Evolvable Hardware
Toward morphable, adaptive infrastructures of tomorrow

Adrian Stoica
Jet Propulsion Laboratory
California Institute of Technology
adrian.stoica@jpl.nasa.gov
http://cism.jpl.nasa.gov/ehw/darpa

Chevron, 11/9/2000
Objectives

- Overview a set of technologies that promise to make hardware adaptive and evolvable
- Present a vision on their potential impact on infrastructures of tomorrow

- Get you interested in EHW
- See what EHW can do for you (and vice-versa:-)
Outline

- Vision, infrastructure, technologies, EHW
- EHW fundamentals, algorithms, devices, examples
- Fundamental questions in EHW
- Overview of EHW efforts
- Directions of EHW research at JPL
- MEMS, smart devices, distributed intelligence
- Chevron and adaptive technologies for tomorrow
- Final remarks
Vision - Looking at the future

Vision

Infrastructure supporting the vision

Technologies that will create the infrastructure

Future

EHW

MEMS

Distributed intelligence

Present

A. Stoica, 11/9/2000
Looking at the 21st Century...

- Book me a double - with a view of Venus
- Roboplaymates: MyDog and VPal
- Nice legs, are they new?
- Don’t die, stay pretty
- .mars
- Smaller is getting bigger every day (the MEMS revolution)
- Deus ex Silico

* Selected titles of articles from WIRED, Special Issue *The Future Gets Fun Again*, January 2000
Space...the final frontier

View of Venus

.mars

Interstellar exploration -
Long life survivable spacecraft (100 years+)


A. Stoica, 11/9/2000
Global Interaction Medium (GIM)
An evolving, symbiotic Omninet of 2050*

- The HW/SW symbiosis will be morphing and adaptive
- Interconnected info-processing systems could act as global brains

- The GIM will become very closely interfaced with our sensing and thinking, so that we become ourselves the neurons of one or several brains.
- Machines evolve, interconnections evolve... the result - a net whose power goes beyond anything that could be designed

* "Only Connect: From swarms of smart dust to secure collaborative zones the Omninet comes to you", by George Johnson, Wired Special Issue, Jan, 2000
Characteristics of the infrastructure

- Adaptive
- Self-configurable
- Self-healing
- Massive distributed intelligence, self-organizing
- Morphable structures

- Lots of “smarts”:
  - smart skins (tankers that heal their “wounds”)
  - smart walls (optimize comfort with minimal energy consumption)
  - smart implants
- And above all...a wired, wired world
Today's technology supporting the vision of tomorrow's infrastructure

- Evolvable Hardware
- Amorphous Computing
- Polymorphous Computing
- Moletronics
- Spintronics
- Nanotechnology
- MEMS
- Biometrics
- Prosthetics
- Bio, bio, bio
- SW, Tools, ...

Molecular nanomechanics:
- DNA, mechanical, chemical, biological

Quantum cellular automata:
- Arrays of quantum dots

Molecular nanoelectronics:
- Chemically-synthesized circuits
Basis for projections of today's technology

- DARPA Tech 2000 Symposium
- ITRS 1999 Int Tech Roadmap for Semiconductors
- Wired
- IEEE 1st International Conference on Humanoids
- Internet

- Technology access
- Humanity needs and dreams

- It is hard to predict, especially the future
A new generation of hardware

Generation changes:
- fixed hardware,
- reconfigurable hardware,
- evolvable hardware

Flexibility, fault-tolerance

Self-reconfigurable, evolvable

Reconfigurable

Fixed HW

2005 - 100nm - BISR, ITRS'99

1st  2nd  3rd

Generation
EHW Driver at JPL: Enable long-life (100+ years) survivable 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
- Radiation impacts
- Temperature variations
- Aging
- Malfunctions, etc.

Versatility: Create new functionality required by changes in requirements or environment

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
Space avionics in 2020

- EHW has the potential to be the underlying technology behind the avionics infrastructure of the space systems for 2020 and beyond. Future avionics may evolve not only electronics but also smart optical/structural/thermal subsystems through reconfiguration and morphing.

- EHW technology will enable:
  - Reconfiguration for multiple functionality of avionics systems using the existing resources.
  - Adaptation for new needed functionality
  - Fault-tolerance and self-healing for recovering functionality by rerouting around damaged components and reusing components with modified/alter Characteristics in new circuit topologies.
  - Autonomous avionics through self-configuration.
EHW fundamentals

- Introduction to EHW
- System-level
- Evolutionary Algorithms
- Reconfigurable devices
- Step-by-step example of circuit evolution
Evolvable Hardware (EHW) =
Reconfiguration Mechanism + Reconfigurable HW

EHW = RH + RM

RM

Mechanisms of transformation
search/optimization techniques

RH

HW that can change

GA + FPGA

EHW

Electronics

Antennas

MEMS

BioMEMS

A. Stoica, 11/9/2000
A mechanism inspired from Nature

“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.

Accelerated evolution, ~ seconds for electronics

Evolvable Hardware

A. Stoica, 11/9/2000
Design to be evolved

Programs

0 While TooFarFromWall
1 Do2
2 MoveForward
3 Do2
4 While In Corridor Range
5 Turn Away From Closest Wall
6 While In Corridor Range
7 Do2
8 Turn Parallel To Closest Wall
9 Move Forward

Model of Hardware

- SPICE Netlist
- VHDL code

Physical Hardware

- Microprocessor

- Evolutionary is Revolutionary!
Evolution of evolvable HW

We are here

Programmable HW
Downloadable SW

Evolution of descriptions of electronic HW

Evolutionary search for a parametric design

Board level EHW

Field Programmable Gate Arrays

Evolvable SOC

Evolvable Systems

IP level

Chip level
EHW implementation: HW/SW

\[ \text{EHW} = \text{RH} + \text{RM} \]

Present: RM in SW

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

Plus
- Fast evaluation
- Low cost for failure

Future: everything seamlessly integrated in HW

A. Stoica, 11/9/2000

Evolvable Hardware
Reconfigurable Hardware

- distinct blocks with extensive interwiring
- switches/routing are programmable
- a permissive environment where connections are created as needed

Elementary block/cell
- Gate
- Adder
- OpAmp
- Passives (R,L,C)
- Neuron
- Transistor
- Nano-electronic Devices

Programmable Logic Devices
- Connected PLD's
  - 10^3+
- Simple PLD
  - 10^2
- F PGA
  - 10^6
- ASIC
  - 10^7
- CA
- Digital NN

Low Level Spec
High Level Spec

Evolvable Hardware
Reconfiguration Mechanisms

- RM: GA, ES, Hill Climbing, Taguchi Methods, etc.
- Most popular searches: population based, use “generate and test” strategies.

**Sketch of a simple GA**

1. Initialize a population of candidate solutions: 10111, 10100
2. Evaluate population
3. Select the best
4. Acceptable solution found?
   - No
   - Yes: Output solution
5. Create a new population based on old one: 10000, 00000
6. Crossover: 11111, 00000
7. Mutation: 10111, 10101

Crossover and mutation are two common genetic operators used in creating a new population.
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
Evolutionary synthesis and adaptation of electronic circuits

Evolutionary Algorithm
Genetic search on a population of chromosomes
- select the best designs from a population
- reproduce them with some variation
- iterate until the performance goal is reached.

Chromosomes
1010011010100
011010110111
1101101101110

Conversion to a circuit description

Control bitstrings

Extrinsic evolution

Models of circuits

Simulators (e.g., SPICE)

Intrinsic evolution

Reconfigurable hardware

Target response

Evaluate individual responses and assess their fitness

Circuit response

Potential electronic designs/implementations compete; the best ones are slightly modified to search for even more suitable solutions
A system-level view - Software

- Simulators: Spice, NEMO, NEC, Diehard
Evolution at device, circuit, and system level

At the *device level*, evolution can assist in creating structures with desired functional characteristics. For example, the synthesis could be of a nanoelectronic device, such as a Resonant Tunneling Diode with the characteristic in Fig. 1 or the less common device with the characteristics in Fig. 2. Some of the parameters that determine this functions can be kept fix, while some (e.g., T1-T5, N1-N2 in Fig. 3) can be used as variables in the genetic search.

At the *circuit level*, evolution can combine devices with specific functional characteristics to achieve an overall functionality. For example, given the choice of devices with characteristics in Fig. 1 and Fig. 2, find the optimal interconnections that provide the function of a 4-input NAND gate with minimum power consumption.

At the *system level*, evolution can be used to bring together heterogeneous building blocks, such as an antenna and associated impedance matching electronics.

