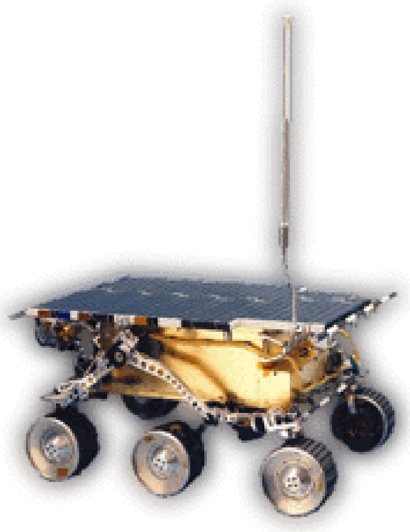
Multi-Objective Intrinsic Evolution of Embedded Systems (MOVES)

Paul Kaufmann, Marco Platzner
Computer Engineering Group
University of Paderborn
{paul.kaufmann, platzner}@uni-paderborn.de

Motivation / Vision

- investigate intrinsic hardware evolution as a mechanism to achieve self-adaptation and –optimization for autonomous embedded systems

- an embedded system ...
  - adapts to slow changes by simulated evolution
    - typically, change of environment
  - adapts to radical changes by switching to pre-evolved alternatives
    - typically, change in computational resources
  - requires intrinsic evolution for autonomous operation



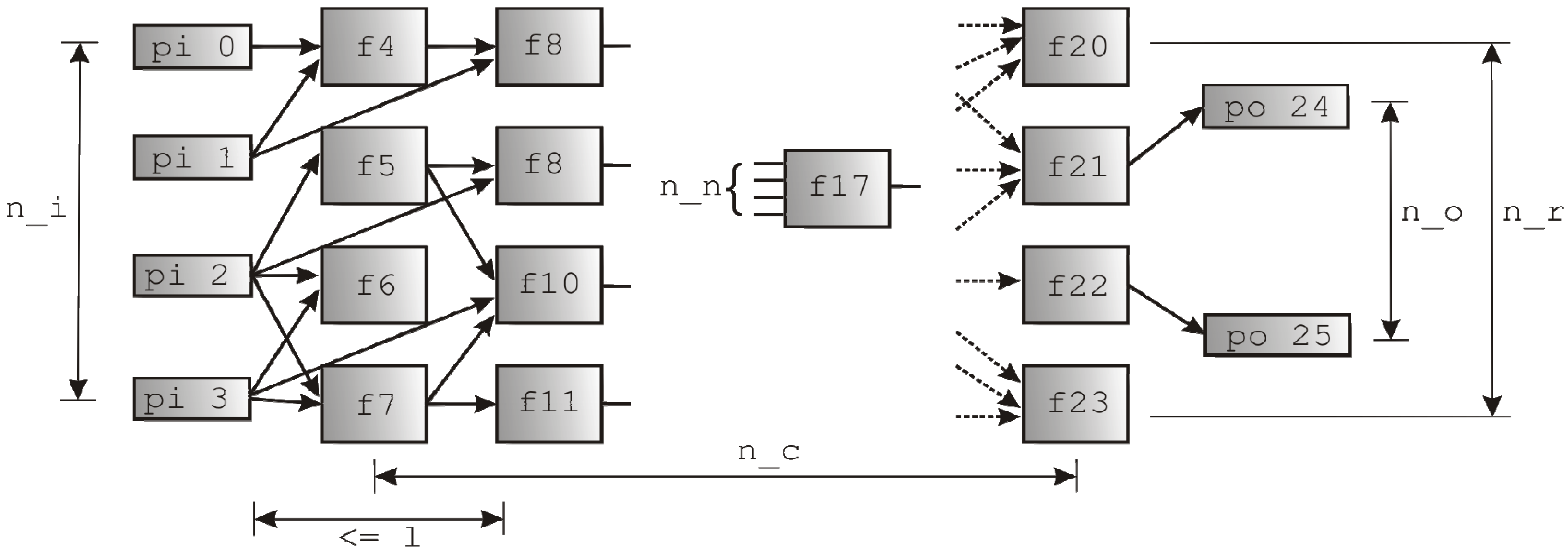
Overview

- topics of phase I
  1. models and algorithms
     - representation models for digital logic, multi-objective evolutionary optimization algorithms, tools for the evaluation of models and algorithms
  2. system-on-chip architecture
     - platform FPGA, hw/sw partitioning
  3. case studies and evaluation
     - test problems, classificator for electromyography (EMG) signals

- current and future topics (phase II)
  1. models and algorithms
     - address the scalability problem
     - investigate models for evolutionary self-adaptation
  2. system-on-chip architecture
     - implement complete adaptive SoC architecture
  3. case studies and evaluation
     - intrinsic implementation of the prosthetic hand controller
     - investigate autonomous robot control and navigation

Evolvable Hardware - Representation Model (1)

