

# Coupling Routing Algorithm and Data Encoding for Low Power Networks on Chip

<sup>1</sup>Maurizio Palesi, <sup>2</sup>Davide Patti, <sup>2</sup>Giuseppe Ascia, <sup>2</sup>Daniela Panno and <sup>2</sup>Vincenzo Catania

<sup>1</sup>Department of Computer Science Engineering, KORE University, Italy

<sup>2</sup>Department of Computer Science Engineering, DIEEI, University of Catania, Italy

## Article history

Received: 24-04-2014

Revised: 03-09-2014

Accepted: 22-04-2015

Corresponding Author:

Maurizio Palesi

Department of Computer

Science Engineering, KORE

University, Italy

Email: maurizio.palesi@unikore.it

**Abstract:** The routing algorithm used in a Network-on-Chip (NoC) has a strong impact on both the functional and non functional indices of the overall system. Traditionally, routing algorithms have been designed considering performance and cost as the main objectives. In this study we focus on two important non functional metrics, namely, power dissipation and energy consumption. We propose a selection policy that can be coupled with any multi-path routing function and whose primary goal is reducing power dissipation. As technology shrinks, the power dissipated by the network links represents an ever more significant fraction of the total power budget. Based on this, the proposed selection policy tries to reduce link power dissipation by selecting the output port of the router which minimises the switching activity of the output link. A set of experiments carried out on both synthetic and real traffic scenarios is presented. When the proposed selection policy is used in conjunction with a data encoding technique, on average, 31% of energy reduction and 37% of power saving is observed. An architectural implementation of the selection policy is also presented and its impact on cost (silicon area) and power dissipation of the baseline router is discussed.

**Keywords:** Network on Chip, Routing Algorithm, Selection Policy, Router Design, Performance Analysis, Power Analysis, Data Encoding

## Introduction

As the number of cores integrated into a single silicon chip increases, the role played by the communication system becomes more and more relevant. In fact, traditional bus based implementations does not scale as the complexity of the system increases. The Network-on-Chip (NoC) design paradigm represents a first answer to deal with scalability problems which characterizes the many-cores era. As technology shrinks, the impact on the overall performance, cost, power and reliability figures due to the communication subsystem becomes as important as and in some cases more important than, that due to the computation sub-system. For instance, considering the power metric, in the Intel's 80-tiles TeraFLOPS processor (Vangal *et al.*, 2008) the communication power accounts for about 30% of the total power budget. In the Massachusetts Institute of Technology RAW chip (Taylor *et al.*, 2002) the NoC is responsible for the 40% of the system power. In the Æthereal NoC the largest percentage of power dissipation (54%) is due to the NoC clock, followed by

the NoC links (18%) (Steenhof *et al.*, 2006). Ejlali *et al.* (2010), it has been shown that on-chip interconnects account for a significant fraction (up to 50%) of the total on-chip energy consumption. Based on this, several techniques have been proposed in literature aimed at reducing the power dissipated by the main NoC elements, namely, the routers, the links and the network interfaces. In this study, we focus on power dissipated by the network links as they contribute to a significant fraction of the total power dissipation. In addition, their importance is expected to increase as soon as technology shrinks.

Several factors determine the overall performance, cost, power, reliability and other functional metrics and physical attributes of a NoC. The routing algorithm, used to deliver the packets to their destinations, is one of the main protagonist in a NoC (Duato *et al.*, 2002).

A routing algorithm is composed by two main components, namely, the *routing function* and the *selection function* (Fig.1). The routing function computes the set of admissible output ports towards which the

packet can be forwarded to reach its destination. Then, the selection function is used to select one output port from the set of admissible output ports returned by the routing function. Of course, in a router implementing a deterministic routing algorithm, the selection block is not present since the routing function returns only a single output port. In a router implementing an oblivious routing algorithm the selection block takes its decision based solely on the information provided by the header flit; whereas, network status information (e.g., link utilization, buffer occupation) are exploited by the selection function of a router implementing an adaptive routing algorithm.

The importance of the routing algorithm is expected to increase as the network size (measured as the number

of communicating nodes) increases. For instance, Fig. 2 shows the percent performance improvement (measured as reduction in average delay) when a deterministic XY routing is replaced with an adaptive routing algorithm (Palesi *et al.*, 2009) for different network sizes. It also shows the contribute of the selection function by comparing two different selection strategies, namely, Buffer Level (BL) (Hu and Marculescu, 2004) and Neighbor-on-Path (NOP) (Ascia *et al.*, 2008). As can be observed, passing from 2x2 to 16x16 meshes, the performance improvement varies from 8 to 42%. It should be pointed out that the above results should be considered qualitative as they strongly depend on the actual network configuration and the considered traffic scenario.

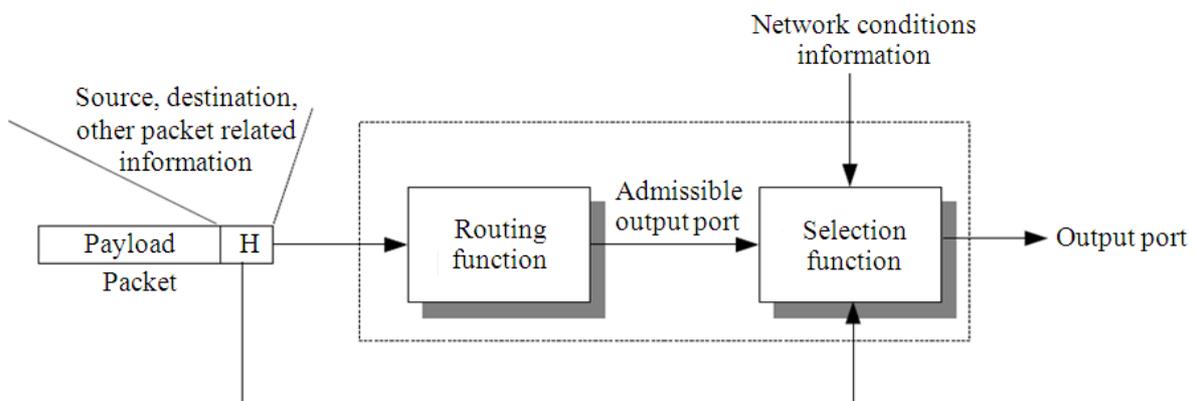


Fig. 1. Routing and selection blocks of a router

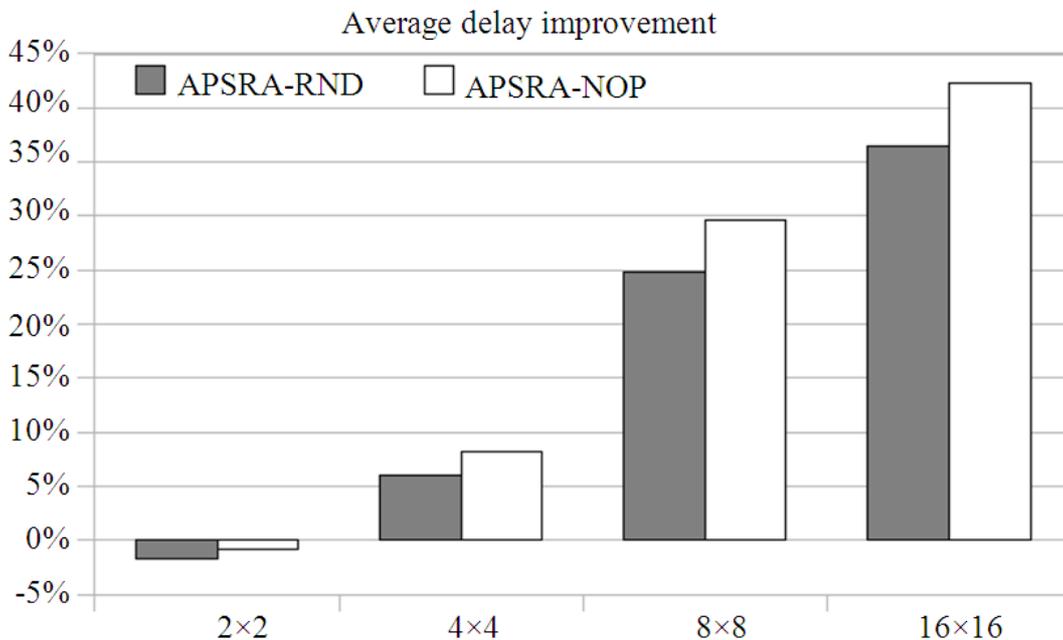


Fig. 2. Impact of the routing function and the selection function on the average delay for different NoC sizes

This study focuses on the selection function as it strongly determines the effectiveness of a routing algorithm (Feng and Shin, 1997). Differently from the previous works in which the selection function is designed with the aim of optimizing performance metrics (Hu and Marculescu, 2004; Ascia *et al.*, 2008), our proposal is mainly oriented to reducing the power dissipated by the network links. In fact, as technology shrinks, the power dissipated by the links is as significant as (or more significant than) that dissipated by routers and network interfaces (Srinivasan and Chatha, 2006; Palma *et al.*, 2007; Hoskote *et al.*, 2007; Carloni *et al.*, 2008). Links dissipate power due to the switching activity (both self and coupling) induced by subsequent data patterns traversing the link (Jantsch *et al.*, 2005). The basic idea behind the proposed selection function is selecting the output port through which it is convenient to forward the incoming packet in such a way that the power dissipated by the output link is minimized.