While optimal designs may be found individually for each component of the system, when considered functioning together, the optimal points may change and need recalculation.
JPL EHW testbed

Link to Hardware Evaluation

- Database
  Chromosome and circuit info

- LabView
  - A/D
  - Digital I/O
  - D/A

- Reconfigurable hardware
  Chips under test

Link to Software Evaluation

- Evolutionary Reconfiguration Mechanism
  (PGAPack)

- SW Tool: EHWPack
  HWresources: PC + NI HW/SW, Supercomputer

- 256-processor HP Exemplar running SPICE 3f5

- SW model of the hardware

User can draw a function using the graphical tablet
A few minutes later the hardware has evolved (automatically synthesized) a circuit that provides the function
Technology insertion

- PCI
- SOC
- IP

module control;
ALU MCU (out, opA, opB, opcode)
initial
for(I=0; I<=count; I=I+1)
begin ...

Evolvable embedded systems
A world controlled in natural language

Sensor web

High-level specs

Device evolve for adaptation

Spec/requirement

constraints: weather, events
EHW today and tomorrow

- Today the human is actively involved in many steps
- Tomorrow humans will only provide the high-level desired specs

Diagram:

- Fitness
- EA /params
- Stopping criteria
- RH Design
- Spec language
- Compiler
- Fitness
- Self-adapting EA
- Morphable device
- Automatic
Evolutionary algorithms

- Search
- GA/GP
- Algorithm Parameters
- Adaptive EA
- Fitness evaluation
- HW implementation of EA
- Simplifications for HW
Search Techniques

Scope of Evolutionary Algorithms:
Discontinuous, non-differentiable,
Multimodal and noisy response
surfaces.

Complexity of the
Search Space

- Random Search
- Probabilistic Methods (Evolutionary Algorithms,
  Hill Climbing, Simulated Annealing)
- Calculus Based Methods (Gradient Notion)

Efficiency

Specialized
Scheme

Robust Scheme

Random Walk

Problem Type

Combinatorial unimodal multimodal
Evolutionary Algorithms (EAs)

- Classes of EAs:
  - Evolutionary Strategies (ES);
  - Evolutionary Programming (EP);
  - Genetic Algorithms (GA);
  - Genetic Programming (GP);

- Basic components of EAs:
  - Representation;
  - Selection;
  - Crossover;
  - Mutation;
  - Fitness Evaluation Function.
Representations

- Binary (GA);
- Real-values vectors (ES);
- Trees (GPs);
- State Machines (EP).

- Performance is largely determined by the representation;
- Choose the representation that is most suitable for the search algorithm;
- A good representation should be:
  - Simple and compact (small chromosomes);
  - Flexible to map solutions of various sizes and shapes;
Selection

- Based on the principle of survival of the fittest;
- Deterministic in ES and EP;
- Probabilistic in GA and GP

Fitness of an individual proportional to slice in the roulette

Spin the roulette

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

1 - Genetic algorithms

Parents

Offspring

Crossover Points

2 - 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.
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;
- Recent findings: driving force of GAs:
  - GAs performance largely affected by the mutation rate.
Fitness Evaluation Function

- Objective function that evaluate how well each individual performs;
- Standard Method:

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

Fitness \( F \) is computed over \( n \) samples;
\( R_i \) – Individual Response;
\( T_i \) – Target Response;
\( W_i \) – Weight reflecting some knowledge of the problem

Fitness of the best along the generations for a typical GA execution:

Skills to design a proper fitness evaluation function are essential for the application success.
Fitness Evaluation Function

Three kinds of circuit analysis: transient; DC transfer; and small signal analysis.

\[ \text{Fitness} = \sum w_i \cdot e_i^k \]

- \(w\) - weight vector;
- \(e_i\) - error of the output sample \(i\) to the desired response;
- \(k\) - power applied to the error, usually \(k = 2\);
- \(i\) - index related to the time domain, DC transfer domain or frequency domain.

To improve the performance:
- Weight vector components are set according to the particularities of the problem;
- Only some points of the analysis domain are considered, such as: peaks and valleys for DC transfer; cut-off frequency for AC analysis; and particular time intervals for transient analysis;
- Probing internal circuits' points;
- Co-evolution of the weights.
Multi-Objective Optimization

- Analog circuit design is *intrinsically* multi-objective;
- Conventional design usually decomposes synthesis tasks into two sub-tasks: *general performance requirements* (Ex: frequency response) and *specific circuit requirements* (Ex: noise and fault-tolerance);
- The designer may *choose* among a number of solutions provided by the Genetic Algorithm.

General Fitness Expression

\[ \text{Fitness} = \sum (w_i \cdot f_i) \]

- \( w_i \) ⇒ Weight Vector component for objective \( i \);
- \( f_i \) ⇒ Fitness of the objective \( i \).

Enhancements in EAs

- Adaptive mutation rate:
  - Escape local optima by increasing rate of mutation;
- Speciation:
  - Keep diversity by creating sub-populations;
  - Multimodal problems: subpopulations sampling different and interesting solutions to a particular problem.
- Variable Length Representations:
  - Map solutions of different sizes;
  - Evolution of electronic circuits of different sizes.
Evolvable Hardware

- 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 hereconfigurable device.
Evolutionary algorithms visualized

New Population (Pop. Size 512, 24 bits)

<table>
<thead>
<tr>
<th>Elite (10%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recombined Individuals</td>
</tr>
</tbody>
</table>

Population Initialization (Randomly)

- Evaluations (30 samples)
- Compare to Target
- Fitness (MSE)

| 0.11 |
| 0.34 |
| 0.10 |
| 0.53 |

Evaluation
- Simulators (SPICE)
- Hardware (PTAs)

Two Points Crossover (Prob. 70%)

| 0.1010011000 |
| 0.1011001000 |
| 0.1100110100 |

Uniform Mutation (Prob. 4%)

| 0.1010101010 |
| 0.1011010101 |

Recombined Individuals

| 0.1010101010 |
| 0.1011010101 |

| 0.1010011000 |
| 0.1100110100 |

| 0.1010011000 |
| 0.1100110100 |

Population

Elite (10%)

A. Stoica, 11/9/2000

Evolvable Hardware
Reconfigurable devices

- COTS
- Custom made, FPTA
- EORA
- Comparison
- Models and levels of accuracy (SPICE with levels)
- Portability problem
- Mixtrinsic evolution
EHW with COTS

FPGA
Xilinx 6200

FPAA
Motorola MPAA020
➤Switched capacitors

Virtex (Xilinx)

Zetex:
➤Twenty operational amplifiers configured as adders, subtractors, multipliers, rectifiers, etc.

A. Stoica, 11/9/2000
EORA Characteristics

- *programmable granularity* (at least for experimental work in EHW, it appears a good choice to build reconfigurable hardware based on elements of the lowest level of granularity.

- *transparent architectures*, allowing the analysis and simulation of the evolved circuits.

- *robust* enough not to be damaged by any bitstring configuration existent in the search space, potentially sampled by evolution.

- should allow evolution of *both analog and digital functions*.
Comparison w/respect to EORA

- Main features of surveyed devices (Analog)

<table>
<thead>
<tr>
<th>Feature /Device</th>
<th>TRAC</th>
<th>MPAA020</th>
<th>PALMO</th>
<th>EM</th>
<th>Lattice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Granularity</td>
<td>coarse</td>
<td>coarse</td>
<td>coarse</td>
<td>fine</td>
<td>coarse</td>
</tr>
<tr>
<td>Protection</td>
<td>NA</td>
<td>software tools</td>
<td>software/hardware</td>
<td>switches parasitics</td>
<td>NA</td>
</tr>
<tr>
<td>Circuit Download</td>
<td>parallel port</td>
<td>serial port</td>
<td>serial port</td>
<td>ISA bus</td>
<td>serial port</td>
</tr>
<tr>
<td>Proprietary Information</td>
<td>NA</td>
<td>Yes</td>
<td>NA</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Search Space</td>
<td>NA</td>
<td>~2^{300}/cell</td>
<td>NA</td>
<td>10^{420}</td>
<td>NA</td>
</tr>
<tr>
<td>Technology</td>
<td>CMOS</td>
<td>CMOS</td>
<td>BiCMOS</td>
<td>Board level</td>
<td>CMOS</td>
</tr>
</tbody>
</table>

(NA – Information not available).
Custom Made EHW

JPL’98

JPL’2000
Integrate 128 cells with APS 8x8 vision sensor

Japan

Germany
➢ Array of 16x16 programmable transistor cells

UK (Board Level)
Models, levels of accuracy, Differences between HW and SW evaluation

Differences between model and real HW:

• a) Simplified models (e.g. to gain speed in SPICE runs),
• b) Incomplete models because of lack of information about fabrication,
• c) HW can change from the moment was modeled/identified (temperature, radiation, operating conditions),
• d) HW can change in time after evaluation (e.g. slow discharge)

Simulator limitations (SW evaluation):

• a) Convergence conditions, which humans may be able to help by setting/adjusting values,
• b) Conditions unknown a-priori (e.g. charges, initial conditions), in which case the system of differential equations can not be solved

HW testing limitations: a) Transients, b) Charge, e.g. remaining from a previously evaluated individual, c) Impedance loading of measured circuit, d) Time delays between physical signals (e.g. excitatory) and outputs, e) Artifacts originating in signal generators, data acquisition paths, sampling, A/D, etc
Portability problem

- Between HW1-HW2  Thompson's Early Experiments on FPGAs
- Between SW - HW, Experiments at JPL on FPTAs

So what?

