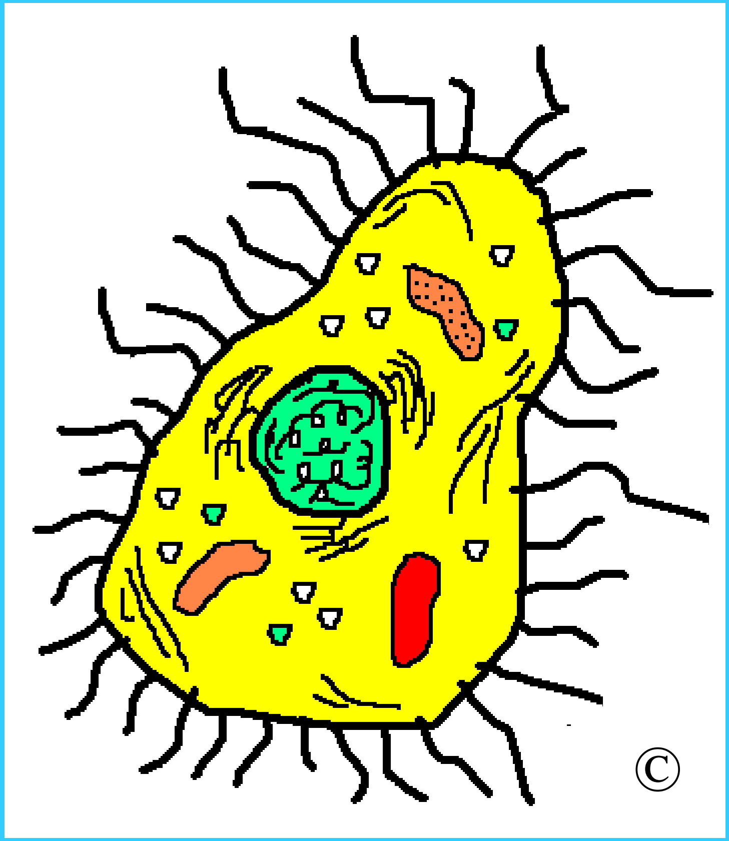
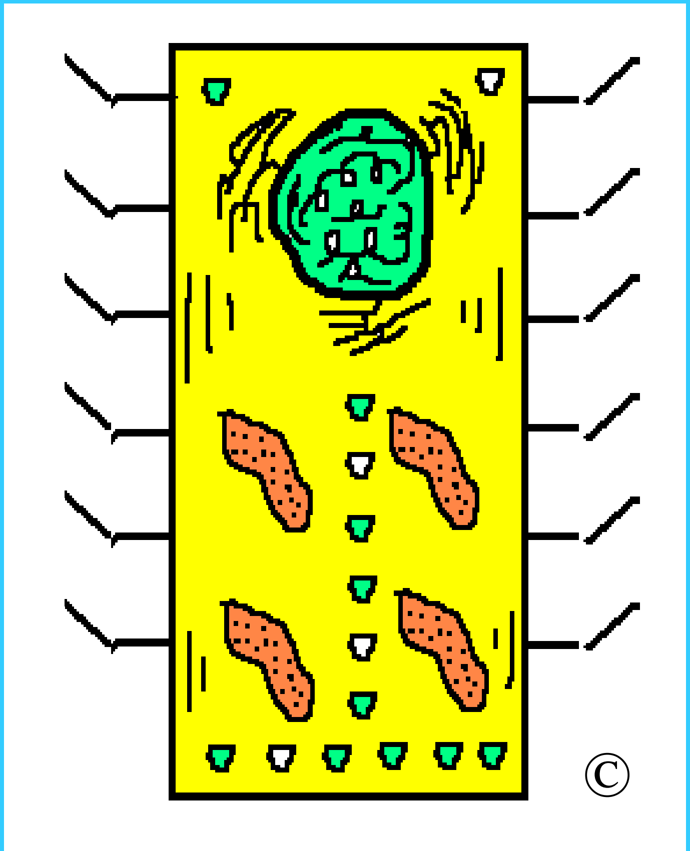
Evolvable hardware :
Darwin dans du silicium

Woozle



FPGA



Algorithme Evolutionniste

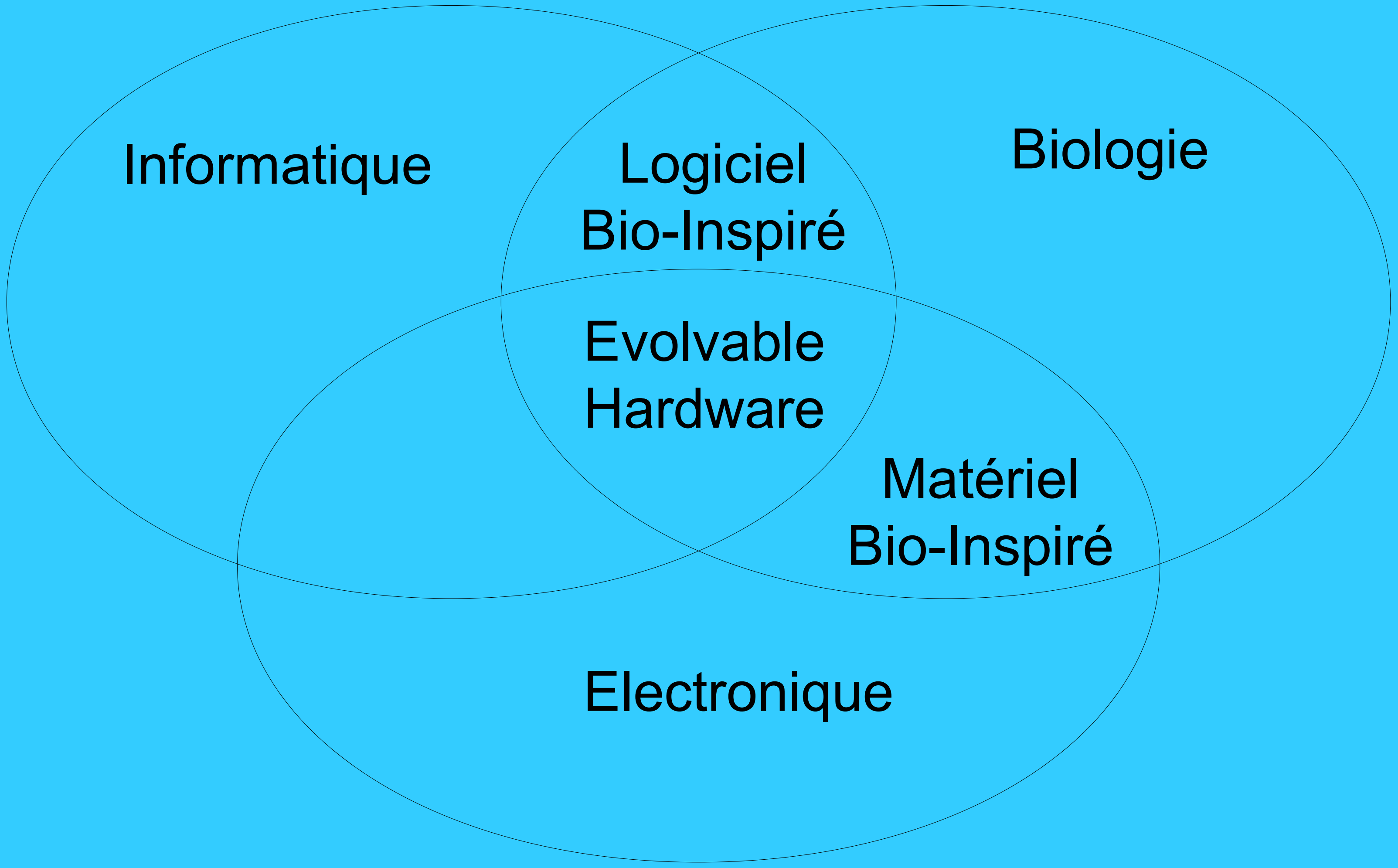
+

Circuit Configurable

=

Evolvable Hardware (EHW)

A la croisée des chemins



Informatique

Logiciel Bio-Inspiré

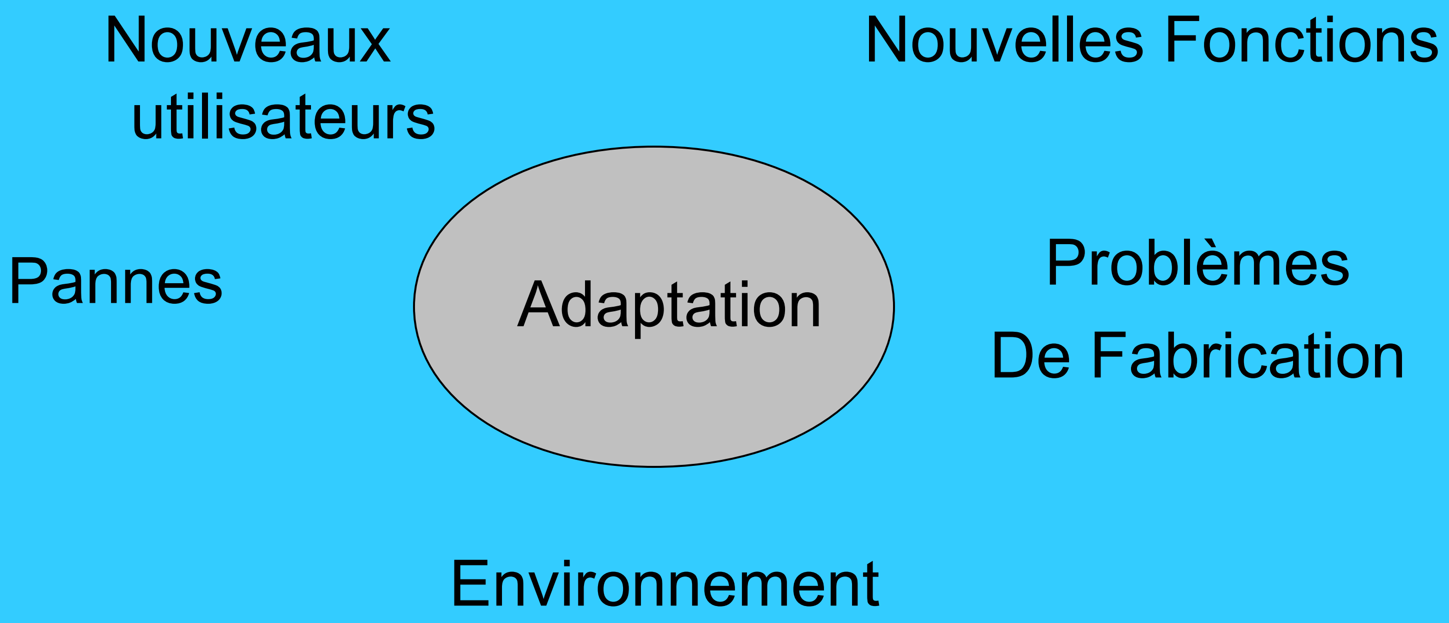
Biologie

Evolvable Hardware

Matériel Bio-Inspiré

Electronique

Evolvable Hardware
le circuit auto-adaptable



Nouveaux utilisateurs

Nouvelles Fonctions

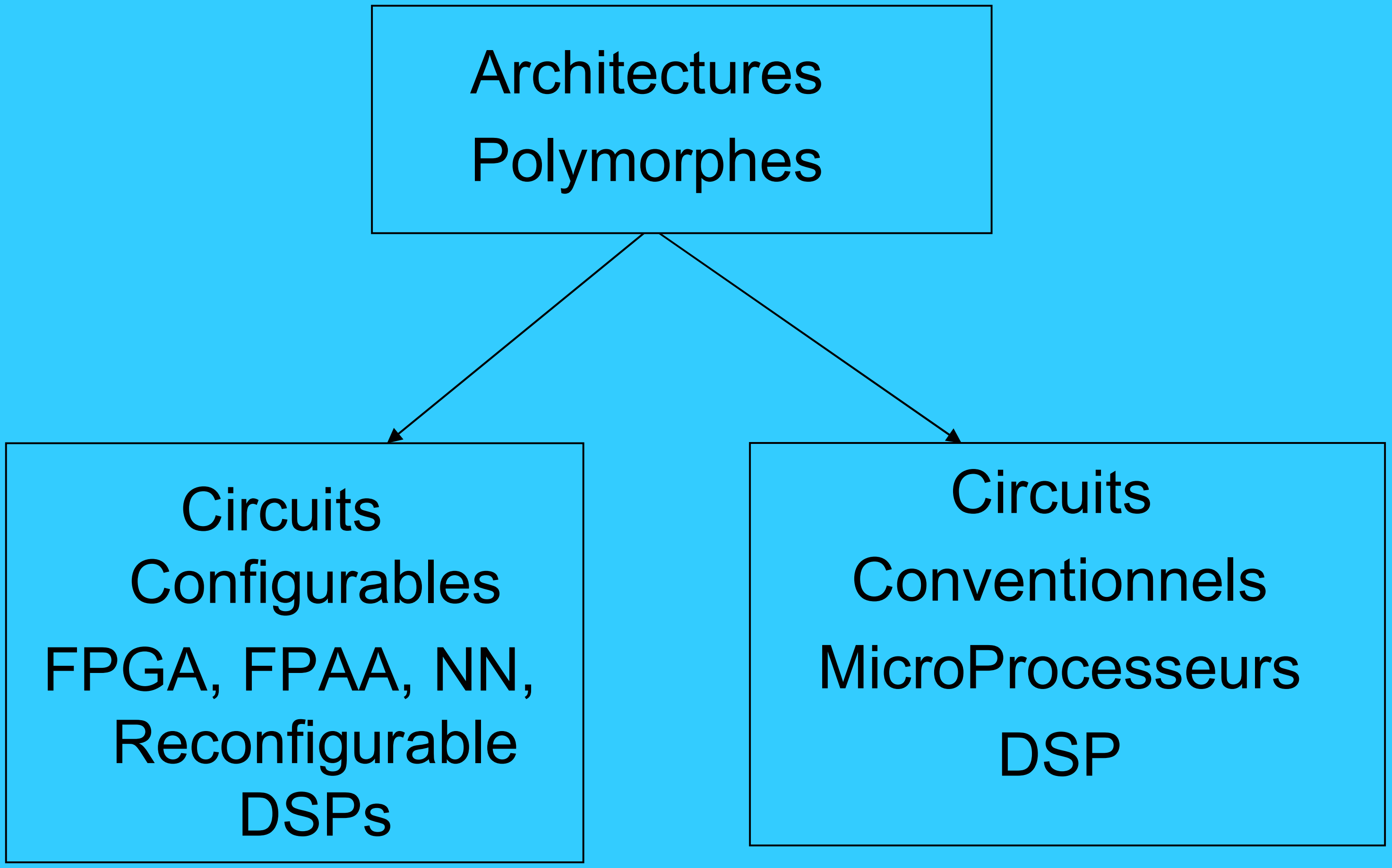
Pannes

Adaptation

Problèmes De Fabrication

Environnement

Processus d'adaptation



Architectures Polymorphes

Circuits Configurables

FPGA, FPAA, NN, Reconfigurable DSPs

Circuits Conventionnels

MicroProcesseurs

DSP

Polymorphe :

Qui est sujet à changer de forme,
qui offre des formes différentes.

Dictionnaire de l'Académie française, huitième édition (1932-1935)

Niveau d'adaptation

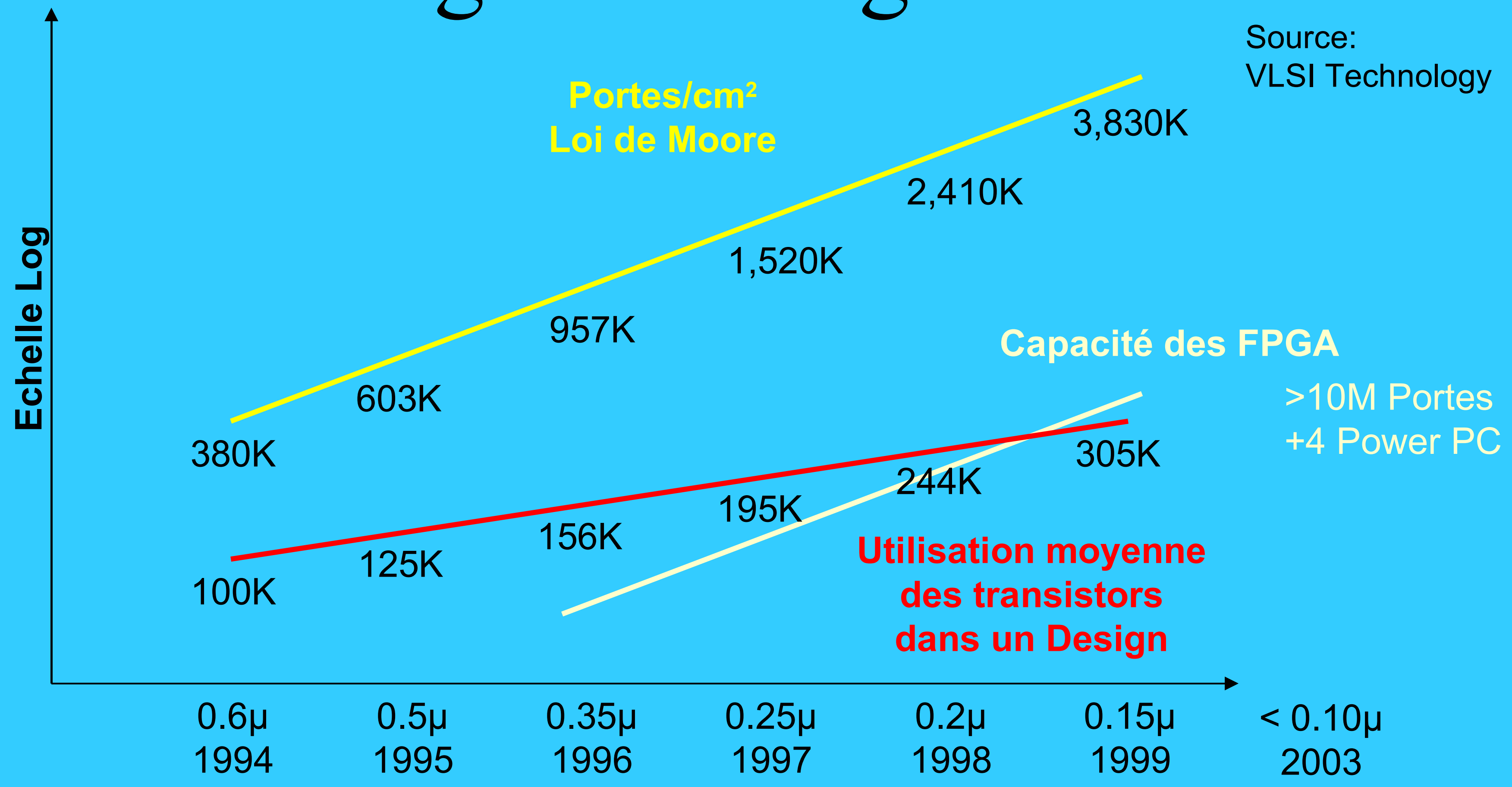
L'adaptation peut être effectué à différents niveaux

- Au niveau du système où des architectures polymorphes recombinent entre elles des ressources hétérogènes
- Au niveau circuit numérique (FPGA, reconfigurable DSP), analogique (FPAA), neuronaux

Environment Aware Devices

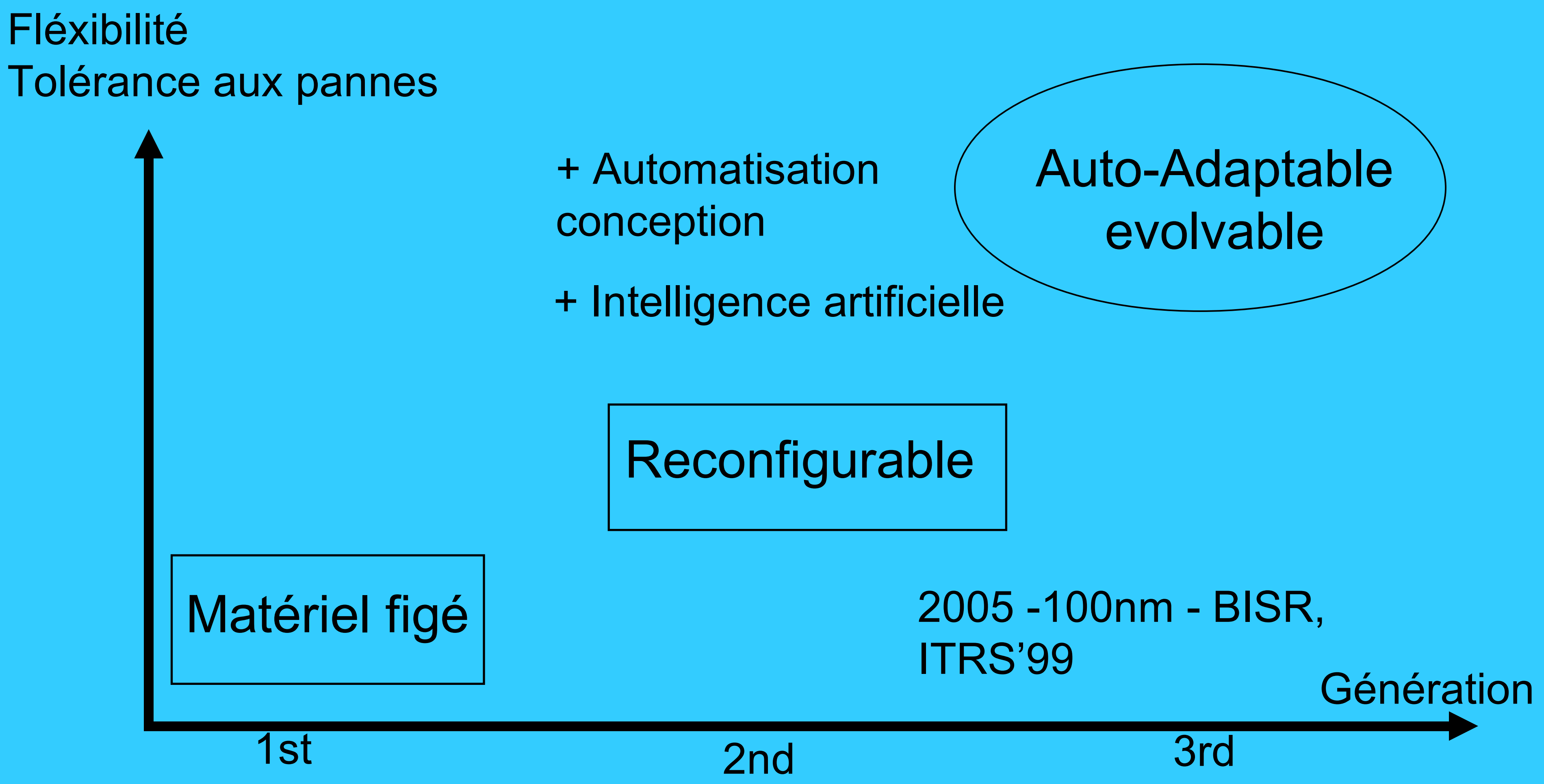
- Environment aware devices adapt to environment.
- For example when the battery is full they operate at high frequency and high resolution. If battery is low, frequency is reduced and resolution is reduced.
- For example an A/D converter that operates like this:
  - If battery level is good – F=100MHz, Resolution = 16 Bit
  - If battery is low – F=10Khz, Resolution is 8 bit

Faille de productivité de l'ingénieur augmente



Les circuits sont plus gros que ce que nous savons produire
Même les FPGA ont plus de transistors que nous ne savons
En utiliser

Une nouvelle génération de circuits



Fléxibilité
Tolérance aux pannes

+ Automatisation conception

+ Intelligence artificielle

Auto-Adaptable
evolvable

Reconfigurable

Matériel figé

2005 -100nm - BISR, ITRS’99

Génération

1st

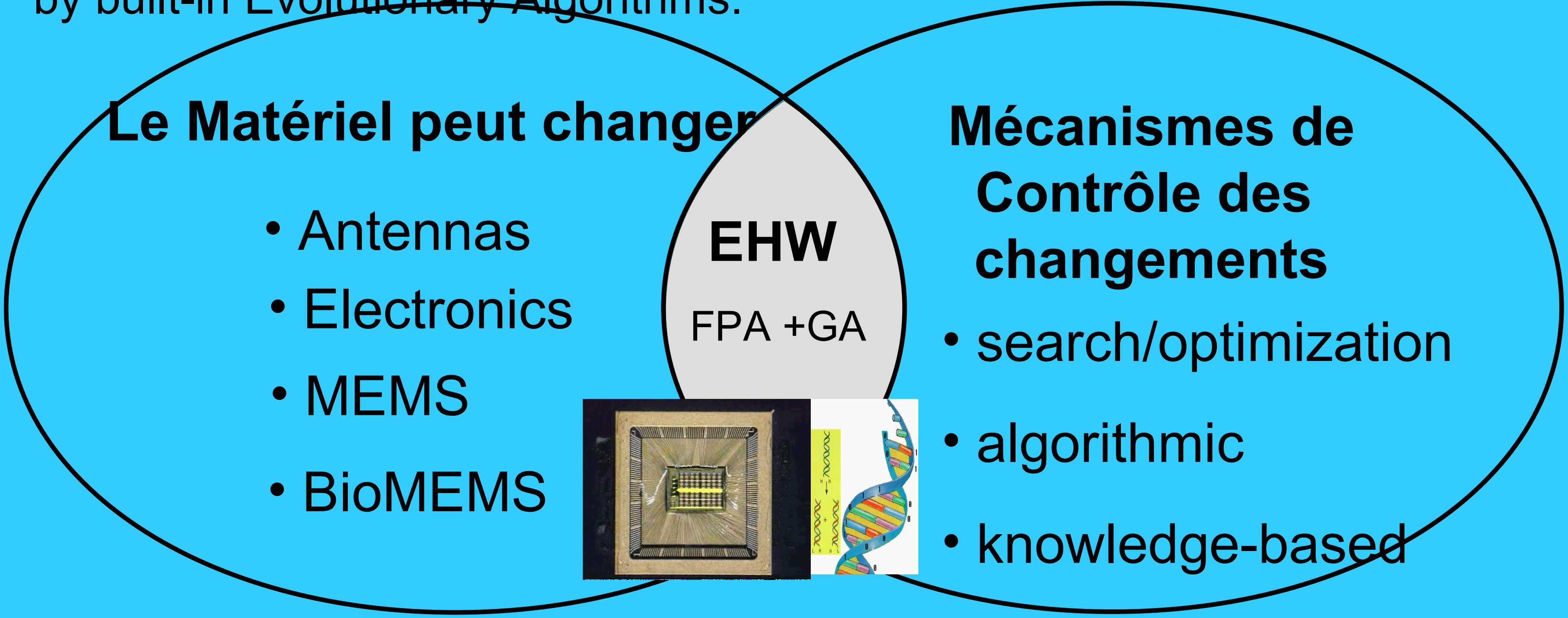
2nd

3rd

Composant d'un système Auto-Adaptable Evolvable Hardware

volvable Hardware = Matériel Reconfigurable + Mécanisme de Reconfiguration

In a narrow sense (EHW) is programmable hardware self-configurable by built-in Evolutionary Algorithms.



Le Matériel peut changer

- Antennas
- Electronics
- MEMS
- BioMEMS

EHW

FPA +GA

Mécanismes de Contrôle des changements

- search/optimization
- algorithmic
- knowledge-based

Same components for intelligent mixed-signal microsystems

Flexible reconfigurable analog/mixed-signal devices

intelligent part – the built-in mechanisms that would control the adaptation/self-configuration

Evolutionary algorithms: inspiration from Nature

“Design” goal: survival

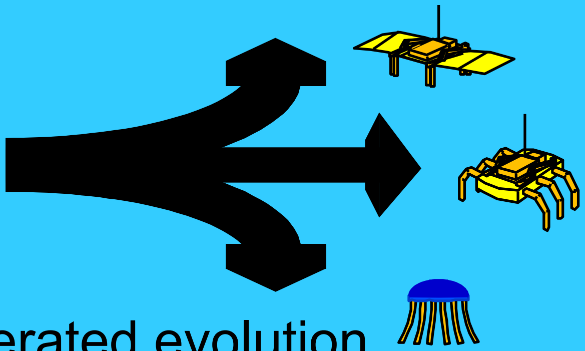
Evolution in nature has lead to species highly adapted to their environment:
adaptation
ensured survival.

Millions of years

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.



Accelerated evolution,
~ seconds for electronics

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

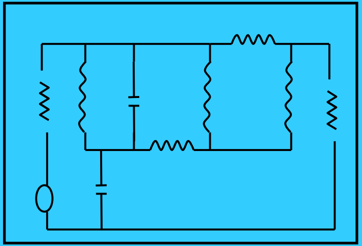
```
0 WhileTooFarFromWall
1   Do2
2   MoveForward
3   Do2
4   WhileInCoridorRange
5   TurnAwayFromClosestWall
6   WhileInCoridorRange
7   Do2
8   TurnParallelToClosestWall
9   MoveForward
```

