Architecture and Design Methodology for Autonomic System-on-Chip (ASoC)

A. Bernauer, W. Rosenstiel
Wilhelm-Schickard Institute for Informatics
Computer Engineering
University of Tübingen

O. Bringmann,
Forschungszentrum Informatik
Karlsruhe

A. Bouajila, J. Zeppenfeld,
W. Stechele, A. Herkersdorf
Institute for Integrated Systems
Munich University of Technology

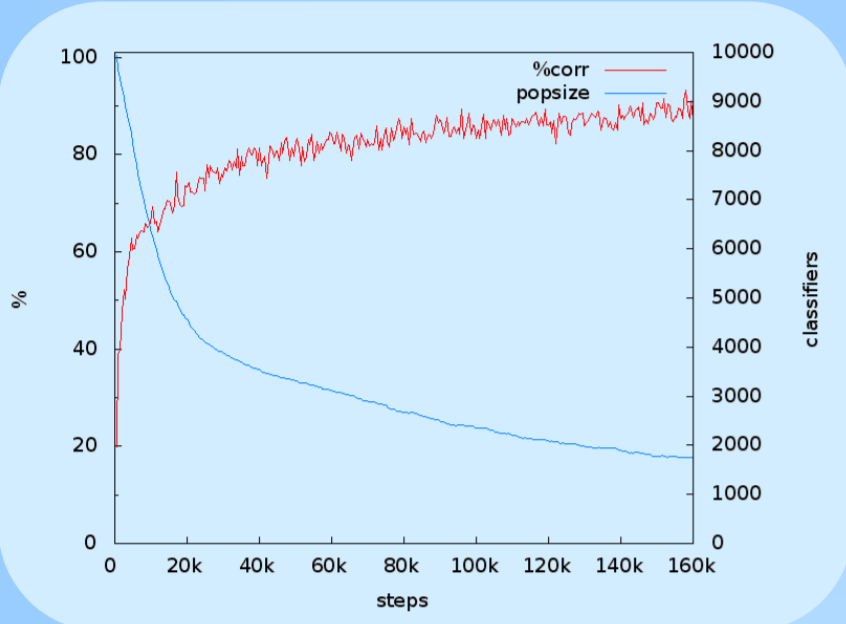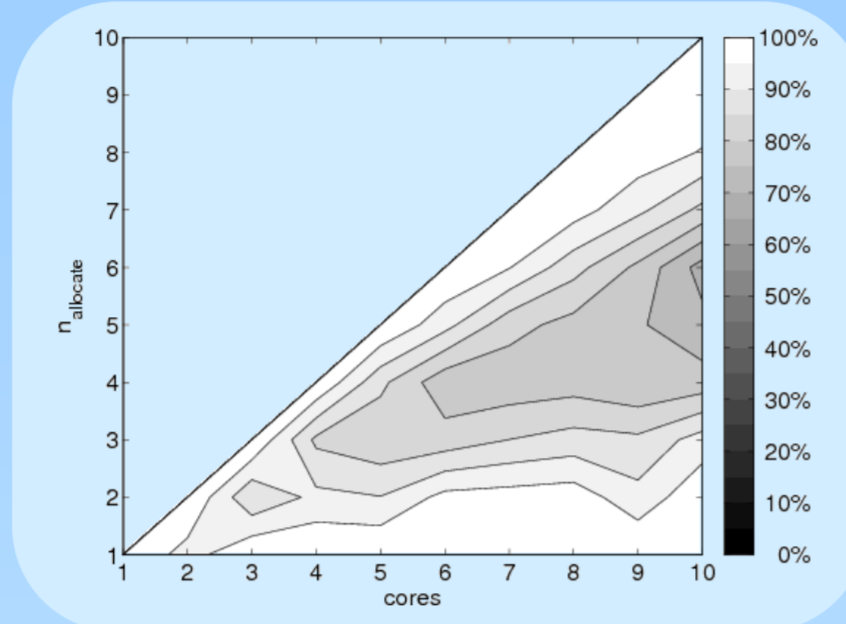asoc@informatik.uni-tuebingen.de