- Limits application of SW evolved solutions
- Prevents analysis of HW evolved solutions
Mixtrinsic evolution
Step-by-step evolution example

- Evolve a computational circuit which responds with a Gaussian current output when the input is ramped between Gnd to Vdd.
Genes and their mapping to hardware

What is needed first:
the "genes" representation for the system to be evolved (STBE), and
the mapping/ transformation from genes to an "embodiment" of the STBE.

**Gene representation:**
could be a binary word "10101100", each bit defines the value of a 2-state device.

**Mapping/ transformation from genes to an "embodiment"**
In extrinsic Evolvable Hardware the "embodiment" is a description of a
model of the STBE submitted to a simulator that evaluates the model
and generates a behavioral response.

\[
\begin{align*}
1 & \rightarrow R = 10 \\
0 & \rightarrow R = 1000,0000
\end{align*}
\]

In intrinsic Evolvable Hardware the "embodiment" is the programmable
circuit itself.
FPTA SPICE Netlist

```
.MODEL NMOS NMOS LEVEL=8 TOX=7.6000E-09 XJ=0.100000U + VTO=0.4777253 DELTA=1.00E-02
...

.MODEL PMOS PMOS LEVEL=8 TOX=7.6000E-09 XJ=0.100000U + VTO=-0.7111998 DELTA=1.00E-02
...

* Basic Circuit Configuration for evolvable hardware
m1 n1d n1g 1 1 PMOS l=1.2u w=1.2u
m2 n2d n2g 1 1 PMOS l=1.2u w=1.2u
...

m7 n7d n7g 0 0 NMOS l=1.2u w=1.2u
m8 n8d n8g 0 0 NMOS l=1.2u w=1.2u

* the tgate-based switches
m9 n1g S1 n2g 0 NMOS w=1.2u l=.6u
m10 n2g S1 n1g 1 PMOS w=3.6u l=.6u
m11 n1d S2 n3s 0 NMOS w=1.2u l=.6u
m12 n3s S2 n1d 1 PMOS w=3.6u l=.6u
...

m55 n7g S24 n8g 0 NMOS w=1.2u l=.6u
m56 n8g S24 n7g 1 PMOS w=3.6u l=.6u

vdd 1 0 DC 3.3v
vin+ n5g 0 DC 1.5
vin- n6g 0 DC 1.5
.DC vin+ 0.0v 3.3v 0.15v
.Print DC v(n4d)
.END
```

`10000....0001`

*Resistance Based switches*
R1g2g 1g 2g R1_
R1d3s 1d 3s R2_
...
R7g8g 7g 8g R24_

1 means closed switch
0 means open switch

Each bit of the chromosome determines
the state of a switch in the reconfigurable device.
From chromosome to voltages (or resistances)

\[ \text{Bit}_i = 1 \quad \rightarrow \quad \text{Switch}_i \quad \ OR \quad R_i = 50 \]

\[ \text{Bit}_i = 0 \quad \rightarrow \quad \text{Switch}_i \quad \ OR \quad R_i = 10^{10} \]
Output netlist with resistances

R1g2g 1g 2g 50
R1d3s 1d 3s 1e+10
R1d5d 1d 5d 1e+10
R1g1d 1g 1d 50
R2d4s 2d 4s 50
R1d6d 1d 6d 50
R3sdd 3s 1 1e+10
R3s4s 3s 4s 1e+10
R3g4g 3g 4g 50
R3d5d 3d 5d 1e+10
R3d7d 3d 7d 50
R4sdd 4s 1 1e+10
R3g3d 3g 3d 50
R4d6d 20 6d 1e+10
R4d8d 20 8d 1e+10
R5d5g 5d 5g 50
R5g6g 5g 6g 1e+10
R5s6s 5s 6s 50
R5s7d 5s 7d 1e+10
R5sss 5s 0 1e+10
R6s8d 6s 8d 50
R6sss 6s 0 1e+10
R7d7g 7d 7g 1e+10
R7g8g 7g 8g 50
Output netlist with transistors

Voltages

\begin{align*}
&vs1 \ S1 \ 0 \ 0.000000 \\
&v_s1 \ _S1 \ 0 \ 3.300000 \\
&vs2 \ S2 \ 0 \ 3.300000 \\
&v_s2 \ _S2 \ 0 \ 0.000000 \\
&\ldots \\
&vs24 \ S24 \ 0 \ 3.300000 \\
&v_s24 \ _S24 \ 0 \ 0.000000 \\
\end{align*}

Switches

\begin{align*}
&m9 \ n1g \ S1 \ n2g \ 0 \ \text{NMOS} \ w=1.2u \ l=.6u \\
&m10 \ n2g \ _S1 \ n1g \ 1 \ \text{PMOS} \ w=3.6u \ l=.6u \\
&m11 \ n1d \ S2 \ n3s \ 0 \ \text{NMOS} \ w=1.2u \ l=.6u \\
&m12 \ n3s \ _S2 \ n1d \ 1 \ \text{PMOS} \ w=3.6u \ l=.6u \\
&\ldots \\
&m55 \ n7g \ S24 \ n8g \ 0 \ \text{NMOS} \ w=1.2u \ l=.6u \\
&m56 \ n8g \ _S24 \ n7g \ 1 \ \text{PMOS} \ w=3.6u \ l=.6u \\
\end{align*}
Circuit Output

Output File

<table>
<thead>
<tr>
<th>vin+</th>
<th>V(n4d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000E+00</td>
<td>1.828E-04</td>
</tr>
<tr>
<td>1.500E-01</td>
<td>1.800E-04</td>
</tr>
<tr>
<td>3.000E-01</td>
<td>1.773E-04</td>
</tr>
<tr>
<td>4.500E-01</td>
<td>1.744E-04</td>
</tr>
<tr>
<td>6.000E-01</td>
<td>1.708E-04</td>
</tr>
<tr>
<td>7.500E-01</td>
<td>1.670E-04</td>
</tr>
<tr>
<td>9.000E-01</td>
<td>1.630E-04</td>
</tr>
<tr>
<td>1.050E+00</td>
<td>1.591E-04</td>
</tr>
<tr>
<td>1.200E+00</td>
<td>1.552E-04</td>
</tr>
<tr>
<td>1.350E+00</td>
<td>1.513E-04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target</th>
<th>Actual Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.081971</td>
<td>2.005E-04</td>
</tr>
<tr>
<td>0.246255</td>
<td>7.598E-02</td>
</tr>
<tr>
<td>0.605690</td>
<td>6.637E-01</td>
</tr>
<tr>
<td>1.219709</td>
<td>1.556E+00</td>
</tr>
<tr>
<td>2.010960</td>
<td>2.299E+00</td>
</tr>
<tr>
<td>2.714512</td>
<td>2.482E+00</td>
</tr>
<tr>
<td>3.000000</td>
<td>2.428E+00</td>
</tr>
<tr>
<td>2.714512</td>
<td>2.252E+00</td>
</tr>
<tr>
<td>2.010960</td>
<td>2.006E+00</td>
</tr>
<tr>
<td>1.219709</td>
<td>1.717E+00</td>
</tr>
<tr>
<td>0.605690</td>
<td>1.406E+00</td>
</tr>
<tr>
<td>0.246255</td>
<td>1.102E+00</td>
</tr>
<tr>
<td>0.081971</td>
<td>8.186E-01</td>
</tr>
</tbody>
</table>

Fitness = \sum (Y_i - T_i)^2
Selection: Ranking

- Rank individuals according to the quality of their response
Advanced EHW techniques (morphing, mixtrinsic, voting)

- Guiding evolution through fitness functions
- Distances: when Euclidian doesn’t work
- Weights
- Multi-criteria optimization (power, speed, size)
- Supervised, reinforcement, unsupervised

Movie
Morphing through fuzzy topologies

- A topology with gray-level switches is named here a fuzzy topology
- Instead of being only ON/OFF, the switches were considered as having a Low/High resistance (Low for ON state)
- The binary genetic code specifies if the switch is Low or High, the numerical meaning of this qualitative code would change gradually as a function of a temperature-like parameter. Initially the temperature is high, and Low and High switch status have values close to each other (2M for Low, 20M for High). Gradually the temperature goes down and the switch resistance polarizes to the extremes (10/100s of Ohms for Low, 100s of MOhms for High). The number of generations (100 in most runs) was chosen to ensure some quasi-static behavior. Promising individuals (with higher fitness) have shown-up much earlier in the search. This is probably because of the richer set of effects due to the active contribution of all switch transistors (not only for signal passing), but also because through the gray-switches signals get to the output test/probe points (albeit attenuated) even through "closed" switches along the path.