Model of Hardware

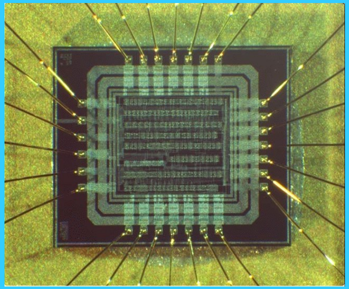
SPICE Netlist

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}
```



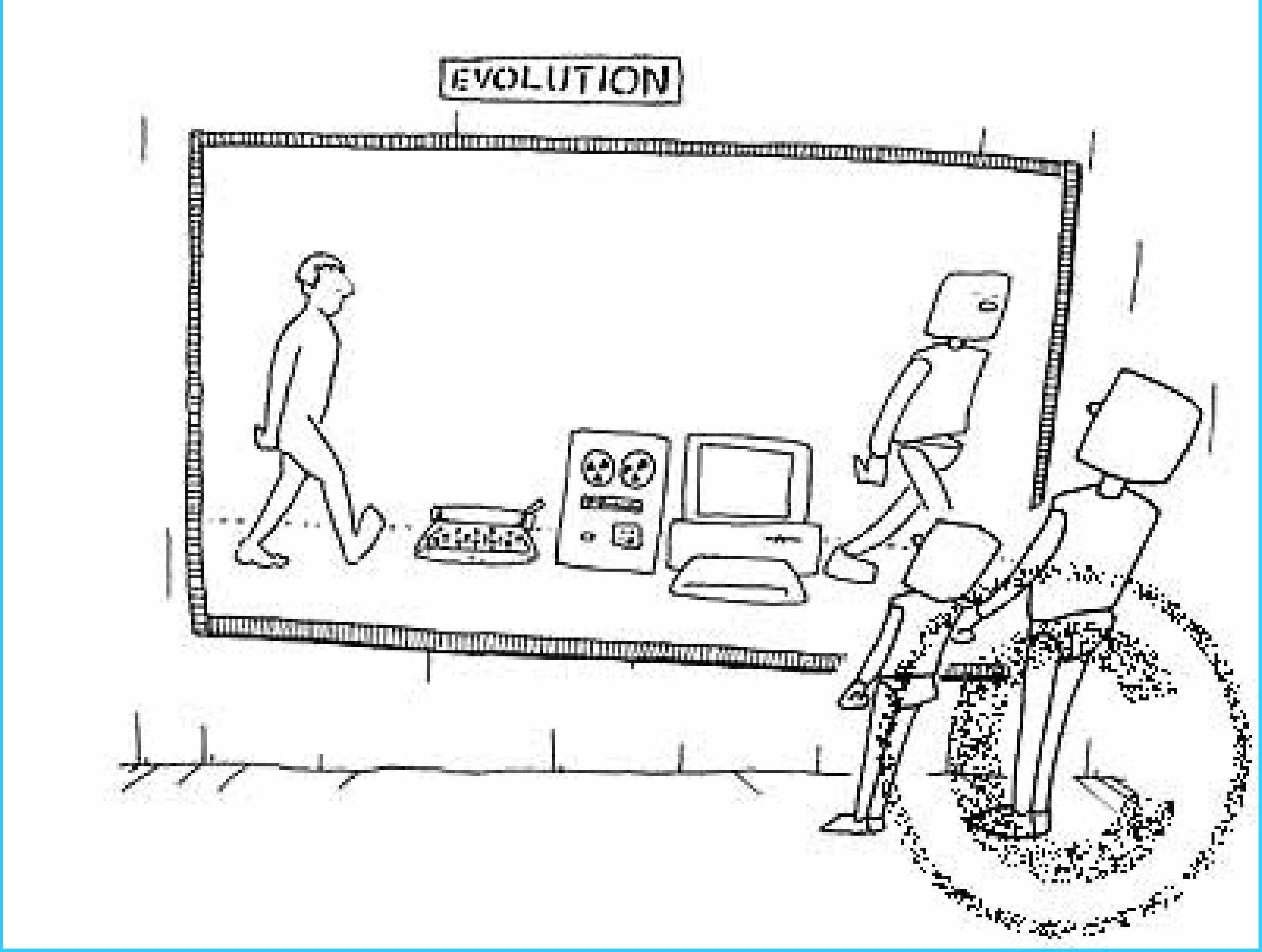
Physical Hardware



Evolutionary is Revolutionary!

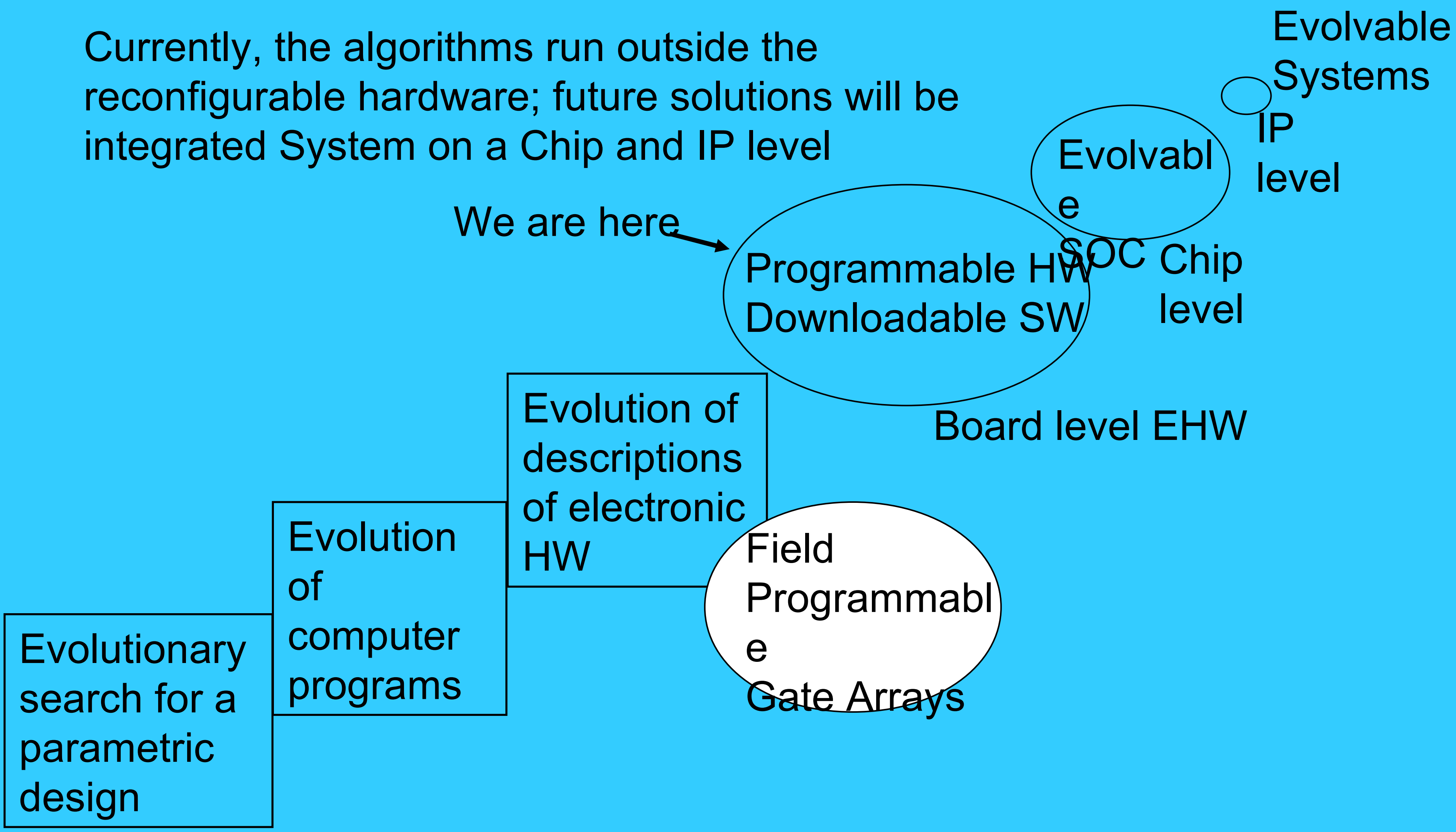
Evolution

http://www.oneonta.edu/~anthro/anth130/cartoons.html



Evolution of EHW

Currently, the algorithms run outside the reconfigurable hardware; future solutions will be integrated System on a Chip and IP level



Evolvable Systems

IP level

Evolvabl e SOC

Chip level

We are here

Programmable HW
Downloadable SW

Board level EHW

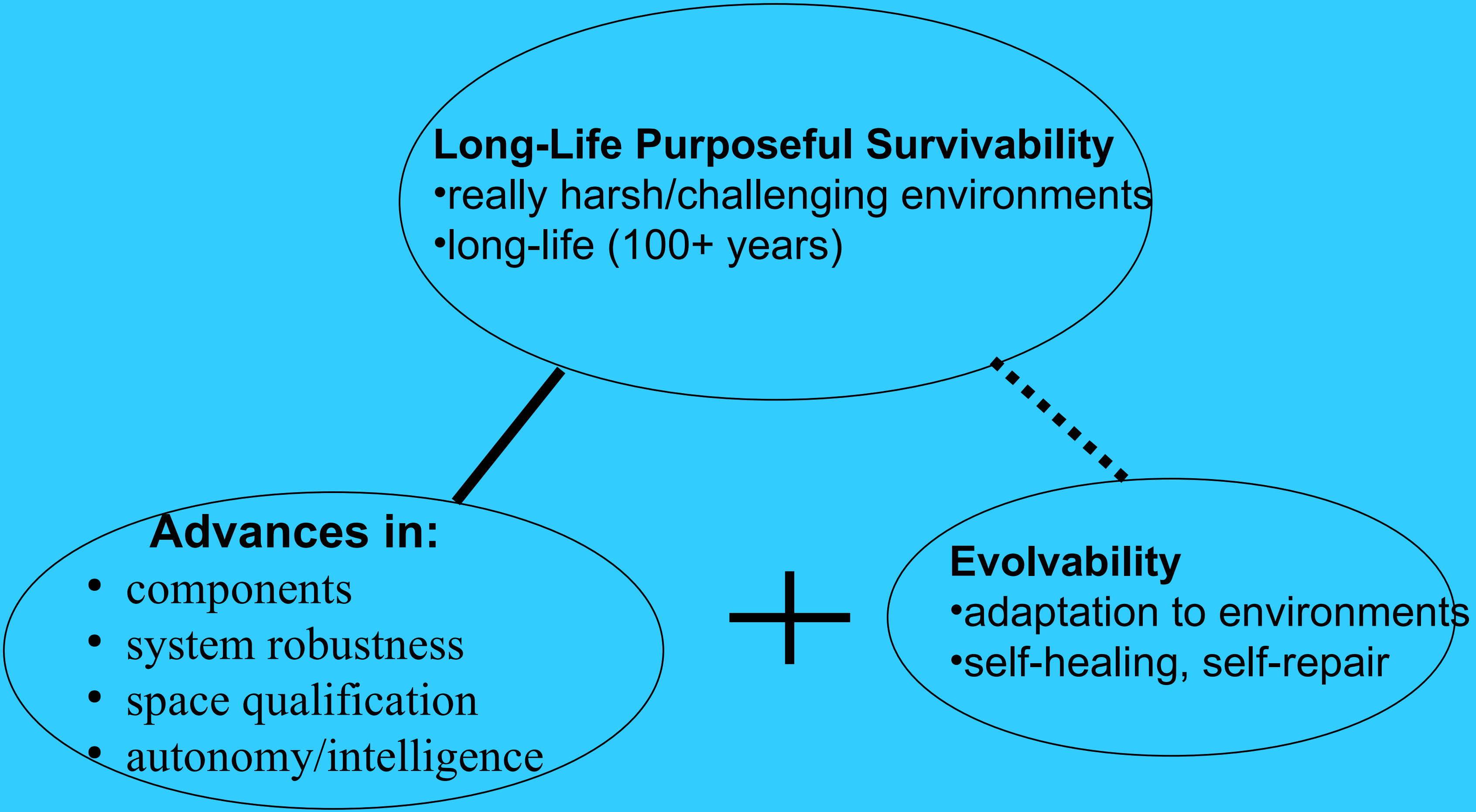
Evolutionary search for a parametric design

Evolution of computer programs

Evolution of descriptions of electronic HW

Field Programmabl e
Gate Arrays

Add evolvability to increase survivability



Long-Life Purposeful Survivability

•really harsh/challenging environments

•long-life (100+ years)

Advances in:

- components
- system robustness
- space qualification
- autonomy/intelligence

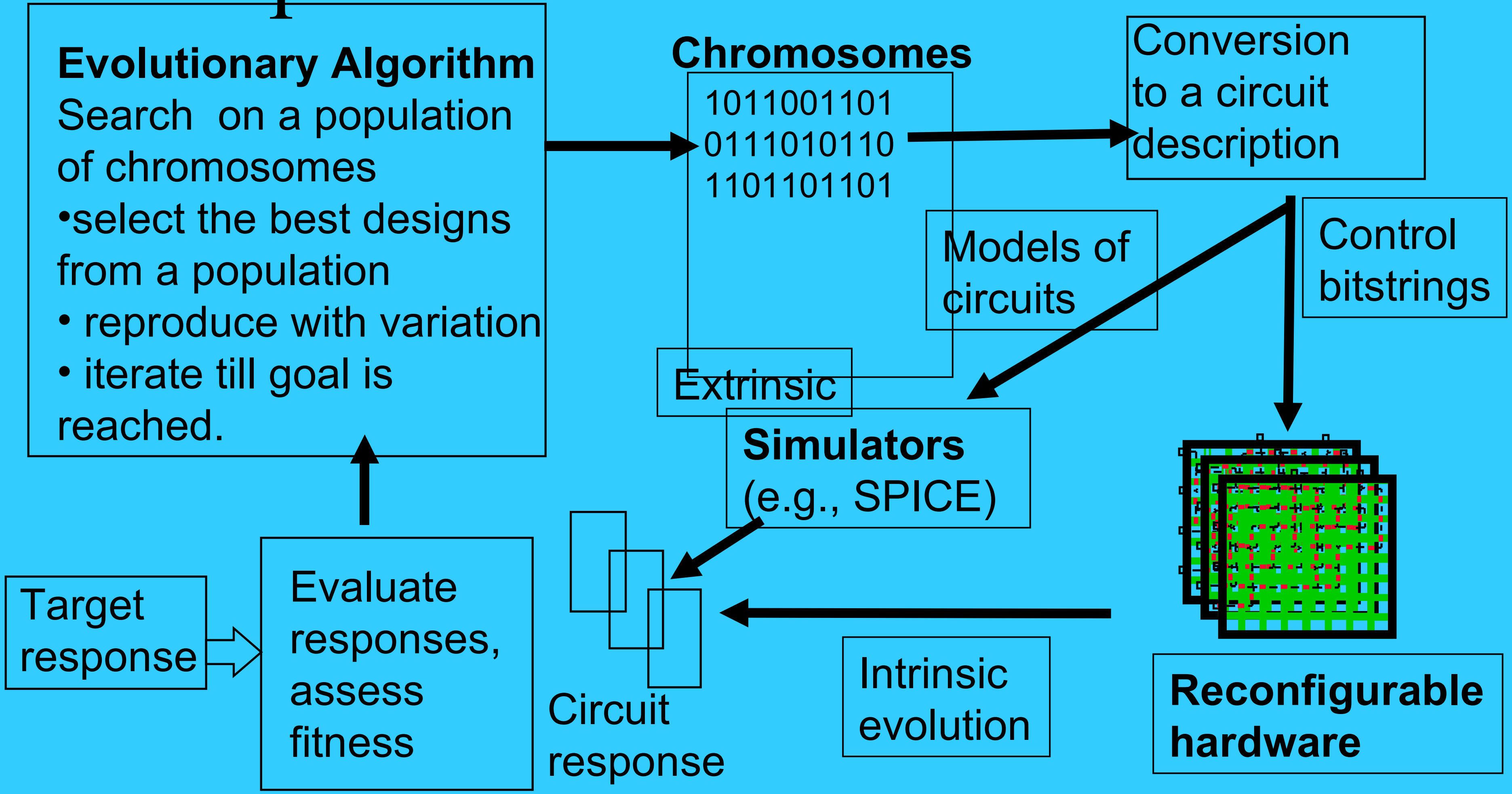
+

Evolvability

•adaptation to environments

•self-healing, self-repair

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)

Reconfigurable hardware

Intrinsic evolution

Circuit response

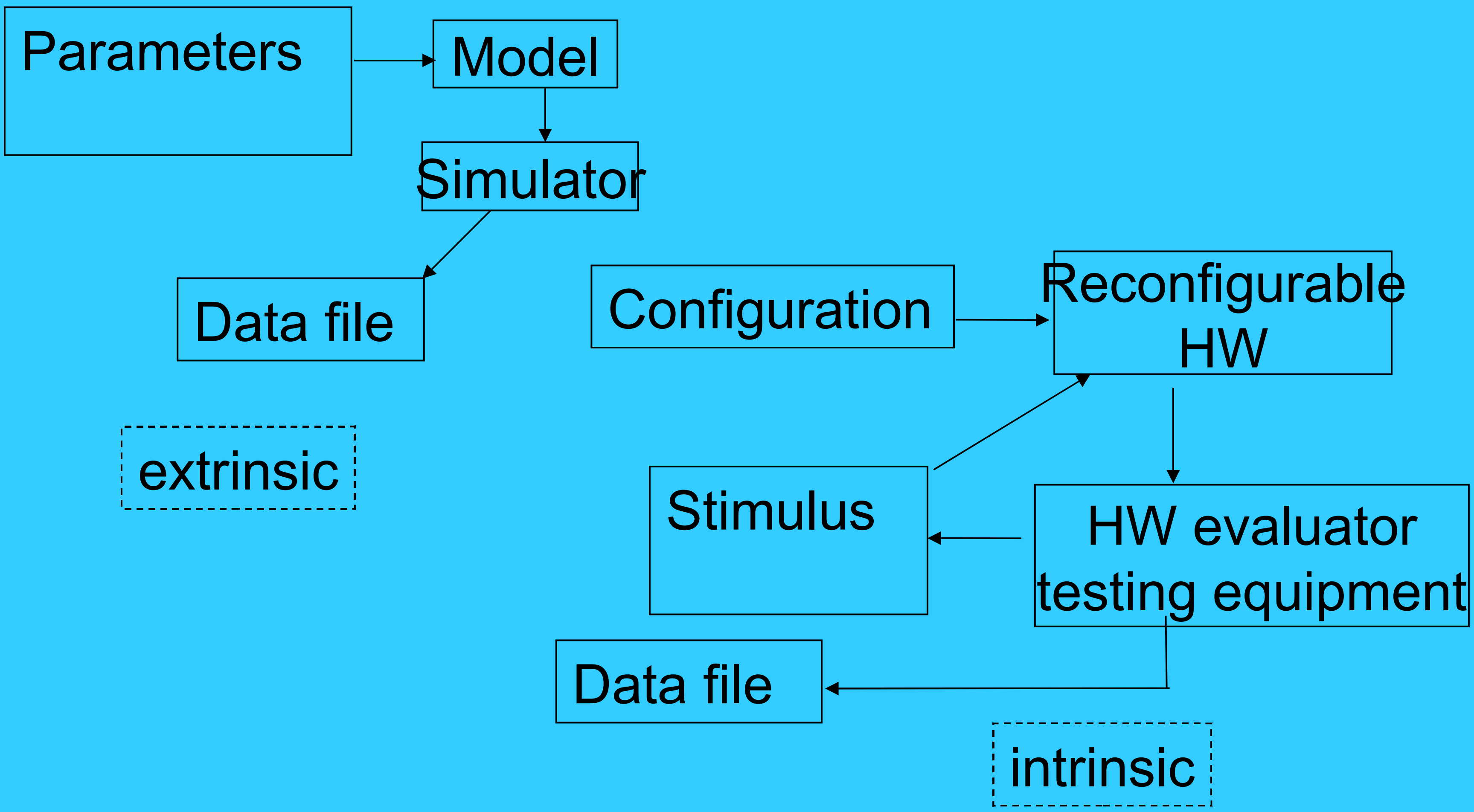
Target response

Evaluate responses, assess fitness

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



Parameters

Model

Simulator

Data file

extrinsic

Configuration

Reconfigurable HW

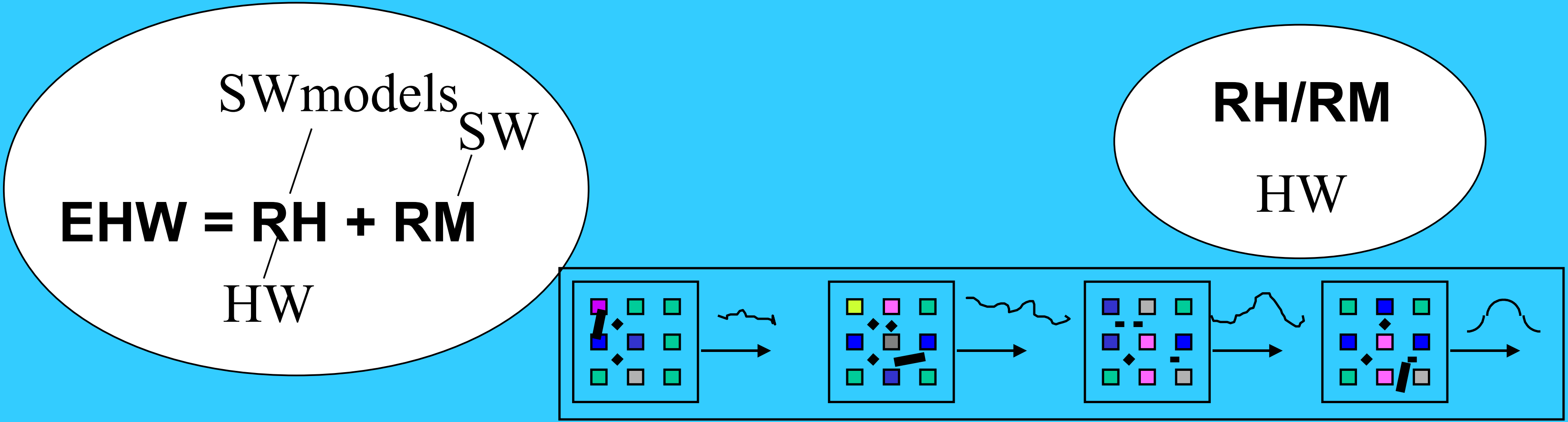
Stimulus

HW evaluator testing equipment

Data file

intrinsic

EHW implementation: HW/SW



SWmodels

SW

EHW = RH + RM

HW

RH/RM

HW

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
- Low cost for failure

Present solutions:RM in SW

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

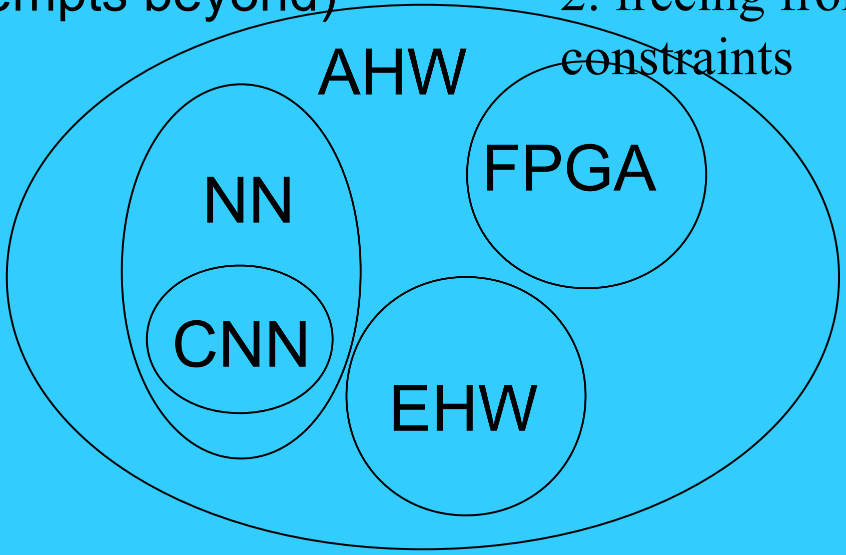
NN seek biological inspiration for

- 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

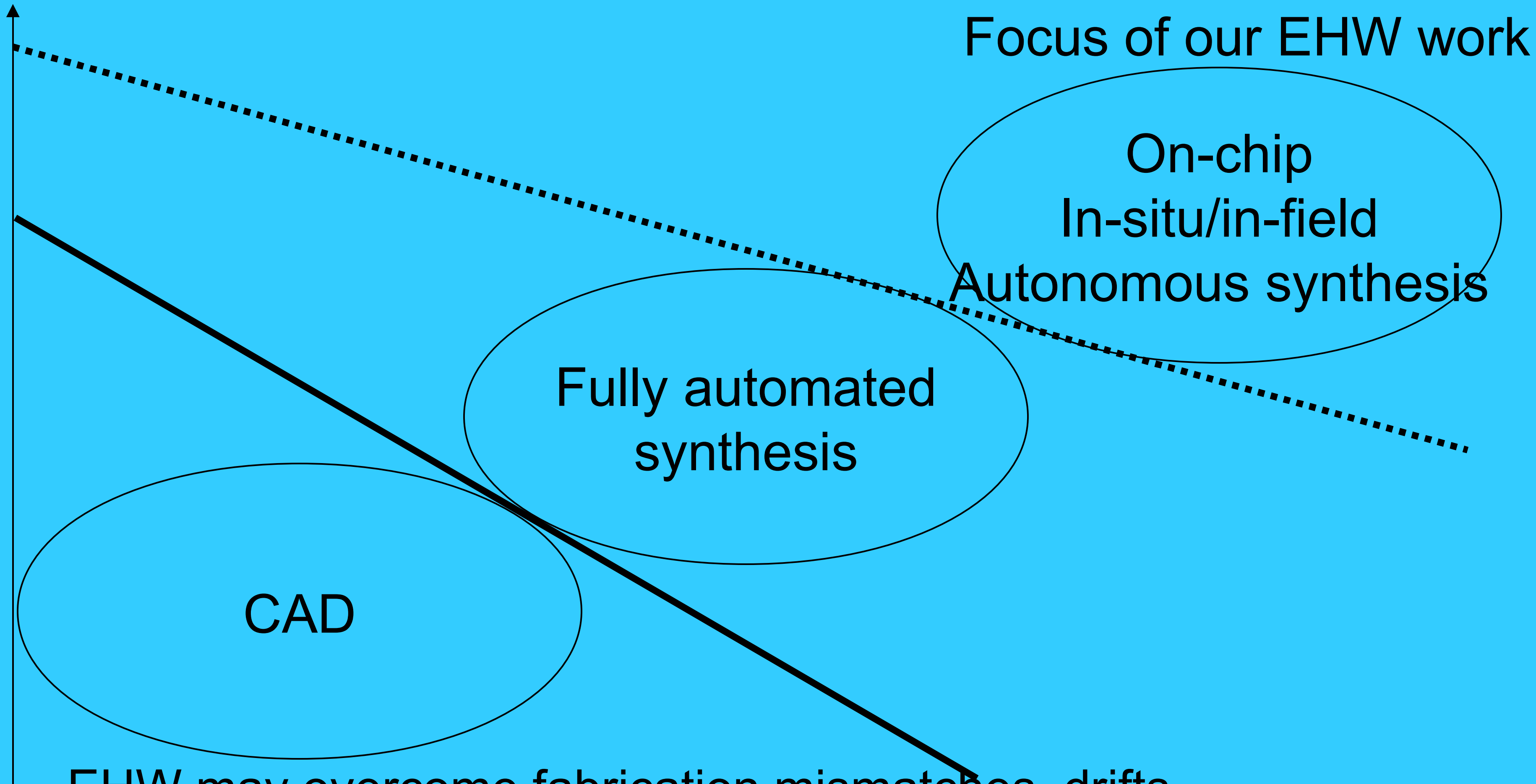


Mechanisms

Building block

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

On-chip EHW vs CAD/synthesis tools



Focus of our EHW work

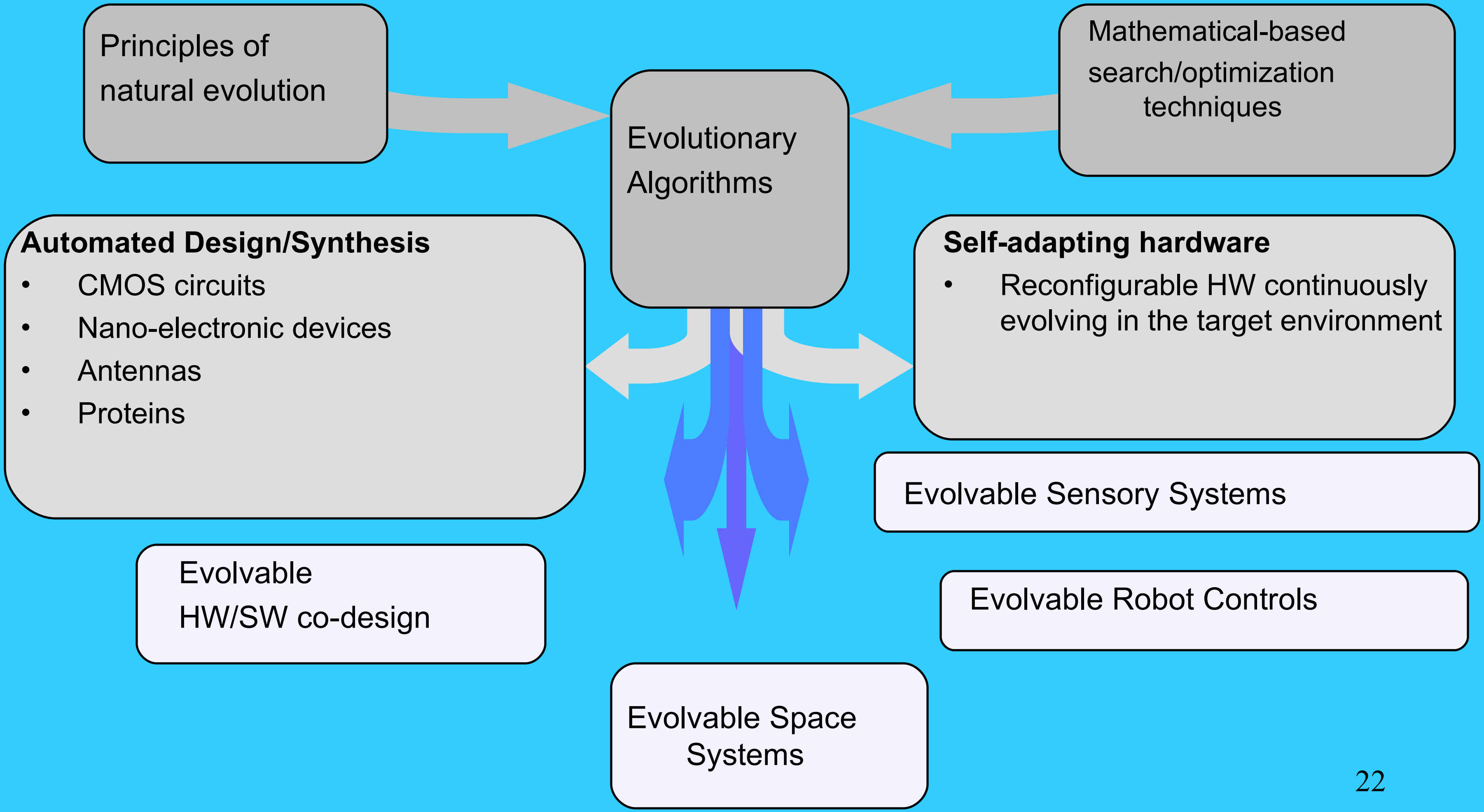
On-chip
In-situ/in-field
Autonomous synthesis

Fully automated
synthesis

CAD

EHW may 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

Evolware: from genetically engineered devices to evolvable space systems



Principles of natural evolution

Mathematical-based search/optimization techniques

Evolutionary Algorithms

Automated Design/Synthesis

- CMOS circuits
- Nano-electronic devices
- Antennas
- Proteins

Self-adapting hardware

- Reconfigurable HW continuously evolving in the target environment

Evolvable Sensory Systems

Evolvable Robot Controls

Evolvable HW/SW co-design

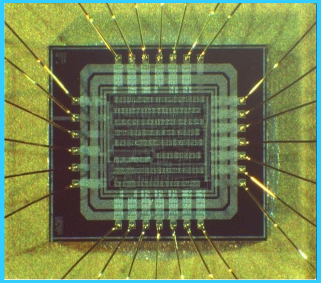
Evolvable Space Systems

What kind of HW is needed for future missions

Temperature & radiation tolerant electronics and long life survivability are key capabilities required for future NASA/ JPL missions.

Ultra long life

Adaptive/Malleable



Autonomous

Extreme environments

- Temperature
- Radiation

Triton 34.5 K

Venus 726K

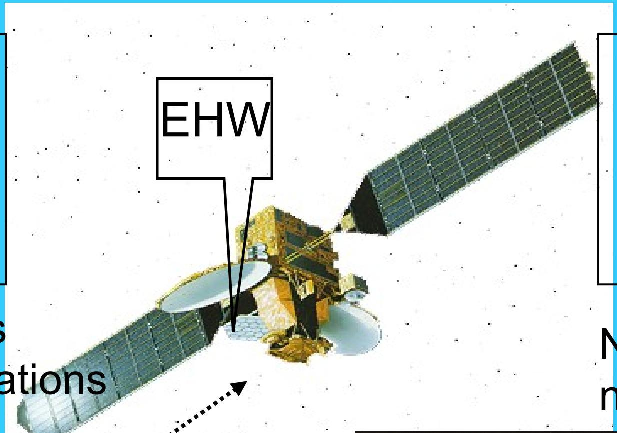
EHW for flexibility and survivability of autonomous systems

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.

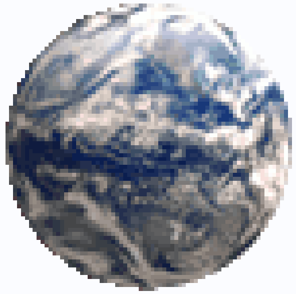
Survivability:
Maintain functionality coping with changes in HW characteristics

- Radiation impacts
- Temperature variations
- Aging
- Malfunctions, etc.



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

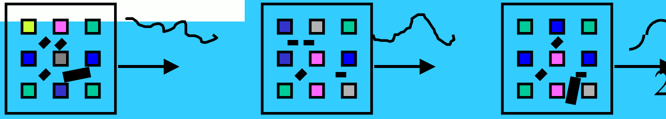
New functions required for new mission phase or opportunity



Up-link new functions for re-planned mission

Accurate model of hardware is not available after launch

Develop space HW that can evolve



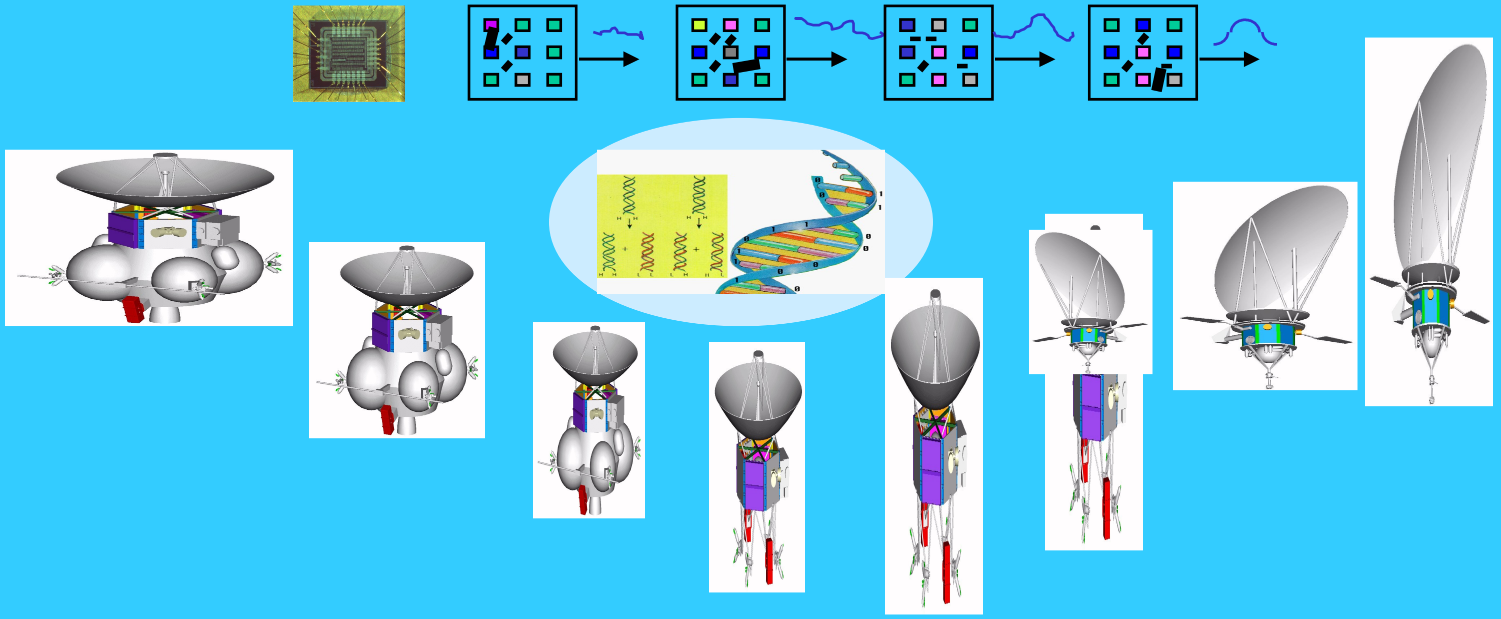
How Evolvable Space Systems would Revolutionize NASA Missions

- Long life, survivable, self-healing space systems
  - would allow long duration/far out missions
  - would harness required power and other resources from environment
- Would enable *evolvable missions* capturing science/exploration opportunities in real time
- Space explorer
  - would produce knowledge from acquired data
  - would use the knowledge to mission refocus/replanning
  - would be able to create new functions, unforeseen before launch
  - would be able to learn on-the-fly to best deal with changing conditions
- Fleet, Swarm, Armada
- Salvaging: some do not adapt, their unharmed resources are reused by survivors

Evolvable system technology:
adaptive platform for space systems in a large variety of missions

Ultimate goal: fully evolvable space systems

- Morphing/plasticity can expand gradually from electronic subsystems to entire space systems.



- Evolution of space systems would include autonomous changes/reconfiguration of both software and hardware: sensors, avionics, structure, ...
- The future may see chameleon-like surface explorers and phoenix-like robotic birds...

Fundamental open questions

- Can we evolve artificial systems in similar ways natural systems evolve? Advantages and disadvantages.
