Fault Tolerant Network-on-Chip for Deep Learning Algorithms

COMPAS – July 7 2021

Mercier Romain
Univ. Rennes, Inria, IRISA
romain.mercier@irisa.fr

Director:
Chillet Daniel
Univ. Rennes, Inria, IRISA
daniel.chillet@irisa.fr

Supervisor 1:
Killian Cédric
Univ. Rennes, Inria, IRISA
cedric.killian@irisa.fr

Supervisor 2:
Kritikakou Angeliki
Univ. Rennes, Inria, IRISA
angeliki.kritikakou@irisa.fr

DGA Tutor:
Helen Youri
DGA MI/STD/MAN/BSA
youri.helen@intrade.gouv.fr

Context of this Work

- **Network-on-Chips (NoCs) paradigm is considered**
  - The most dominant for on-chip communications [1]
  - Network Interfaces (NIs) are used as link between IPs and NoC [2]
  - Messages cross the NoC from source IP towards destination IP
- **NoCs can be affected by physical failure mechanisms due to several effects** [3]
  - Occur more frequently due to nanoscale technologies and power scaling
  - Affect data (flits) which transit on device circuit [4]
  - Crossbar, buffers and interconnections have the highest risk to be impacted due to them high sizes
- **Multiple permanent faults on datapath are considered** [5]
  - Stuck-at fault model is used
  - Flits are equally impacted

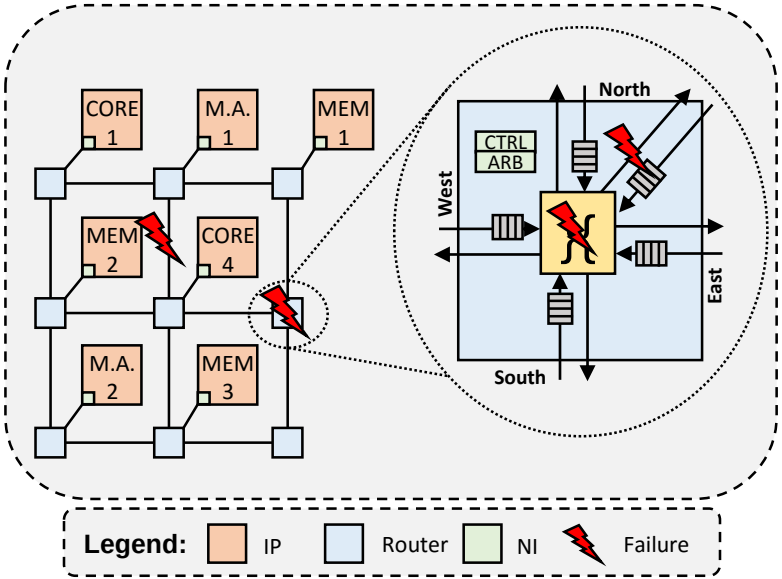


Figure 1: Original 3x3 mesh NoC and router representations

How multiple permanent faults can be managed in NoCs?

NoC Message Formating

Symbol	Definition
S_{msg}	Message size
S_{pck}	Payload size in packets
S_{flit}	Size of a flit
S_{SF}	Size of a subflit
$N_P = S_{msg}/S_{pck}$	Number of packets
$N_F = S_{pck}/S_{flit}$	Number of flits
$N_{SF} = S_{flit}/S_{SF}$	Number of subflits

Table 2: Extended notation summary

- **Message is plit into one or several packets**
- **Packets are split into flits**
- **One header is added for routing control**

