Evolvable Hardware for Autonomous Systems

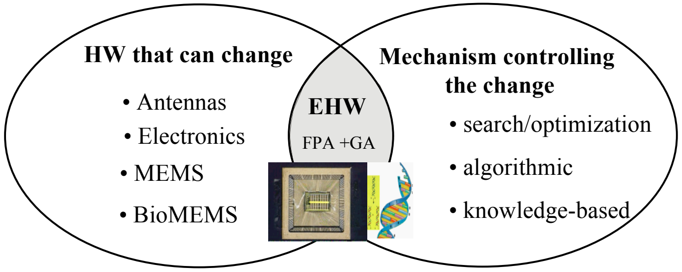
Adrian Stoica
Jet Propulsion Laboratory
adrian.stoica@jpl.nasa.gov, 818-354-2190, ehw.jpl.nasa.gov

JPL Co-investigators: D. Keymeulen, R. Zebulum, M.I.Ferguson, T. Daud, T. Arslan (Visiting Prof – U Edinburgh)

GECCO-2005

Evolvable Hardware (EHW) Technology

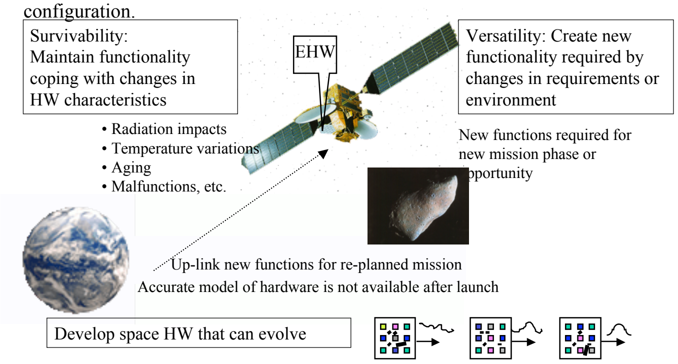
Evolvable Hardware = Reconfigurable HW + Reconfiguration Mechanism
In a narrow sense (EHW) is programmable hardware self-configurable by built-in Evolutionary Algorithms.



intelligent part – the built-in mechanisms controling the adaptation/self-configuration

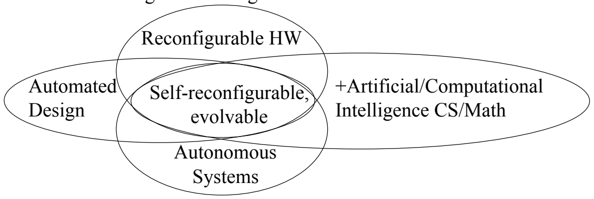
EHW for flexibility and survivability of autonomous systems

JPL/NASA driver – long–life spacecraft
Dramatic changes in hardware/environment, e.g. in case of faults or need for new functions, may require in-situ synthesis of a totally new hardware configuration.


Survivability:
Maintain functionality coping with changes in HW characteristics
EHW
Versatility: Create new functionality required by changes in requirements or environment
• Radiation impacts
• Temperature variations
• Aging
• Malfunctions, etc.
New functions required for new mission phase or opportunity
Up-link new functions for re-planned mission
Accurate model of hardware is not available after launch
Develop space HW that can evolve

EHW: definitions

- Intelligent automatic design of reconfigurable hardware…

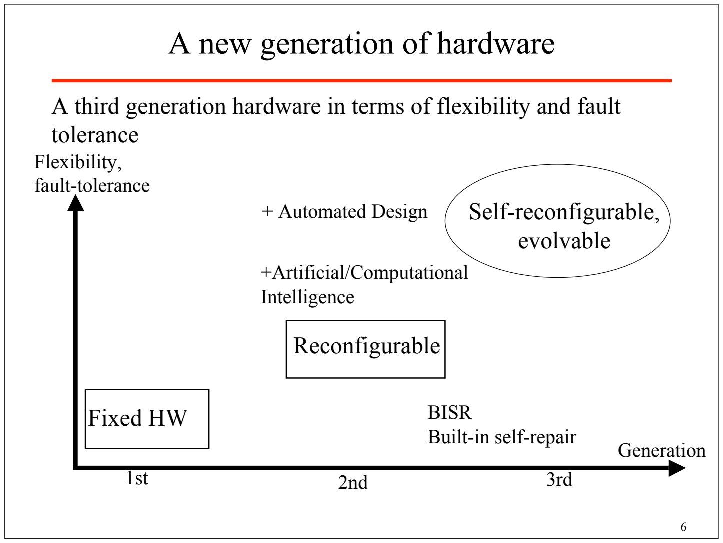

Reconfigurable HW
Automated Design
Self-reconfigurable, evolvable
+Artificial/Computational Intelligence CS/Math
Autonomous Systems

- **Objective:** development of flexible and survivable HW capable of intelligent self-configuration, self-tuning, and self-repair, that can can adaptively change through reconfiguration/compensation for optimal signal processing, sensing/control, and survival in the presence of faults/degradation.

- **Successful applications:** in automated design, automated calibration and tuning, in-field adaptation of hardware systems, sensing, control and robotics.

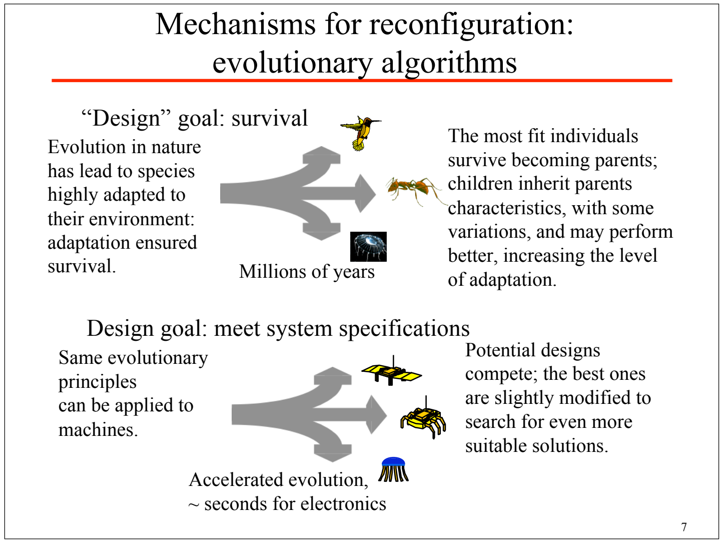
Tutorial Overview

- Introduction to EHW
- Algorithms for self-configuration and evolution
- Reconfigurable and Morphable Hardware
- Demonstrations of Evolvable Systems
- Application Examples
- System Aspects
- Resources for EHW Engineers



A new generation of hardware

A third generation hardware in terms of flexibility and fault tolerance



Mechanisms for reconfiguration: evolutionary algorithms

"Design" goal: survival

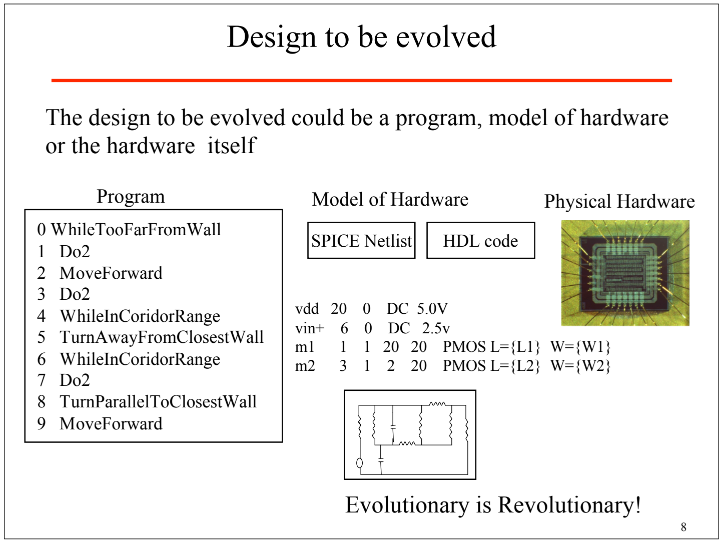
Evolution in nature has lead to species highly adapted to their environment: adaptation ensured survival.

The most fit individuals survive becoming parents; children inherit parents characteristics, with some variations, and may perform better, increasing the level of adaptation.

Design goal: meet system specifications

Same evolutionary principles can be applied to machines.

Potential designs compete; the best ones are slightly modified to search for even more suitable solutions.



Design to be evolved

The design to be evolved could be a program, model of hardware or the hardware itself

Program

```
0 WhileTooFarFromWall
1  Do2
2  MoveForward
3  Do2
4  WhileInCoridorRange
5  TurnAwayFromClosestWall
6  WhileInCoridorRange
7  Do2
8  TurnParallelToClosestWall
9  MoveForward
```

Model of Hardware

SPICE Netlist   HDL code

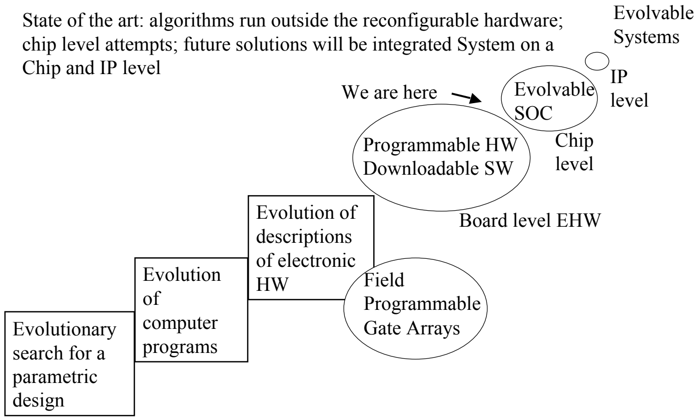
```
vdd   20    0    DC  5.0V
vin+   6    0    DC  2.5v
m1     1    1   20   20   PMOS L={L1}  W={W1}
m2     3    1    2   20   PMOS L={L2}  W={W2}
```

Physical Hardware

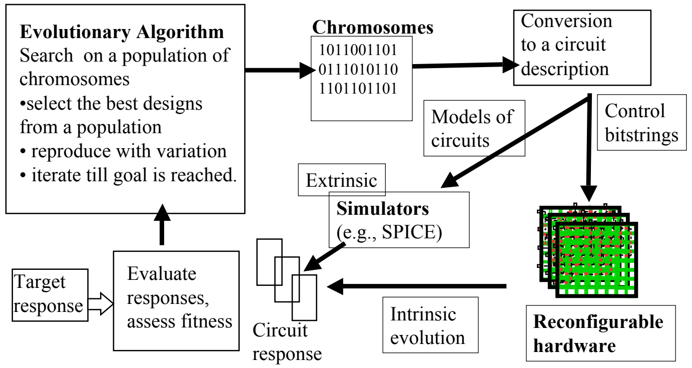
Evolutionary is Revolutionary!

Evolution of EHW

State of the art: algorithms run outside the reconfigurable hardware; chip level attempts; future solutions will be integrated System on a Chip and IP level


Evolvable Systems
IP level
We are here
Evolvable SOC
Chip level
Programmable HW Downloadable SW
Board level EHW
Evolution of descriptions of electronic HW
Field Programmable Gate Arrays
Evolution of computer programs
Evolutionary search for a parametric design

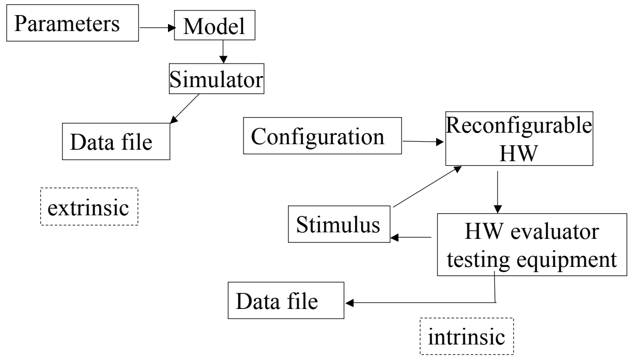
Evolutionary synthesis and adaptation of electronic circuits


Evolutionary Algorithm
Search on a population of chromosomes
•select the best designs from a population
• reproduce with variation
• iterate till goal is reached.
Chromosomes
1011001101
0111010110
1101101101
Conversion to a circuit description
Models of circuits
Control bitstrings
Extrinsic
Simulators (e.g., SPICE)
Target response
Evaluate responses, assess fitness
Circuit response
Intrinsic evolution
Reconfigurable hardware

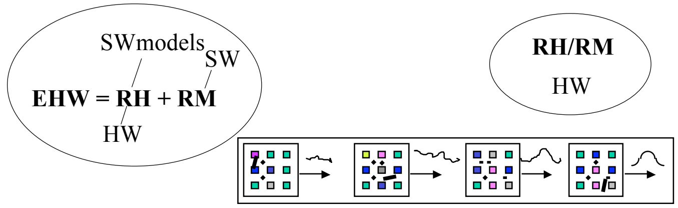
Potential electronic designs/implementations compete;
the best ones are slightly modified to search for even more suitable solutions

Extrinsic and intrinsic EHW

Path from chromosome to behavior data file



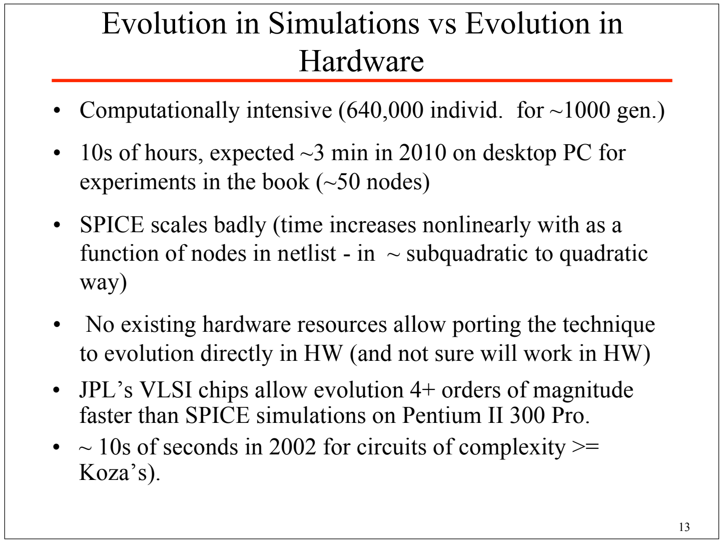
EHW implementation: HW/SW



Current approach to EH implementation:
- Use RH- reassign cell function/interconnection
- Use powerful parallel searches (e.g., GAs) to evolve the hardware

