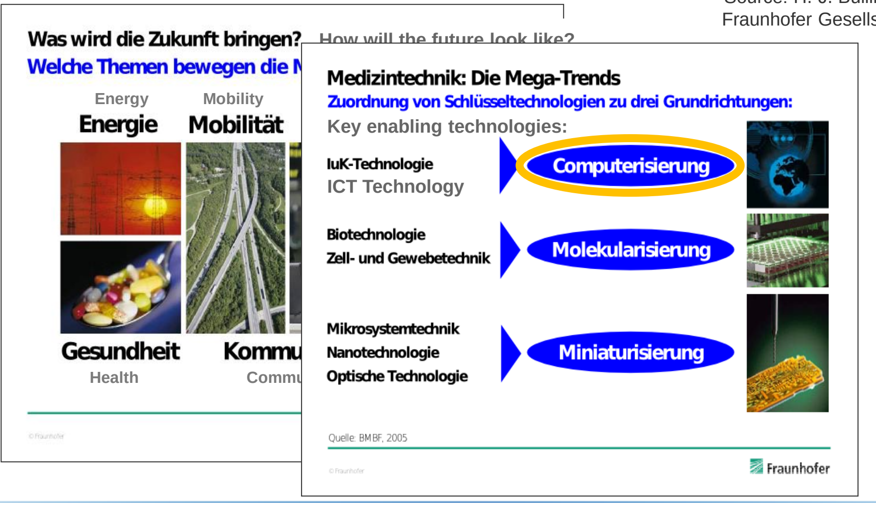
MPSoC Forum 2014
Margeaux, France, July 10, 2014

Conquering MPSoC Design and Architecture Complexity with Bio-Inspired Self-Organization

Andreas Herkersdorf
Institute for Integrated Systems, TU München
http:/www.lis.tum.de

The System Design Complexity Challenge

Source: H.-J. Bullinger, Fraunhofer Gesellschaft



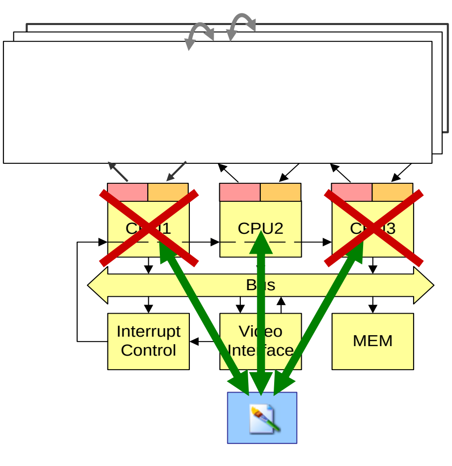
The System Design Complexity Challenge

- Moore's law is the technological enabler for Billion MOSFET designs, but …
  - … how to deal with the design complexity of Billion MOSFET (MP)SoCs
    - considering implications of various forms of variability
    - and get the design and fabrication right at first time?
- When the MPSoC design & manufacturing challenge is solved, how to …
  - program massively parallel processor components,
  - guarantee real-time, security, mixed-criticality and power constraints?
- Progress of IC design and integration gets more and more constraint by direct or indirect complexity issues
- What alternative to established, best practice engineering approaches do we have to tackle complexity?