- Many solutions were actually observed while running through this "annealing". If the goal is to design a blueprint "binary" topology (a wire connecting two components either exists or does not) the annealing technique could be used as a catalyst to accelerate evolution. If on the other hand evolution takes place on hardware that supports gray-level switches, then the degree of opening of the switches could be an extra degree of freedom for the problem, enabling an increased number of solutions. It is possible that these solutions are more sensitive (to various drifts, etc.) than solutions with binary switches.

A. Stoica, 11/9/2000

Evolvable Hardware
Fundamental questions/topics

- Can we evolve artificial systems in similar ways natural systems evolve? Advantages and disadvantages.
- Can we use evolution to obtain intelligent systems, human competitive (and beyond) intelligence
- How can we build devices/HW that evolve (autonomously)?
- Can we seamlessly embed the guiding mechanism for evolution with the morphing system (i.e. the "goals", the "goodness"
- Does EHW scale-up?
Where in the world

TEH '95 Switzerland
ICES '96 Japan
ICES2 '98 Switzerland
EH'99 USA
EH'00 UK/USA
EH'01 Japan/USA
ICES3 '99
ICES4 '00
ICES4 '01
Evolvable Hardware Research

- Evolution of analog circuits in simulation: Koza (Stanford University), Zebulum (Sussex, now JPL), Stoica (JPL);
- Exploration of device physics: Thompson (Sussex University, UK);
- Fault Tolerance: Layzell (Sussex University, UK), Keymeulen (JPL);
- Evolution of Digital Circuits: Miller (University of Birmingham, UK);
- Industrial Applications: Higuchi (Electrotechnical Laboratory, Japan);
- Reconfigurable Analog Devices: Stoica (JPL), Langeheine (Heidelberg University, Germany), Hamilton (University of Edinburgh, UK).
Overview

First/ significant experiments on:...

Functional EHW

DSP/ASIC

FPPA

FPGA

FPAA

FPTA

Analog ASIC (NN)

Analog ASIC functional adjustment

SPICE

1 Thompson, U. Sussex, UK
2 Higuchi, ETL, Japan
3 Koza, Stanford, Lohn, NASA ARC
4 Marchal, CSEM, Switzerland
5 Zebulum, U. Sussex, UK (now at JPL)
6 Stoica, JPL

* Limited selection due to space on slide
Overview

First/ significant experiments on:...

Functional EHW

DSP/ASIC

FPPA

FPGA

FPAA

FPTA

Analog ASIC (NN)

Analog ASIC functional adjustmen

SPICE

1 Thompson, U. Sussex, UK
2 Higuchi, ETL, Japan
3 Koza, Stanford, Lohn, NASA ARC
4 Marchal, CSEM, Switzerland
5 Zebulum, U. Sussex, UK (now at JPL)
6 Stoica, JPL

* Limited selection due to space on slide

A. Stoica, 11/9/2000

Evolvable Hardware
Exploration of Device Physics

- 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

1 kHz - 100 KHz
Evolution of Analog Circuits in Simulation

- Use of Genetic Programming to evolve:
  - Topology;
  - Sizing;
  - Placement;
  - Routing.
- 60 dB amplifier based on bipolar transistors;
- Very large population sizes (10,000,000);
- Simulations using SPICE;
Fault Tolerance

- Layzell @University of Sussex;
- Investigation of population fault tolerance as a consequence of the incremental nature of the evolutionary design process;
- Case studies: Inverter, Amplifier and oscillator;
- Experimental platform: Evolvable motherboard (re-configurable platform with plug-in components);
- Faults induced by removing plug-in components.

Programmable crosspoints switch arrays

Plug-in components
Evolution of Digital Circuits

- Vassilev and Miller & Napier University (UK);
- Evolution of a 3-bit multiplier;
- Cartesian Genetic Programming:

Each cell \( C \) can assume the following logic functions: AND, OR, XOR, 2-inputs MUX, etc;
No feedback allowed;
Chromosome determines cell functionality, input and output connections, and cell interconnections.
Industrial Applications

- Higuchi @ ETL, http://www.etl.go.jp/~ehw/
- Evolutionary recovery;
- Variations in analog components performance are adjusted by GA
- Analog HW Evolution (parametric not topological)

Precise central frequency <1% shift

From 20% to 90% yield!

600 MHz CPU early stages
yield 10%, "clock skew"
800MHz From 2 to 50% yield!

Bias current controller
for IF filter in cellular phones

Programmable delay in
Clock timing adjusting chip
ETL applications

Analogue EHW for IF Filters

Analogue EHW Chip

IF Filter

Evaluation

CPU (GA)

Download

Adjust

Architecture Bits

(Gm: Transconductance Amplifier)

Input signal

Image frequency f_m

Desired signal frequency f_d

Local oscillator frequency f_L

To be suppressed!

Output signal

Intermediate frequency f_F

A. Stoica, 11/9/2000

Evolvable Hardware
Industrial Applications

- Higuchi, ETL
- EMG Prosthetic hand
- EMG - operated by remaining muscles
- Need for adaptation: EMG characteristics

3DOF, 6 movements.

- specific to individuals
  - muscle conditions
  - sensor positions

- EHW (gate level) implements a pattern recognition HW specific to individuals

- EHW hand using EHW chip can very quickly adapt to individuals (5 min)
Reconfigurable Analog Devices

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

Vdd  N  W  S  E  gnd

1:6 analog MUX

D

Selection of transistor dimensions

3-bit Dec

Vdd  N  W  S  E  gnd

1:6 analog MUX

A. Stoica, 11/9/2000
Evolvable Hardware
Embryonics, CSEM/EPFL, Switzerland

Each Nucleus stores its own copy

Differentiation Process

Healing Process
Automated synthesis of digital circuits

**Circuit Specification:** The automated synthesis of digital circuits is possible at two design levels: the block diagram level and the logic gate level. At the block diagram level, a hardware description language, such as VHDL or SFL, is used as the genetic encoding of the circuit. The automated synthesis optimizes the HDL code, compiles and downloads into a Programmable Logic Device. At the logic gate level, the configuration bits in the programmable device are used as the genetic encoding of the circuit. The automated synthesis optimizes the configuration bits and download into a Programmable Logic Device.

**Circuit Evaluation:** The evaluation of the digital design is done by simulation or by downloading the configuration bits of the candidate design into a Programmable Logic Device. The design performance is evaluated by using the input-output mapping for a combinatorial circuit and by tracing the states sequence or the state transition paths for a sequential circuit.

**Achievements:** There has been successful automated synthesis of digital circuit for 6-in MUX (25 gates) [KIT96], a 4-bit comparator (23 gates) [HIG96], and a sequential adder (50 gates) [ZEB96]. Automated synthesis of an HDL-program representing a circuit with 8 control states (using 100 gates) able to control a simulated ant to find food using the shortest trail [HEM96].

**Automated synthesis of HDL code**

- **(0.0) module** → K_MOD name list_comp list_pin list_action
- **(1.0) name** → K_NAME
- **(2.0) list_comp** → comp
- **(20.3) comp** → K_INTRIN inst_name

**HDL code for ant control**

**Automated synthesis of Logic Gates**

- **2-channel 2-bit comparator**
  - OUTPUT: 1: A >= B
  - 0: A < B

**Chromosome Representation**

- 001110100111110101010000

**Evolvable Hardware**
Automated synthesis of analog circuits

Various of analog circuits have been synthesized in software.

The elementary building blocks varied from passive R,L,C components (used mainly in filter synthesis) to transistors and operational amplifiers.

The most common circuits synthesized are filters and computational circuits
- Koza and colleagues at Stanford U. used Genetic Programming to grow “embryonic” circuits into circuits that satisfy desired requirements. This approach was used for evolving filters, computational circuits, etc. Koza’s evolutions were performed in simulations, without the concerns of physical implementations, as proofs-of-concept that evolution can lead to designs that compete or even exceed the performance of human designs. No analog programmable devices exist that would support the implementation of the resulting designs and thus intrinsic evolution was not possible. [KOZ96] [KOZ97]. In principle, though, one can test validity of the designs in circuits built from discrete components, or in an ASIC.