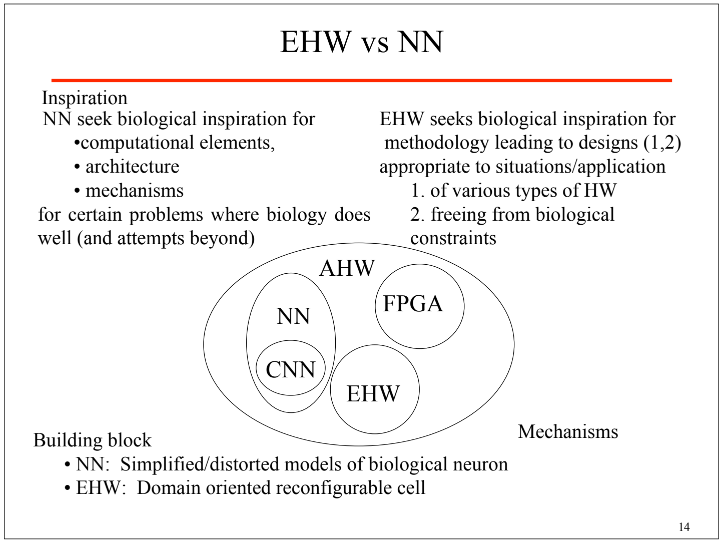
In addition EHW requires

• Fast evaluation	Present solutions:RM in SW
• Low cost for failure	Future: everything **seamlessly** integrated in HW



Evolution in Simulations vs Evolution in Hardware

- Computationally intensive (640,000 individ. for ~1000 gen.)
- 10s of hours, expected ~3 min in 2010 on desktop PC for experiments in the book (~50 nodes)
- SPICE scales badly (time increases nonlinearly with as a function of nodes in netlist - in ~ subquadratic to quadratic way)
- No existing hardware resources allow porting the technique to evolution directly in HW (and not sure will work in HW)
- JPL’s VLSI chips allow evolution 4+ orders of magnitude faster than SPICE simulations on Pentium II 300 Pro.
- ~ 10s of seconds in 2002 for circuits of complexity >= Koza’s).



EHW vs NN

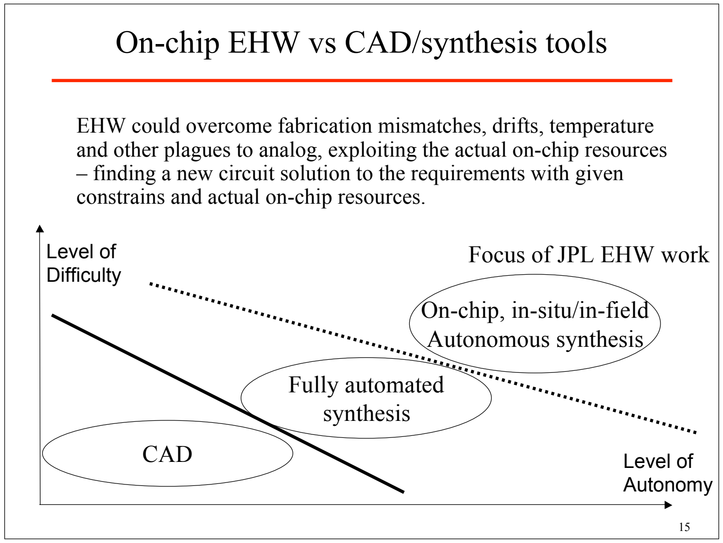
Inspiration
NN seek biological inspiration for
- computational elements,
- architecture
- mechanisms

for certain problems where biology does well (and attempts beyond)

EHW seeks biological inspiration for methodology leading to designs (1,2) appropriate to situations/application
1. of various types of HW
2. freeing from biological constraints

Building block
- NN: Simplified/distorted models of biological neuron
- EHW: Domain oriented reconfigurable cell

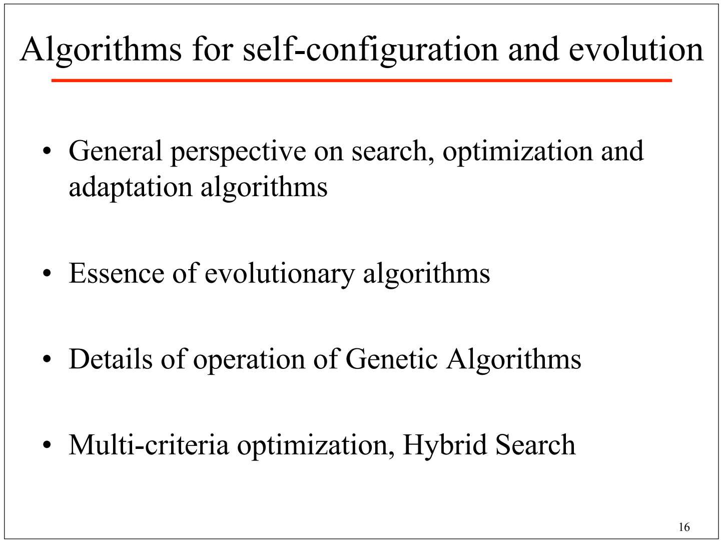
Mechanisms



On-chip EHW vs CAD/synthesis tools

EHW could overcome fabrication mismatches, drifts, temperature and other plagues to analog, exploiting the actual on-chip resources – finding a new circuit solution to the requirements with given constrains and actual on-chip resources.

Level of Difficulty
Focus of JPL EHW work
On-chip, in-situ/in-field Autonomous synthesis
Fully automated synthesis
CAD
Level of Autonomy



Algorithms for self-configuration and evolution

- General perspective on search, optimization and adaptation algorithms
- Essence of evolutionary algorithms
- Details of operation of Genetic Algorithms
- Multi-criteria optimization, Hybrid Search

Algorithms for reconfiguration. Objectives: control self-configuration for desired functionality

- A control C that creates a structure / topology / architecture S, that has the function F. Specification in terms of S or F. F may include constraints, preferences, etc.
- Behavior/Function may change in time, in simple case it doesn’t
- Often C, even for a set of states which can be decomposed, but could be a sequence C1 C2 C3 if system has memory
- Digital or analog controls (analog signals often obtained by conversion from digital)

Search/optimization algorithms and NFL Theorems

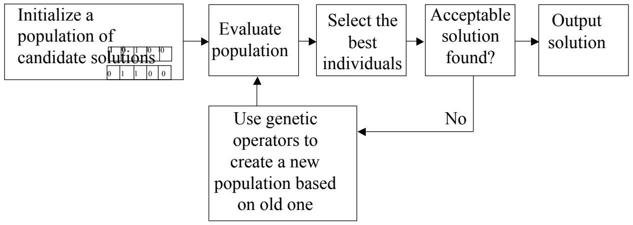
- Start with an initial "guess" at a solution, which is updated over iterations with the aim of improving the performance measure (objective function).
- Multiple variables influence the function: a multivariable optimization problem of minimizing or maximizing an objective function.
- No free lunch Theorem: No search algorithm is uniformly better than all other algorithms across all possible problems. (Cheaper lunches in certain places: Some algorithms may work better than others on certain classes of problems as a consequence of being able to exploit the problem structure.)
- E.g. traditional nonlinear programming methods (e.g., constrained conjugate gradient) are well suited to deterministic optimization problems with exact knowledge of the gradient of the objective function; more generally, stochastic gradient methods are effective if one has direct (unbiased) measurements of the gradient of the objective function.

Principles of evolutionary processes

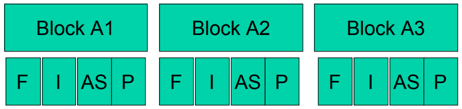
- Genetic program (genotype) -> expressed behavioral traits (phenotype)
- Pleiotropy: a single gene may simultaneously affect several phenotypic traits.
- Polygeny: a single phenotypic characteristic may be determined by the simultaneous interaction of many genes.
- Epistasis: expression of one gene masks the phenotypic effects of another
- There are no one-gene, one-trait relationships in naturally evolved systems.
- Very different genetic structures may code for equivalent behaviors; various circuits that implement a function with electronic components.

Principle of operation of evolutionary algorithms

Coding solutions as chromosomes. Operates on code not on solution.
A string is a candidate solution.
•Switch states 11011 Bitstring
•Program (+x(*x (- x 1)))
•Vector (4.3 3.2 500)


Initialize a population of candidate solutions
1 0 1 0 0
0 1 1 0 0
Evaluate population
Select the best individuals
Acceptable solution found?
Output solution
No
Use genetic operators to create a new population based on old one

From code to instantiation



- F=Function, I= Interconnect of terminals, AS= Analog Signal, P=Parameter of passive component

- Example for Block A1: 101 0101 1000 0011 101 1111
- F= 101 (e.g. 2 input Amp with gain g1, out of 8 choices)
- I: 0101 1000 0011 (eg 2 in and 1 out, NESW connections

  In 1 from E W, In 2 from N, Out goes to S and W
- Analog Signal: 101 An analog bias of 6/8 on bias node
- Parameter of passive component: 1111 selection of 10k R

Measurement and evaluation of individuals

- Testbench, or in-system measurement:
  - Stimulation signals
  - Load on the output signals
- Measured in the testbench:
  - Time response, taking samples (Often A/D conv for processing in digital
  - Frequency response, (directly (Spectrum Analyzer), indirect (FFT)
  - Other measures, such as current from source
  - Effects: derived effects (if electrical device controls something else)

- Quality of an individual: an overall fitness value is determined based on individual fitness function associated to the testbench, and their weighting.

Objective Functions

Objective function evaluates how well each individual performs.

Goal: maximize the objective function

Standard Method: compute a distance to a target

$$F = - \sum_{i=0}^{n-1} (W_i \cdot |R_i - T_i|)$$

Fitness F is computed over n samples;

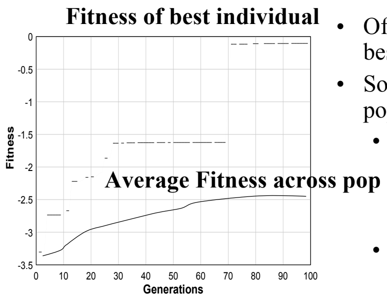
R_i – Individual Response;

T_i – Target Response;

W_i – Weights reflecting some knowledge of the problem

The design of a good fitness evaluation function is critical for evolution.

Improvement in individual and population



Example of improvement in fitness of the best individual over the generations (and

- Often we care only of best individual
- Sometimes we care of a population:
  - For monitoring purposes to understand better what is going on
  - For fault-tolerance we may want several good “mutants” – a fault gives a mutant which still has high fitness

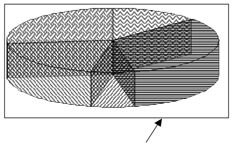
Selection

- Based on the principle of survival of the fittest;
- Better candidate solutions get more offspring with same/close genetic code
- •Deterministic in ES and EP;
- Probabilistic in GA and GP

•Selection Techniques:
- •Proportional Selection;
- •Rank based selection;
- •Exponential Selection;
- •Tournament Selection;

Proportional Selection
Roulette wheel selection

Spin the roulette

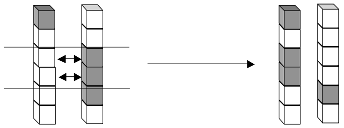


Slice in roulette and fitness of an individual are proportional

Those who have higher fitness have higher probability to be selected for mating

Crossover

- Genetic algorithms

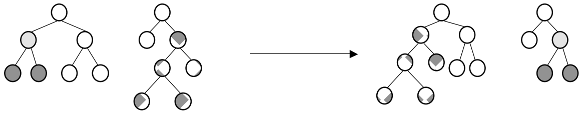


Parents

Offspring

2-point Crossover

- Genetic Programming



Parents

Offspring

•Recombination of genetic material that contributes to the variability in the population;
•Harmful effects: destroying potentially useful building blocks
  •Automatically Defined Functions (ADFs): protection against disruptive effect of crossover.

Schemata, building blocks, ADFs

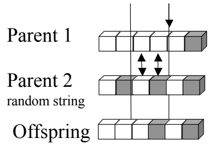
- GA/GP (EAs using crossover) use the building block theory – useful components of what makes a solution (chunks of chromosomes) can be efficiently manipulated and used to lead to the solution.
- A problem decomposition
- Looking for similarities patterns in chromosomes of similarly performing solutions
- 1100 10
- 0010 3
- 0101 4
- 1101 20 11xx (or 110x) may be a good building block –schemata set of all combinations based on same pattern

- Goldberg: Ensure BB supply, growth, understand BB spead, ensure good BB decisions, know BB challenges, ensure good BB mixing
- Crossover probability – rules of thumb

Mutation

- Each bit of a new string can be changed (mutated) with a probability given by the *mutation rate*;
- Low values for the mutation rate are often used;
- Traditional interpretation: only support for crossover;
- More recent voices: driving force of GAs: (something other EA camps have always stated) (or it depends on problem/rep)
  - – GAs performance largely affected by the mutation rate.

