



**HAL**  
open science

# Networks-on-Chip Cortex Inspired Communication To Reduce Energy Consumption

Erwan Moréac, Johann Laurent, Pierre Bomel, André Rossi, Emmanuel Boutillon, Andrea Mineo, Maurizio Palesi

## ► To cite this version:

Erwan Moréac, Johann Laurent, Pierre Bomel, André Rossi, Emmanuel Boutillon, et al.. Networks-on-Chip Cortex Inspired Communication To Reduce Energy Consumption. Design Automation Conference (DAC 2017), Jun 2017, Austin, TX, United States. , Design Automation Conference, DAC 2017. hal-01576029

**HAL Id: hal-01576029**

**<https://hal.science/hal-01576029>**

Submitted on 22 Aug 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Networks-on-Chip Cortex Inspired Communication To Reduce Energy Consumption

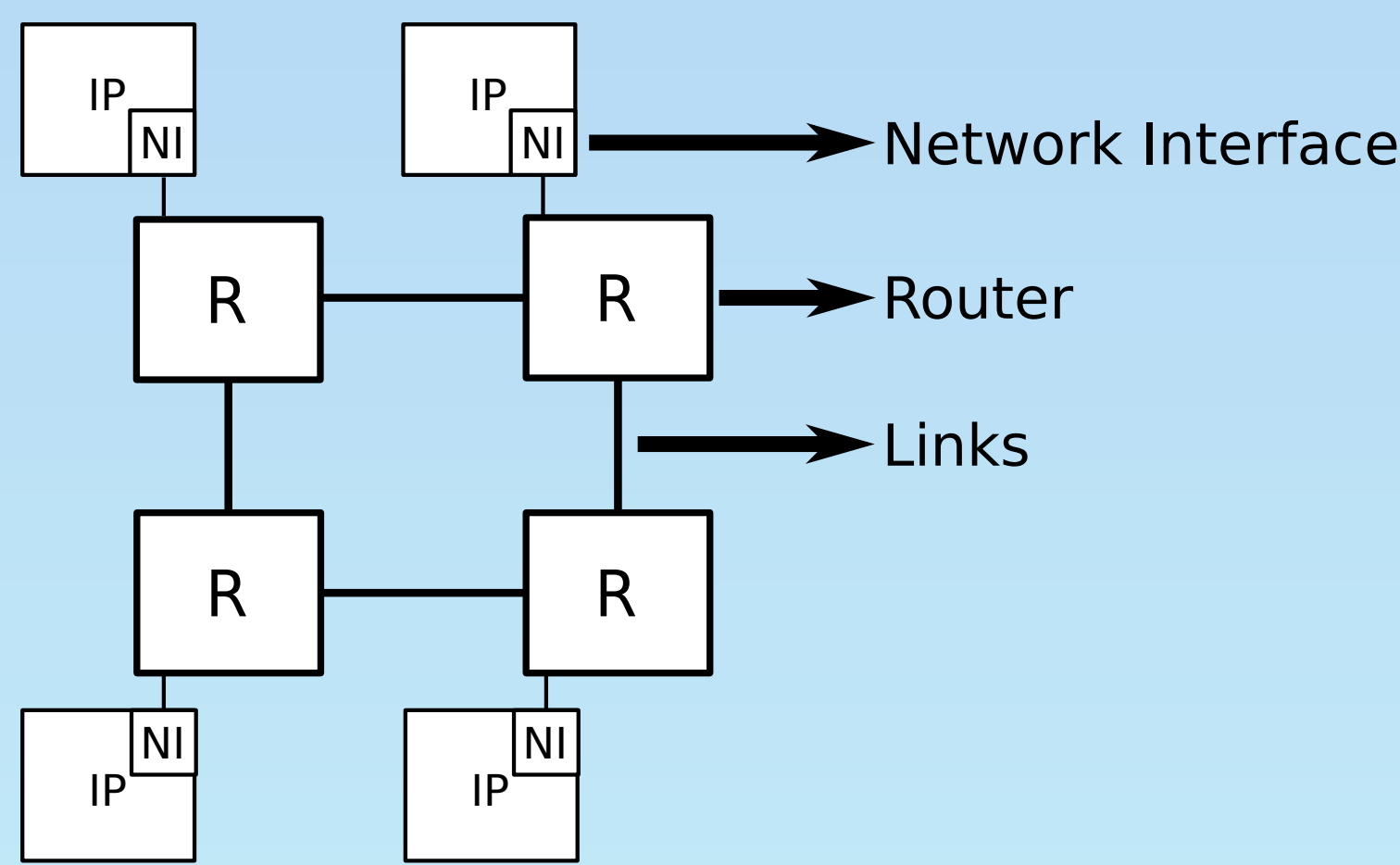
Erwan Moréac<sup>1</sup>, Johann Laurent<sup>1</sup>, Pierre Bomel<sup>1</sup>, André Rossi<sup>2</sup>, Emmanuel Boutillon<sup>1</sup>, Andrea Mineo<sup>3</sup> and Maurizio Palesi<sup>3</sup>

