low voltage swing will be also investigated.

References

[1] S. Mittal, "A survey of techniques for approximate computing," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 62:1–62:33, Mar. 2016.

[2] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *2013 18th IEEE European Test Symposium (ETS)*, May 2013, pp. 1–6.

[3] V. Catania, A. Mineo, S. Monteleone, and D. Patti, "Distributed topology discovery in self-assembled nano network-on-chip," *Computers and Electrical Engineering*, vol. 40, no. 8, pp. 292–306, 2014.

[4] ITRS. (2015) International technology roadmap for semiconductors 2.0. [Online]. Available: http://www.itrs2.net

[5] M. Kim, D. Kim, and G. E. S. Sobelman, "Network-on-chip link analysis under power and performance constraints," in *IEEE International Symposium on Circuits and Systems*, May 2006.

[6] E. Moréac, A. Rossi, J. Laurent, and P. Bomel, "Crosstalk-aware link power model for networks-on-chip," in *Conference on Design and Architectures for Signal and Image Processing*, Oct 2016, pp. 121–128.

[7] N. Jafarzadeh, M. Palesi, A. Khademzadeh, and A. Afzali-Kusha, "Data encoding techniques for reducing energy consumption in network-on-chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 3, pp. 675–685, 2014.

[8] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Energy efficient transceiver in wireless network on chip architectures," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*. Dresden, Germany: IEEE Computer Society, March 2016, pp. 1321–1326.

[9] "NanGate 45nm open cell library." [Online]. Available: http://www.nangate.com

[10] A. Yazdanbakhsh, D. Mahajan, H. Esmaeilzadeh, and P. Lotfi-Kamran, "Axbench: A multiplatform benchmark suite for approximate computing," *IEEE Design Test*, vol. 34, no. 2, pp. 60–68, April 2017.