Crossover with random string



Parent 1

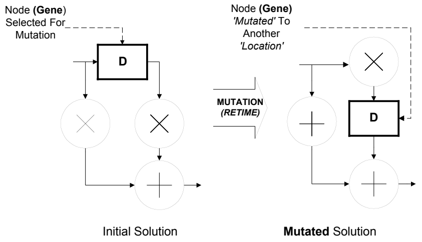
Parent 2
random string

Offspring

Adaptive operators

- Adapting the probability associated to evolutionary operators improves convergence
- Crossover probability
- Mutation probability
- Change representation
- Change selection probability and method – all is permitted

Specific Crossover and Mutation


Node (Gene) Selected For Mutation
D
MUTATION (RETIME)
Node (Gene) 'Mutated' To Another 'Location'
D
Initial Solution
Mutated Solution

Example of Mutation Operation

Examples from T. Arslan's work

Population, generations, runs

- Fixed or variable population size
- Small populations – more generations, vice-versa
- ~ 100 individuals very common
- Usually GP asks for more: eg 640,000 in some of Koza's experiments
- Hundreds of generations
- Sampling a small % of space
- See if it is still improving tracing amount of changes in last generations
- Stop: nr of generations, time, lack of improvement
- Re-start, change initial population, seed with solutions

Multi-criteria optimization, trade-offs, Pareto optimality

- The simultaneous optimization of multiple, possibly competing, objective functions deviates from the single-function optimization in that it seldom admits a perfect (or Utopian) solution;
- Instead, multi-objective optimization problems tend to be characterized by a family of alternatives that must be considered equivalent in the absence of information concerning the relevance of each objective relative to the others;
- Two different methods: Plain aggregating approaches and Pareto-based approaches;
- Plain aggregating approaches perform the scalarization of the objective vectors: each objective, $f_i(x)$, multiplied by the weight w_i.

$$f(x)=\sum_{i=1}^{n} w_i f_i(x)$$

Pareto: A is superior to B if in all objectives A is better or equal to B

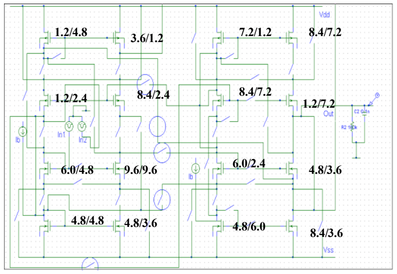
e.g. compare houses by price, lower interest, safer neighborhood

A $200k, 6%, 1/5 B $300k, 6%, 1/5, C $300k 5% 2/5

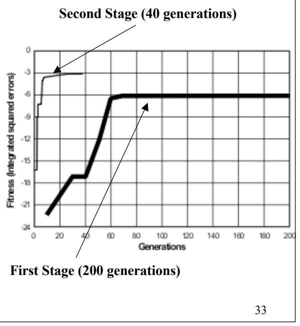
A and C solutions on Pareto front

Multi-stage search: Search for topology followed by parameter optimization (multi-method ok)

- First Stage: GA-based Evolution of the circuit topology;
- Second Stage: GA-based Optimization of the transistor sizes for the best topology resulted in the first stage. Initialization is made with the best topology and random parameters.



Multiplier Evolved Through Multi-stage search W/L in um



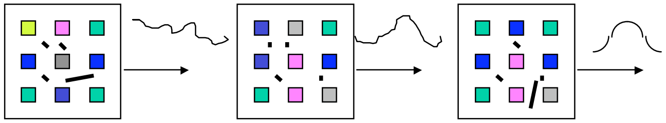
Reconfigurable Hardware

- Reconfigurable hardware (switch-based). Devices, SW Tools, Potential for EHW
- Field Programmable Gate Arrays (FPGA) – Xilinx examples
- Field Programmable Analog Arrays (FPAA) – Anadigm Examples
- Field Programmable Transistor Arrays (FPTA) – JPL examples

Reconfigurable Devices, Tools, Potential for EHW

- Function change by configuration change
- Switch-based devices, switches interconnecting functional modules of primitive functions (logical or analogical)
- Vendor programming tools allow switches to be turned ON/OFF, in a mode visible or invisible to user, via intermediary program conversions.
- Determining the status of the switches, – which switches are ON (OFF) is the search/optimization problem for EHW. Either a local search for compensation; variations around a configuration determined by knowledge/analytical means, or, a new configurations needs to be searched e.g. when unidentified faults prevent mapping of computed solutions
- Status of switches – on or off – can be straightforward associated with a binary representation used by genetic algorithms

Reconfigurable hardware is hardware that changes cell function and cell interconnect



Language for programming reconfigurable hardware needs to define:
Alphabet –choices of cells
Vocabulary/Grammar – rules of interconnect

Genetics: {G,A,T,C} (GATTACA)

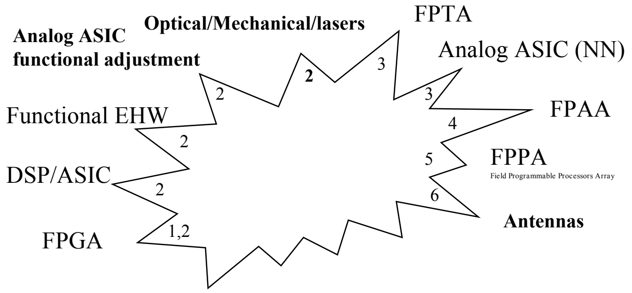
IBM Computer: {1,0} (1010011)

FPGA: AND, OR, NOT

FPTA: Cells of Transistor Arrays

HW Platforms for EHW Experiments

First/ significant experiments on:...


Analog ASIC functional adjustment
Optical/Mechanical/lasers
FPTA
Analog ASIC (NN)
Functional EHW
FPAA
DSP/ASIC
FPPA
Field Programmable Processors Array
FPGA
Antennas
2
2
3
3
4
2
5
2
6
1,2

1 Thompson, U. Sussex, UK
2 Higuchi, ETL, Japan
3 Stoica, JPL

4 Zebulum, JPL
5 Marchal, CSEM, Switzerland
6 Linden

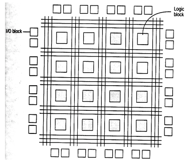
COTS digital reconfigurable hardware

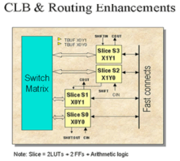
PLA

FPGA

Xilinx 6200

Virtex, VirtexII Pro (Xilinx)





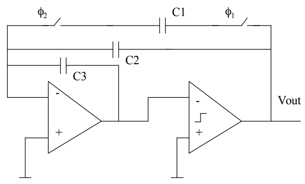
Altera, Actel, Other companies, etc…

Programmable SOC

COTS analog reconfigurable hardware

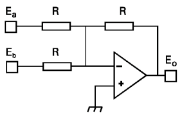
FPAAs

Pilkington
Motorola MPAA020
Now Anadigm
• Switched capacitors



Zetex TRAC

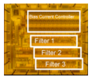
- Totally Reconfigurable Analog Circuit
- 20 cells, each an op-amp with a small reconfigurable network
- Cell can do one of: Add, negate, substract, multiply, pass, log, antilog, rectify, or basic inverting opamp for use with external components

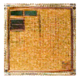


Lattice

Custom Made EHW-oriented reconfigurable hardware

Japan Higuchi EHW-chips

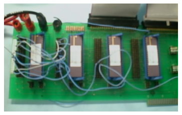




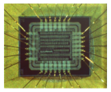
Industrial, specific

Research, general

JPL'98 FPTA-0



JPL'2001 FPTA-2
Integrated 64 cells (each 44 programmable transistors)



Boards MUX-based
UK Sussex (Evolvable motherboard)

Germany (Heidelberg)
Array of 16x16 programmable transistor cells

Brazil -PAMA

UK Edinburgh Palmo

Configurable Mixed-Signal Array

The PSoC™ CY8C25122/CY8C26233/CY8C26443/CY8C26643 family of programmable system-on-chip devices [redacted]e multiple MCU-based sy~~st~~em components with one single-chip, configurable device.

A PSoC device includes configurable analog and digital peripheral blocks, a fast CPU, Flash program memory, and SRAM data memory in a range of convenient pin-outs and memory sizes.

The driving force behind this innovative programmable system-on-chip comes from user configurability of the analog and digital arrays: the PSoC blocks.

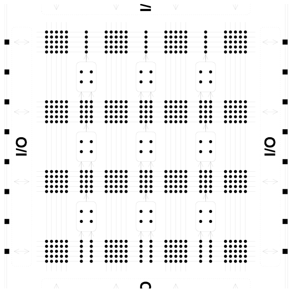
Example Applications on the PSoC (Application notes on www.cypress.com)

- PSoC Single-Phase Power Meter Reference Design
- Modem - 300 Baud
- Magnetic Card Reader Reference Design

Cypress CY8C26443
Final Datasheet
May. 29, 2003

Cellular Architecture in Xilinx 6200 chip

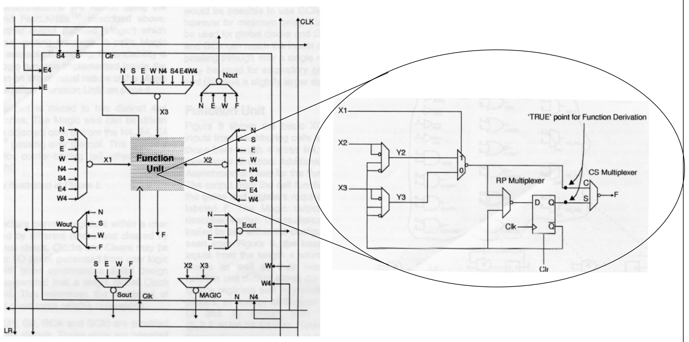
- Architecture, transparence
- NESW
- Can take any bitstring



- configuration switches
- Routing short links with neighbours, long busses skipping over areas, hierarchy of routing resources,

- configurable blocks LUT one of 8/16 of 2/3 input gates
- RAM

Xilinx XC6200 cell

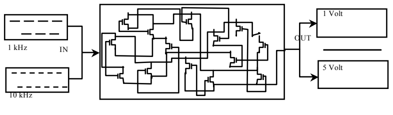


SRAM-controlled switch (mux)-based FPGA

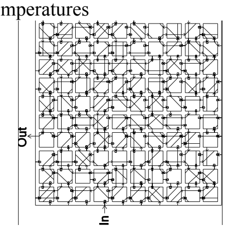
Thompson's experiment

- Adrian Thompson @ Sussex U.
- Frequency discriminator
- 10x10 corner of FPGA Xilinx 6200, no clk
- Conventional design searches in constraint regions
- EA can explore larger space, possibly better solution
- Evolution of robust circuits:

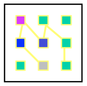
Use of FPGAs from different foundries, at different temperatures



Tone-Discriminator for 1 kHz and 10 kHz using Transistors

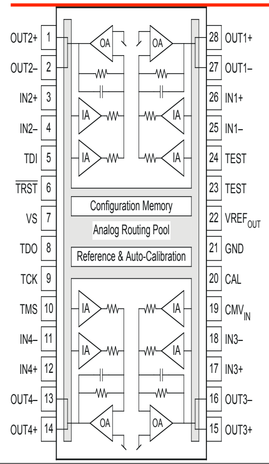


1kHz - 10KHz



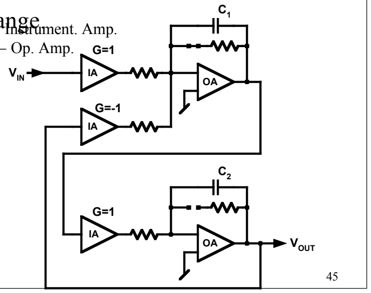
0 1

Lattice ispPAC10

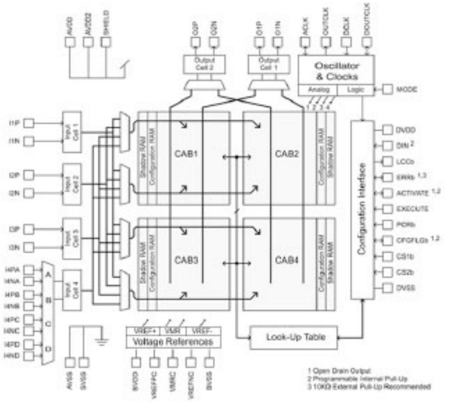


- four programmable analog modules and a programmable interconnection system
- can be configured to implement 2nd and 4th order active LP and BP filters in the 10 Khz—100 Khz range

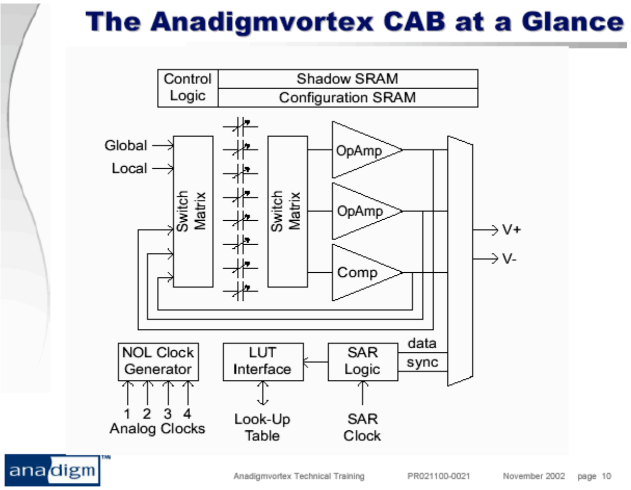
IA – Instrument. Amp.
OA – Op. Amp.



Architecture of AN220E04 FPAA



http://www.anadigm.com/_doc/FPAAfamilyOverview.pdf



The Anadigmvortex CAB at a Glance

Anadigmvortex Technical Training PR021100-0021 November 2002 page 10

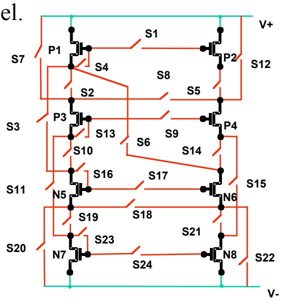
Why Custom

Programmable analog only allowed configuration around OpAmp level. There are many interesting circuits topologies to evolve below this level.

Evolution-oriented devices

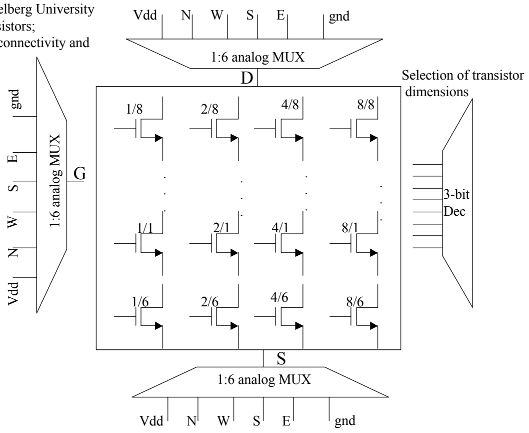
- can reprogram many times
- can understand what's inside
- Flexible programmability

- Example: JPL PTA,
- Reconfigurable at transistor level
- Both analog and digital



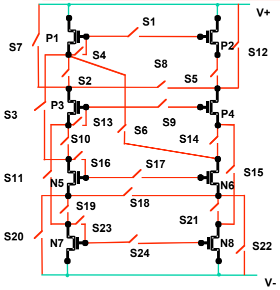
FPTA of U. Heldelberg

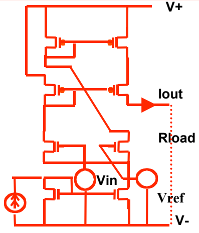
- Langeheine @ Heidelberg University
- Array of 16x16 transistors;
- Programmability in connectivity and channel lengths.