1: Lab-STICC, CNRS UMR 6285, University of South Brittany, Lorient, France  
 2: LERIA, University of Angers, Angers, France  
 3: Department of Computer Science and Telecommunications Engineering, University of Catania, Catania, Italy

## Introduction

Complex many-cores and SoCs contain one (if not several) NoCs. NoC power consumption can represent a significant proportion (40%) of the overall power consumption [M Taylor 2002].

### Main components of a NoC



As technology advances, **links scale badly** in terms of energy and delay as compared to transistors [ITRS 2013].

Links energy optimization is a topic of utmost importance.

## Related works

Links power consumption computation:

$$P_d = \sum_{i=1}^{N_w} \alpha_i \times C_i \times V_{dd}^2 \times f_{clk}$$

- $P_d$ , wires dynamic power consumption
- $N_w$ , the number of wires
- $\alpha_i$ , wire  $i$  switching activity
- $C_i$ , wire  $i$  capacitance
- $V_{dd}$ , power supply
- $f_{clk}$ , clock frequency

Crosstalk effects have an important impact on links power consumption

Several links energy optimizations have been proposed for NoCs:

- Bus coding to avoid Crosstalk in [M Taassori 2009]
- Serialized links [S Ogg 2008]
- Approximate computing [A Mineo 2013]

Only effective on very long wires

No tradeoff energy/throughput considered

**Idea:** mimic the brain processing, only few synapses among myriad are active at the same time.

## Cortex Inspired Communication

Brain is subjected to **two major constraints**:

- low level of available energy (i.e. sugar in the blood)
- 25 W overall maximum power dissipation [E R Kandel 2000]

The human brain stores and uses sensory information by using representational codes that rely on very few active neurons [P Lennie 2003].