The proposed selection function is assessed on a set of traffic scenarios both synthetic and extracted from real applications. The analysis is carried out comparing with representative selection policies and considering performance, power dissipation and energy consumption as evaluation metrics. In addition, an architectural implementation of the proposed selection function is presented and the implications on the router architectures in terms of the area and power overhead are discussed.

The rest of the study is organized as follows. The proposed selection policy is presented in section III. In section IV the hardware implications on the architecture of the router are analyzed. The assessment of the proposed selection policy on both synthetic traffic and real traffic scenarios is presented in Section V. Finally, in section VI, we draw our conclusions.

## Related Work

There is a great deal of work in literature about the definition of routing algorithms for NoC architectures. The majority of them have been conceived with the primary goal of improving performance and, sometimes, reducing implementation cost. A methodology to design Application Specific Routing Algorithms for NoCs (APSRA) (Palesi *et al.*, 2009) has been proposed with the aim of maximizing the degree of adaptivity of the routing function as it is strictly related to the performance metrics of the communication system. In fact, APSRA does not take into account cost and power issues as the routing function needs to be implemented using routing tables which are usually an expensive resource both in terms of silicon area and power dissipation. To address power and cost issues, a technique to compress routing tables and the architecture of the hardware block for managing compressed routing

tables has been presented (Mejia *et al.*, 2009). The possibility of avoiding the use of expensive routing tables in favour of small logic blocks in every switch to implement a routing function (even for irregular network topologies) is presented in (Cano *et al.*, 2011). In the above works, power optimization is not the main goal. In fact, it represents a secondary positive effect due to the reduction of the area needed to implement the routing algorithm. Other proposals focus specifically on the design of routing algorithms in which power dissipation is the primary metric. Several routing algorithms have been proposed to deal with thermal issues related to power dissipation. A thermal-aware routing algorithm aimed at ensuring both thermal safety and less performance impact is proposed in (Chao *et al.*, 2010). The knowledge of traffic information is used to design application specific routing algorithms to reduce the hot-spot temperature for NoCs (Qian and Tsui, 2011). Traditionally, a minimum hop count routing policy is employed for electronic NoCs, as it minimizes both power consumption and latency. However, due to the special architecture of current optical NoC routers, such a minimum-hop path may not be energy-wise optimal. Based on this, the use of effective routing algorithms specifically defined to address energy related issues is explored in the context of optical NoCs (Liu and Yang, 2010). Power related issues are particularly important in 3D architectures. The use of routing algorithms for managing vertical communication with the goal of minimizing power dissipation is presented in (Rahmani *et al.*, 2011). When a multipath routing function is used, the goal of the selection function is usually that of selecting the output port which allows the packet to reach its destination by minimizing the end-to-end delay. The authors proposed a different way to design the selection function in which the primary objective is minimizing power dissipation while performance is in seen a secondary objective (Salemi *et al.*, 2011). This study presents an evolution of the selection strategy presented in (Salemi *et al.*, 2011). Although the basic idea is the same, in this study we couple a power-aware selection function with a data encoding technique aimed at reducing both energy consumption and power dissipation. The study also presents a possible implementation of the selection logic analysing its impact on area and power of the router.

## Power-Aware Selection Policy

### *The Idea in a Sketch*

The idea behind the proposed selection policy can be introduced as follows. Let us consider a router which has to choose the output port through which forwarding the incoming packet. The routing function will provide the set of admissible output ports through which the packet

can be forwarded. The proposed selection function, that will be presented in more details in the following, estimates the power dissipation due to the transmission of the incoming packet through each of the admissible output ports. Then, it selects the output port which minimizes the power metric.

### The Proposed Selection Policy

The pseudo-code of the algorithm implementing the proposed selection function is shown in Fig. 3. It gets as inputs the set of output ports returned by the routing function (*oport*s) and the header flit that has to be forwarded (*hflit*). It returns the selected output port. The

algorithm starts checking the size of the set of the admissible output ports returned by the routing function. If *oport*s contains a single output port, such output port is returned. If either all the output ports belonging to *oport*s are reserved or none of them is reserved, the output port which determines the minimum link power dissipation is selected (function SelectionMinPower). The information about reservation status of the output ports is obtained from the wormhole reservation table in the router (Duato *et al.*, 2002). In the remaining of the cases, the output port connected to the input port of the router which has the minimum buffer occupancy is selected (function SelectionMinBuffer).

```

1 OutputPort SelectionFunction(set<OutputPort> oports,
2                               TFlit hflit)
3 {
4   if (oports.size() = 1)
5     oport ← oports[0];
6   else if (AllReserved(oport) || NoneReserved(oport))
7     oport ← SelectionMinPower(oport, hflit);
8   else
9     oport ← SelectionMinBuffer(oport);
10
11  return oport;
12 }
    
```

Fig. 3. Pseudo-code of the algorithm implementing the proposed selection function

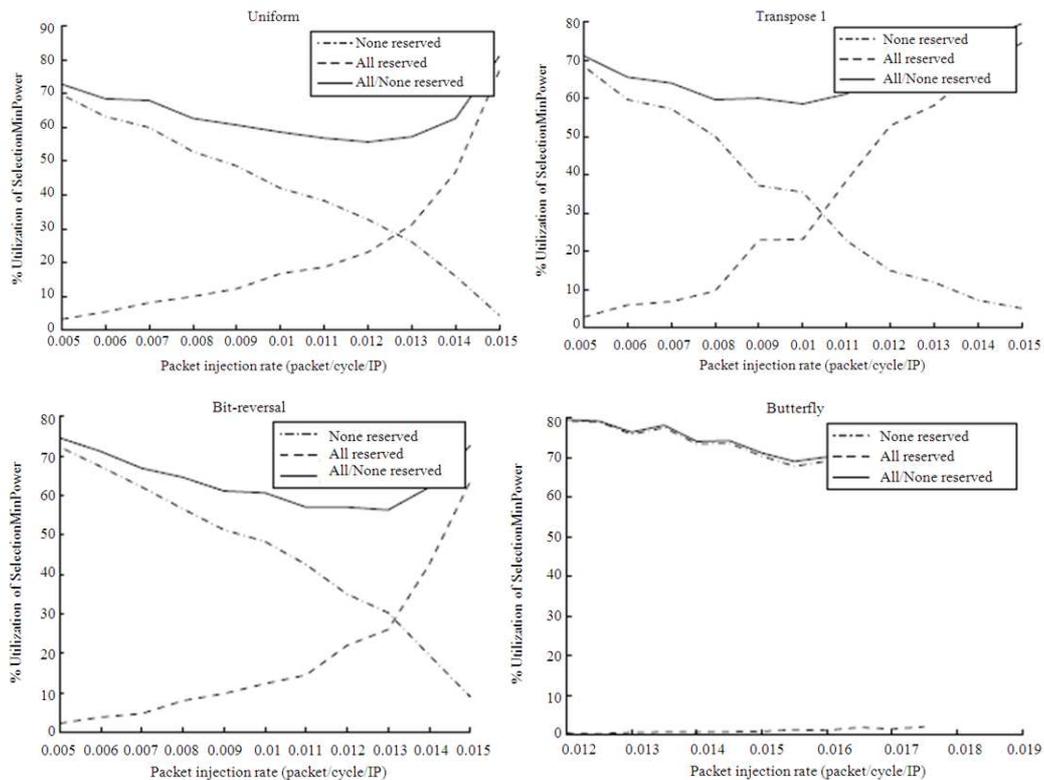


Fig. 4. Percent utilization rate of the SelectionMinPower selection function (solid line) for different packet injection rates under several traffic scenarios

As can be observed, the selection policy gives priority to performance optimization by means of the SelectionMinBuffer function. In fact, the SelectionMinPower function is used when either all the admissible output ports are available or they are all reserved. To assess how often the SelectionMinPower function is invoked in practical cases (i.e., condition in line 6 of Fig. 3 is met), the following experiment has been carried out. For different packet injection rates and several traffic scenarios, the percent utilization rate of the SelectionMinPower selection function is monitored as shown in Fig. 4. Precisely, the figure shows the percentage of the times that the single conditions in line 6 of Fig. 3 (data series *None reserved* and *All reserved*) are met. The sum of these components (data serie *All/None reserved*) represents the percentage of times that the SelectionMinPower function is actually used. As can be observed, in practical cases, the proposed selection function is used for more than 70% of the time on average. Based on this, we expect that the use of a power aware selection policy is justified and it would bring to significant power savings as it will be shown in the experiments section.