- Lohn and Colombano at NASA Ames used a linear representation and a Genetic Algorithm to evolve filters [LOH98]

- Stoica and colleagues at JPL used a Genetic Algorithm to evolve CMOS transistor-based computational circuits and validated them in a test CMOS chip.
On-line evolution of digital circuits

On-line Evolution
The on-line evolution is applied to tasks operating in real-time and where: (a) algorithms vary depending on the processed data, (b) algorithms change to meet new performance requirement, (c) hardware components may fail.

Achievements
• Data compression for digital color electrophotographic printers reconfigures on-line its hardware structure to improve the method of coding according to the local characteristics of the image. It obtains a compression ratio twice as high as JBIG [TAN98].
• On-line adaptation of a myoelectric artificial hand operated by muscular control signals from a disabled person was made possible using a single chip integrating a CPU, the control unit for the hand, a memory for the chromosome, and the evolution unit to adapt the control on-line [KAJ98].
• On-line fault-tolerant digital architecture accepting a 0.1% component defects using cellular programming system where a set of digital circuit cells is reported [SIP97]. The evolution is carried out onboard and the evolutionary phase intertwined with normal execution.

A. Stoica, 11/9/2000
On-line evolution of analog circuits

On-line Evolution
The on-line evolution defines a set of instructions to apply to programmable analog devices able to process analog signals. The candidate circuits are implemented immediately and tested on real signals. The analog devices are programmed at the transistor level to exploit analog properties of the circuits and to use the devices for tasks well outside the range of the original design [KOZ96] [THO96]. They can also be programmed at the building block level explicitly designed for their analog properties and designed as a grid of cells that can have their properties and interconnects altered digitally (Field Programmable Analog Arrays, FPAA can be used) [FLO98].

Achievement
- An analog circuit able to generate an asymmetric and non-linear steady-state characteristic has been designed using analog building block circuits [FLO98].
- An analog circuit able to oscillate at the specified target frequency, e.g., 55.3 kHz, has been designed from primitive electronic components, such as astable multivibrators, without using inputs and clock signals [HUE98].
- An analog circuit able to discriminate between an input signal of frequency of 10 kHz and 1 kHz and robust to changes of temperature has also been designed [THO96].

A. Stoica, 11/9/2000
Evolvable Hardware
Evolution on Neural ASICs

Intrinsic Evolution on a Functional-Level Mixed-Signal ASIC (A Neural Network Chip) [STO98]

- Goal: experiment with intrinsic Evolvable Hardware
- Reconfigurable hardware: JPL-NN64
  - 64 neurons, 4096 synapses
  - analog neuron input and output, digital synapses (8-bit)
- Evolutionary on-chip learning (weight modifications)

Functional approximation of a DC characteristic

A DC function learned on the chip

Learning sensory-motor maps
Truncated visual inputs and steering controls
Examples from the training set used for learning

- Steer -0.7
- Steer 0.3

Closeness to target

Response on the oscilloscope

Khepera following a marked trail
Evolvable Hardware for adaptive compression

A genetic programming system was developed to perform adaptive image compression based on predictive coding [Fukunaga, JPL].

• The GP system evolves s-expressions that represent nonlinear predictive models for lossless image compression.
• The error image is compressed using a Huffman encoder.
• A different model was evolved for each image.

Compression ratios of various compression techniques applied to set of test images

<table>
<thead>
<tr>
<th>Image Name</th>
<th>Original Size</th>
<th>evoluted</th>
<th>CALIC</th>
<th>LOCO-I</th>
<th>Com-press</th>
<th>gzip</th>
<th>pack</th>
<th>szip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earth</td>
<td>72643</td>
<td>30380</td>
<td>31798</td>
<td>32932</td>
<td>42502</td>
<td>40908</td>
<td>55068</td>
<td>40585</td>
</tr>
<tr>
<td>Earth4</td>
<td>11246</td>
<td>5513</td>
<td>5631</td>
<td>5857</td>
<td>7441</td>
<td>6865</td>
<td>8072</td>
<td>7727</td>
</tr>
<tr>
<td>Earth6</td>
<td>20400</td>
<td>9288</td>
<td>10144</td>
<td>10488</td>
<td>11339</td>
<td>10925</td>
<td>13264</td>
<td>12793</td>
</tr>
<tr>
<td>Earth7</td>
<td>21039</td>
<td>10218</td>
<td>11183</td>
<td>11476</td>
<td>13117</td>
<td>12520</td>
<td>15551</td>
<td>13269</td>
</tr>
<tr>
<td>Earth8</td>
<td>19055</td>
<td>9594</td>
<td>10460</td>
<td>10716</td>
<td>11699</td>
<td>11350</td>
<td>13298</td>
<td>12465</td>
</tr>
</tbody>
</table>

• The results obtained show that for science data images, an evolvable-hardware based image compression system is capable of achieving compression ratios superior to that of the best known lossless compression algorithms.
Human-competitive machine intelligence

Creativity: evolved circuits = new patents

A list of 16 attributes reasonably expected to be possessed by a system for automatically creating computer programs: GP possesses them all

- Competitiveness with human-produced solutions to important real-world problems
  - Patentable
  - Publishable in peer-review paper independent of the fact it was automatically generated

- Ways around patented solutions (e.g. rewarding topologies that are different than patented ones)
  - 14 instances in which GP determines competitive solutions
  - 9 rediscover patented solutions in analog circuit design
Why EHW in HW?

- Circuit design can be demonstrated in SW, but...
- takes huge resources (the photo on the bottom left shows Koza’s computers, which run for days evaluating hundreds of thousands of circuits for thousands of generations!!)
- Computationally intensive (640,000 individ. for ~1000 gen.
10s of hours, expected $\sim3 \text{ min in 2010}$ on desktop PC for experiments in the book

- SPICE scales badly (time increases nonlinearly with as a function of nodes in netlist - in $\sim$ 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 will allow evolution 4+ orders of magnitude faster than SPICE simulations on Pentium II 300 Pro. ($\sim3 \text{ min in 2001 for circuits of complexity } \geq \text{ Koza’s}$).
Current directions of EHW research @ JPL

• Automatic design (patents, optimization), adaptive hardware
• Toward an evolvable SOC
• Examples of SW and HW evolution (gaussian, filter, A/D, D/A) adaptive and reconfigurable circuits: fuzzy neuron, polymorphic gates, arithmetic and logic circuits
• Survivable HW (faults and extreme environments )
• Polymorphic electronics
• Evolvable sensing (MECA, APS)
• EvoNanoTech
• Randomizers, digital EHW
• Antennas (presented in the MEMS section)
Automatic design (patents, optimization), adaptive hardware

Objective 1: Automated device/circuit/system design/synthesis/optimization
Objective 2: Chips/hardware/systems capable of (self-) adaptation to the environment

Empirical design is sub-optimal. Complete design space search is unfeasible.
• Rapid design
• Generation of novel devices/circuits

Extrinsic Evolvable Hardware
• Multi-criteria optimization (power, speed, etc.)
• Novel approaches to electronics, e.g. polymorphic, hybrid system design/integration

Validation for synthesis of new circuits directly in reconfigurable hardware

Intrinsic Evolvable Hardware
• Automated design in space, without human designer intervention
• Adaptive electronics
Toward an evolvable SOC

FY99

FY00

FY01

Array of PTAs

Reconfigurable FPTA Chip with ~10000 transistors

Evolvable Hardware
Examples of SW and HW evolution

- Computational circuits: gaussian, neurons
- Filters,
- A/D, D/A
- adaptive and reconfigurable circuits: fuzzy neuron
- polymorphic gates
- multiplier
- logic circuits
- ...

A. Stoica, 11/9/2000
CMOS Op.Amp. Mapped into PTA cells

Transient response with a sine wave input:
Red - Without switches
Blue - With switches.
Evolution of Computational circuits

Evolution of Fuzzy-Neuron Circuit

\[ S_s(x, y) = \begin{cases} 
MAX(x, y) & \text{if } (s = 0) \\
 x + y - x \cdot y & \text{if } (s = 1) \\
1 - \log \left( 1 + \frac{(s^{1-x} - 1)(s^{1-y} - 1)}{s - 1} \right) & \text{if } ((0 < s < \infty), s \neq 1) \\
MIN(1, x + y) & \text{if } (s = \infty) 
\end{cases} \]

- Uses two FPTA cells (16 transistors)
- compact implementation

Survivable HW

- Extreme environments - temperature, radiation
- Fault repair via evolutionary reconfiguration
- Robust behavior through evolutionary design
- Temperature compensation via architectural changes
- Diehard electronics
Evolvable Hardware for Extreme Environments: Expanding Device Operational Envelope through Adaptive Reconfiguration

- EHW can preserve/recover system functionality by reconfiguration/morphing.