- Cartesian Genetic Program (CGP) [Miller and Thomson, '96]
  - the mostly used representation model for evolving digital hardware
  - array of combinational blocks connected by feed-forward wires, chromosome defines configuration of the array



- implemented highly parametrizable CGP model
  - array parameters: n_c, n_r, n_i, n_o, n_n, l
  - combinational blocks can be {AND, OR, …} or n_n bit table lookup

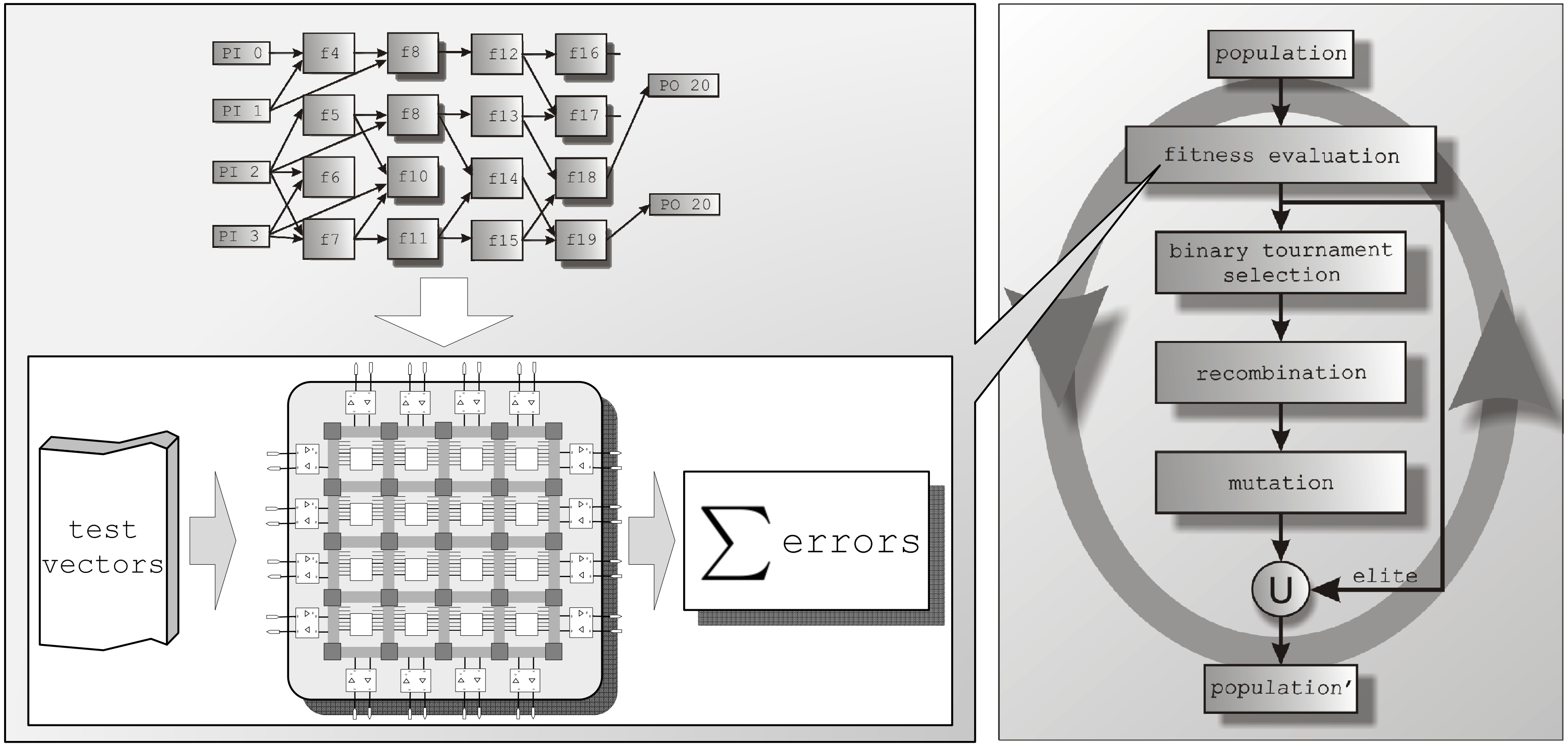
Evolvable Hardware - Representation Model (2)

- Embedded CGP [Walker and Miller, '05]
  - chromosome is a DAG and does not encode placement
  - allows for subfunction extraction (which is a problem for CGP)

- implemented ECGP model
  - module creation based on cones in the DAG
  - modules are not created randomly, but depending on the number of generations the module substructure has persisted in the population
  - dynamic mutation rates

Evolvable Hardware - Genetic Algorithms (1)

- reference algorithm GA
  - conventional single-objective genetic algorithm
  - uses elitism, tournament selection, uniform crossover, mutation



Evolvable Hardware – Genetic Algorithms (2)