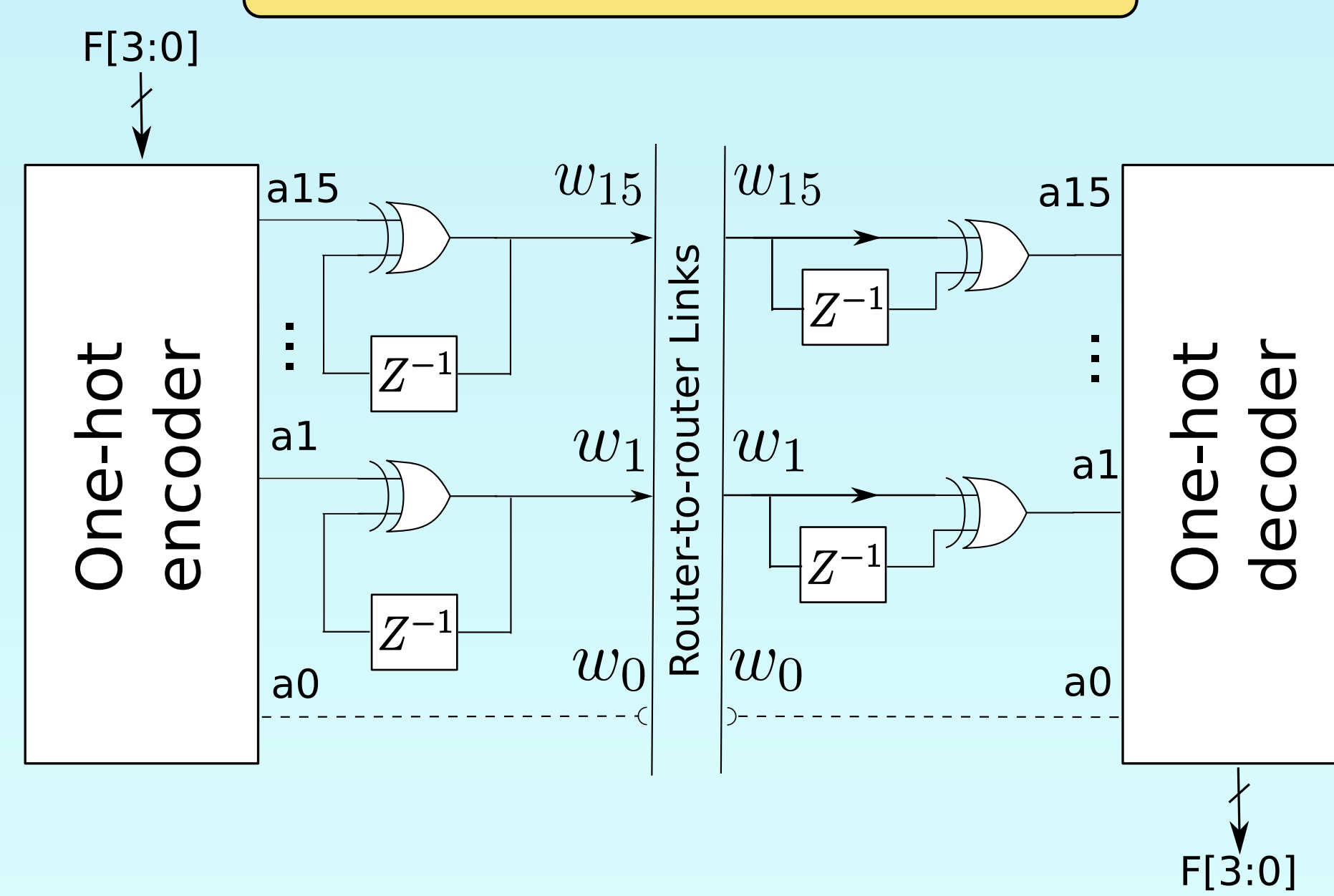
Human neural system	Network-on-chip
• Active nerve fibers rate: 1%	• Active wires rate per transmission: 50%
• Power hungry neurons spikes	• Simultaneous wire transitions cause <b>Crosstalk</b>
• Energy efficient coding representation	• No data coding
	• Network maximum bandwidth <b>under-used</b>

Opportunities to reduce the NoC energy transmission cost

Tradeoff between bandwidth and energy efficiency

**Proposition:** Reduce the wire use rate to decrease energy consumption and delete Crosstalk

## CIC architecture



### Practical example

Clock cycle $i$	Data to send $F[3:0]$ (4 bits)	One-hot coding $a_{15}-a_0$ (16 bits)	Data on links $w_{15}-w_0$ (16 bits)
0	0001	0000 0000 0000 0010	0000 0000 0000 0010
1	1010	0000 0100 0000 0000	0000 0100 0000 0010
2	0101	0000 0000 0010 0000	0000 0100 0010 0010
3	0001	0000 0000 0000 0010	0000 0100 0010 0000

Allow only **one transition** among a set of wires



**Problem:** only 1 bit of data sent

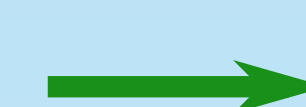


Use the bit position as an additional data. For a bus of  $N$  wires, it gives  $\log_2(N)$  bits of data

This is still not enough to send a data in one cycle, we must **decompose** a flit in several parts