For this work, we split the flits into several subflits

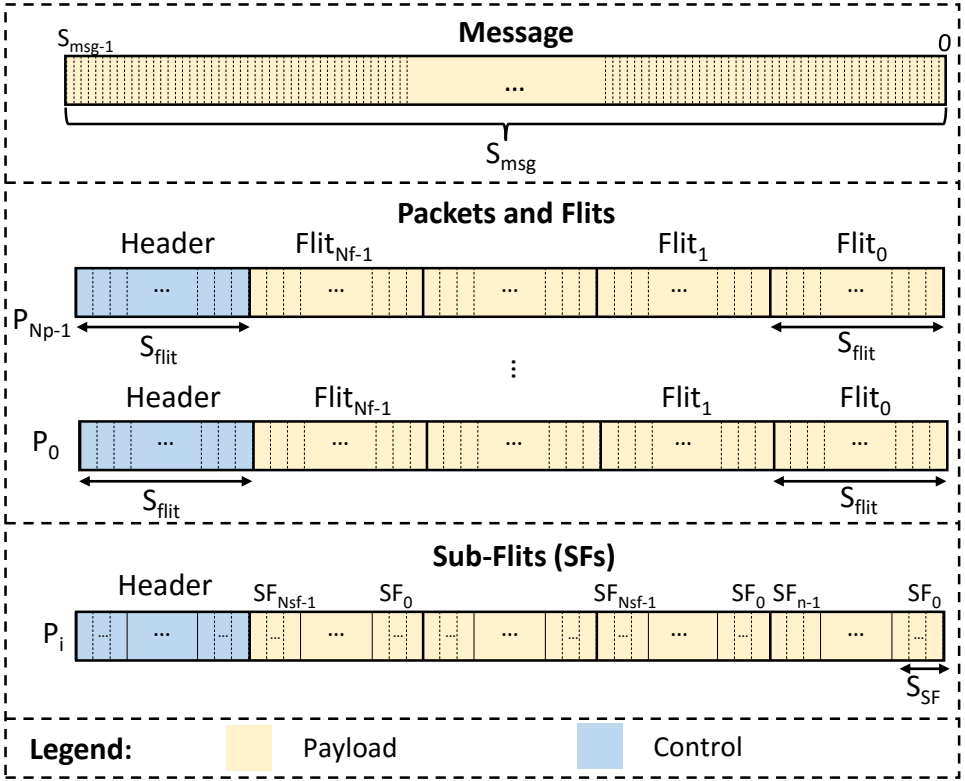


Figure 6: Message Formatting into packets, flits and sub-flits

Bit-Shuffling (BiSu) Method[6]

- **Fault mitigation in the NoCs**
  - Manage fault on datapath
  - Can be extended to virtual channel architectures
- **Fault mitigation at run time**
  - Shuffler block re-orders subflits before faulty path
  - Faults impact Least Significant Bits (LSBs) instead of Most Significant Bits (MSBs)
  - Deshuffler block re-organizes subflits at their original configuration
- **Flit organization managed in the NI (Merger block)**
- **Distribution on two flits for sensible data protection (header)**

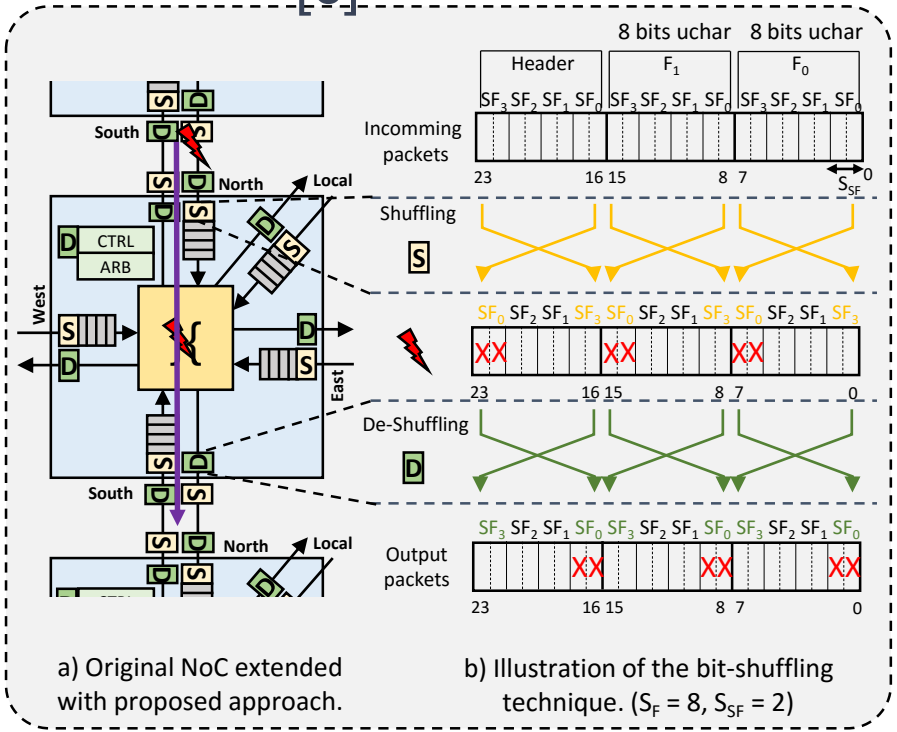


Figure 2: Illustration of the proposed Bit-Shuffling method

[6] R. Mercier, C. Killian, A. Kritikakou, Y. Helen, and D. Chillet. "*Multiple Permanent Faults Mitigation Through Bit-Shuffling for Network-on-Chip Architecture*". In IEEE Int. Conf. on Comput. Des. (ICCD), pages 205–212, Oct 2020.