- 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”)
- How does EHW scale-up?
- Can we use evolution to obtain intelligent systems, human competitive (and beyond) intelligence

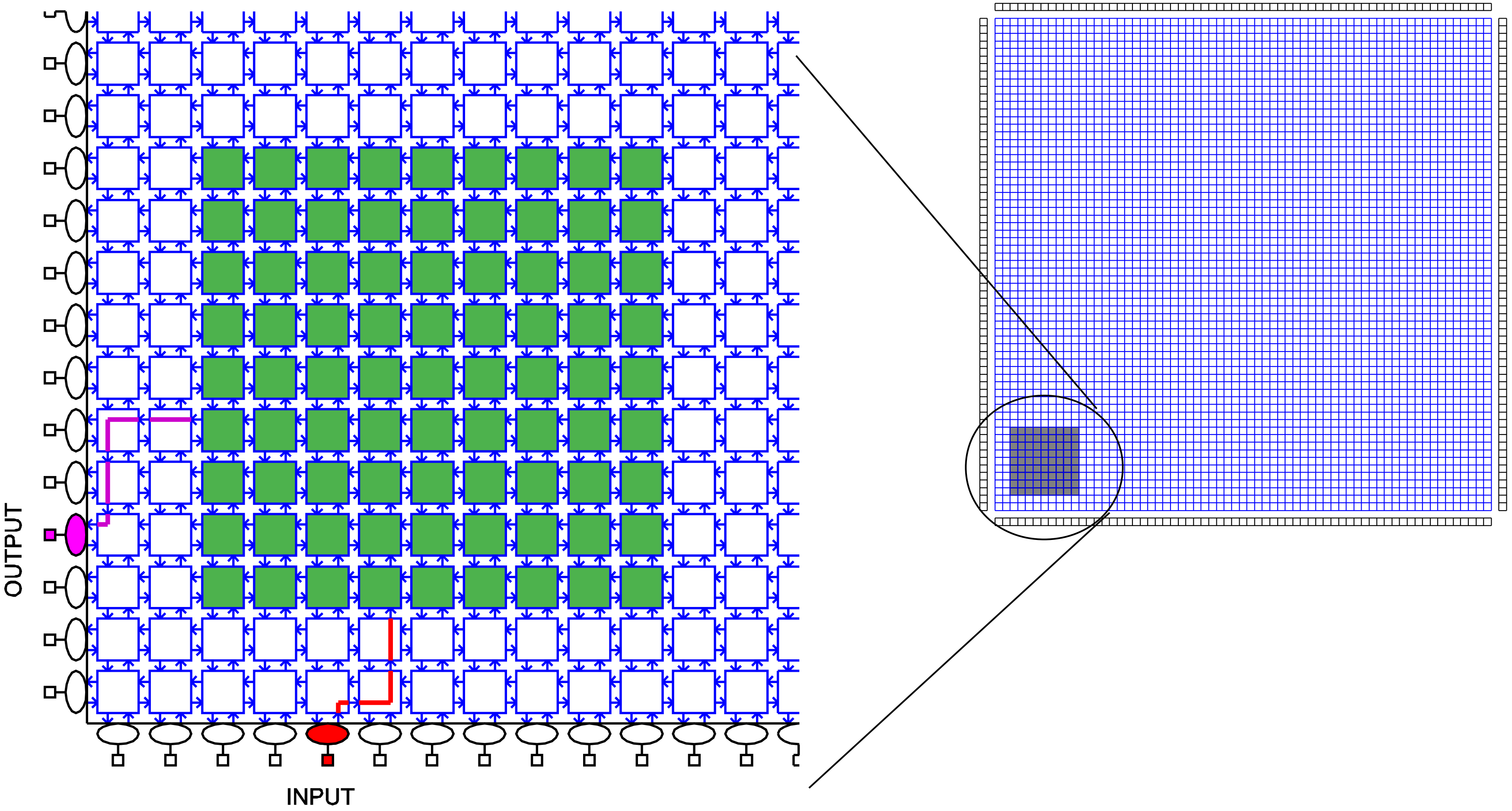
Quelques Exemples

- ETL – Prothèse de la main (Controller, GA)
- Koza – Conception de circuit Analogique (Simulation, Embryology, GP)
- De-Garis - CAM brain machine (FPGA, Embryology)

Matériels configurables

- FPGAs (Field Programmable Gate Arrays)
- Adjustable Controllers
- FPAAs (Field Programmable Analog Arrays)
- Computer Simulation

Le FPGA Xilinx XC6216



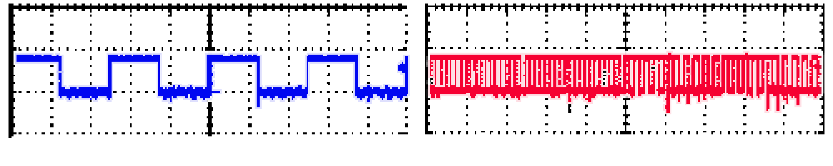
Evolution #1
Monitor Views

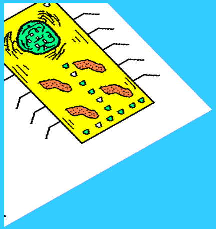
1 kHz

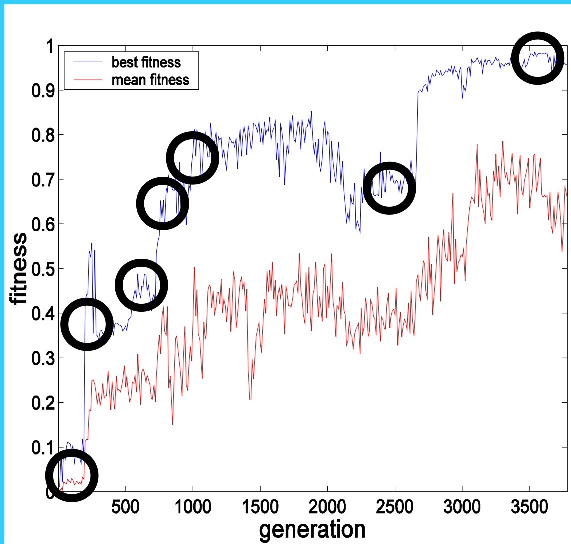
10 kHz

INPUT SIGNALS:

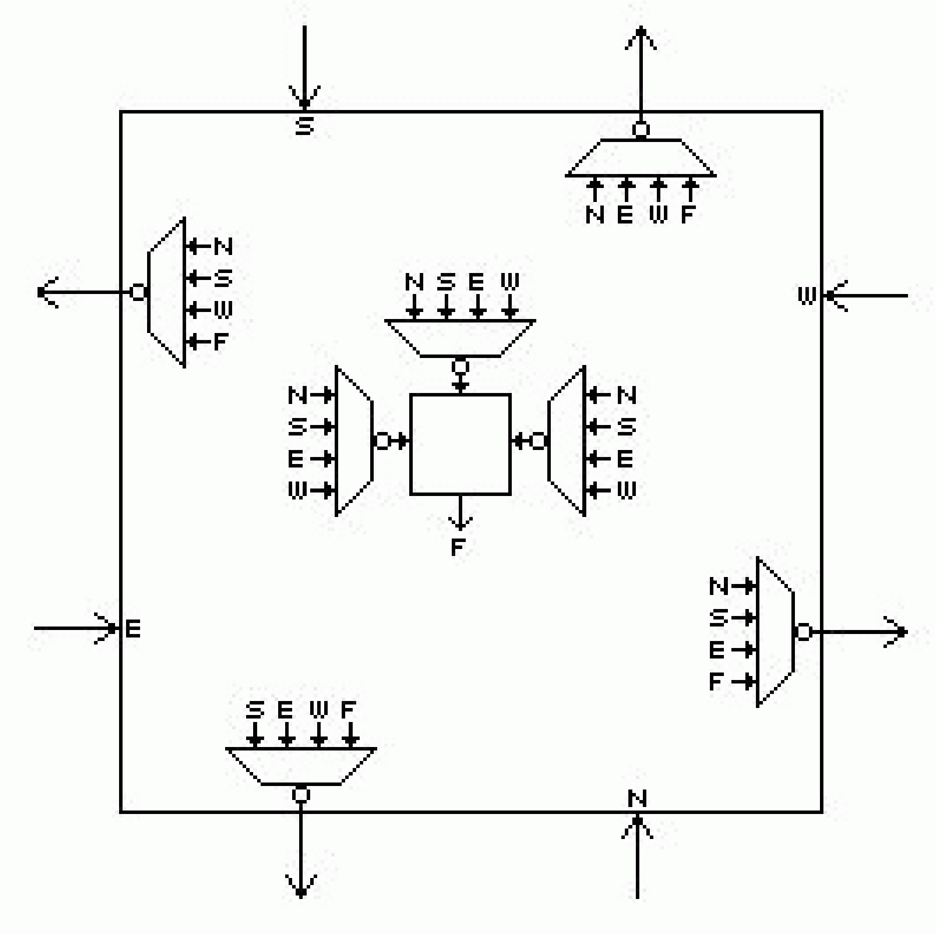
OUTPUT SIGNALS:



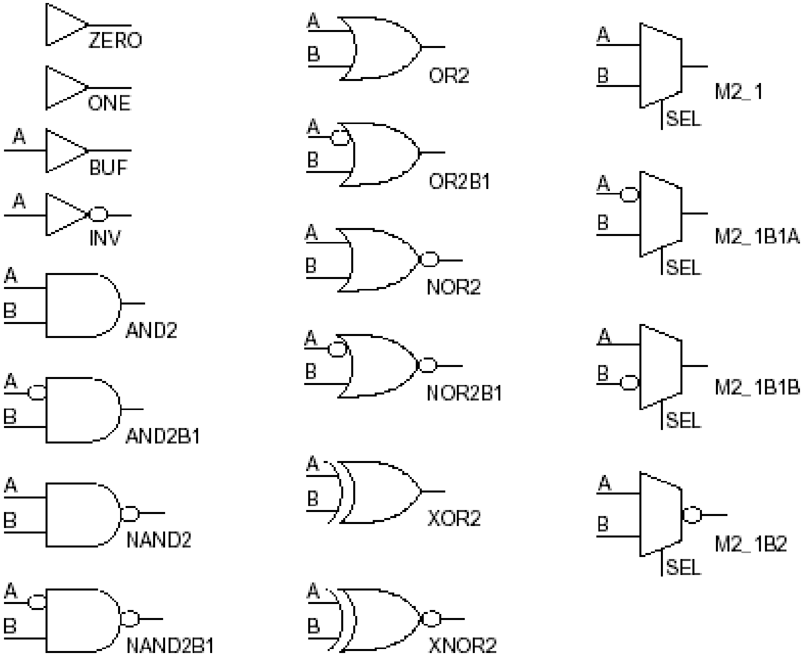


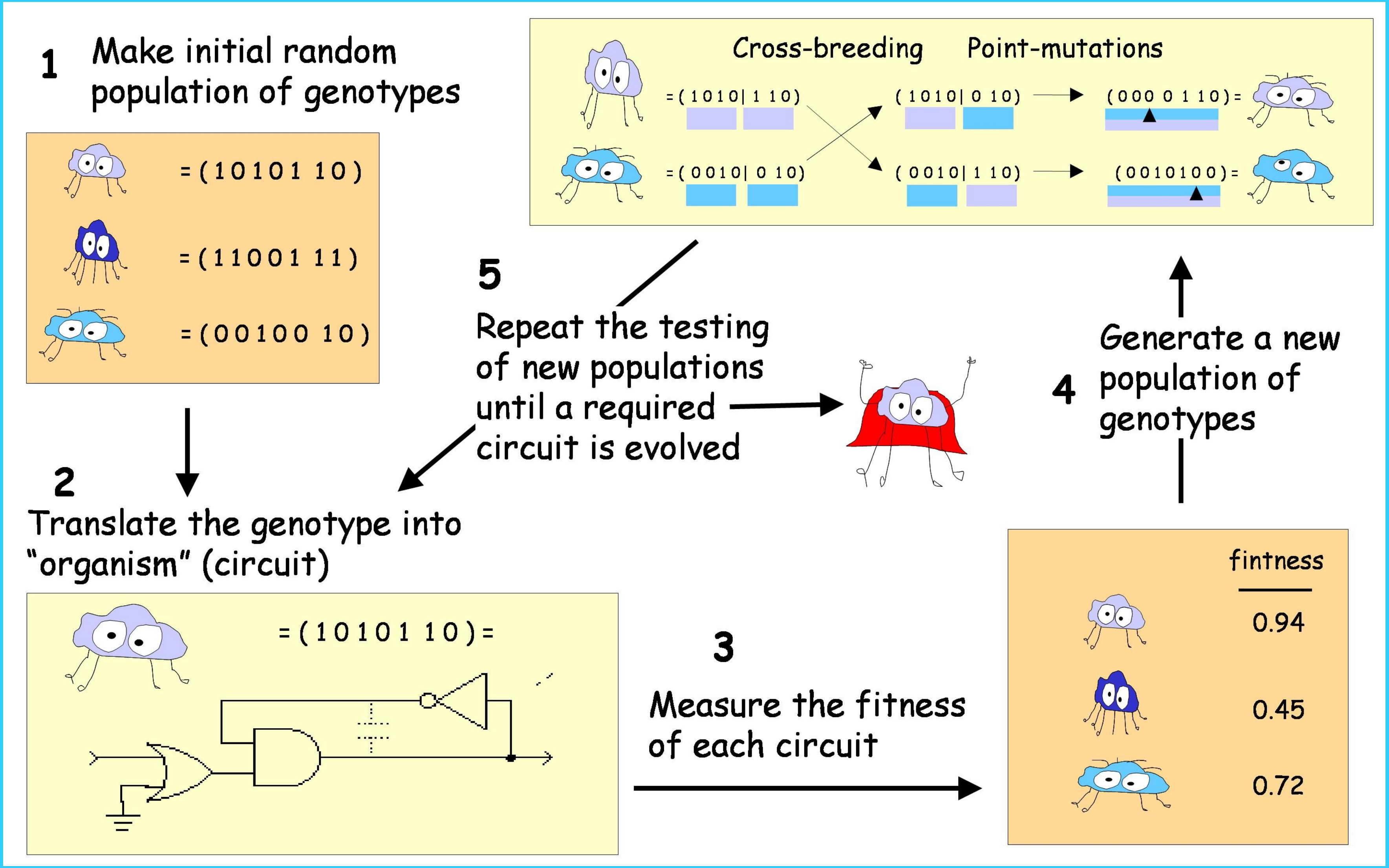


L'unité de base



Fonctions possibles





1 Make initial random population of genotypes

=(10101 10)

=(11001 11)

=(00100 10)

2 Translate the genotype into "organism" (circuit)

=(10101 10)=

3 Measure the fitness of each circuit

fintness

0.94

0.45

0.72

4 Generate a new population of genotypes

Cross-breeding Point-mutations

=(1010| 1 10) (1010| 0 10) → (000 0 1 10)=

=(0010| 0 10) (0010| 1 10) → (0010100)=

5 Repeat the testing of new populations until a required circuit is evolved

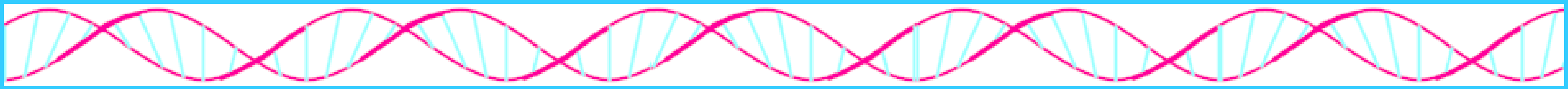
Le code génétique
et
La réalisation du circuit

- 18 bits sont nécessaires pour décrire la fonctionnalité du circuit

- Le circuit est composé de 100 cellules

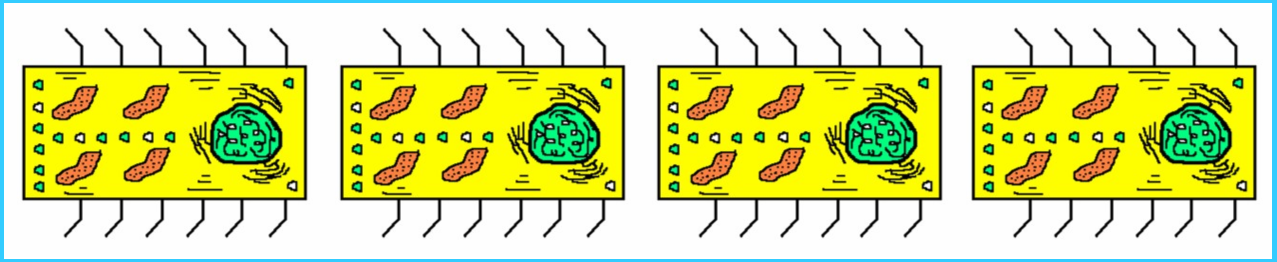
→ Le gêne du circuit mesure 1800 bits

→ l'espace de rechercher est de 2^{1800} circuits différents



GA parameters

- Code length: 1,800 bits
- Population Size: 50
- Mutation probability per bit: 0.15 %
- Crossing probability: 70 %



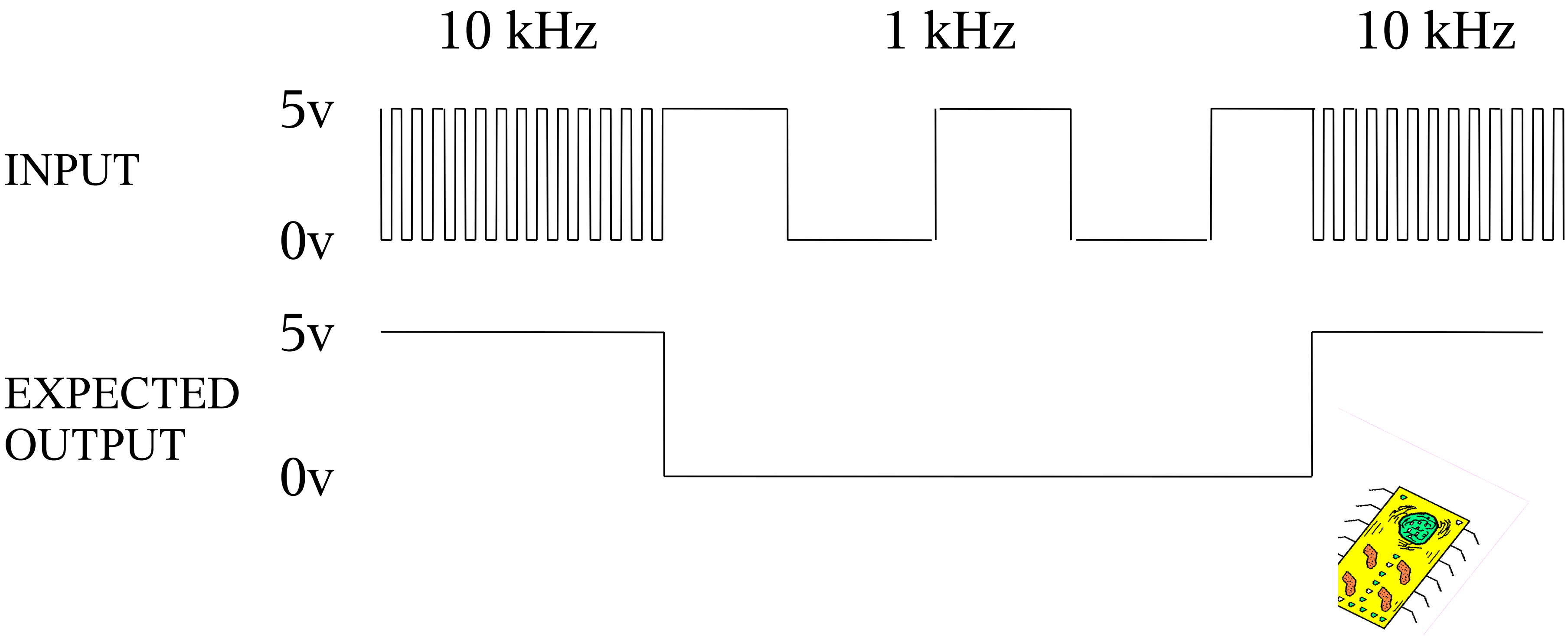
The Challenge:

Analog Task to a Digital Circuit

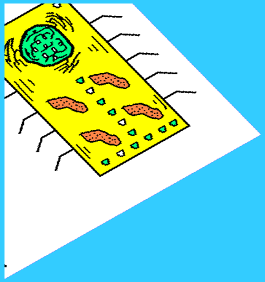
The result: autonomous behavior of a circuit

Self-feeding with stray analog elements

Evolution task: Evolve a frequency discriminating circuit



Fitness Evaluation

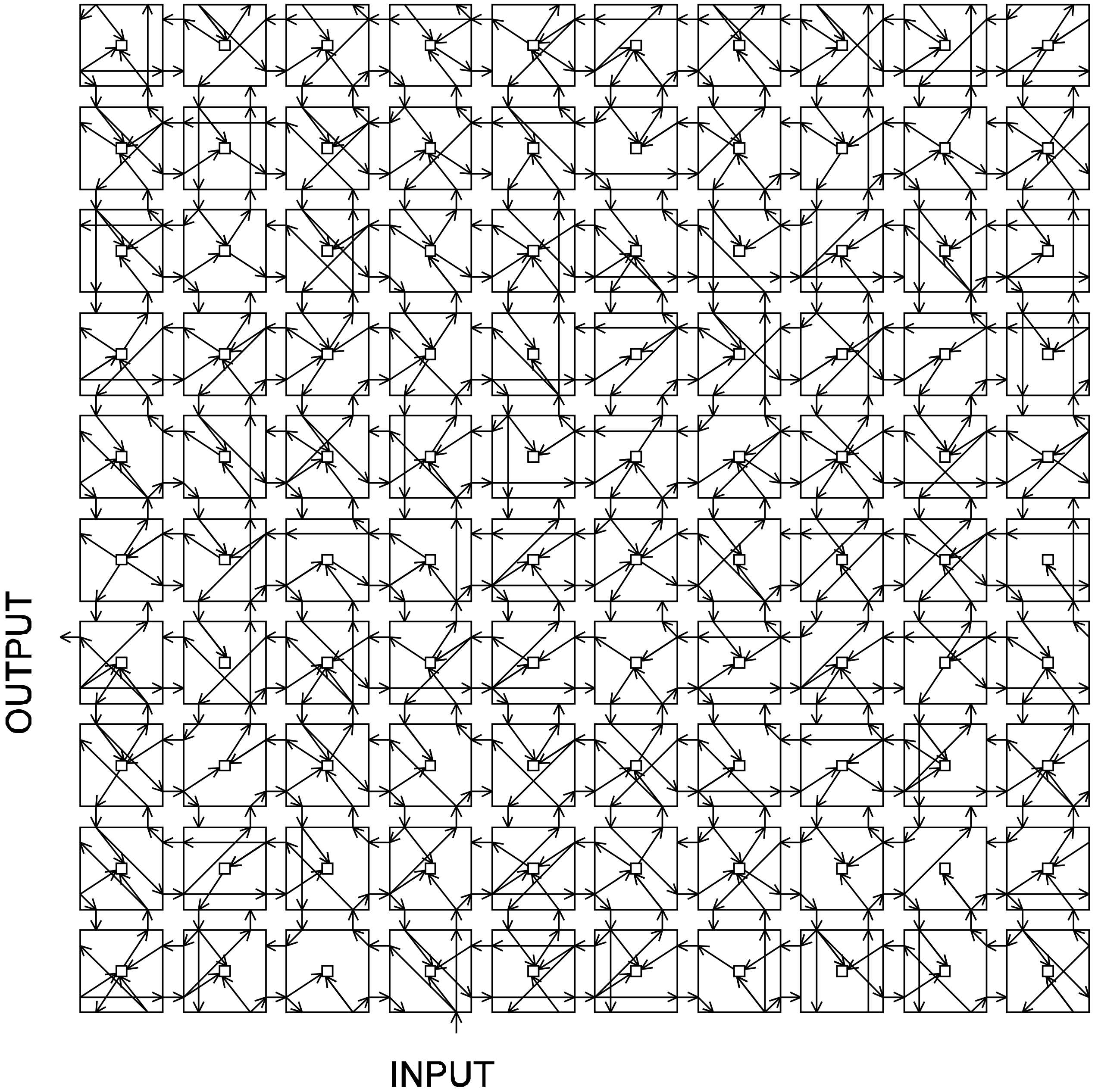


The output of a circuit for each input is sampled 5 times by a 1 MHz oscillator for a time period of 200 ms, and the number of logic "1"s are counted.
Fitness is then calculated by:

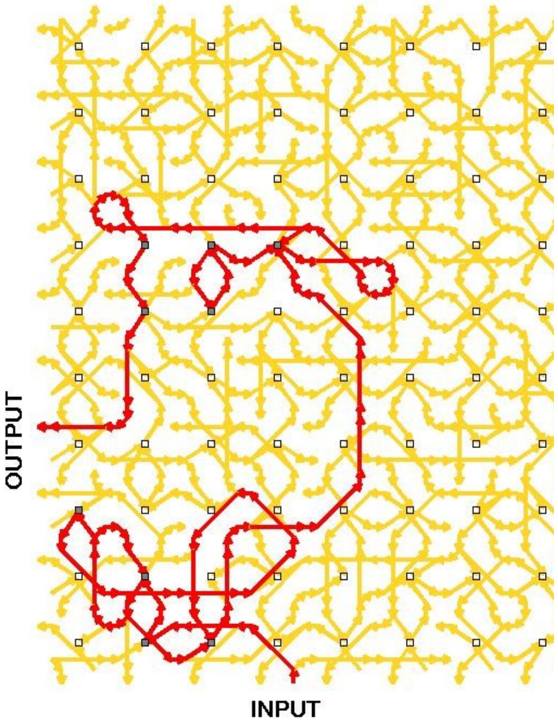
$$f=\frac{1}{5}\left|\sum I_{1\ kHz}-\sum I_{10\ kHz}\right|$$

A total time of 2 sec to evaluate a single circuit.

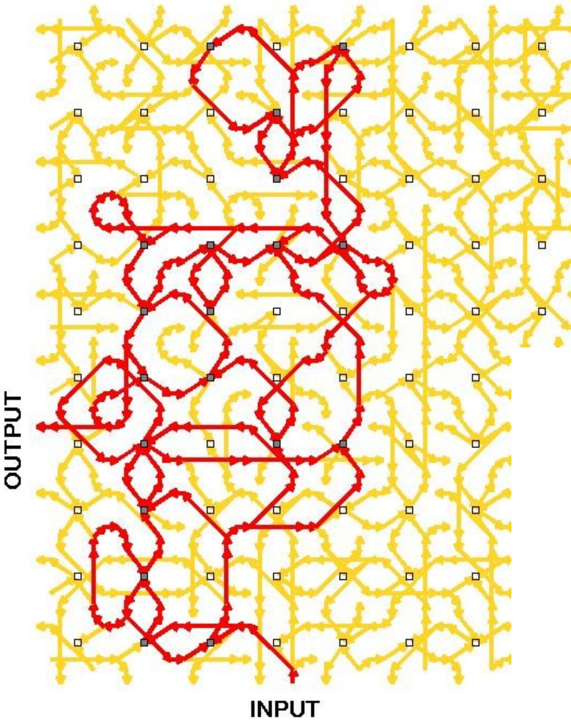
The fitness resolution is ~12 bits (Δf=0.00025).


OUTPUT
INPUT

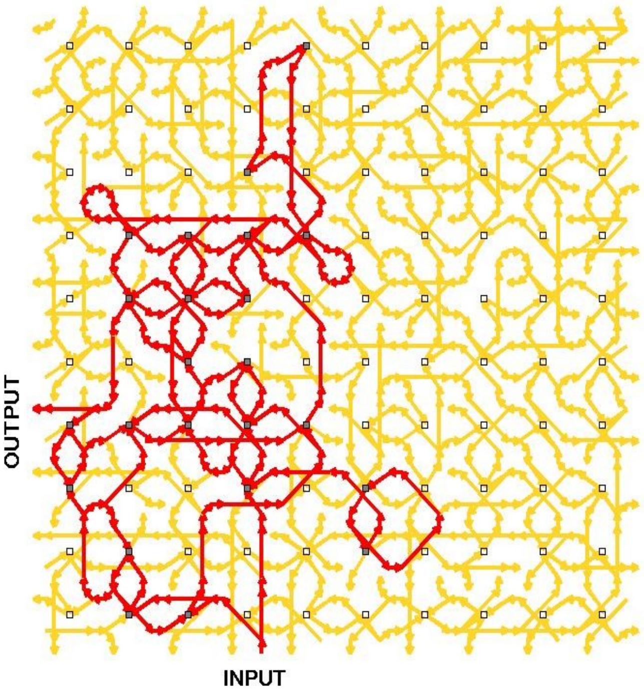
generation 3000



generation 4200

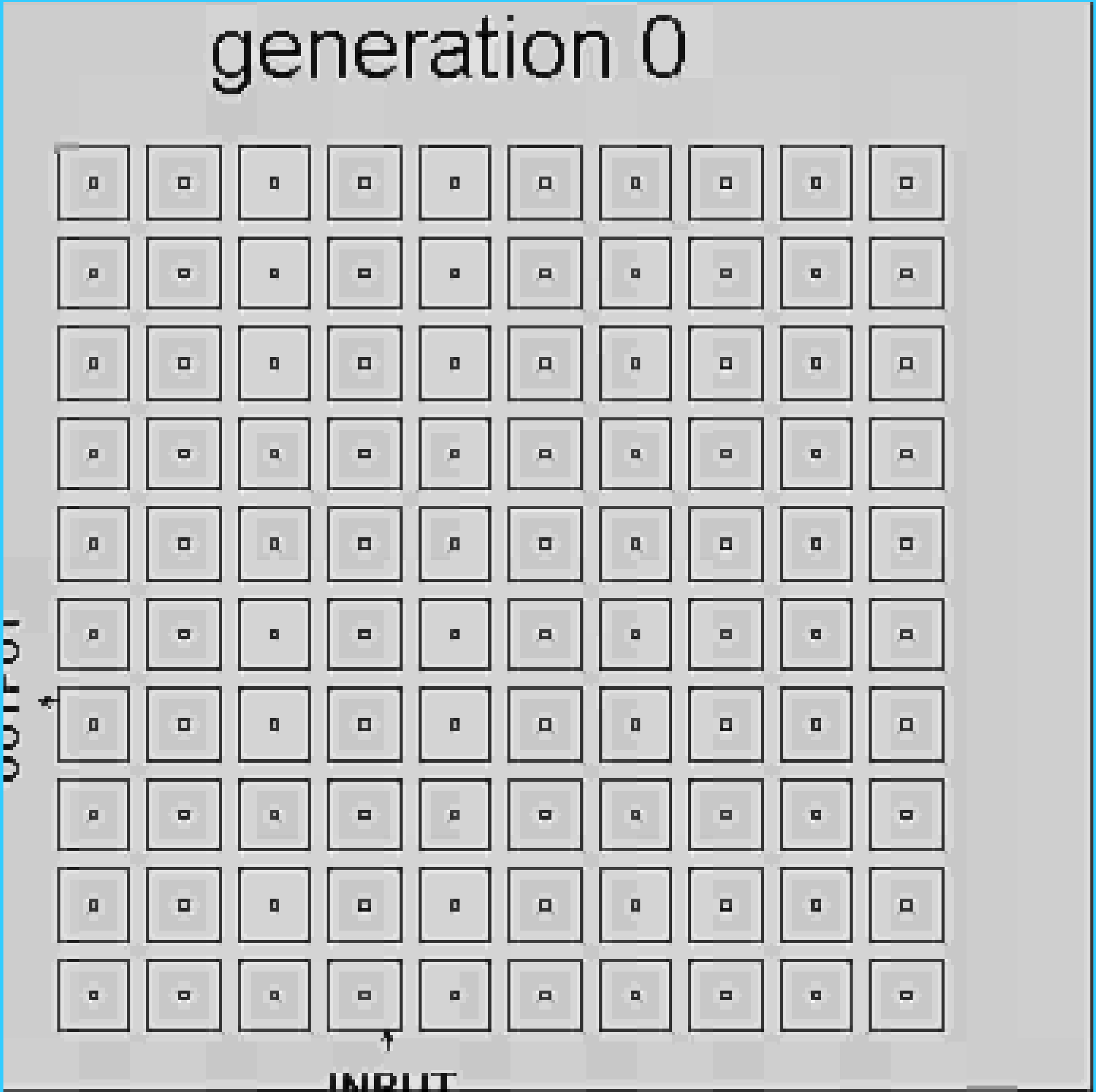


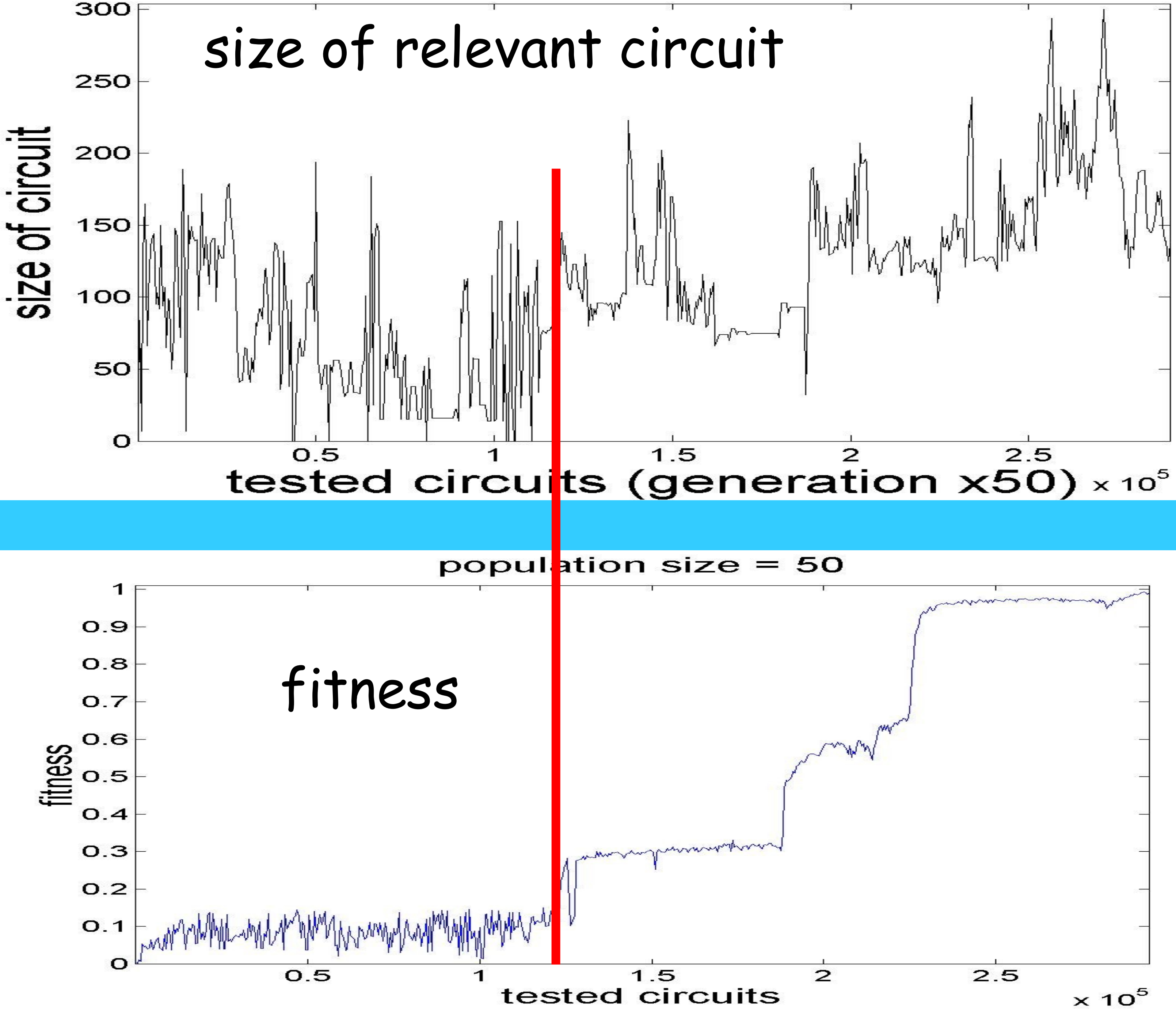
generation 5000



The evolution of the effective circuit

Bursts in the time evolution of the relevant circuit





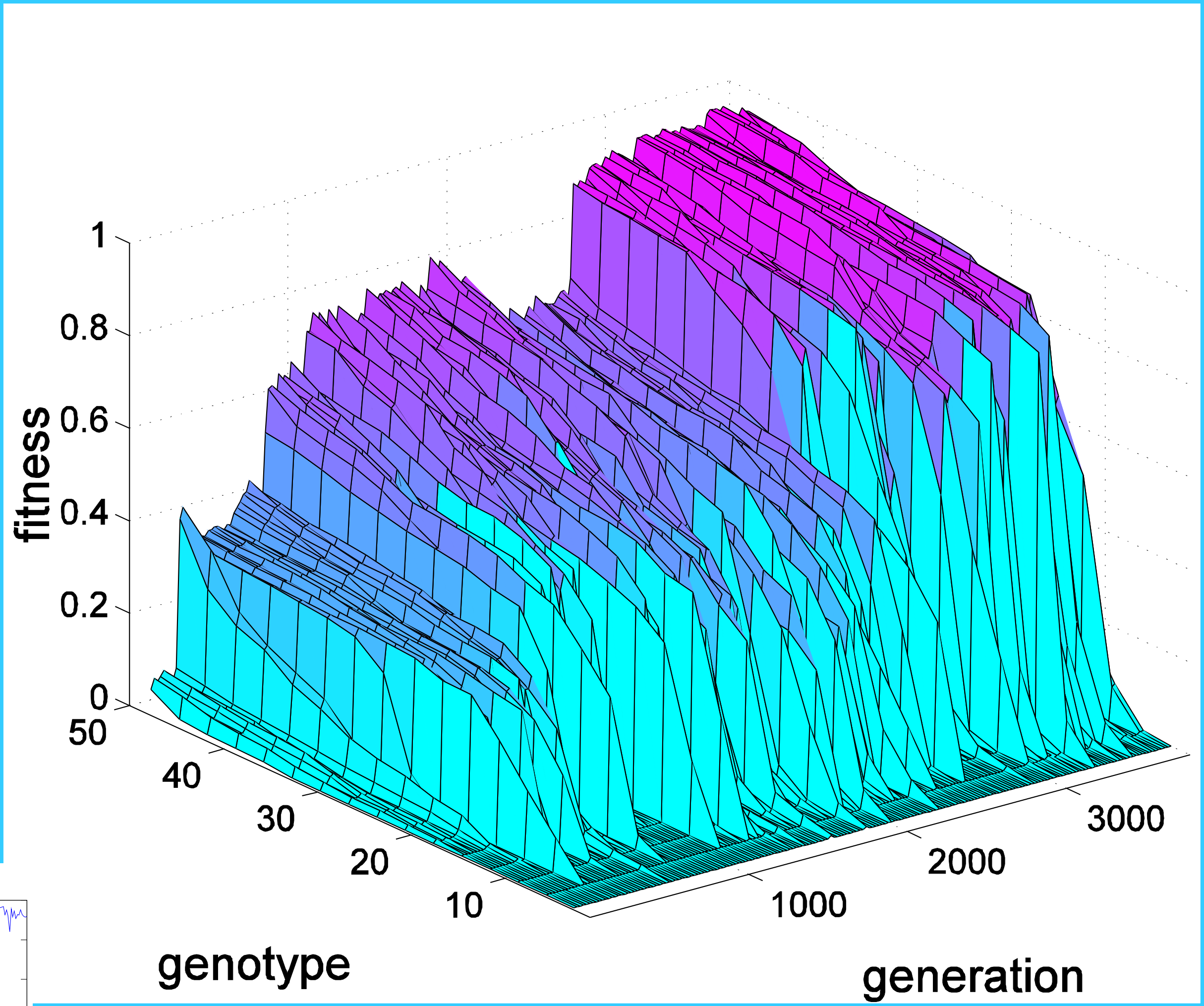
size of relevant circuit

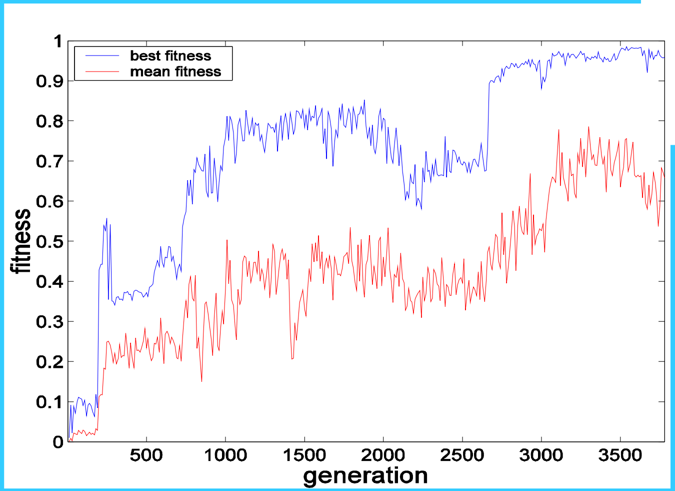
fitness

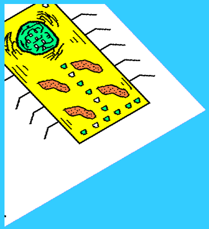
Evolution #1

generations:
3,785

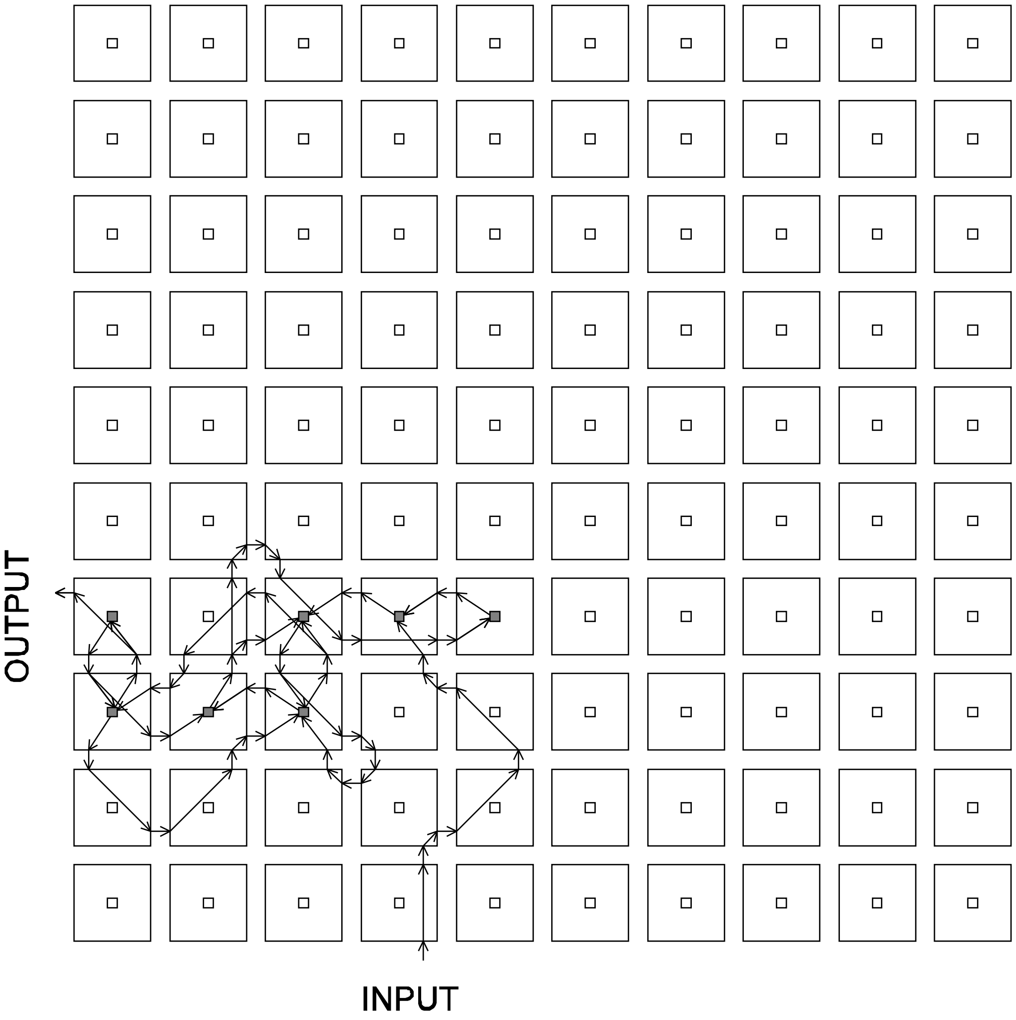
tested circuits:
189,250

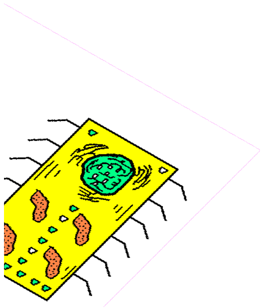




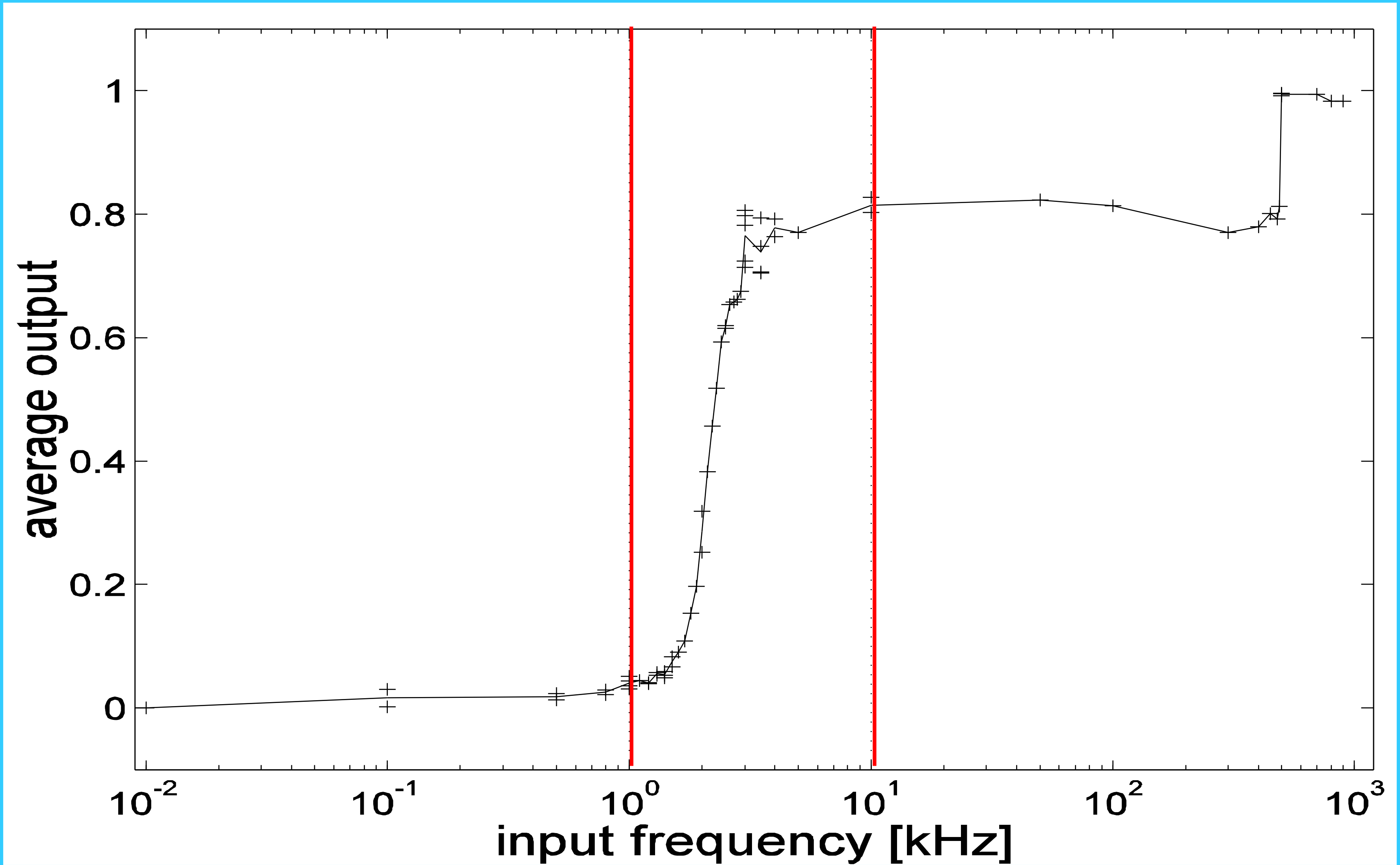


Evolution #1 - Final Circuit Schematics

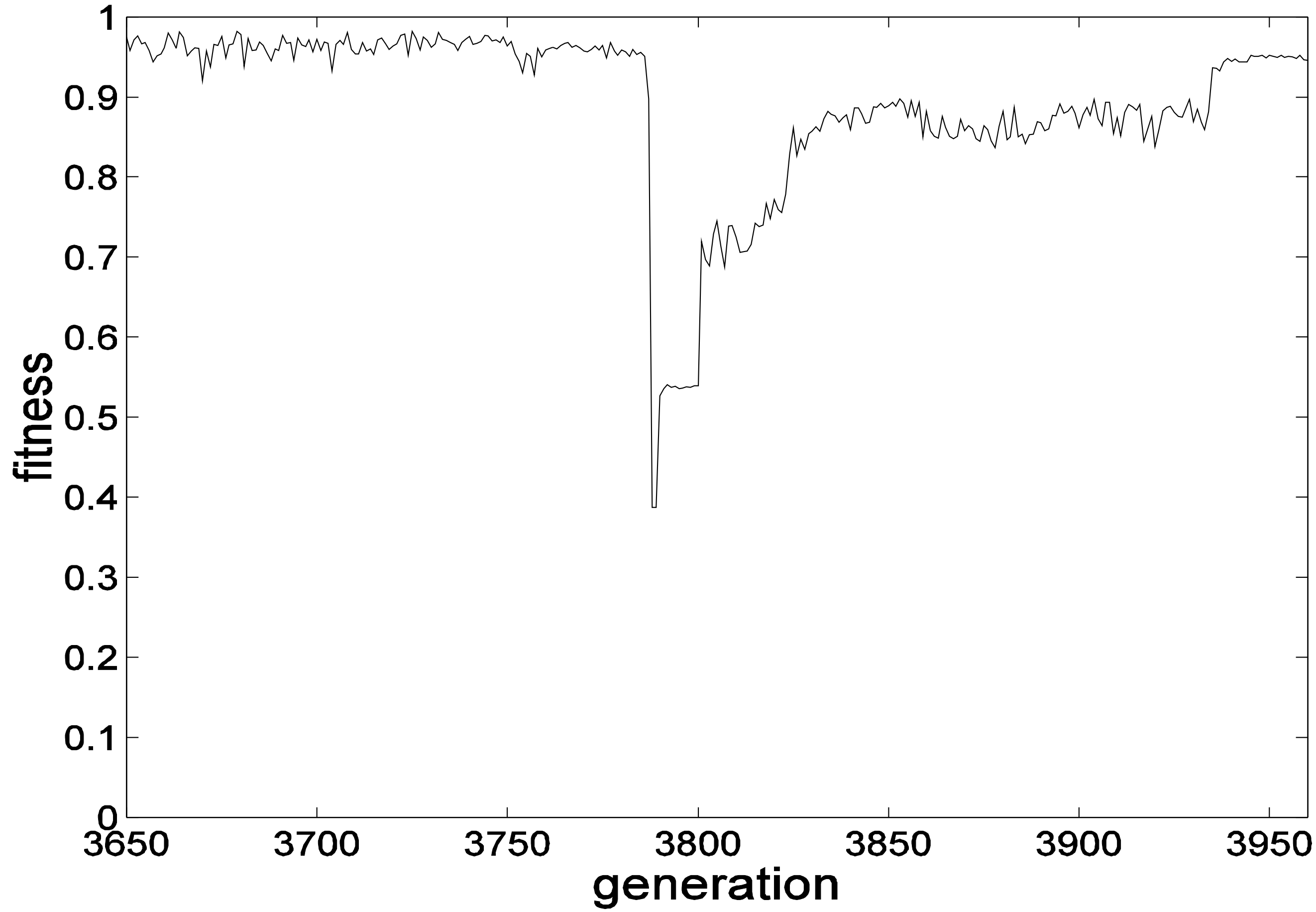




Evolution #1 – Final Circuit Response

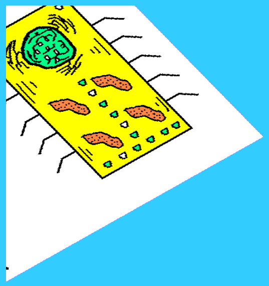


Evolution #1.1
adaptation to a new environment



Evolution #1: encountered problems – cont’d

- Low tolerance to external changes
- Difficulty in circuit reconstruction



Evolution 2:
development of circuit structure

Algorithm modification: longer time delay between circuits, in order to reduce stray effects.

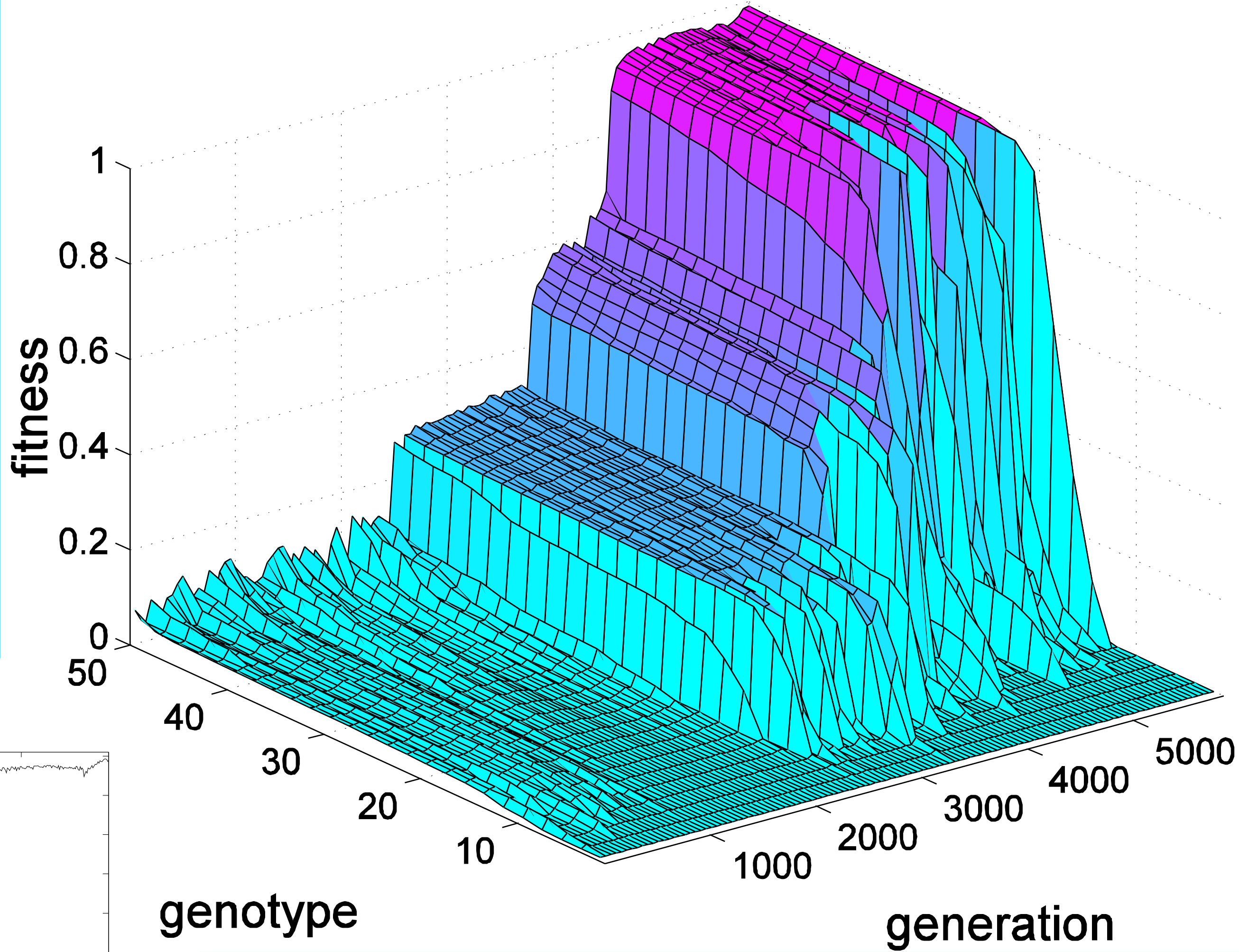
Evolution #2

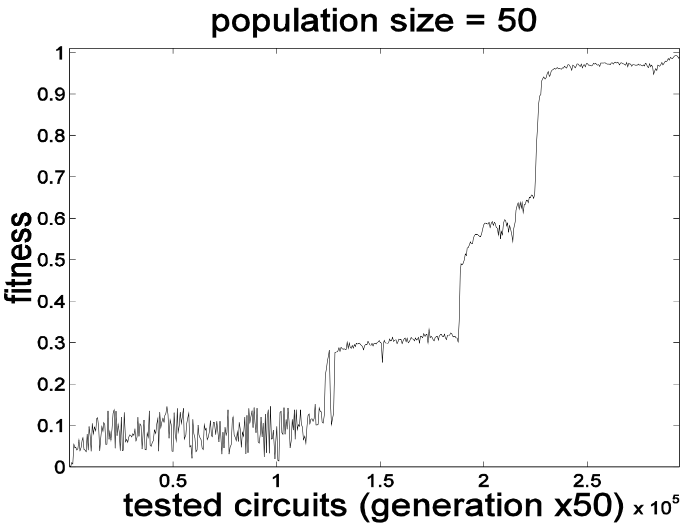
generations:

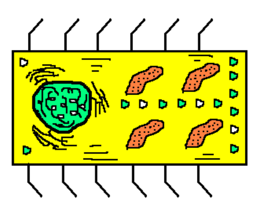
5,800

tested circuits:

290,000



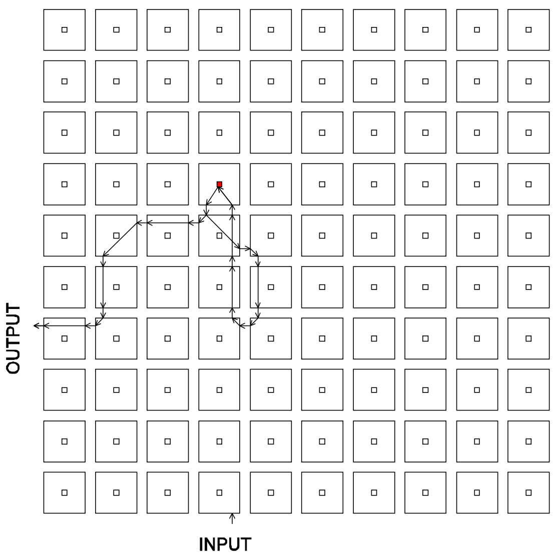




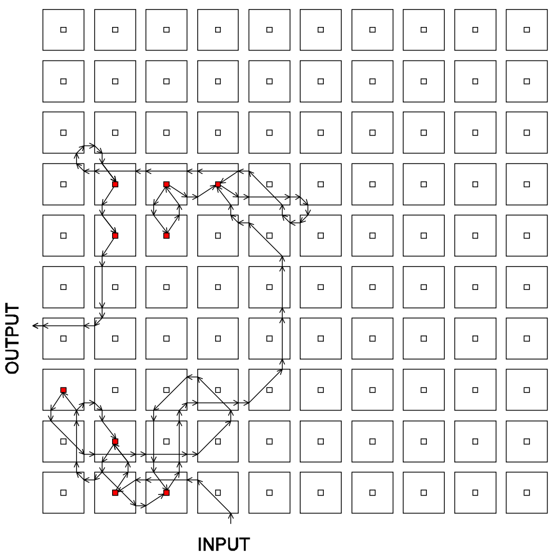
Evolution #2:

Circuitry Development

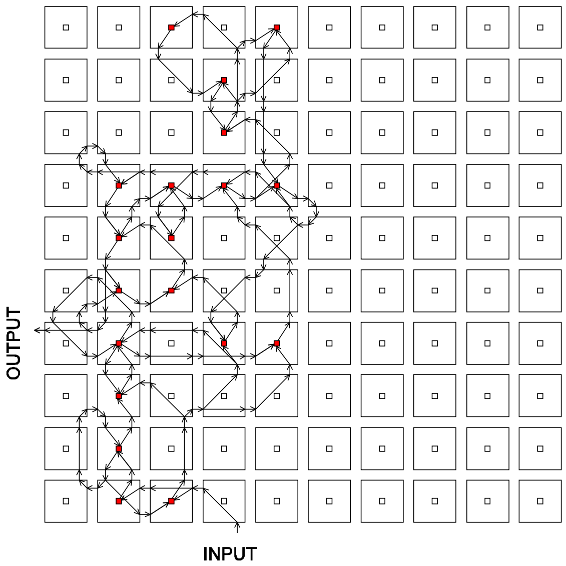
generation 1700
fitness = 0.04



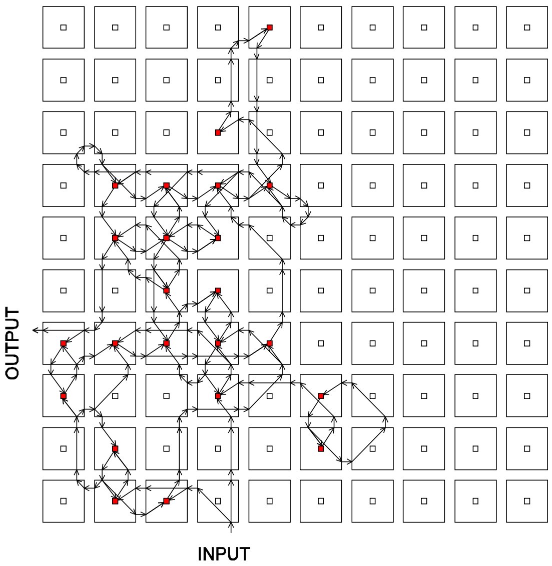
generation 3000
fitness = 0.30



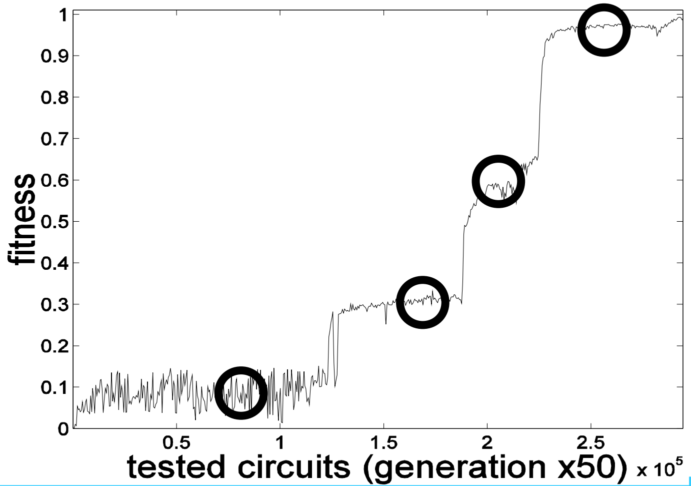
generation 4200
fitness = 0.60



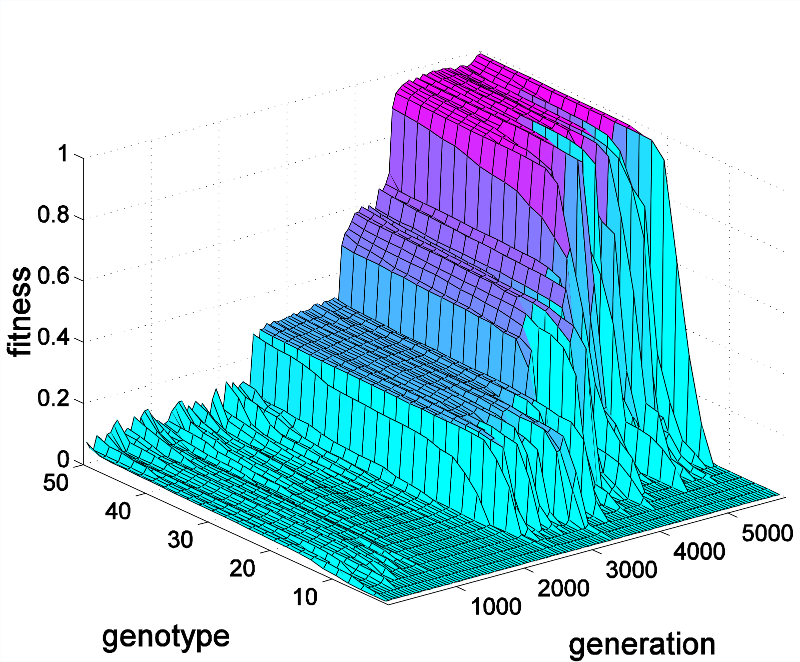
generation 5000
fitness = 0.94

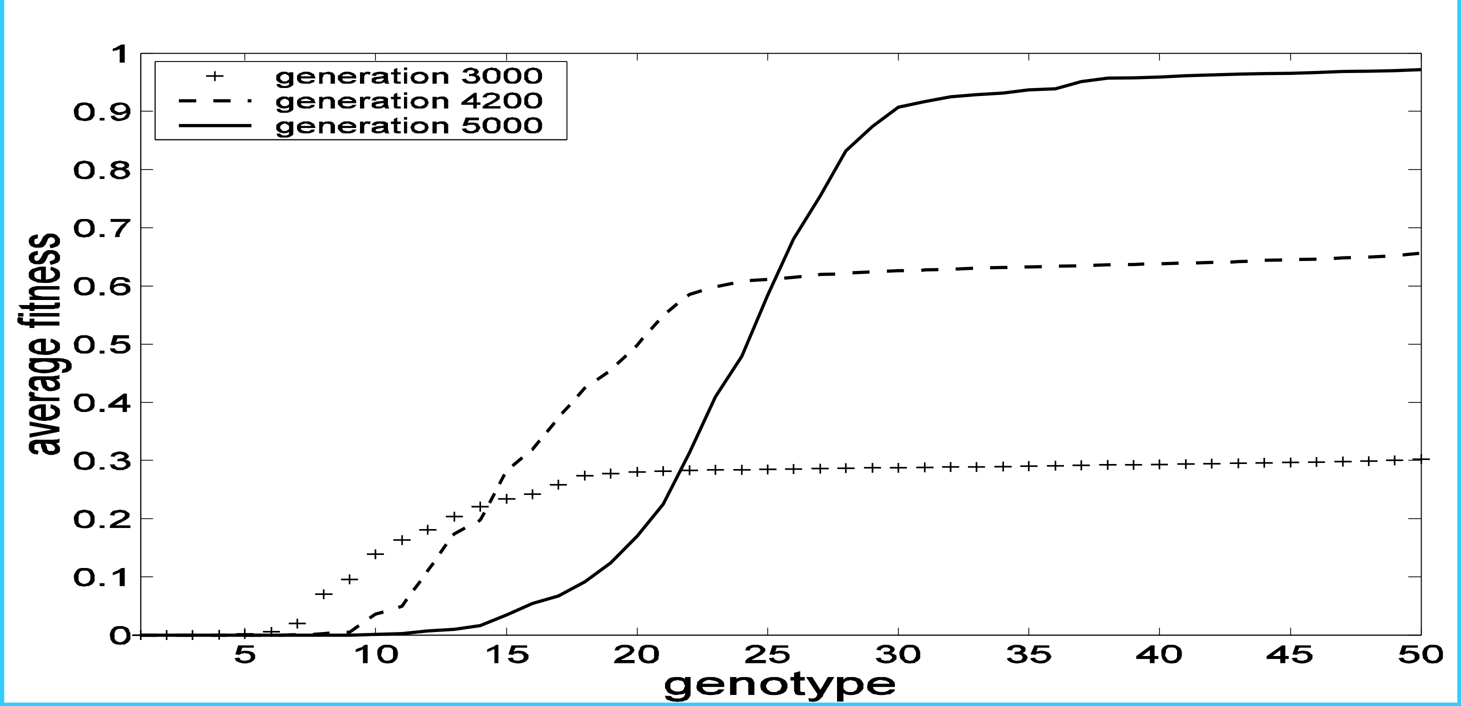


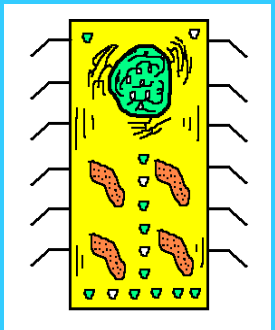
population size = 50



Evolution #2: Fitness Distribution

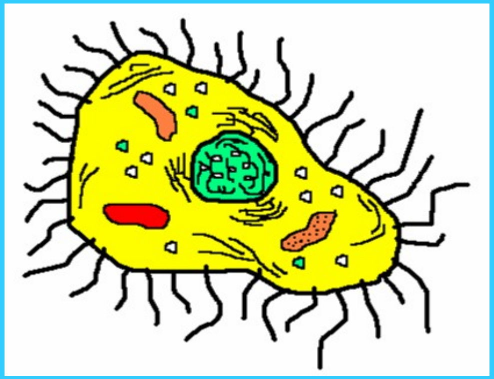






Results

- Both evolution experiments succeeded in finding high scoring circuits. A typical evolution requires testing of about 200,000 circuits.
- Evolution #1:
  - Final circuit is extremely small in size: only 7 logic units.
  - Environmental changes had negative (and positive?) on the fitness of the circuit.
- Evolution #2:
  - Reduction of stray effects leads to a more stable fitness search.
  - The fact that the developed circuit grew in size through the evolution can lead to a better search algorithm.



Looking A HEAD

The hybridized evolvable analog-digital (HEAD) element

