Sensor Network on Chip

A HW-SW Collaborated Method for Resilient MPSoC

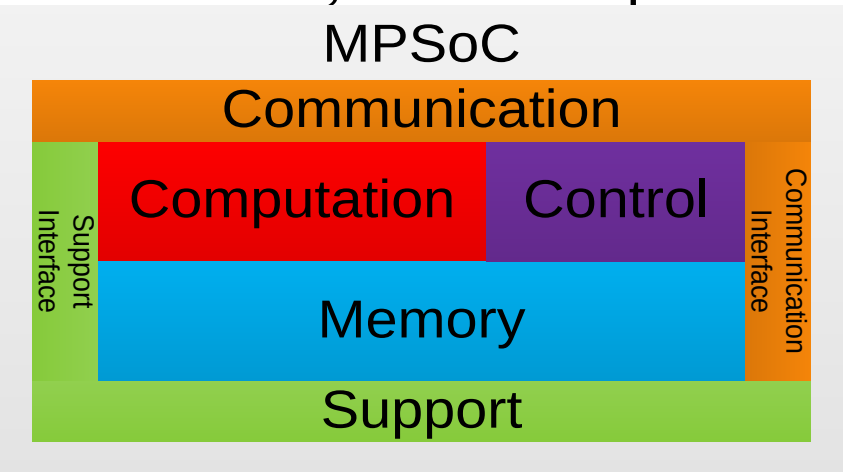
Jiang Xu

MOBILE COMPUTING SYSTEM LAB

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

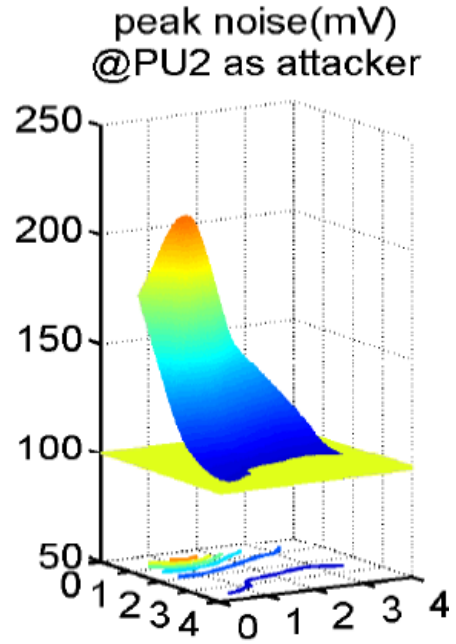
Subsystems of Multiprocessor System-on-Chip

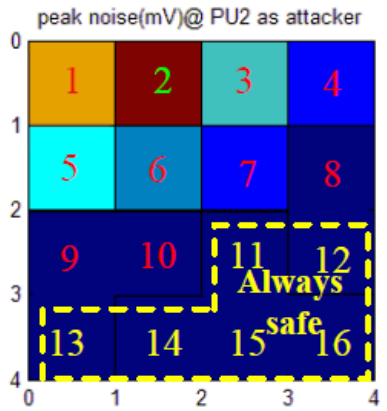
- Computation: process data, implement major functions
- Control: coordinate all the subsystems
- Memory: temporally store data, instructions, and system status
- Communication: transfer data, instructions, and other information inside, into, and out of an MPSoC
- Support: maintain appropriate operating conditions, such as power supply, clock, temperature, *etc.*
- Can physically overlap
  - E.g. computation and control
- There are grey areas
  - E.g. communication interfaces



Transient Threats to MPSoC

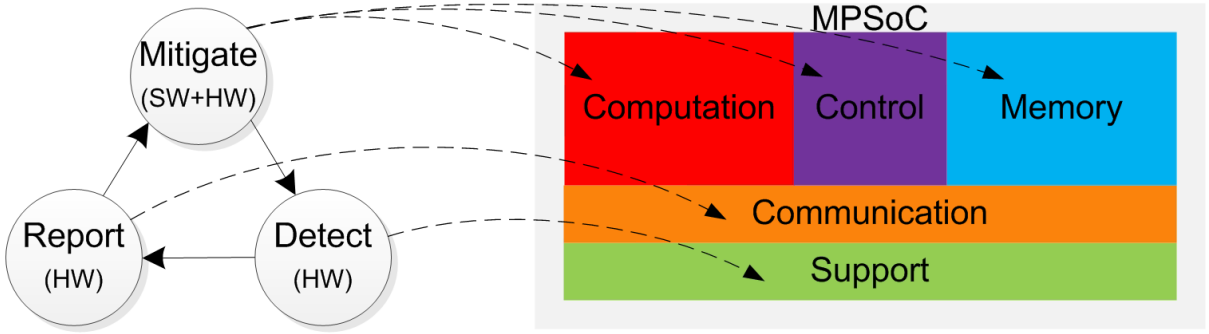
- MPSoC is susceptible to various transient threats
  - Soft errors, power/ground noise, *etc.*
  - Due to smaller feature sizes, narrower noise margins, larger transistor numbers, and wider usages
  - Result in unpleasant user experiences and even serious dangers
- Traditional solutions use rigorous designs for worst cases
  - Reinforced devices and circuits, triple modular redundancy, *etc.* [1]~[3]
  - Substantial performance, power, and area overheads
- Is it possible to make MPSoC more resilient with a low overhead?





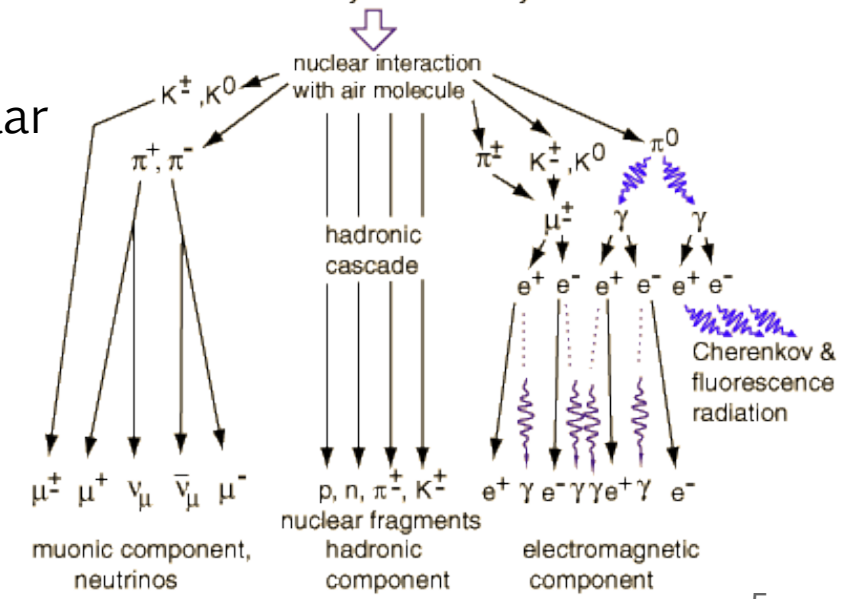
Sensor Network on Chip Overview

- Root of the problem
  - Traditional solutions are often designed for worst cases
  - Worst cases only account for a small portion of MPSoC runtime
- Increasingly difficult to determine worst cases
- Sensor network on chip (SENoC)
  - Detect transient threats at run time
  - Mitigate transient threats adaptively and collaboratively
  - HW-SW collaborations to reduce cost