Hardware Implementation

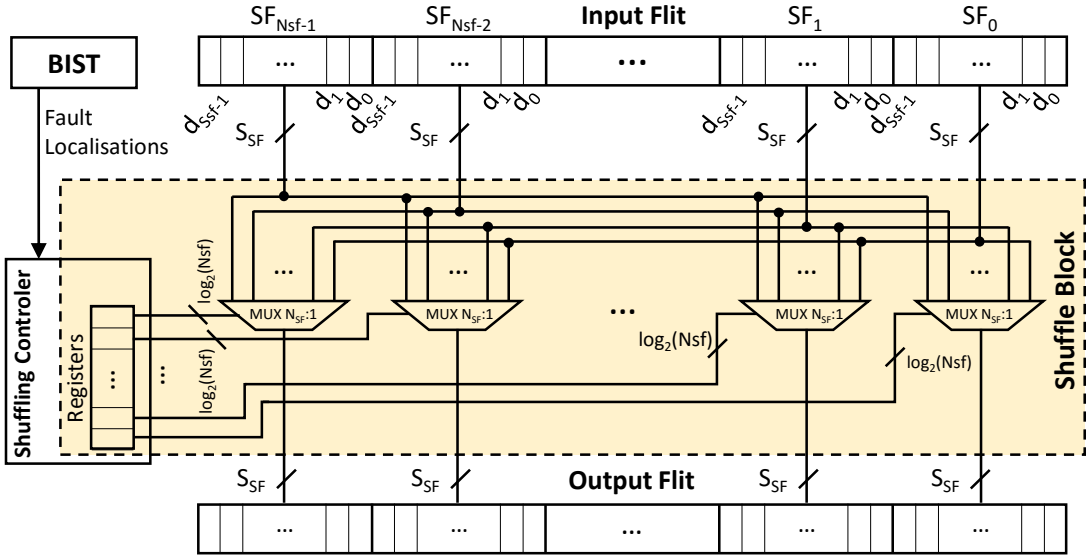


Figure 8: Hardware architecture of a shuffle block

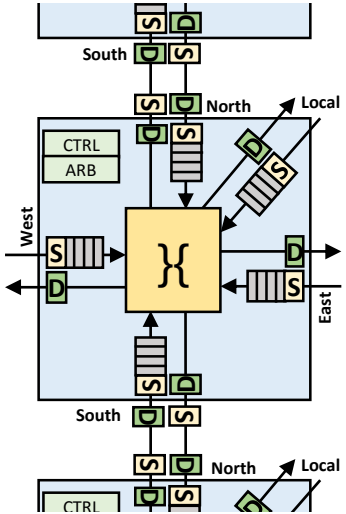
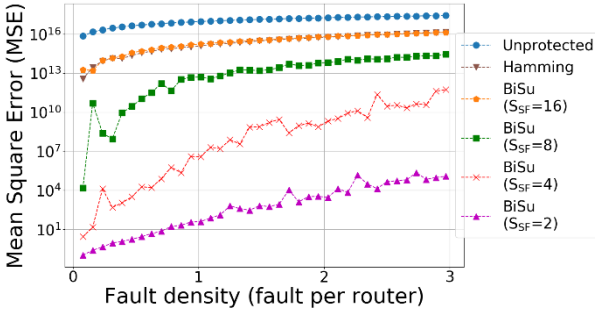


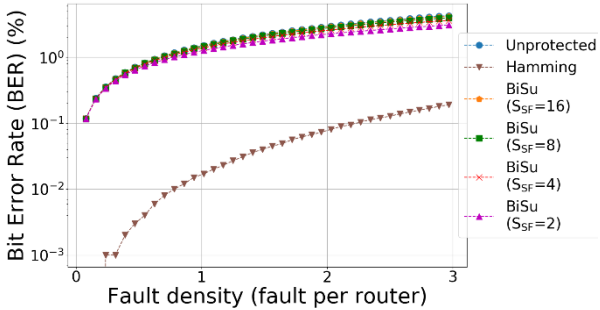
Figure 9: Original NoC extended with the proposed appraoch

- **Fault localization with external methods (Built-In Self-Test (BIST)** [7])
- **Registers are updated offline according to fault localizations by the shuffling controler**

