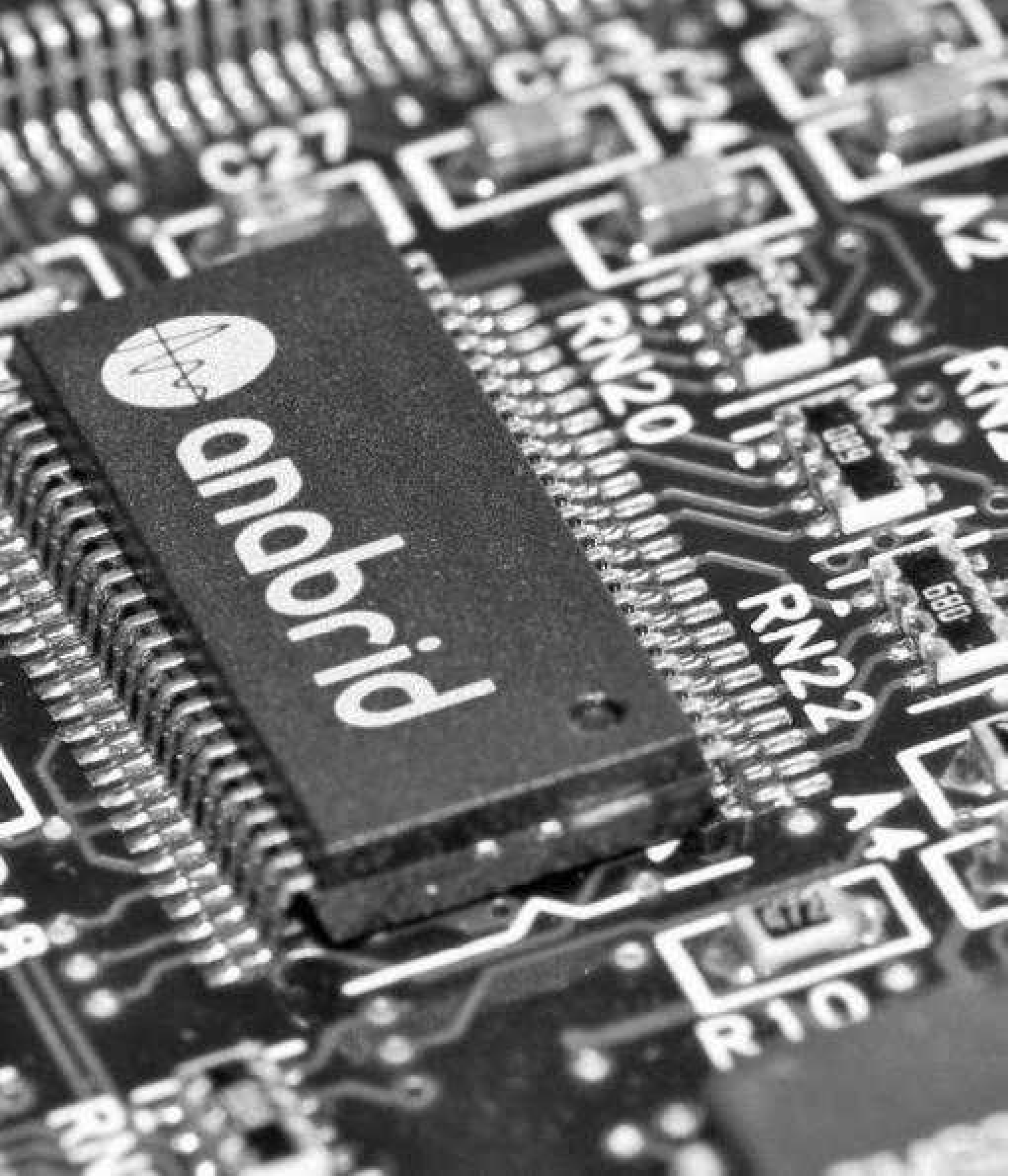
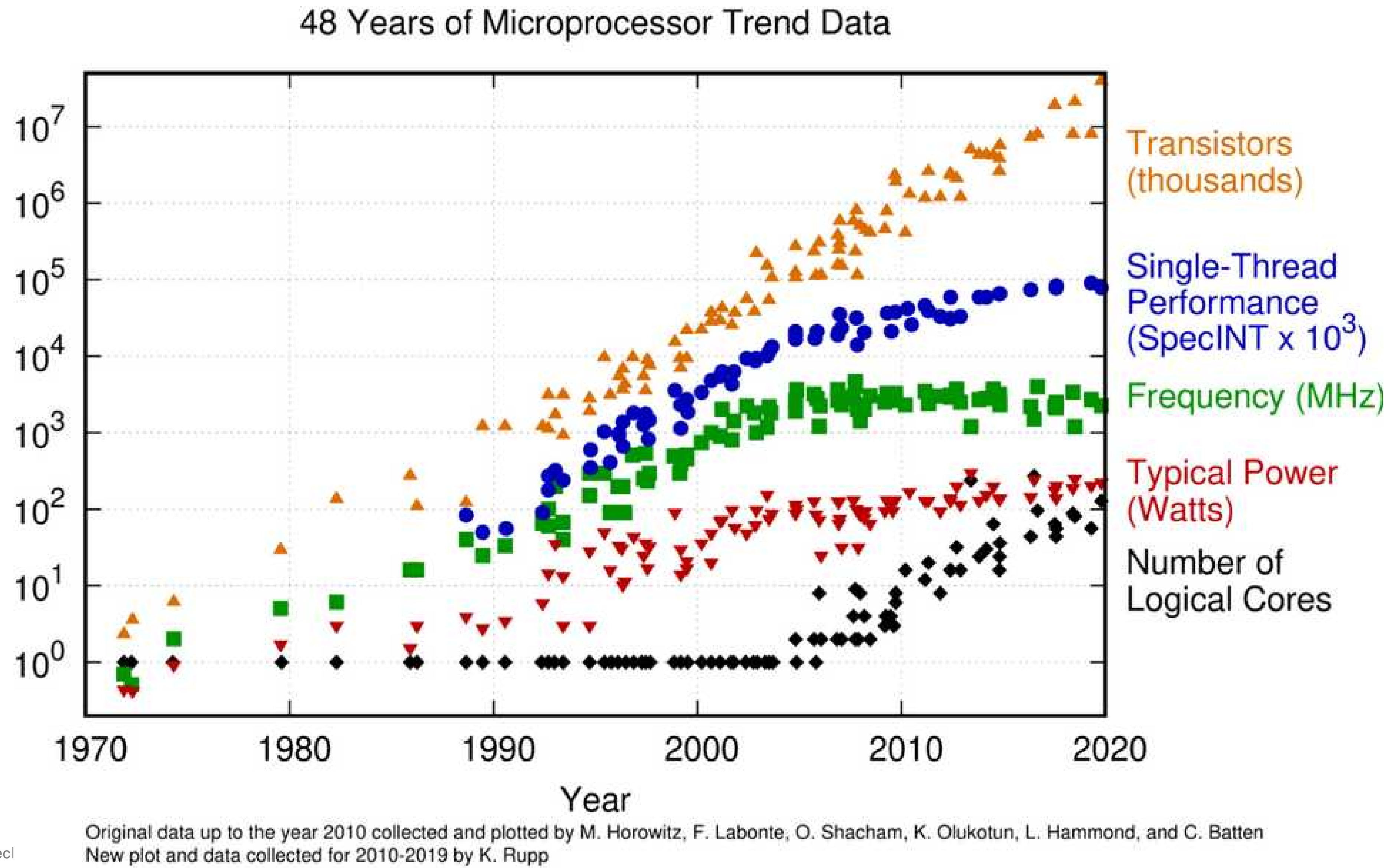


Analog architectures on FPGAs and ASICs

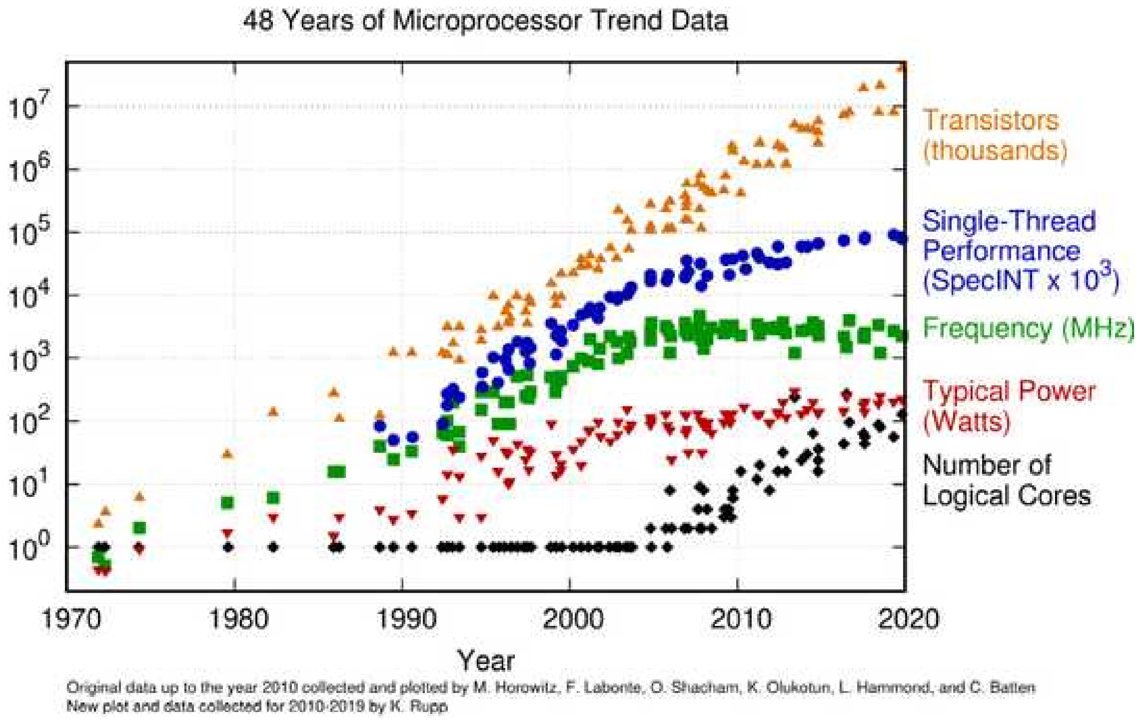
Dr. Sven Köppel
koeppel@anabrid.com

2025-07-14, 10:30 AM
Paderborn Center for Parallel Computing (PC2)
PC2 Seminar at U Paderborn
Room X0.101, Mersingweg 5, 33098 Paderborn