Interfacing the HEAD element and real neural networks

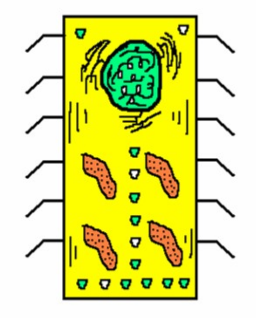
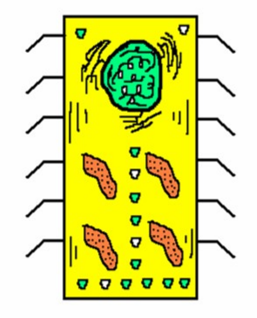
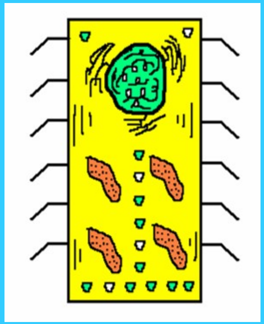
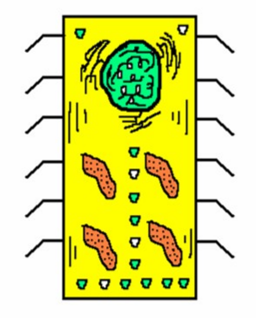
Interfacing the HEAD element with computer models

Ben-Jacob et al., Engineered self-organization in natural
and man made systems (in press)

HEAD element is available upon request

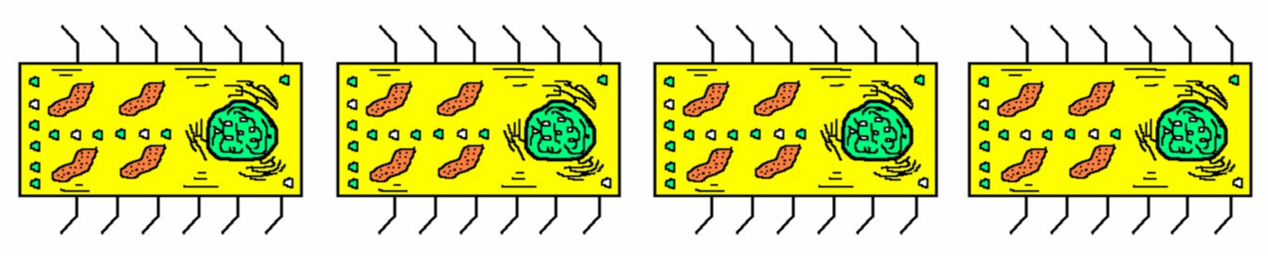
BIG open questions

- How to evolve circuits for more complex tasks
- Search algorithms for circuits on the grand scale (millions of gates)
- Evolution of circuits with combined analog-digital operations

Open Questions

- Effects of algorithm parameters on the optimization search.
- Effects of circuit size and neutral areas.
- Balancing between unconstrained and constrained searches in order to:
  - shorten search time without decreasing the search space;
  - improve circuit stability;
  - keep adaptation ability to environmental changes on an adequate level.



2. Reconfigurable and Morphable Hardware

Part 1Reconfigurable Electronic Hardware

2.1. Reconfigurable hardware (switch-based). Devices, SW Tools, Potential for EHW

2.2. Field Programmable Gate Arrays (FPGA) – Xilinx examples

2.3. Field Programmable Analog Arrays (FPAA) – Anadigm Examples

2.4. Field Programmable Transistor Arrays (FPTA) – JPL examples

Part 2 Other Reconfigurable and Morphable Hardware

2.5. Reconfigurable antennas

2.6. Other reconfigurable structures

2.7. Speed of reconfiguration, partial reconfiguration, context-switching latency issues

2.8. Morphable hardware (no switches). Fine changes and tuning.

2.9. Morphable Materials and devices

2.10. Polymorphic circuits

2.1 Reconfigurable Electronic Hardware

2.1. Reconfigurable hardware (switch-based). Devices, SW Tools, Potential for EHW

2.2. Field Programmable Gate Arrays (FPGA) – Xilinx examples

2.3. Field Programmable Analog Arrays (FPAA) – Anadigm Examples

2.4. Field Programmable Transistor Arrays (FPTA) – JPL examples

2.5. Reconfigurable antennas

2.6. Other reconfigurable structures

2.7. Speed of reconfiguration, partial reconfiguration, context-switching

2.8. Morphable hardware (no switches). Fine changes and tuning.

2.9. Morphable Materials and devices

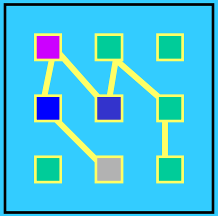
2.10. Polymorphic circuits

Reconfigurable hardware (switch-based). Devices, SW Tools, Potential for EHW

- Function change by configuration change
- Switch-based devices, switches interconnecting functional modules of primitive functions (logical or analogical)
- Programming tools from vendors allow switches to be turned on or off, in a mode visible or invisible to user, via intermediary program conversions.
- Determining the status of the switches – which switches are on and which are off becomes the search/optimization problem for EHW. In many cases only a local search is needed for optimization, to allow for compensation – variations around a configuration determined by knowledge-based/analytical means. Other cases (where for example unidentified faults prevent mapping of computed solutions, a new configurations needs to be searched
- Status of switches – on or off – can be straightforward associated with a binary representation used by genetic algorithms

Elements of Reconfigurable devices

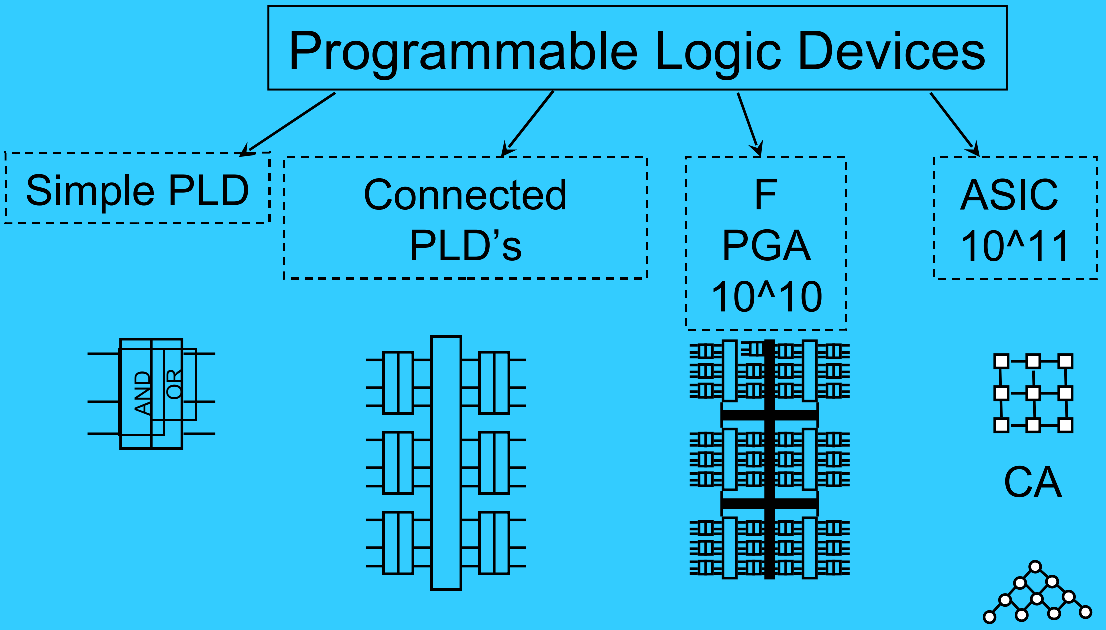
FPGA
FPAA
FPTA
ASIC
Reconfigurable
SOC MSA



- 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

Simple PLD

Connected PLD's

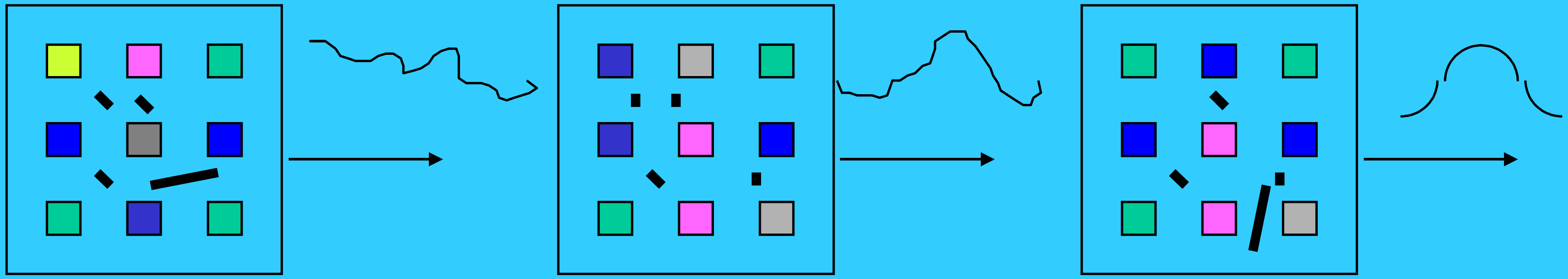
F
PGA
10^10

ASIC
10^11

Low Level Spec

High Level Spec Digital NN

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)

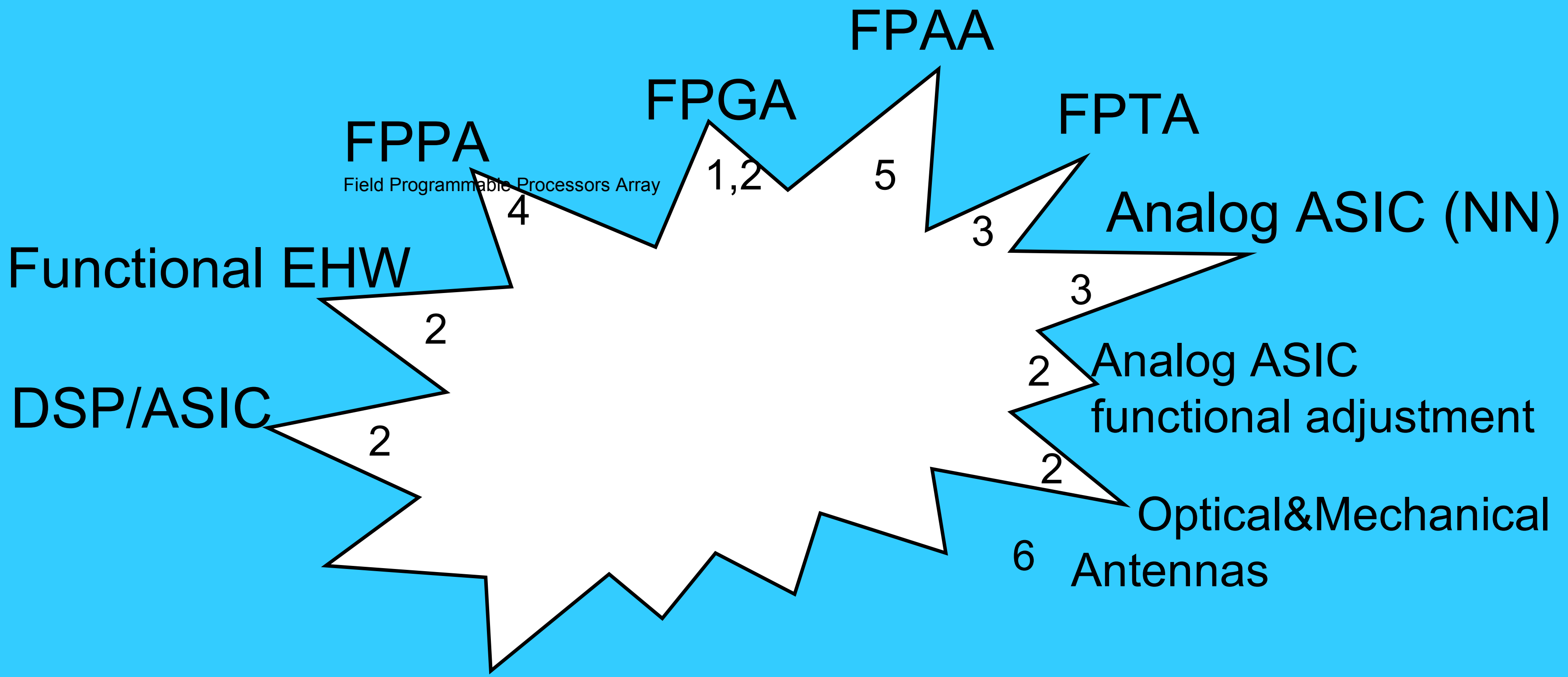
FPGA: AND, OR, NOT FPTA: Cells of Transistor Arrays

On-chip resources of reconfigurable devices

- Fine-grained or course-grained cells
- Local interconnects, global
- Blocks Functional
- Application-specific blocks
- Processors inside
- Power management
- Context memory
- Analog or digital

HW Platforms for EHW Experiments

First/ significant experiments on:...



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

4 Marchal, CSEM, Switzerland
5 Zebulum, U. Sussex, UK (now at JPL)
6 Linden

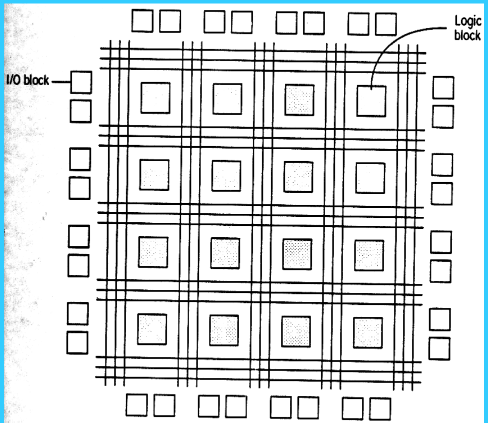
COTS digital reconfigurable hardware

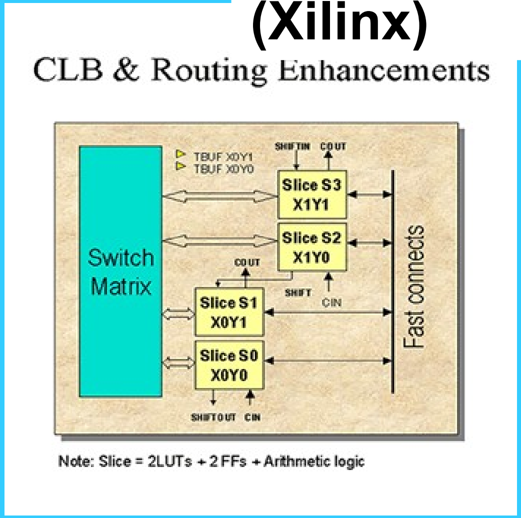
PLA

Xilinx 6200

FPGA

Virtex, VirtexII Pro (Xilinx)





Note: Slice = 2LUTs + 2 FFs + Arithmetic logic

Altera, Actel, Other companies,, etc…

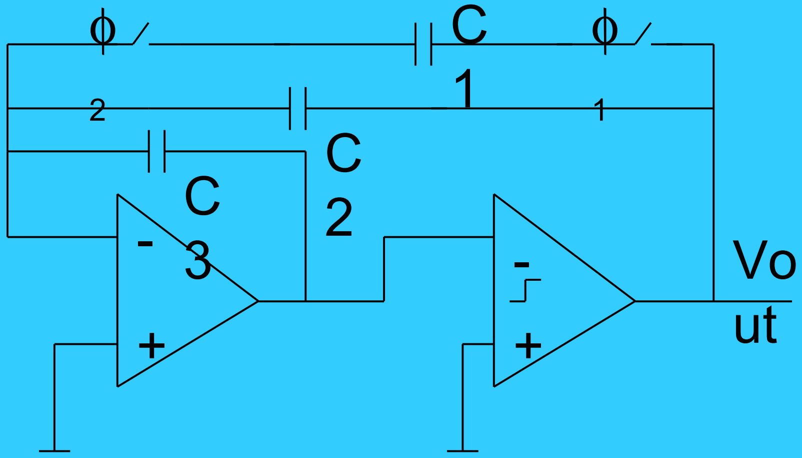
Programmable SOC

COTS analog reconfigurable hardware

FPAAs

Pilkington
Motorola MPAA020

- Switched capacitors

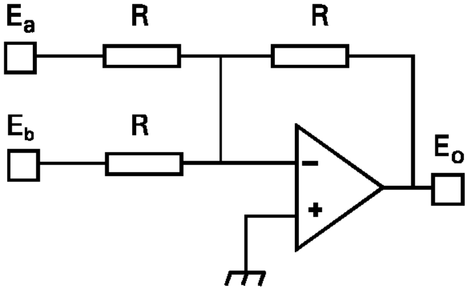


Now Anadigm

Lattice

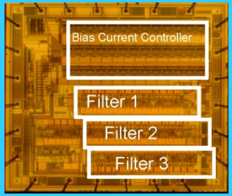
Zetex TRAC

- 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 compor



Custom Made EHW-oriented reconfigurable hardware

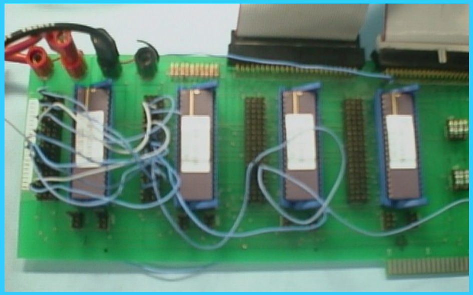
Japan Higuchi EHW-chips



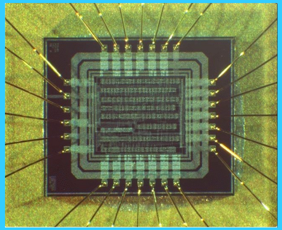
Industrial, specific

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 with On-board Controller

The PSoC™ CY8C25122/CY8C26233/CY8C26443/CY8C26643 family of programmable system-on-chip devices replace multiple MCU-based system 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)