**Problem:** Increase the packet transmission time



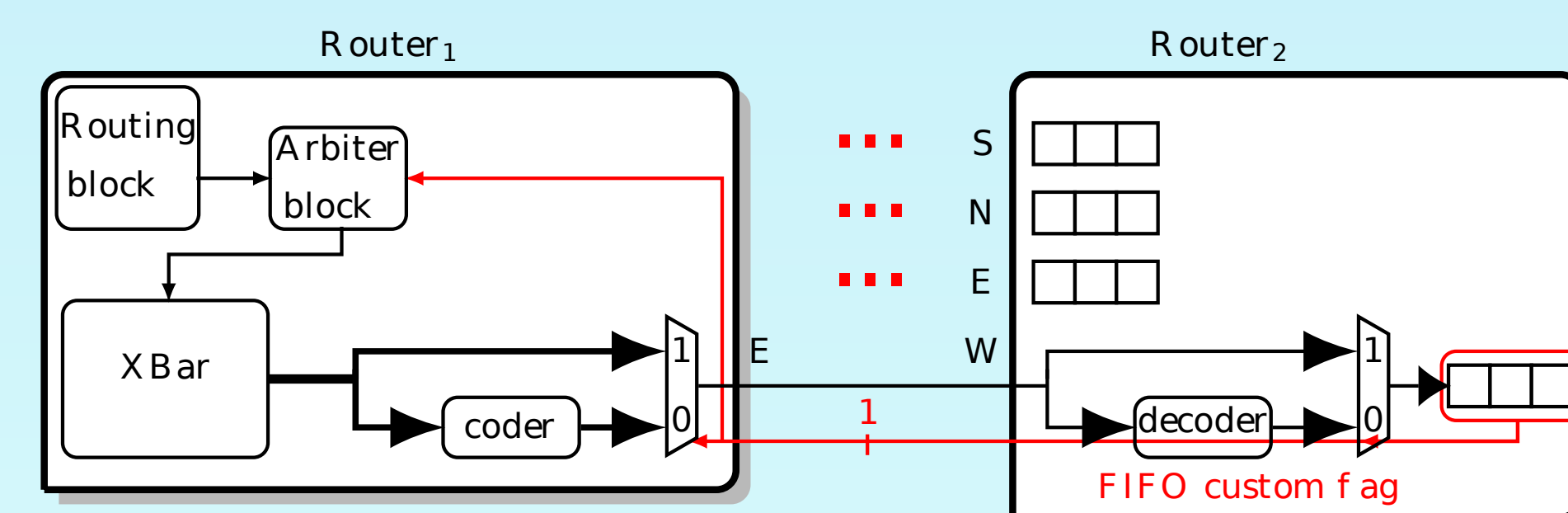
Allow the CIC transmission considering the network local state