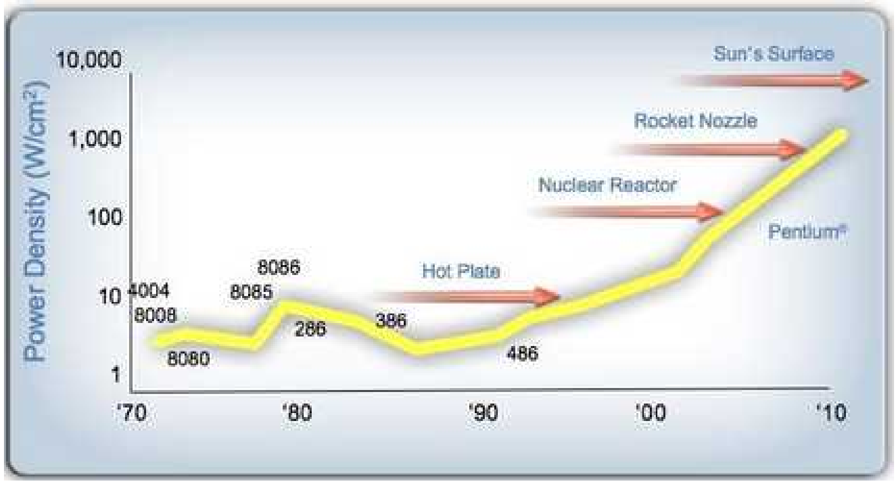


Bildquellen: wccftecl

Moore's Law: Advances only at Parallelization, no more at clock frequency

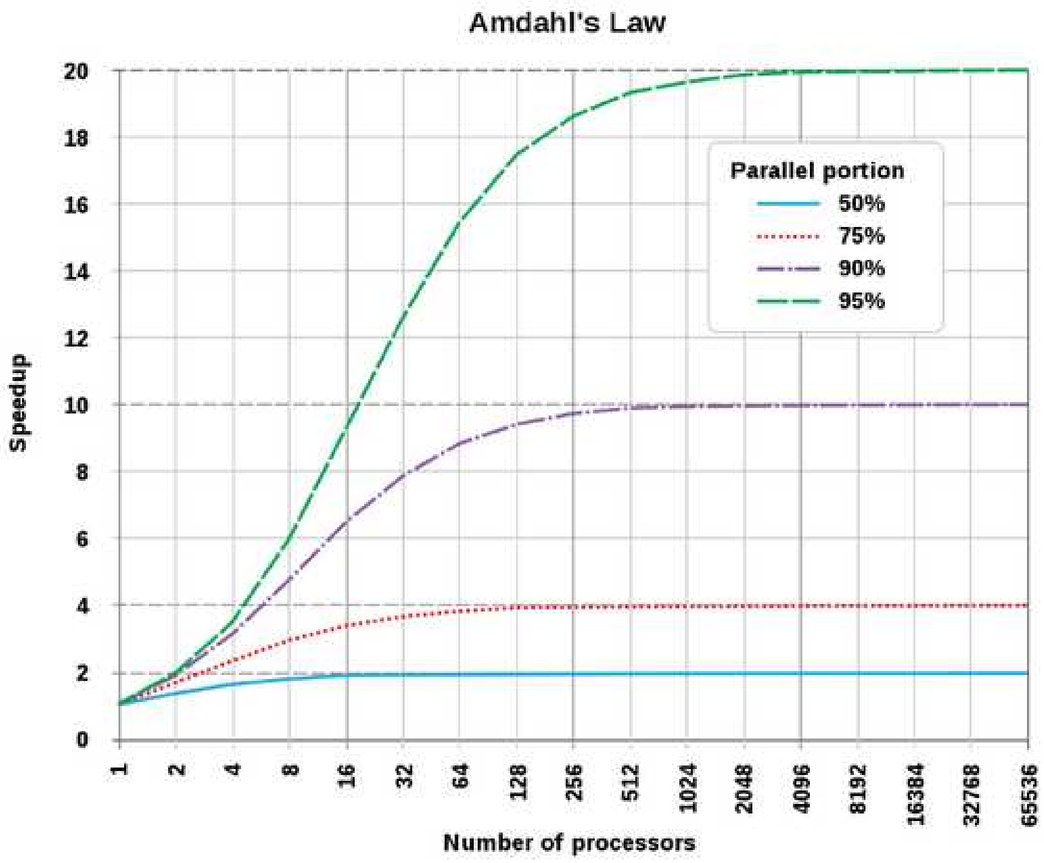


„Energy barrier"



Bildquellen: wccftech.com

Limits of Parallelizability: Amdahl's Law



Manycore architectures and high-performance clusters enable progress.

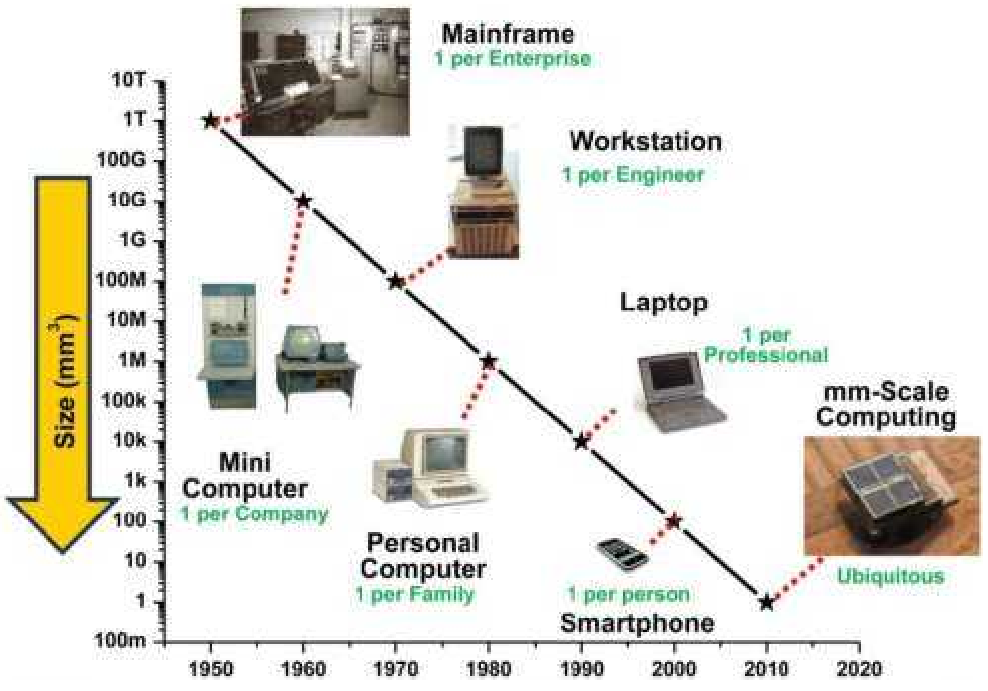
But the energy cost is very high!

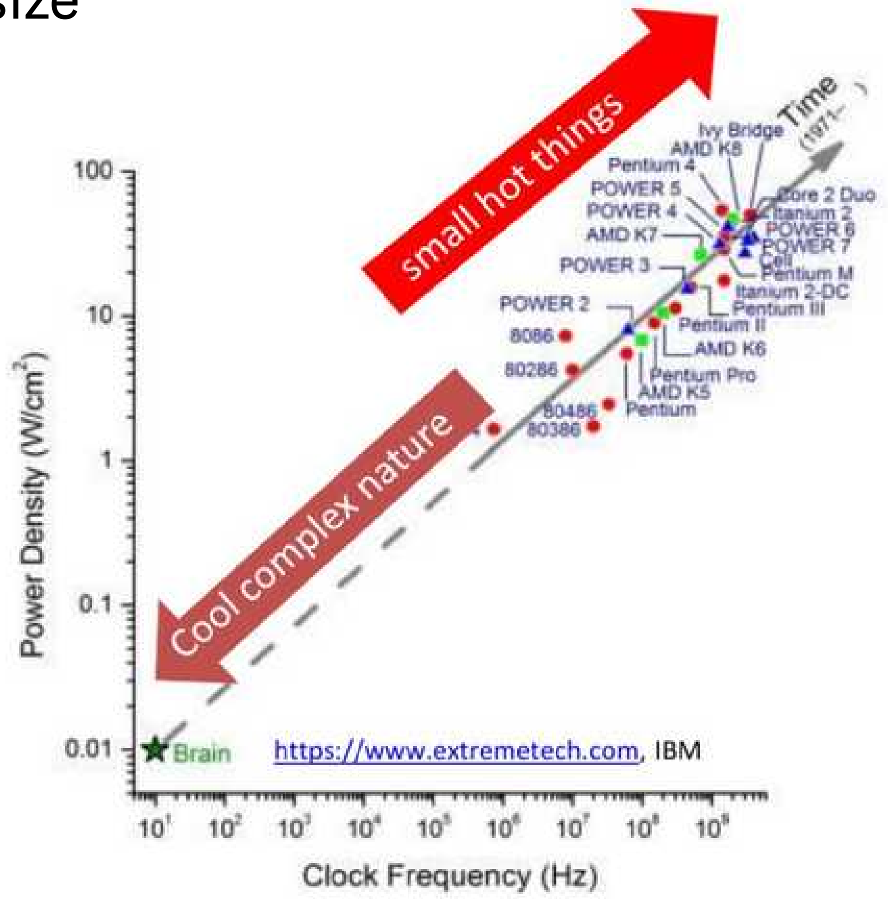
With 95% parallel portion:

3000 x computers (performance)
but only 20 x speedup

Bildquellen: Rebooting the IT-Revolution, Semiconductor Research Coorp 2015; extremetech.com; IBM; Special credits to Damian D. für die Slide und Abbildungen!

Heat generation: Energy and component size





Modern semiconductor manufacturing processes enable progress.
However, the energy cost is very high.

Bildquellen: Rebooting the IT-Revolution, Semiconductor Research Coorp 2015; extremetech.com; IBM; Special credits to Damian D. für die Slide und Abbildungen!

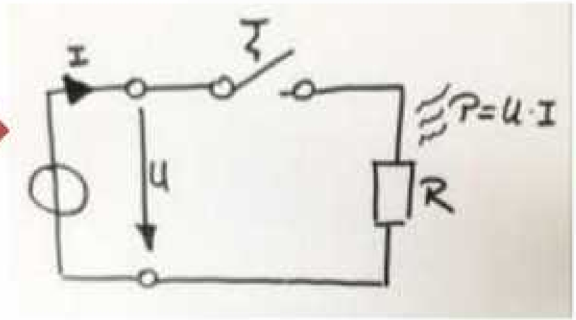
The final frontier for computer heat production