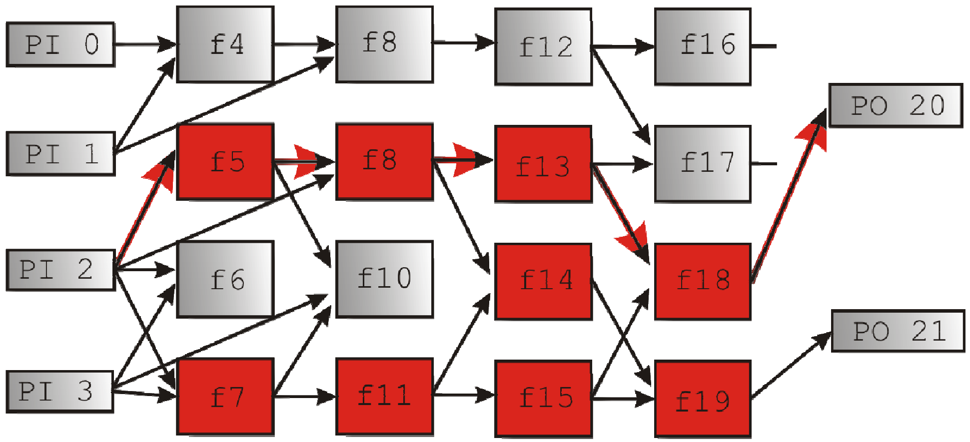
- optimization for multiple objectives
  - in circuit design: functional quality vs. speed vs. area (vs. power consumption)
  - often, the objectives are conflicting which leads to compromises
- approaches for optimizing circuits with multiple objectives
  - two-stage fitness [Kalganova and Miller '99], [Coello Coello '01]
  - we focus on a direct multi-objective evolution of digital circuits
- implemented algorithms
  - SPEA2 [Zitzler et al. '01]
  - µGA + µGA2 [Coello Coello '01]
  - NSGA II [Deb et al. '00]
  - TSPEA2, our own multi-objective optimizer [Kaufmann and Platzner '06]
  - OMOEA II [Liu et al. '06] } ongoing
  - IBEA [Zitzler and Künzli '04] } ongoing

Evolvable Hardware – Estimation of Objectives

- area and speed estimation for the CGP model

$$area(c) = 1 - \frac{used\_blocks(c)}{n_c \cdot n_r}$$

$$speed(c) = 1 - \frac{delay(c)}{n_c + 1}$$



- more precise estimation by Xilinx backend tools
  - transformation of CGP chromosomes to FPGA netlists using JHDL
  - useful for experimentation, but runtimes and memory requirements are prohibitive for intrinsic evolution