In the following subsections, the two main components which form the selection function, namely, the SelectionMinPower and the SelectionMinBuffer functions are described. The deadlock free property of the routing algorithm is also discussed.

### The SelectionMinPower Function

The SelectionMinPower function gets two inputs: The set of admissible output ports returned by the routing function (*oport*s) and the header flit to be forwarded (*h f lit*). It returns the output port belonging to *oport*s such that the power dissipated by the link traversed by *h f lit* is minimized. The dynamic power dissipated by the link is:

$$P_l = [T_{0 \rightarrow 1}(C_s + C_l) + T_c C_c] V_{dd}^2 F_{ck}, \quad (1)$$

Where:

- $V_{dd}$  = The supply voltage
- $F_{ck}$  = The clock frequency
- $C_s$  = The self capacitance
- $C_l$  = The load capacitance
- $C_c$  = The coupling capacitance

$T_{0 \rightarrow 1}$  and  $T_c$  are the average number of effective transitions per cycle for  $C_s$  and  $C_c$  respectively.  $T_{0 \rightarrow 1}$  counts the number of 0→1 transitions in the link in two consecutive transmissions.  $T_c$  counts the correlated switching between physically adjacent lines. There are

four types of coupling transitions (Kim *et al.*, 2000). A Type I transition occurs when one of the lines switches while the other stays unchanged. In a Type II transition one line switches from low to high and the other from high to low. A Type III transition occurs when both lines switch simultaneously. Finally, in a Type IV transition both lines do not switch. So, the coupling transition activity  $T_c$  is a weighted sum of the different type of coupling transition contributions as follows:

$$T_c = k_1 T_1 + k_2 T_2 + k_3 T_3 + k_4 T_4 \quad (2)$$

where, the  $T_i$ ,  $i = 1, 2, 3, 4$ , are the average number of transition type  $i$  and  $k_i$  are weights.

According to (Kim *et al.*, 2000) we assume  $k_1 = 1$ ,  $k_2 = 2$ ,  $k_3 = k_4 = 0$  and  $C_c/C_s = 4$ . That is,  $k_1$  is assumed as reference for other types of transition. The effective capacitance in Type II transition is usually twice that of a Type I transition. In Type III transition, as both signal switch simultaneously,  $C_c$  is not charged (here we assume that there are no misalignment between the two transitions). Finally, in Type IV transition there is no dynamic charge distribution over  $C_c$ . Overall, Equation 1 can be simplified as follows:

$$P_l = [T_{0 \rightarrow 1} + 4(T_1 + 2T_2)] C_s V_{dd}^2 F_{ck} \quad (3)$$

Let us indicate with  $T_{0 \rightarrow 1}^i$  the number of 0→1 transitions in the link connected to the output port  $op_i \in oports$  due to the transmission of *h f lit*. Similarly, let us indicate with  $T_1^i$  and  $T_2^i$  the number of Type I transitions and Type II transitions in the link connected to the output port  $op_i \in oports$  due to the transmission of *h f lit*. So, the goal of the selection function is to select the output port belonging to *oport*s such that  $P_l$  is minimized, that is:

$$\min_{i=0, \dots, |oport| - 1} [T_{0 \rightarrow 1}^i + 4(T_1^i + 2T_2^i)] \quad (4)$$

To simplify the computation of (4), the following approximated problem is considered:

$$\min_{i=0, \dots, |oport| - 1} (T_1^i + 2T_2^i) \quad (5)$$

In which the term  $T_{0 \rightarrow 1}^i$  is neglected in favor of the term  $T_1^i + 2T_2^i$  which is weighted by 4. Let us consider

mesh topologies and minimal routing. The routing function will return at most two admissible output ports; thus, an approximated solution of problem (5) can be obtained using the algorithm shown in Fig. 5. If the number of Type II transitions when h f lit is forwarded through the port *oports* [0] ( $T_2^0$ ) is greater than the number of Type II transitions when h f lit is forwarded through the port *oports* [1] ( $T_2^1$ ), then output port *oports* [1] is selected. Otherwise, if  $T_2^0$  is less than  $T_2^1$ , then *oports* [0] is selected. If  $T_2^0 = T_2^1$ , Type I transitions are compared. If  $T_1^0$  is greater than  $T_1^1$  then *oports*[1] is selected otherwise *oports* [0] is selected.

### The SelectionMinBuffer Function

The SelectionMinBuffer function gets as input the set of admissible output ports returned by the routing function (*oports*) and returns the output port connected to the input port of the downstream router with the minimum buffer occupancy. Let us indicate with  $b_o^i$  the occupancy of the buffer of the input port of the router connected to the port *oports* [i] of the current router. If we consider mesh topologies and minimal routing, the algorithm implemented by the SelectionMinBuffer function is shown in Fig. 6.

If  $b_o^1$  is greater than  $b_o^0$ , then *oport* [0] is returned. Otherwise, if  $b_o^1$  is less than  $b_o^0$ , then *oport* [1] is returned. Finally, if  $b_o^1$  is equal to  $b_o^0$ , then a random selection is performed.

### Deadlock Freedom

Deadlocks may appear in interconnection networks such as NoCs and may lead to performance degradation or even system failure (Benini and Micheli, 2002). One of the most common techniques for deadlock avoidance is based on the turn model (Chiu, 2000) which prohibits the routing algorithm from making certain turns in the network in such a way that the Channel Dependency Graph (CDG) is acyclic. The proposed selection function does not affect the deadlock free properties of the routing algorithm. A selection function selects one of the multiple allowed turns provided by the routing function. For this reason, the allowed routing paths remain unchanged whatever the selection function is. Based on this, the proposed selection function is agnostic as respect to the underlying routing function (i.e., it can be coupled with any routing function) and does not affect the CDG and therefore the deadlock free property of the routing algorithm.

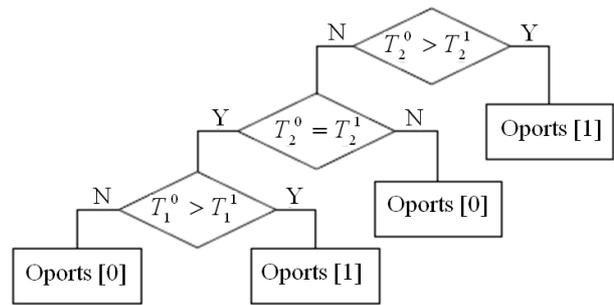


Fig. 5. Flow chart of the SelectionMinPower function

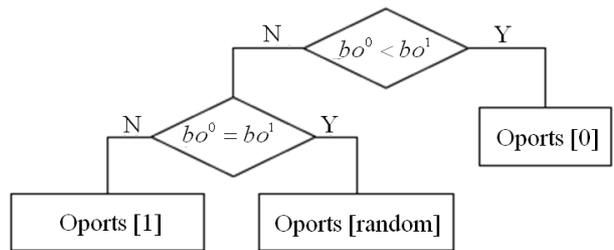


Fig. 6. Flow chart of the SelectionMinBuffer function

## Implications on the Router Architecture

In this section we propose a possible hardware implementation of the proposed selection function and analyze the overhead in terms of silicon area and power dissipation over a baseline router implementing a deterministic XY routing.

### Hardware Implementation of the Algorithm

Figure 7 shows the block diagram of the router and its main internal blocks. For each direction the router has three main ports, namely,  $x\_in$ ,  $x\_out$  and  $x\_bo$ , where  $x$  can be n (north), e (east), w (west) and s (south). There is also a local port,  $l$ , connected to the IP core through the network interface.  $X\_in$  and  $x\_out$  represent the input and output ports from where the flit is received and through which the flit is transmitted, respectively.  $X\_bo$  receives the number of flits stored in the input buffer of the router connected to the  $x$  output port.