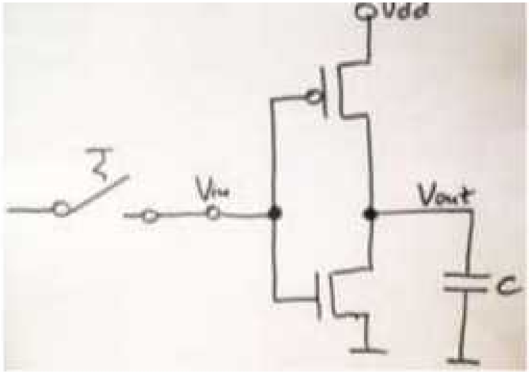
Landauer Principle! [R. Landauer 1961]

$$E_{\text{Gesamt}} = E_{\text{Signalverarbeitung}} + E_{\text{Wärme}}$$

Landauer Limit (at room temperature):

$$W \sim 3 \times 10^{-21} \text{Joule/bit}$$

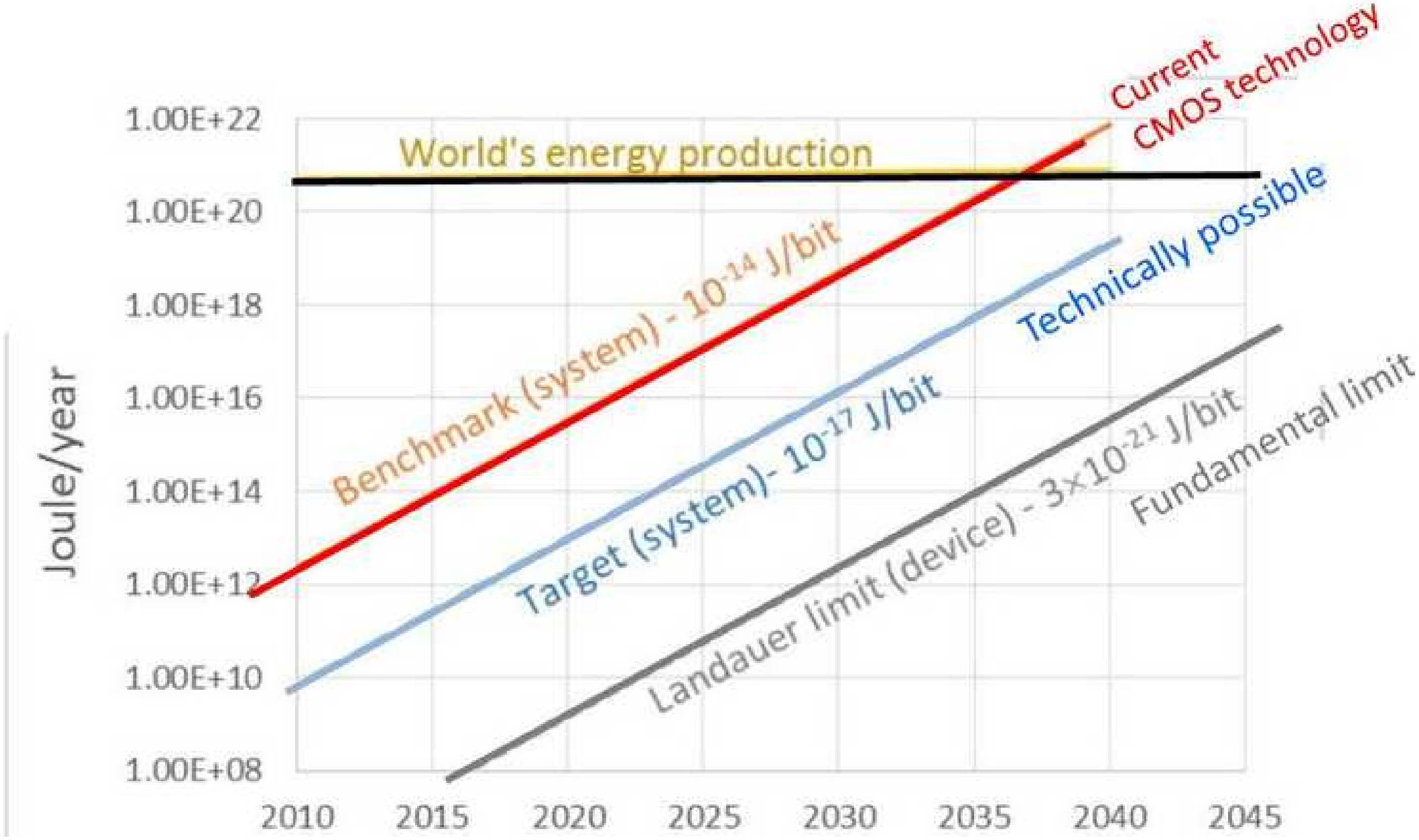




Heat dissipation in CMOS circuits increases significantly with the integration of components.

Bildquellen: mann.tv (iPhone verbrennt Mutter), wccftech.com; Special credits to Damian D. für die Slide und Abbildungen!

Switching generates Heat: A globally significant amount of it!



„Information catastrophe“

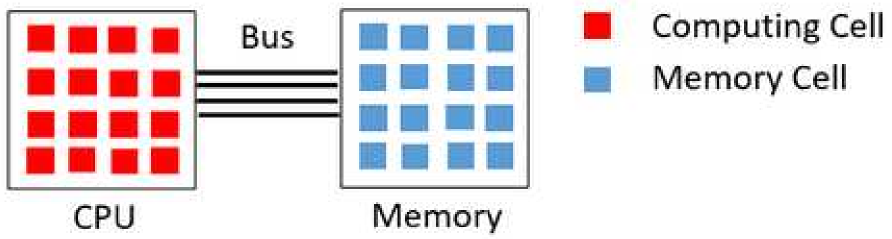
Already today, **7% of global energy** is consumed by the information sector.

Climate change?

Bildquellen: semiconductors.org; Special credits to Damian D. für die Slide und Abbildungen!

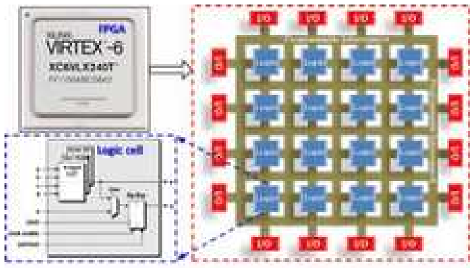
Digital architectures: Summary

- Energy barrier: Clock speed can no longer increase.
- Amdahl's Law: Parallelizability has its limits.
- Von Neumann bottleneck: Conventional Computer architecture limits performance.



Digital Alternatives exist, for instance

GPGPUs

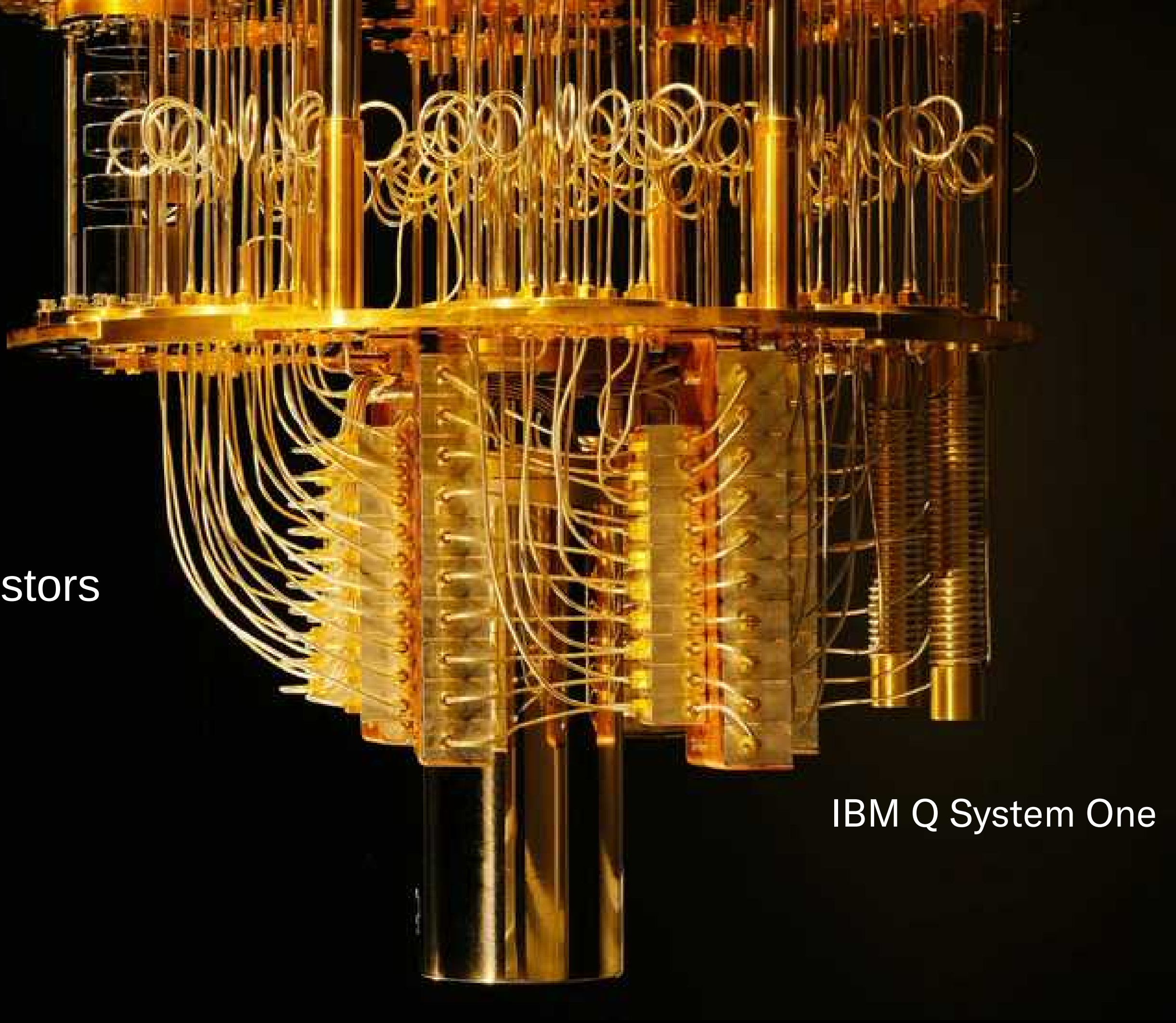


FPGAs

Exotic Computing:
Fundamental research

- Silicon alternatives, for instance Graphene
- Photonic Computing
- Quantum Computing
- Neuromorphic Computing
- Computational Memory, e.g. Memristors
- Analog Computing