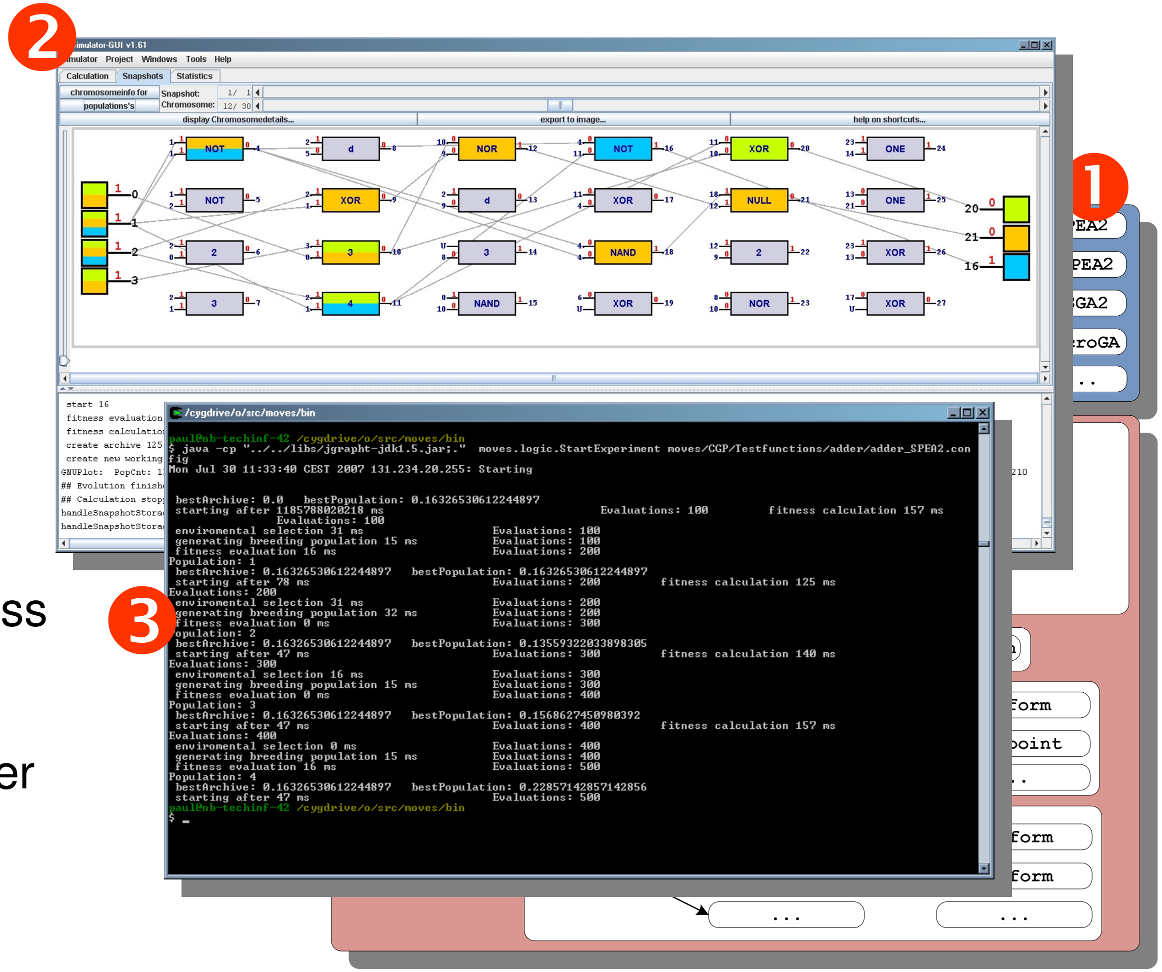
Evolvable Hardware – Test Problems

- functions with correctness property
  - arithmetic and logic functions, e.g. adder, multiplier, parity, ...
  - popular test functions for comparing representation models and algorithms
  - ➔ the evolutionary design of such functions is not our primary target, as here classically engineered solutions might be sufficient

- functions without correctness property
  - the functional quality depends on (changing) input data
  - a-priory or optimal solutions are unknown
  - e.g. classificator, cache controller, robot navigation controller, …

MOVES Framework

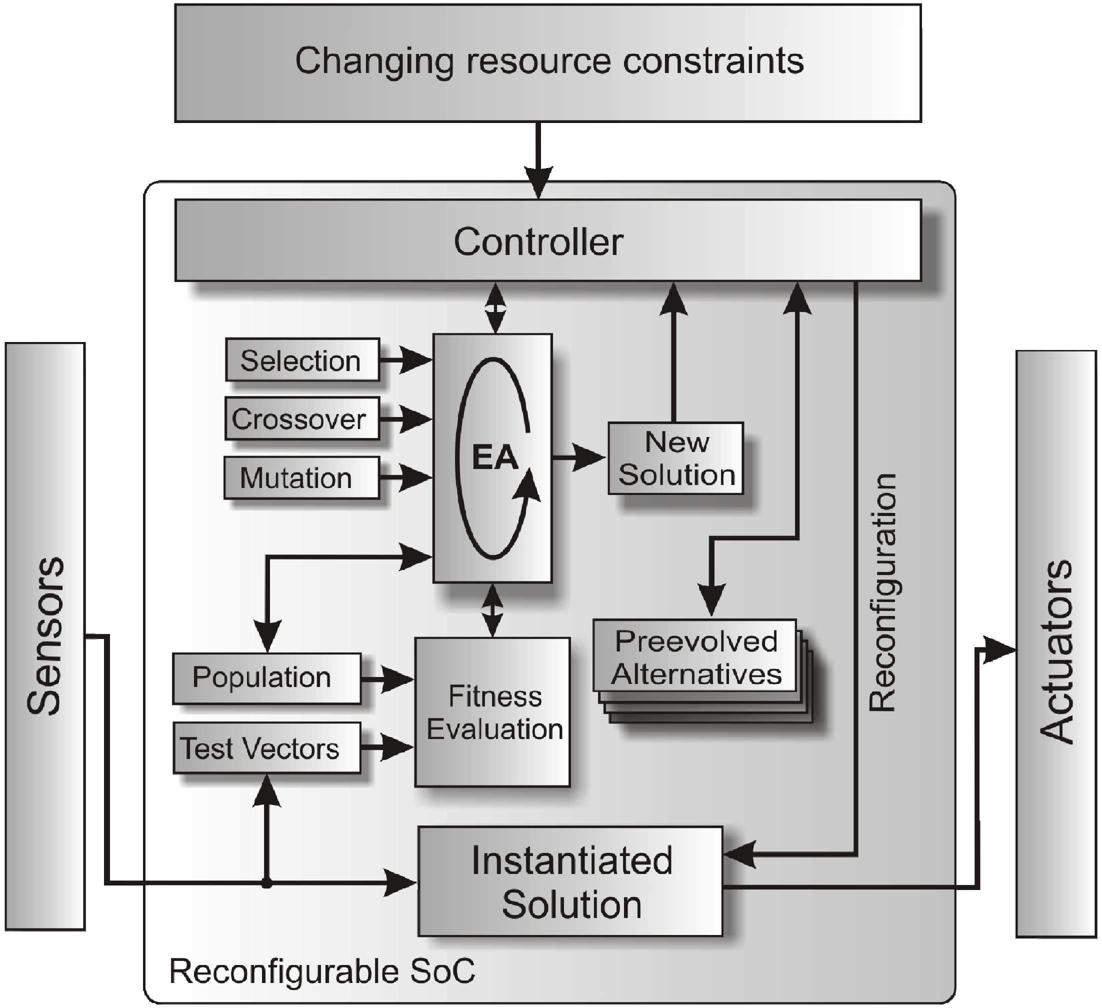
1. generic evolutionary algorithm modules
   - representation models
   - operators
   - evolutionary algorithms
2. graphical user interface
   - test different experiment parameter settings
   - evaluate newly implemented methods
   - investigate optimization process
3. batch mode tools
   - produce representative number of experiment runs
   - interface to grid computing software **Condor**