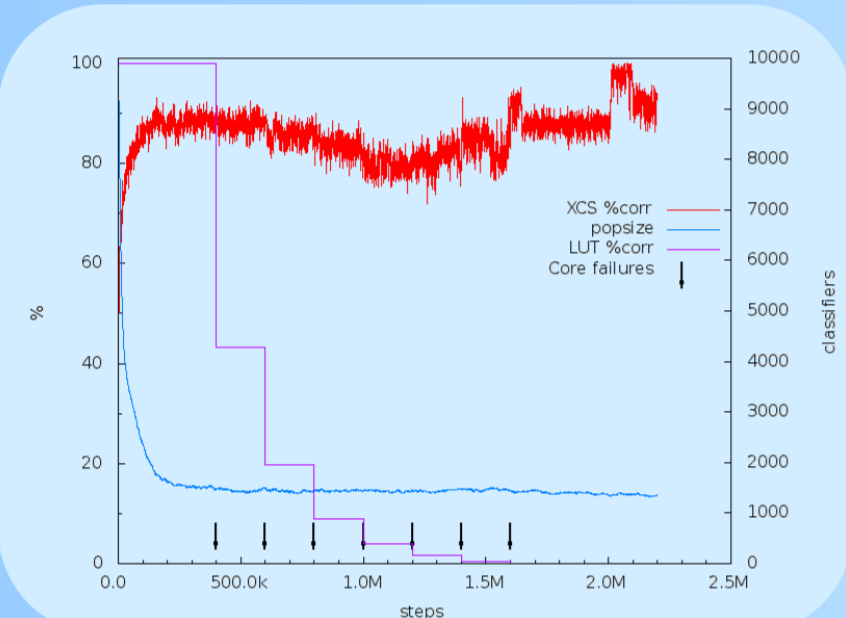## Reducing Design Effort

XCS helps reduce design effort
➢ Solves problem of task allocation
➢ Adapts to unexpected event of core failure
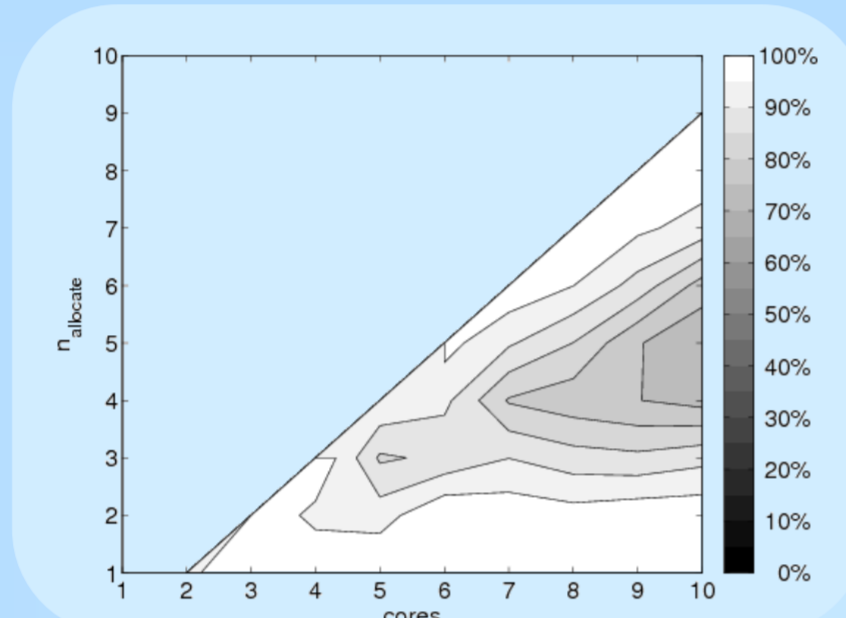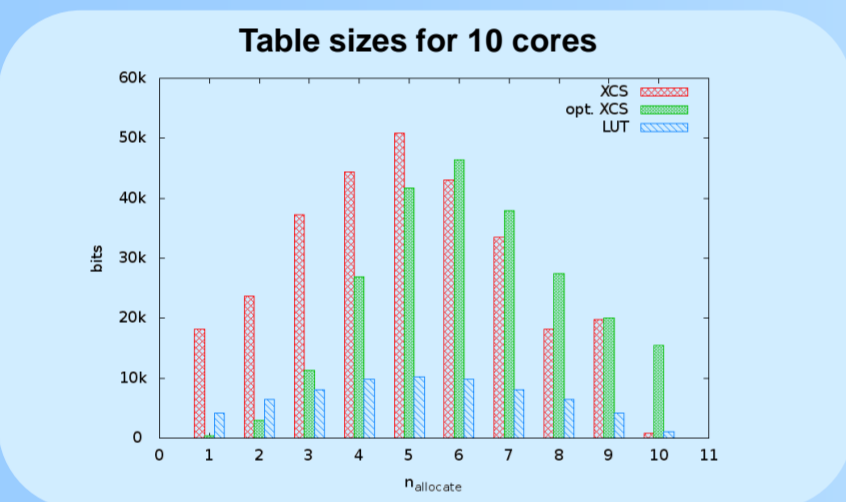➢ Manageable table sizes

Learning (10,3)-task-allocation

Rate of valid task allocation

Adapting to core failure;
comparing to lookup table (LUT)

Rate of valid task allocations after
1 core has failed

Table sizes for 10 cores

A. Bernauer, O. Bringmann, W. Rosenstiel; *Generic Self-Adaptation to Reduce Design Effort for System-on-Chip*, p. 126-135, IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO), p.126-135, San Francisco, 2009.