- If device characteristics change with temperature, one can preserve the function by finding a different circuit solution, which exploits the altered/modified characteristics.
Planetary Extreme Environments

Radiation
total dose, Mrad

Sun Spot

Europa
Ganymede
Comets
Moon
Earth
Mercury
Venus
Io

Temperature, °C
Expanding Operational Envelope through Adaptive Reconfiguration
(A Circuit Solution)

- Claim: Circuits solutions can further expand the operational envelope, and should be considered in addition to device solutions

- Demo: Circuits solutions can expand the operational envelope of current devices

- Limitations are of the ensemble device/configuration, not of the device(material) only
If the device characteristic changes, change the circuit topology

Notation
T : Temperature of operation. Could be an interval
D(T) ={d1(T), d2(T), ..., dn(t)} : Set of devices with various temp characteristics
C : Circuit (topology, configuration). Describes interconnection of devices.
F : Function of circuit

\[ f_1, T1, D(T1) \rightarrow C \]

For desired function, given operational temperature T1, and D, a set of devices of with certain
temperature dependent characteristics, find a circuit topology/configuration C.

**current approach**
\[ fA, T1, D(T1) \rightarrow C \]

**proposed approach**
\[ f, T, D(T) \rightarrow C(T) \]

\[ fA, T1, D(T1) \rightarrow C(T1)=c1 \]

@T2
\[ T2, D(T2), C \rightarrow fB \]

C is fixed, we are stuck
\[ T2, D(T2), C(T1) \rightarrow fB \]

C can change, search again
find c2=C(T2) which gives T2, D(T2), C(T2) ---> fA
Steps of the temperature experiment:

1. Get human design or evolutionary design of a circuit at 27°C
2. Expose chip to low/high temperature and observe degraded response
3. Apply evolution, and obtain a new circuit solution, which recovers functionality

[Images: Immersing the chip under test in liquid nitrogen and Test chamber for high temperature]
Functional Recovery at Low Temperatures

\[ f_A = \{T_1, D(T_1), C(T_1)\} = \{T_2, D(T_2), C(T_2)\} \]
\[ \{T_2, D(T_2), C(T_1)\} = f_B \]

Functional response of original circuit design

T = 27°C

T = -196°C

Functional response of original circuit affected by temperature

T = -196°C

Evolvable Hardware

A. Stoica, 11/9/2000
Circuits evolved at -196°C, +27°C and +245°C

T: -196°C  T: +27°C  T: +245°C

Input1: 2.5Volt  Input2: Ramp 0Volt to 5Volt

★ changes

A. Stoica, 11/9/2000
Evolvable Hardware
At high temperatures, classical gates will produce wrong logical level; circuits with correct logic can be obtained by evolution.

SPICE Simulation of a standard NAND Gate at 27C and 300C

VDD = 3.3V

Wrong logic response '0' instead of '1'

Threshold between logic levels "0" and "1"
Digital circuits are affected by temperature and can give wrong output response.

**Combination at gate input**

<table>
<thead>
<tr>
<th>“00”</th>
<th>“10”</th>
<th>“01”</th>
<th>“11”</th>
</tr>
</thead>
<tbody>
<tr>
<td>T=180C</td>
<td>“1”</td>
<td>T=170C</td>
<td>“0”</td>
</tr>
<tr>
<td>T=160C</td>
<td>“0”</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Threshold between logic levels “0” and “1”

---

**Gate evolved at 27C, gives wrong logic response at over 180C (‘1’ instead of ‘0’)**

---

**180C**

Evolved at 180, recovered function

---

A. Stoica, 11/9/2000

Evolvable Hardware
Functional Recovery of an AND

Compliant AND circuit evolved at 27C degrades as temperature increases

3.3V

Evolved@27C  Measured@100  Measured@240C

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

Evolved@240C  Measured@150C  Measured@137C
Response degradation for conventional AND gate design, new design for high temperatures
Discussion

- Initial experiments, although very simple, demonstrate the new concept of extending functionality at extreme temperatures through hardware (self) reconfiguration.

- Fine granularity probably helps - bigger search space, more flexibility.

- How difficult is for more complex circuits?
Self-repair experiments

- Multiplier with two cascaded FPTAs (88 bits)
  - 6 external connections between 2 PTAs
  - Find solution after 59 generations
  - Cut 1 connection after 60 generations.
  - Start the GA and recover a desired circuit after 20 generations

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
DieHard Architecture

Currently, to achieve evolutionary reconfiguration around a fault the implementation of the adaptation mechanism itself must be fault-free. If it can not be made fault-tolerant, the mechanism must be isolated in a protected area.

Seamlessly integrated “Diehard” architecture

Conceptual development of a “die-hard” architecture, i.e. a way of distributing the adaptation/self-configuration mechanism into the reconfigurable hardware.
Evolvable sensing

Smart sensing – evolve reconfigurable sensors
Replace/self-configure signal conditioning electronics in flight instruments.
State-of-the-art:
Electrometer, Vision Sensor (APS)

Proposed effort:
Reconfigurable analog circuits
A/D conversion
Communication

Evolvable reconfigurable sensor electronics

Future:
Various Transducers
EVOLVABLE analog circuits
A/D conversion
Communication

Evolvable Sensor Chips

Concurrent effort (funded by DARPA):
Reconfigurable analog circuits
Reconfiguration algorithm integrated on the chip

Evolvable Circuits - Evolvable Chip

Evolvable Hardware

Reconfigurable analog circuits (on a chip)
Reconfiguration algorithm (e.g. evolutionary algorithm) running externally on computer

Sensor on a chip

Detector/transducer
Dedicated analog circuits
A/D conversion
Communication

A. Stoica, 11/9/2000
Evolvable electronics for MECA Electrometer

MECA Electrometer, Electric field sensor and electronics, Mars’01 Lander (canceled)

2000 efficiency:
- Bench-level prototype evolvable sensor integrating sensor and EHW board

2003 deliverable:
- E-Sensor Chip with built-in HW-GA, capable to perform on-chip synthesis of electronic circuits
  - On-chip smart sensor processing, increase quality of returned data, adaptive sensing, lean smart data
Evolvable vision sensors

- Photo-detector
- Dedicated analog circuits
  - A/D conversion and video readout
- Interfacing constraints
- Neuromorphic circuits implementing local vision algorithms
- Inspiration from retin-like circuitry
- Vision-oriented reconfigurable analog circuits
- Evolutionary constraints
- Evolvable programmable transistor array (PTA) circuits

A. Stoica, 11/9/2000

Evolvable Hardware
Genetic engineering of novel nanoelectronic devices

JPL GENES Project (Genetically Engineered NanoElectronic Structures), Stoica/Klimeck

Objectives
- Automated/rapid nanotechnology device synthesis and development.
- Generation of novel devices.

Approach
- Augmented the advanced NanoElectronic MOdeling (NEMO) tool to analyze individual structures in parallel.
- Uses parallel genetic algorithm package (PGApack) to optimize and select desired structures in NEMO.

Result
- A Resonant Tunneling Diode with a desired characteristic was automatically designed/evolved.

The Device evolved matched the experimental current voltage characteristic of a resonant tunneling diode.

Five structural parameters used in search/evolution
- Thickness: well, barrier, spacer.
- Doping: low doped spacer, unintentional doping in center

A. Stoica, 11/9/2000

Evolvable Hardware
Randomizers, digital EHW

Currently expanding EHW testbed to include FPGA boards

- Execute the Genetic Algorithm and Evaluate sets of pseudo random numbers using DIE HARD software suite
- Generate sets pseudo random numbers
- RNG
  - Completed design and validation of a VHDL implementation for a pseudo random number generator. The design is being ported to the FPGA board.
  - Integrated the DIE HARD software suite that will be used in the evaluation of pseudo random generator into the EHW test bed

2000 - Digital EHW for Complex Functionality
- Synthesis of Random Number Generators on FPGAs.
- Suitable task for evolution: no design guidelines, but measures of randomness exist
- FPGA offer fast, parallel evaluation of candidate solutions.
- Encryption applications - DARPA and NSA

Long term goal - Evolvable flight computer
- Would leverage FPGA-based flight hardware and endowing it with evolution capabilities

Xilinx Virtex FPGA Board

Aiming to enable automatic synthesis of complex digital functionality for flight computers at the FPGA hardware level

A. Stoica, 11/9/2000

Evolvable Hardware
MEMS, smart devices, distributed intelligence, amorphous computing

EHW may have a strong impact on

- Micro/nano-scale systems
- Biological/artificial hybrids
- Adaptive Internet HW, Internet infrastructure
- "Smart households" and evolvable sensors
- EHW could help in all applications requiring automatic adaptive reconfiguration.

A new kind of HW: Aware (environment, power), smart (self-configurable), adaptive, robust

- The following slides focus on
  - MEMS
  - Antennas
  - Smart sensing
  - Sensor web
  - Hybrid devices (bio-)
  - Amorphous computing
What are MEMS?