- tools can be downloaded from the MOVES project page: uni-paderborn.de/cs/ag-platzner/research/moves

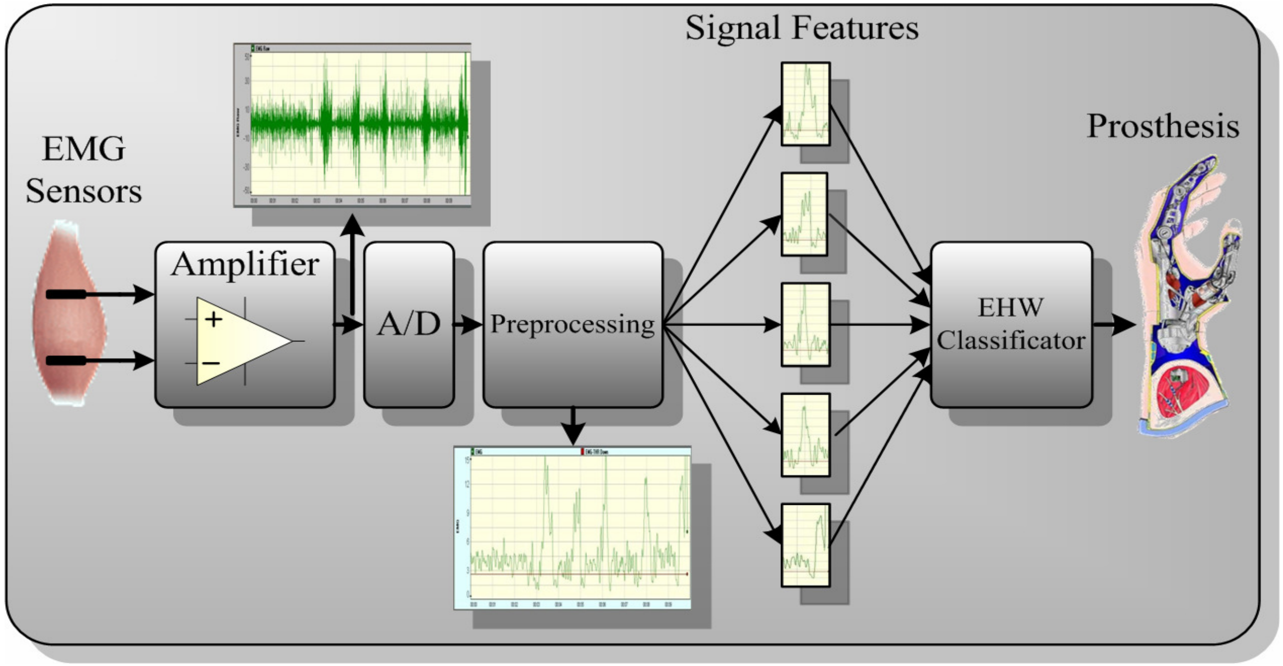
Evolvable Hardware – SoC Architecture

- started implementation on reconfigurable system-on-chip
  - initial hw/sw partitioning on a platform FPGA including CPU and logic
  - partial reconfiguration, self-reconfiguration
- in software (PowerPC)
  - CGP representation model
  - GA and SPEA2
- in hardware
  - instantiated solution
  - time-consuming functions, e.g.. k-th nearest neighbor clustering of SPEA2



Case Study: EMG Signal Classificator (1)