Design and Run Time Learning
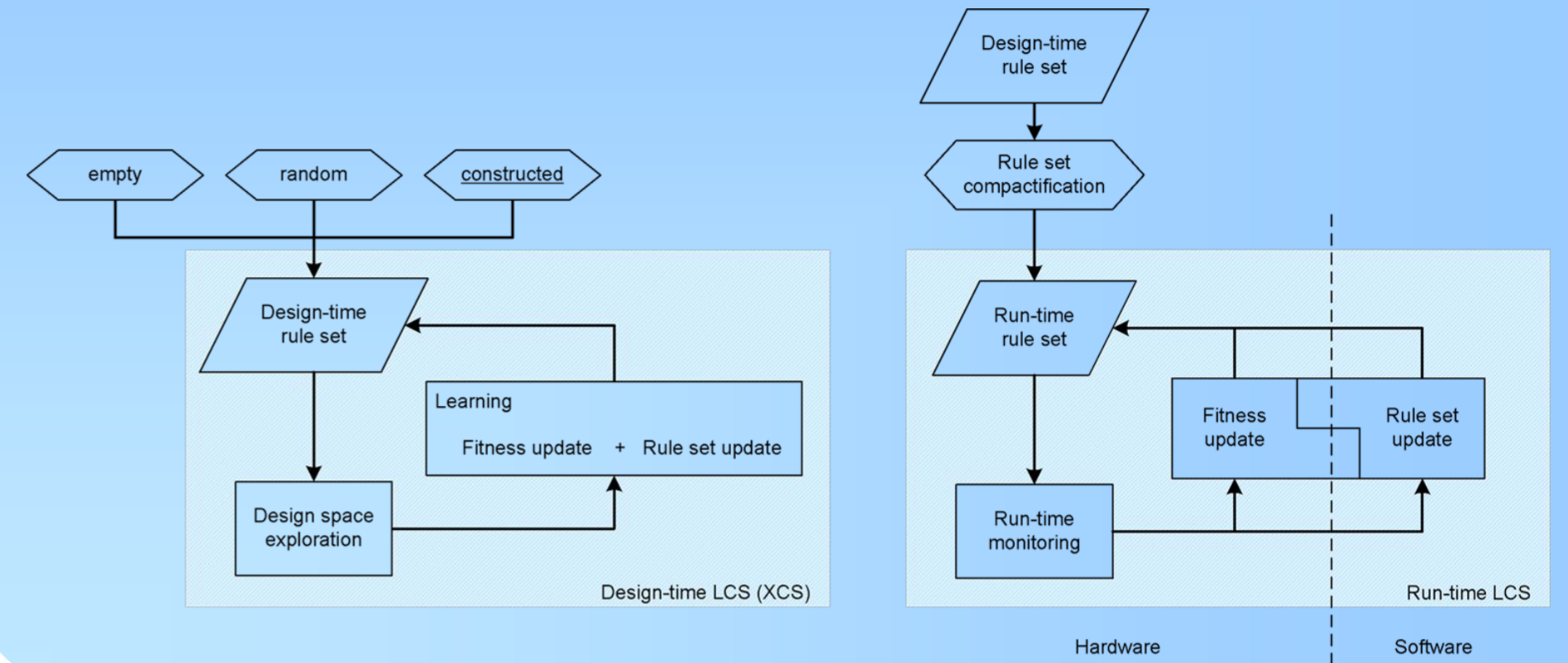
ASoC FPGA Prototype