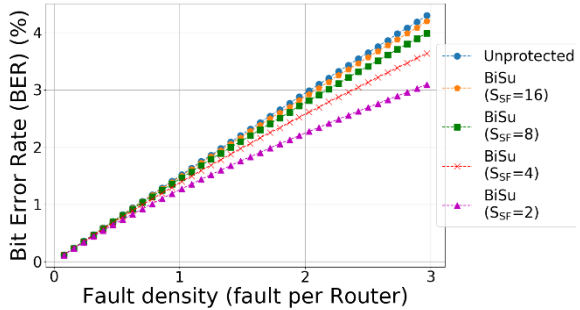
Flit-Level Evaluation



a) Mean Square Error (MSE) study



b) Bit Error Rate (BER) study with Hamming



c) Bit Error Rate (BER) without Hamming

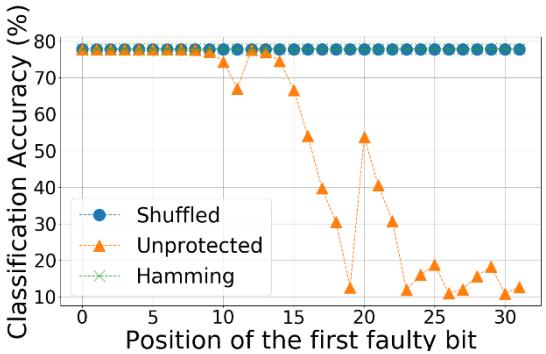
Figure 3: Payload flit accuracy in a 8x8 NoC with 32-bit flits in presence of Single Hard Errors (SHEs)

- **Experimental setup:**
  - Flit Definitions:
    - Packet size: 16 flits
    - XY routing algorithm
    - Flit injection pattern: TORNADO
  - Fault Definitions:
    - Fault model: Stuck-at (bit-flip)
    - Fault size: 1 bit
    - Fault distribution: Random
    - Number of fault injection pattern: 10,000

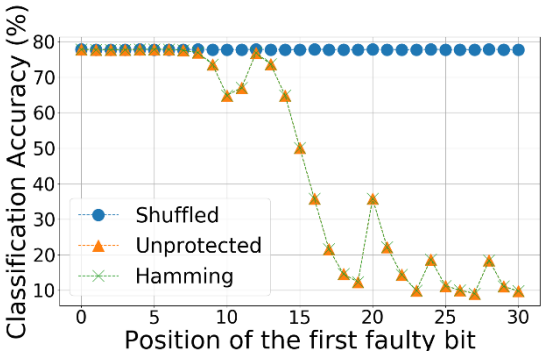
- **Experimental results:**
  - The MSE is reduced with the subflit size
  - The BER is reduced with the subflit size
  - Hamming code has better BER due to bit corrections
  - BiSu method has a better MSE than Hamming code

BiSu more efficient in presence of multiple faults

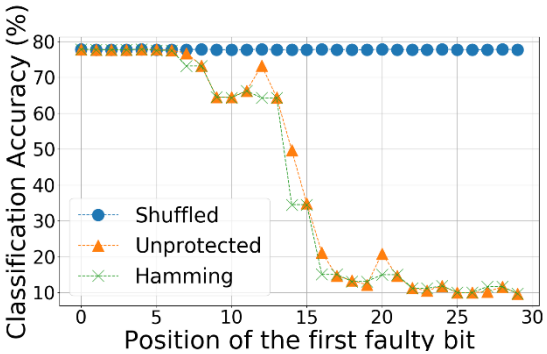
Application-Level Evaluation



a) Fault size: 1 bit



b) Fault size: 2 bits



c) Fault size: 3 bits

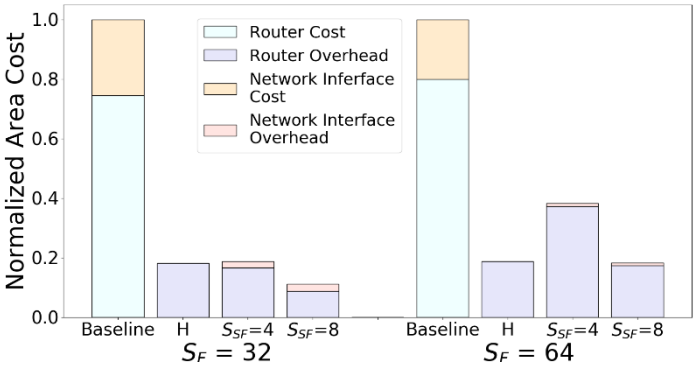
Figure 3: Classification accuracy of a CNN on the CIFAR10 database with two adjacent hard errors of size 2 and 3