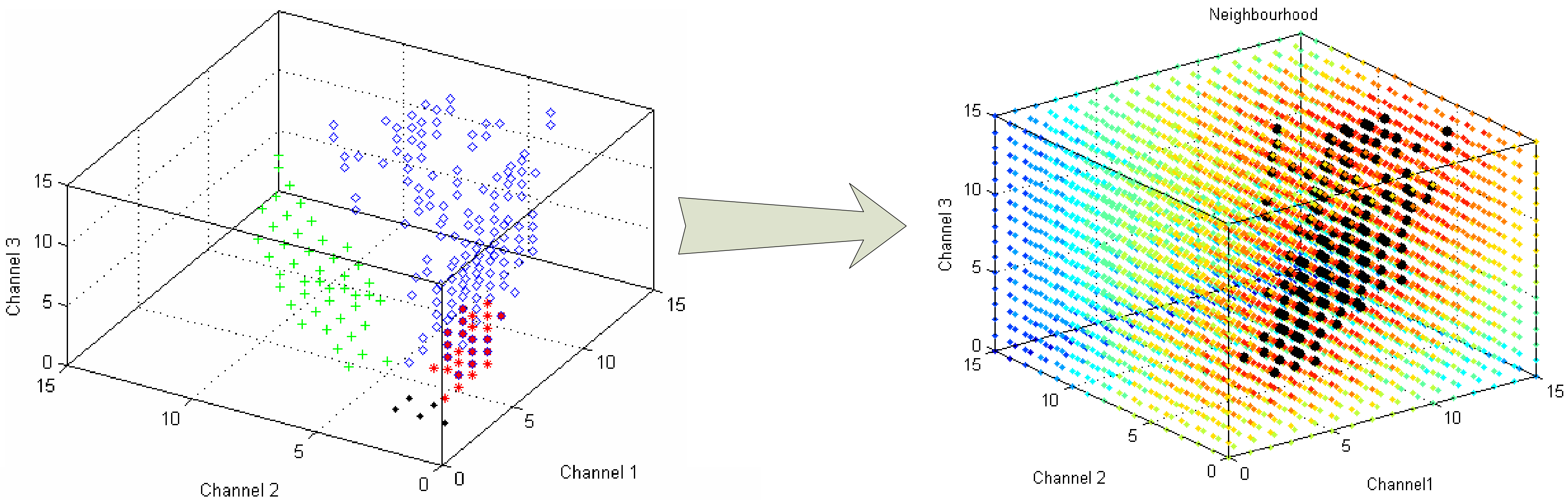
- started to develop a self-adaptive EMG signal classificator for prosthetic hand control
- two models of evolution
  - off-line,
    during training phases
  - online,
    in parallel to the operation



- current system setup
  - PC with attached EMG sensors, amplifiers and A/D converters
  - evolvable hardware classifier simulated on the PC

Case Study: EMG Signal Classificator (2)

- EMG classificators require self-adaptation due to ...
  - varying sensor positions
  - varying skin conductance
  - cross talking of neighbor muscles
  - heart-beat noise
  - muscle fatigue
  - movement patterns change over time
  - varying electronic (analog) component parameters



Collaborations & Publications

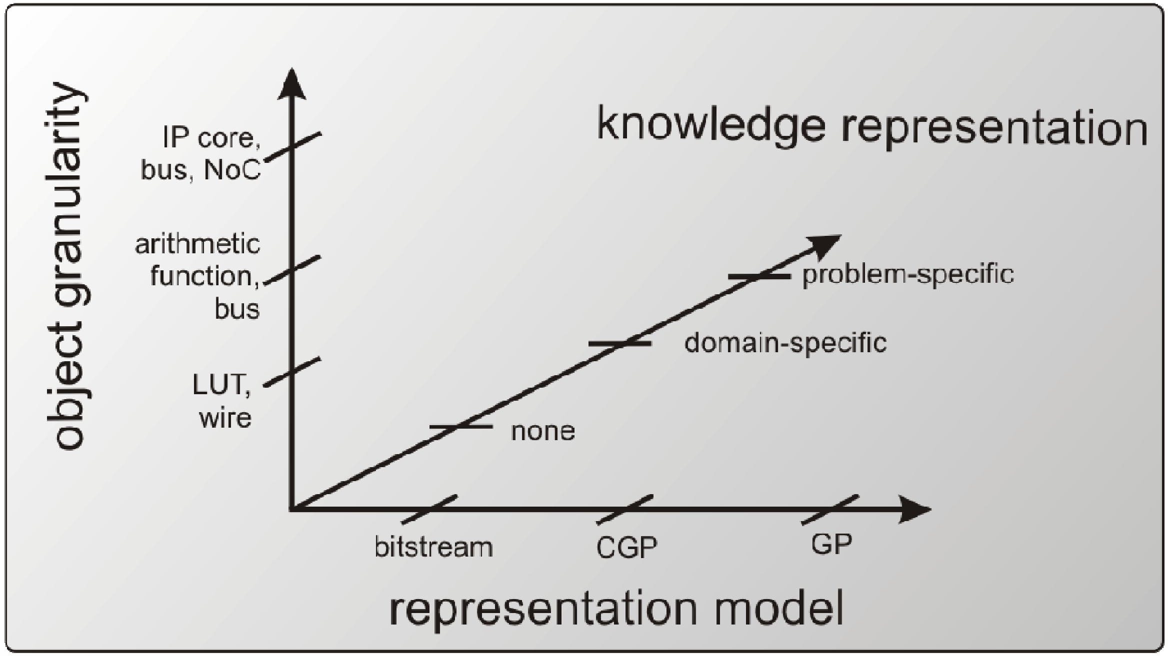
- Prof. Dr. Jim Tørresen, University of Oslo
  - scalability, resource-aware EHW SoC, classification architectures
  - supported by DAAD
- PD Dr. Bernhard Sick, University of Passau
  - classification of EMG signals
- Prof. Dr. Hartmut Schmeck, University of Karlsruhe
  - approaches for evolvable hardware