SoC with autonomic capabilities on Xilinx FPGA board

➢ Autonomic and functional layer
➢ Autonomic Element Interconnect (AEI): connect different AE elements in order to provide system-wide optimizations
➢ Functional IPs augmented with autonomic elements (incl. LCTs)
➢ Functional layer: MPSoC based on multiple Leon3 processors
➢ LCT evaluator: 51 slices
➢ AEI interface: 260 slices (64 bit payload)
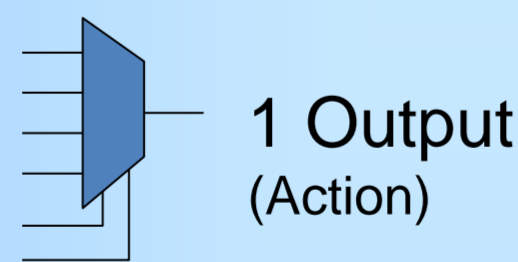➢ Round trip latency: 4–328 cycles with 4 AE nodes
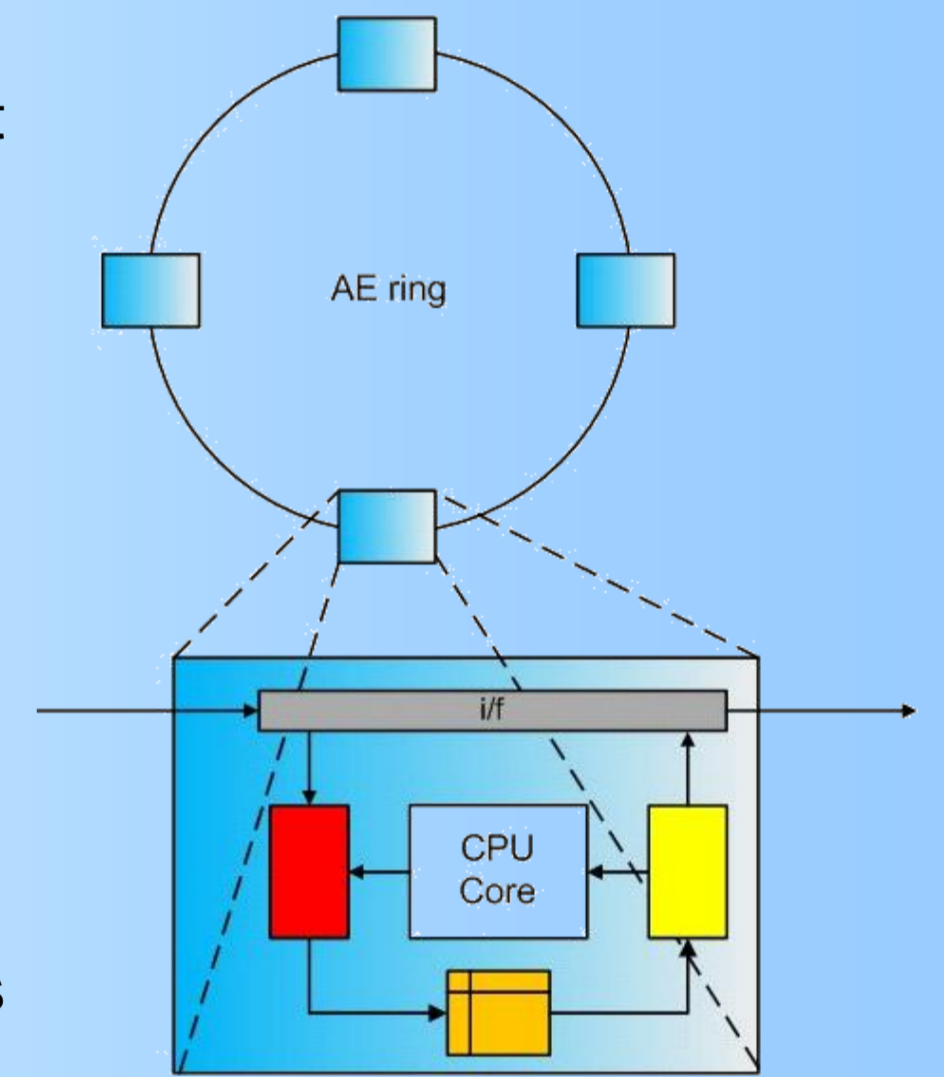
J. Zeppenfeld, A. Herkersdorf; *Autonomic Workload Management for Multi-core Processor Systems*, ARCS 2010.
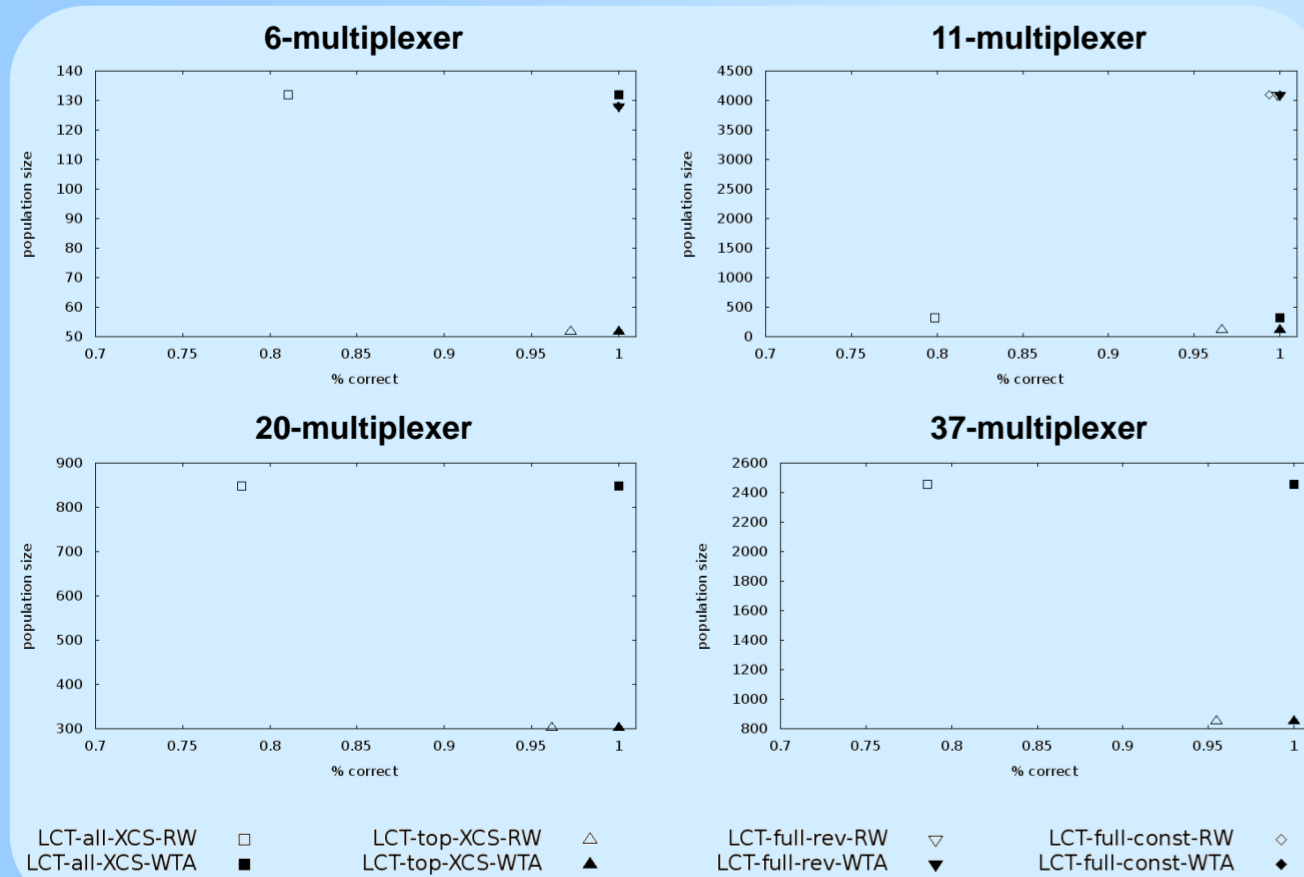
Multiplexer Benchmark