## Transmission strategies

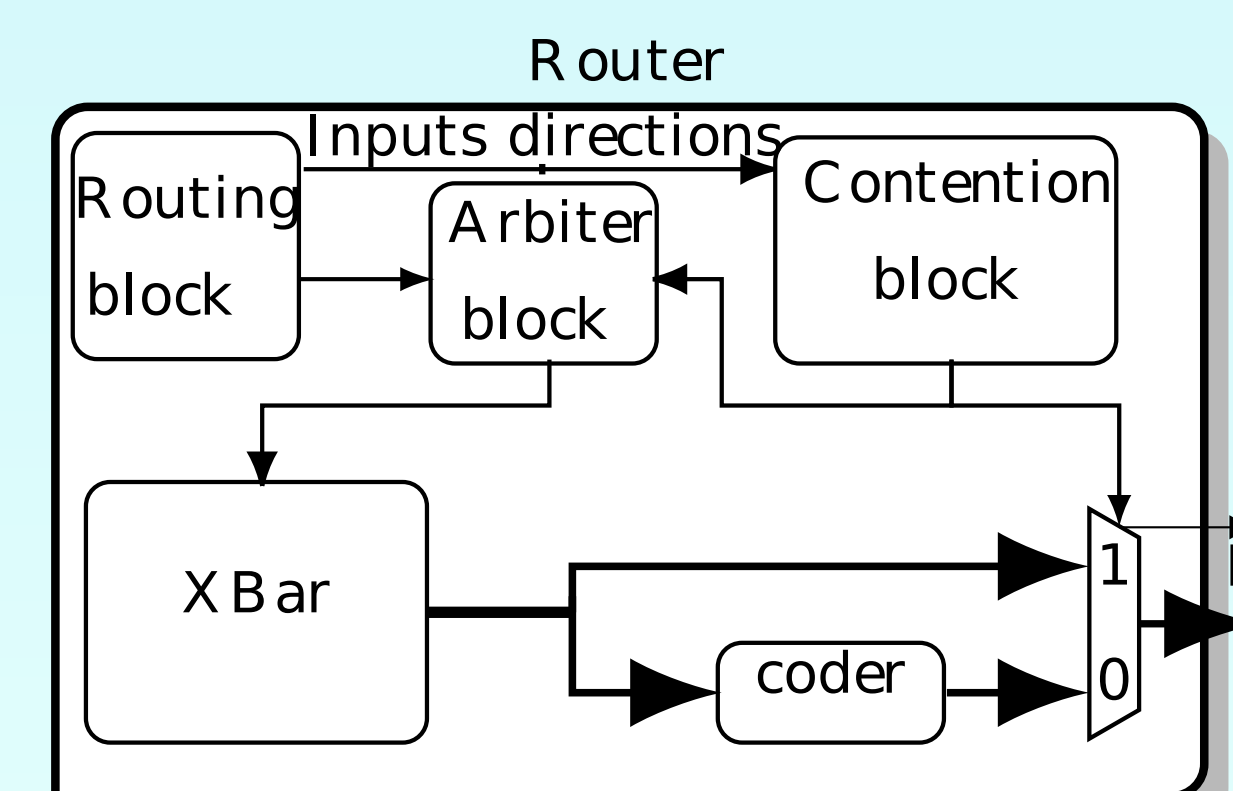
CIC restriction according to the **local state** of the network. Three strategies are proposed:

- OCC : Buffer occupancy checking of the following router
- CONT : Contention checking at each output
- CONT+OCC : Restriction increased with both checking

### Buffer occupancy checking architecture



### Contention checking architecture



## Results and conclusion

### Application description

- Dijkstra: compute the shortest path
- 5app: Susan-corners, Susan-edges, LU, Water and Water-spatial
- 5 clust app: same apps as 5 app, each app uses 4 cores in parallel
- Full HD: 150MHz video flow through 6 image processing IPs