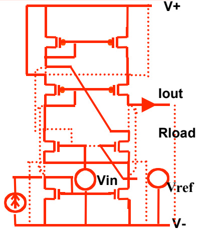
Selection of transistor dimensions

Programmable Transistor Array Cell – FPTA0



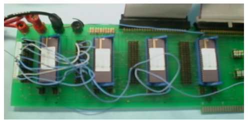


Human Design

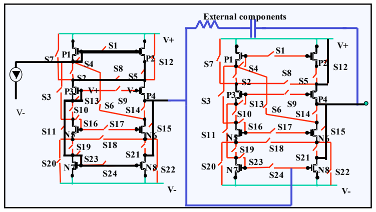


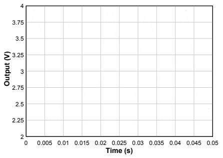
Leakage through finite resistance OFF

- 24 programmable switches: sufficient number for meaningful topologies
- Chromosomes give the value HIGH-LOW (not only ON-OFF) of the switches
- All the terminals are connected via switches to expansion terminals
- CMOS (0.5 μ) - MOSIS

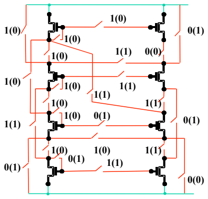


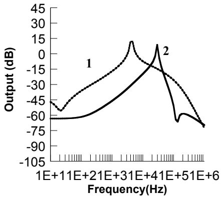
Op.Amp, Filters Mapped into PTA cells





OA response, sine wave input:
Red - Without switches
Blue - With switches.



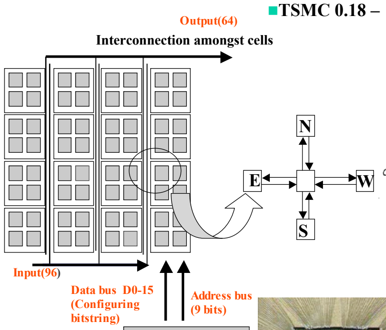


Filter Characteristics:
- Configuration 1:
  Filter with 11dB gain at 5kHz , roll-off about –30dB/dec.
- Configuration 2 :
  Filter with 9dB gain at 25kHz, roll-off about –40dB/dec .

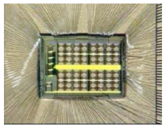
Programmable Transistor Array Cell – FPTA2

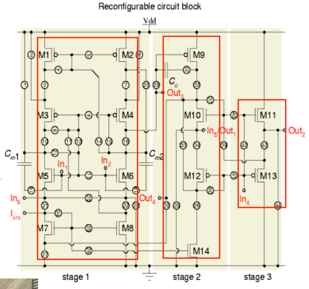
- Implementation of an evolution-oriented reconfigurable architecture (EORA)

■TSMC 0.18 – 1.8v;

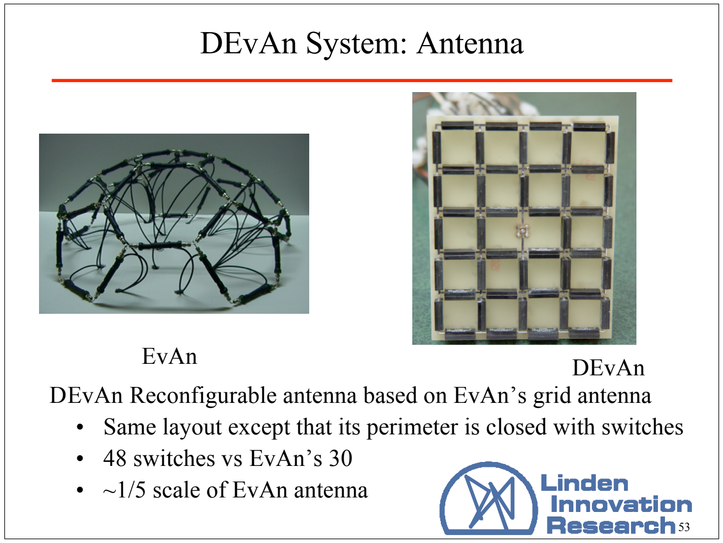

Output(64)
Interconnection amongst cells
N
E
W
S
Input(96)
Data bus D0-15 (Configuring bitstring)
Address bus (9 bits)
Control Logic

Chip Architecture





Cell Schematic



DEvAn System: Antenna

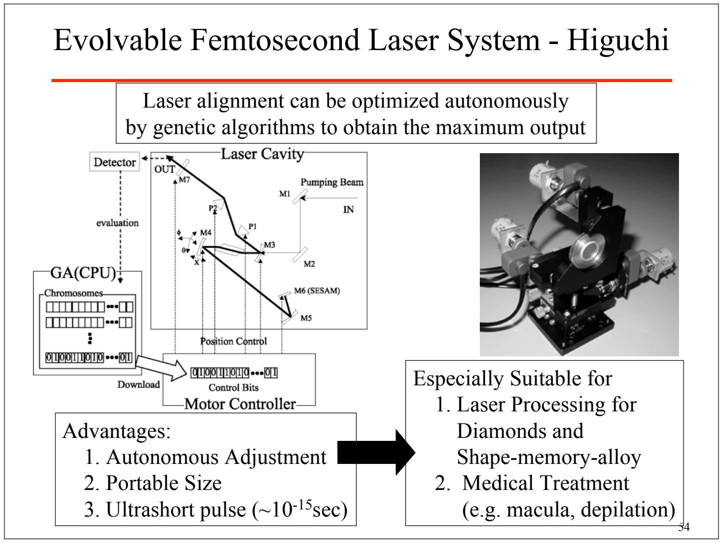
EvAn

DEvAn

DEvAn Reconfigurable antenna based on EvAn’s grid antenna

- Same layout except that its perimeter is closed with switches
- 48 switches vs EvAn’s 30
- ~1/5 scale of EvAn antenna

Linden Innovation Research

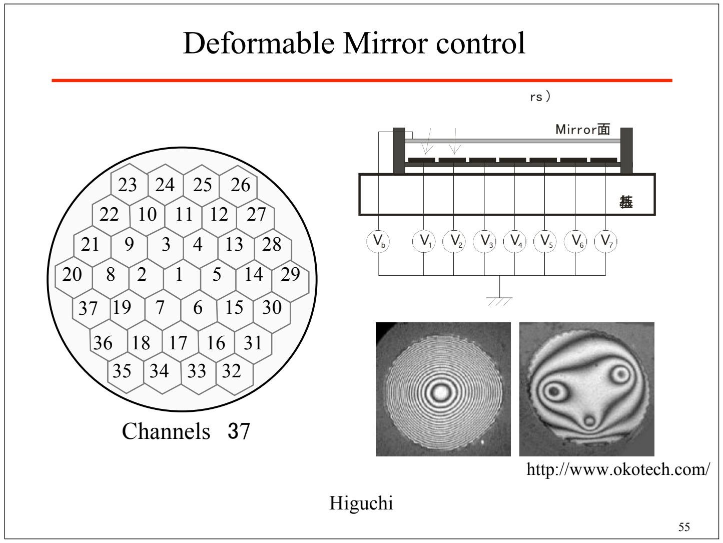


Evolvable Femtosecond Laser System - Higuchi

Laser alignment can be optimized autonomously by genetic algorithms to obtain the maximum output

Advantages:
1. Autonomous Adjustment
2. Portable Size
3. Ultrashort pulse (~10^{-15}sec)

Especially Suitable for
1. Laser Processing for Diamonds and Shape-memory-alloy
2. Medical Treatment (e.g. macula, depilation)

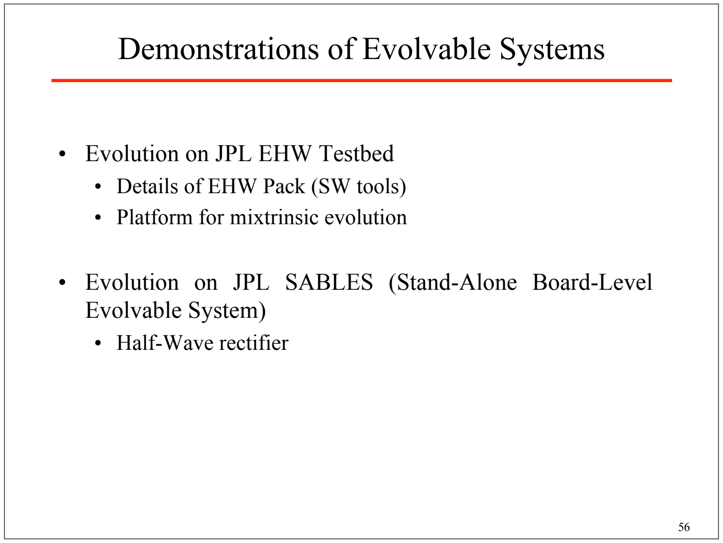


Deformable Mirror control

Channels　37

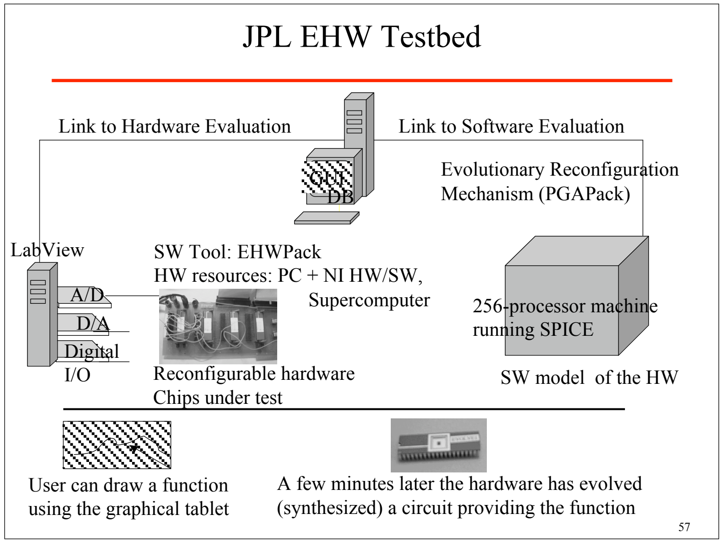
http://www.okotech.com/

Higuchi



Demonstrations of Evolvable Systems

- Evolution on JPL EHW Testbed
  - Details of EHW Pack (SW tools)
  - Platform for mixtrinsic evolution
- Evolution on JPL SABLES (Stand-Alone Board-Level Evolvable System)
  - Half-Wave rectifier

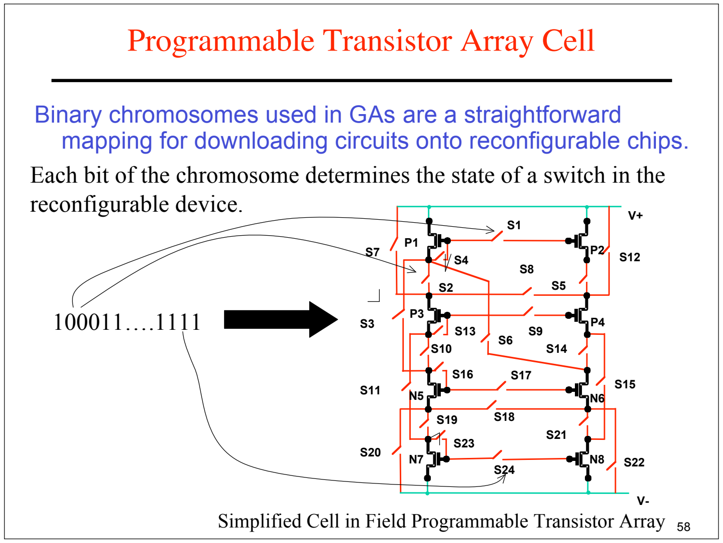


JPL EHW Testbed

Link to Hardware Evaluation
Link to Software Evaluation
GUI
DB
Evolutionary Reconfiguration Mechanism (PGAPack)
LabView
A/D
D/A
Digital I/O
SW Tool: EHWPack
HW resources: PC + NI HW/SW,
Supercomputer
256-processor machine running SPICE
Reconfigurable hardware Chips under test
SW model of the HW
57

User can draw a function using the graphical tablet

A few minutes later the hardware has evolved (synthesized) a circuit providing the function

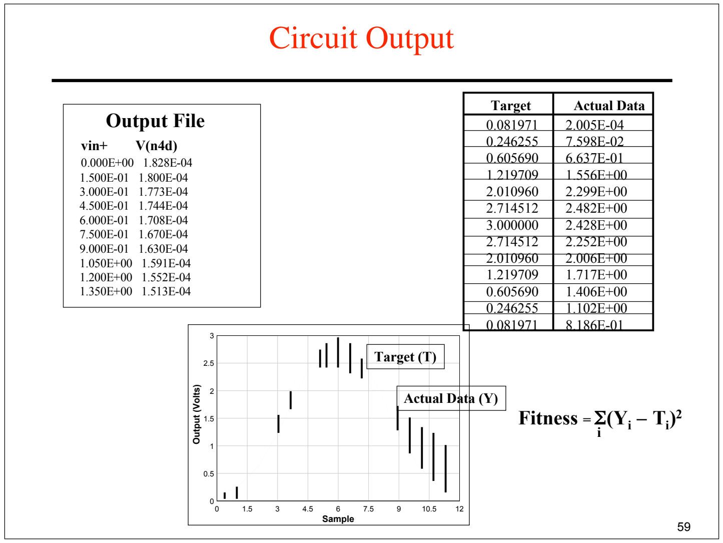


Programmable Transistor Array Cell

Binary chromosomes used in GAs are a straightforward mapping for downloading circuits onto reconfigurable chips.

Each bit of the chromosome determines the state of a switch in the reconfigurable device.