Standard benchmark for comparison with other LCS implementations
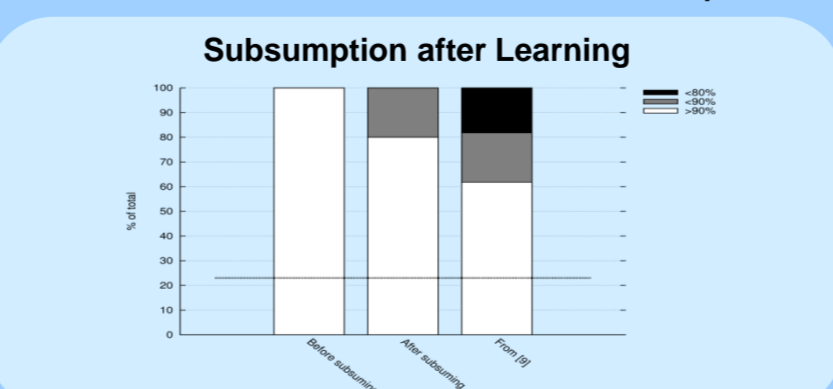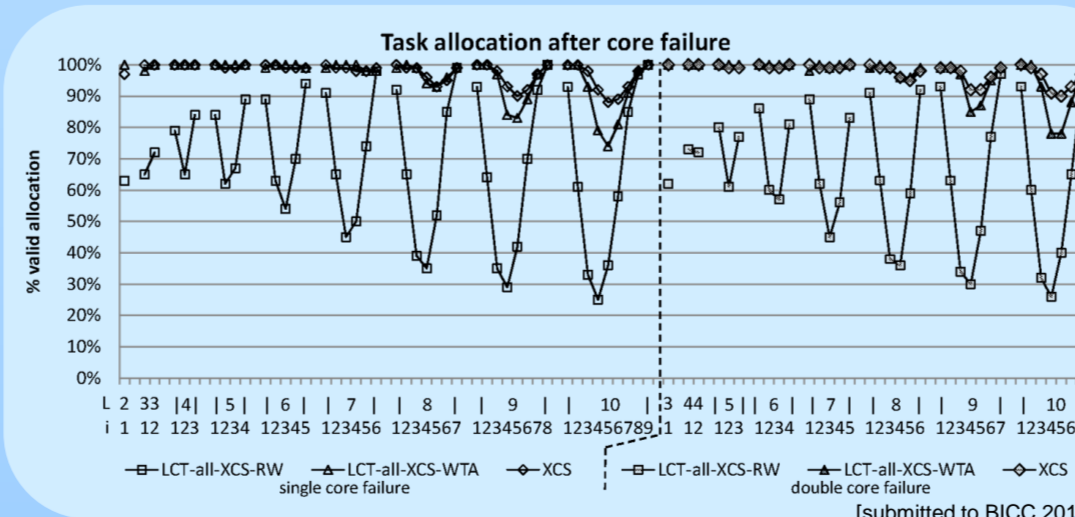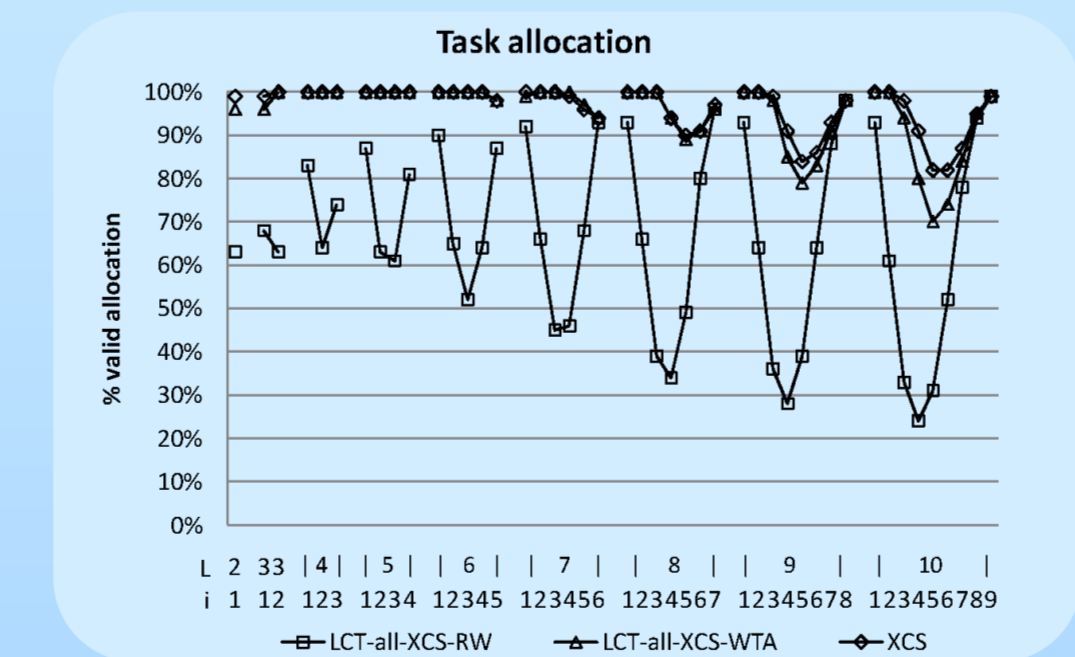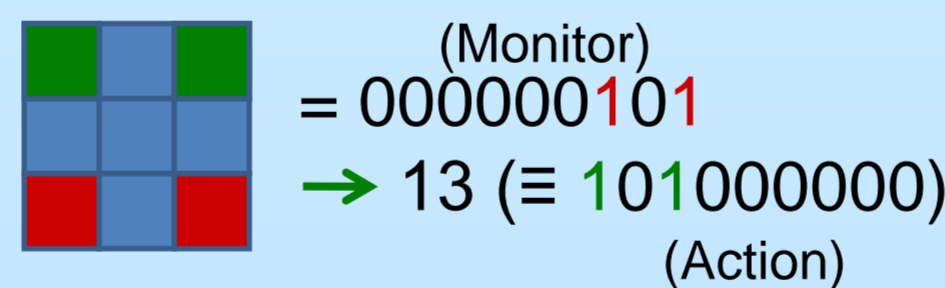
2^K + K Inputs
(Monitor)