- **Experimental setup:**
  - CNN Definitions:
    - Validation accuracy: 79%
    - Data size: 16 bits
    - Flits size: 32 bits
  - Fault Definitions:
    - Fault model: Stuck-at (bit-flip)
    - Fault injected at each possible position
    - Faults injected at the flit level

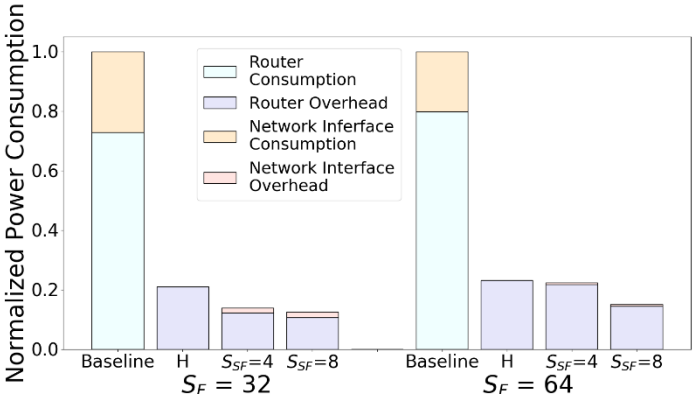
- **Experimental results:**
  - Hamming code inefficient for two or more faults
  - BiSu method maintains the classification rate
  - Classification accuracy degraded when the bit number 9 is affected
  - BiSu can handle up tol 9 faulty bits (positions 0 up to 8)

BiSu more efficient in presence of multiple faults

Hardware Costs Evaluation



a) Area overhead



b) Power overhead

Figure 4: Overhead comparison between BiSu method and Hamming code for a 8x8 NoC using router of CONNECT[8]

- **Experimental setup:**
  - High Level Synthesis (HLS) for cost estimation:
    - Technology: 28 nm FDSOI
    - Sowtfare: Mentor Graphic tools
    - Clock: 1 GHz
- **Experimental results:**
  - BiSu global overhead depends of the flit size and the subflit size
  - The area overhead is correlated with the subflit number
  - The power consumption is particularly reduced with the BiSu method

The lightweight BiSu configurations outperforms the Hamming method

Conclusion

- **Multiple permanent fault mitigation with the Bit-Shuffling (BiSu) technique**
- **Target the fault tolerant applications in harsh environnements**
- **High efficiency in presence of multiple permanent faults**
- **Low hardware costs in terms of area costs, power consumption and latency**
- **Outperform the current method of the state of the art, e.g. Hamming code**
- **Can be adapted to ASICs and FPGAs implementation**

Bibliography

[1] J. Xu, W. Wolf, J. Henkel, and S. Chakradhar. A Methodology for Design, Modeling, and Analysis of Networks-on-chip. In Proc. IEEE Int. Symp. Circuits and Syst. (ISCAS), volume 2, pages 1778–1781, May 2005.
[2] Babak Aghaei, Midia Reshadi, Mohammad Masdari, Seyed Hadi Sajadi, Mehdi Hosseinzadeh, and Aso Darwesh. Network adapter architectures in network on chip: comprehensive literature review. Cluster Computing, 23(1):321–346, 2020.
[3] M. Radetzki, C. Feng, X. Zhao, and A. Jantsch. Methods for Fault Tolerance in Networks-on-Chip. ACM Comput. Surv., 46(8):1–38, July 2013.
[4] Space Product Assurance: Techniques for Radiation Effects Mitigation in AASIC and FPGAs Handbook. Technical report, ESA Requirements and Standards Division, Sept. 2016.
[5] Single Event Effects Mitigation Techniques Report. Federal Aviation Admin., William J. Hughes Tech. Center, Feb. 2016.
[6] R. Mercier, C. Killian, A. Kritikakou, Y. Helen, and D. Chillet. "*Multiple Permanent Faults Mitigation Through Bit-Shuffling for Network-on-Chip Architecture*". In IEEE Int. Conf. on Comput. Des. (ICCD), pages 205–212, Oct 2020.
[7] B. Bhowmik, S. Biswas, J. K. Deka, and B. B. Bhattacharya. A Low-Cost Test Solution for Reliable Communication in Networks-on-Chip. J. of Electron. Testing, 35(2):215–243, Apr. 2019.
[8] M. K. Papamichael and J. C. Hoe. CONNECT: Re-examining Conventional Wisdom for Designing Nocs in the Context of FPGAs. In Proc. ACM/SIGDA Int. Symp. Field Program. Gate Arrays (FPGA), pages 37–46, Feb. 2012.

Thank you!

Follow us on www.inria.fr

DGA

UMR IRISA

Inria inventors for the digital world