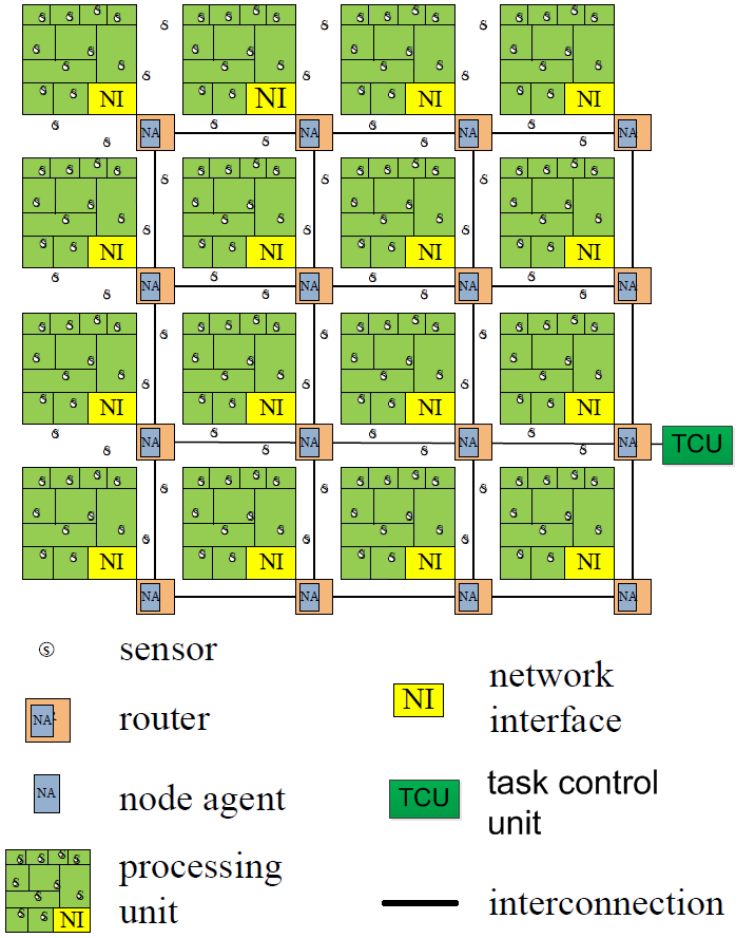
Case Study: Soft Error Tolerance

- Soft error is caused by high-energy particles or electromagnetic interference
  - State inversions in registers or memory elements
  - Or indirectly through glitches propagating across combinational logic
- Existing solutions
  - Hardware-based solutions [3]~[7]
    - Reinforced gate/circuit, micro/macro-architecture, triple modular redundancy
  - Software-based solutions [8]~[12]
    - Spatial/temporal redundancy of process/thread, task, code block, instruction
  - Large power, performance, and area overheads



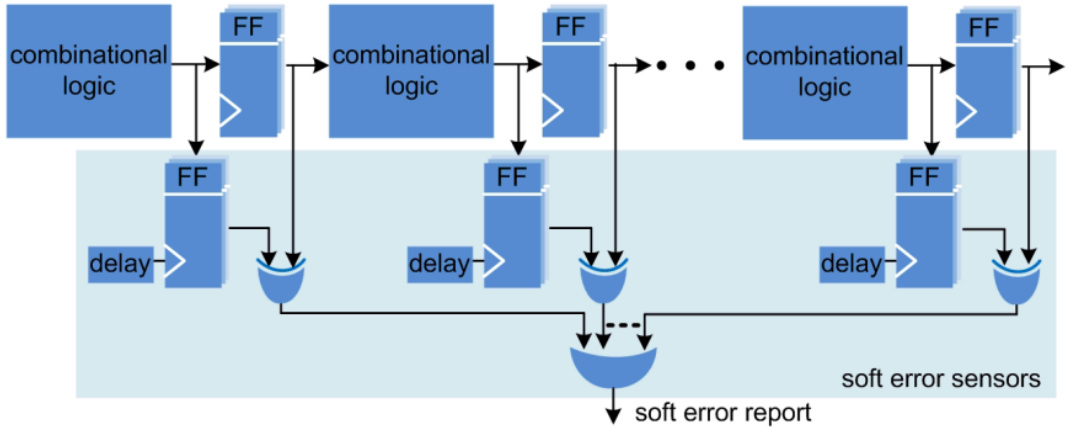
SENoC HW Architecture Overview

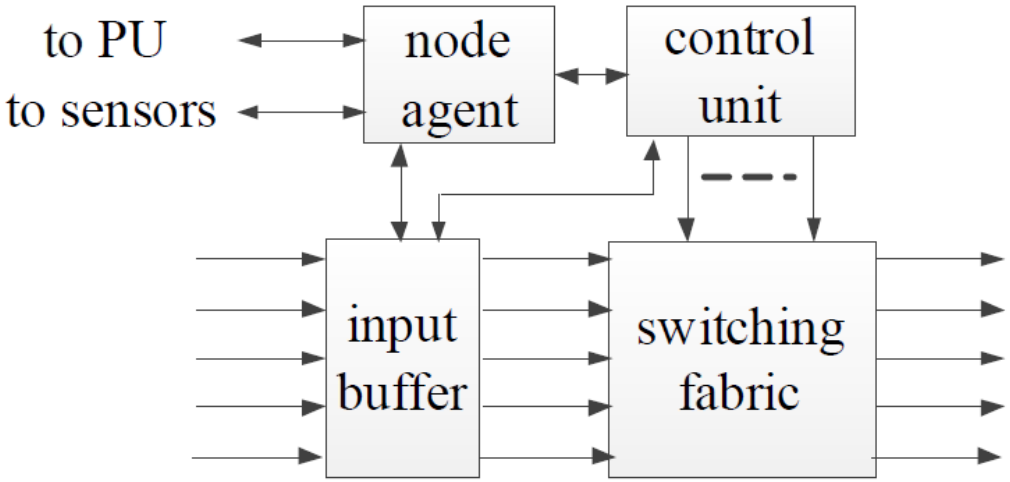
- On-chip sensors
  - Embedded inside and between PUs
  - Measure various parameters, such as voltage
  - Automatically report special conditions
  - Report normal conditions by requests
- Node agents
  - Preprocess and compress info from local sensors
  - Report info to other agents and TCU
  - Analyze info from other PUs and TCU
- Task control unit
  - Issue tasks onto PUs and coordinate PU actions
  - Adjust the scheduling assumptions based on node agent reports



SENoC HW Architecture Overview

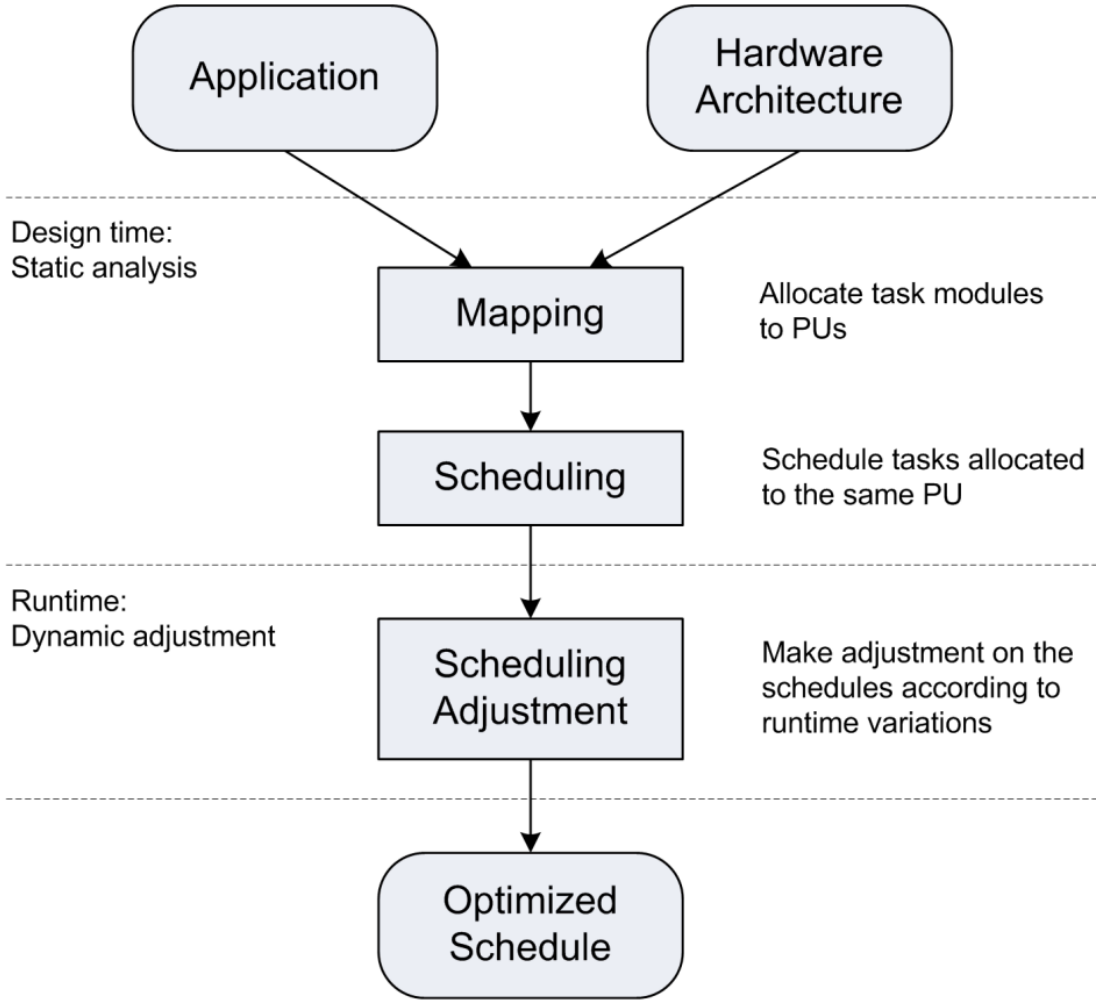
- Soft error sensor
  - Shadow flip-flop with a delayed clock
  - Effective for detection of soft error occurred on combinational/sequential logics
- Network-on-chip
  - Serve as the communication backbone
  - Integrate node agents with NoC routers
  - Use virtual channels to ensure timely delivery