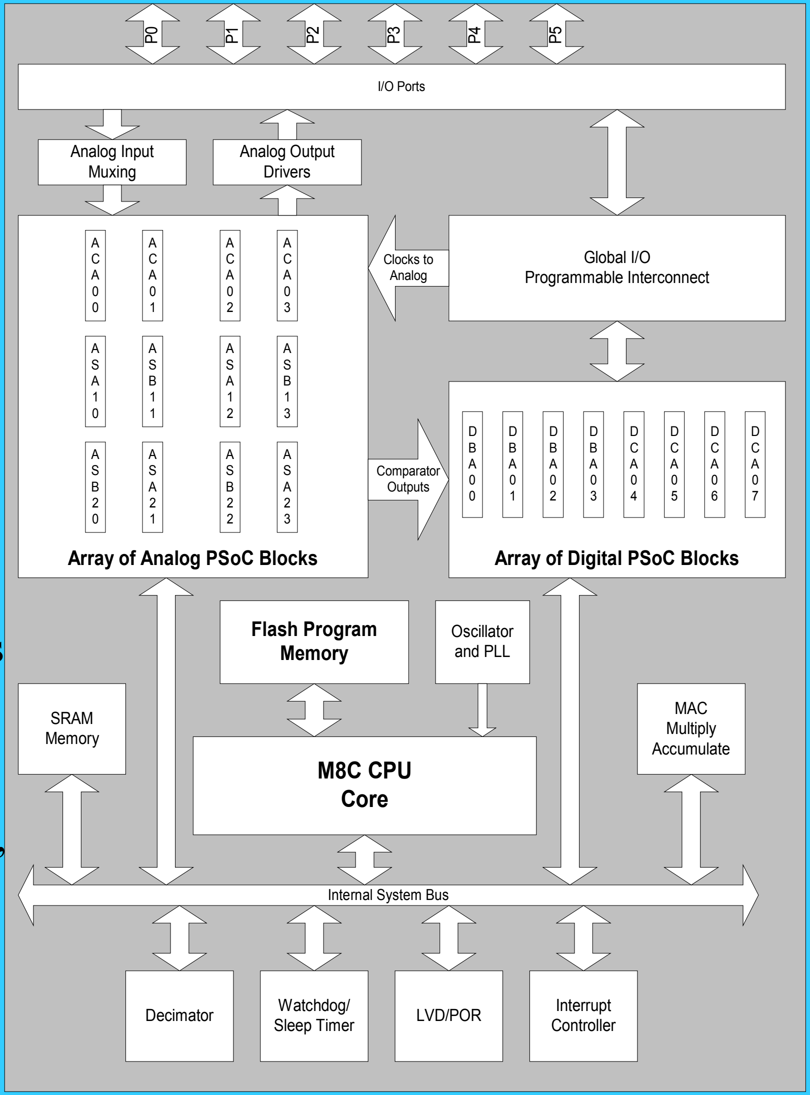
- PSoC Single-Phase Power Meter Reference Design
- Modem - 300 Baud
- Magnetic Card Reader Reference Design

Cypress
CY8C26443
Final Datasheet
May. 29, 2003

PSoC architecture and building blocks

All devices in this family include both analog and digital configurable peripherals (PSoC blocks). These blocks enable the user to define unique functions during configuration of the device.

Included are twelve analog PSoC blocks and eight digital PSoC blocks Potential applications for the digital PSoC blocks are timers,counters, UARTs, CRC generators, PWMs, and other functions.



The analog PSoC blocks can be used for SAR ADCs, multi-slope ADCs, programmable gain amplifiers, pro-grammable filters, DACs, and other functions.

Higher order User Modules such as modems, complex motor controllers, and complete sensor signal chains can be created from these building blocks. This allows for an unprecedented level of flexibility and integration in micro-controller-based systems.

http://www.cypress.com/cfuploads/img/products/CY8C26443-24PI.pdf

More details on PSoC blocks and features

Programmable System-on-Chip (PSoC) Blocks

- On-chip, user configurable analog and digital peripheral blocks
- PSoC blocks can be used individually or in combination
- 12 Analog PSoC blocks provide:
  - Up to 11 bit Delta-Sigma ADC
  - Up to 8 bit Successive Approximation ADC
  - Up to 12 bit Incremental ADC
  - Up to 9 bit DAC
  - Programmable gain amplifier
  - Programmable filters
  - Differential comparators
- 8 Digital PSoC blocks provide:
  - Multipurpose timers: event timing, real-time clock, pulse width modulation (PWM) and PWM with deadband
  - CRC modules
  - Full-duplex UARTs
  - SPI . master or slave configuration
  - Flexible clocking sources for analog PSoC blocks

Powerful Harvard Architecture Processor

with Fast Multiply/

•M8C processor instruction set

•Processor speeds to 24 MHz

•Register speed memory transfers

•Flexible addressing modes

•Bit manipulation on I/O and memory

•8x8 multiply, 32-bit accumulate

•Flexible On-Chip Memory

•on device

•50,000 erase/write cycles

•256 bytes SRAM data storage

•In-System Serial Programming (ISSP .)

•http://www.cypress.com/cfuploads/img/products/CY8C26443-24PI.pdf

2. Reconfigurable and Morphable Hardware

2.1. Reconfigurable hardware (switch-based). Devices, SW Tools, Potential for EHW

2.2. Field Programmable Gate Arrays (FPGA) – Xilinx examples

2.3. Field Programmable Analog Arrays (FPAA) – Anadigm Examples

2.4. Field Programmable Transistor Arrays (FPTA) – JPL examples

2.5. Reconfigurable antennas

2.6. Other reconfigurable structures

2.7. Speed of reconfiguration, partial reconfiguration, context-switching

2.8. Morphable hardware (no switches). Fine changes and tuning.

2.9. Morphable Materials and devices

2.10. Polymorphic circuits

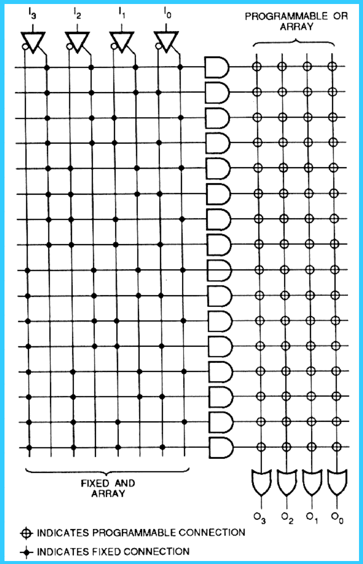
Prewired Arrays

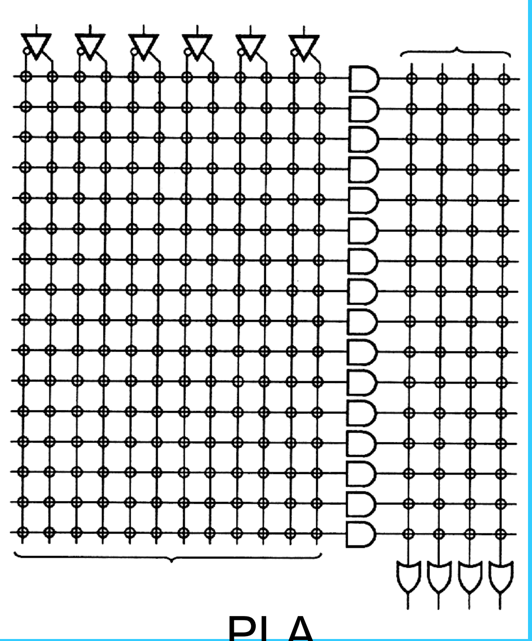
Categories of prewired arrays (or field-programmable devices):

- Fuse-based (program-once)
- Non-volatile EPROM based
- RAM based

Programmable Logic Devices

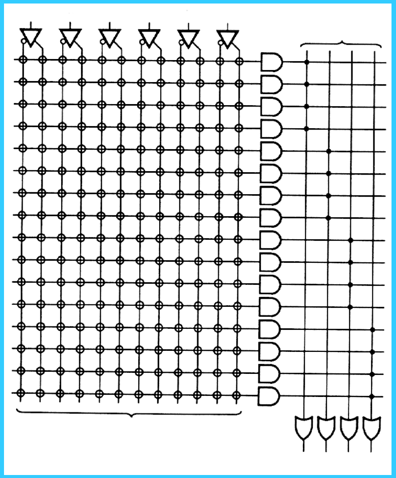
PROM

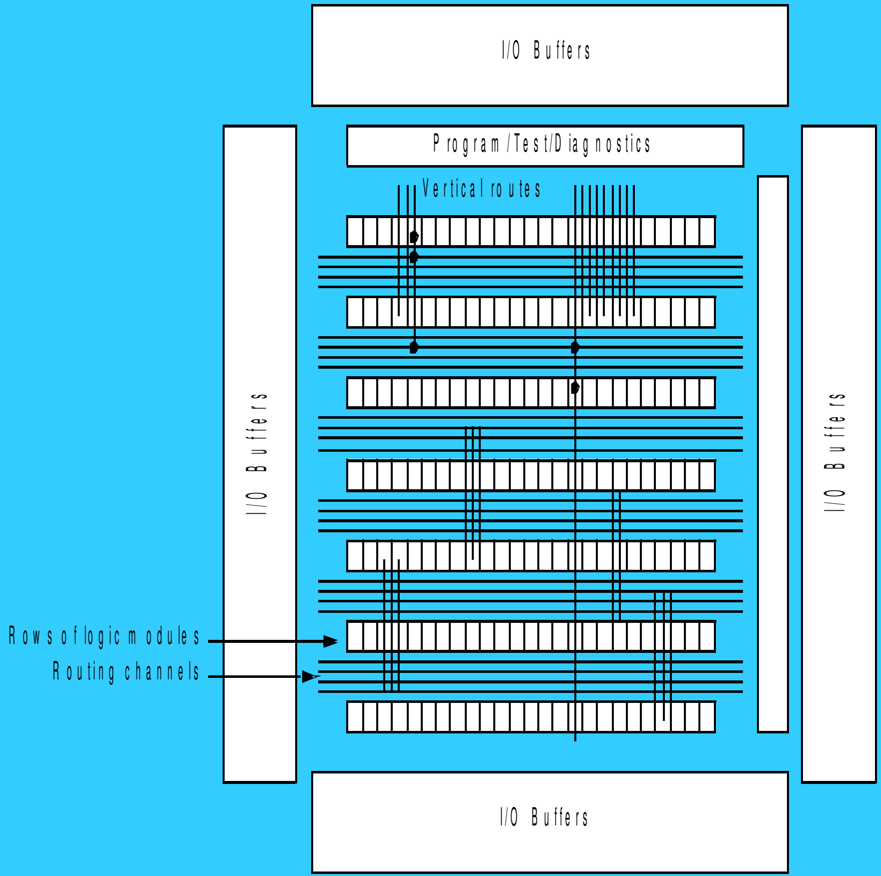




PLA

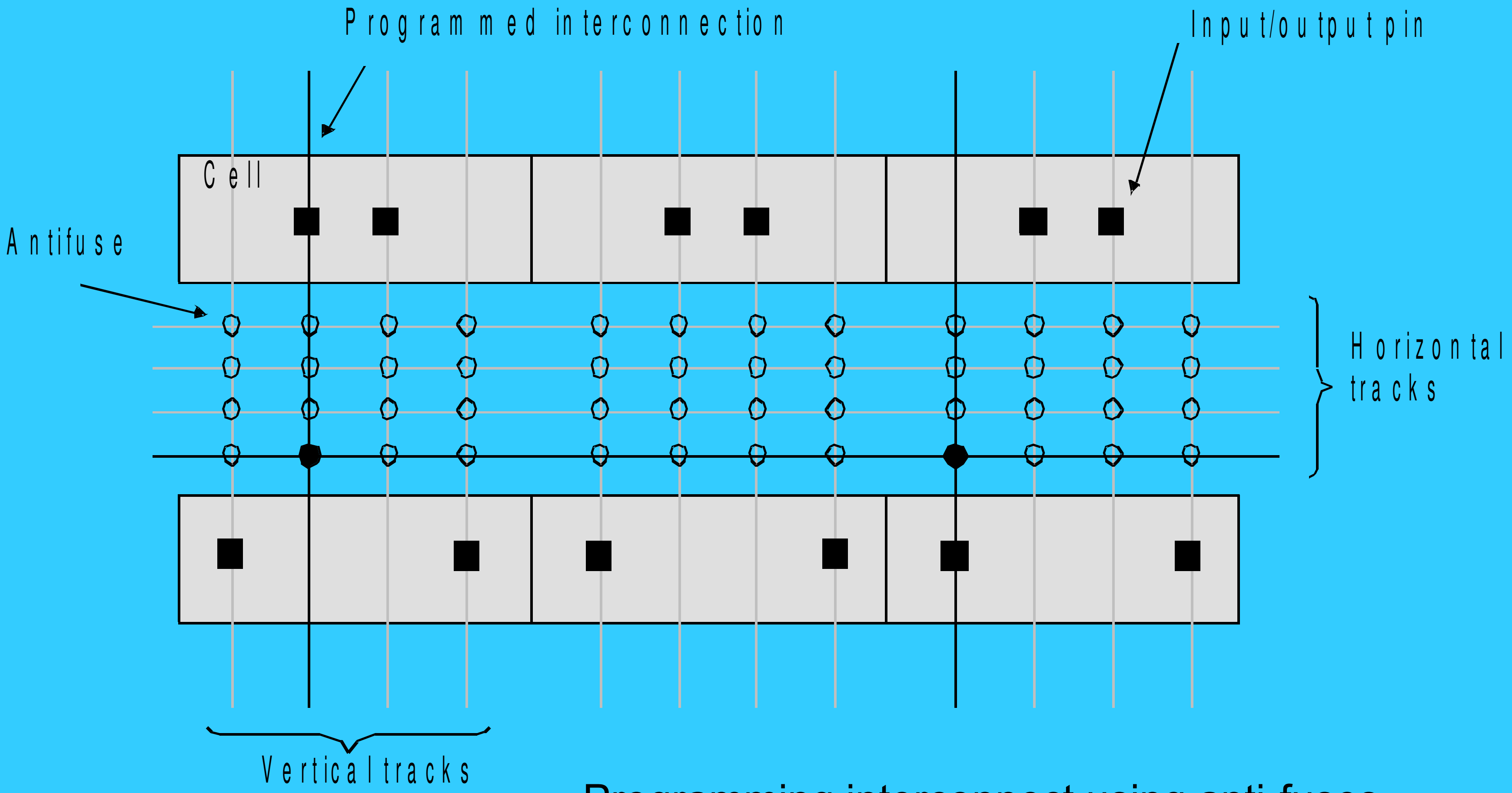
PAL





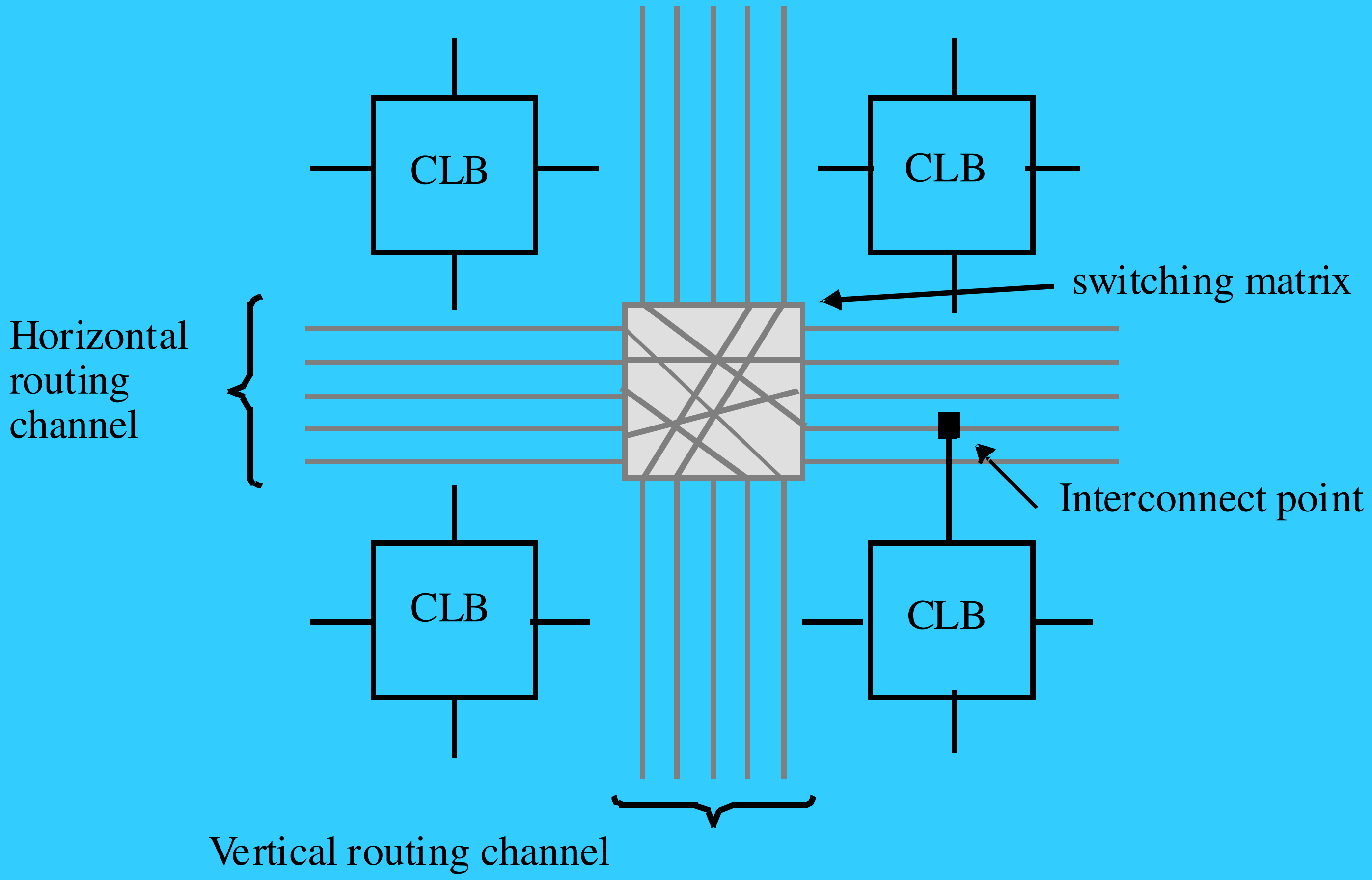
Standard-cell like
Floorplan in fuse-based FPGA

Interconnect

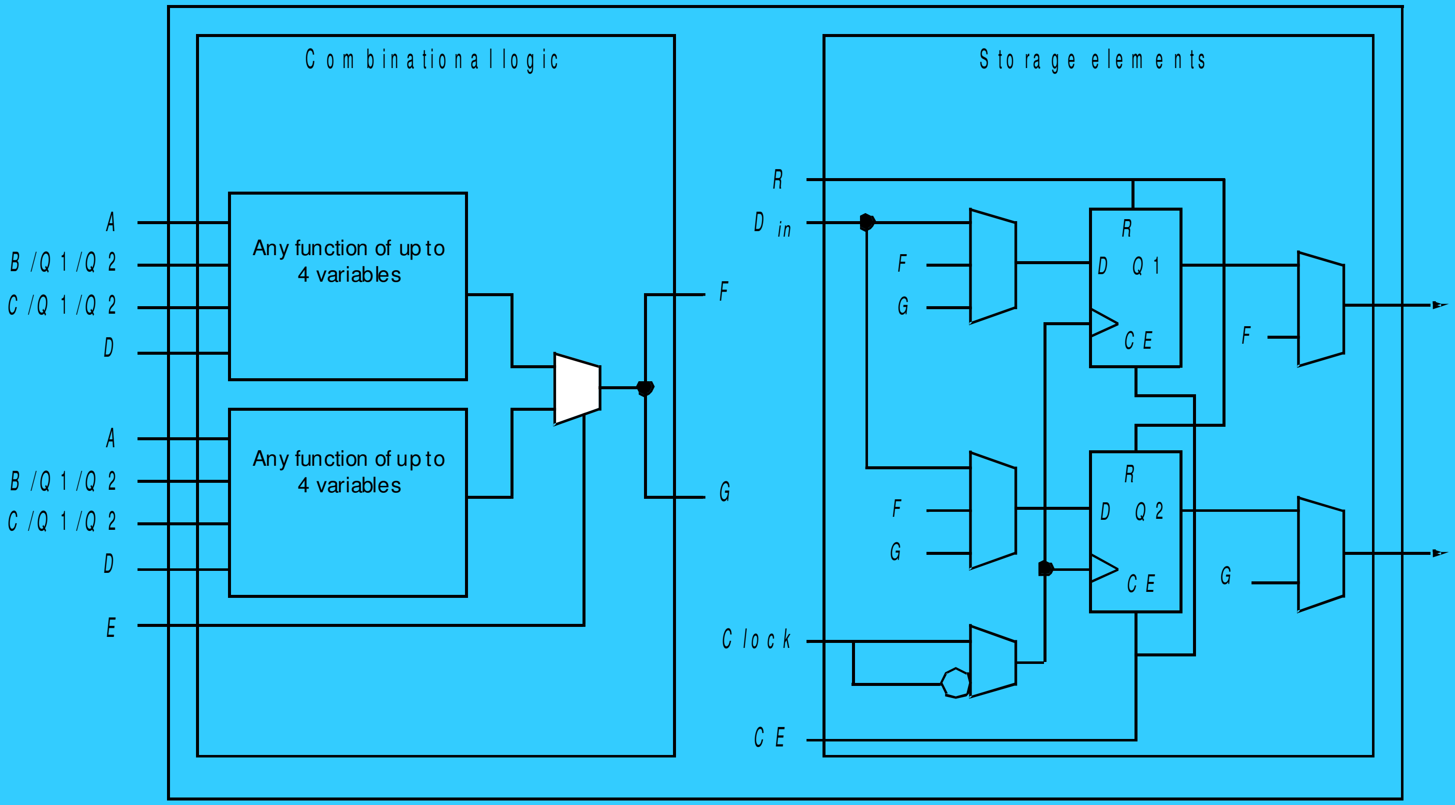


Programming interconnect using anti-fuses

Field-Programmable Gate Arrays
RAM-based



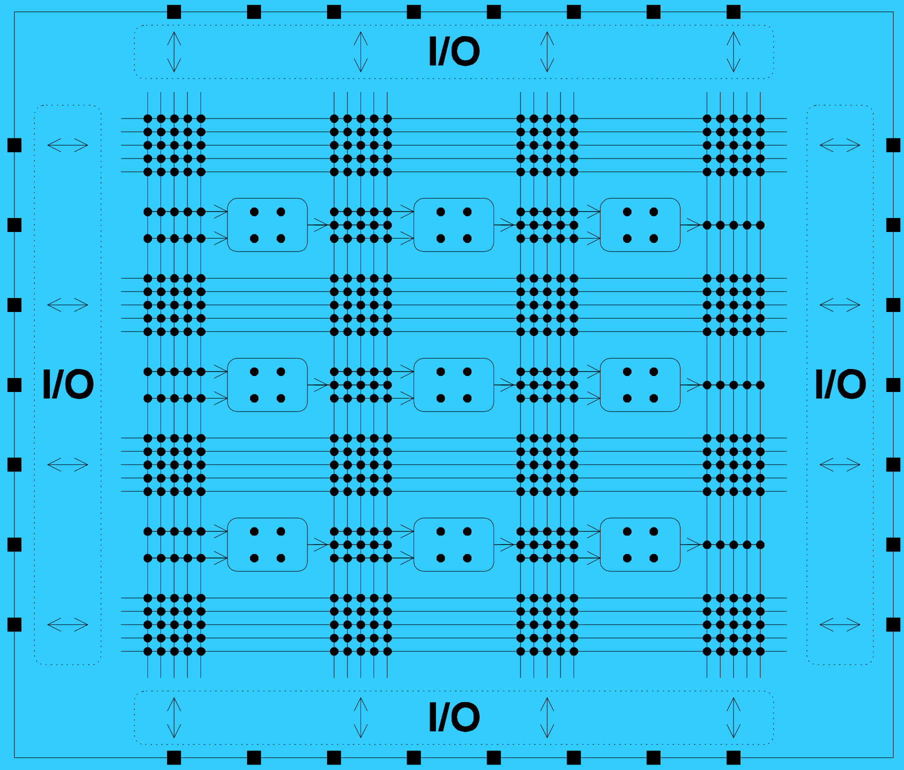
RAM-based FPGA Basic Cell (CLB)



Courtesy of Xilinx

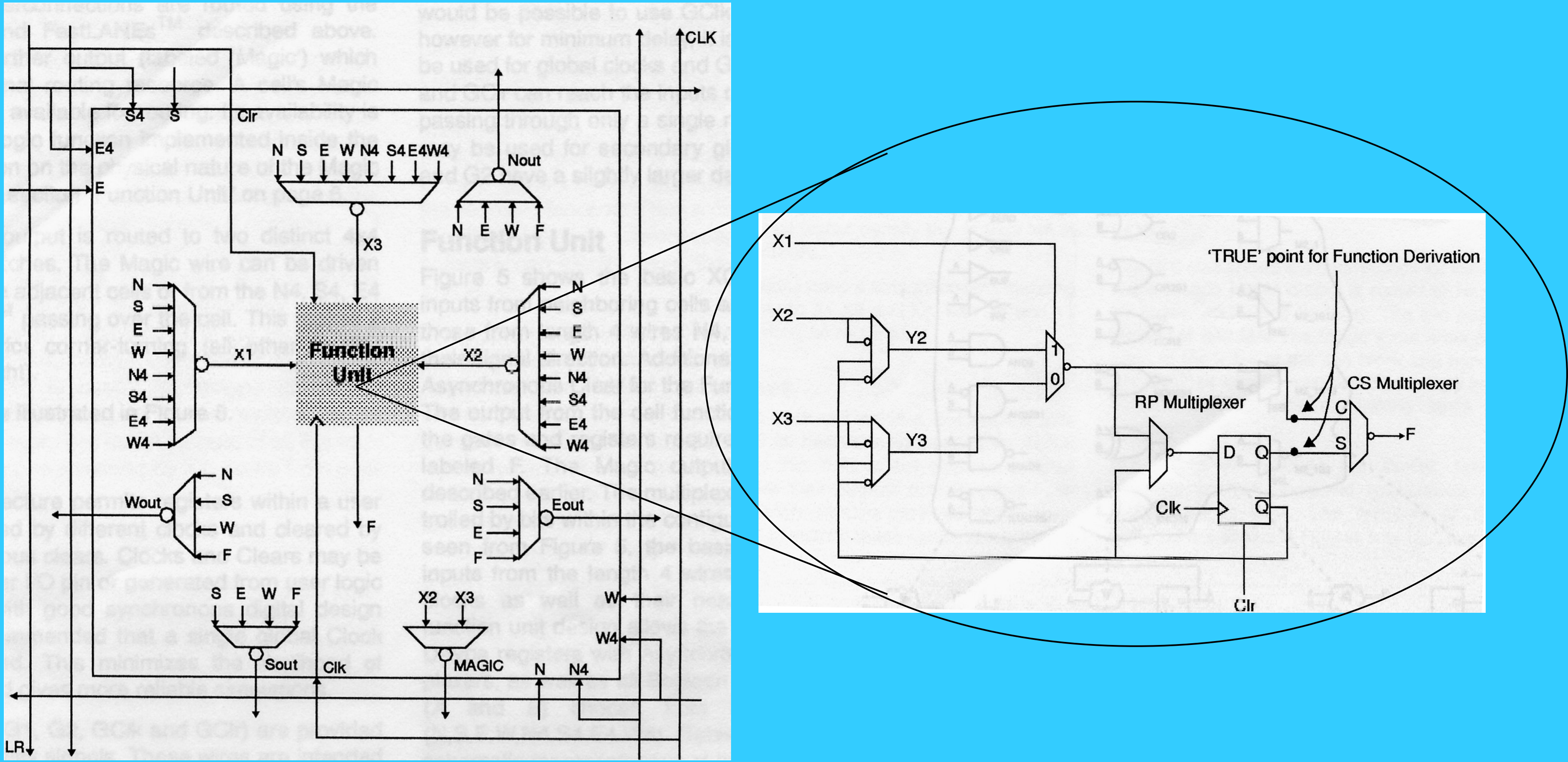
6200 Architecture and Thompson’s experiments

- 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

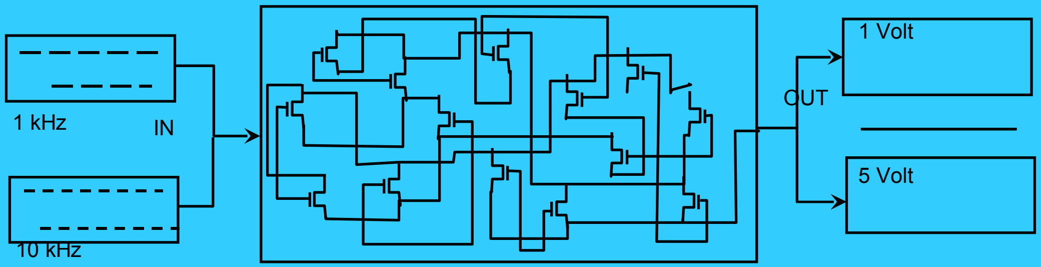


SRAM-controlled switch (mux)-based FPGA

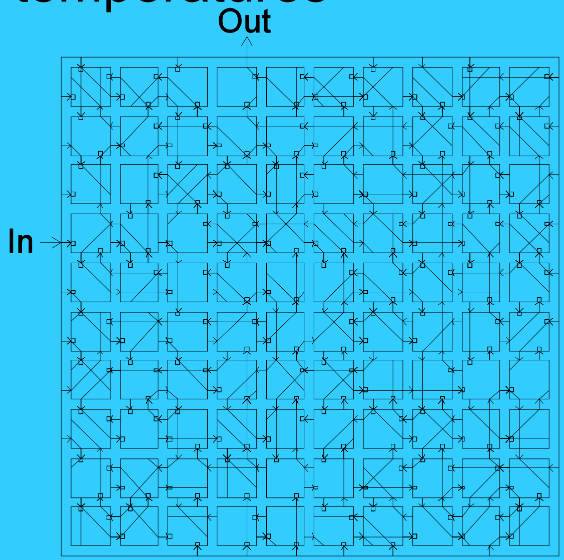
Thompson's experiment

- 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:

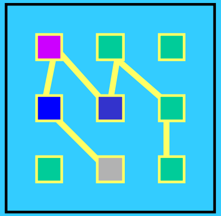
Use of FPGAs from different foundries, at different temperatures



Tone-Discriminator for 1 kHz and 10 kHz using Transistors



1kHz - 10KHz



0 1

Virtex 2 Pro

- Cells
- On-chip processors (Power-PC) offer potential for on-chip ES
- HW/SW co-design

Virtex-II slice

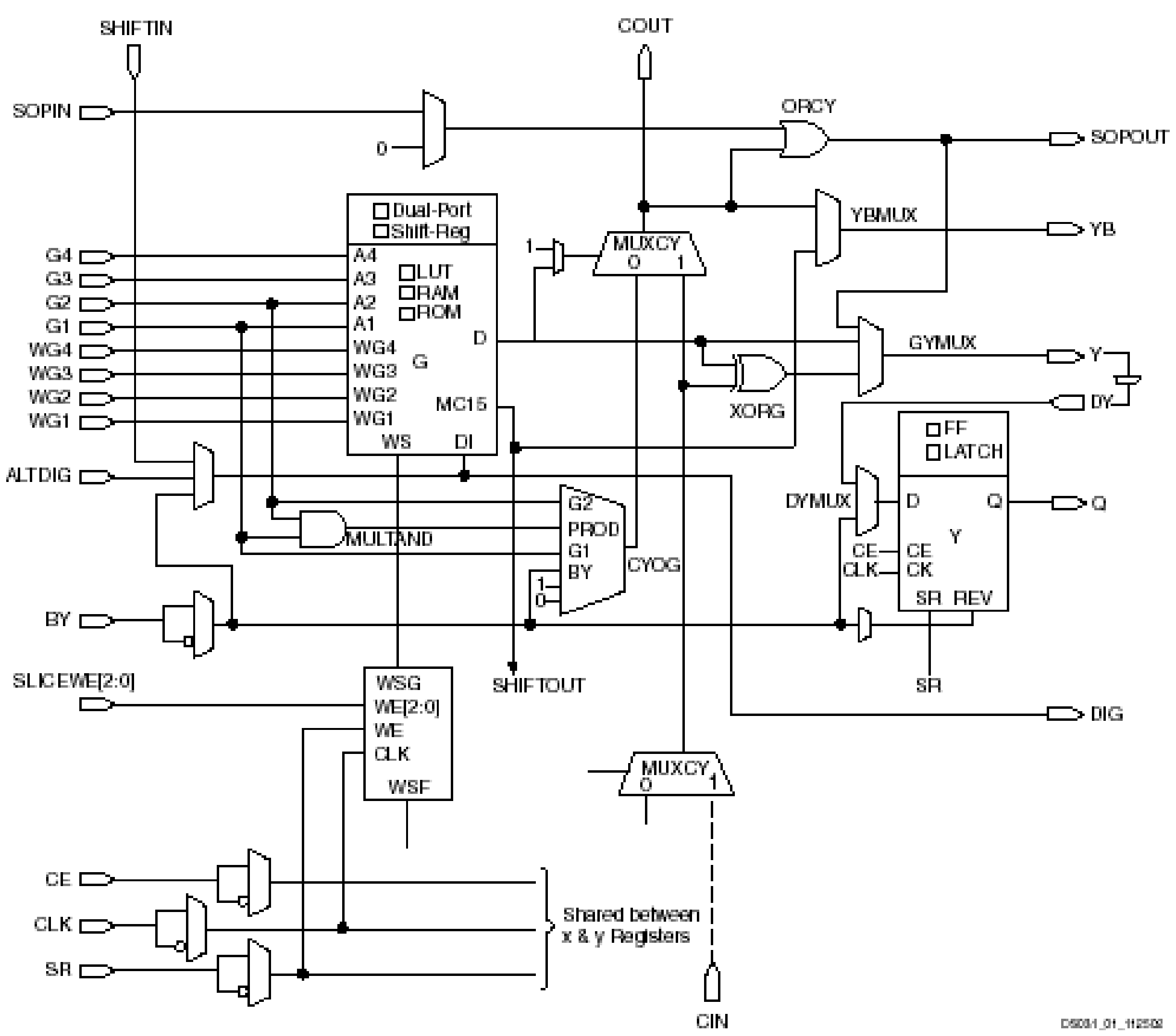


Figure 16: Virtex-II Slice (Top Half)

A Combined LUT/MUX FPGA Cell

2. Reconfigurable and Morphable Hardware

2.1. Reconfigurable hardware (switch-based). Devices, SW Tools, Potential for EHW

2.2. Field Programmable Gate Arrays (FPGA) – Xilinx examples

2.3. Field Programmable Analog Arrays (FPAA) – Anadigm Examples

2.4. Field Programmable Transistor Arrays (FPTA) – JPL examples

2.5. Reconfigurable antennas

2.6. Other reconfigurable structures

2.7. Speed of reconfiguration, partial reconfiguration, context-switching

2.8. Morphable hardware (no switches). Fine changes and tuning.

2.9. Morphable Materials and devices

2.10. Polymorphic circuits

Motorola Field Programmable Analog Array

- Coarse-grained analog programmable array;

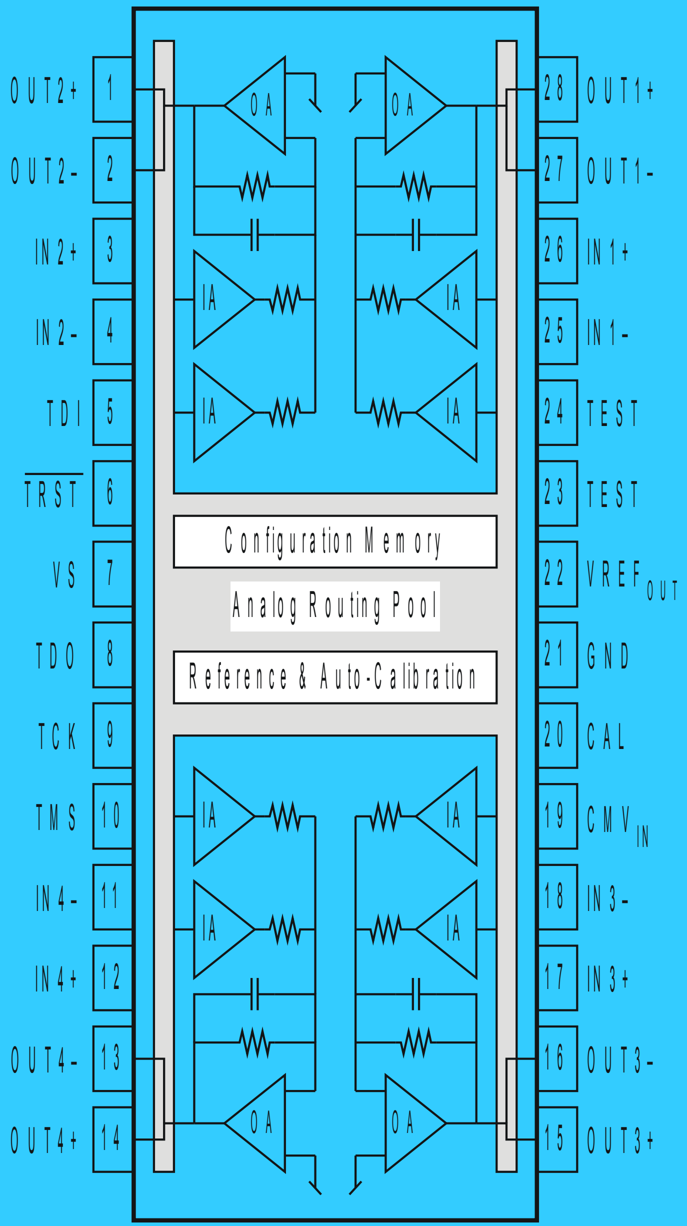