1 Output
(Action)

6-multiplexer

11-multiplexer

20-multiplexer

37-multiplexer

LCT-all-XCS-RW   LCT-top-XCS-RW   LCT-full-rev-RW   LCT-full-const-RW
LCT-all-XCS-WTA   LCT-top-XCS-WTA   LCT-full-rev-WTA   LCT-full-const-WTA

LCT performance in $\overline{m_{[1]}}$ for insufficiently learned XCS

LCT-all-XCS-RW   LCT-full-rev-WTA
LCT-all-XCS-RW   LCT-full-rev-RW
LCT-all-XCS-WTA   LCT-full-const-RW
LCT-top-XCS-RW   LCT-full-const-RW
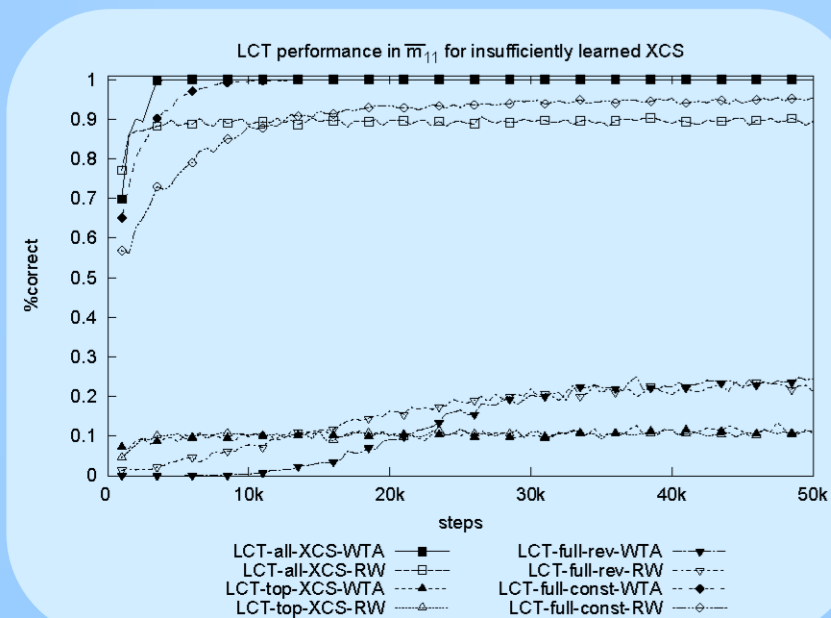
A. Bernauer, J. Zeppenfeld, O. Bringmann, A. Herkersdorf, W. Rosenstiel; *Combining software and hardware LCS for lightweight on-chip learning*, submitted to Biologically-Inspired Collaborative Computing (BICC 2010).

Task Allocation

Finding a valid task allocation over multiple cores

(Monitor)
= 000000101
➜ 13 (≡ 101000000)
(Action)

Task allocation

LCT-all-XCS-RW   LCT-all-XCS-WTA   XCS

Task allocation after core failure

LCT-all-XCS-RW   LCT-all-XCS-WTA   XCS   LCT-all-XCS-RW   LCT-all-XCS-WTA   XCS
single core failure   double core failure
[submitted to BICC 2010]

Subsumption after Learning

B. Rakitsch, A. Bernauer, O. Bringmann, W. Rosenstiel; *Pruning population size in XCS for complex problems*, submitted to IEEE World Congress on Computational Intelligence (WCCI 2010), Organic Computing Session.