Two-stage Task Scheduling

- Static mapping and scheduling at design time
  - Performance optimization for normal conditions
  - The complexity of the algorithm can be high
- Dynamically adjust schedules at run time
  - Roll back faulty tasks
  - Minimize soft error impacts
  - The complexity must be low for runtime executions



Static Scheduling and Dynamic Adjustment (SSDA)

Static Scheduling and Dynamic Adjustment (SSDA)

Algorithm 1 The offline static load balanced mapping and static order scheduling algorithm

Require: application graph $G(V, E)$, architecture P

```
1: time = 0, sl = 0
2: scheduleLength = GetScheduleLength(G)
3: while sl < scheduleLength do
4:    readyQueue = UpdateReadyQueue()
5:    for each task v in readyQueue do
6:       if v is already mapped to m(v) then
7:          selectedProc = m(v)
8:       else
9:          minWeight = ∞
10:         for each processor p in P do
11:            if w(v, p) < minWeight then
12:               selectedProc = p
13:               minWeight = w(v, p)
14:            end if
15:         end for
16:         m(v) = selectedProc
17:      end if
18:      s(v, selectedProc) = GetOrder(selectedProc)
19:      procAvailTimes = GetEarliestAvailableTimes(P)
20:   end for
21:   time = TimeAdvance(procAvailTimes)
22:   sl = FinishTaskExecution(G)
23: end while
24: return mapping M, schedules S
```

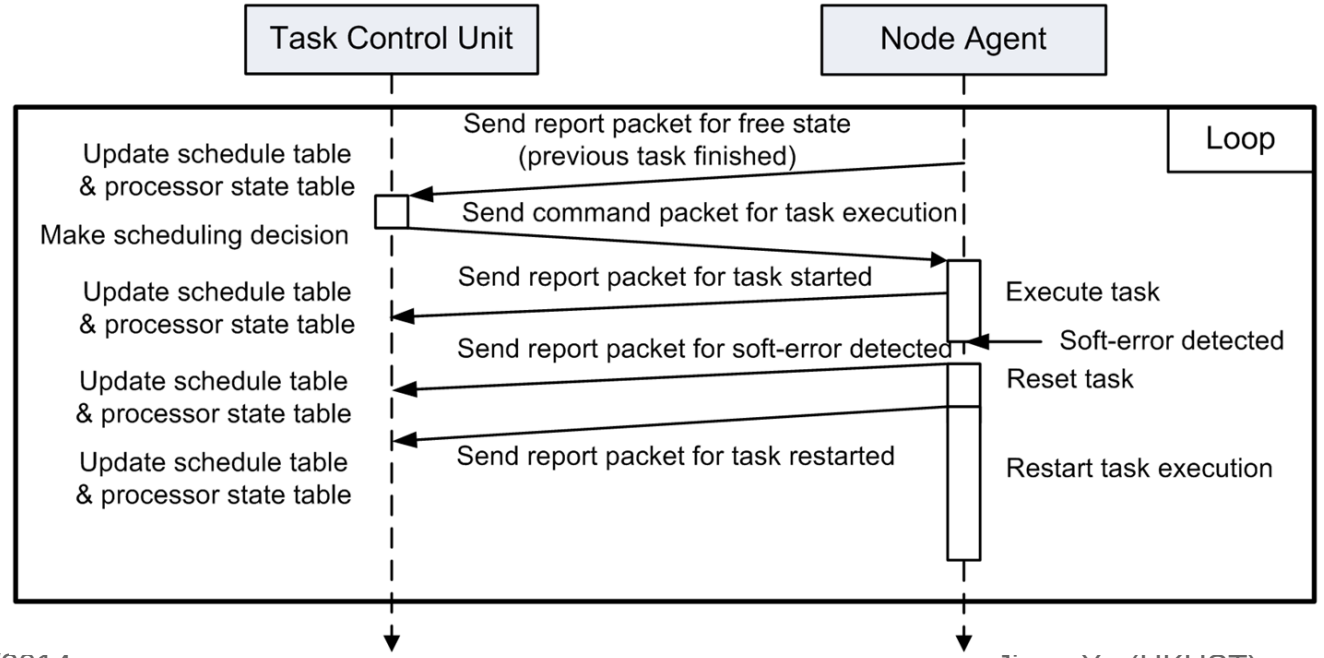
Algorithm 2 The light-weight online dynamic scheduling adjustment algorithm

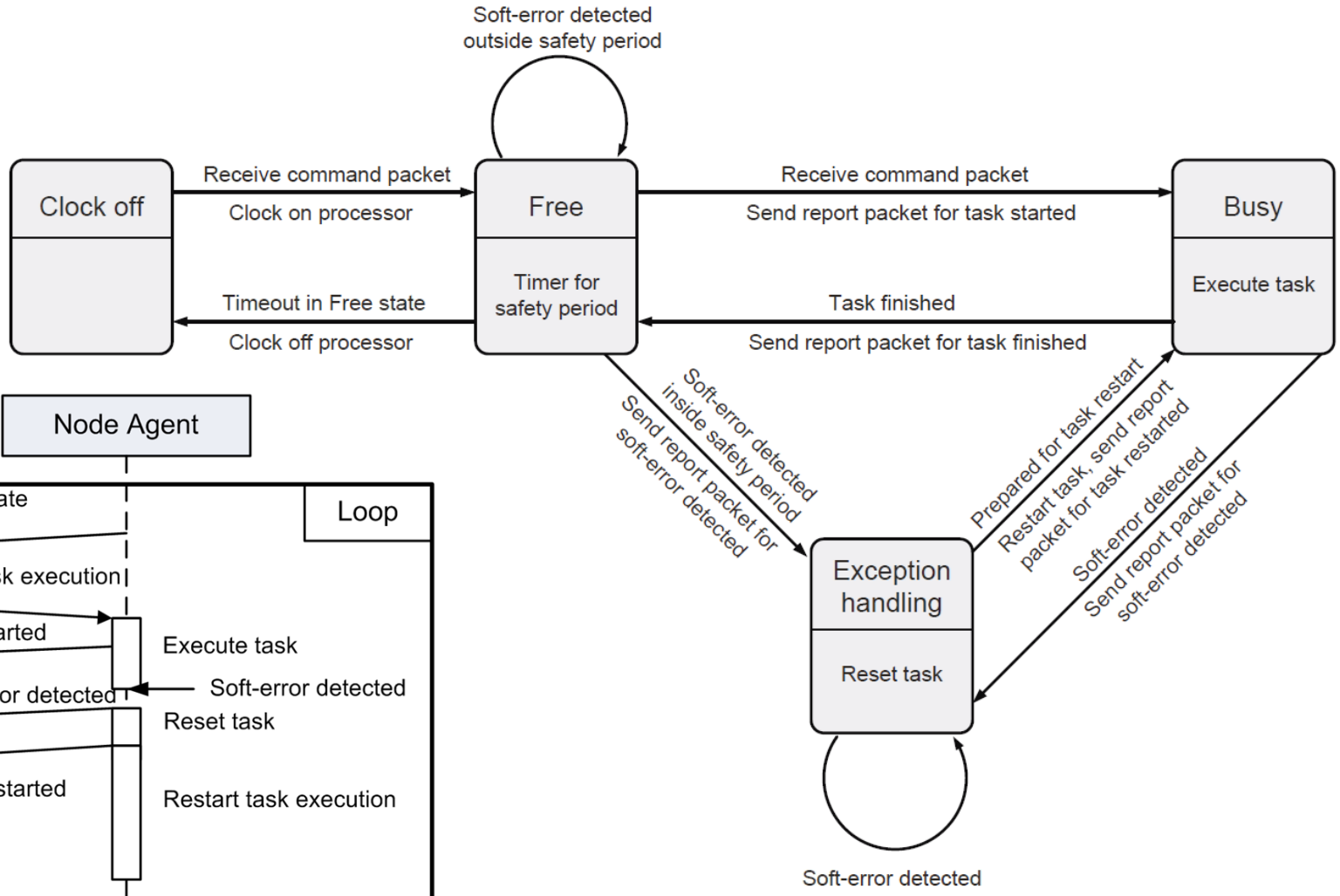
Require: application graph $G(V, E)$, architecture P, mapping $M(V, P)$, static order schedules $S(V, P, N)$

```
1:  for each processor p do
2:     if p reports free then
3:        v = NextTask(p, S)
4:        while v is not ready do
5:           v = NextTask(p, S)
6:        end while
7:        Schedule v on p
8:        Update st(v)
9:     end if
10:    if p reports processor state then
11:       Update st(v)
12:    end if
13: end for
14: return makespan and schedule table st(V)
```