MPSoC Resilience

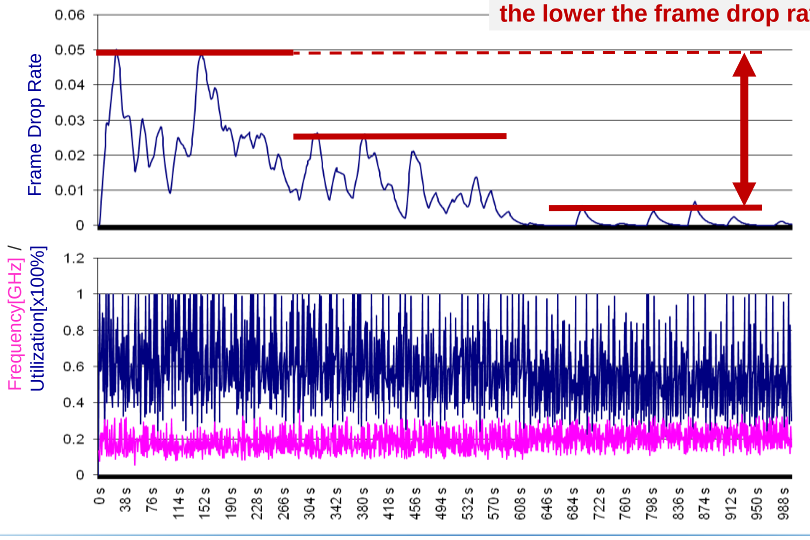
[Zeppenfeld08]

- Video frames distributed among 1 – 3 operational RISC cores
- RISC cores may arbitrarily fail … (mimicked by purposely switch-off every few 100 ms)
- … which is compensated by workload redistribution and f, V_{DD} scaling (DVFS) of remaining cores
- Heuristic by which f, V_{DD} are altered will be revealed later



MPSoC Resilience

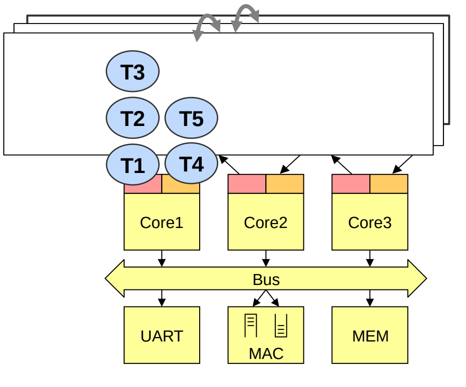
The more often we switch cores off/on, the lower the frame drop rate



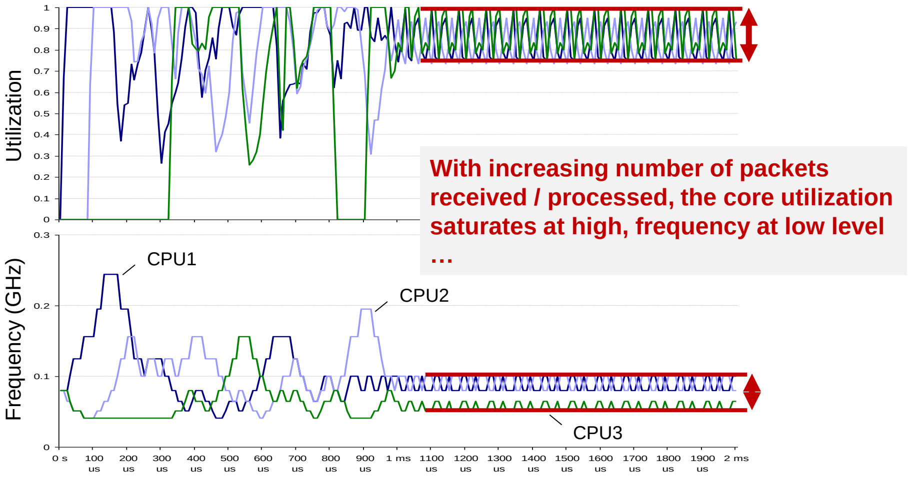
MPSoC Task Mapping / Workload Balancing

[Zeppenfeld11]

- IP packet processing is split into 5 tasks, which are executed sequentially and initially all mapped to Core 1
- IP flow received at MAC, sent to T1 and received from T5 for retransmit
- Cores may issue task relocation when in high-load condition and perform DVFS
- Same heuristic and method applied as in previous example of resilient video applications