Simplified Cell in Field Programmable Transistor Array



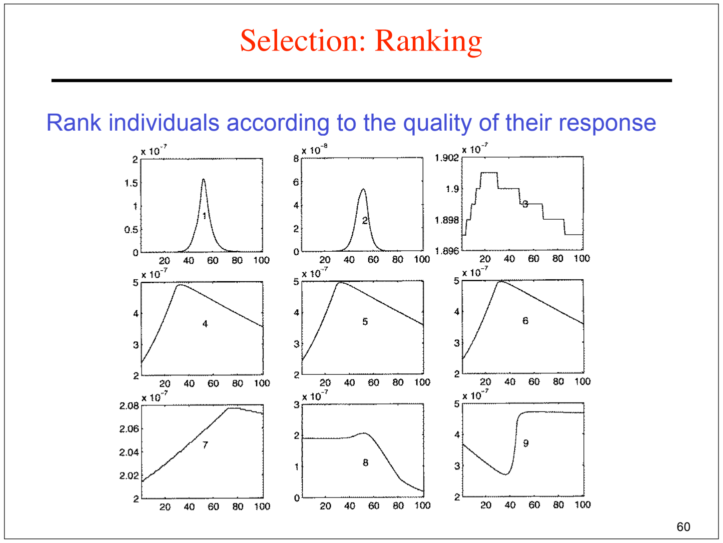
Circuit Output

Output File

```
vin+        V(n4d)
0.000E+00   1.828E-04
1.500E-01   1.800E-04
3.000E-01   1.773E-04
4.500E-01   1.744E-04
6.000E-01   1.708E-04
7.500E-01   1.670E-04
9.000E-01   1.630E-04
1.050E+00   1.591E-04
1.200E+00   1.552E-04
1.350E+00   1.513E-04
```

Target	Actual Data
0.081971	2.005E-04
0.246255	7.598E-02
0.605690	6.637E-01
1.219709	1.556E+00
2.010960	2.299E+00
2.714512	2.482E+00
3.000000	2.428E+00
2.714512	2.252E+00
2.010960	2.006E+00
1.219709	1.717E+00
0.605690	1.406E+00
0.246255	1.102E+00
0.081971	8.186E-01

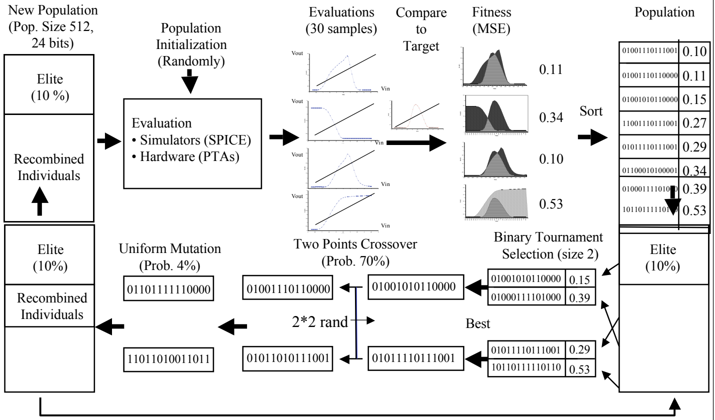
$$\text{Fitness} = \sum_i (Y_i - T_i)^2$$



Selection: Ranking

Rank individuals according to the quality of their response

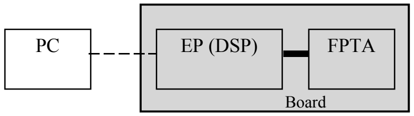
Evolutionary algorithms visualized



- On-chip evolution and fault-recovery: movie clip

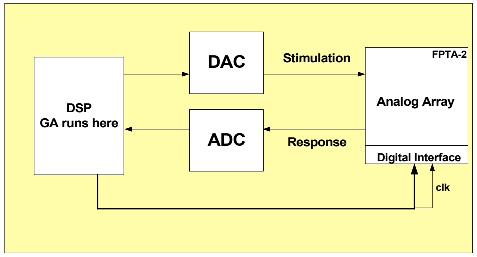
Structure of SABLES

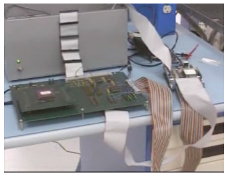
- Reconfiguration mechanism (evolutionary algorithm): TI DSP;
- Reconfigurable Hardware: Field Programmable Transistor Array (FPTA2)
- Performance: 1-2 orders of magnitude reduction in memory, 4+ orders of magnitude improvement in speed compared to systems evolving in simulations.



8”x8”x3”

SABLE details

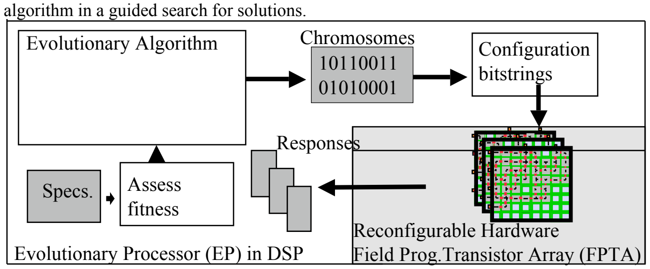




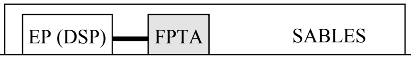
- FPTA – fast evaluation compared to simulation of SPICE netlist
- DSP + FPTA
  - Fast download for evaluation of individuals
  - Good architecture for moving to a self-reconfigurable system-on-a-chip
    - Fault-tolerant solution on a chip
    - Sensors, actuators

Evolution on SABLES

Candidate configurations are tested on-chip; the best ones are modified by the evolutionary algorithm in a guided search for solutions.



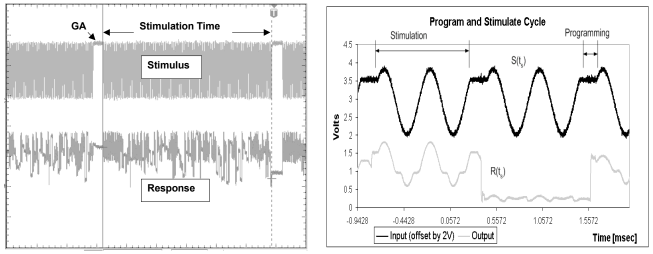
Information flow and implementation with DSP/FPTA chip pair



Speed. Portability. Stand-alone. Autonomy?

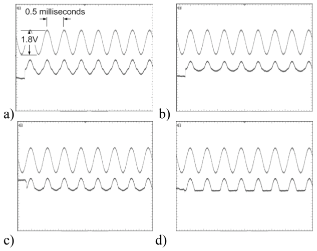
Evolution on SABLES (Half-wave rectifier)

•Excitation input of 2kHz sine wave of amplitude 2V; 20-second experiment
•9% elite percentage; 70% crossover; 4% mutation;100 individuals population;

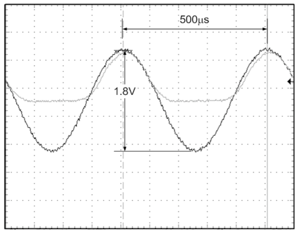


Stimulus-response waveforms during the evaluation of a population in one generation (left) and for 2 individuals in the population (right)

Half-wave rectifier evolution waveforms



Best individual of generation a) 1, b) 5, c) 50 d) 82

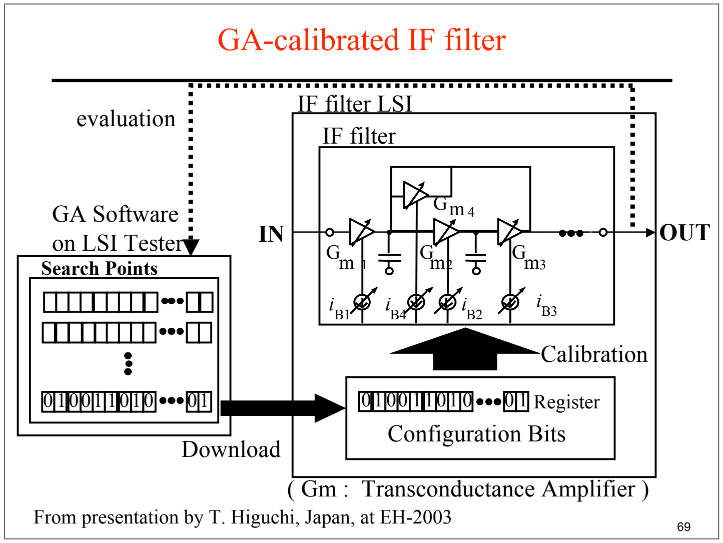


Solution at generation 82
(after 8200 circuits evaluated in ~8s)

On-chip evolution on SABLES: movie clip

Application Examples

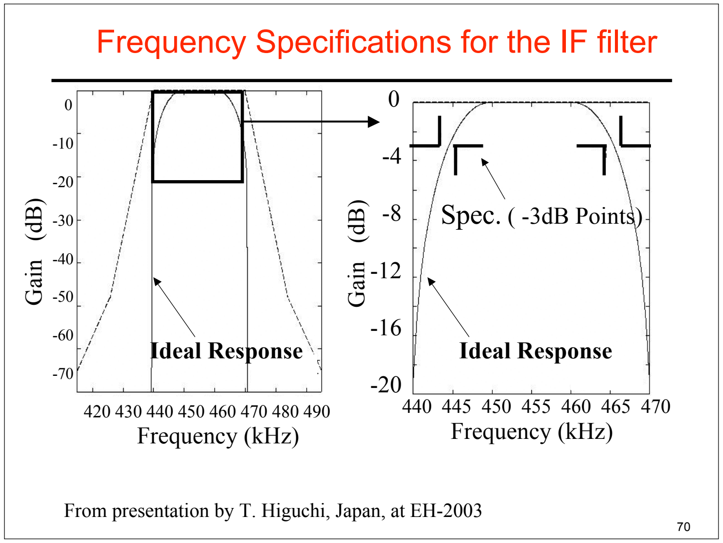
- Design for post-fab compensation/calibration
- Design for implementation/fabrication
- Compensation at extreme temperatures
- Fault-tolerance and fault-recovery
- Evolvable antennas
- Adaptive filters
- Evolution of controllers



GA-calibrated IF filter

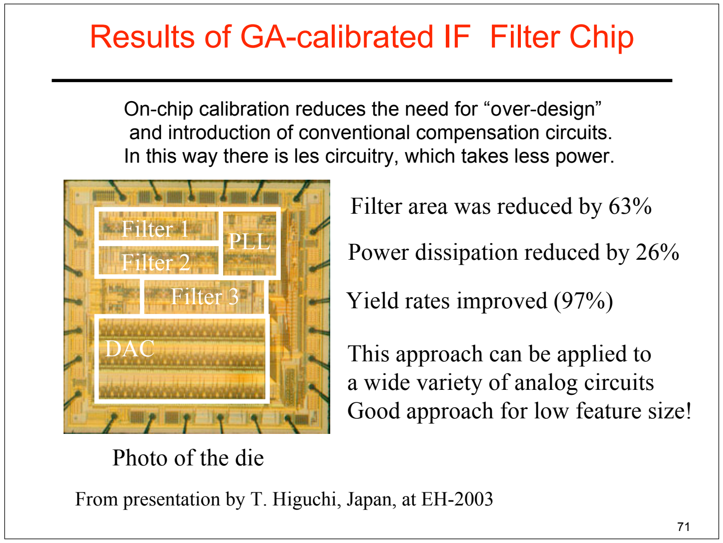
(Gm : Transconductance Amplifier)

From presentation by T. Higuchi, Japan, at EH-2003



Frequency Specifications for the IF filter

From presentation by T. Higuchi, Japan, at EH-2003



Results of GA-calibrated IF Filter Chip

On-chip calibration reduces the need for “over-design” and introduction of conventional compensation circuits. In this way there is les circuitry, which takes less power.

Filter area was reduced by 63%

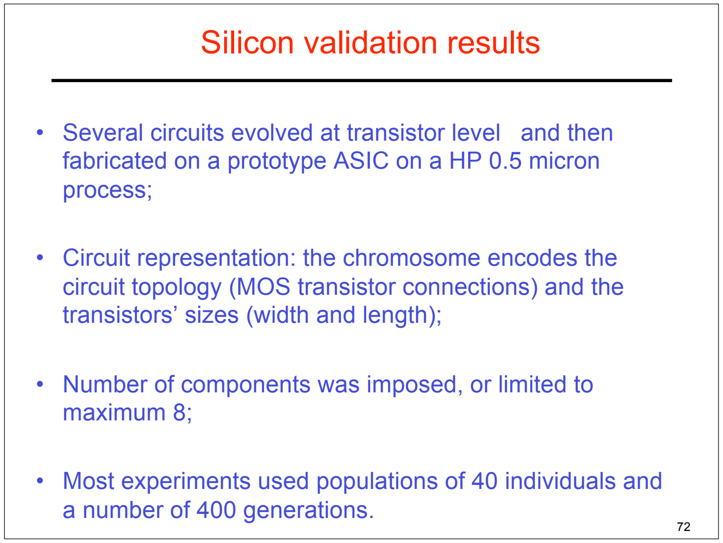
Power dissipation reduced by 26%

Yield rates improved (97%)

This approach can be applied to a wide variety of analog circuits
Good approach for low feature size!

Photo of the die

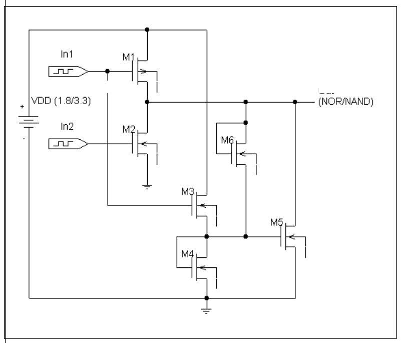
From presentation by T. Higuchi, Japan, at EH-2003



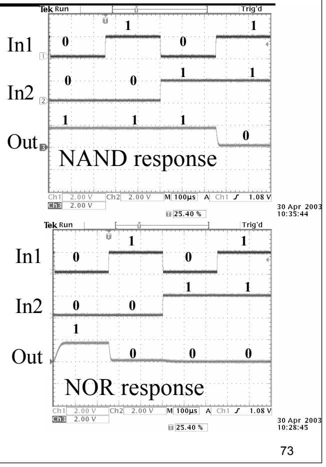
Silicon validation results

- Several circuits evolved at transistor level and then fabricated on a prototype ASIC on a HP 0.5 micron process;
- Circuit representation: the chromosome encodes the circuit topology (MOS transistor connections) and the transistors’ sizes (width and length);
- Number of components was imposed, or limited to maximum 8;
- Most experiments used populations of 40 individuals and a number of 400 generations.