### Experimental set up

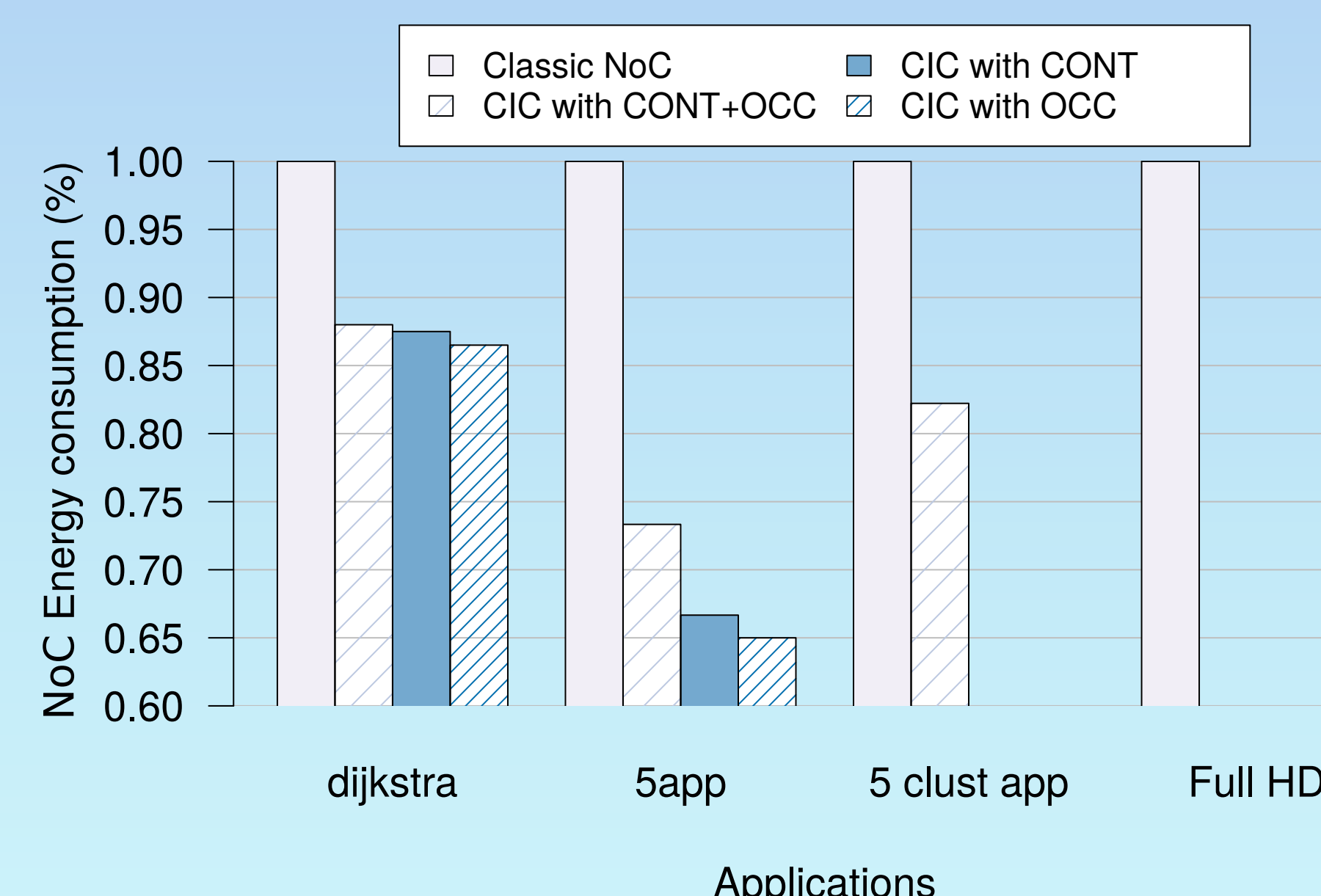
- Cycle-accurate, bit-accurate NoC simulator
- Application traces extraction from MPSoCBench
- Hardware overhead considered
- NoC 4x3 clocked at 1GHz, XY algorithm
- 65nm CMOS technology,
- Wires length of 2mm, 32 bits

### Cons

- Strong delay overhead with OCC strategy
- The NoC need a frequency increase to run unidirectional high throughput video application
- Area overhead of 21% per router

### Pros

- Up to 67% of links energy saving
- Up to 35% of total energy saving
- Delay between computations exploited



Application	CIC architecture with OCC strategy				CIC architecture with CONT strategy				CIC architecture with CONT+OCC strategy			
	Links energy (%)	Encoder use rate (%)	Extra delay (cycles)	Processing throughput (%)	Links energy (%)	Encoder use rate (%)	Extra delay (cycles)	Processing throughput (%)	Links energy (%)	Encoder use rate (%)	Extra delay (cycles)	Processing throughput (%)
Dijkstra	-67.7	99.6	42	0	-61.5	66.5	18	-0.001	-59.5	62.9	13	0
5 applications	-63.6	94.98	21	-0.001	-57.5	64.4	8	-0.001	-50.6	58	7	0
5 Clusters applications	-63.2	78.4	25653	-27.84	-62.3	62.5	12341	-12.5	-31.2	36	17	-0.01
Full HD	-64.1	78	42477	-42.2	-57.2	88	29070	-29	-57.2	87.5	29070	-28.9

## Future works

- Explore other CIC configurations
- Smarter strategy to handle high traffic apps
- Adapt the method to design a CIC router