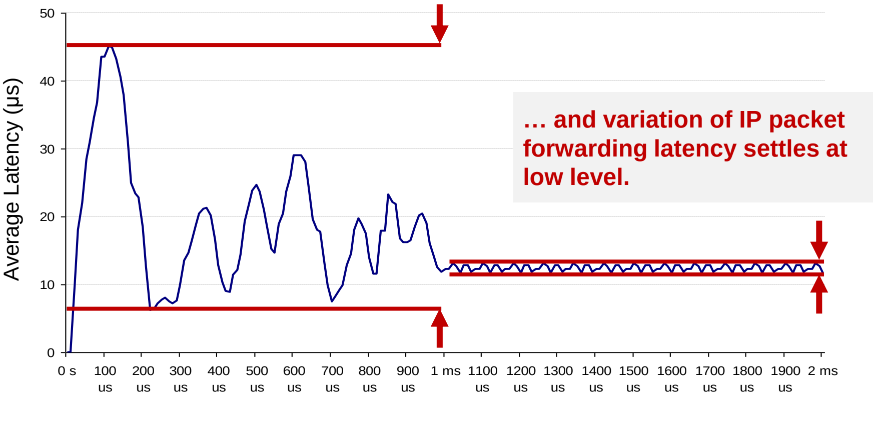


MPSoC Task Mapping / Workload Balancing



Technische Universität München

MPSoC Task Mapping / Workload Balancing



„Manycore“ System in Nature …

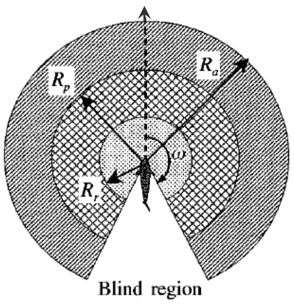
- Fish school does not seam to have a complexity or reliability problem
- Entire system behaves orderly and defends itself reliably against predators
- How is orchestration of “fish school” facilitated among its components?

Nature designs and optimizes differently …

[Inada02, Pathel04]

Every system constituent (fish) follows a local rule set on how to behave under stress

- Rule set is simple, easily manageable for each constituent
- Every constituent follows same rule set
- Global system behavior not necessarily reflected in local rule set



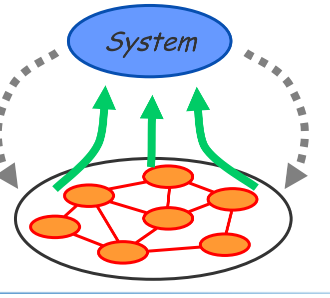
If (d < R_r) then
Repulsion: Avoid clashes
Else if (d < R_p) then
Orientation: Parallel to neighbor
Else if (d < R_a)
Attraction: Approach companion
Endif

Self-Organization / Emergence

[Fromm04]

Local behavior of the constituents of a self-organizing system may lead to observable, emergent global behavior which is not reflected in local behavior / rules

- Population of interacting system constituents
- System is hierarchically structured (multi-layer organization)
- Emergent behavior observable at levels above constituent level (system level or system environment) as a result of hidden causal relationships across levels

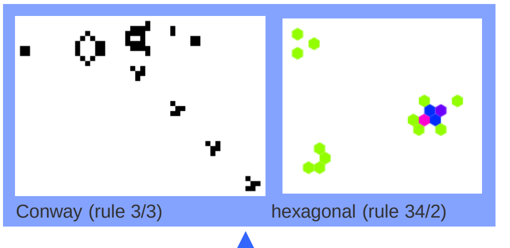


Conway's Game of Life

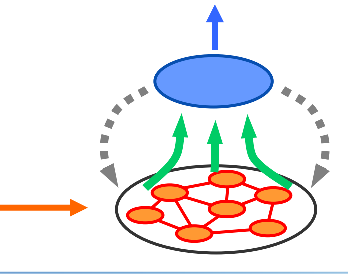
[Gardner70]

Constituent pattern determines system level behavior:

- Rotary, translatory movements, oscillation, persistence, …
- … in combination and with varying parameters



If (cell alive AND N = 3) then
   live unchanged to next generation
Else if (cell alive AND N < 2 OR N > 3) then
   death by loneliness or overcrowding
Else if (cell dead AND N = 3) then
   birth of new cell in next generation
End