Extensive materials research
Many interdisciplinary connections

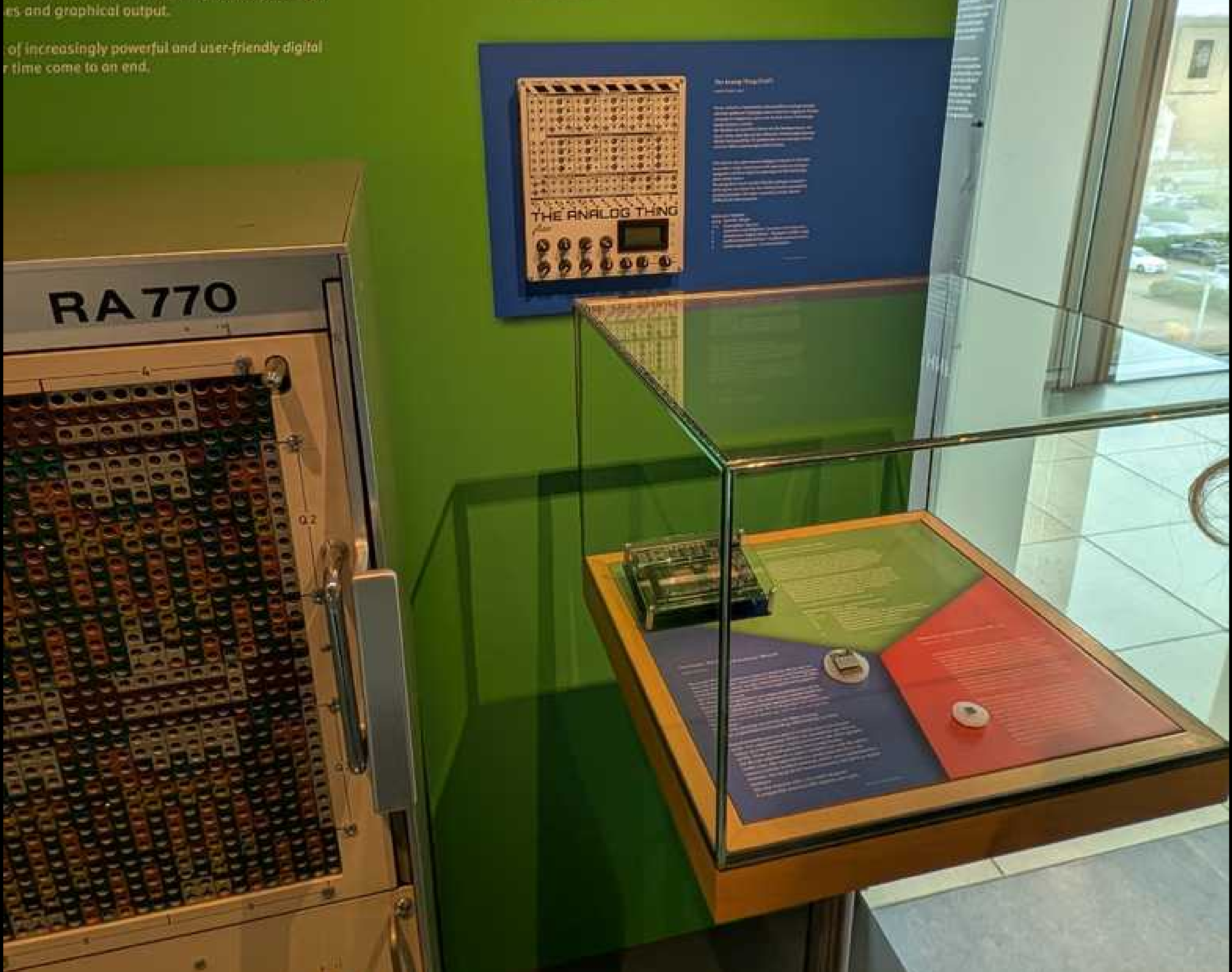


IBM Q System One

Heinz-Nixdorf-Forum (Computer Museum) in Paderborn

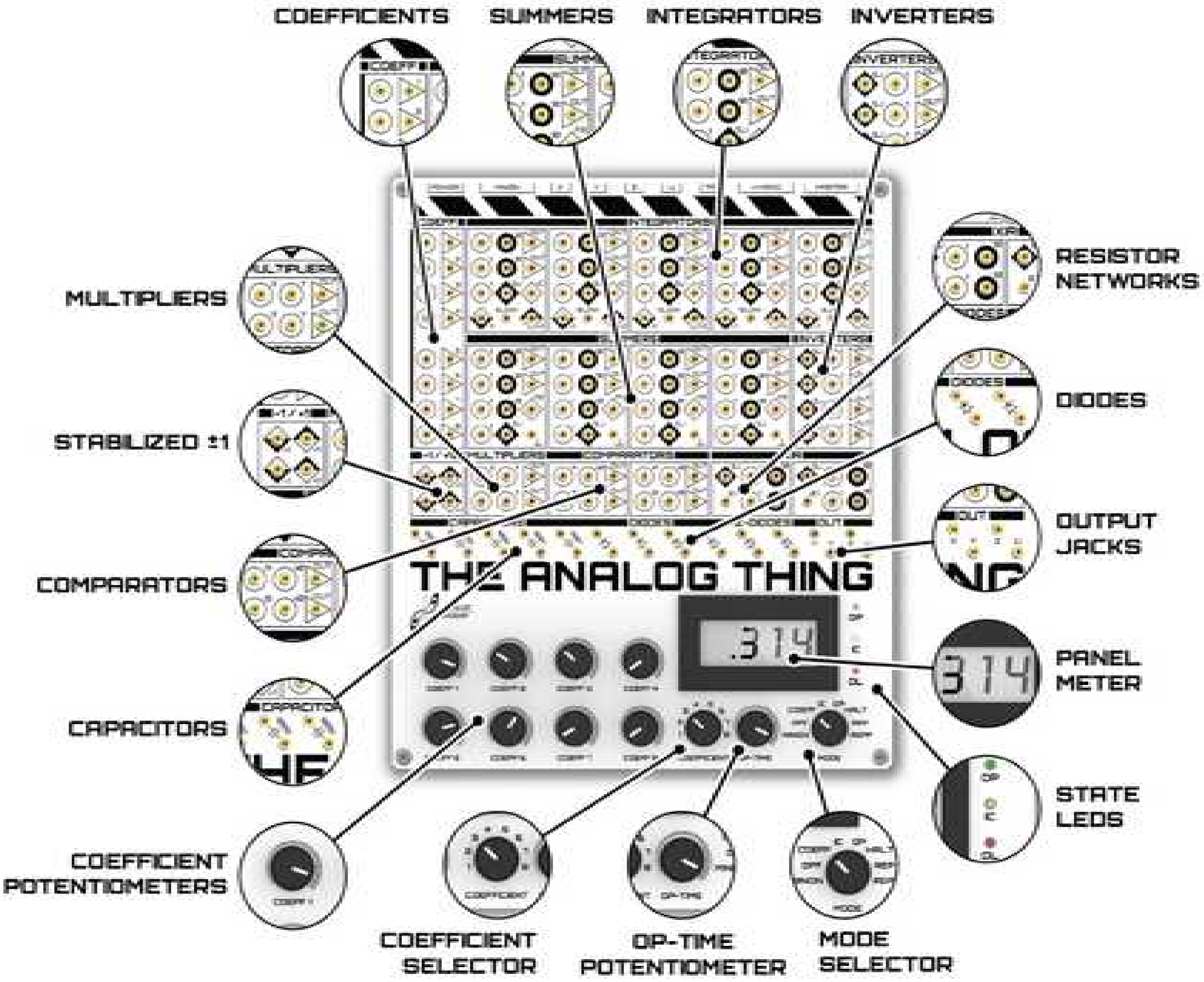
Telefunken Analog Computer from 1960s → www.analogcomputermuseum.org


ANALOGCOMPUTER
ANALOGUE COMPUTERS

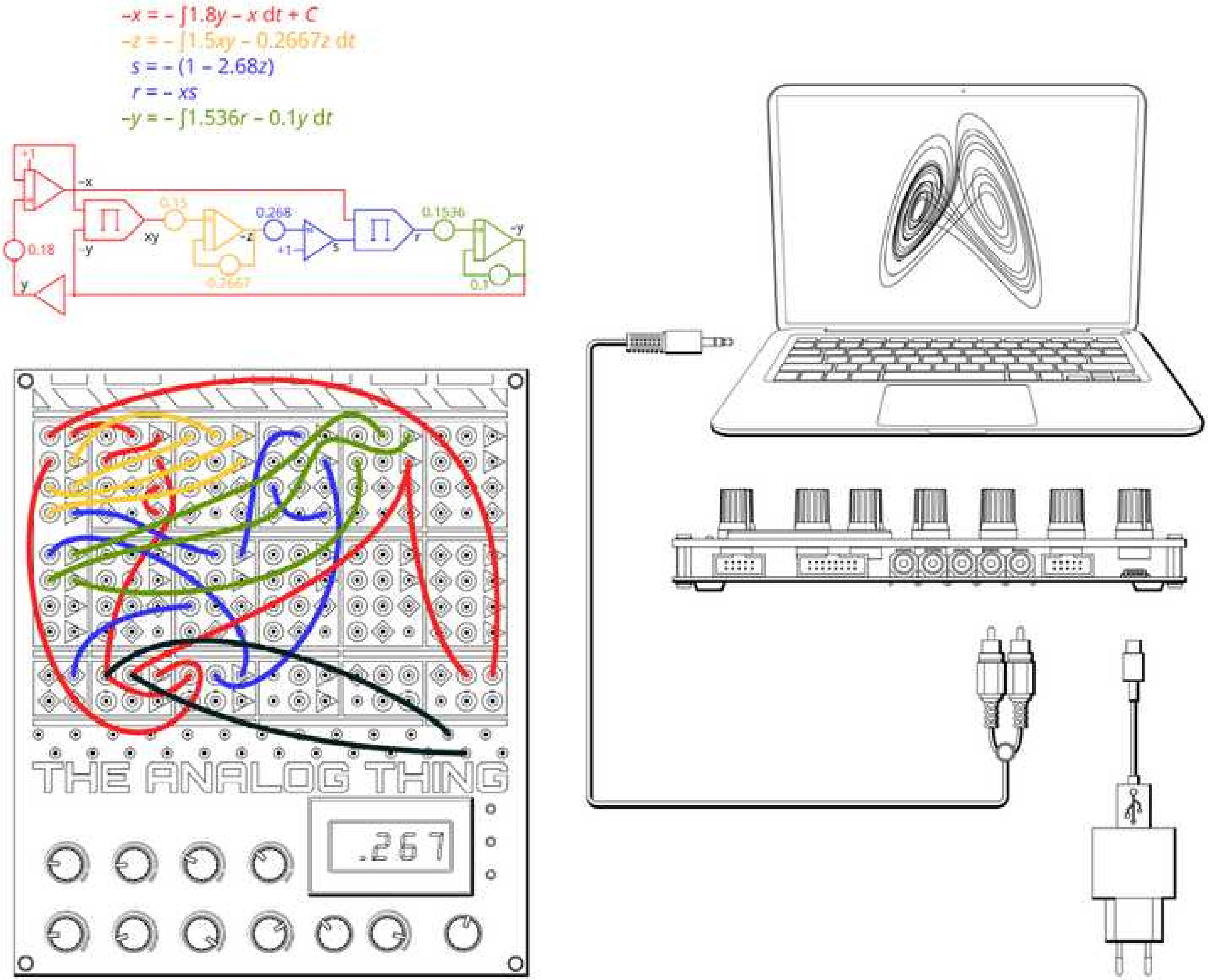

es and graphical output.
of increasingly powerful and user-friendly digital
r time come to an end.
THE ANALOG THING
RA 770

The Analog Thing

- Affordable computer for education and teaching.
- Open Source Hardware invites tinkering and „hacking“.
- Discrete CMOS and no processor, only a digital state machine.
- 3V3 A&D interface for Arduino, Raspberry Pi, etc.
- No electronic lab equipment neccessary (can use sound card instead oscilloscope)



(picture from our website)



(picture from our manual)