- Each cell consists of an operational amplifiers, internally connected through switching capacitors;

- Each cell programmed by ~300 bits, which determine capacitance values and switching;

- Total of 20 cells;

- Evolutionary experiments to synthesize low-pass filters, oscillators, half-wave rectifiers, gain circuits and adders.

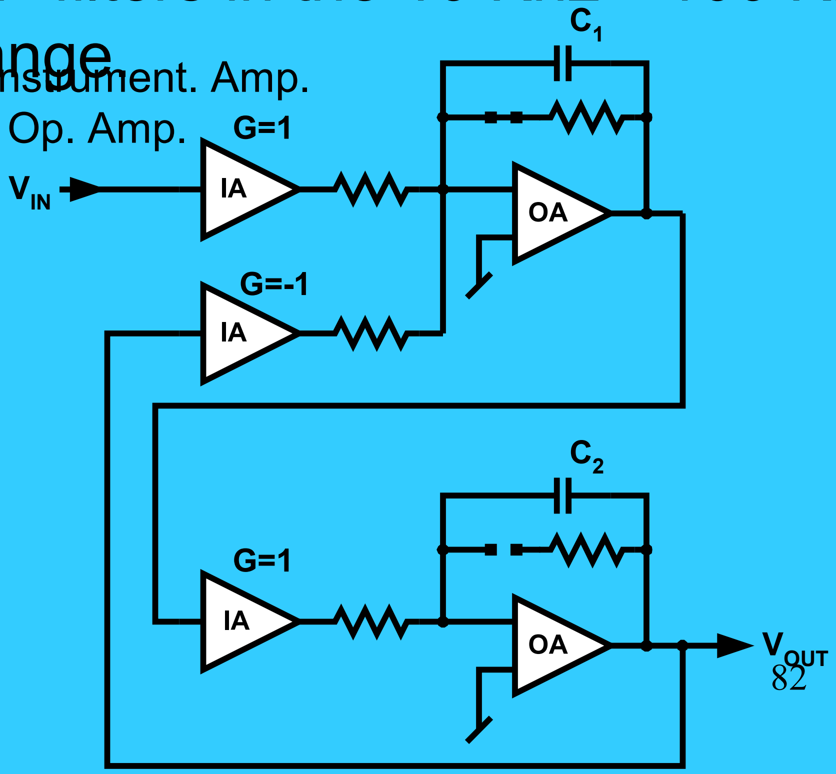
Lattice ispPAC10



- 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.



Characteristics of AN220E04 FPAA from Anadigm

- The AN220E04 chip consists of a 2 x 2 matrix of fully configurable, switched capacitor configurable analog blocks (CABs), enmeshed in a fabric of programmable interconnect resources;
- Flexible platform for signal conditioning, allowing the configuration of many important high level building blocks called CAM (configurable analog modules) such as oscillators, adders, and amplifiers
- Design based on previous FPAA chip fabricated by Motorola in 1998, the MPAA020 with 20 cells and 6864 programming bits;
- Improvements from earlier versions:
  - Fully differential architecture;
  - Dynamic Reconfiguration (Two memories, shadow and configurations SRAM);
  - Higher bandwidth: 2MHz;
- However there are less cells compared to the Motorola chip;

Anadigm Vortex

- Switched capacitors
  - Can act like big resistors
  - Good thermal stability
  - Easy to match manufacturing variations in ratios
  - Can do things like “negative resistance” & rectifiers without diode drops

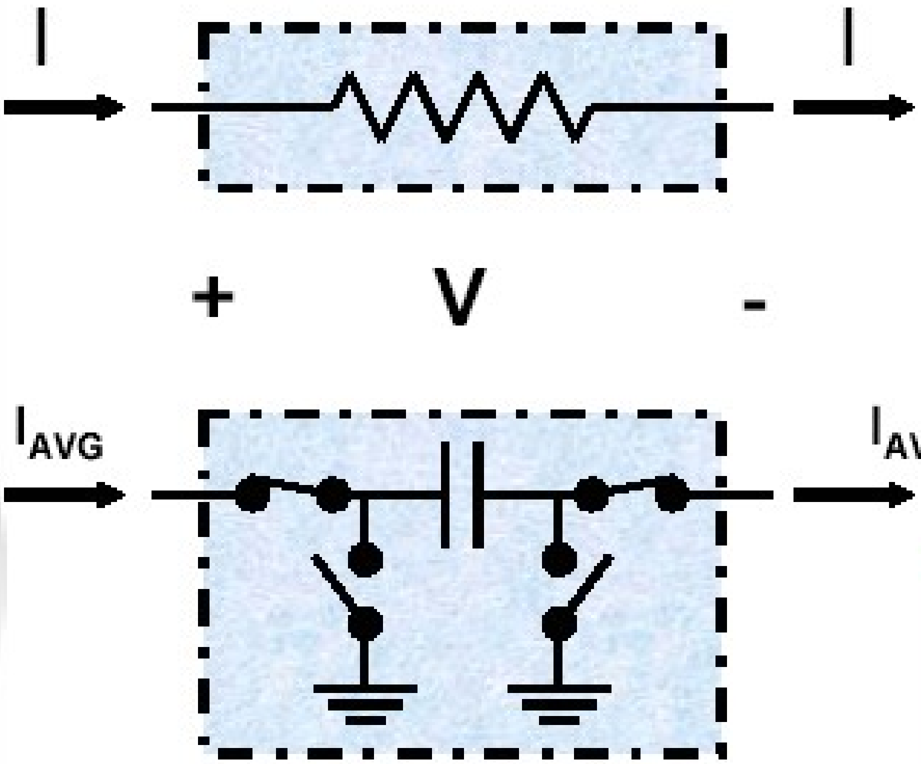
- Chip
  - Sampled data, analog value
  - Can be reconfigured quickly & flexibly by host µP & itself
  - Good software support; aims to bring FPGA advantages to analog
  - 20 cells, configurable IO

Internal reprogrammable routing impedances and switched capacitor circuit architecture using this operational amplifier limit the effective usable bandwidth of a circuit realized in the FPAA to less than 2MHz

Anadigm FPAA

- Library of analog functions implemented in the Configurable Analog Blocks: Filters (Butterworth, Chebyshev, Elliptic), oscillators, amplifiers, adders, user-defined input-to-output transfer functions etc;
- Configuration:
  - data downloaded from PC using RS232;
  - Self contained system: EPROM
- Vendor suggested applications:
  - Adaptive filtering and control;
  - Adaptive DSP front-end;
  - Self-calibrating systems;
  - Compensation for aging of systems components;
  - Ultra-low frequency signal conditioning.

The Switched Capacitor as a "Resistor"



$$R = \frac{V}{I}$$

$$I_{AVG} = \frac{C\,V}{T}$$

$$\boxed{R_{eq} = \frac{\cancel{V}\,T}{C\,\cancel{V}} = \frac{T}{C}}$$

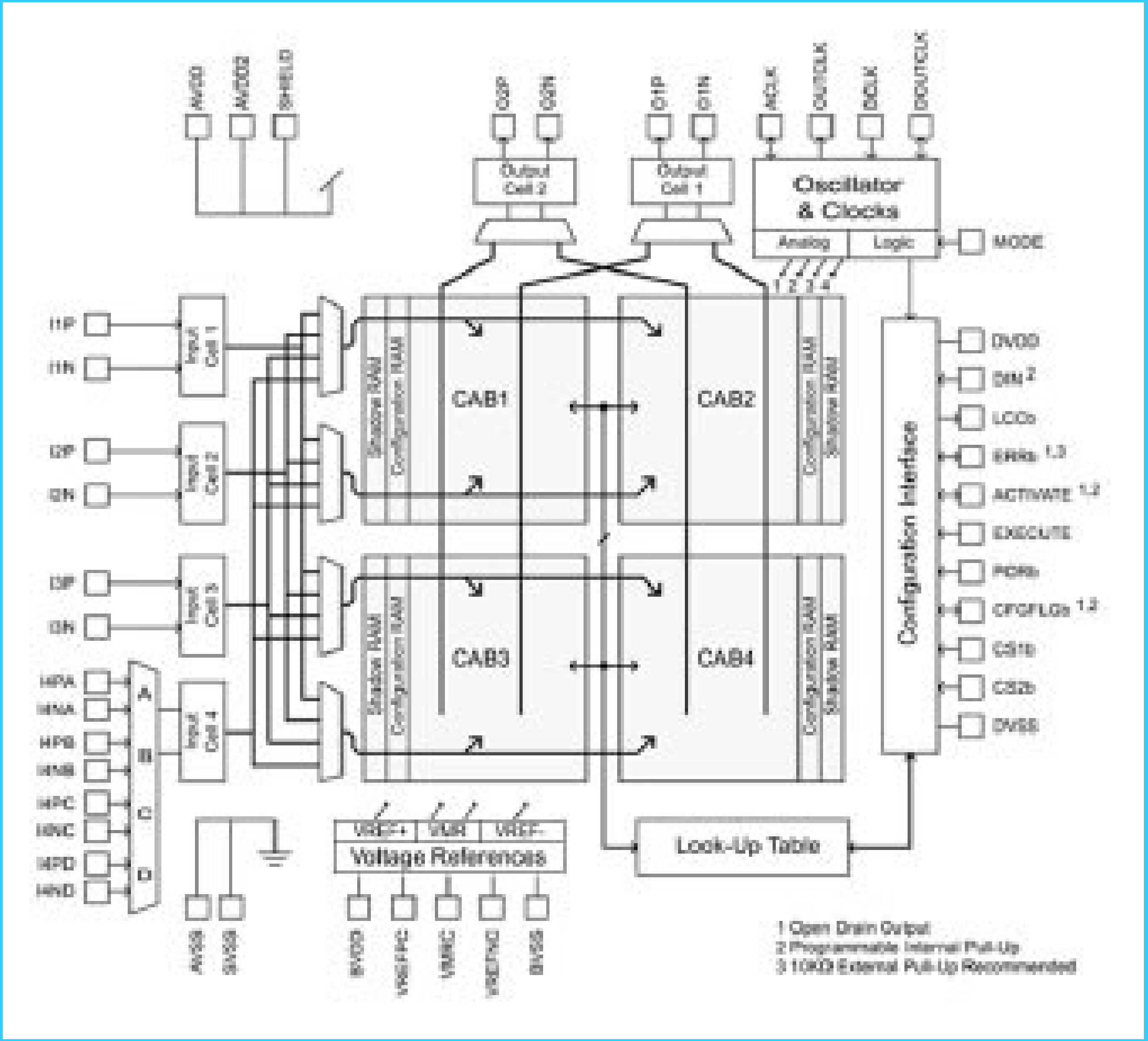
Overview of Anadigm AN220E04 chip

With the AN220E04 dynamically reconfigurable field programmable analog array, you can integrate analog signal conditioning and processing functions into an off-the-shelf, pre-tested device that interacts with other parts of the system through software, putting analog under the absolute control of the system. Based on a fully differential switched capacitor architecture, Anadigm®'s second-generation AN220E04 brings you a new level of device functionality and performance. Compared with our first-generation FPAAs, the AN220E04 provides a significantly improved dynamic performance as well as higher bandwidth. The device also incorporates an 8-bit SAR-based ADC and a 256-byte LUT. Combined, you can use these features to implement complex, non-linear analog functions such as sensor response linearization, arbitrary waveform synthesis, signal-dependent functions, analog multiplication, and signal companding.

Dynamic reconfiguration

Using dynamic reconfiguration, you can manipulate the loop response of your design: change filter characteristics (even the order) in response to changing environmental conditions, page functions in and out in sync with MUX settings, or simply adjust coefficients. All this without interrupting the operation of the FPAA, and all under control of automatically-generated software which is deployed using the same simple "drag-and-drop" approach as the circuit design itself.

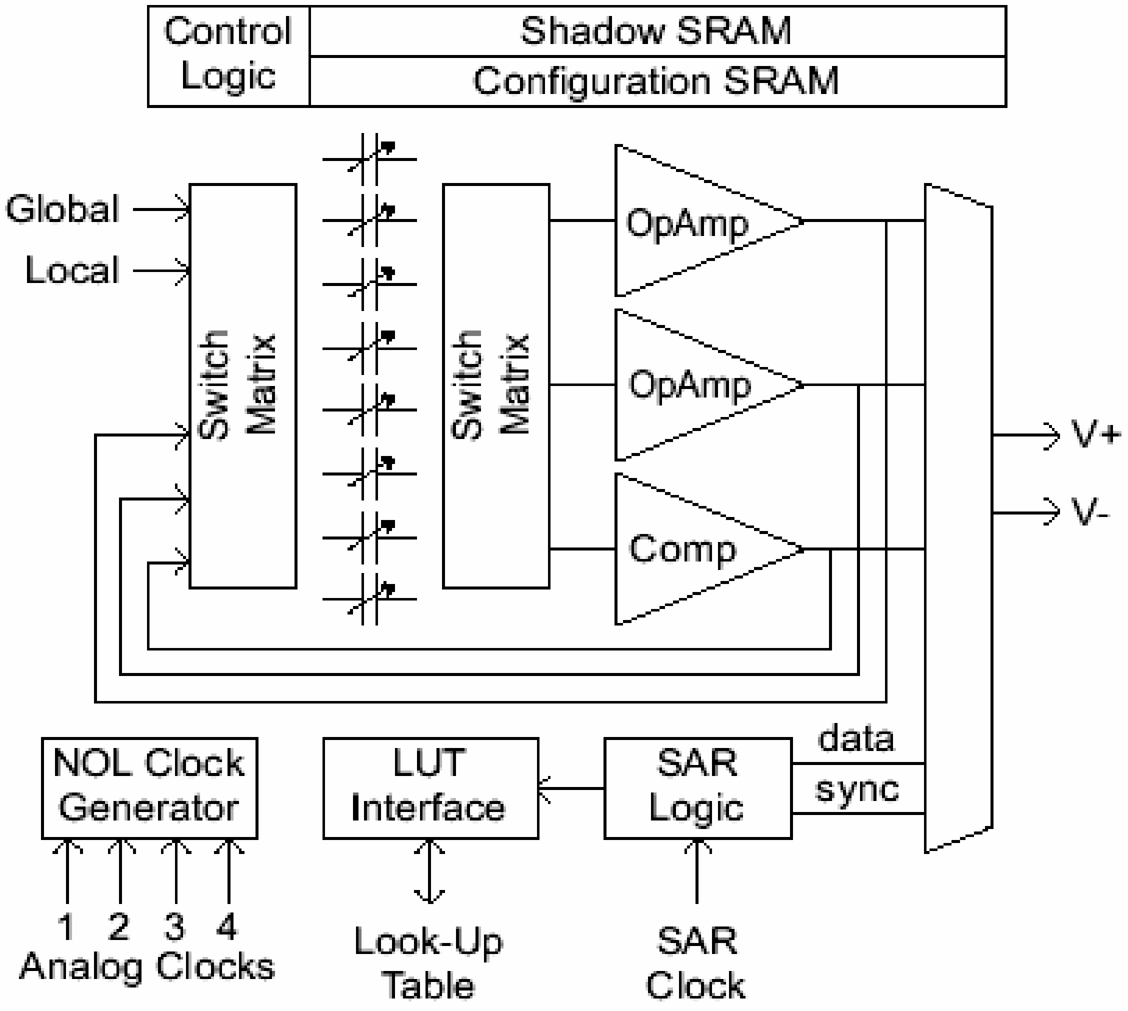
Architecture of AN220E04 FPAA



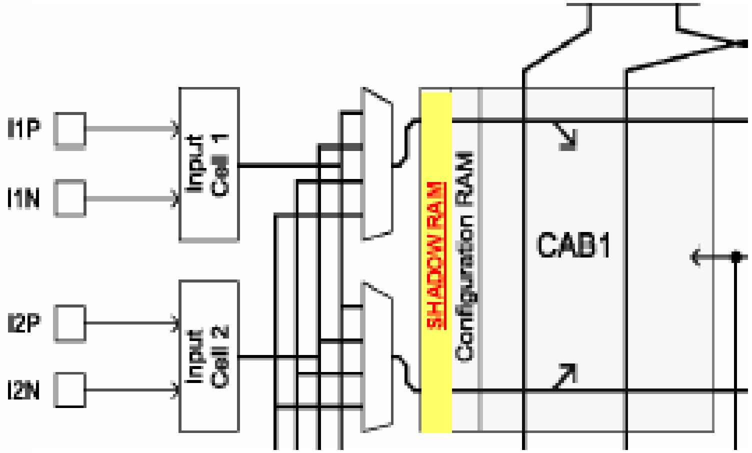
1 Open Drain Output
2 Programmable Internal Pull-Up
3 10KΩ External Pull-Up Recommended

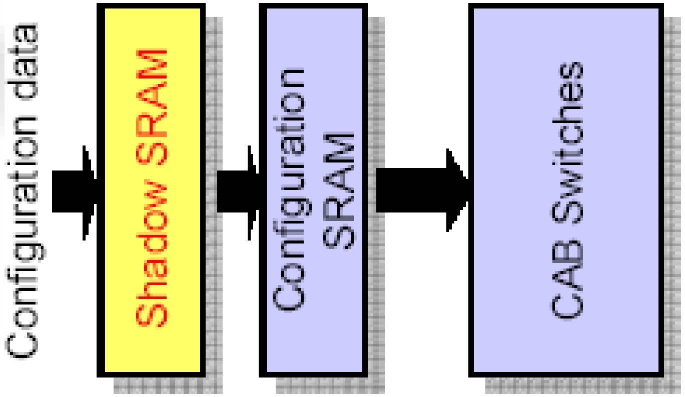
http://www.anadigm.com/_doc/FPAAfamilyOverview.pdf

The Anadigmvortex CAB at a Glance



Dynamic Reconfiguration





Features:

- **Each CAB has two RAM blocks**
  - Configuration SRAM
  - Shadow SRAM
- **After power-up, data moves from the outside world**
  - To the Shadow SRAM
  - Then to the Configuration SRAM
- **While the device is operating**
  - The contents of the Shadow SRAM can be updated without impacting device operation
- **Shadow SRAM contents can be transferred to Configuration SRAM 'on-the-fly'**

More on AN220E04

- The four configurable analog blocks (CABs) in the product -- that is the bits that do the processing -- are based on switched-capacitor array technology acquired by Motorola from Pilkington in the UK (a major glass manufacturer.) Anadigm was subsequently spun off as a venture-backed company. Obviously the use of switched-capacitor technology inevitably leads to limitations in the frequency performance that can be achieved but a lot of the world's products operate at speeds of audio frequencies and lower.