ASoC: Autonomic System on Chip

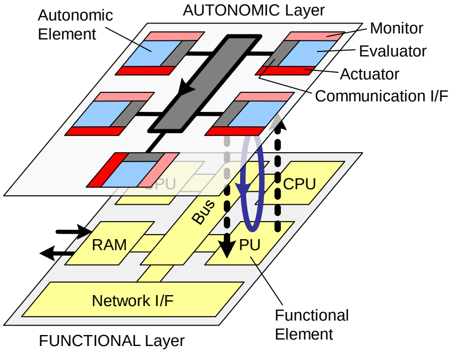
[Herkersdorf/Rosenstiel04]

Evolutionary, platform-centric approach with compatibility to SoC design method:

- Functional layer containing conventional IPs
- Autonomic layer extends IPs with self-x properties
  - Improved Reliability
  - Performance / Power optimization at runtime

Dedicate part of chip capacity to self-x abilities at autonomic layer

Future SoC shall have the ability to learn to live with environmentally imposed variations or work around defects autonomously

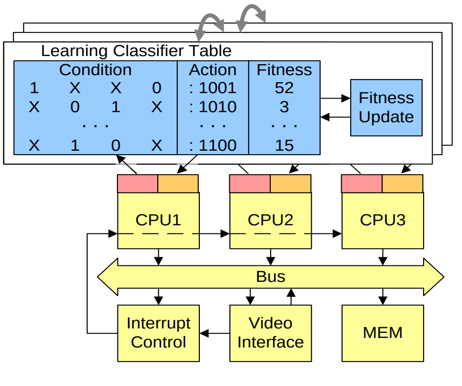


Joint project with University of Tübingen and FZI Karlsruhe

ASoC: Self-Optimization through Runtime Learning

[Zeppenfeld11, Holland78, Wilson95, Butz06]

- **Monitors**
  - Error rate, (Temperature)
  - Frequency
  - Utilization
  - Workload
- **Actuators**
  - Frequency (and voltage)
  - Task migration
- **Evaluator**
  - Learning classifier system adapted for efficient HW implementation
  - Reinforcement learning through fitness update of individual rules
- **Communicator**
  - Sharing of global information



ASoC: Self-Optimization through Runtime Learning

[Zeppenfeld11]

- Fitness-driven rule and operation parameter selection per core
- Objective function to assess system-wide impact of rule / action selection:

$\delta_{Load} = |f_{cpu\ n} \cdot util_{cpu\ n} - f_{cpu\ avg} \cdot util_{cpu\ avg}|$

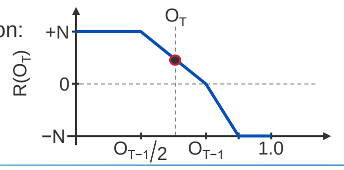
$\delta_{Util} = util_{target} - util_{cpu\ n}$

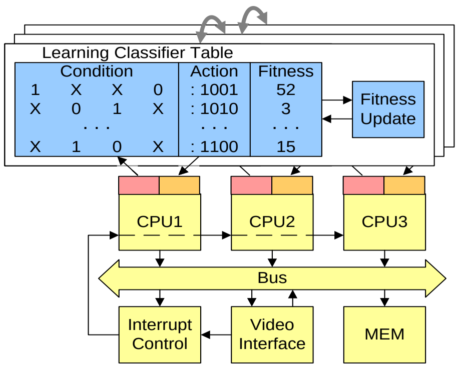
$\delta_{Freq} = f_{cpu\ n}$

$O_{CPU} = w_1 \cdot \delta_{Load} + w_2 \cdot \delta_{Util} + w_3 \cdot \delta_{Freq}$

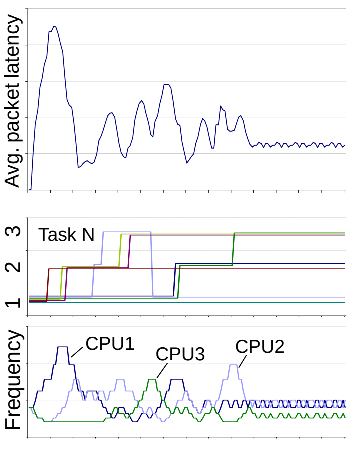
$O_{Sys} = w_a \cdot O_{CPU1} + w_b \cdot O_{CPU2} + \ldots$