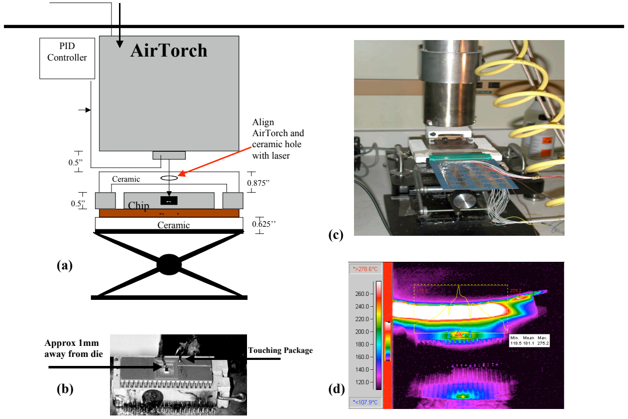
Silicon validation results



Evolved circuit



High temperature testbed



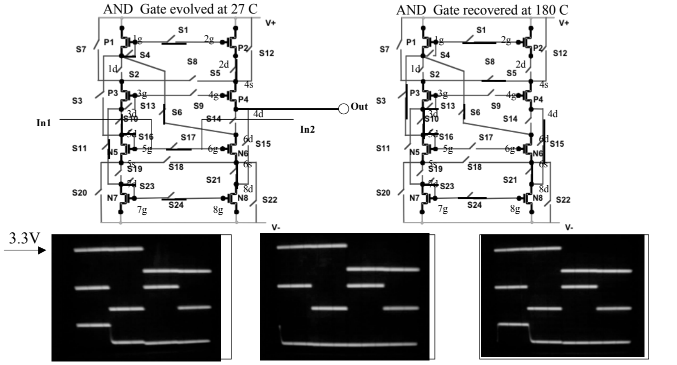
(a). High temperature experimental setup with heat pump and chip under test
(b). Temperature measurement with thermocouples above and below the die
(c) Picture of the apparatus (d)Photo of the heated chip with infrared camera

Expanding the temperature operational range through circuit reconfiguration

Evolution can automatically (on-chip, in-situ) find circuit solutions that recover lost functionality, thus expanding the operational envelope of current devices

AND Gate evolved at 27 C

AND Gate recovered at 180 C



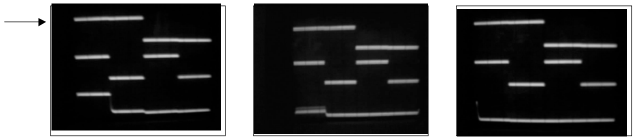
Evolved@27C

Degradation as measured@240

Recovered by evolution @240C

Solutions evolved are point solutions; continuous monitoring and evolution is needed

3.3V Compliant AND circuit evolved at 27C degrades as temperature increases

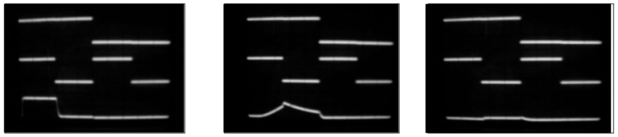


Evolved@27C

Measured@100

Measured @240C

Evolved at 240C becomes compliant; this circuit degrades when temperature decreases



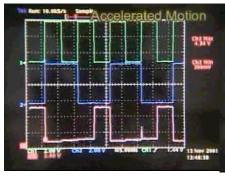
Evolved@240C

Measured@150C

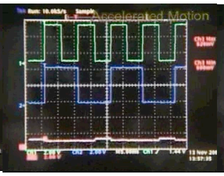
Measured @137C

Coping with faults and degradation in extreme environments

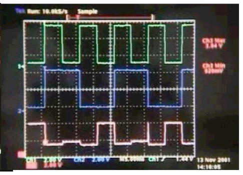
Evolution can recover functionality of circuits affected by faults and degradation, by finding a new circuit bypassing the fault or using damaged components in a different configuration. Experiments at low (-196 C) and high (>+300 C) demonstrate that electronic functions altered by temperature can be recovered through reconfiguration.



Original NOR gate at 27 C



Degraded NOR gate at 326 C

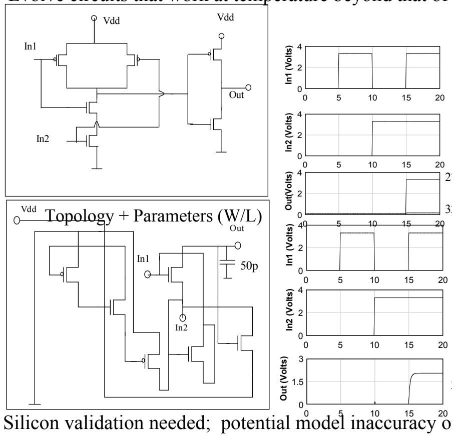


NOR gate recovery at 329C

Recovering functionality at high temperature: movie clip

Circuits designed specifically for high temperatures (only)

Evolve circuits that work at temperature beyond that of conventional cells

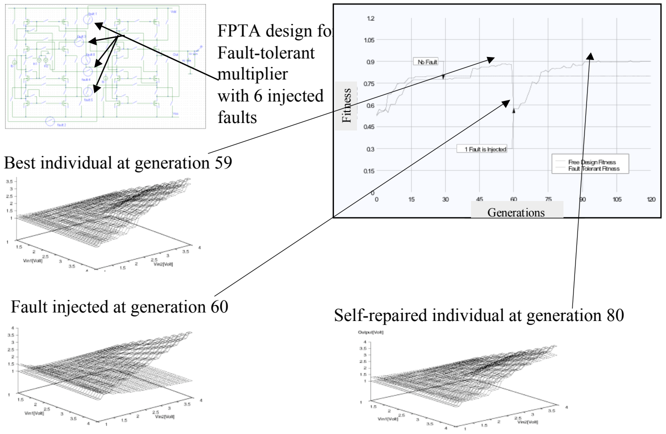


Conventional AND gate deteriorates at 320oC

Evolution synthesized AND gate operating at 320oC.

Silicon validation needed; potential model inaccuracy of the model

Self-repair of Multiplier Circuits



FPTA design for Fault-tolerant multiplier with 6 injected faults

Best individual at generation 59

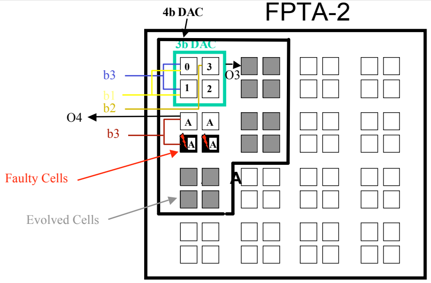
Fault injected at generation 60

Self-repaired individual at generation 80

Self-repair of Digital to Analog Converters

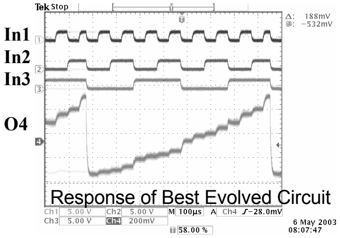
- Hardware experiments using FPTA-2 chip;
- Evolve first a 2-bit DAC, using it as a building block to evolve a 3-bit DAC, and reusing it to evolve a 4-bit DAC.
- Total number of FPTA cells: 20
  - 4 cells mapping a previously evolved 3-bit DAC (evolved from a 2-bit DAC);
  - 4 cells mapping human designed Operational Amplifier (buffering and amplification);
  - 12 cells have their switches' states controlled by evolution.
- Fault application: open all the switches of 2 cells of the evolved circuit;

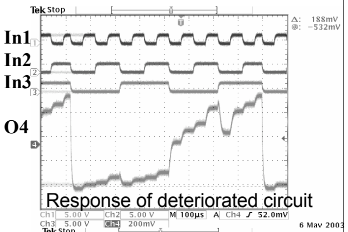
Self-repair of Digital to Analog Converters



- Faulty Cells (Black): All switches opened (stuck-at 0 fault);
- 3-bit DAC Cell: Cells ‘0’, ‘1’, ‘2’ and ‘3’ map the previously evolved 3-bit DAC, whose output is O3.
- OpAmp Cell (Label ‘A’): constrained to OpAmp implementation
- Evolved Cell (Grey): switches’ states controlled by evolution.
- O4 is the output of the 4-bit DAC.

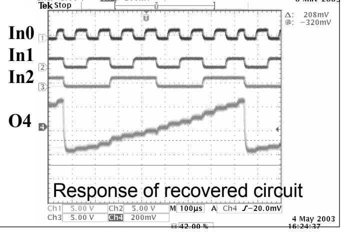
Fault-Tolerant data converter (4bit DAC): responses





Recovery by Evolution of 4bit DAC:

- 12 Cells used by evolution
- 2 Cell constrained to OpAmp
- 4 Cells constrained to 3bit DAC
- 2 Faulty Cells
- Input:10 kHz – Sampling: 20 kHz
- Full-Scale Output Voltage: 0.6 Volt

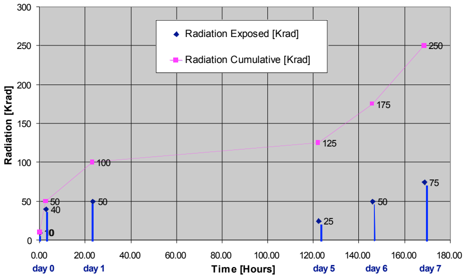


Radiation Tolerant Experiments: Protocol

- Total Ionization Dose (TID)
- Radiation type: electrons Beam from accelerator
- High Radiation Rate (HRD): 300 rad/sec
- Particle energy: 1MeV
- Bias: chip power on and input/output connect to Vdd and Gnd.
- Circuit Level Analysis: Rectifier, NAND, 4bits DAC
- Cumulative Effect Expected:
  - Shift of threshold voltage (silicon dioxide effect)
  - Leakage current (Field Oxides effect)

Radiation Experiments: Circuits

- Cumulative Radiation: up to 250 Krad

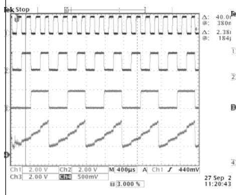


- Circuits Tested:
  - Switching Element (Transmission Gates): Recovery of Identity at 250Krad
  - Analog Signal: Recovery of Rectifier at 100, 175 and 250Krad
  - Logic Signal: Recovery of NAND gate at 175 and 250 Krad
  - Mixed Signal: Recovery of 4bits DAC at 175 and 250 Krad

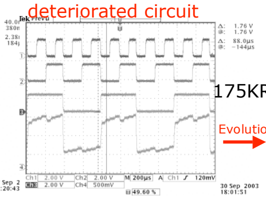
FPTA-B1 Chip: 4 Bits DAC

Response of deteriorated circuit

Recovered by Evolution



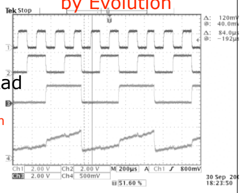
B1- 100Krad – 27Sep – 10:21am



B1- 175Krad – 30Sep – 5:02pm

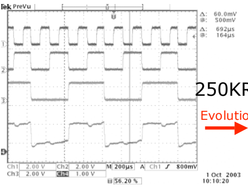
175KRad

Evolution



B1-175Krad –30Sep-5:24pm

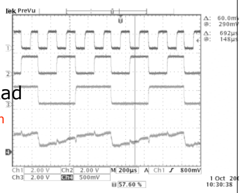
- Incremental radiation with annealing



B1- 250Krad – 1 Oct - 9:10am

250KRad

Evolution



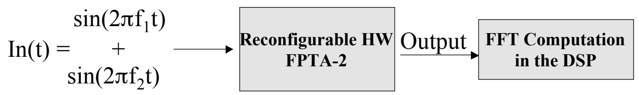
B1- 250Krad -1 Oct-9:31pm

Evolution of Filters

- Binary representation used both in simulation and HW experiments:
  - Simulation experiments using SPICE models of the first FPTA chip;
  - Hardware experiments using FPTA-2 chip;
- Circuits evaluated in the frequency domain:
  - Simulation: small signal analysis in SPICE;
  - Real hardware: FFT of the circuit transient response.

Filter Evolution in Hardware

- Reconfigurable device: FPTA-2
- Experiments performed on SABLES platform: about 5 minutes evolution time;



f_1: Filter Cut-off frequency

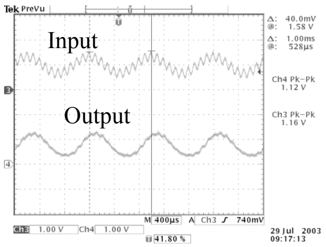
f_2: Stop-band starting frequency

Fitness function maximizes output FFT component at f_1 and minimizes the one at f_2.

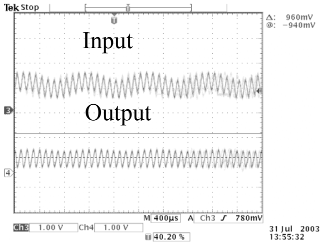
Filter Evolution in Hardware

- Low-pass ($f_{1/2}$ = 1kHz/10kHz) and high-pass filters ($f_{2/1}$ = 1kHz/10kHz);
- Use of 10 cells of the FPTA-2 chip;

Evolved Low-Pass Filter

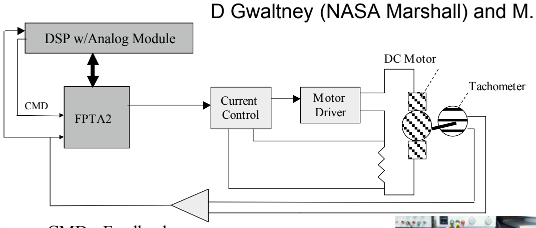


Evolved High-Pass Filter



Evolvable Controllers

D Gwaltney (NASA Marshall) and M.I.Ferguson (JPL)

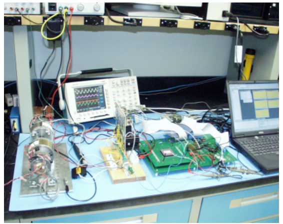


Second motor operated as a generator and used to provide load

Error = CMD - Feedback

Experimental configuration

The evolved controllers use two adjacent cells in the FPTA to perform a similar function to four op-amps, a collection of 12 resistors and one capacitor. Evolved controller has good response to inputs of varying amplitude and frequency and varying load



System Aspects

- Integration
- Need for upfront complete specifications
- Behavioral vs structural specification
- Languages for evolvable hardware
- Verification and validation
- On-line vs off-line evolution
- Techniques to reduce evaluation time

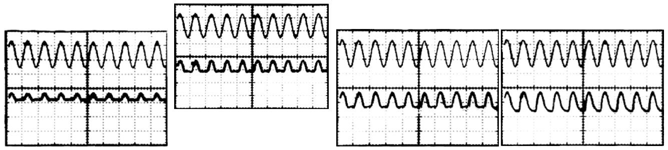
Revisiting Challenges and open Problems

- It is still hard to evolve complex circuits, although we can evolve now orders of magnitude faster.
- It is hard to prove a solution is stable, robust, and hard to predict its behavior outside the domain in which it was evolved.
- The difficulty is in writing fitness functions and guiding evolution for complex systems
- Complete upfront Specifications are required

• The challenge of conventional design is replaced with that of designing an evolutionary process that automatically performs the design in our place. *This may be harder than doing the design directly*, but makes autonomy possible.