THE ANALOG THING

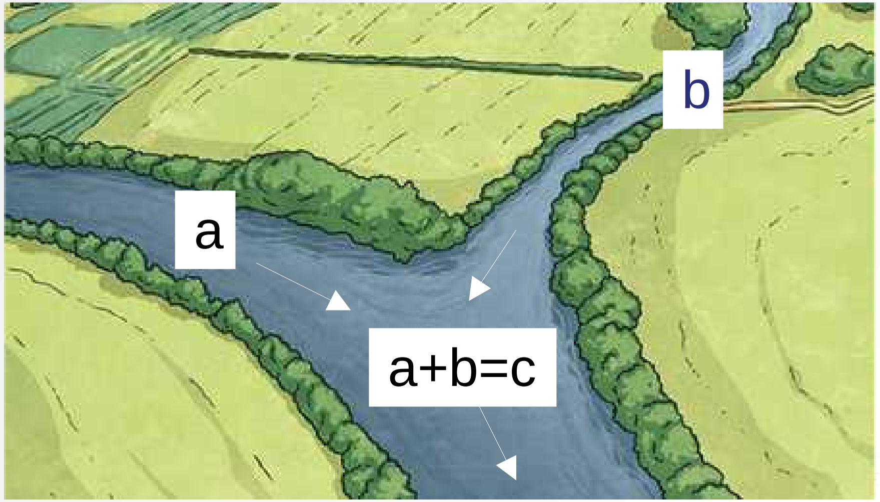
anabrid

Analogues

Digital Arithmetics

4	7	3	·	1	7
		4	7	3	0
	+	$_1$ 3	3	1	1
		8	0	4	1

Analog processes in nature

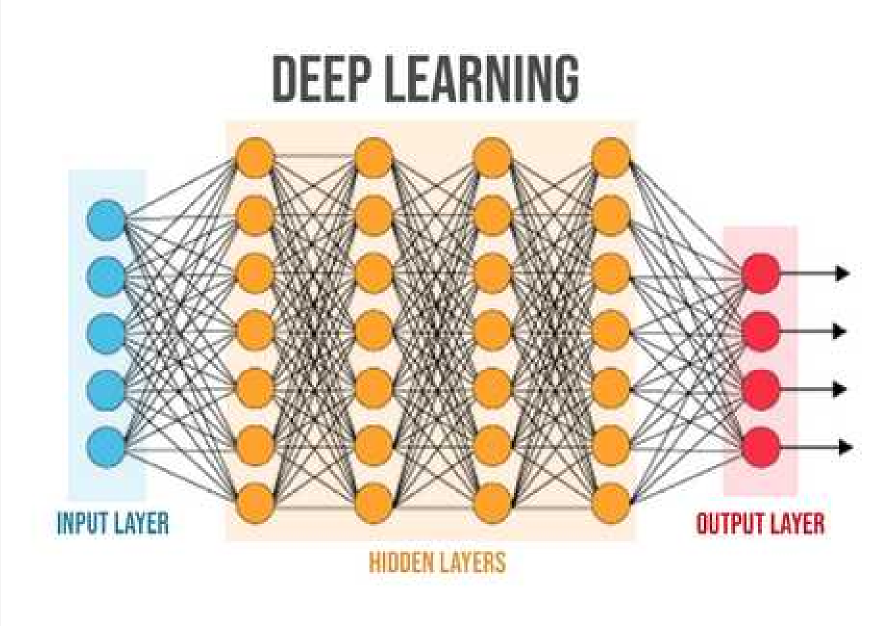


- Uses algorithms → Happens in real-time
- Inherently clocked and serial → Continous and parallel
- Based on memory → Initial state preparation and Measurement-based
- Energy barrier, von-Neumann bottleneck → **Resetting Moore’s law**

anabrid

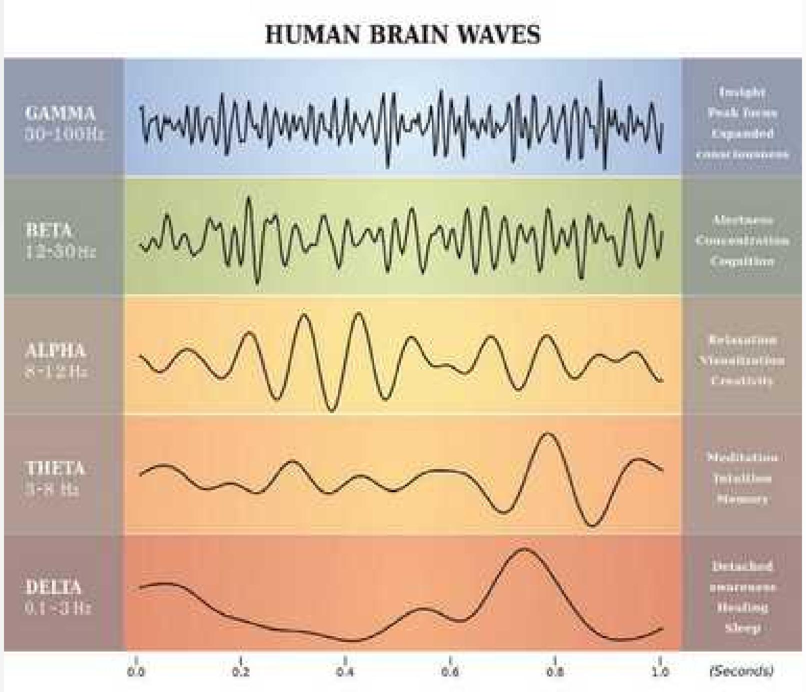
Compute as the Human Brain

Digital Arithmetics

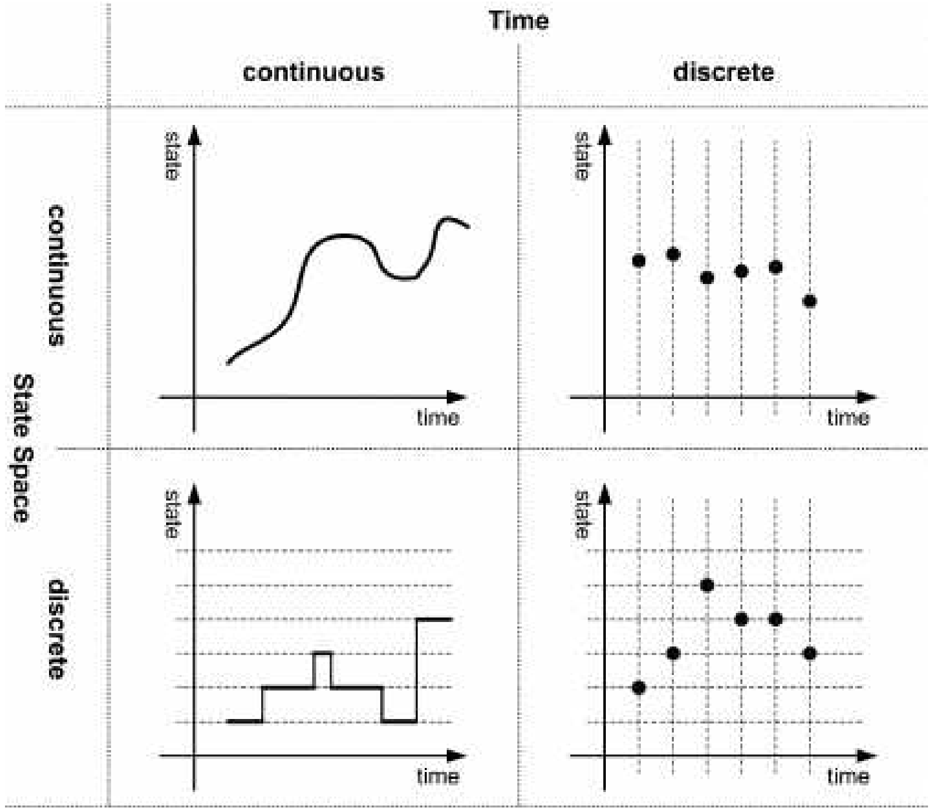


Symbolic computing & sparse linear algebra

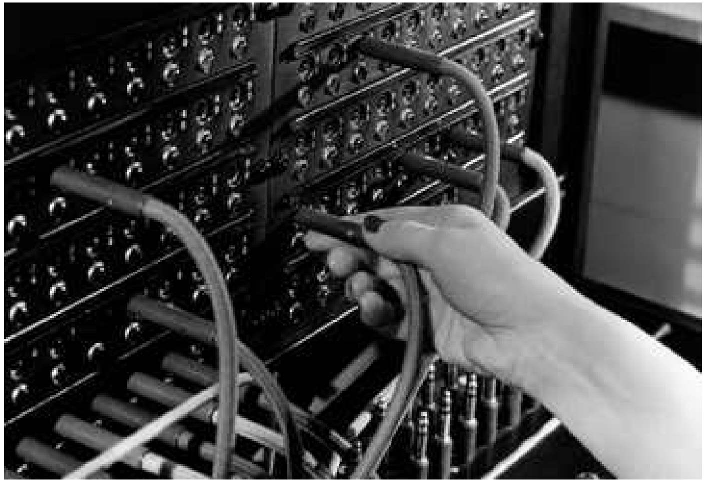
Analog processes in nature



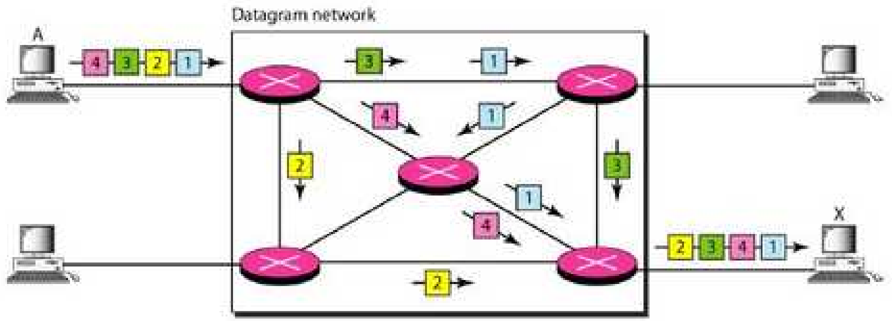
Wave-based computing & dense interconnects



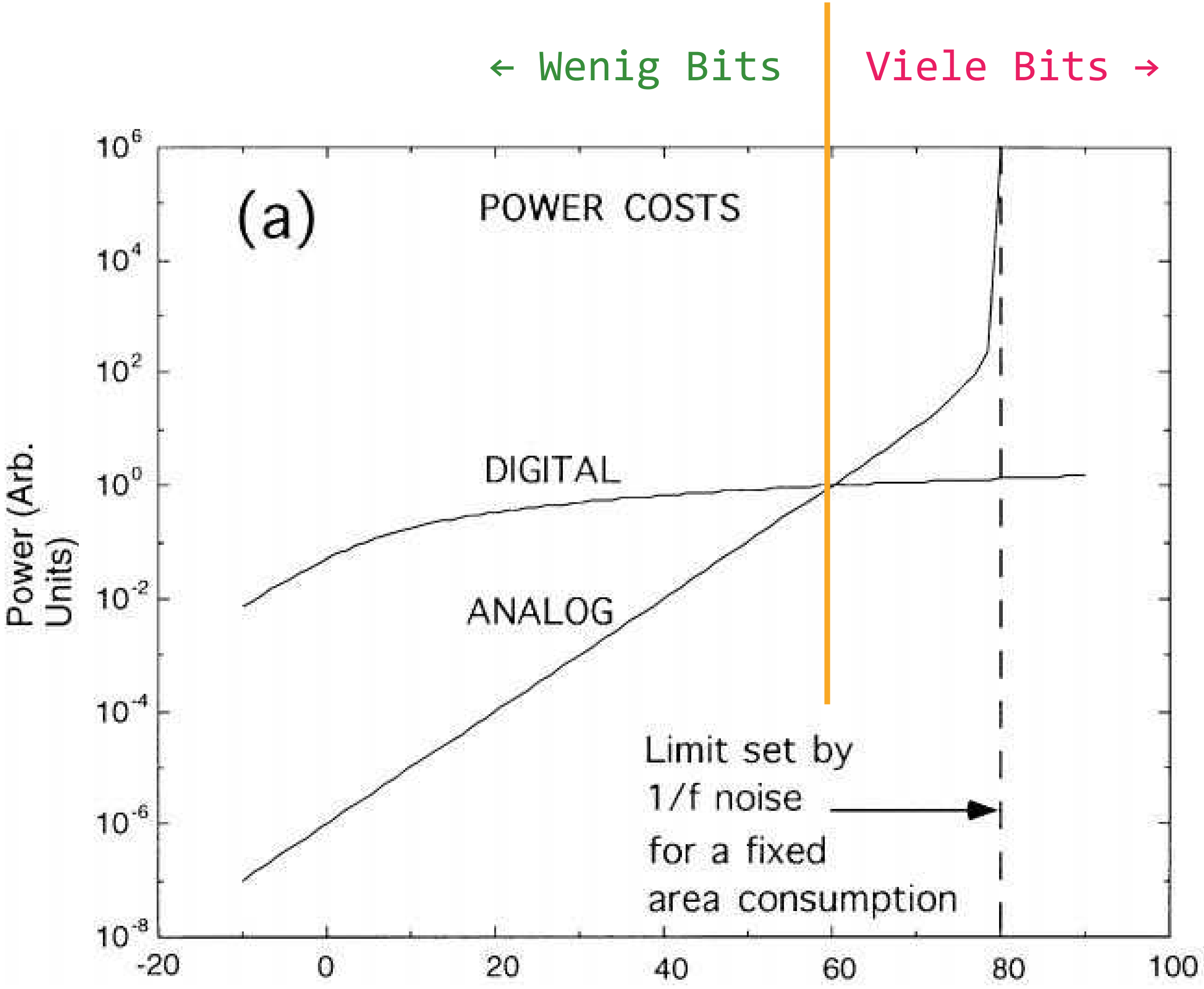
[10.14279/tuj.eceasst.27.385]