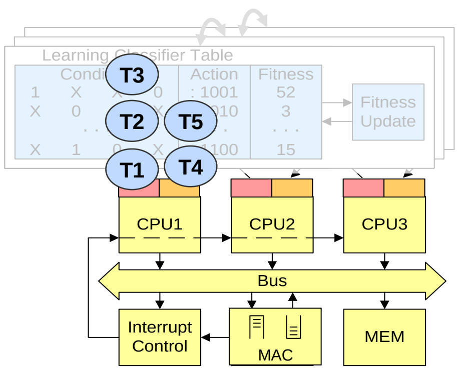
- Reward function:



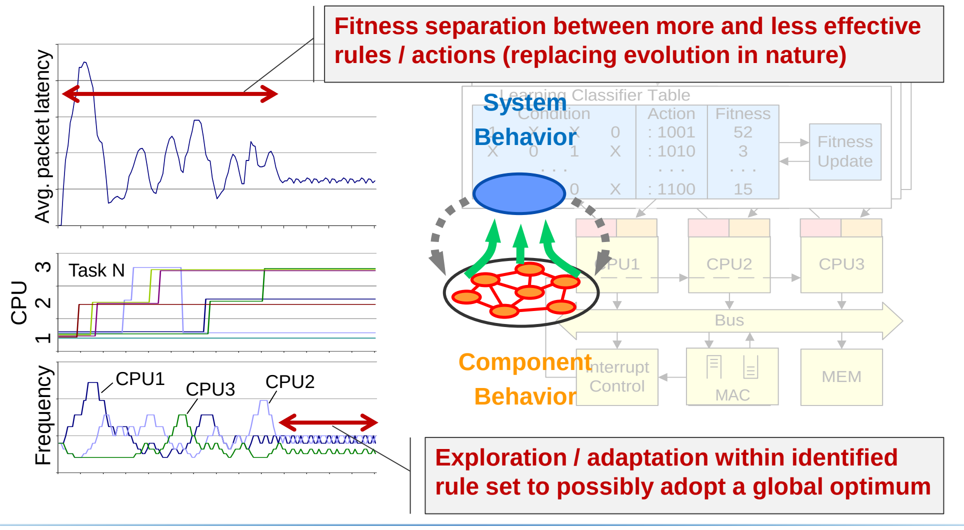


ASoC: Self-Optimization through Runtime Learning

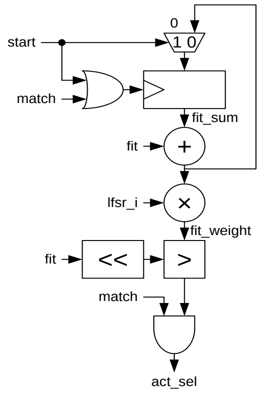




ASoC: Self-Optimization through Runtime Learning



LCT / AE Implementierung

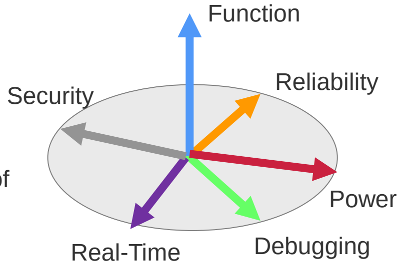


	Flip-Flops	LUTs	BRAMs	Mult.	Overhead
Leon3	1749	8936	28	1	–
Leon3 AE	2122	10213	29	2	14.3%
LCT	*66*	*116*	*1*	*1*	*1.4%*
Act Task.	*57*	*299*	*0*	*0*	*3.5%*
Act Freq.	*7*	*19*	*0*	*0*	*0.2%*
Mon Util.	*35*	*74*	*0*	*0*	*0.8%*
Mon Load	*20*	*40*	*0*	*0*	*0.5%*
AE IF	*173*	*399*	*0*	*0*	*4.5%*