The internal structure of the router is shown in Fig. 7b. The selection function has five inputs: The set of admissible output ports returned by the routing function (*AOP*), the information about the occupancy of the buffers of the neighbors routers ( $bo$ ), the header flit to be forwarded (*h f lit*) and the previously transmitted flit (*pt f lit*). The output provides the selected output port (*out\_dir*).

The internal of the module implementing the selection function is shown in Fig. 7c. It simply propagates the output of either the SelectionMinBuffer module or the

SelectionMin- Power module based on the availability of the output ports in *AOP* (condition in line 6, Fig. 3).

The SelectionMinPower module, implementing the algorithm in Fig. 5, is shown in Fig. 7d. The block SelPTFlits returns the previously transmitted flits through the output ports in *AOP*, whereas the blocks T1 and T2 count the number of Type I and Type II transitions respectively if *hflit* is forwarded through the output ports in *AOP*. Finally, the SelOut module returns the selected output port based on the result of the comparisons.

Overall, the data flow can be summarized as follow. When a header flit of a packet reaches the router, the block Routing Function returns the set of admissible output ports where the packet can be forwarded. The block Selection Function selects one of these admissible output ports based on the reservation status of the channel (*whrt*), the previous transmitted flit (*pt flit*) and the occupancy of the input buffers connected to the output ports in *AOP*. If all the output ports in *AOP* are reserved or none of them is reserved, the output provided by the SelectionMinPower block is used to select the output port

through which forwarding the packets. Otherwise, the output port provided by the SelectionMinBuffer block is used. In the first case, the selected output port is that resulting from the condition tree shown in Fig. 5. In the second case, the selected output ports is determined by the condition tree in Fig. 6.

### Synthesis Results

The following routers have been designed in VHDL described at the RTL level, synthesized with Synopsys Design Compiler and mapped onto an UMC 65 nm technology library:

- XY: A router implementing a deterministic XY routing algorithm
- OE-BL: A router implementing Odd-Even (Chiu, 2000) routing function and buffer level selection function
- OE-PWR: A router implementing Odd-Even (Chiu, 2000) routing function and the proposed power-aware selection function

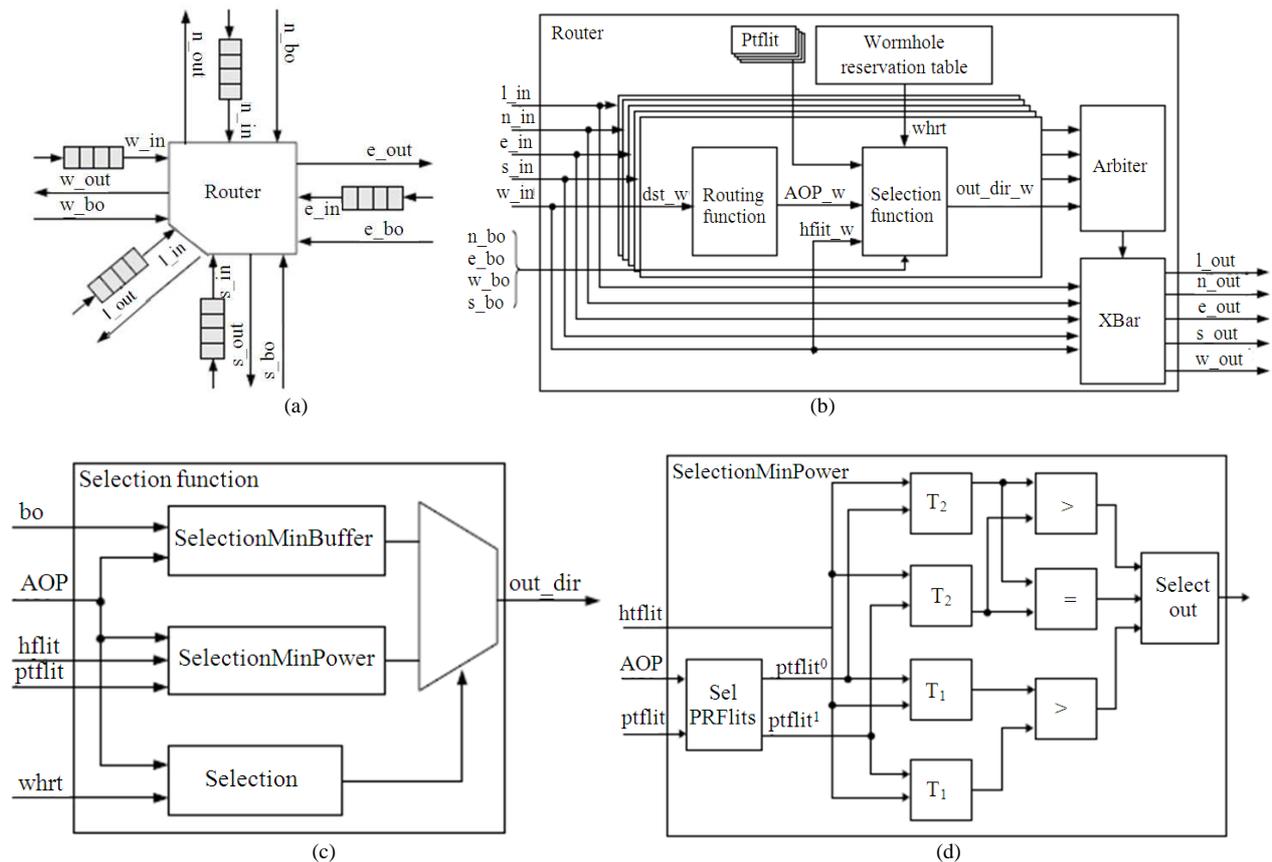


Fig. 7. Block diagram of the router architecture, Top level interface (a) Internal structure for the west input port (b) Internal structure of the selection function module (c) External structure of module implementing the SelectionMinPower function (d)

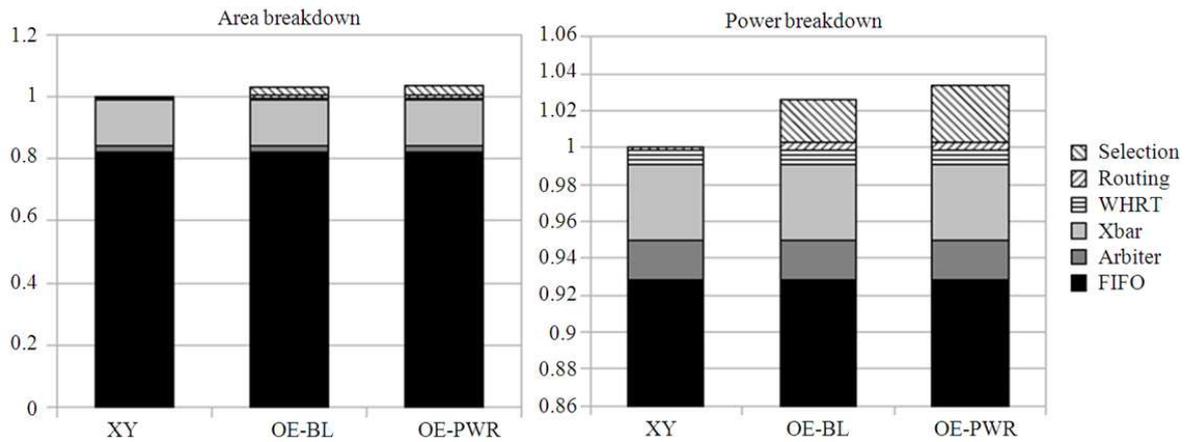


Fig. 8. Area and power breakdown (values normalized w.r.t. XY)

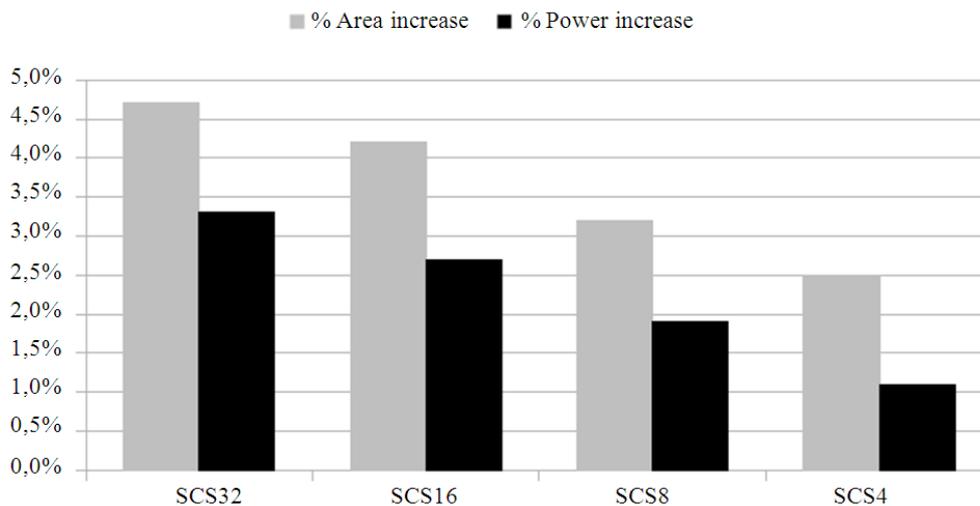


Fig. 9. Percentage impact on silicon area and power dissipation of the network interface due to the data encoding/decoding logic

All the above routers are clocked at 800 MHz and have input FIFO buffers of 4 flits with 32-bit flit.