Telephone switchboard

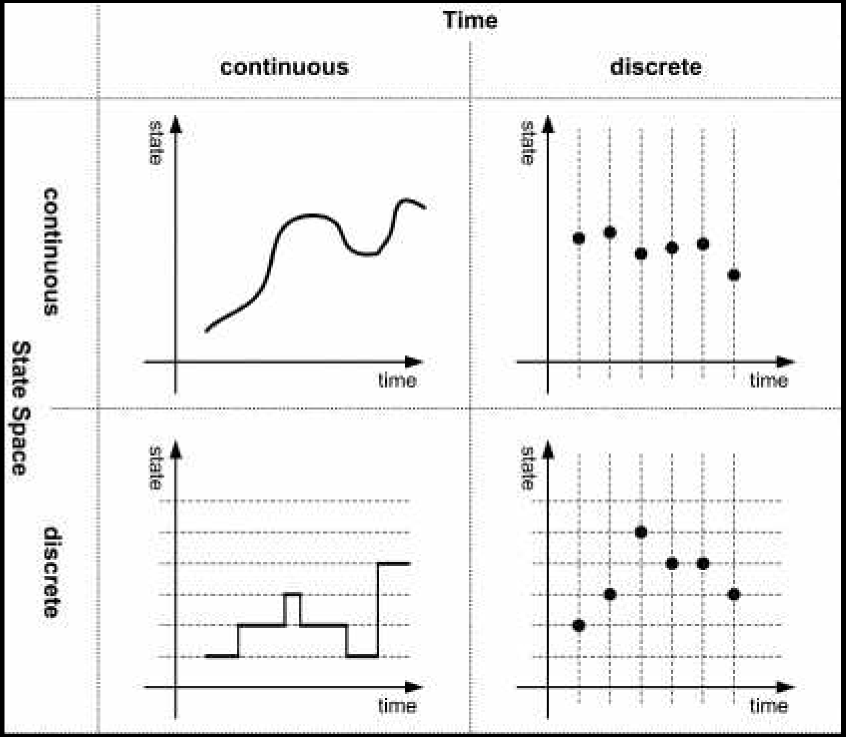


Ethernet switch

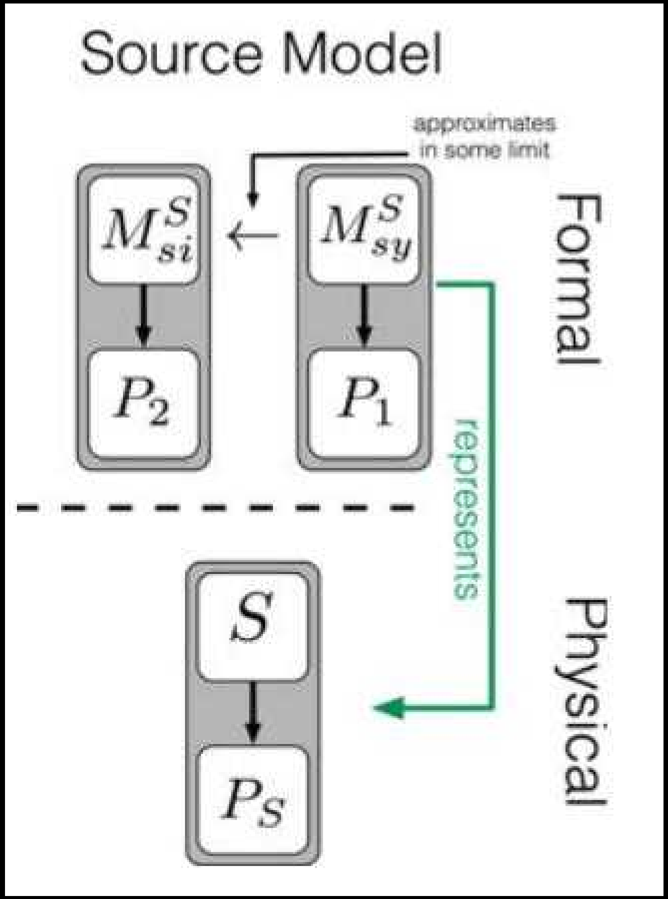


Sarpeshkar, Neural Computation 10, 1601–1638 (1998)

analog vs **analogue**



[10.14279/tuj.eceasst.27.385]



[Hangleiter, Carolan, Thébault: Analogue Quantum Simulation, p.18]

Emulation of Hawking Radiation in Dispersive Optical Media

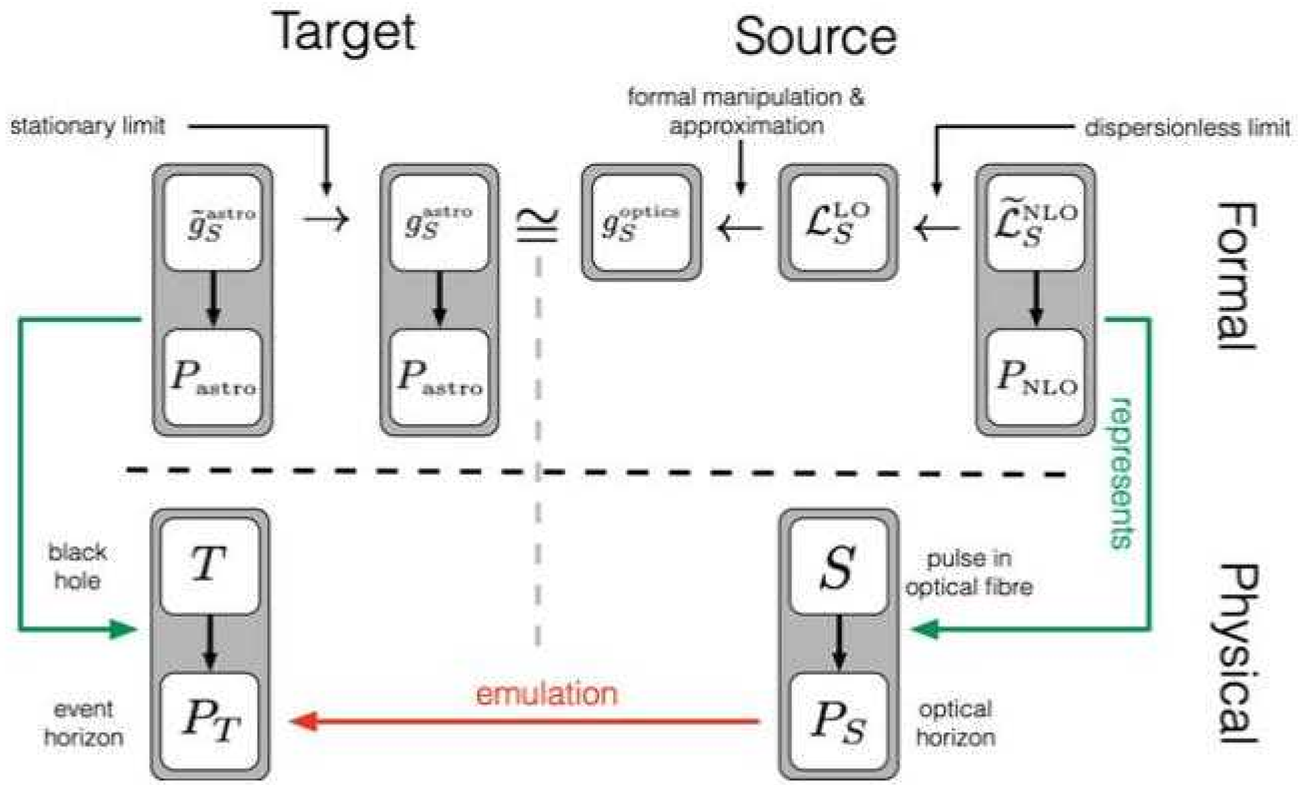
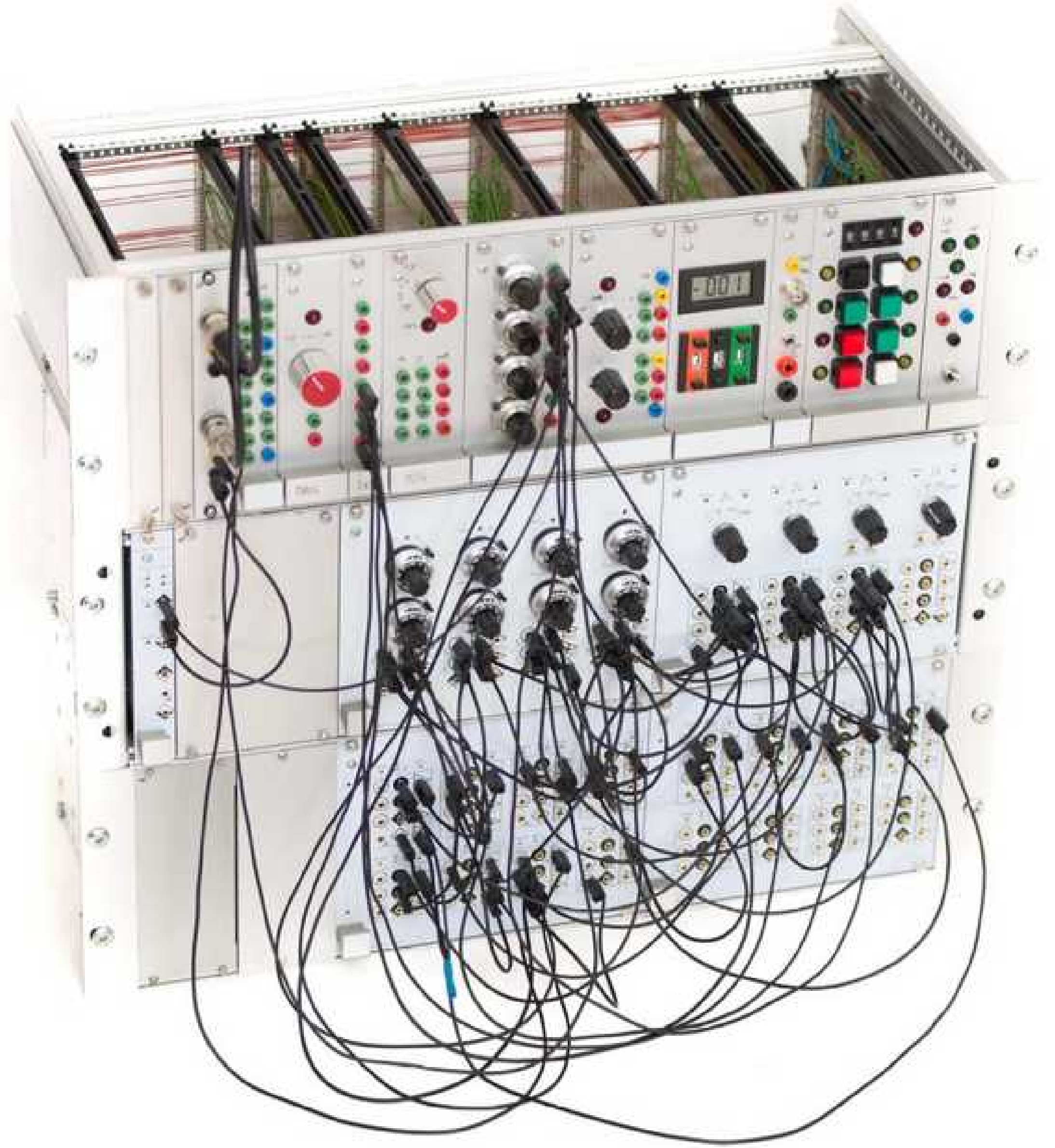