Synthesis results for Xilinx Virtex 4 VLX100

Conclusions

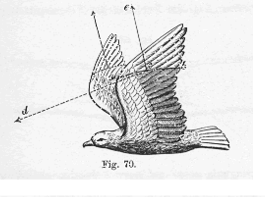
- Complexity of multi-objective hardware/software systems demands push of classical design time tasks into field operation
  - Complexity either increases through technology-enabled increases in function integration or consideration of pressing non-functional aspects
    - Security, resilience, power dissipation, test & debug
- Nature offers scalable, surprisingly simple self-organization paradigms to control complex systems
- Suggestion to study and apply emergent behaviors on broader scope for optimization of non-function aspects of MPSoC and distributed Embedded Systems / CPS

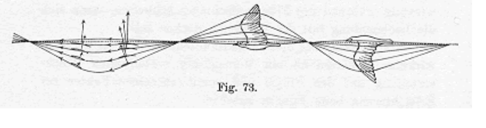


- Be inspired by nature, but also be aware that …

… copying nature 1-on-1 not necessarily yields success!

„Kleiner Schlagflügelapparat“, 1893





„Der Vogelflug als Grundlage der Fliegekunst“, O. Lilienthal, 1889

Source: Fotoarchiv Otto-Lilienthal-Museum, 1997

Thanks!

«All arts rely on mimicking nature.»

Seneca (1 – 65 AD), roman philosopher, scientist, politician

I particularly would like to thank our ASoC partners from University of Tübingen (W. Rosenstiel, A. Bernauer), the VirTherm-3D team from KIT Karlsruhe (J. Henkel, T. Ebi) and the team at TU Munich (A. Bouajila, H. Rauchfuss, W. Stechele, T. Wild, J. Zeppenfeld).

Furthermore, sincere thanks deserve the DFG and the BMBF for supporting our work as part of the SPP1183 (Organic Computing), SPP1500 (Dependable Embedded Systems) and the Clusterforschungsprogramm AIS "Autonome Integrierte Systeme"

References

[Butz06] Butz, M. V.: "Rule-based Evolutionary Online Learning Systems – A Principled approach to LCS Analysis and Design", Springer, ISBN 978-3-540-25379-2, 2006.

[Fromm04] Fromm, J.; "The Emergence of Complexity", Kassel university press, ISBN 3-89958-069-9, 2004, URN: urn:nbn:de:0002-069.

[Herkersdorf04] Herkersdorf, A., Rosenstiel, W., "Towards a Framework and Design Methododology for Autonomous SoC", GI Workshop on Organic Computing, Ulm, Germany, September 2004

[Holland78] Holland, J.; Reitman, J.: Cognitive systems based on adaptive algorithms. In Pattern-directed inference systems, Academic Press, New York, 1978.

[Inada02] Patel, J. "Characterization of Soft Errors Caused by Single Event Upsets in CMOS Processes". IEEE Trans. Dependable Secur. Comput. 1, 2 (Apr. 2004), 128-143

[Patel04] Patel, J. "Characterization of Soft Errors Caused by Single Event Upsets in CMOS Processes". IEEE Trans. Dependable Secur. Comput. 1, 2 (Apr. 2004), 128-143

References

[Wilson95] S. Wilson, “Classifier fitness based on accuracy”, Evolutionary Computation, 3, 1995, pp. 149-175

[Zeppenfeld08] Zeppenfeld J.; et. al.: “Learning Classifier Tables for Autonomic Systems on Chip”, Lecture Notes in Informatics vol. 134, pp.771–778, Springer; GI Annnual Meeting, Munich, September 12, 2008

[Zeppenfeld11] Zeppenfeld J.; Herkersdorf A.; "Applying Autonomic Principles for Workload Management in Multi-Core Systems on Chip", International Conference on Autonomic Computing, ICAC, Karlsruhe, Germany, June 14-18, 2011