- There are four analog input cells, all of which are differential (but can be operated single-ended), and the fourth input actually has a 4:1 differential input multiplexer driving it. This, of course, is the ideal connection for slow sensor interfacing. The cells have programmable anti-aliasing filters with a high-gain amplifier (with optional chopper stabilizer mode); the high precision input range is 0.5 to 3.5 V single-ended, up to 6.0 V differential and the chopper reduces the input offset from 15 mV (max.) down to 100 uV (max.)

- Through a crossover system all the inputs can be associated with any multiple (programmable interconnect resources, the company calls them) of the CABs and those in turn drive either (or both) of the output cells, which have the same ranges as the inputs. The four CABs share a single look-up table which is driven from a configuration interface. The 256 byte look-up table can be used to generate arbitrary signals or be used as a gamma corrector to linearize a signal, or to define a transfer function. Non-linear functions can be defined with an 8-bit SAR ADC in loop to the table. Clocks and three different voltage references are also available. Each of the four CABs has both shadow and configuration RAM, allowing new settings to be loaded in the shadow RAM before being implemented by transfer to the configuration RAM.

- acquisitionZONE Products for the week of September 9, 2002
- AN220E04: Anadigm Ships Dynamically-Reconfigurable Field-Programmable Analog Array Allows Rapid Analog Design Implementation and On-the-Fly Reconfiguration

More on AN220E04

- Frequency maxes out at 2 MHz while clocks can be run at up to 40 MHz. The numbers are good enough for quite high-quality audio work with THD down at 80 dB and audio band SNR at 100 dB (broadband is 80 dB), crosstalk is better than 70 dB and the input amplifiers have a bandwidth of 170 MHz and a slew rate of better than 150 V/µs -- considerable improvements over the initial product offering (the AN10E400) as well as adding the dynamic reconfigurability.

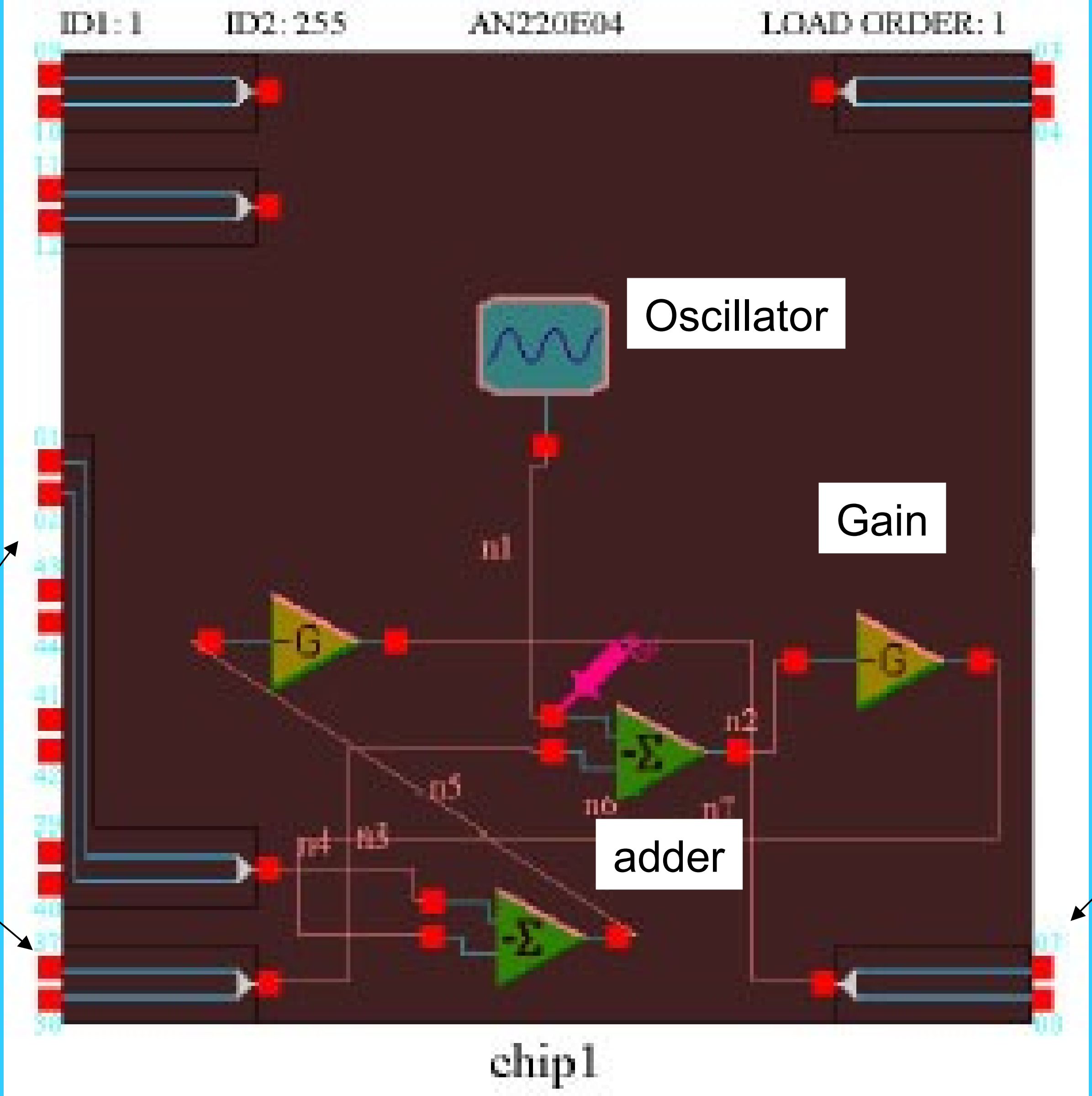
- Power consumption of the 5-V parts varies with the amount of IC in use but ranges from about 25 to 60 mA in a low-power mode and 75 to 200 mA in the full-power mode.

- Applications range from sensing of all kinds to industrial control, medical monitors, signal conditioning, and filtering of many kinds. The dynamic reconfigurability can be used specifically in complex test equipment variables, and the use of the same device with slightly different programming can give OEMs the base for offering products with varying levels of performance from one design.

- The AN220E04 is in production in a QFP-44 and is priced at $15 in 10-k piece lots. An evaluation kit complete with entry-level software and documentation is available for $499. A multimedia demonstration of the EDA tool can be viewed on the company's web site. www.anadigm.com

- http://www.analogzone.com/acqp0909.htm

Anadigm Designer Schematic Software



Oscillator

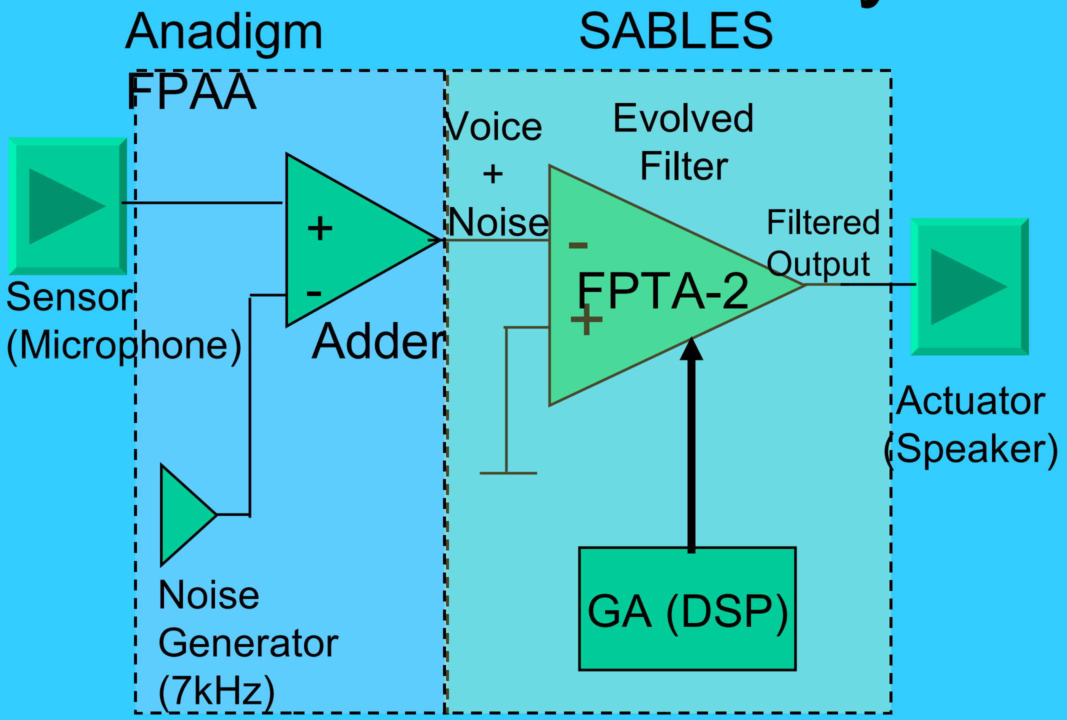
Gain

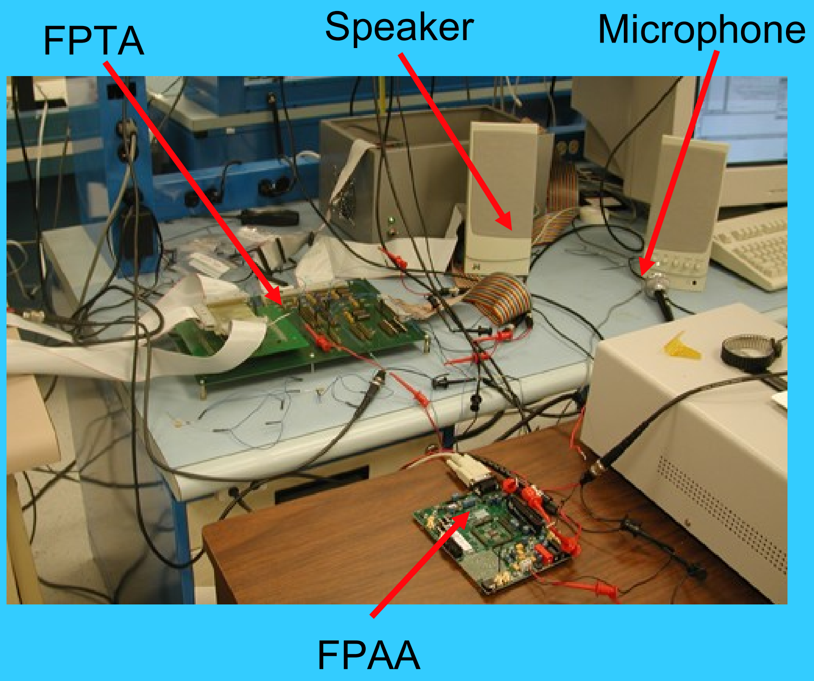
adder

Inputs

Output

Example of hybrid FPAA –FPTA system



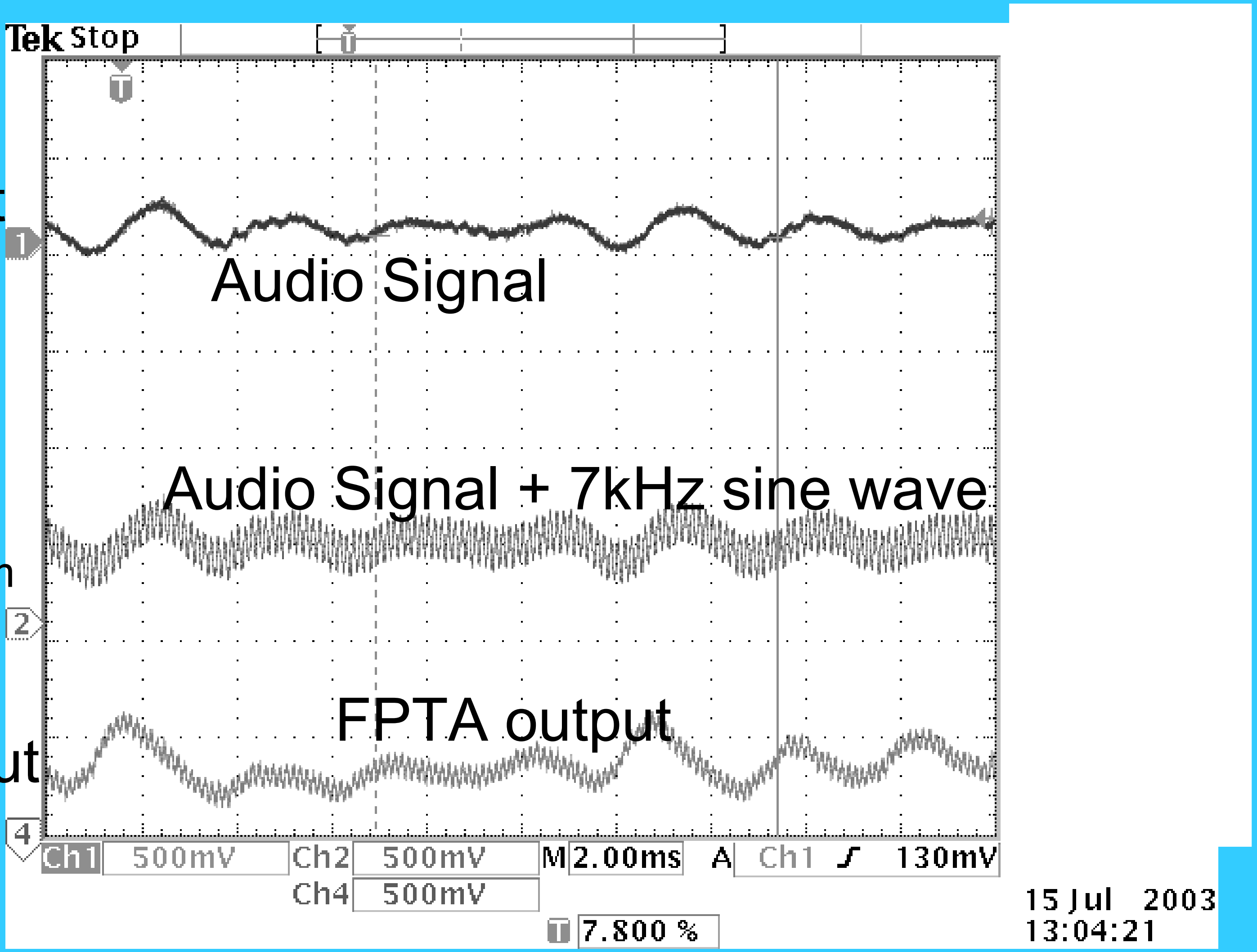


Application: EHW for Sensing
Real-Time Noise Elimination from Audio Signal

DSP + FPTA (Example of Application)

- Microphone acquires a voice signal coming from a radio in real-time which is mixed with a noise signal and conditioned for the FPTA-2 (shift the DC value)
- FPTA-2 is evolved to separate the two signals

EHW Platform for Sensing



Target

Audio Signal

Input
(from Anadigm FPAA)

Audio Signal + 7kHz sine wave

Output

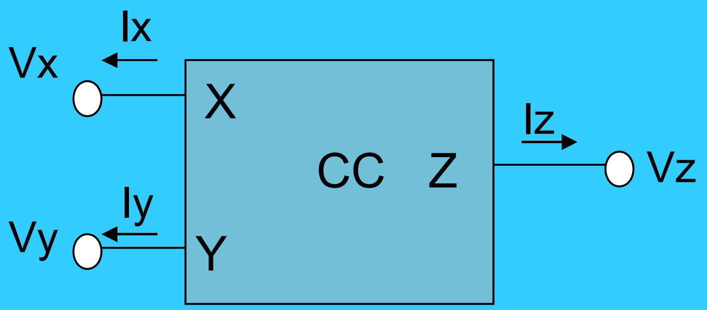
FPTA output

FPAA Challenges

- Accuracy (device matching);
- Digital noise (clock feedthrough);
- Analog noise (power supply, thermal, shot, 1/f noise);
- Bandwidth limitations:
  - Most designs: around 100kHz;
  - Zetex: 4 MHz
  - Anadigm: 2MHz

Research FPAA with current conveyors

- Gaudet and Gulak (University of Toronto): Use of current conveyors instead of OpAmps to achieve CMOS FPAAs operating at 10MHz bandwidths.



- Switch resistance has little effect on current-mode circuits;
- When a voltage is applied at node Y, that voltage is replicated at node X. This is similar to the virtual short of an OpAmp, but there is no need for negative feedback to achieve it. When current is injected into node X, that same current gets copied into node Z.
- Continuous time analog functions can be realized by connecting other elements to the various terminals of the current conveyor.
- In addition to bandwidth, current conveyors are advantageous compared to OpAmps, due to the fact that they do not require compensation.

2. Reconfigurable and Morphable Hardware

2.1. Reconfigurable hardware (switch-based). Devices, SW Tools, Potential for EHW
2.2. Field Programmable Gate Arrays (FPGA) – Xilinx examples
2.3. Field Programmable Analog Arrays (FPAA) – Anadigm Examples