On-chip search may suffer from Transient Solutions and Memory Effects

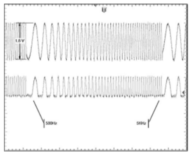
- Circuit under evaluation my require a certain amount of time to achieve steady state, while faster evaluation may evaluate a transient behavior
- FPTA-state dependence: Behavior exhibited in the evaluation can be influenced by the circuit downloaded previously ;
- Artifacts due to parasitic as well as static capacitors in the chip which can be charged during one configuration period and not discharged before the next configuration is tested;
- Observation: GA usually eliminates transient solutions after some generations.



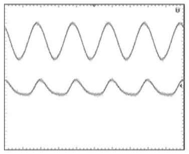
Example of transient behavior. Degradation shown over ~ 1 s.

Solution is guarantied only where tested

Circuit evolved at 2kHz does not work at more than 10kHz



Response of the half-wave rectifier for a frequency sweep from 500Hz to 5kHz(left).



Deteriorated response at 50kHz.

Circuit behavior should be evaluated for the overall frequency domain in which it is expected to work

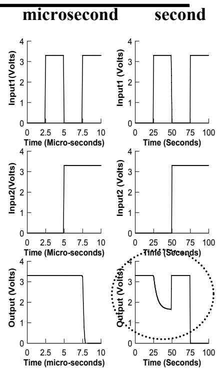
Evolution does not work on implicit assumptions

Evolutionary design requires explicit formulation of assumptions often implicit to human designers. For example:

- evolved logic gates tested with slow signals were naturally slow
- Evolution in small timescales: transient solutions;
- Evolution in large timescales: slow gates;

Solutions:

- Mixtrinsic evolution: using combined excitation modes;
- Increase the duration of transient analysis to 'catch' operating point;
- Decrease step of transient analysis: check circuit behavior after transition;
- Increase output load to ensure a fast gate.

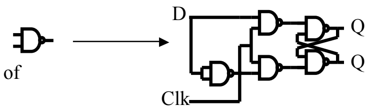


Evolved NAND gate evaluated in the timescale of microsecond (until 10^{-5} sec)

Challenges …and how we address them

- Scalability
- Existence/convergence
of optimal solution
- Satisfying real world requirements
  - loads, power
- Low reliability/safety of evolved solution
- Understandability
- Complete specifications
- Hardware Artifacts

- Hierarchy
- Representations, adaptive GA parameters
- Smart fitness

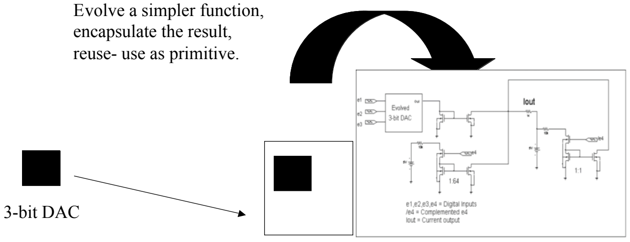


- Evolve sensors rather than controls

Addressing complexity by hierarchical design

- Design and reuse

  Evolve a simpler function, encapsulate the result, reuse- use as primitive.

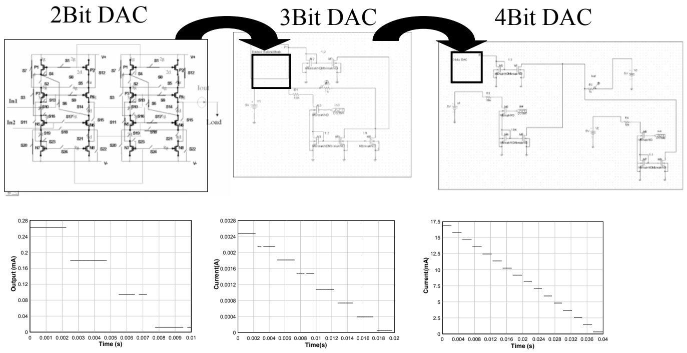


4-bit DAC

- Use a standard higher-level building block (e.g. OA)

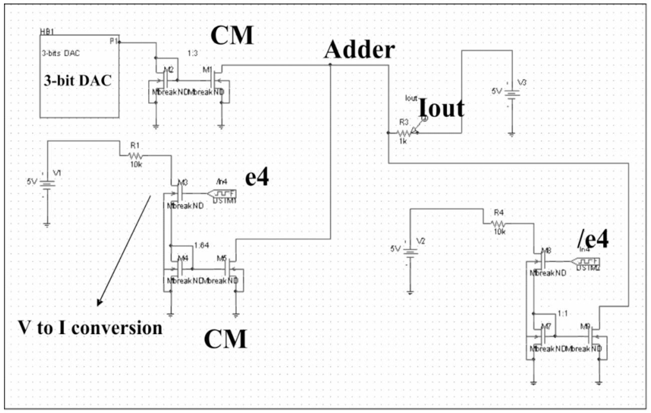
Still, it's not always clear how to select higher-level BB or how to encapsulate sub-circuits obtained by evolution.

Divide and conquer: evolving a scalable design

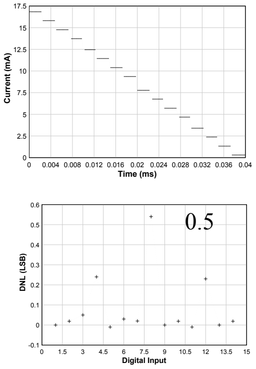


We demonstrated a scalable approach based on encapsulation and block reuse

4-bit DAC



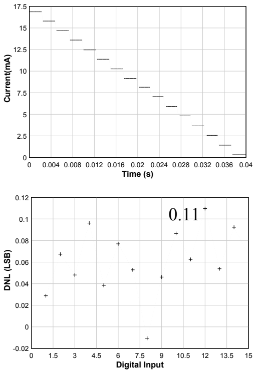
4-bit DAC Response



Circuit Response

Parametric Optimization (W,L)

Differential Non-Linearity

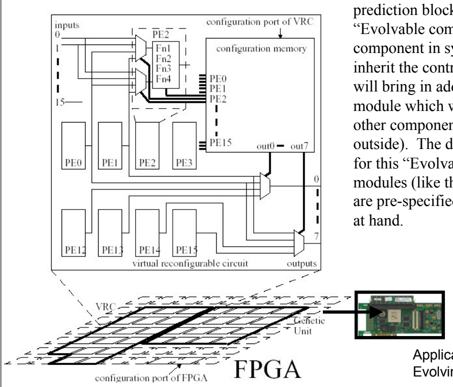


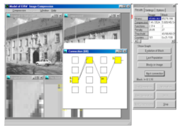
EHW at IP core level: Virtual reconfigurable circuit with genetic unit in FPGA - Sekanina

http://www.fit.vutbr.cz/~sekanina/ehw

Make EHW independent of target platform

Imagine you could replace a fixed components of a system with an adaptive one (say an adaptive prediction block in a compression system). An "Evolvable component" would substitute the fixed component in system level description; it would inherit the controls and interfaces of fixed one but will bring in addition a genetic unit and a fitness module which would use information/feedback from other components in the system (and possibly from outside). The design tool would generate the HDL for this "Evolvable component" for which specific modules (like those implementing genetic operators) are pre-specified and can be tailored for the problem at hand.



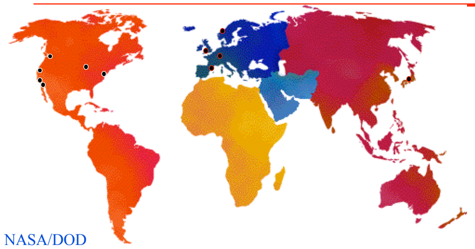


Applications in image processing
Evolving filters, circuits for image compression

Resources for EHW Engineers

- Events
  - EHW-dedicated conferences
  - EHW-related events
- A guide to published literature and on-line resources
  - Journals
  - Books
  - Selected Articles
  - On line resources
- A guide to tools
  - Software
  - Hardware
- A guide to people and programs
  - Research topics, labs and individual researchers
  - Sponsors and funding programs

EHW dedicated Conferences



Conf with sessions on EHW: GECCO, CEC

NASA/DOD
Workshop/Conf on Evolvable Hardware-EH

International Conference
On Evolvable Systems-ICES

	‘95	’96	’98	’99	’00	’01	’02	’03	’04	’05
EH				EH-99 Pasadena	EH-00 Palo Alto	EH-01 Long Beach	EH-02 DC	EH-03 Chicago	EH-04 Seattle	EH-05 Wash DC
ICES	TEH Lausanne Switzerland	ICES Tokyo Japan	ICES2 Lausanne Switzerland		ICES3 Edinburgh UK	ICES4 Tokyo Japan		ICES5 Trondheim Norway		ICES6 Barcelona Spain

EHW-dedicated conferences

Evolvable Hardware Conferences:

Proceedings of 1999, 2000, 2001, 2002, 2003, 2004 NASA/DoD Conferences (IEEE Computer), (http://computer.org/)

- **Forthcoming:**

ICES 2005 The 6th International Conference on Evolvable Systems: From Biology to Hardware
Barcelona, Spain, June 2005
Proceedings – Lecture Notes in Computer Science, Springer

2005 NASA/DOD Conference on Evolvable Hardware
Washington DC, June 29- July 12005

EHW-related Events

Evolutionary Computation Conferences:

2005 IEEE Congress on Evolutionary Computation (CEC 2005)
Sept 2-5, Edinburgh, Scotland
http://www.cec2005.org/

2005 Genetic and Evolutionary Computation COnference (GECCO - 2005)
June, 25-29 Washington DC
http://www.isgec.org/gecco-2005/

EHW-related Events

Design Automation Conferences:

41st Design Automation Conference (DAC)
June 7-11 , 2004, San Diego CA
http://www.dac.com/

International Conference on Computer Aided Design (ICCAD-2004)
November 7-11, 2004 S. Jose CA
http://www.iccad.com

Design Automation and Test Europe (DATE-2004)
February 16-20, 2004 Paris
http://www.date-conference.com

Journals

- Genetic Programming and Evolvable Machines:
  http://www.kluweronline.com/issn/1389-2576
- IEEE Transactions on Evolutionary Computation:
  http://www.ieee-nns.org/
- Evolutionary Computation Journal (MIT Press) :
  http://www.mitpress.mit.edu/EVCO/
- International Journal of SMART ENGINEERING SYSTEM DESIGN
  Cihan Dagli (Ed), http://web.umr.edu/~dagli

Special issues on EHW in following journals:

- Soft Computing Journal, Special Issue on Evolvable Hardware
  Adrian Stoica (Ed) http://www.springer.de
- IEE Proceedings Computer-Digital Techniques
  Special Issue on Evolvable Hardware, Andy Tyrrell (Ed)
  http://www.iee.org/Publish/Journals/ProfJourn/Proc/CDT/evolvable_hardware.pdf

Books

- Thompson, A., “Hardware Evolution: Automatic design of electronic circuits in reconfigurable hardware by artificial evolution”, Springer-Verlag, 1998, http://www.cogs.susx.ac.uk/users/adrianth/ade.html
- Zebulum et Al, “Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms”, CRC Press, 2001 http://www.crcpress.com/shopping_cart/products/product_detail.asp?sku=0865
- Sekanina, L., Evolvable Components From Theory to Hardware Implementations”, Springer, 2003, http://www.springer.de/cgi-bin/search_book.pl?isbn=3-540-40377-9&cookie=done
- John Koza, “*Genetic Programming: On the Programming of Computers by Means of Natural Selection”* published by The MIT Press , 1992;
- John Koza, “*Genetic Programming II: Automatic Discovery of Reusable Programs”* published by The MIT Press, 1994.
- John Koza, “*Genetic Programming III: Darwinian Invention and Problem Solving”* published by Morgan Kaufmann Publishers, 1999.
- John Koza, “*Genetic Programming IV*: *Routine Human-Competitive Machine Intelligence”* (with Martin A. Keane, Matthew J. Streeter, William Mydlowec, Jessen Yu, and Guido Lanza) published by Kluwer Academic Publishers, 2003.

Books

- Goldberg, D., “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1989.
- Holland, J., “Adaptation in Natural and Artificial Systems”, University of Michingan Press, Ann Arbor, EUA, 1975.
- Higuchi, T. , “Evolvable Hardware and its Applications”, chapter in “Computational Intelligence The Expert Speak” by Fogel and Robinson, IEEE Press, 2003.
- Miller, J. F., Thomson, P., and Fogarty, T., “Designing Electronic Circuits Using Evolutionary Algorithms. Arithmetic Circuits: A Case Study”, chapter 6 in Genetic Algorithms Recent Advancements and Industrial Applications. Editors: D. Quagliarella, J. Periaux, C. Poloni and G. Winter, Wiley, 1997.

Selected articles

- **Evolvable Hardware:**
- **Explorations in Design Space: Unconventional Electronics Design Through Artificial Evolution**
  Thompson, A. , Layzell , P., Zebulum, R.S., published at: IEEE Transactions on Evolutionary Computation, Special Issue in "Evolvable Hardware", Moshe Sipper (Ed), pp.167-196, V.3, N.3, September, 1999.