Protocols to Mitigate Soft Error

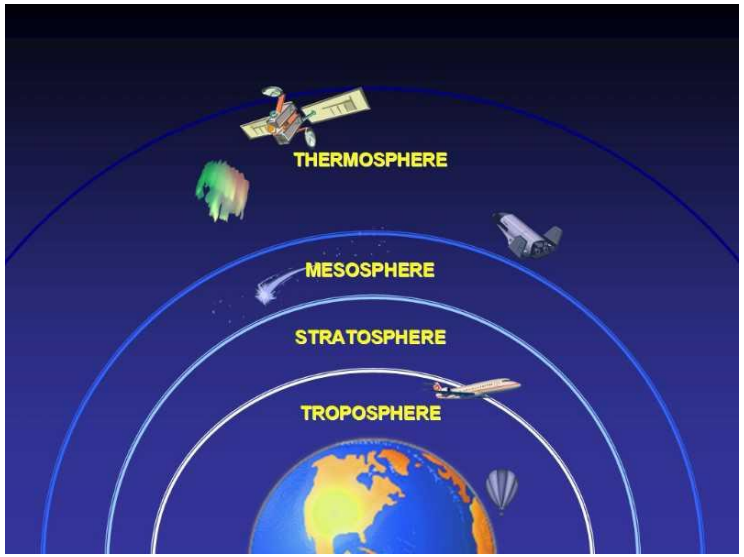
- HW detection
- SW mitigation
- PU, agents, TCU, NoC work together

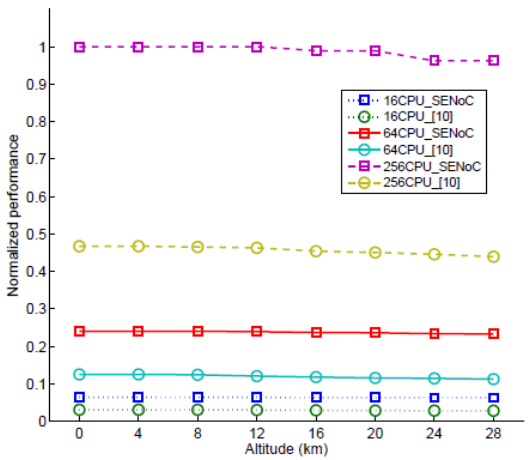




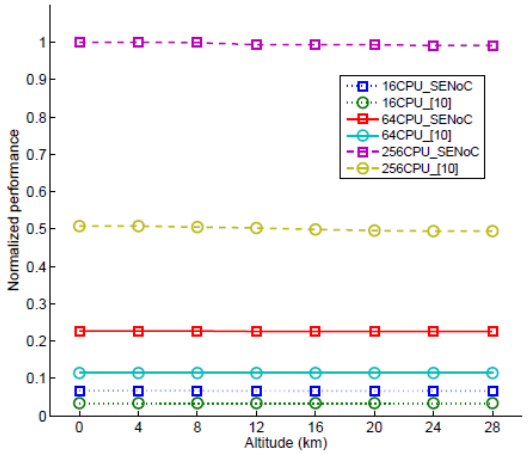
Performance Degradation at High Altitudes

- Cosmic ray flux levels at different altitudes are based on JEDEC JESD89 standard
- About 2X performance improvement
- Performance degradation is small within civil and general aviation altitudes

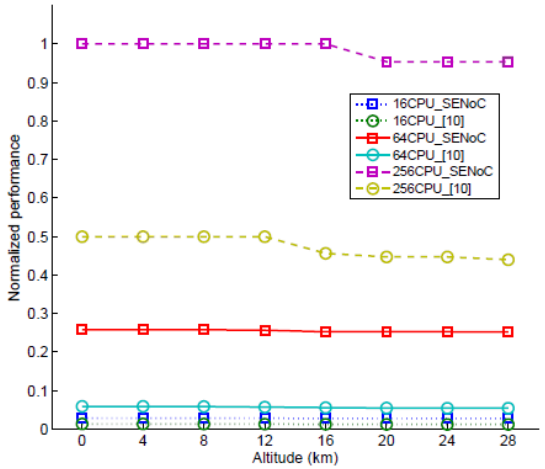




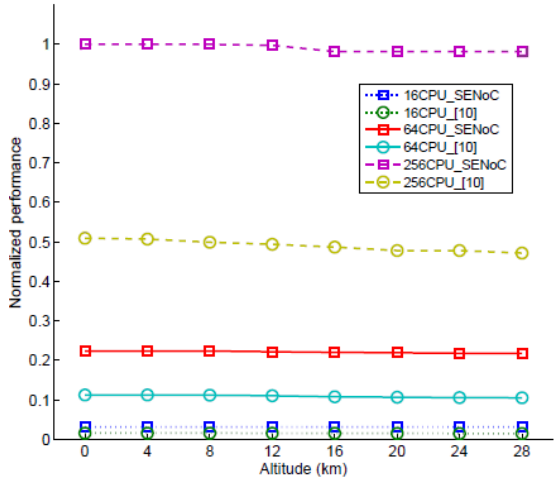
H.264 decoder



Sample rate converter



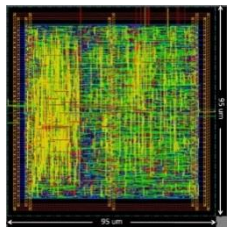
Sparse matrix solver



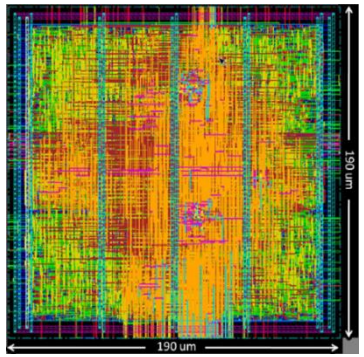
MJPEG encoder

Cost and Scalability

- Power overhead is less than 5%
- Area overhead is small
- TCU scalability is linear