- MEMS - "Micro Electro Mechanical Systems" or "Microelectromechanical Systems";
- MOEMS – "Micro Opto Electro Mechanical Systems";

- **Physical sensors**: Inertial, pressure, stress and strain, temperature and radiation sensors;
- **Chemical sensors**: liquid and gas;
- **Actuators**: micromotors, gear trains, linear displacement devices, switches and relays.
- **Biomedical devices and microfluidics**: systems of liquid microchannels, micropumps, microvalves, and micro flow controllers;
- **Optical devices**: micromirrors and fiber switches;
- **RF devices**: switches, mixers and filters.
MEMS Definition

- MEMS technologies are defined by the use of micron-scale machining techniques to produce sub-millimeter features:
  - Semiconductor wafer processing;
  - 2.5-D technologies (complex 3-D geometries may be difficult to produce using existing MEMS technologies);
- Micromachining history:
  - Bulk micromachining;
  - Thin film technology (surface micromachining);
  - Working in the third dimension: High-aspect ratio technologies;
  - Monolithic sensor technologies.

Bergstrom’s slides
Benefits and limitations

- Early motivations: miniaturization (size and mass) and the potential to outperform traditional systems;

- Current motivations: system cost (large volume mfg), system capability (biosystems & wireless) and power consumption (wireless systems).

- Benefits of micro-scale:
  - new methodologies (biomems), lower manufacturing cost, beneficial for system integration (transducers, sensors and actuators), 2.5 is adequate for most implementations.

- However, many manufactured or machined sensors, actuators, and analytic systems still outperform micromachined systems:
  - Can employ methodologies not yet possible with micromachined devices;
  - May operate at scales suitable for the method;
  - In some instances, may be less expensive;
  - True 3D implementations.
Caltech MEMS

http://touch.caltech.edu/

- MicroBat

- Integrated Structures for Protein Mass Spectral Analysis

Distributed Turbulent Flow Control by MEMS Integrated with Neural Network

Figure Left: Electrospray chip with 2 mm long overhanging Parylene Micro Capillary
Figure Right: Close view of the capillary tip

Micro Heat Exchanger Using Impinging Jets

Handheld Fluidic System for Biological Agents Detection
Smart Skin

Objectives

- Develop a generic smart skin and control approaches for broadband radiation control from structure.

- Apply the smart skin to reduce interior noise due to boundary layer excitation in aircraft cabin.

http://www.val.me.vt.edu/presentations/slides/spie97/sld003.htm
Neural VLSI for MEMS control

- 3-D VLSI multi-chip packaged (Cube) neural networks architecture for image processing with four orders of magnitude speed in object recognition (visual & hyperspectral data).
- Integration of neural networks in ULPE and sensors leading to highly sensitive instruments for NASA missions such as Artificial Olfaction System or Active Skin.
Monolithic Sensor Systems

Inertial Sensors

Integrated Micro Instruments, Inc (IMI) multi-axis accelerometer product vehicle monolithically integrated with CMOS

(http://www.imi-mems.com/)
Applications for inertial/accelerometers MEMS

Navigation
• GPS backup during outages;
• Motion sensing for map matching

Industrial
• Vibration monitoring;
• Platform stabilization;
• Shock sensing.

Consumer
• Personal & car navigation

Automotive
• Commercial fleet management;
• Active suspension.

Military
• Augmented GPS navigation;
• UAV missile and bomb guidance

Seismic
• Seismic surveys;
• Petroleum exploration;
• Down hole mapping.
Optical Systems

- Surface and bulk micromachining techniques utilized for micromirrors, fiber guides and fiber switches

http://www.dlp.com/dlp/resources/dmmd.asp

➢ SXGA device with black aperture: 1280x1024; 1,310,720 mirrors
Microfluidic systems

- Micro flow channels and reservoirs formed in glass micromachining for bioanalytical systems;
- http://micromachine.stanford.edu/
BioMEMS Systems

- Sensors
  - Tissue sense and active probes;
  - Chemical sensing of biological fluids (e.g., DNA sequencing, blood chemistry sensing).

- Actuators
  - Needles
  - Tweezers

- Biocapsules membranes
  (Protection of transplanted cells)

http://chopin.bme.ohio-state.edu/bme_home/ferrariwebpagedir/
Micromanipulators

- Micromanipulators for biomedical applications.
- http://www.memspi.com/wholelwzw.html
ISSYS Pressure Sensor

- Ultra wide range pressure sensor;
- Anisotropic bulk etching of silicon;
- ASIC control chip is copackaged with the sensor.

[Integrated Sensing Systems (ISSYS) ultra-biomedical pressure sensor device; Catheter applications;
(http://www.mems-issys.com/)]
ISSYS Pressure Sensor

- Ultra wide range pressure sensor;
- Anisotropic bulk etching of silicon;
- ASIC control chip is copackaged with the sensor.

[Integrated Sensing Systems (ISSYS) ultra-biomedical pressure sensor device;
Catheter applications;](http://www.mems-issys.com/)
Data transmission

(http://products.analog.com/products/)

High Speed Digital Isolators

Micromachined Relays

* Pin 1 and Pin 3 are internally connected. Either or both may be used for VDD1.
** Pin 5 and Pin 7 are internally connected. Either of both may be used for GND2.
Genetic design of antennas

D. Linden, MIT

- A Genetic Algorithm was used in conjunction with the Numerical Electromagnetic Code, Version 2 (NEC2) (as simulator) to create and optimize atypical wire antenna designs with impressive characteristics [LIN97]. Evolutionary techniques may revolutionize the design of wire antennas.
  - GA-optimized Yagi antennas surpass by ~1dB the gain of conventional Yagis
  - Crooked-wire antennas, consisting of wires joined at various locations and with various lengths (both determined by GA), evolved to unusual shapes, unrealizable using conventional design, and demonstrated excellent performance both in simulations and physical implementation

![Possible novel designs, e.g. tree like antennas [LIN97]](image)

Other ideas about evolvable antennas
- Use real reconfigurable antenna, morphing in real time under evolutionary control
- Components of reconfigurable antennas could be macroscopic (e.g., actuated wires) or MEMS
- Evolution of antenna arrays
- Co-evolution of antenna & electronic interface (e.g., for matching impedance, etc.)
Evolvable antenna system

- Goal: to explore the *in-situ* optimization of a reconfigurable antenna (vs. optimization on a controlled antenna range)
- Software evolves relay configurations, user subjectively ranks designs based on signal quality (future work will automate this process)
- 30-relay antenna created, along with all necessary software to control and evolve it
- System is able to optimize effectively for frequencies in the upper portion of the VHF broadcast TV band (177 - 213 MHz)
- 1.5m diagonal length (about 1 wavelength at above frequencies)
- Continuing work: improve optimization effectiveness, expand number of relays and antenna size to enhance low frequency performance

Evolvable antenna concept developed by Linden & Stoica
Antenna work performed by D. Linden. JPL funding.
Collective Intelligence of Reconfigurable Agents

Collective Intelligent Agent System:
- The goal is to develop a modular and self-reconfigurable robotics system with a collective intelligence able to change their topological structure for locomotion, manipulation, or sensing purposes to carry out optimally a variety of tasks/missions.
- The agent system uses a set of unreliable robotic modules with limited power, sensing, communication and computation.
- The collective intelligence is based on the principles underlying the behavior of natural systems such as swarming of ants. Amorphous computing offers a language that is used to observe, control, organize and exploit the behavior of collective hardware modules.

Mechanical and Computational Features:
- Reconfigurability/Shape Metamorphosis
- Distributed Control, Sensing and Action
- Distributed Learning
- Fault tolerance and self-repair
- Adaptive Communication
- Scalability

NASA Applications:
- Control of limited power and computation reconfigurable micro robot/spacecraft.
- Reliable, fault tolerant and self repaired inexpensive reconfigurable robotic/ spacecraft module.
- Distributed and large science gathering.
- Distributed space/planet station maintenance and health monitoring.

A. Stoica, 11/9/2000
Evolvable Hardware
Reconfigurable Robotic Molecules (Dartmouth)

Robotic Molecule:
- Robotic Molecule is a 4 degree-of-freedom, small-scale module capable of aggregating with other identical modules to form three-dimensional dynamic structures.

Features:
- Molecule consists of two atoms connected by a right-angle rigid bond
- One atom has 5 inter-Molecule connectors and 2 degrees of freedom: rotate about the connector and about the bond
- Manually controlled

A. Stoica, 11/9/2000

Evolvable Hardware
Amorphous Computing (MIT)

Amorphous Computing:
- allows to obtain coherent behavior from the cooperation of large numbers of unreliable parts that are interconnected in unknown, irregular, and time-varying ways
- Give a method for instructing myriads of programmable entities to cooperate to achieve particular goals.