2.4. Field Programmable Transistor Arrays (FPTA) – JPL examples
2.5. Reconfigurable antennas
2.6. Other reconfigurable structures
2.7. Speed of reconfiguration, partial reconfiguration, context-switching
2.8. Morphable hardware (no switches). Fine changes and tuning.
2.9. Morphable Materials and devices
2.10. Polymorphic circuits

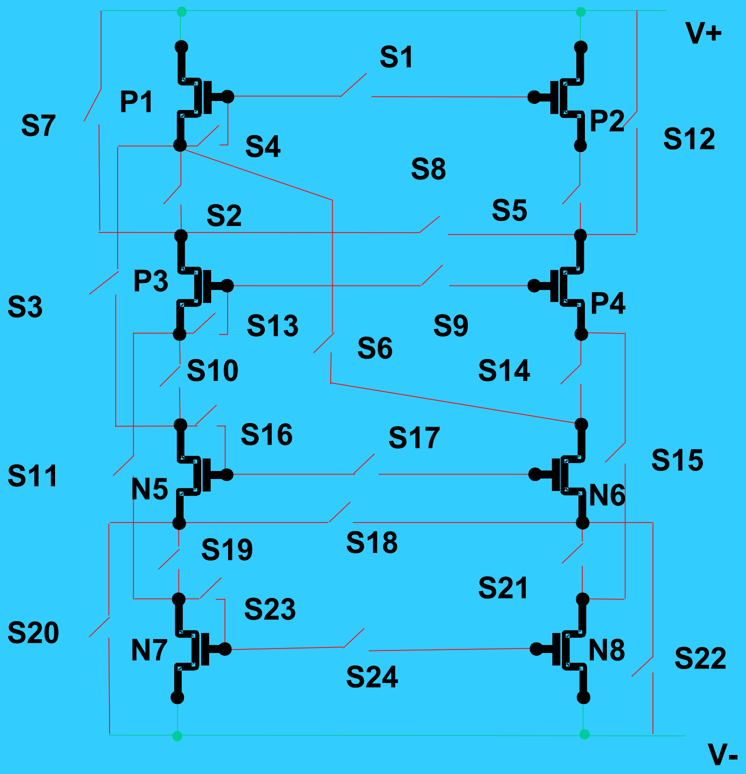
Why Custom

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



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.

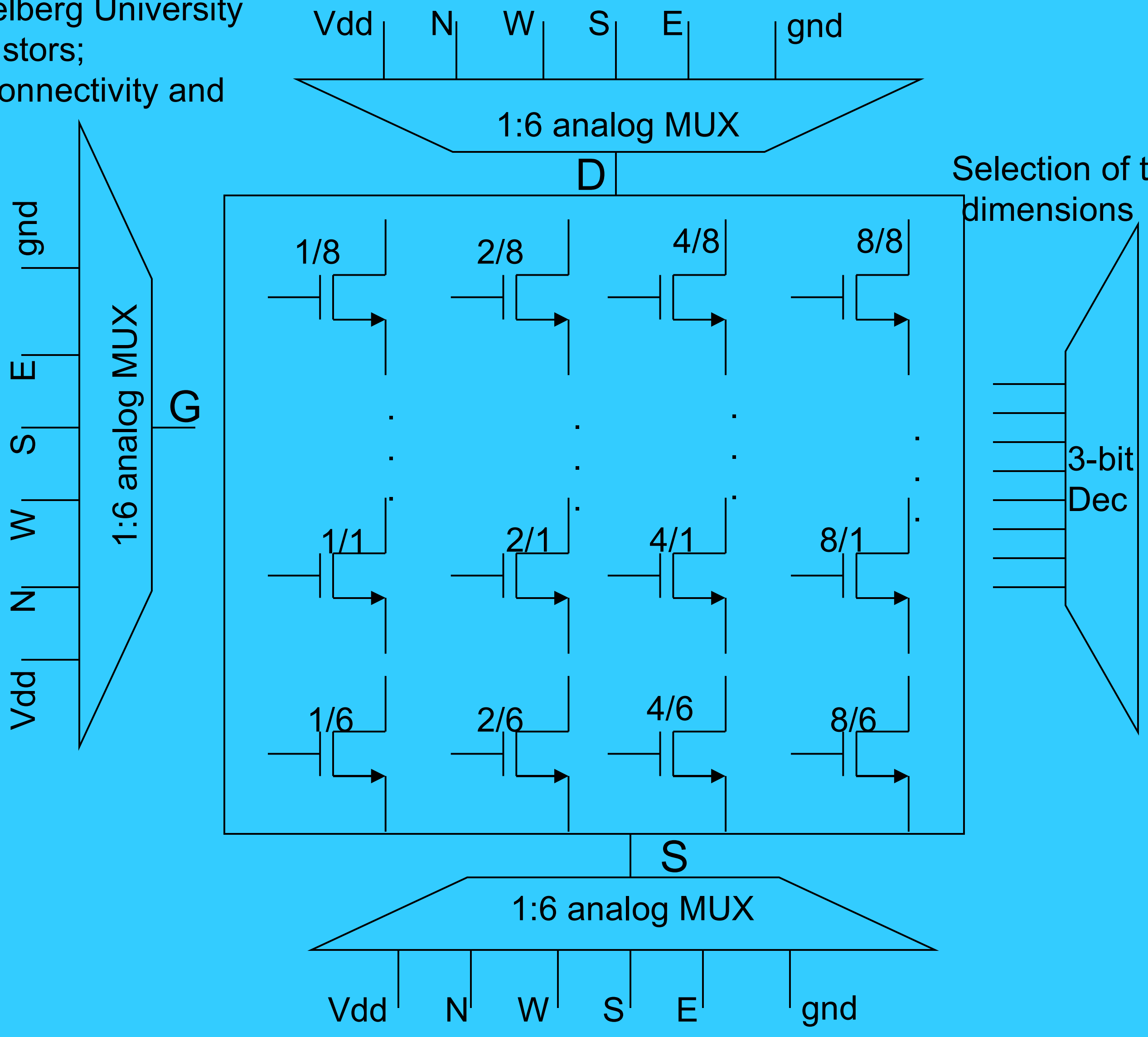
Survey of various analog reconfigurable platforms

Feature /Device	TRAC	VORTEX	PALMO	EM	Lattice
Granularity	coarse	coarse	coarse	fine	coarse
Protection	NA	software tools	software/ hardware	switches parasitics	NA
Circuit Download	parallel port	serial port	serial port	ISA bus	serial port
Proprietary Information	NA	Yes	NA	No	Yes
Search Space	NA	$\sim 2^{300}$/cell	NA	10^{420}	NA
Technology	CMOS	CMOS	BiCMOS	Board level	CMOS

(NA – Information not available).

FPTA of U. Heldelberg

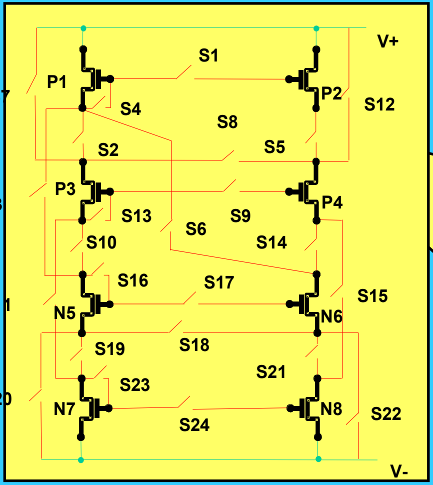
- Langeheine @ Heidelberg University
- Array of 16x16 transistors;
- Programmability in connectivity and channel lengths.



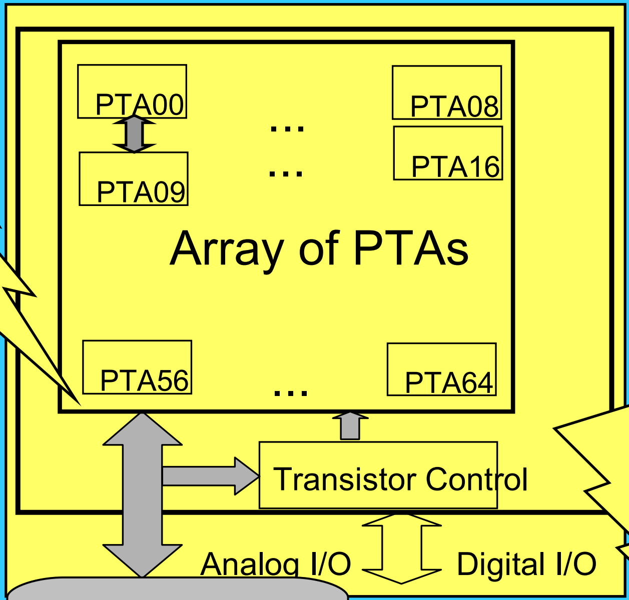
Selection of transistor dimensions

Toward an evolvable SOC

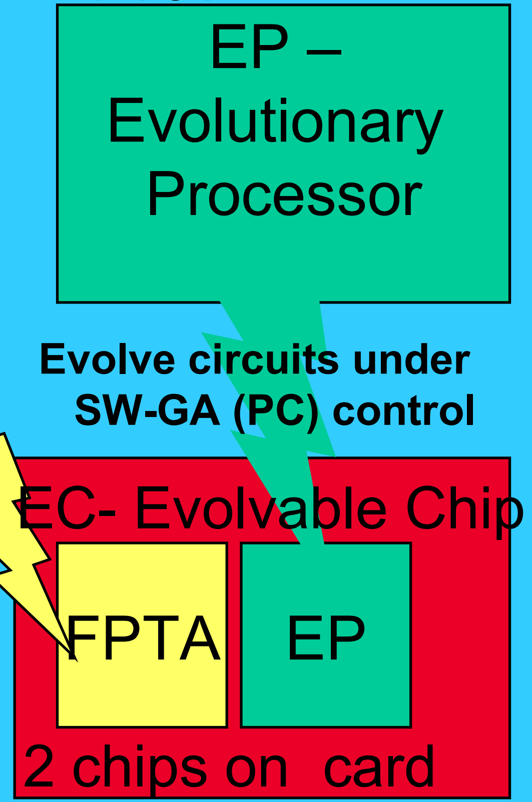
FY99



FY00



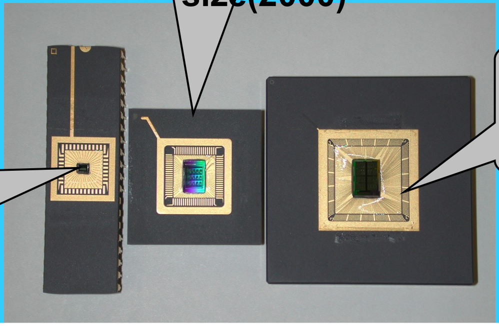
FY01



Evolve circuits under SW-GA (PC) control

Planned

FPTA1: 12 cells 0.5u feature size(2000)



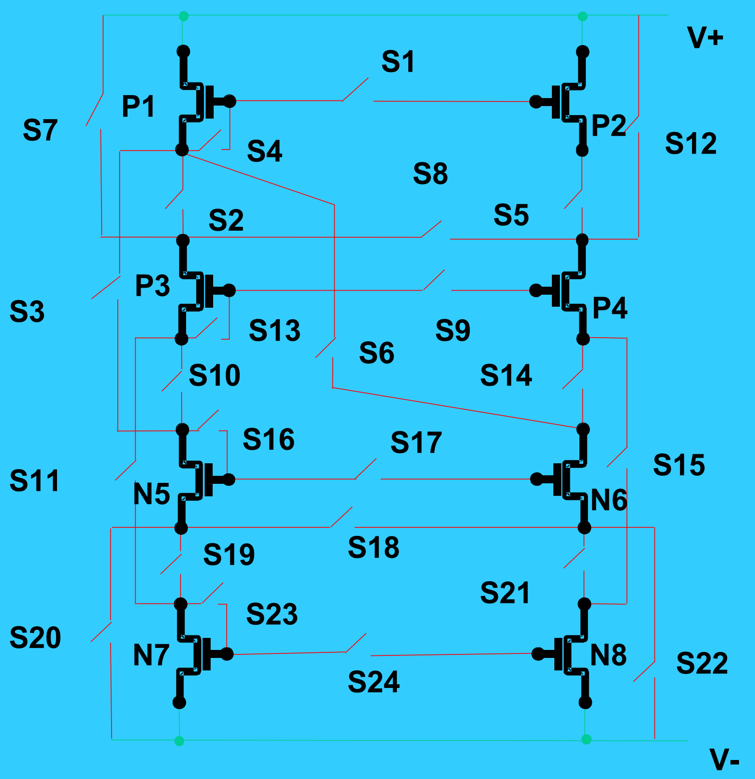
PTA: one cell 0.5u feature size (1998)

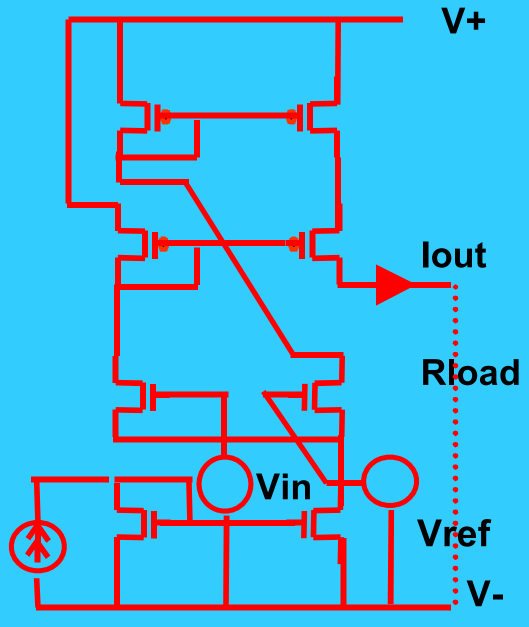
FPTA2: 64 cells 0.18u feature size (2001)

FPTA3: Stacked sensor-analog-digital processing

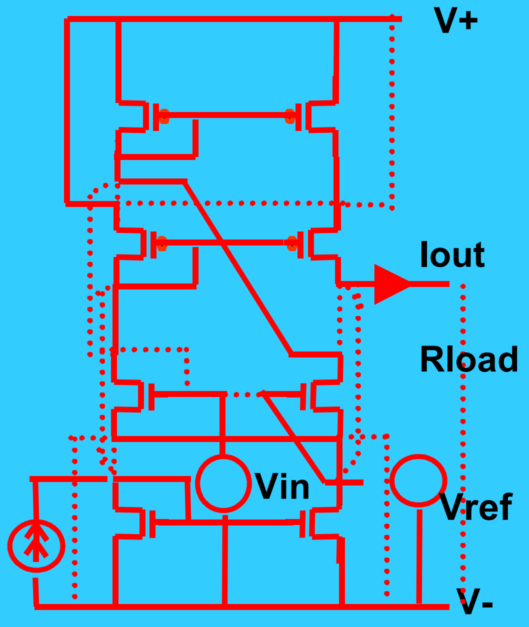
Three generations of evolution-oriented devices

Programmable Transistor Array Cell – FPTA0



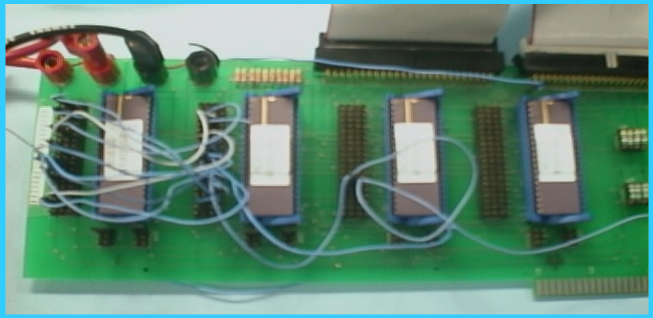


Human Design

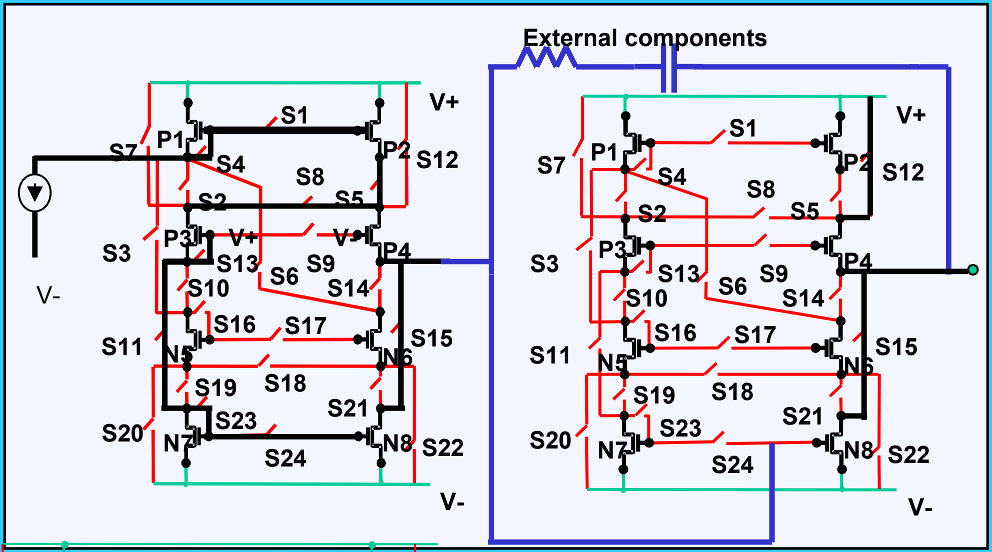


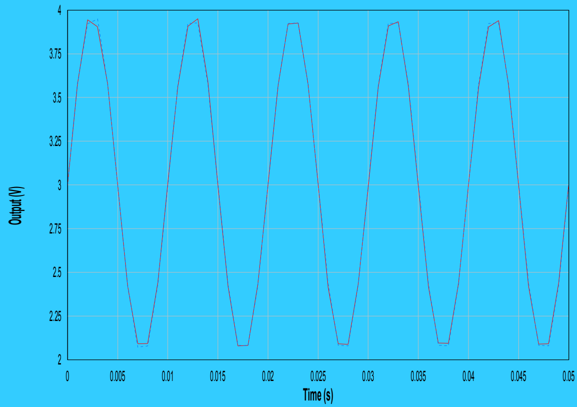
Leakage through finite resistance OFF

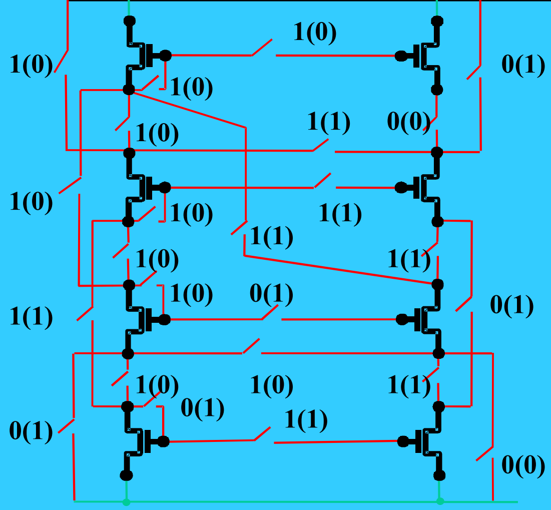
- 24 programmable switches: sufficient number for meaningful topologies
- Chromosomes give the value HIGH-LOW (not only ON-OFF) of the switches
- All the terminals are connected via switches to expansion terminals
- CMOS (0.5 μ) - MOSIS

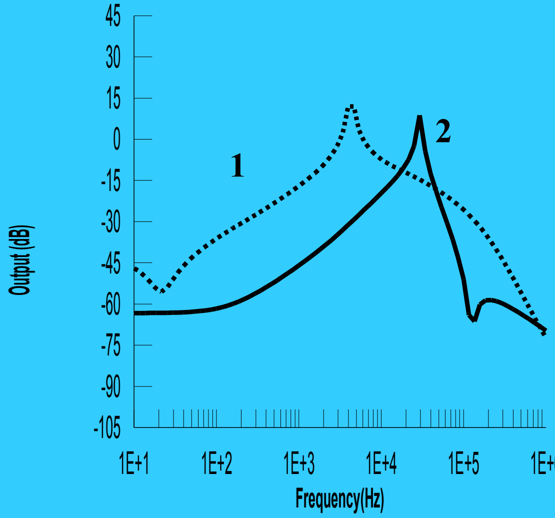


Op.Amp, Filters Mapped into PTA cells









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 .

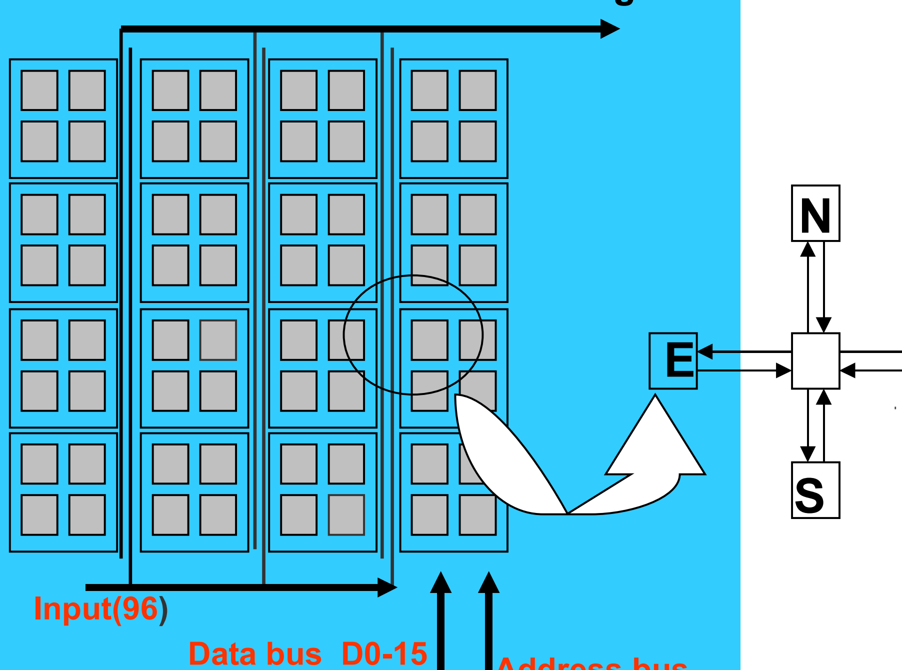
Programmable Transistor Array Cell – FPTA2

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

🟩 **TSMC 0.18 – 1.8v;**

Output(64)

Interconnection amongst cells



Input(96)

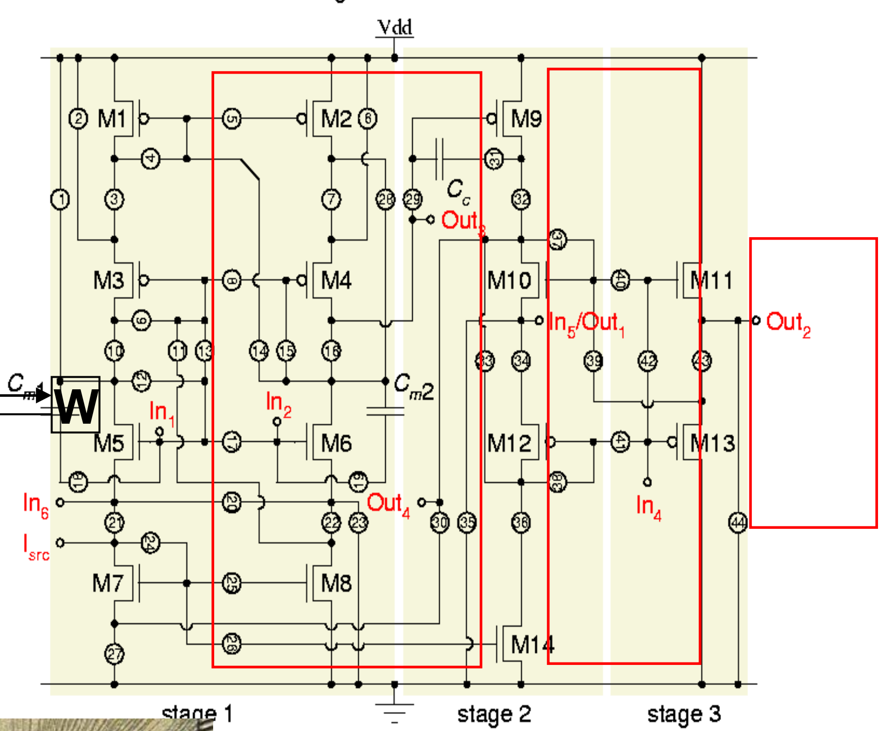
Data bus D0-15 (Configuring bitstring)

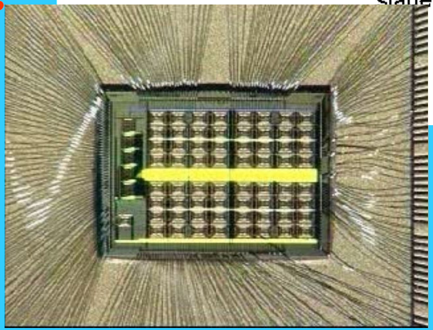
Address bus (9 bits)

Control Logic

Chip Architecture

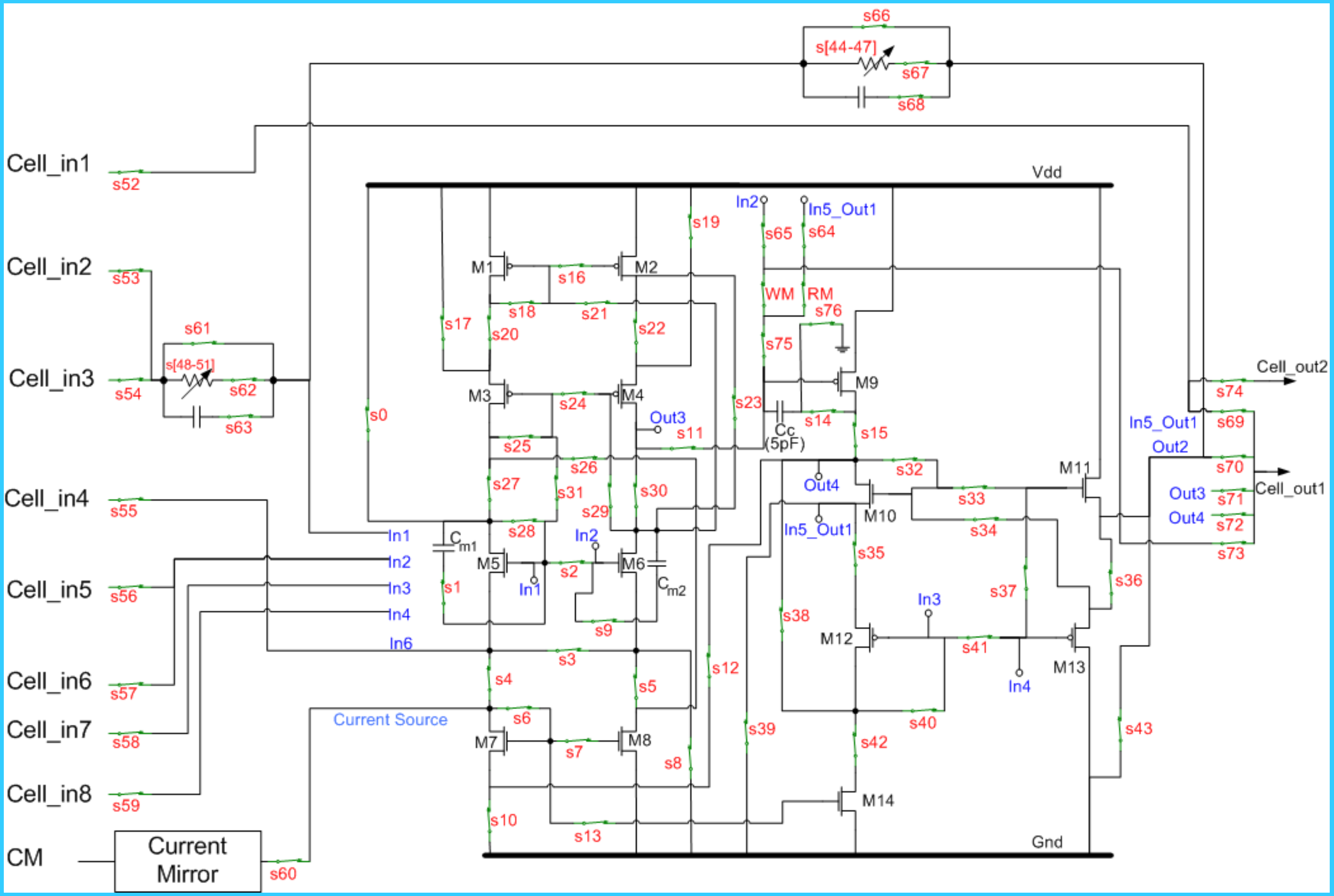
Reconfigurable circuit block



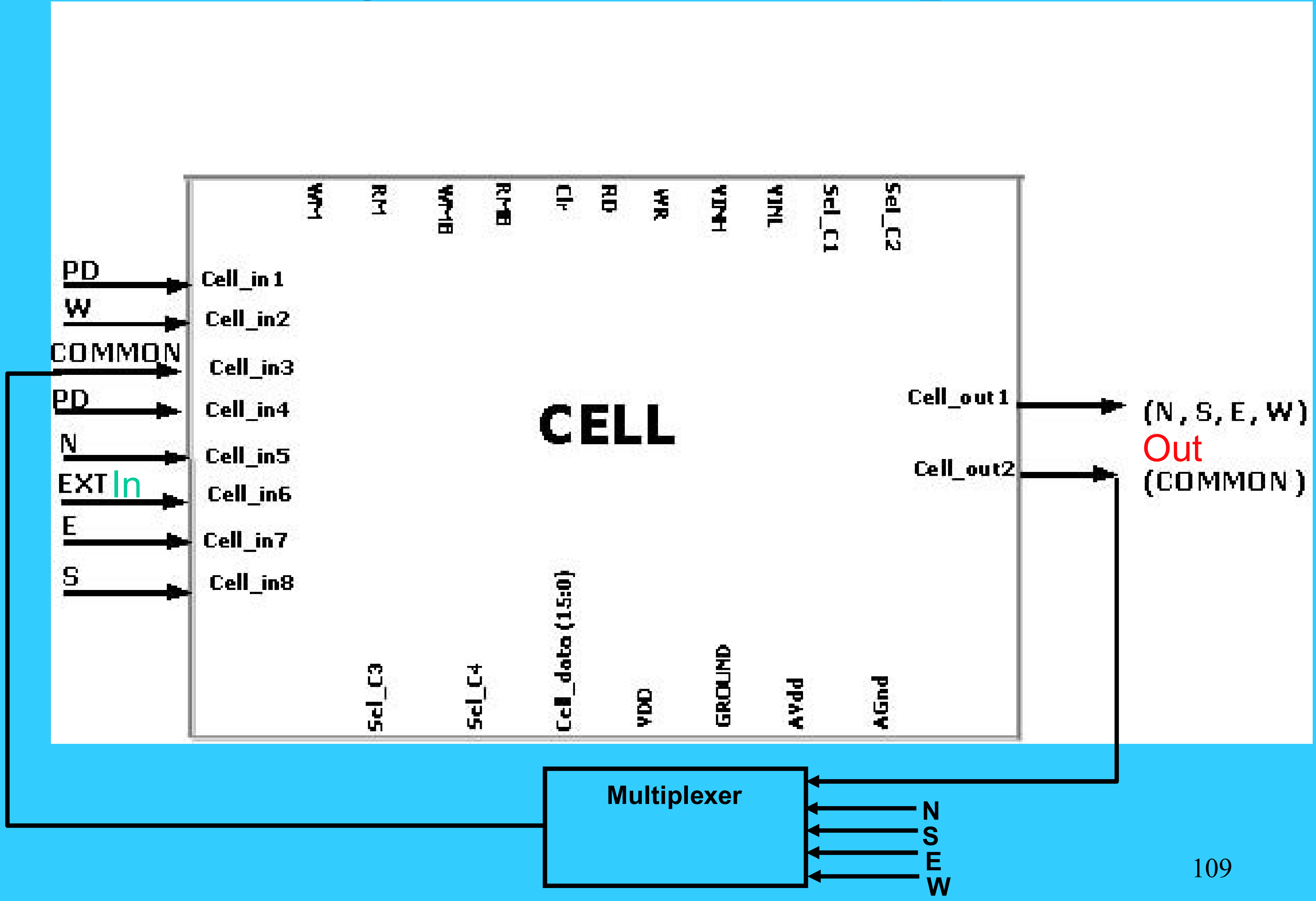


Cell Schematic

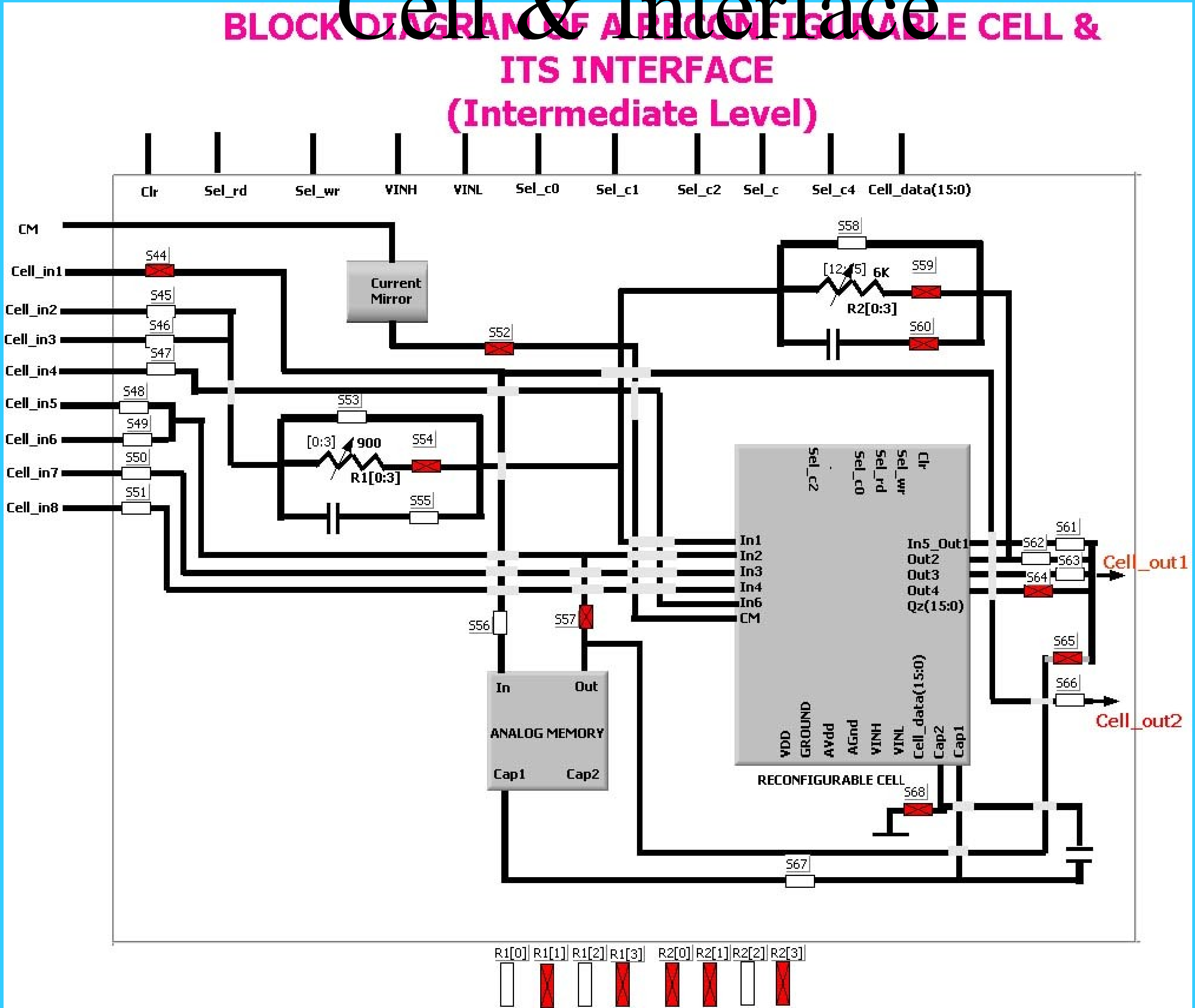
JPL FPTA2 Cell Schematic



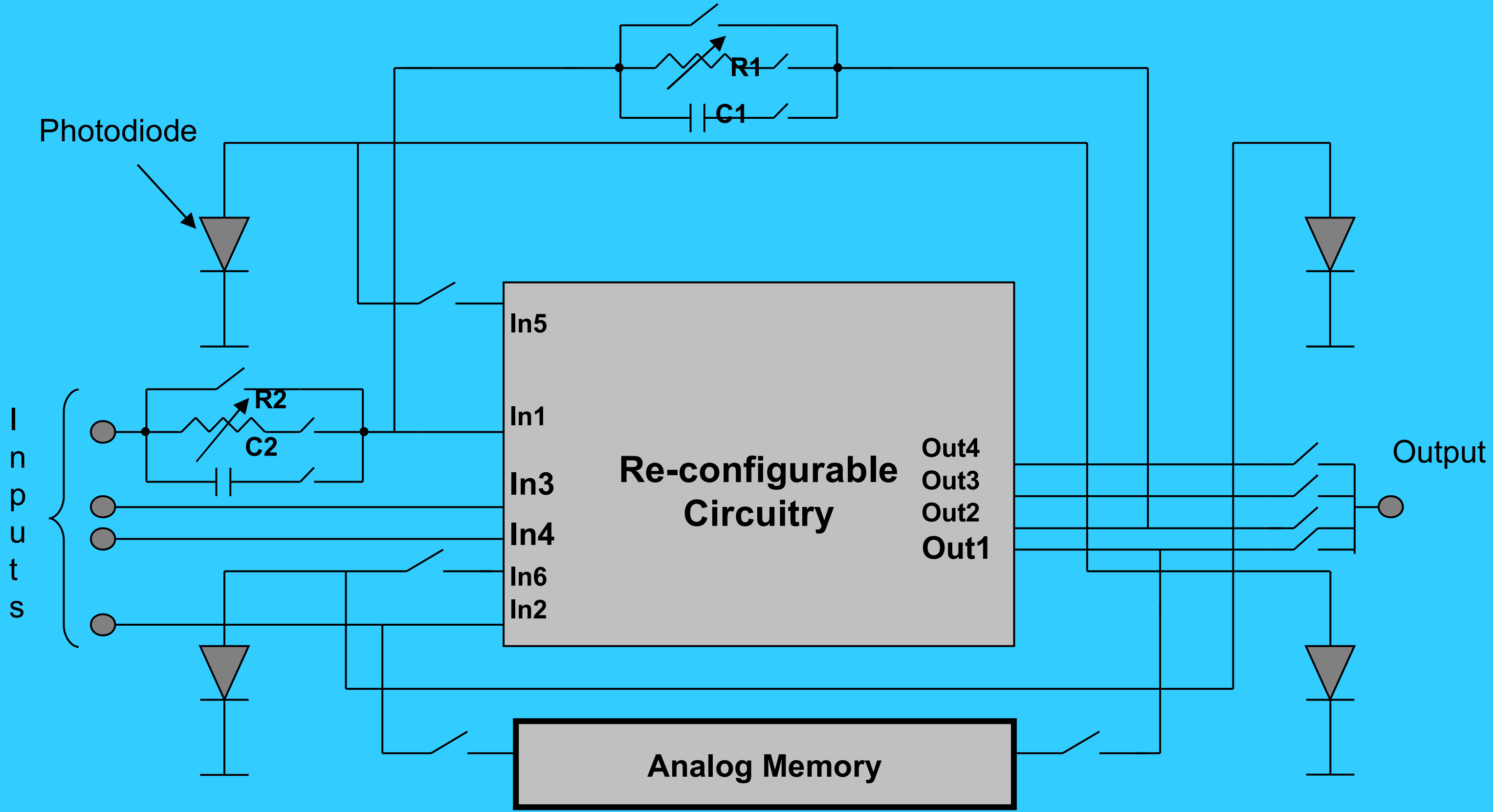
Reconfigurable Cell (Top Level)



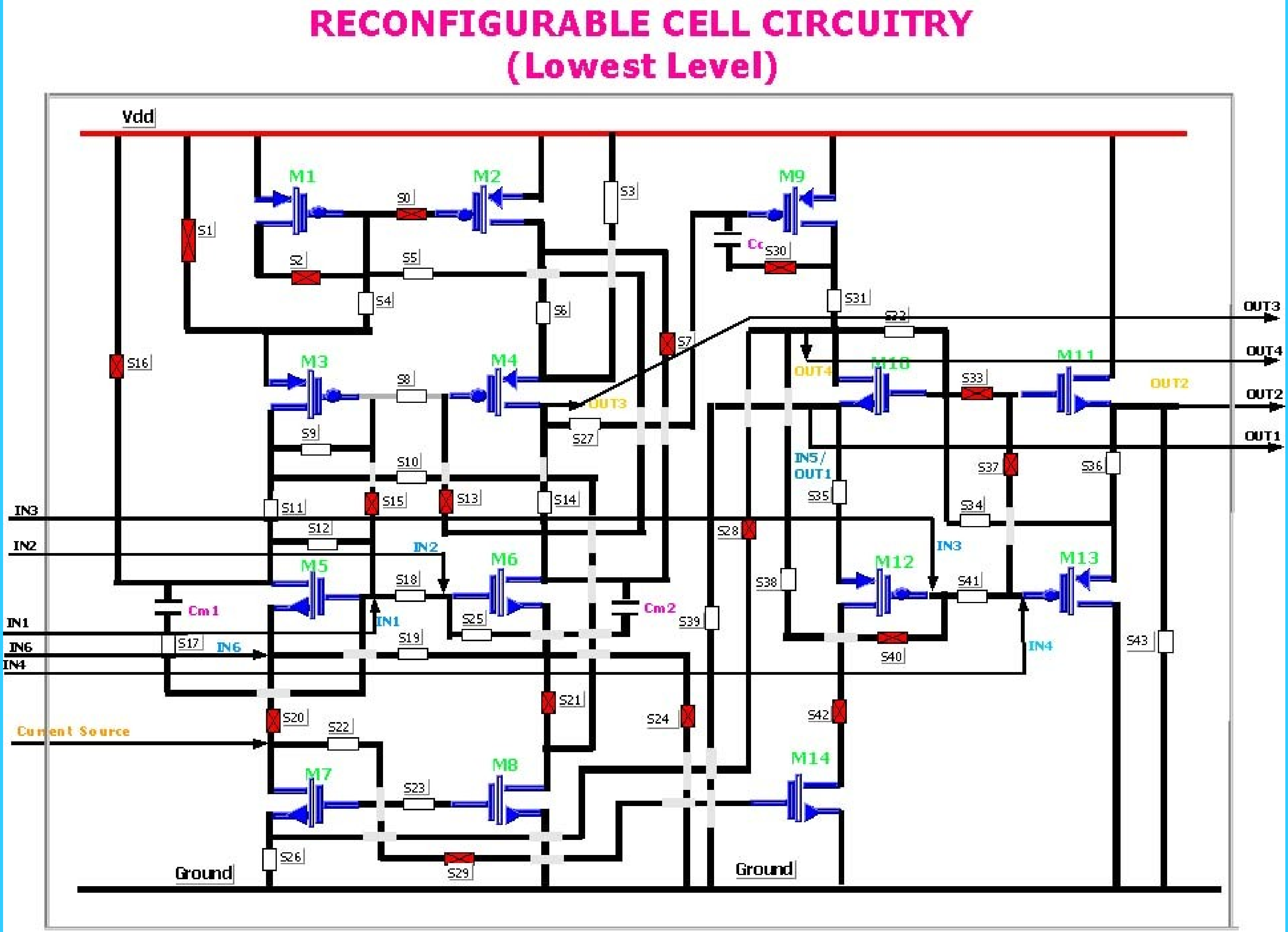
Block Diagram of Reconfigurable Cell & Interface



Block Diagram of Reconfigurable Cell & Interface

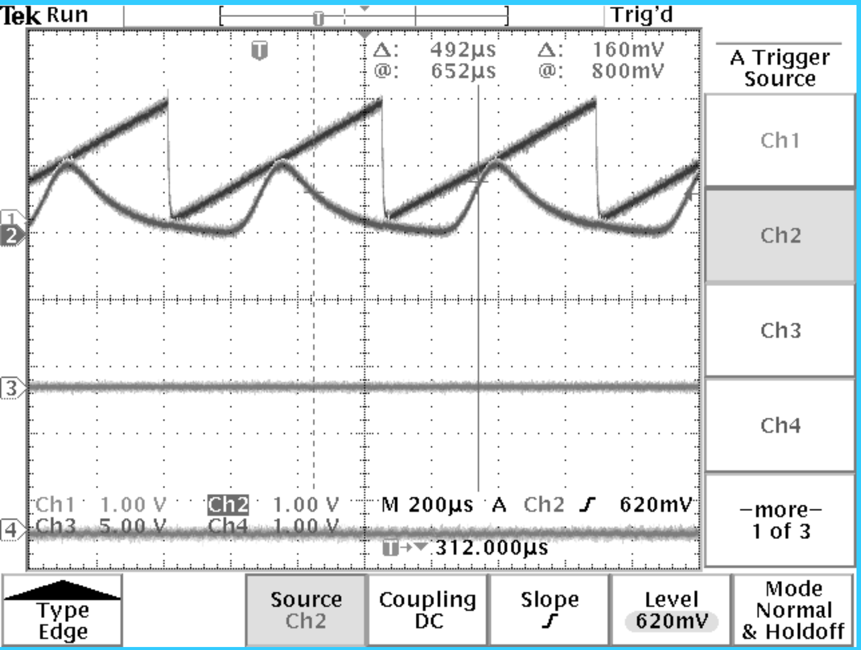


Reconfigurable Cell Circuitry (Lowest Level)



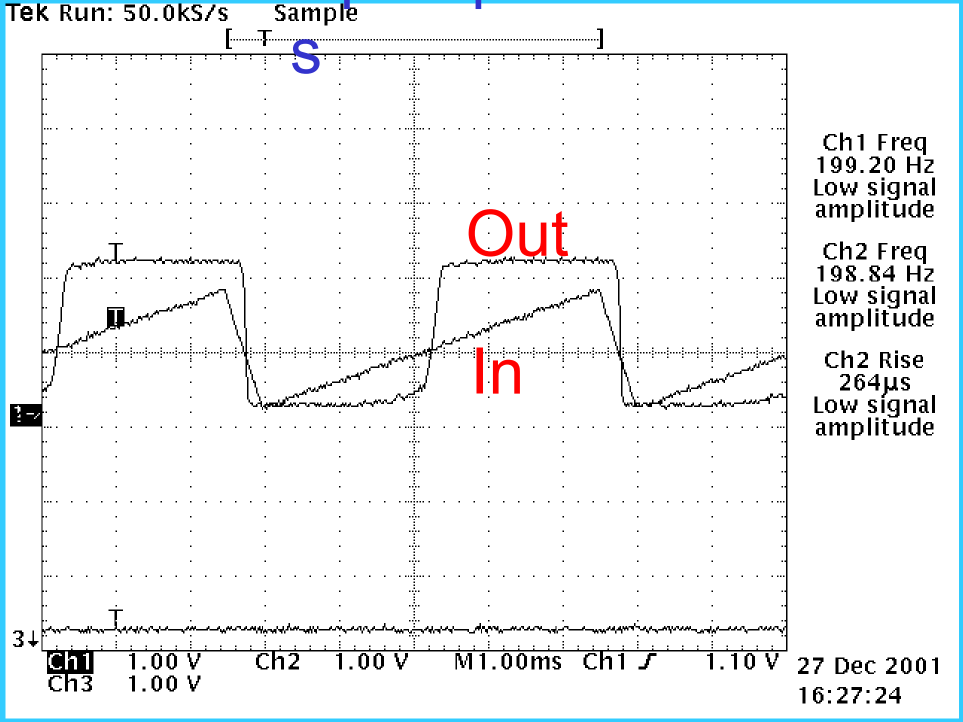
FPTA2 - Tested Circuits

Gaussian



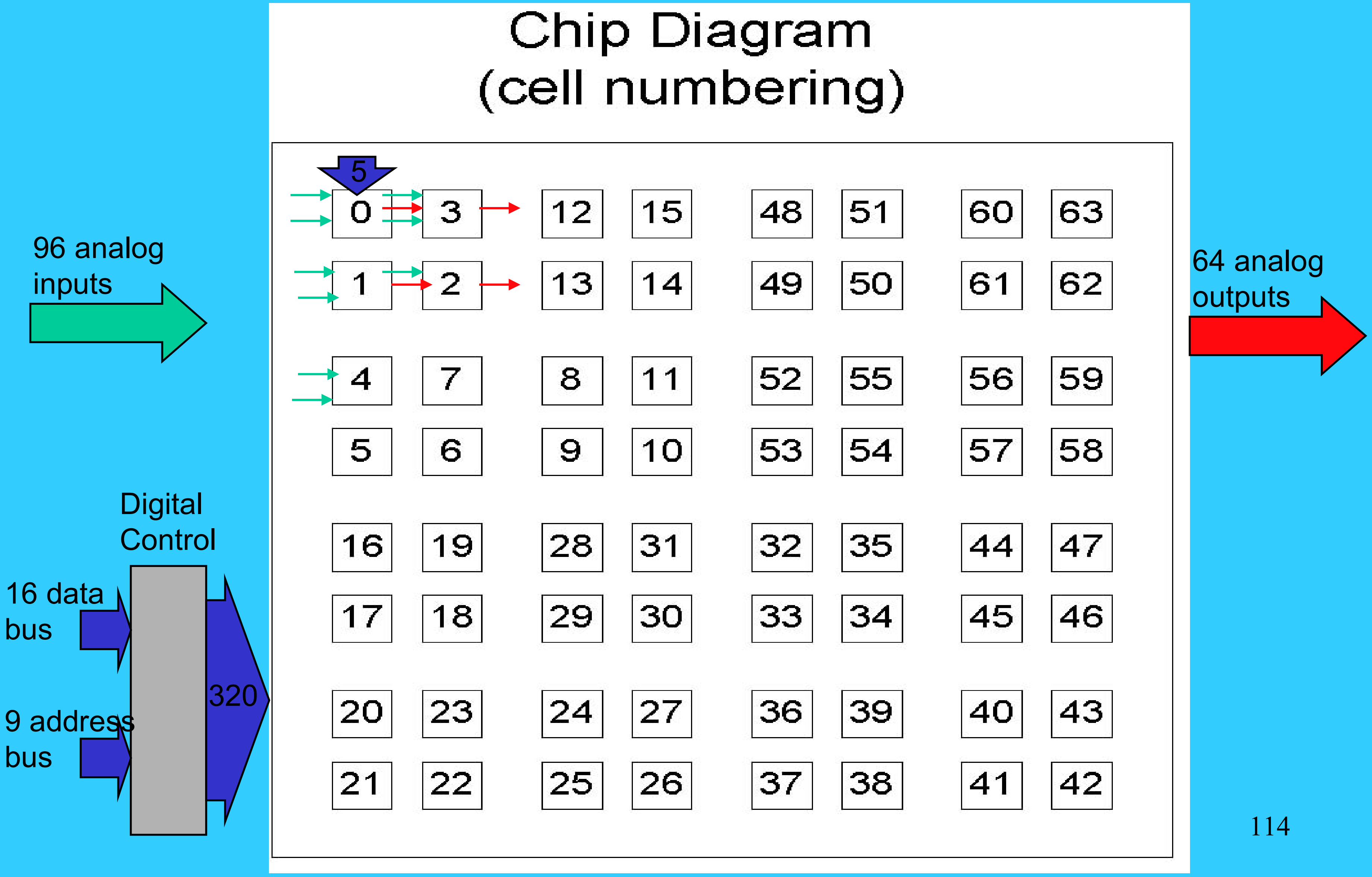
- FPTA: cell 1
- Inputs:
  - In1, In2
  - ramp 0 to 2 Volt
  - Frequency: 3 KHz
- Output: Out3

OpAmps



- FPTA: cell 1
- Inputs:
  - differential inputs
  - In1(In) and In2(0.9Volt)
  - Frequency: 200Hz (up to 100KHz)
- Output: Out2

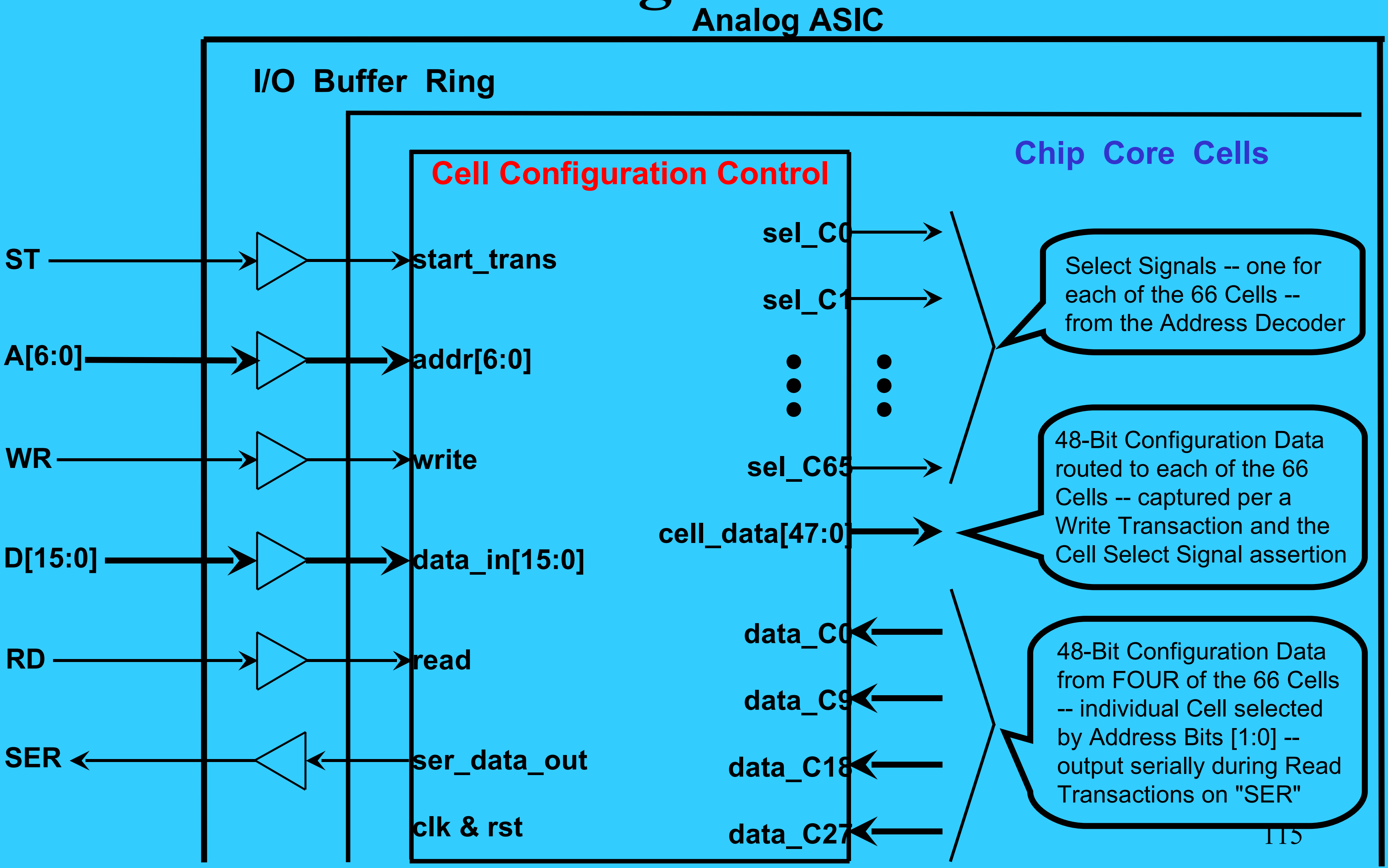
Chip Diagram



Chip Diagram
(cell numbering)

0	3	12	15	48	51	60	63
1	2	13	14	49	50	61	62
4	7	8	11	52	55	56	59
5	6	9	10	53	54	57	58
16	19	28	31	32	35	44	47
17	18	29	30	33	34	45	46
20	23	24	27	36	39	40	43
21	22	25	26	37	38	41	42

External Port Listing & Internal Signals

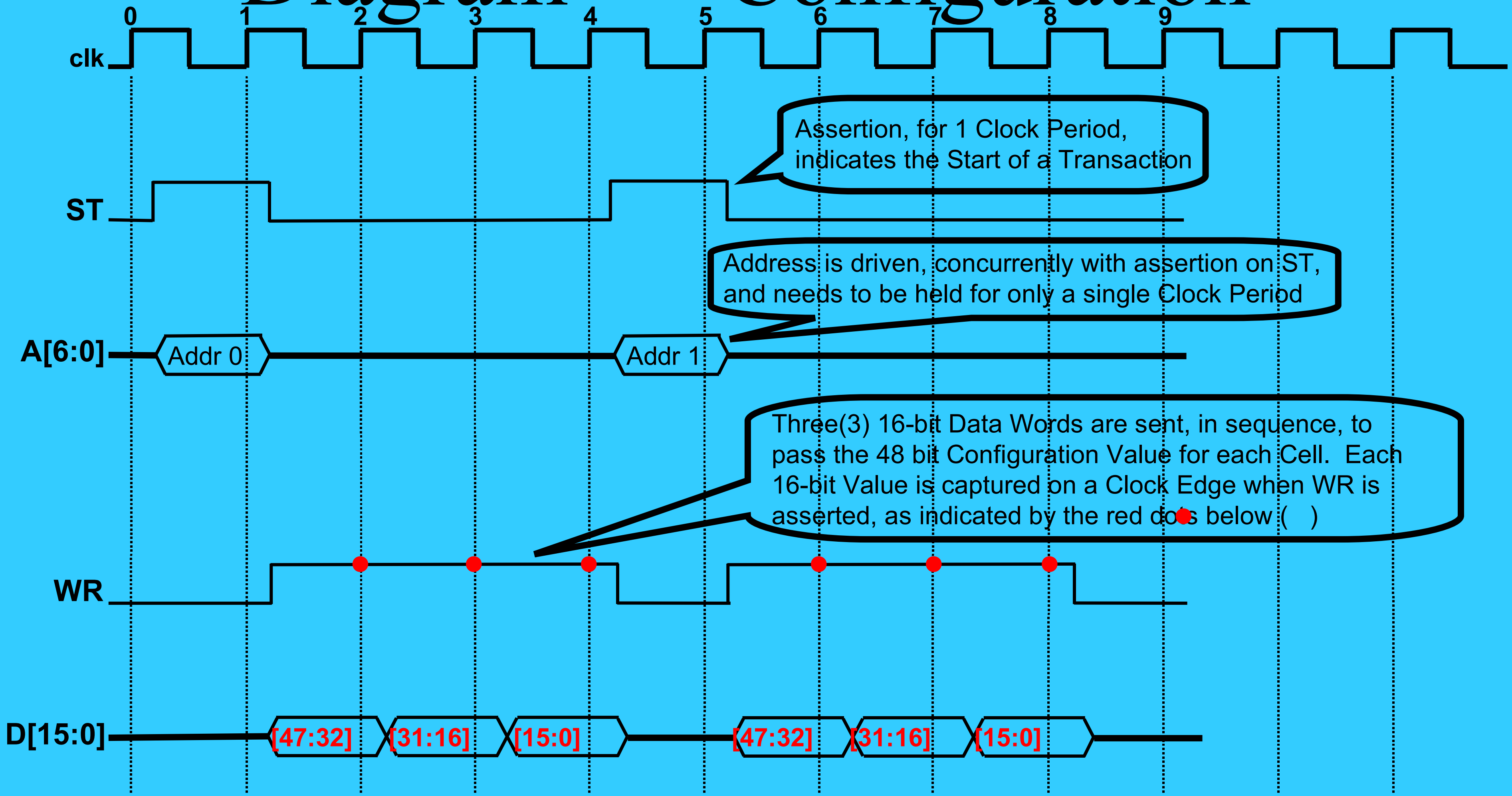


Select Signals -- one for each of the 66 Cells -- from the Address Decoder

48-Bit Configuration Data routed to each of the 66 Cells -- captured per a Write Transaction and the Cell Select Signal assertion

48-Bit Configuration Data from FOUR of the 66 Cells -- individual Cell selected by Address Bits [1:0] -- output serially during Read Transactions on "SER"

WRITE Protocol Timing Diagram -- Configuration



Assertion, for 1 Clock Period, indicates the Start of a Transaction

Address is driven, concurrently with assertion on ST, and needs to be held for only a single Clock Period

Three(3) 16-bit Data Words are sent, in sequence, to pass the 48 bit Configuration Value for each Cell. Each 16-bit Value is captured on a Clock Edge when WR is asserted, as indicated by the red dots below ( )

Each Write Transaction, for Configuration, takes 4 Clock Periods. The 1st clock Period is for the Start Transaction signal to assert, "ST", and the Address to be drive, "A". During the subsequent 3 Clock Periods, 16-bits of Data are Written, to comprise the 48-bit Configuration Value for each internal Analog Cell