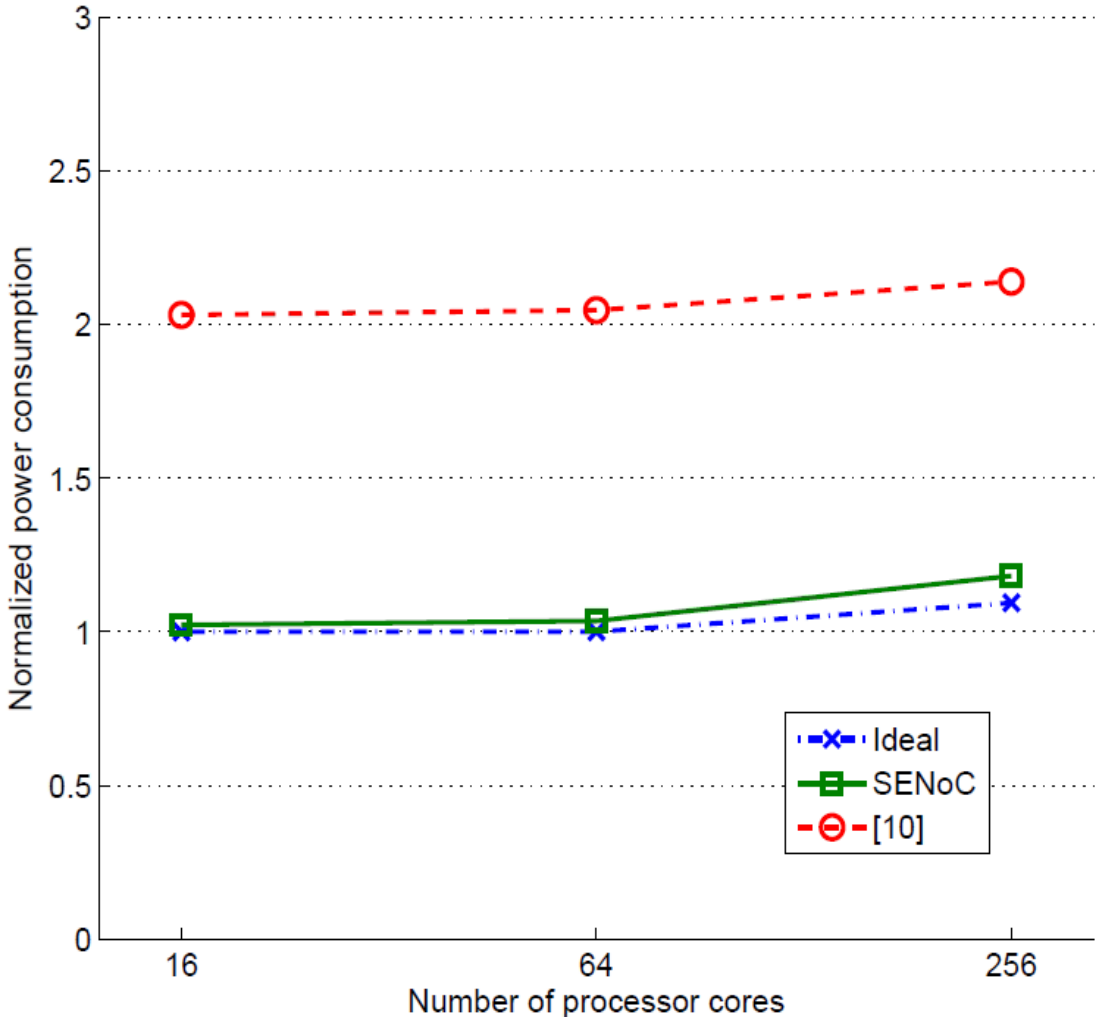


TCU



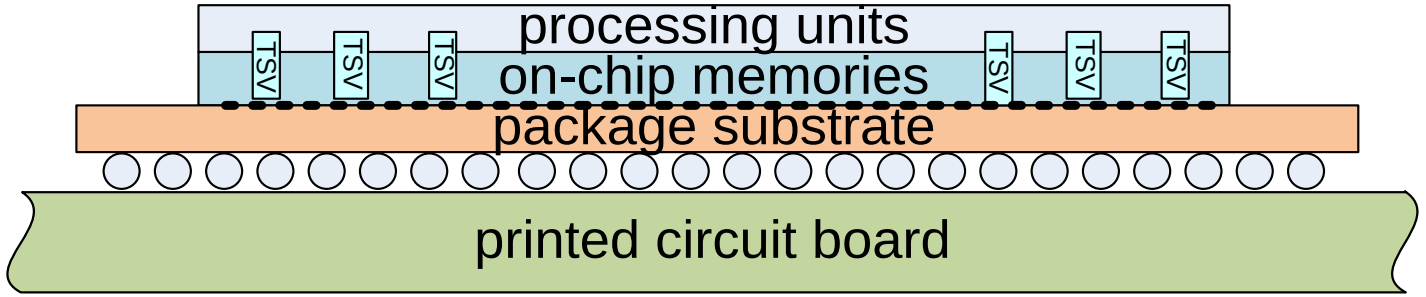
Router

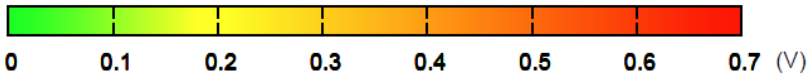
	Area (μm^2)	Energy consumption (μW/MHz)
Processor core	435600	240
Router	22803	24
TCU on 16-core MPSoC	8599	2.2
TCU on 64-core MPSoC	50608	11.1
TCU on 256-core MPSoC	127315	61.4

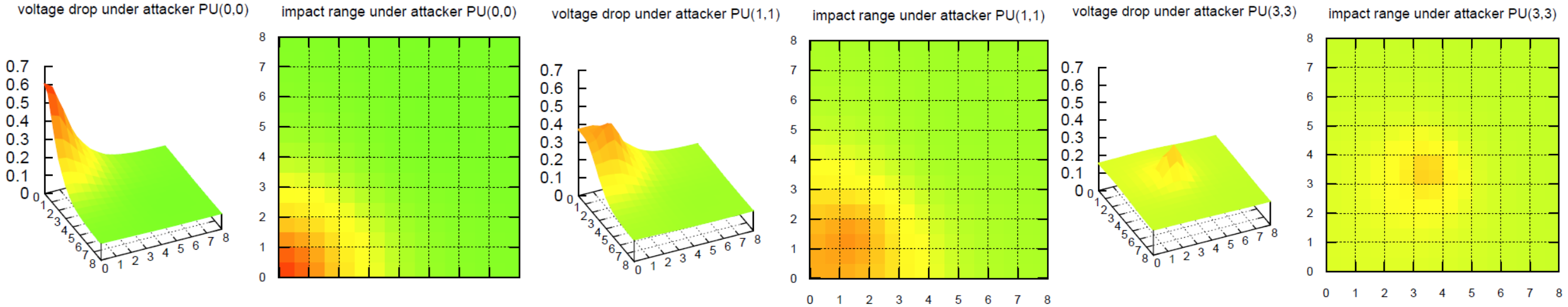


Case Study: Power/Ground Noise in 3D MPSoC

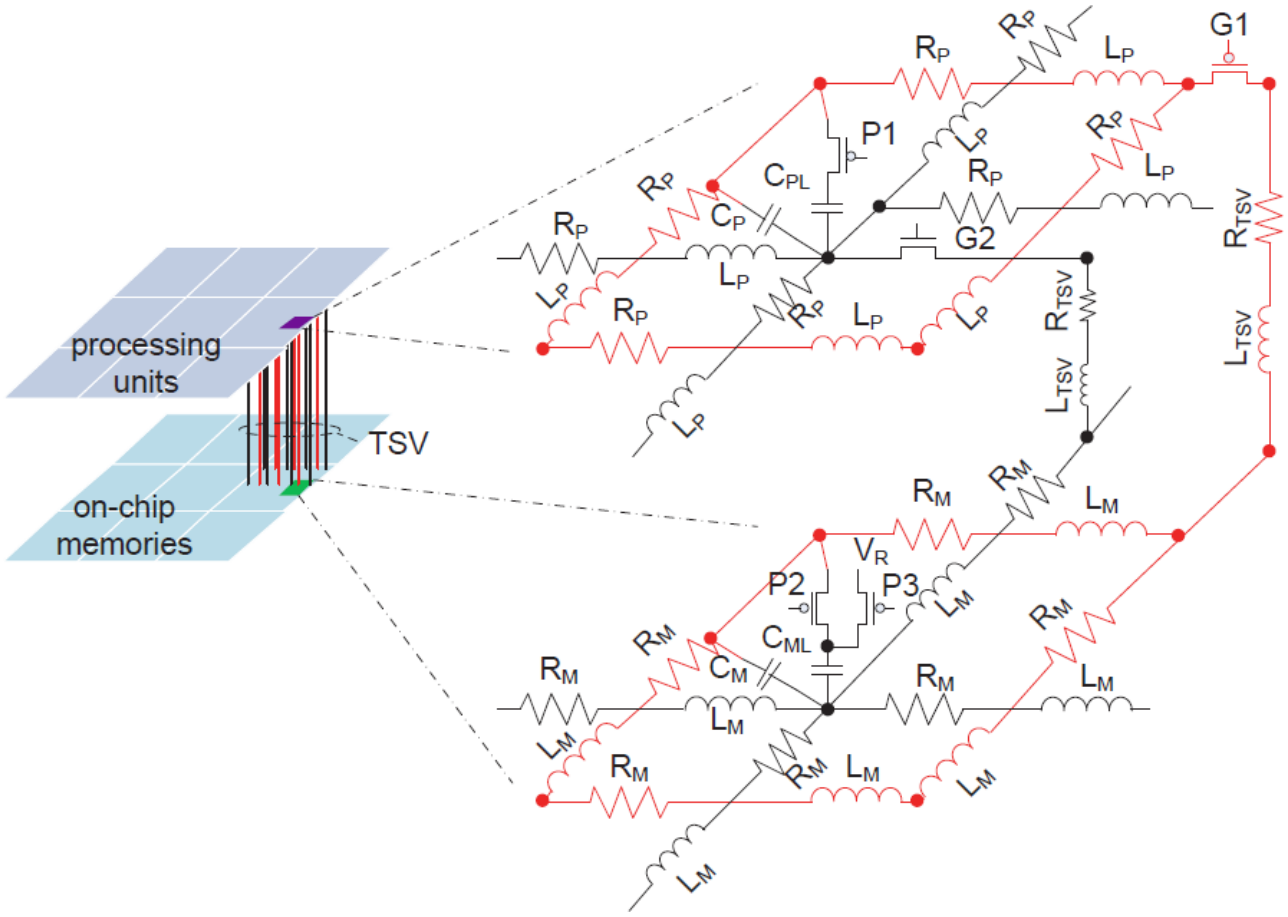
- On-chip power grid is noisy
- P/G noise is a major concern of low-power MPSoC
- SENoC is used to minimize the impacts of P/G noise caused by power gating