- Paul Kaufmann and Marco Platzner. Multi-objective Intrinsic Hardware Evolution. Proc. *International Conference on Military Applications of Programmable Logic Devices (MAPLD)*, Washington, DC, September, 2006.
- Paul Kaufmann and Marco Platzner. Toward Self-adaptive Embedded Systems: Multi-objective Hardware Evolution. Proc. *International Conference on Architecture of Computing Systems (ARCS)*, Zurich, March 12-15, 2007.
- Paul Kaufmann and Marco Platzner. MOVES: A Modular Framework for Hardware Evolution. Proc. *NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, Edinburgh, August 4-9, 2007.
  - **received best paper award in the "Evolvable Hardware" category**

Summary and Main Challenge

- summary
  - investigated models for digital circuit representation: CGP and ECGP
  - benchmarked a set of multi-objective optimizers for digital circuit design
  - created a tool suite for efficiently experimenting with models and algorithms
  - partially implemented a self-adaptive SoC architecture
  - first experiments with an evolvable classifier for prosthetic hand control

- main challenge: scalability
  - ➔ chose representation model and objects of "right" granularity
  - ➔ exploit problem-specific knowledge
  - ➔ (evolve incrementally)



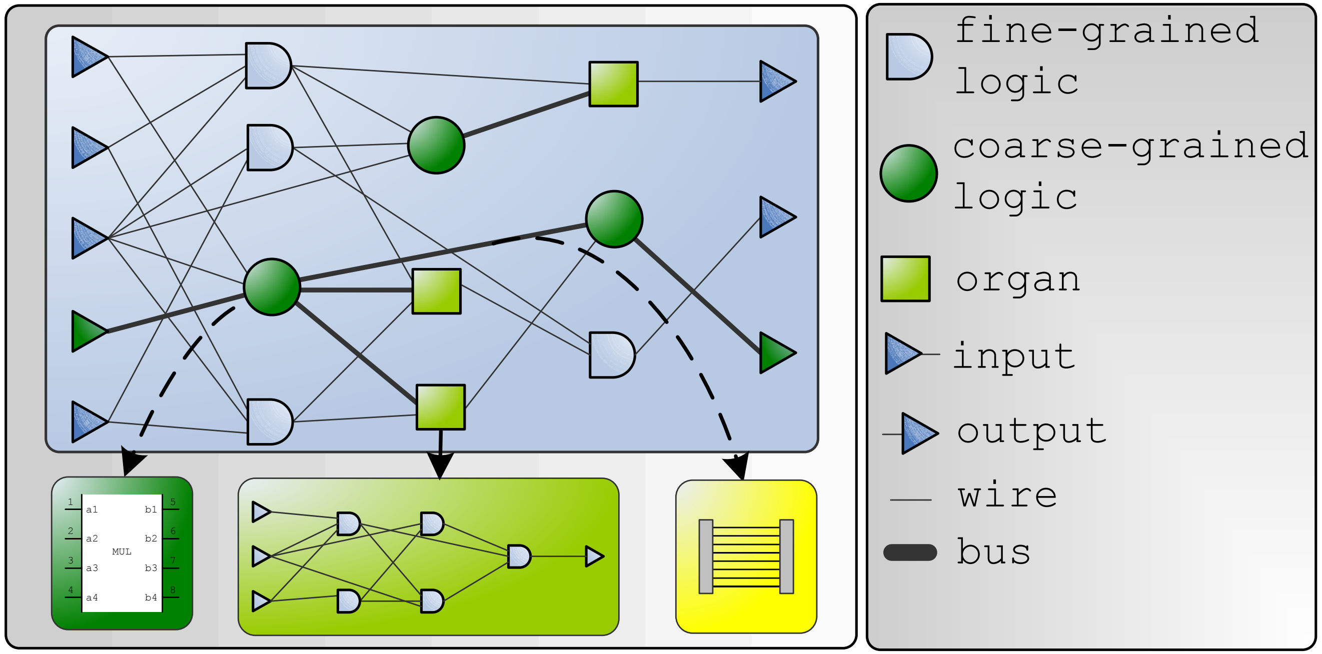
Overview

- topics of phase I
  1. models and algorithms
     - representation models for digital logic, multi-objective evolutionary optimization algorithms, tools for the evaluation of models and algorithms
  2. system-on-chip architecture
     - platform FPGA, hw/sw partitioning
  3. case studies and evaluation
     - test problems, classificator for electromyography (EMG) signals