Figure 8 shows the area and power breakdown for the three router implementations. The data are normalized with respect to XY. As can be observed, the FIFO buffers accounts for a significant fraction of the area and power dissipation. The contribute of the selection function of the OE-PWR router on both silicon area and power dissipation is about 3%. Overall, OE-PWR router is 3 and 1% larger and more power hungry than XY and OE-BL respectively. However, as it will be shown in the next section, this overhead is more than balanced by the power and energy saving on the network links.

Assuming a wormhole switching technique, the proposed scheme can be further improved if the data flits are encoded in such a way that they determine the minimum switching activity and minimum coupled switching activity when the packet traverses the links of

the routing path. The basic idea is encoding the transmitted flits at the Network Interface (NI) of the source node and decoding the incoming flits at the NI of the destination node. As the header flit is not encoded (i.e., only the data flits of the packet are encoded) the routers' logic remains unchanged. Adding data encoding feature to the proposed scheme results in an overhead of the NI.

Assuming a 32-bit flit size, the considered data encoding technique (Palesi *et al.*, 2011) can be applied to the entire flit (SCS32), separately to the two 16-bit partitions of the flit (SCS16), to the four 8-bit partitions of the flit (SCS8) and to the eight 4-bit partitions of the flit (SCS4).

Figure 9 shows the percentage impact on silicon area and power dissipation of the NI due to the data encoding/decoding logic. The baseline NI has minimum buffering and supports OCP 2 and AHB protocols. As can be observed area and power overhead range from 2.5

to 4.8% and from 1.1 to 3.2%, respectively. However, as it will be shown in the experiments section, the power overhead is counterbalanced by the energy saving that can be obtained using this scheme.

## Experiments

In this section we assess the proposed selection function on both synthetic and real traffic scenarios. The analysis is carried out using a cycle accurate NoC simulator based on Noxim (Fazzino *et al.*, 2005). The power estimation models implemented in Noxim have been updated with the power figures of the selection functions discussed in the previous section. We indicate with OE-PWR-DE  $n$  the case in which OE-PWR is used in conjunction with SCS  $n$  data encoding technique proposed in (Palesi *et al.*, 2011). The suffix  $n$  belongs to the set {4,8,16,32} and refers to the number of bits in which the link is partitioned as discussed in subsection IV-B. When OE-PWR-DE  $n$  is considered, the power contribute of the network interface, augmented with the SCS  $n$  data encoding/decoding logic, is taken into account to compute the overall power/energy figures.

The experiments are carried out on an 8×8 mesh-based NoC. We consider input FIFO buffers of 4 flits and packets of 8 flits injected at different Packet Injection Rates (PIR). Energy figures are computed running the simulation until 1 MB of traffic is *drained* by the network. A number of simulations is repeated for each  $pir$  value and energy values are averaged until the 95% confidence intervals are mostly within 2% of the means.

### Performance Analysis

We start comparing the different selection functions using the average delay as performance metric. Figure 10 shows the average delay under *uniform* and *transpose* traffic. As can be observed, OE-PWR performs as well as OE-BL although the buffer level selection is used for a fraction of the time (Fig. 4). As expected, as data encoding is used, performance degradation is observed. In particular, the impact on average delay is more evident as the number of partitions increases. In fact, as the number of partitions increases, the number of inversion bits to be transmitted increases as well. This causes an increase of the injected traffic with a consequence negative impact on performance. On the other hand, as already observed in Fig. 9, a higher number of partitions results in less power and area overhead in the network interface.

Due to space limitation, we do not report the detailed results for the other traffic scenarios. We limit the analysis for the other traffic scenarios by reporting the results for two performance indices, namely, saturation  $pir$  and average delay. Precisely, Fig. 11 shows a summary of the improvements in terms of the increase in saturation  $pir$  (A network is said to start saturating when increase in applied load does not result in linear increase in throughput (Pande *et al.*, 2005))

and decrease in average delay assuming XY as baseline. As can be noticed, the saturation  $pir$  increases by 27 and 20% on average and the average delay decreases by 47 and 40% on average with OE-BL and OE-PWR respectively. With OE-PWR-DE  $n$  there is not significant performance degradation as compared to OEPWR.

### Power Analysis

It has been shown how OE-PWR performs as well as OE-BL under different traffic conditions. However, the main goal of OE-PWR and its extension OE-PWR-DE  $n$ , is reducing power dissipation and energy consumption. Figure 12 shows the average energy per flit for different packet injection rates and under *uniform* and *transpose* traffic. The average energy per flit is computed by dividing the total energy consumption to drain 1 MB of traffic by the total number of received flits. As can be observed, the energy saving obtained with OE-PWR is less than 3% as compared to OE-BL. This small power improvement is due to the fact that the amount of power saved by OE-PWR as respect to OE-BL is just that related to the transmission of the header flit. In fact, the selection of the output port which minimises the power dissipation is based on the header flit and not on the remaining flits of the packet. Thus, as the packet size increases, the power saving of OE-PWR decreases as will be shown later. Important power savings can be observed when the proposed selection policy is coupled with data encoding. As can be observed in Fig. 12, OE-PWR-DE is about 17% more energy efficient than XY and OE-BL under *uniform* traffic. Under *transpose* traffic, OE-PWR-DE is about 20 and 18% more energy efficient than XY and OE-BL respectively. Overall, the power overhead due to the encoding and decoding logic in the network interfaces is completely absorbed by the power saved in the links of the network.

Figure 13 summarizes the energy saving that can be obtained using the proposed approach. Values are normalized by the energy consumption when XY is used. Energy values are captured for a  $pir$  value where none of the networks are saturated. On average, as compared to XY, OE-PWR-Ben allows to save from 19 to 22% of energy respectively.

The packet size impacts the power figures of the different algorithms discussed so far. Figure 14 shows the impact of packet size on power saving assuming XY as baseline and under *uniform* traffic.

As packet size increases, the power saving of OE-PWR decreases since the attempt for power reduction is made just for the header flit. On the other hand, when OEPWR-BE  $n$  is used, the power saving increases as packet size increases. This is due to the fact that the links of the routing path are traversed by a long “worm” of flits which have been encoded assuming that they are not interleaved by other flits belonging to other packets. As

we are assuming wormhole as switching technique, the above assumption is satisfied and link power dissipation is minimized. The only loss in power saving is when a link is traversed by the header flit of the packet as it is

not encoded and the frequency of this event is inversely proportional to the packet size. However, in this case, an attempt for power reduction is made by selecting the output port which minimizes the power consumption.