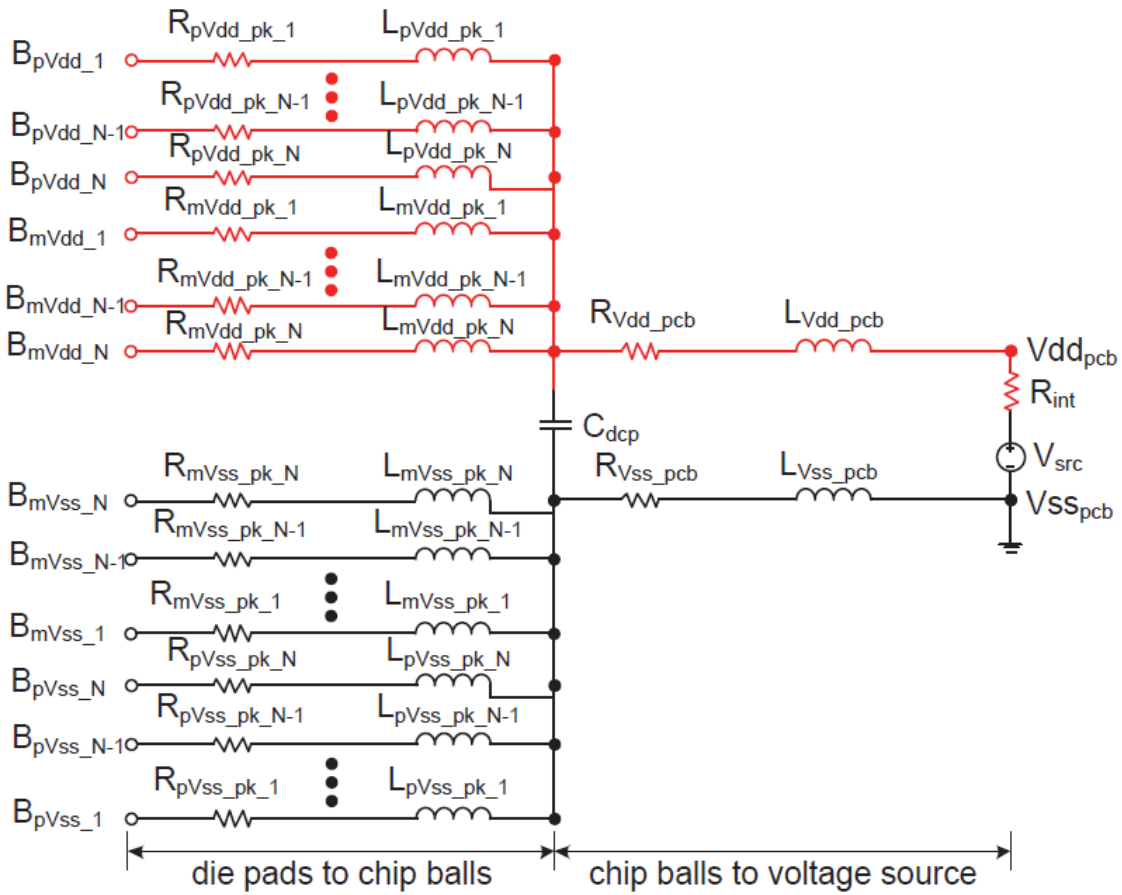




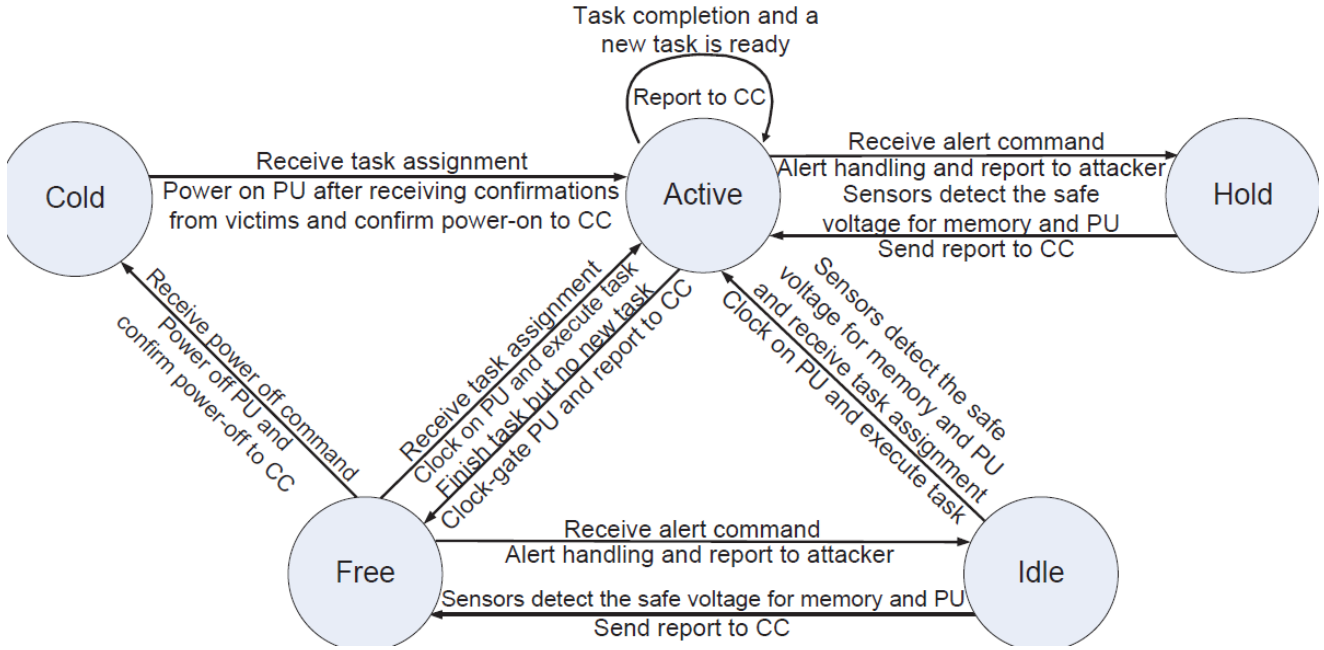


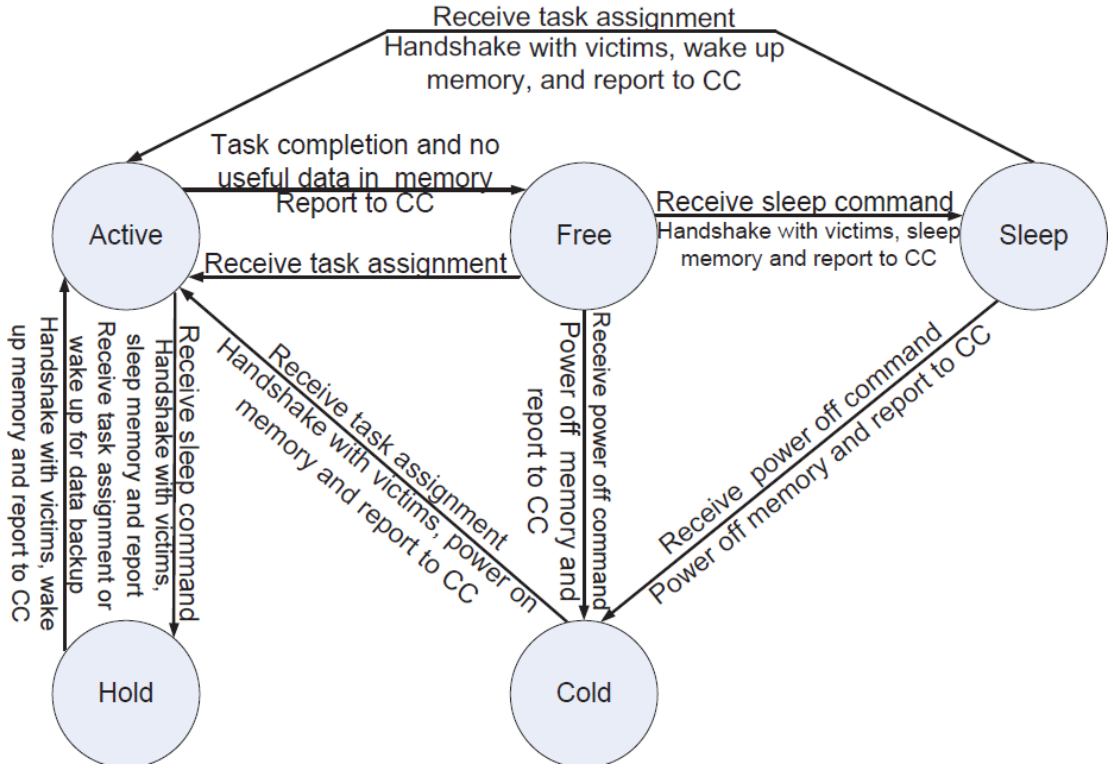
Power Grid Model



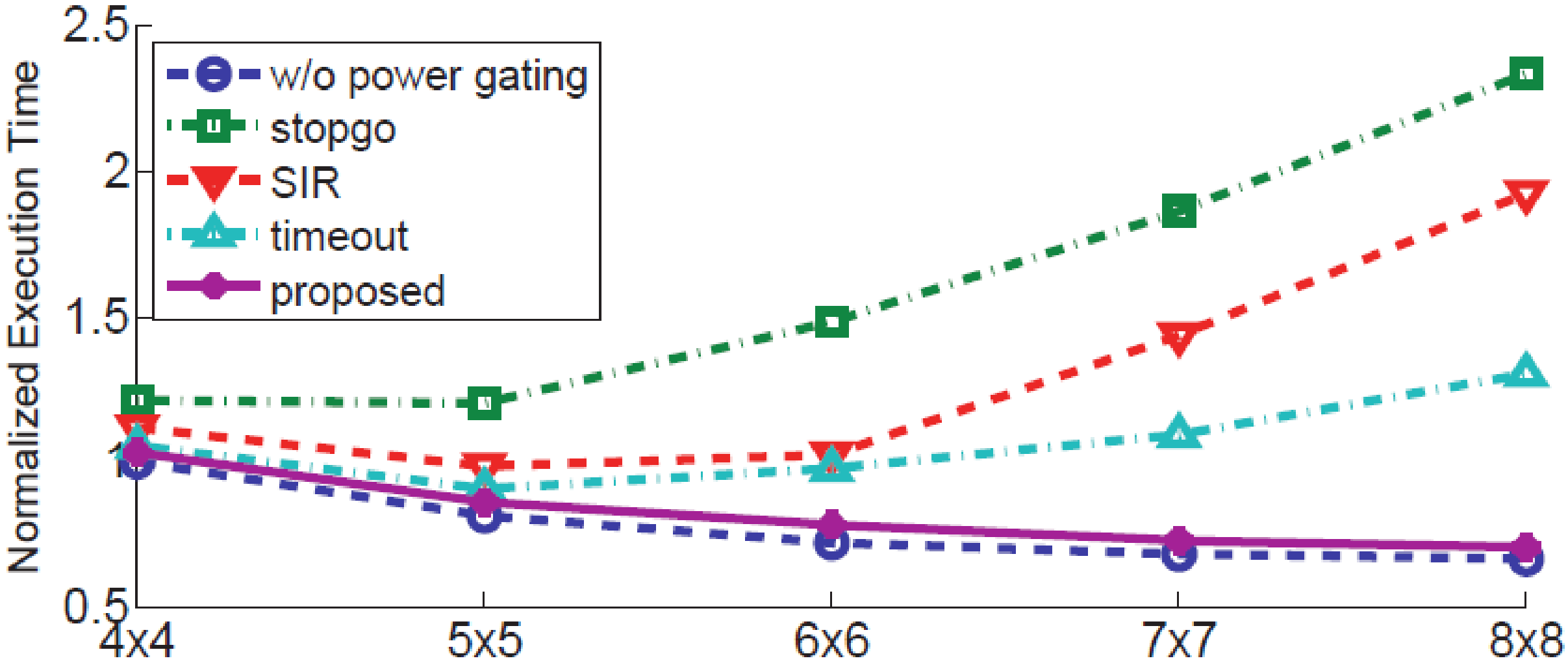


PU and Cache States

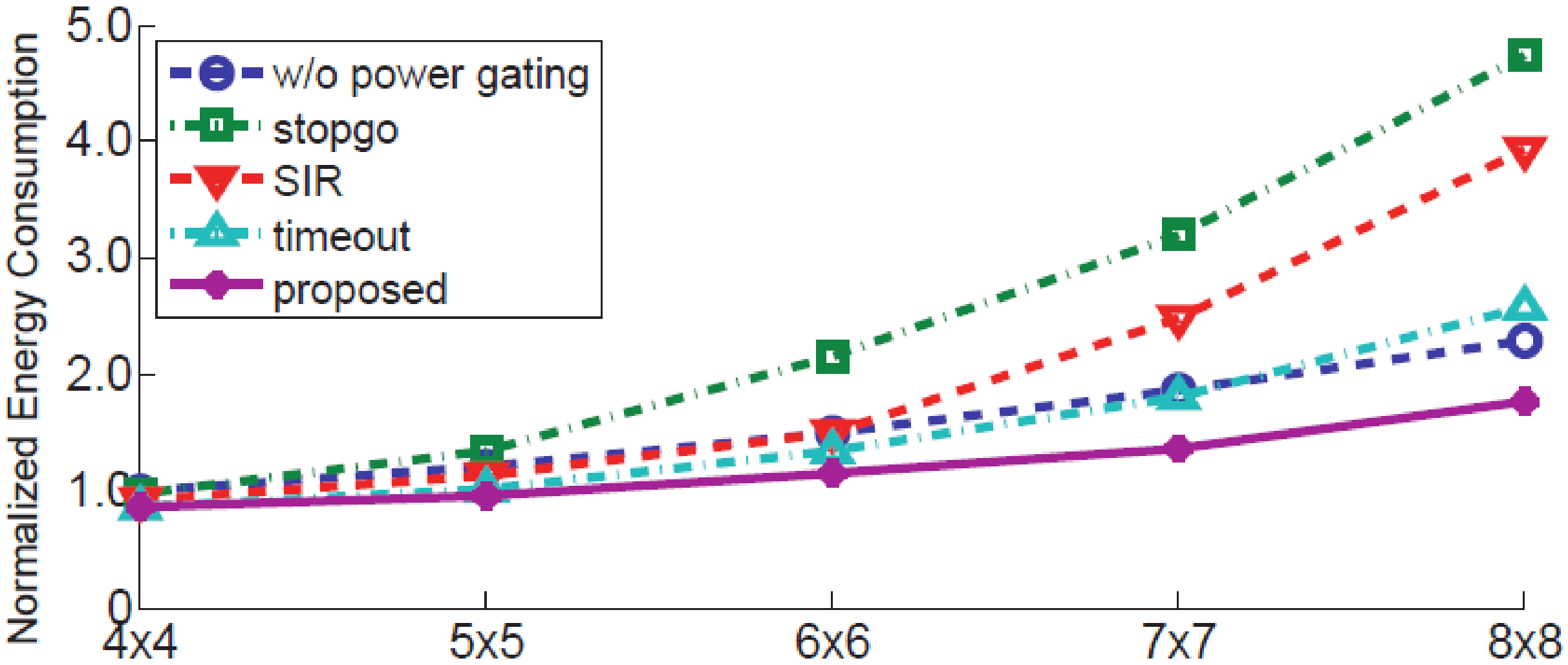




Performance and Scalability



Energy Efficiency and Scalability



Summary

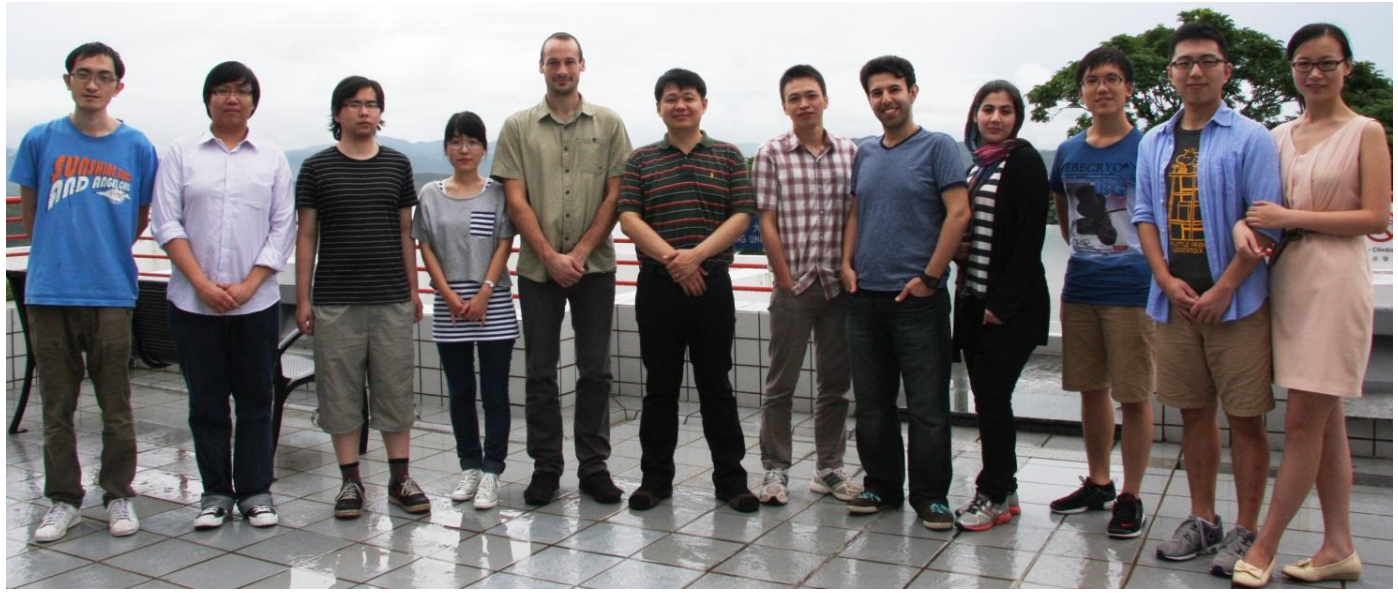
- SENoC is a systematic method to detect, report, and mitigate transient threats at run time
- HW-SW collaboration is the key for SENoC to achieve better performance with lower cost
- SSDA helps to effectively optimize performance at run time
- Two case studies show that SENoC outperforms traditional solutions

References

- S. Almukhaizim, Y. Makris, “Soft Error Mitigation through Selective Addition of Functionally Redundant Wires,” IEEE Trans. Reliability, vol. 57, no. 1, pp. 23–31, March 2008.
- K. Bhattacharya, N. Ranganathan, “RADJAM: A Novel Approach for Reduction of Soft Errors in Logic Circuits,” Int. Conf. VLSI Design, 2009.
- K. Malkowski, P. Raghavan, M. Kandemir, “Analyzing the Soft Error Resilience of Linear Solvers on Multicore Multiprocessors,” IEEE Int. Sym. Parallel Distributed Processing, 2010.
- D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, C. Das, “Exploring Fault-Tolerant Network-on-Chip Architectures,” Int. Conf. Dependable Systems and Networks, 2006.
- S. Mitra, M. Zhang, S. Waqas, N. Seifert, B. Gill, K. S. Kim, “Combinational Logic Soft Error Correction,” IEEE International Test Conference, 2006.