Features:
- Large number: of computing elements sprinkled on a surface or in a volume.
- Local communication: Each can talk to a few nearby neighbors, but not reliably
- Each has modest computing power and modest amount of memory
- The particles are not synchronized, nor are they regularly arranged

EXAMPLE: Differentiation and Growth
- Each computing element's state includes some binary markers. Each computing element's program has many independent rules.
- Rules are triggered when messages are received. A rule is applicable if a certain boolean combination of markers is satisfied.
- When a rule is applied it may set markers and send further messages.
- Messages have hop counts that determine how far they will diffuse.
- Markers may have lifetimes after which they expire.

A. Stoica, 11/9/2000
PolyBot: Modular Reconfigurable Robotics (Xerox PARC)

Polypod:
- Polypod is a bi-unit modular robot: it is built up of exactly two types of modules that are repeated many times. Dynamic reconfigurability allows the robot to be highly versatile, reconfiguring itself to whatever shape best suits the current task.

Features:
- Expanded and Angled Segment
- 3 Segments
- Spider
- Evolvable Hardware
- Caterpillar
Evolution of Machines: The Golem Project (Pollack)

http://www.demo.cs.brandeis.edu/pr/robotics.html

“In the Golem project (Genetically Organized Lifelike Electro Mechanics) we conducted a set of experiments in which simple electro-mechanical systems evolved from scratch to yield physical locomoting machines. Like biological lifeforms whose structure and function exploit the behaviors afforded by their own chemical and mechanical medium, our evolved creatures take advantage of the nature of their own medium - thermoplastic, motors, and artificial neurons. We thus achieve autonomy of design and construction using evolution in a limited universe physical simulation, coupled to off-the-shelf rapid manufacturing technology. This is the first time robots have been robotically designed and robotically fabricated.”

Evolutionary Lego Crane
Chevron

- Sensor webs
- Space energy
- Teramac, chemical computers
Multi-functional infrastructure

- The “Dr Jeckyl and Mr. Hyde” of information processing:
  - Doing useful (company) work in the day; becoming a world-wide supercomputer resource “for hire” work in the night.
  - J&H would provide users extra free disk (not shared to prevent cyber-attacks) and OS - (and perhaps provide free internet)

What has to do with EHW?
1. Population based search
2. High level instruction - computer/cluster/network self-organization to best process information

And with Chevron? (Perhaps it can own it!:-)
Smart sensor web
Who's gonna fuel these?
(Ok, these are solar for now:-)
Outer Solar System Exploration Roadmap

Outer Planets Program: Exploring Organic-Rich Environments

Cassini/Huygens

Galileo-Europa

Europa Orbiter

Europa Lander

Titan Explorer

Comet Nucleus Sample Return

Pluto/Kuiper Express

NASA's interest related to search for life

Chevron's: energy and chemistry related
Leverage Planetary Resources

Alone of all the satellites in the solar system, Titan has a significant atmosphere. At the surface, its pressure is more than 1.5 bar (50% higher than Earth's). It is composed primarily of molecular nitrogen (as is Earth's) with no more than 6% argon and a few percent methane.

Titan is the fifteenth of Saturn's known satellites and the largest:
- orbit: 1,221,830 km from Saturn
- diameter: 5150 km
- mass: 1.35e23 kg

Titan Explorer

Interestingly, there are also trace amounts of at least a dozen other organic compounds (i.e. ethane, hydrogen cyanide, carbon dioxide) and water. The organics are formed as methane, which dominates in Titan's upper atmosphere, is destroyed by sunlight.

...there appears to be a lot of chemistry going on...
Planetary engineering of Mars (et al)

Biology and the Planetary Engineering of Mars http://spot.colorado.edu/~marscase/cfm/articles/biorev3.html

- 6. Production of greenhouse gases. Microorganisms could be used to metabolise nitrate deposits to NH3. As discussed in section four, NH3 is a powerful greenhouse gas, so not only would this process contribute to the warming of the planet, but at low levels NH3 would be photochemically broken down into N2, a further greenhouse gas (H2O) and H2 (Kasting, 1982). Another greenhouse gas that could be produced by biological mechanisms is methane, CH4. Methane may have been a constituent of the Martian paleoatmosphere (Kasting, 1991).

- 7. Biomass production and soil protection...

...Objectives of some current proposals
Develop several optimized strains of bacteria suitable for survival on Mars, which will perform photosynthesis to generate chemicals necessary for future human habitats.
Future goal: send colonies of biologically adapted bacteria to be tested on a future Mars mission.
Long-term goal: deploy these bacteria as tools for in situ resource utilization for oxygen delivery and terraforming Mars.

Evolved bacteria will be capable of capturing essential chemicals in the presence of considerable harsh environment. Cyanobacteria on Earth produce oxygen and survive in extremely harsh environments. They were tremendously important in shaping ecological change and the course of evolution throughout the early history of life on earth.

A. Stoica, 11/9/2000

Photograph of plants on Mars.
Once the oxygen level is around 20 mbar then plants can be introduced onto Mars. These will serve a number of functions including the production of more oxygen and stabilising geological features. (Photograph J. A. Hiscox and M. W. Parnell).
HPL Teramac

1THz multi-architecture computer

- \(10^6\) gates operating at \(10^6\) cycle/sec
- Largest defect-tolerant computer
- Contains 256 effective processors
- Computes with look-up tables
- 220,000 (3\%) defective components
- Comatose at birth

From a presentation by
Joel Birnbaum Chief Scientist Hewlett-Packard
Chemical computers, Chevron Inside™

Silicon-based technology will hit its physical limits in about 10 years, predicts James Heath, a chemist at the University of California, Los Angeles. To boost computing efficiency past that point, he predicts that engineers will have to create computer components—switches, wires, and memory—on a molecular scale, growing them more or less like crystals in a beaker instead of etching them on a silicon wafer. But chemical structures always have defects, and in a chemically assembled microprocessor, several percent of the components would probably not work. Moreover, just as no two snowflakes are the same, says Heath, "each machine that comes out of the lab will be different." The solution, Heath believes, is not to aim for a defect-free computer but a defect-tolerant one, like Teramac.

http://www.academicpress.com/inscight/06121998/graphb.htm
Challenges for EHW

Convergence  Scalability  Fitness function
Robustness  Evolution-oriented devices

- The field lacks a strong formal/theoretical basis, and most work is rather empirical/experimental.
- In most situations there is no (theoretical) way to know if a satisfactory/optimal solution exists
- There is no guaranty that Evolvable Hardware will find a solution, even if a solution exists
- There are no proofs of good scalability
- Current programmable hardware is not designed for evolution, and thus is not optimal for it
- Evolving hardware may be risky if intermediary (unsatisfactory solutions) can negatively impact system operation
• Evolution principles can be used in automated design and hardware evolution/adaptation
• Evolvable hardware can provide in-situ designs of electronics for novel functionality, or control reconfiguration to maintain functionality in changed conditions
• New circuits designs, can be obtained in simulations or directly evolving on programmable chips.

Long term impact
• Micro/nano-scale systems
• Biological/artificial hybrids
• Adaptive Internet HW, Internet infrastructure
• "Smart households" and evolvable sensors
• EHW could help in all applications requiring automatic adaptive reconfiguration.
• A new kind of HW: Aware (environment, power), smart (self-configurable), adaptive, robust

EHW technology has the potential to be the underlying technology behind tomorrow's infrastructure, not only for electronics but also for smart optical, structural, thermal systems through reconfigurable, morphing, adaptive MEMS and materials.
Further References

- EHW webs:
  - A. Thompson’s list of EHW groups
    http://www.cogs.susx.ac.uk/users/adrianth/EHW_groups.html
  - Evolutionary Electronics at Sussex http://www.cogs.susx.ac.uk/users/adrianth/index.html;
  - ETL, Japan: http://www.etl.go.jp/~ehw/
  - Genetic Programming Inc: http://www.genetic-programming.com/
  - NASA/AMES: http://ic.arc.nasa.gov/people/jjohn/adcs.html
  - EFPL/Lausanne: http://islwww.epfl.ch/
- MEMS webs:
  - Caltech MEMS http://touch.caltech.edu
  - MEMS Clearinghouse: http://mems.isi.edu/
  - MEMS Primer at UC Berkeley http://www-bsac.eecs.berkeley.edu/~elliot/mems.html
  - Tutorial on MEMS http://home.earthlink.net/~trimmerw/mems/tutorials.html
  - MEMS Precision Instruments http://www.memspi.com/
  - U. Wisconsin MEMS http://mems.engr.wisc.edu/what
  - Nanotech
    - http://www.foresight.org/