Fig. 5.4 Schema for analogue emulation of Hawking radiation case study (see main text for figure explanation)

[Hangleiter, Carolan, Thébault: Analogue Quantum Simulation]

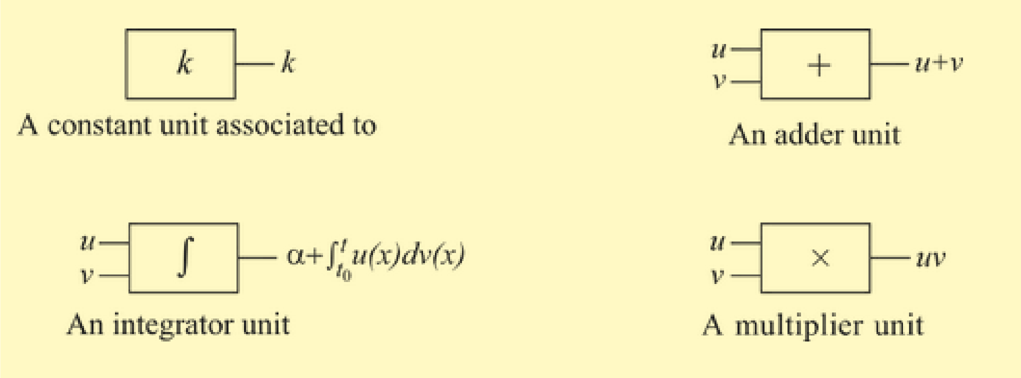


Die Rückkehr des Analogrechners

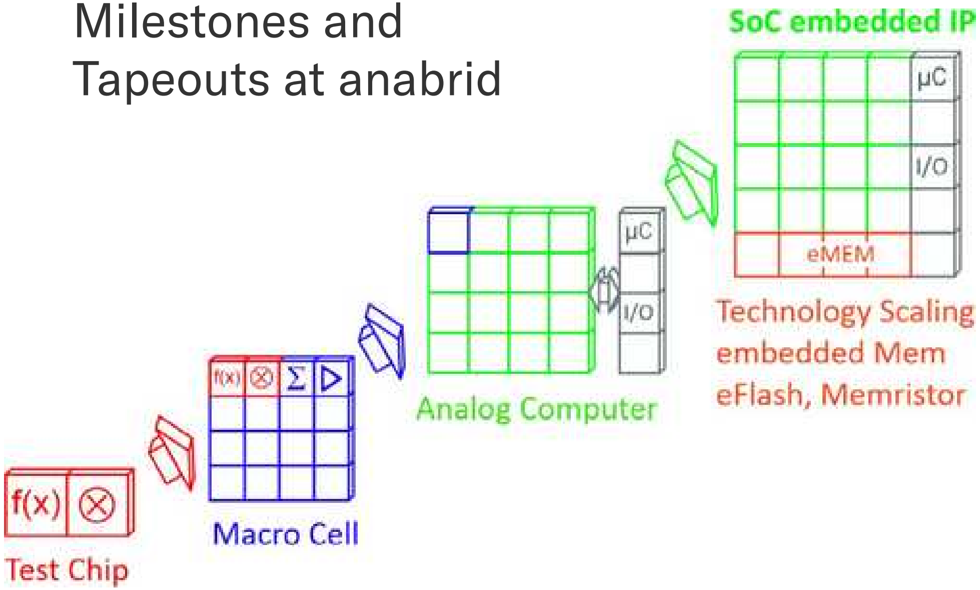
- Namensgebend ist die Analogiebildung mit elektrischen Schaltkreisen.
- Intrinsisch parallele Datenfluss-verarbeitung statt sequentieller Algorithmen.
- Extrem energieeffizient: „Jedes Elektron zählt“.
- Kontinuierliche Werte und kontinuierliche Zeit: Kein Takt.

Building blocks: The General Purpose Analog Computer (GPAC)

Computing model by Claude Shannon [1941]



Milestones and
Tapeouts at anabrid

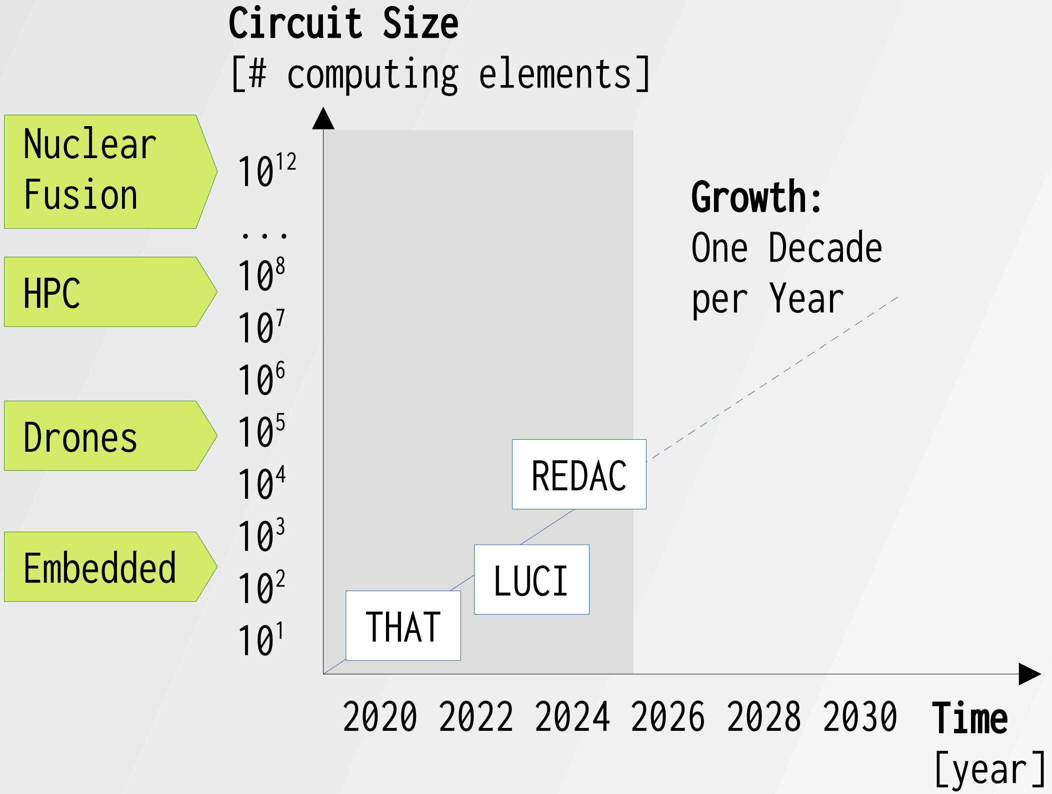


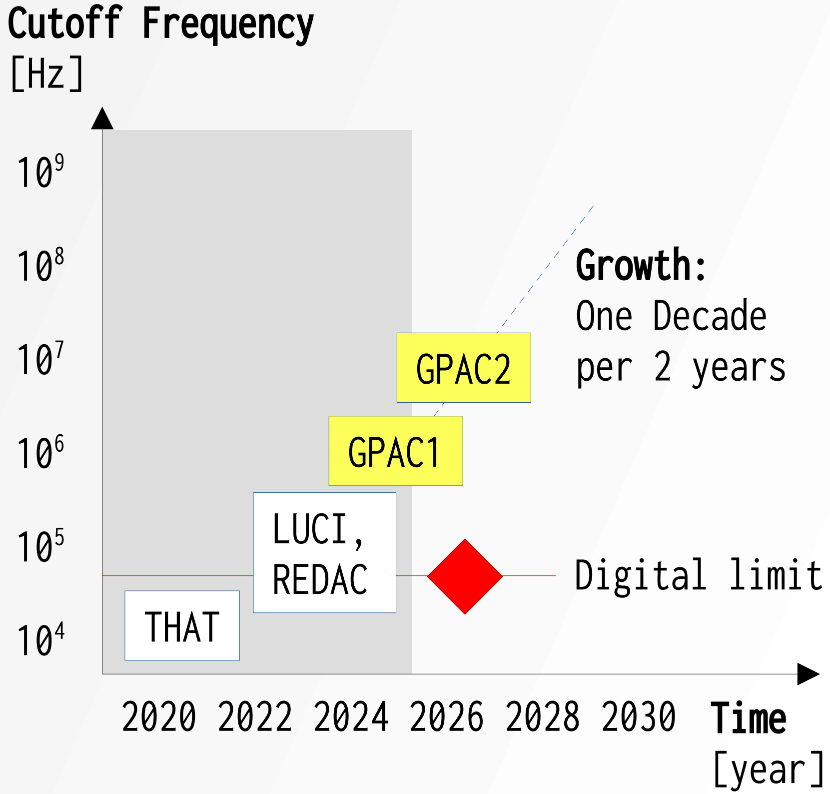
- Black box model
- No algorithm, no memory (beyond state variables)
- All computing elements compute at same time

Bildquellen: Daniel Graca http://www.dp-pmi.org/uploads/3/8/1/3/3813936/shannon_2016.pdf