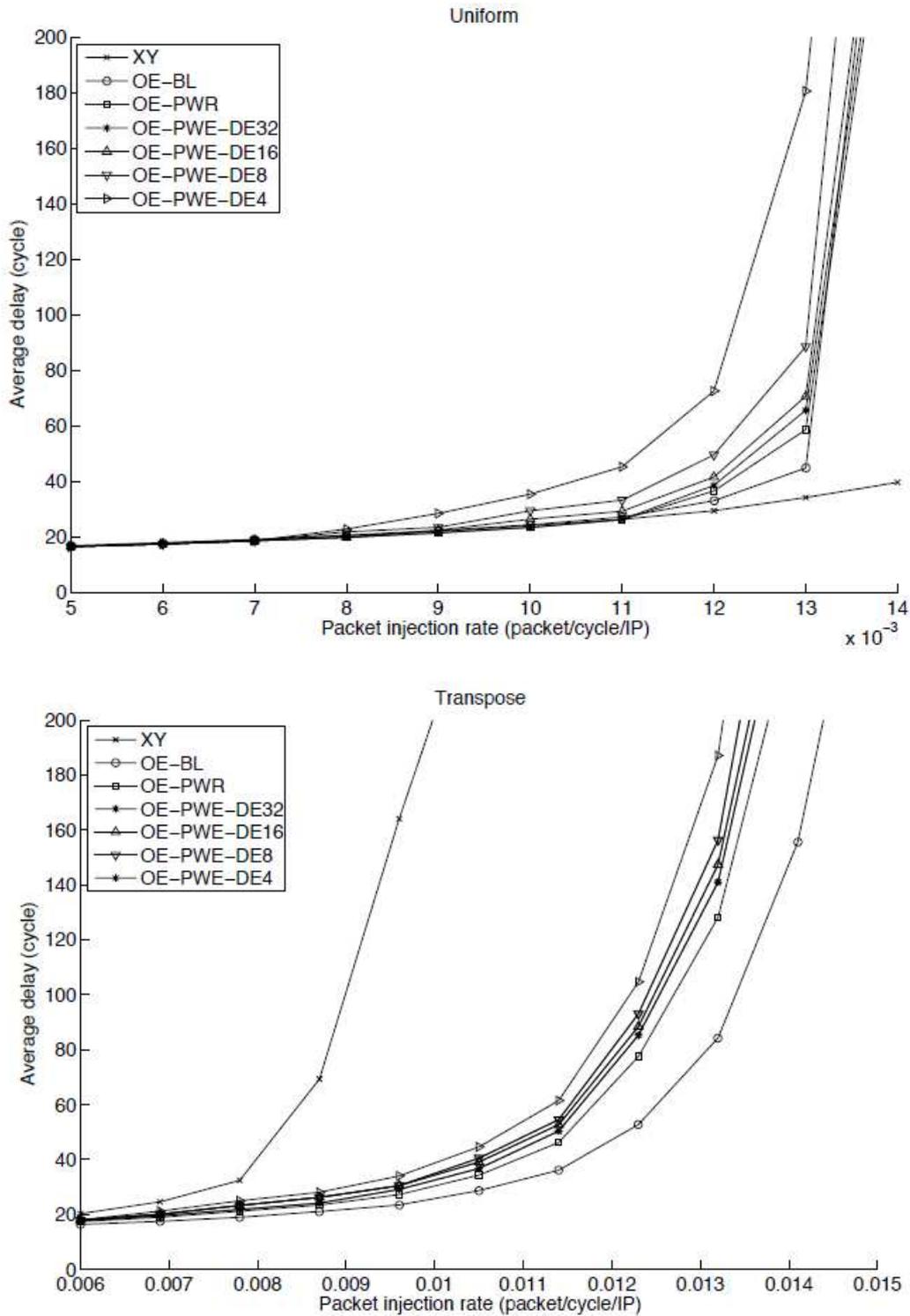


Fig. 10. Average delay for different packet injection rates under uniform and transpose traffic

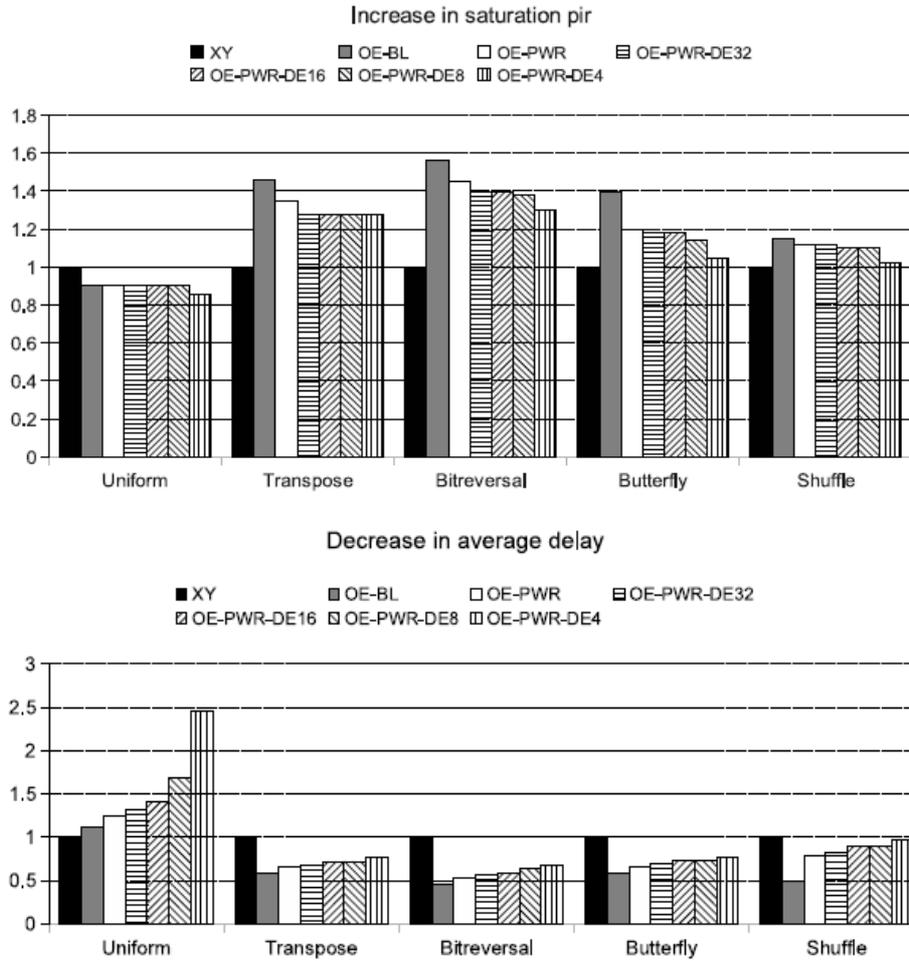
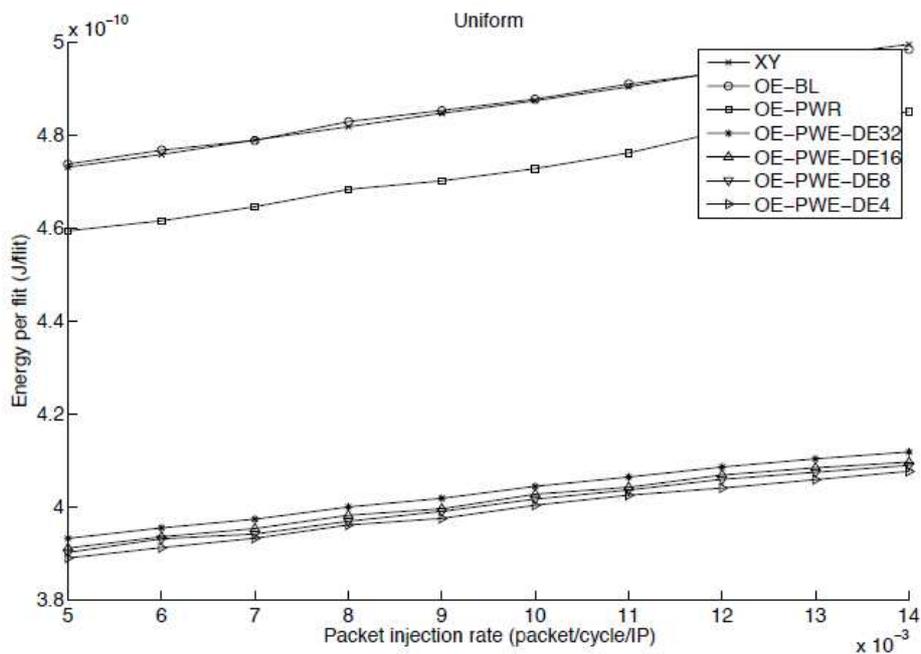


Fig. 11. Increase in saturation *pir* and decrease in average delay using XY as baseline



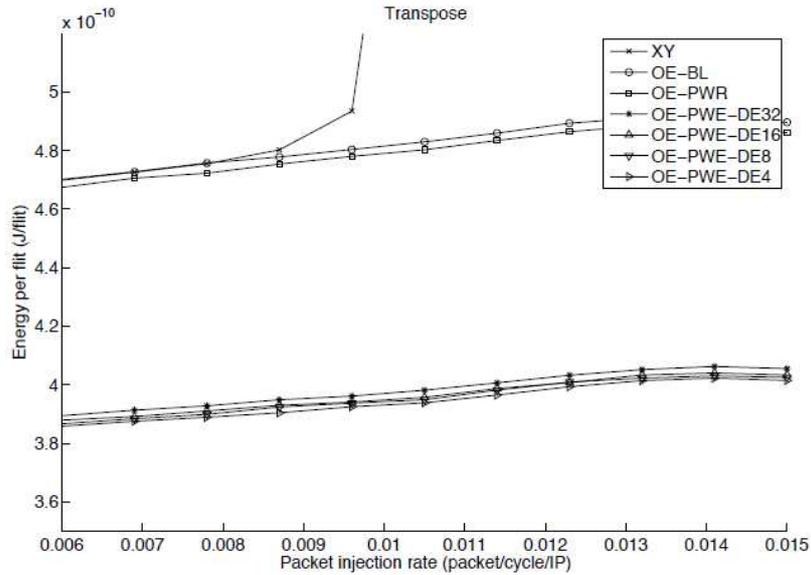


Fig. 12. Energy per flit under uniform and transpose traffic for different packet injection rates