- N. Wang, S. Patel, “ReStore: Symptom-Based Soft Error Detection in Microprocessors,” IEEE Trans. Dependable and Secure Computing, vol. 3, no. 3, pp. 188–201, July-Sept. 2006.
- M. Nicolaidis, “Design for Soft Error Mitigation,” IEEE Trans. Device and Materials Reliability, vol. 5, no. 3, pp. 405–418, Sept. 2005.
- J. Dong, L. Zhang, Y. Han, G. Yan, X. Li, “Variation-Aware Scheduling for Chip Multiprocessors with Thread Level Redundancy,” IEEE Pacific Rim Int. Sym. Dependable Computing, 2009.
- G. Manimaran, C. Murthy, “A Fault-Tolerant Dynamic Scheduling Algorithm for Multiprocessor Real-Time Systems and its Analysis,” IEEE Trans. TPDS, vol. 9, no. 11, pp. 1137–1152, Nov. 1998.
- S. Ghosh, R. Melhem, and D. Mossé, “Fault-tolerance through scheduling of aperiodic tasks in hard real-time multiprocessor systems,” IEEE Trans. Parallel Distrib. Syst., vol. 8, no. 3, pp. 272–284, 1997.
- M. Rebaudengo, M. Sonza Reorda, et al, “Soft-Error Detection through Software Fault-Tolerance Techniques,” Int. Sym. Defect and Fault Tolerance in VLSI Systems, 1999.
- G. Reis, J. Chang, N. Vachharajani, R. Rangan, D. August, “Swift: Software Implemented Fault Tolerance,” Int. Sym. Code Generation and Optimization, 2005.
- Yu Wang, Jiang Xu, Yan Xu, Weichen Liu, Huazhong Yang, “Power Gating Aware Task Scheduling in MPSoC”, IEEE Transactions on Very Large Scale Integration Systems, vol. 19, no. 10, October 2011.