Core Parameterization

Adjusting frequency to maximize utilization

Reward improvement through learning

Learning to cope with a faulty component
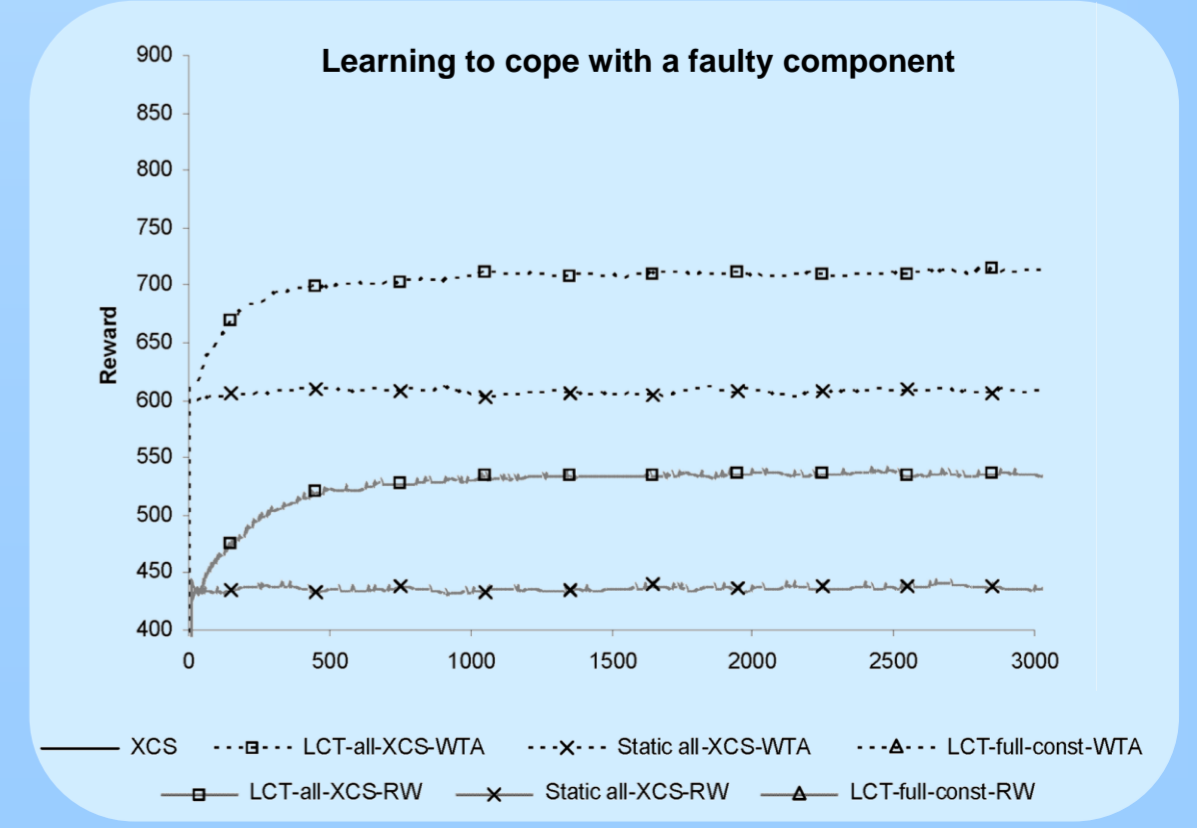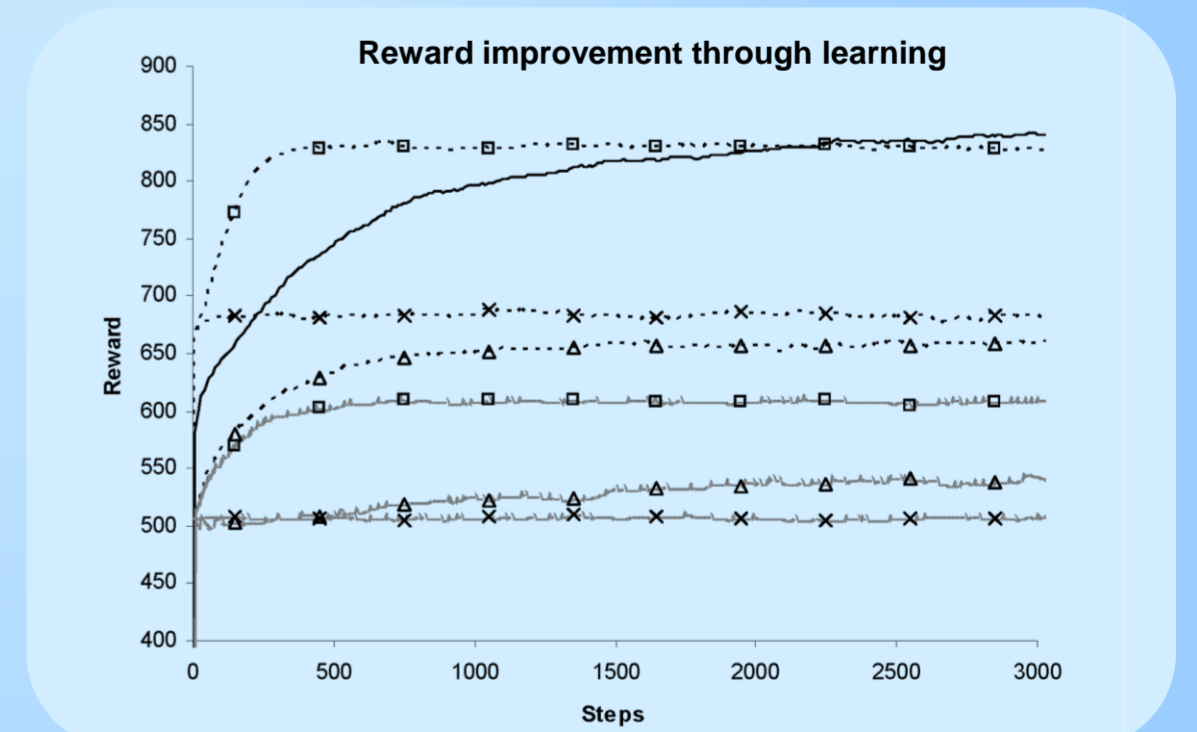
XCS   LCT-all-XCS-WTA   Static all-XCS-WTA   LCT-full-const-WTA
LCT-all-XCS-RW   Static all-XCS-RW   LCT-full-const-RW

[submitted to BICC 2010]