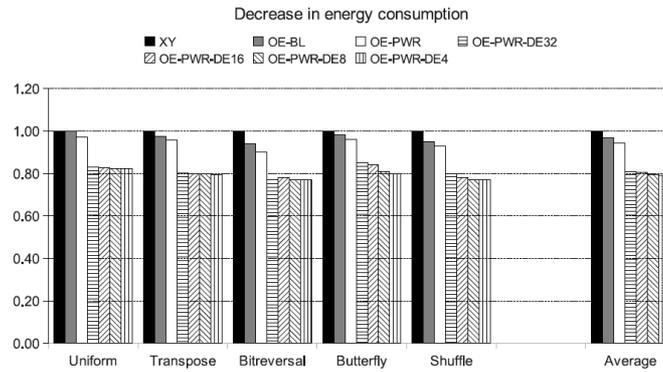


Fig. 13. Decrease in energy consumption using XY as baseline

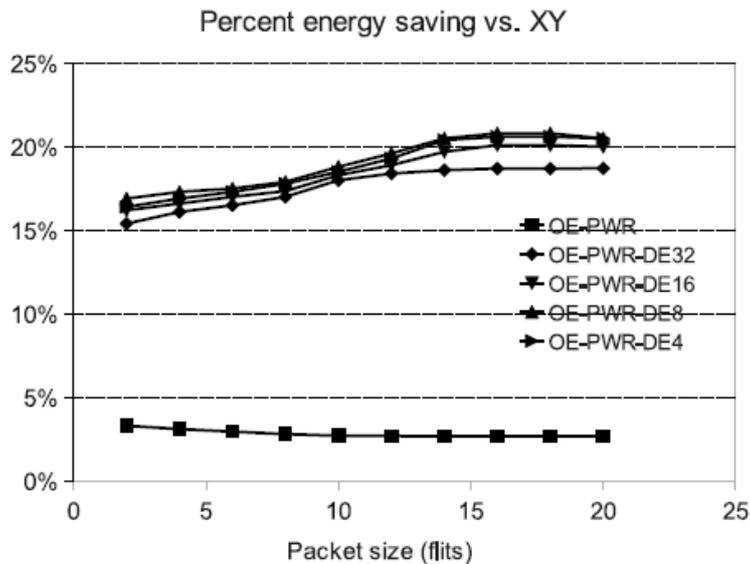


Fig. 14. Impact of packet size on power saving assuming XY as baseline and under uniform traffic

### Case Study

In this subsection we analyse the different algorithms on a real traffic scenario. In fact, the traffic is extracted from a real application running on a complex heterogeneous system shown in Fig. 15.

The system is composed by a generic MultiMedia System which includes a H.263 video encoder, a H.263 video decoder, a MP3 audio encoder and a MP3 audio decoder (Hu and Marculescu, 2005). A MIMO-OFDM receiver in which some of the IPs have been parallelized to multiple IPs (Yoon *et al.*, 2006). A Picture-In-Picture application (PIP) and a Multi-Window Display application (MWD) (Jaspers and De With, 1999; Van Der Tol and Jaspers, 2002). The set of applications have been mapped into an 8x8 mesh topology using the mapping technique proposed in (Tornero *et al.*, 2009). The analysis has been performed considering 32-bit links and routers working at a clock frequency of 800 MHz.

In this case study, both packet size and packet injection rate vary with communication flow. For instance,

the communication flows involved in MMS-Enc and MMS-Dec use a packet size tuned on the basis of a macroblock. Packet injection rate has been computed for each communication flow on the basis of the bandwidth requirements for each application as reported in (Hu and Marculescu, 2005; Yoon *et al.*, 2006; Jaspers and De With, 1999; Van Der Tol and Jaspers, 2002).

Table 1 reports the percentage reduction of total energy consumption, energy per flit and average power dissipation of OE-PWR and OE-PWR-DE<sub>n</sub> as respect to XY. The average power dissipation is computed dividing the total energy by the completion time. As can be observed, OE-PWR-DE<sub>n</sub> provides important power and energy savings as compared to the baseline implementation based on XY routing algorithm. In particular, the configuration which uses the encoding technique with links partitioned in 8-bit sublinks provides the best results in terms of both power dissipation and energy consumption.

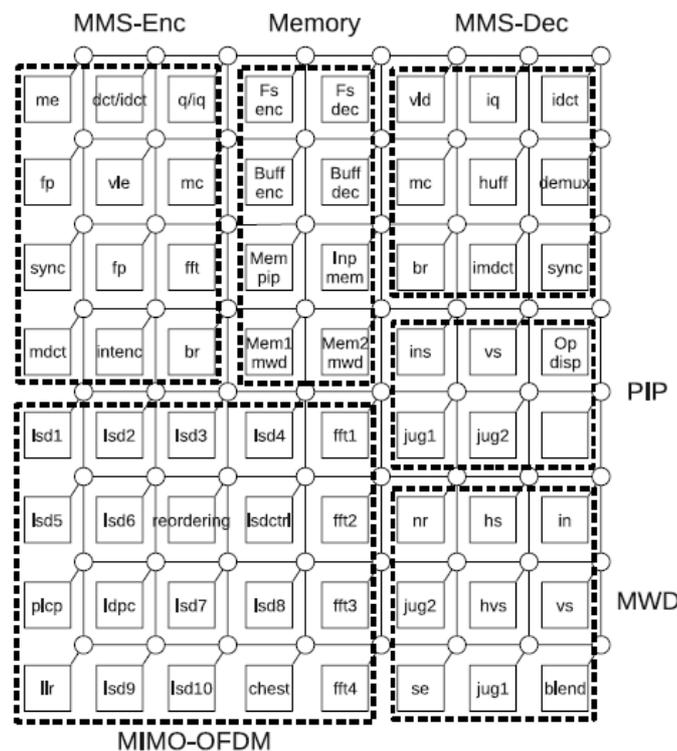


Fig.15. Heterogeneous system composed by a multimedia sub-system, a MIMO-OFDM receiver, a PIP and a MWD module

Table 1. Power and energy saving over XY

	OE-PWR	OE-PWR-DE <sub>n</sub>			
		n = 32 (%)	n = 16 (%)	n = 8 (%)	n = 4 (%)
Total energy	3	27	29	31	28
Energy per flit	5	33	35	36	32
Average power	3	30	34	37	30

## Conclusion

The selection function has a strong impact on performance and power figures of a routing algorithm (Duato *et al.*, 2002; Feng and Shin, 1997). Previous work in the context of the definition of new selection schemes are mainly focused on performance optimization. However, power dissipation and energy consumption represent important design objectives to be optimized that cannot be neglected. In this study we have proposed a new selection function which tries to conjugate both the aspects of improving performance and reducing energy consumption. The basic idea is selecting the routing path which minimizes the power dissipated by the network links. The proposed selection function is general and can be coupled with any multipath routing function. The analysis has been carried out on both synthetic and real traffic scenarios. It has been observed that, using the proposed selection function, up to 31 and 37% of energy consumption and power dissipation can be saved, respectively. Hardware overhead to support the proposed selection function has been also studied. It has been shown that, a router implementing the proposed selection function is 3 and 1% larger and more power hungry than a router implementing XY and OE-BL respectively. However, as shown in the experiments section, the power saving on links counterbalances these overheads.

## Funding Information

The authors have no support or funding to report.

## Author's Contributions

All authors equally contributed in this work.

## Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

## References

- Ascia, G., V. Catania, M. Palesi and D. Patti, 2008. Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip. *IEEE Trans. Comput.*, 57: 809-820. DOI: 10.1109/TC.2008.38
- Benini, L. and G.D. Micheli, 2002. Networks on chips: A new SoC paradigm. *IEEE Comput.*, 35: 70-78. DOI: 10.1109/2.976921
- Cano, J., J. Flich, J. Duato, M. Coppola and R. Locatelli, 2011. Efficient routing implementation in complex systems-on-chip. Proceedings of the 5th IEEE/ACM International Symposium on Networks on Chip, May 1-4, IEEE Xplore Press, Pittsburgh, PA, pp: 1-8. DOI: 10.1145/1999946.1999948
- Carloni, L., A.B. Kahng, S. Muddu, A. Pinto and K. Samadi *et al.*, 2008. Interconnect modeling for improved system-level design optimization. Proceedings of the Asia and South Pacific Design Automation Conference, Jan. 21-24, Seoul, South Korea, pp: 258-264.
- Chao, C.H., K.Y. Jheng, H.Y. Wang, J.C. Wu and A.Y. Wu, 2010. Traffic-and thermal-aware run-time thermal management scheme for 3D NoC systems. Proceeding of the 4th ACM/IEEE International Symposium on Networks-on-Chip, May 3-6, IEEE Xplore Press, Grenoble, pp: 223-230. DOI: 10.1109/NOCS.2010.32
- Chiu, G.M., 2000. The odd-even turn model for adaptive routing. *IEEE Trans. Parallel Distrib. Syst.*, 11: 729-738. DOI: 10.1109/71.877831
- Duato, J., S. Yalamanchili and L. Ni, 2002. *Interconnection Networks: An Engineering Approach*. 1st Edn., Morgan Kaufmann, San Francisco, CA., ISBN-10: 1558608524, pp: 600.
- Ejlali, A., B.M. Al-Hashimi, P. Rosinger, S.G. Miremadi and L. Benini, 2010. Performability/energy tradeoff in error-control schemes for on-chip networks. *IEEE Trans. Comput. Aided Design Integrated Circu. Syst.*, 18: 1-14. DOI: 10.1109/TVLSI.2008.2000994
- Fazzino, F., M. Palesi and D. Patti, 2005. Noxim: The network-on-chip simulator.
- Feng, W.C. and K.G. Shin, 1997. Impact of selection functions on routing algorithm performance in multicomputer networks. Proceedings of the 11th International Conference on Supercomputing, Jul. 07-11, New York, USA, pp: 132-139. DOI: 10.1145/263580.263616
- Hoskote, Y., S. Vangal, A. Singh, N. Borkar and S. Borkar, 2007. A 5-GHz mesh interconnect for a teraflops processor. *IEEE Micro*, 27: 51-61. DOI: 10.1109/MM.2007.77
- Hu, J. and R. Marculescu, 2004. DyAD: Smart routing for networks-on-chip. Proceedings of the ACM/IEEE Design Automation Conference, Jun. 7-11, San Diego, CA, USA, pp: 260-263. DOI: 10.1145/996566.996638
- Hu, J. and R. Marculescu, 2005. Energy-and performance-aware mapping for regular NoC architectures. *IEEE Trans. Comput. Aided Design Integrated Circu. Syst.*, 24: 551-562. DOI: 10.1109/TCAD.2005.844106

- Jantsch, A., R. Lauter and A. Vitkowski, 2005. Power analysis of link level and end-to-end data protection in networks on chip. *IEEE Int. Symp. Circui. Syst.*, 2: 1770-1773. DOI: 10.1109/ISCAS.2005.1464951
- Jaspers, E.G.T. and P.H.N. De With, 1999. Chip-set for video display of multimedia information. *IEEE Trans. Consumer Electron.*, 45: 706-715. DOI: 10.1109/30.793577
- Kim, K.W., K.H. Baek, N. Shanbhag, C.L. Liu and S.M. Kang, 2000. Coupling-driven signal encoding scheme for low-power interface design. *Proceedings of the IEEE/ACM International Conference on Computer-aided Design*, Nov. 05-09, IEEE Xplore Press, San Jose, CA, USA, pp: 318-321.
- Liu, L. and Y. Yang, 2010. Energy-aware routing in hybrid optical network-on-chip for future multi-processor system-on-chip. *Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, Oct. 25-26, San Diego, CA, USA, pp: 1-9. DOI: 10.1145/1872007.1872029
- Mejia, A., M. Palesi, J. Flich, S. Kumar and P. Lopez *et al.*, 2009. Region-based routing: A mechanism to support efficient routing algorithms in NoCs. *IEEE Trans. Very Large Scale Integration Syst.*, 17: 356-369. DOI: 10.1109/TVLSI.2008.2012010
- Palesi, M., G. Ascia, F. Fazzino and V. Catania, 2011. Data encoding schemes in networks on chip. *IEEE Trans. Comput. Aided Design Integrated Circu. Syst.*, 30: 774-786. DOI: 10.1109/TCAD.2010.2098590
- Palesi, M., R. Holsmark, S. Kumar and V. Catania, 2009. Application specific routing algorithms for networks on chip. *IEEE Trans. Paralle. Distrib. Syst.*, 20: 316-330. DOI: 10.1109/TPDS.2008.106
- Palma, J.C.S., L.S. Indrusiak, F.G. Moraes, A.G. Ortiz and M. Glesner *et al.*, 2007. Inserting data encoding techniques into NoC-based systems. *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, Mar. 9-11, IEEE Xplore Press, Porto Alegre, pp: 299-304. DOI: 10.1109/ISVLSI.2007.58
- Pande, P.P., C. Grecu, M. Jones, A. Ivanov and R. Saleh, 2005. Performance evaluation and design trade-offs for network-on-chip interconnect architectures. *IEEE Trans. Comput.*, 54: 1025-1040. DOI: 10.1109/TC.2005.134
- Qian, Z. and C.Y. Tsui, 2011. A thermal-aware application specific routing algorithm for network-on-chip design. *Proceedings of the Asia and South Pacific Design Automation Conference*, Jan. 25-28, IEEE Xplore Press, Yokohama, pp: 449-454. DOI: 10.1109/ASPAC.2011.5722232
- Rahmani, A.M., K. Latif, K.R. Vaddina, P. Liljeberg and J. Plosila *et al.*, 2011. Power-efficient inter-layer communication architectures for 3D NoC. *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, Jul. 4-6, IEEE Xplore Press, Chennai, pp: 355-356. DOI: 10.1109/ISVLSI.2011.12
- Salemi, D., M. Palesi and V. Catania, 2011. Power-aware selection policy for networks on chip. *Proceeding of the 6th International Conference on Design and Technology of Integrated Systems in Nanoscale Era*, Apr. 6-8, IEEE Xplore Press, Athens, pp: 1-4. DOI: 10.1109/DTIS.2011.5941422
- Srinivasan, K. and K.S. Chatha, 2006. Layout aware design of mesh based NoC architectures. *Proceedings of the International Conference on Hardware-Software Co-design and System Synthesis*, Oct. 22-27 IEEE Xplore Press, Seoul, pp: 136-141. DOI: 10.1145/1176254.1176288
- Steenhof, F., H. Duque, B. Nilsson, K. Goossens and R.P. Llopis, 2006. Networks on chips for high-end consumer-electronics TV system architectures. *Proceedings of the Conference on Design, Automation and Test in Europe*, Mar. 6-10, IEEE Xplore Press, Munich, pp: 1-6. DOI: 10.1109/DATE.2006.243840
- Taylor, M.B., J. Kim, J. Miller, D. Wentzlaff and F. Ghodrati *et al.*, 2002. The raw microprocessor: A computational fabric for software circuits and general-purpose programs. *IEEE Micro*, 22: 25-35. DOI: 10.1109/MM.2002.997877
- Tornero, R., V. Sterrantino, M. Palesi and J.M. Orduna, 2009. A multi-objective strategy for concurrent mapping and routing in networks on chip. *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing*, May 23-29, IEEE Xplore Press, Rome, pp: 1-8. DOI: 10.1109/IPDPS.2009.5161128
- Van Der Tol, E.B. and E.G. Jaspers, 2002. Mapping of MPEG-4 decoding on a flexible architecture platform. *Media Processors*. DOI: 10.1117/12.451067
- Vangal, S.R., J. Howard, G. Ruhl, S. Dighe and H. Wilson *et al.*, 2008. An 80-tile sub-100-W TeraFLOPS processor in 65-nm CMOS. *IEEE J. Solid-State Circuits*, 43: 29-41. DOI: 10.1109/JSSC.2007.910957
- Yoon, S.R., J. Lee and S.C. Park, 2006. Case study: Case study: NoC based Next-generation WLAN receiver design in Transaction Level. *Proceedings of the 8th International Conference on Advanced Communication Technology*, Feb. 20-22, IEEE Xplore Press, Phoenix Park, pp: 1125-1128. DOI: 10.1109/ICACT.2006.206169