- Weichen Liu, Zonghua Gu, Jiang Xu, Xiaowen Wu, Yaoyao Ye, “Satisfiability Modulo Graph Theory for Task Mapping and Scheduling on Multiprocessor Systems”, IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 8, August 2011.
- Weichen Liu, Jiang Xu, Jogesh Muppala, Wei Zhang, Xiaowen Wu, Yaoyao Ye, “Coroutine-based Synthesis of Efficient Embedded Software from SystemC Models”, IEEE Embedded Systems Letters, vol.3, no.1, March. 2011.
- Weichen Liu, Jiang Xu, Xiaowen Wu, Yaoyao Ye, Xuan Wang, Wei Zhang, Mahdi Nikdast, Zhehui Wang, “A NoC Traffic Suite Based on Real Applications”, in Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI), July 2011.
- Weichen Liu, Jiang Xu, Xuan Wang, Yu Wang, Wei Zhang, Yaoyao Ye, Xiaowen Wu, Mahdi Nikdast, Zhehui Wang, “A Hardware-Software Collaborated Method for Soft-Error Tolerant MPSoC”, in Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI), July 2011.
- Yu Wang, Jiang Xu, Shengxi Huang, Weichen Liu, Huazhong Yang, “A Case Study of On-Chip Sensor Network in Multiprocessor System-on-Chip”, International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES), 2009.

Acknowledgement

- Current group members
  - Xiaowen Wu, Xuan Wang, Zhe Wang, Zhehui Wang, Mahdi Nikdast, Duong Huu Kinh Luan, Peng Yang, Haoran Li, Rafael Kioji Vivas Maeda, Zhifei Wang
- Past members and visitors
  - Yaoyao Ye, Weichen Liu, Xing Wen, Kwai Hung Mo, Yu Wang, Sébastien Le Beux, Yiyuan Xie, Huaxi Gu



UGC 大學教育資助委員會 University Grants Committee

intel

XILINX