- **What's AI done for me lately? Genetic programming's human-competitive results**
  Koza, J.R.; Keane, M.A.; Streeter, M.J.;
  Intelligent Systems, IEEE [see also IEEE Expert] , Volume: 18 Issue: 3 , May-June 2003
  Page(s): 25 -31

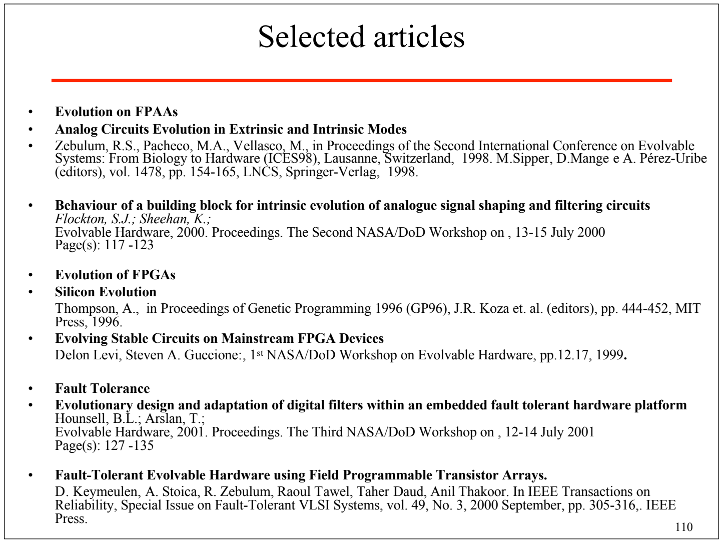
- **A circuit representation technique for automated circuit design**
  Lohn, J.D.; Colombano, S.P.;
  Evolutionary Computation, IEEE Transactions on , Volume: 3 Issue: 3 , Sept. 1999
  Page(s): 205 -219

- **Reconfigurable devices**
- **Embedded reconfigurable array targeting motion estimation applications**
  Khawam, S.; Arslan, T.; Westall, F.;
  Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on , Volume: 2 , May 25-28, 2003
  Page(s): 760 -763

- **Reconfigurable VLSI Architectures for Evolvable Hardware: from Experimental Field Programmable Transistor Arrays to Evolution-Oriented Chips**
  Adrian Stoica, Ricardo Zebulum, Didier Keymeulen, Raoul Tawel, Taher Daud, and Anil Thakoor.. IEEE Trans. on VLSI, IEEE Press , Volume 9, Number 1, ISSN 1063-8210, pp. 227-232, Feb. 2001.

- **Multi-criteria Optimization**
- **An Overview of Evolutionary Algorithms in Multiobjective Optimization**
  Fonseca, C. M, Fleming P. J., Evolutionary Computation, MIT Press, Vol. 3, No.1, pp.1-16, 1995.

- **Digital Filter Design Using Multiple Pareto Fronts**
  Thorsten Schnier, Xin Yao, Pin Liu, 2001 NASA/DoD Workshop on Evolvable Hardware, pp.136-145.



Selected articles

- **Evolution on FPAAs**
- **Analog Circuits Evolution in Extrinsic and Intrinsic Modes**
- Zebulum, R.S., Pacheco, M.A., Vellasco, M., in Proceedings of the Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98), Lausanne, Switzerland, 1998. M.Sipper, D.Mange e A. Pérez-Uribe (editors), vol. 1478, pp. 154-165, LNCS, Springer-Verlag, 1998.

- **Behaviour of a building block for intrinsic evolution of analogue signal shaping and filtering circuits**
  Flockton, S.J.; Sheehan, K.;
  Evolvable Hardware, 2000. Proceedings. The Second NASA/DoD Workshop on , 13-15 July 2000
  Page(s): 117 -123

- **Evolution of FPGAs**
- **Silicon Evolution**
  Thompson, A., in Proceedings of Genetic Programming 1996 (GP96), J.R. Koza et. al. (editors), pp. 444-452, MIT Press, 1996.
- **Evolving Stable Circuits on Mainstream FPGA Devices**
  Delon Levi, Steven A. Guccione:, 1st NASA/DoD Workshop on Evolvable Hardware, pp.12.17, 1999**.**

- **Fault Tolerance**
- **Evolutionary design and adaptation of digital filters within an embedded fault tolerant hardware platform**
  Hounsell, B.L.; Arslan, T.;
  Evolvable Hardware, 2001. Proceedings. The Third NASA/DoD Workshop on , 12-14 July 2001
  Page(s): 127 -135

- **Fault-Tolerant Evolvable Hardware using Field Programmable Transistor Arrays.**
  D. Keymeulen, A. Stoica, R. Zebulum, Raoul Tawel, Taher Daud, Anil Thakoor. In IEEE Transactions on Reliability, Special Issue on Fault-Tolerant VLSI Systems, vol. 49, No. 3, 2000 September, pp. 305-316,. IEEE Press.

Selected articles

- **HW implementations of Evolutionary Processors**
- **An FPGA-Based Genetic Algorithm Machine**
  Shackleford B., Carter D., Snider G, Okushi E., Yasuda M., Koizumi H., Seo K., Iwamoto T., Yasuura H. FPGA 2000, Eighth ACM International Symposium on Field-Programmable Gate Arrays, Feb. 10-11, 2000.
- **An Evolvable Hardware Platform based on DSP and FPTA**
  Ferguson, M.I., Stoica A., Zebulum R., Keymeulen D. and Duong, V. Proceedings of the Genetic and Evolutionary Computation Conference, July 9-3, 2002 New York, NY;

- **Evolution on FPTAs**
- **Evolving Circuits in Seconds: Experiments with a Stand-Alone Board Level Evolvable System**
  Stoica, A., Zebulum, R. S., M.I. Ferguson, Keymeulen, D., Duong, V., 2002 NASA/DoD Conference on Evolvable Hardware, Alexandria Virginia, USA, 2002, pp. 67-74 IEEE Computer Society, ISBN 0-7695-1718-8
- **Intrinsic Evolution of Quasi DC Solutions for Transistor Level Analog Electronic Circuits Using CMOS FPTA Chip**
  Langeheine, J., Meier, K., Schemmel, J., Proceedings of the 2002 NASA/DoD Conference on Evolvable Hardware, pp.75-84, IEEE Computer Press, July, 2002.

- **Evolution of antennas (design)**
- **Evolutionary Design of an X-Band Antenna for NASA's Space Technology 5 Mission**
  J. Lohn, D. Linden, G. Hornby, W. Kraus, A. Rodríguez-Arroyo,
  and S. Seufert, 2003 NASA/DoD Conference on Evolvable Hardware, pp. 155-163, IEEE Computer
- **Fitness Estimations for Evolutionary Antenna Design**
  L. Zinchenko and S. Sorokin, 2003 NASA/DoD Conference on Evolvable Hardware, pp. 164-176, IEEE Computer.
- **Evolving Wire Antennas Using Genetic Algorithms: A Review**
  Derek S. Linden, Edward E. Altshuler , 1st NASA/DoD Workshop on Evolvable Hardware, pp.225.232, 1999, IEEE Computer.

Selected articles

- **Evolution on physical antennas**
- **Optimizing Signal Strength In-Situ Using an Evolvable Antenna System**
  Derek S. Linden   , 2002 NASA/DoD Conference on Evolvable Hardware (EH'02) pp. 147-151, 2002, IEEE Computer.

- **System On a Chip**
- **An evolutionary power management algorithm for SoC based EHW systems**
  Lirong Tian; Arslan, T.;
  Evolvable Hardware, 2003. Proceedings. NASA/DoD Conference on , July 9-11, 2003
  Page(s): 117 -124

- **Multi-objective design strategy for high-level low power design of DSP systems**
  Bright, M.S.; Arslan, T.;
  Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on , Volume: 1 , 30 May-2 June 1999
  Page(s): 80 -83 vol.1

- **Towards evolvable IP cores for FPGAs**
  Sekanina, L.;
  Evolvable Hardware, 2003. Proceedings. NASA/DoD Conference on , July 9-11, 2003
  Page(s): 145 -154

- **Parallel EHW**
- **An EHW architecture for real-time GPS attitude determination based on parallel genetic algorithm**
  Jiangning Xu; Arslan, T.; Qing Wang; Dejun Wan;
  Evolvable Hardware, 2002. Proceedings. NASA/DoD Conference on , 15-18 July 2002
  Page(s): 133 -141

Selected articles

- **Applications**
- **Real-world applications of analog and digital evolvable hardware**
  Higuchi, T.; Iwata, M.; Keymeulen, D.; Sakanashi, H.; Murakawa, M.; Kajitani, I.; Takahashi, E.; Toda, K.; Salami, N.; Kajihara, N.; Otsu, N.;
  Evolutionary Computation, IEEE Transactions on , Volume: 3 Issue: 3 , Sept. 1999
  Page(s): 220 -235
- **An evolvable hardware chip for prosthetic hand controller**
  Kajitani, I.; Murakawa, M.; Nishikawa, D.; Yokoi, H.; Kajihara, N.; Iwata, M.; Keymeulen, D.; Sakanashi, H.; Higuchi, T.;
  Microelectronics for Neural, Fuzzy and Bio-Inspired Systems, 1999. MicroNeuro '99. Proceedings of the Seventh International Conference on , 7-9 April 1999
  Page(s): 179 -186
- **Low Power design**
  Synthesis of low-power DSP systems using a genetic algorithm
  Bright, M.S.; Arslan, T.;
  Evolutionary Computation, IEEE Transactions on , Volume: 5 Issue: 1 , Feb 2001
  Page(s): 27 -40
- **Evolutionary Algorithms**
  From an individual to a population…" Jun He; Xin Yao. Evolutionary Computation, IEEE Transactions, Volume: 6 Issue: 5 , Oct 2002 Page(s): 495-511

On-line resources

EvoNet Tutorials:
http://evonet.dcs.napier.ac.uk/evoweb/resources/flying_circus/tutorials/index.html

Good repository of GA links: http://www.aic.nrl.navy.mil/galist/

EvoWeb, website of EvoNet - the European Network of Excellence in Evolutionary Computing http://evonet.dcs.napier.ac.uk/

Another GA Tutorial http://www.ifs.tuwien.ac.at/~aschatt/info/ga/genetic.html

Evolutionary Multi-Objective Optimization SW http://www.lania.mx/~ccoello/EMOO/
http://xputers.informatik.uni-kl.de/fpl/index_conf.html#evo

Sketchy Tutorial Slides http://lancet.mit.edu/~mbwall/presentations/IntroToGAs/P001.html

Genetic and Evolutionary Algorithms: Principles, Methods and Algorithms
http://www.geatbx.com/docu/algindex.html

Evolutionary Algorithms for MATLAB (incl. Genetic Algorithms and Genetic Programming) http://www.geatbx.com/links/ea_matlab.html

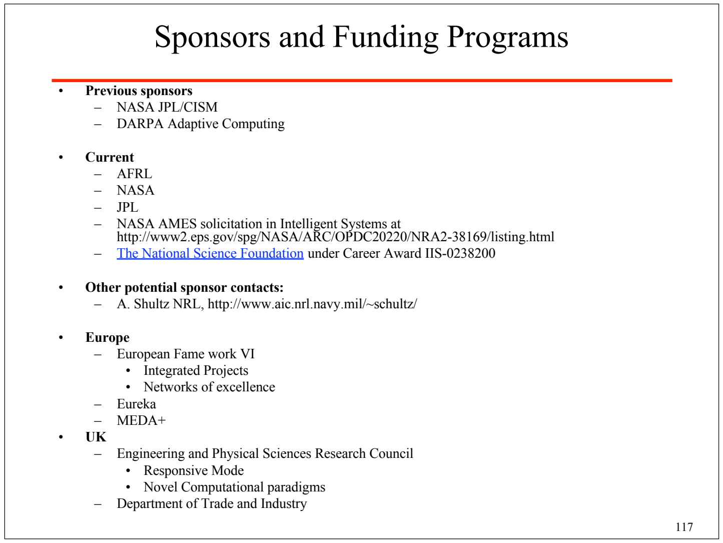
An Overview of Evolutionary Algorithms, Genetic Algorithms and Evolutionary Computing. Darrell Whitley.
http://www.cs.colostate.edu/~genitor/Pubs.html

Guide to Software Tools

- Free Evolutionary SW
  - (evonet.dcs.napier.ac.uk)
- Free Spice
  - Ngspice (sourceforge.net/projects/ngspice)
  - Other (www.repairfaq.org/ELE/F_Free_Spice2.html)
- Simulators, GUI
  - ModelSim (www.model.com)
  - GUI toolkits free/commercial (http://www.atai.org/guitool)
- C/C++
  - GCC (gcc.gnu.org)
  - Rational Rose (OO/UML) (www.rational.com)
- NI
  - National Instruments (www.ni.com) LabView

Guide to Hardware Tools

- FPGA development kits
  - Xilinx (www.xilinx.com) ISE
  - Altera (www.altera.com) MAX+PLUS® II
- FPAA development kits
- Multi-objective synthesis of VLSI Signal and Image Processing Cores
  - www.see.ed.ac.uk/~SLIg
- SoC development kits
  ARM Integrator Solutions,
  - www.arm.com/devtools
- SoC development kits
  - Cadence (www.cadence.com)
  - Mentor Graphics (www.mentor.com)
- Data acquisition boards
  - National Instruments (www.ni.com) LabView
  - Microstar Laboratories (www.mstarlabs.com) DAP Tools
- Microcontrollers, DSP, etc..
  - Keil Software (www.keil.com) DK51
  - Texas Instruments (www.ti.com) Code Composer



Sponsors and Funding Programs

- **Previous sponsors**
  - NASA JPL/CISM
  - DARPA Adaptive Computing
- **Current**
  - AFRL
  - NASA
  - JPL
  - NASA AMES solicitation in Intelligent Systems at http://www2.eps.gov/spg/NASA/ARC/OPDC20220/NRA2-38169/listing.html
  - The National Science Foundation under Career Award IIS-0238200
- **Other potential sponsor contacts:**
  - A. Shultz NRL, http://www.aic.nrl.navy.mil/~schultz/
- **Europe**
  - European Fame work VI
    - Integrated Projects
    - Networks of excellence
  - Eureka
  - MEDA+
- **UK**
  - Engineering and Physical Sciences Research Council
    - Responsive Mode
    - Novel Computational paradigms
  - Department of Trade and Industry