- current and future topics (phase II)
  1. models and algorithms
     - address the scalability problem
     - investigate models for evolutionary self-adaptation
  2. system-on-chip architecture
     - implement complete adaptive SoC architecture
  3. case studies and evaluation
     - intrinsic implementation of the prosthetic hand controller
     - investigate autonomous robot control and navigation

Models and Algorithms

- novel representation model: multi-granular embedded CGP
  - use simple and complex building blocks
  - incorporate domain- and problem-specific knowledge



- constrain model parameters for
  - reducing the search space
  - efficient mapping to hardware

SoC Architecture

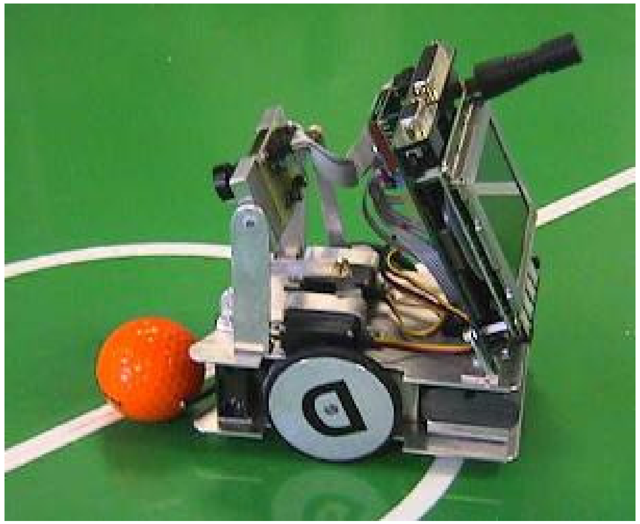
- develop a library of parametrizable hardware modules
  - that match the objects of the representation model (e.g.. LUTs, arithmetic units, ...)
  - that can be dynamically instantiated on the FPGA
    - "virtual" FPGA vs. partial reconfiguration

- investigate observer/controller architecture for critical applications
  - observe critical states
    - recognize unsafe outputs
    - use additional sensor information
  - react to critical states
    - emergency stop
    - instantiate fall-back solution

Case Study: Robot Control and Navigation

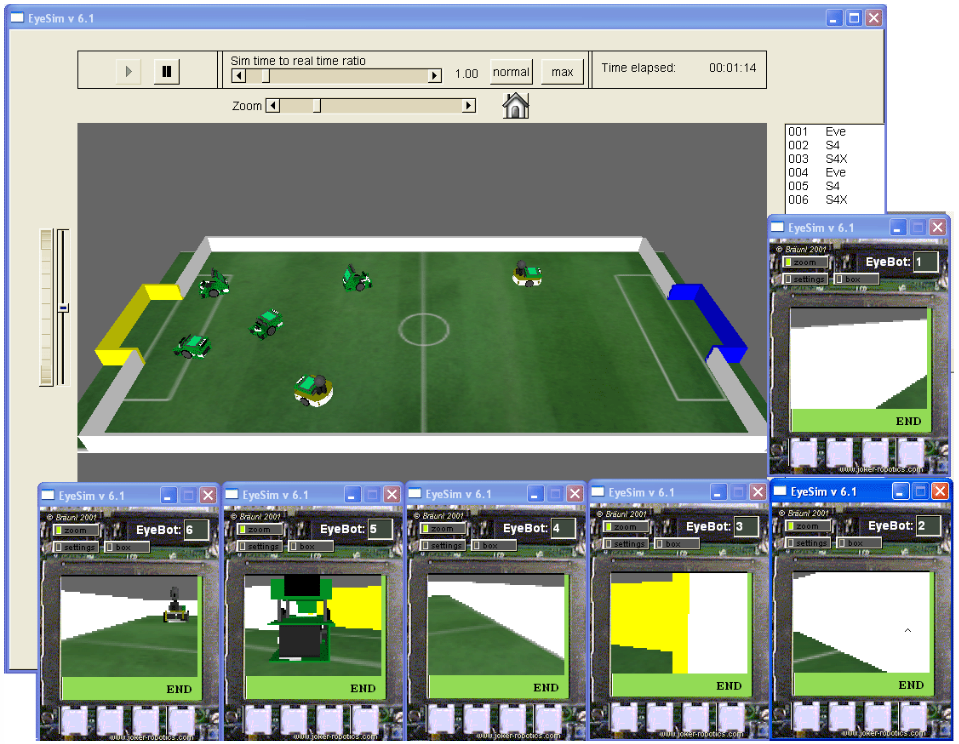
- develop self-adaptive evolvable hardware (EHW) robot controllers
  - integrate such "EHW agents" into EyeSim, a simulation environment for the EyeBot robot platform

- investigate different models for evolutionary self-adaptation
  - e.g.. embedded circuit design, self-adaptive, self-triggered, online evolution (after [Sekanina '04])
  - issues: how to gather test data, how to define fitness, how to validate the evolved functions



EyeBot

EyeSim



Thank you for your attention!