Anabrid resets Moore's law

Scaling analog CMOS for HPC

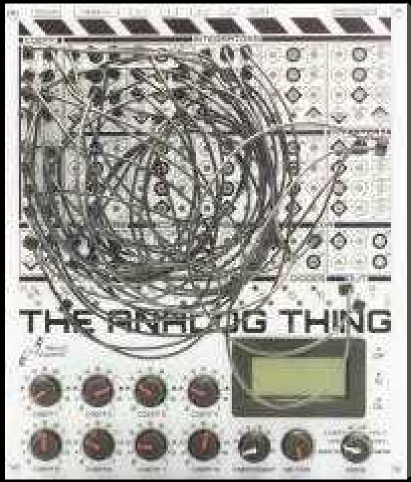




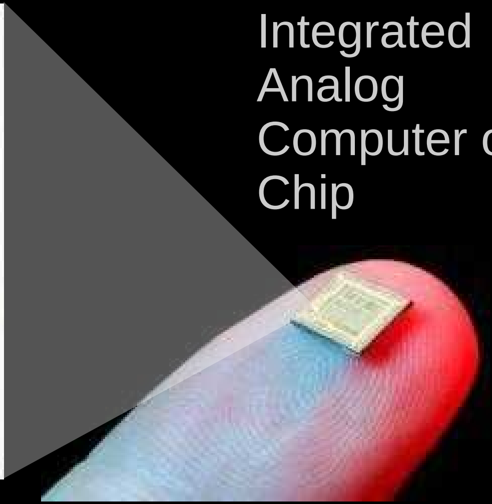
Power and Time to Solution Measurements and Projections for the GPAC

Integrated Digital benchmark

Discrete Analog Computer



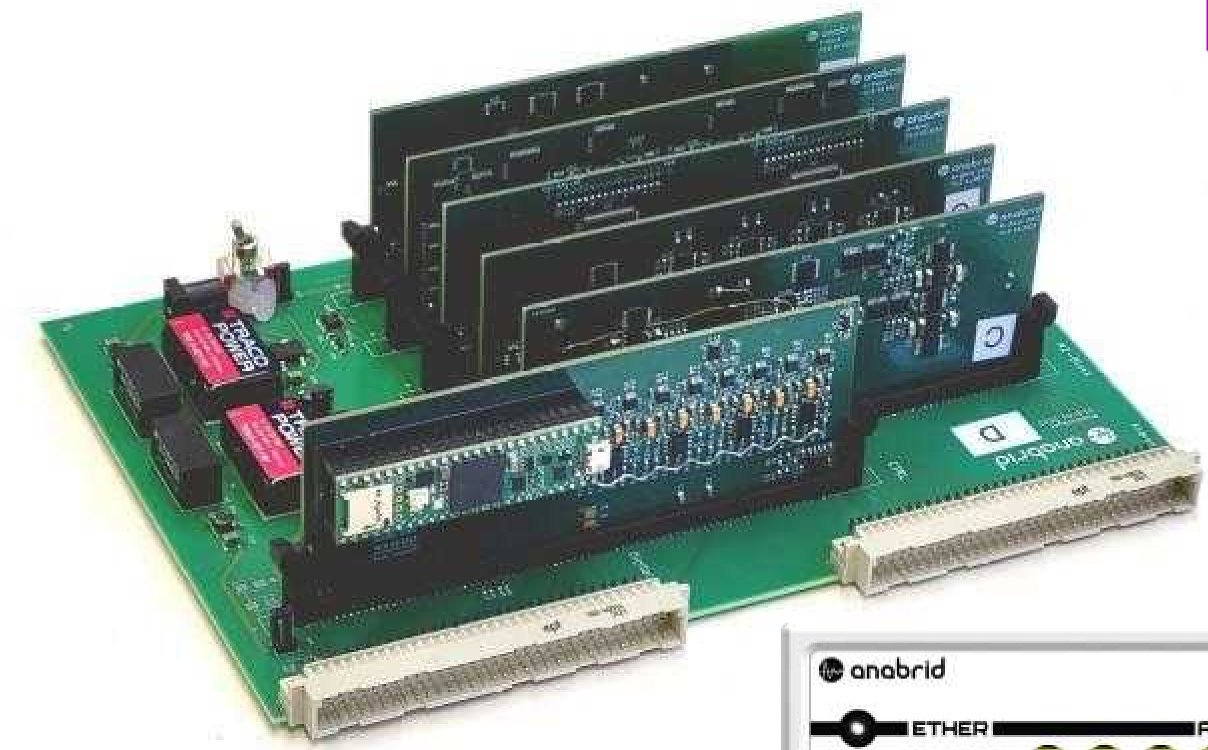
Integrated Analog Computer on Chip

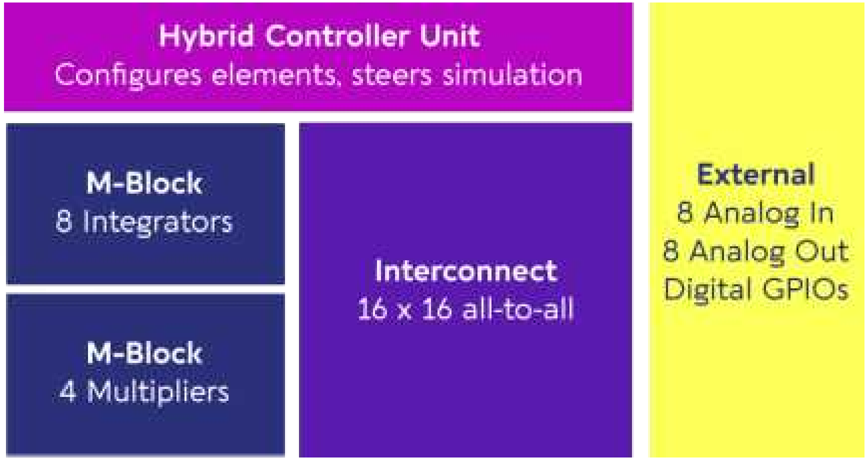


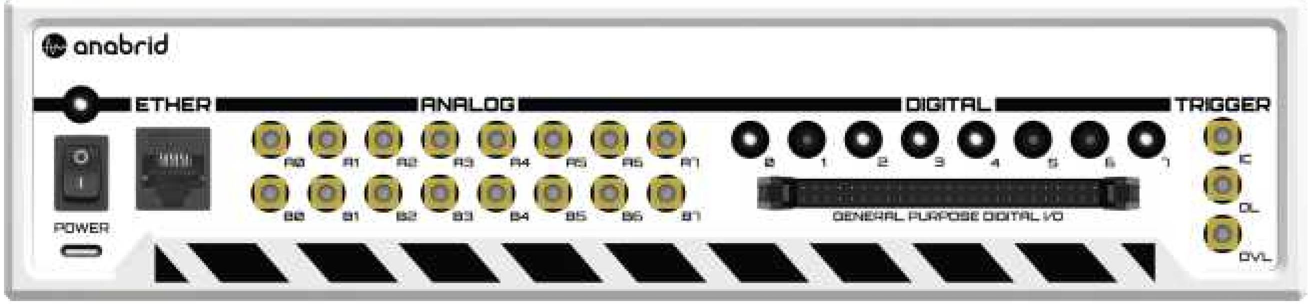
		Measured		Projected
		Digital	Analog (MI)	Analog Chip
Time to Solution:	T [μs]	75 ± 45	100	$10^{-(0.5\pm0.5)}$
Cutoff frequency (bandwith):	$k_0 \sim 1/\Delta t$ [Hz]	3×10^4	10^4	$10^{6.5\pm0.5}$
Power consumption:	P [W]	10	0.4	10^{-2}
Energy for computation:	$E = P \cdot T$ [μJ]	900 ± 600	40	$10^{-(2.5\pm0.5)}$
(effective) FLOPs:	F [FLOP/sec]	10^9	$3 \times 10^{(4\pm1)}$	7×10^5
Efficiency (FLOPS per Energy):	F/E [FLOP/J]	10^8	$7.5 \times 10^{8\pm1}$	3×10^{11}

Köppel et al [arXiV:2102.07268]

A milestone last year: First fully-reconfigurable discrete analog computer







Precision
Up to 12bit resolution.
DC accuracy 0.1%
AC bandwidth > 1Mhz (cutoff frequency).

„Resetting Moore’s law“ at 65nm

anabrid

Towards 1e5 computing elements on chip

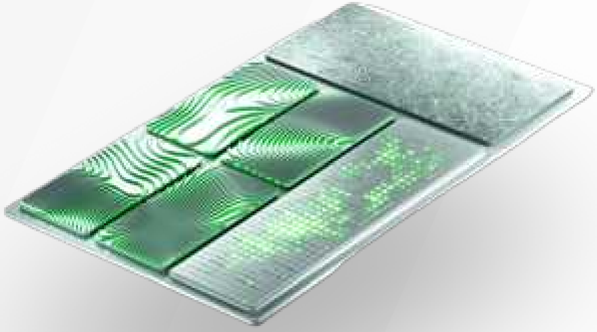
The Analog Thing
»Hacker’s toolkit«

SPRIN-D

REDAC Analog Supercomputer

DLR Deutsches Zentrum für Luft- und Raumfahrt German Aerospace Center

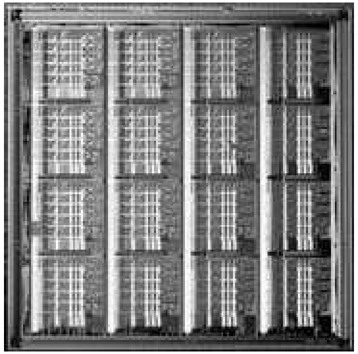
QCi DLR Quantencomputing Initiative



G-PAC: 65nm CMOS
2027 -> 22nm

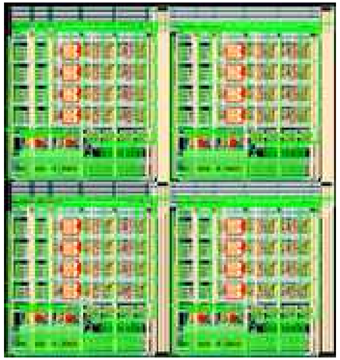
First Chip-level integrated Analog Computers

Existing: Highly Specialized Chips, low bandwith, rather large statical/dynamical errors, near to no software support



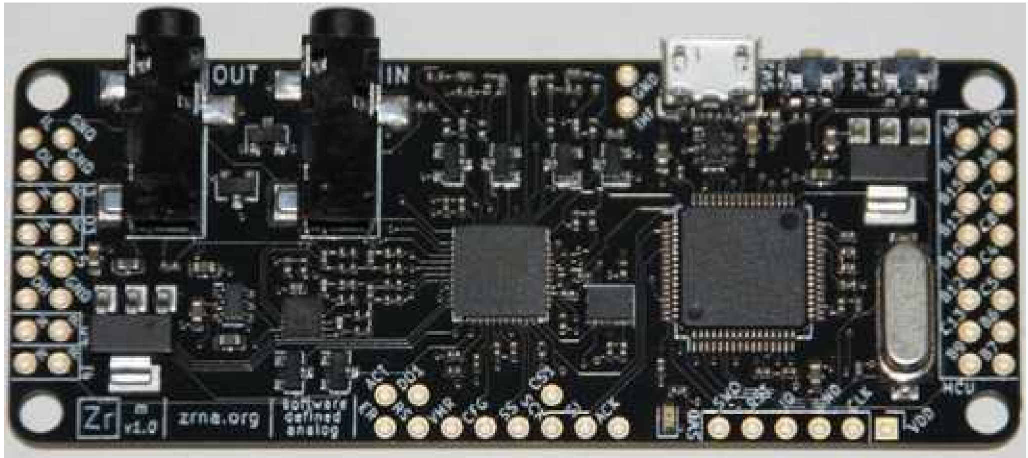
Glenn Cowan 2005

(A VLSI Analog Computer, PhD Thesis, Columbia University)



Ning Guo et al. 2016

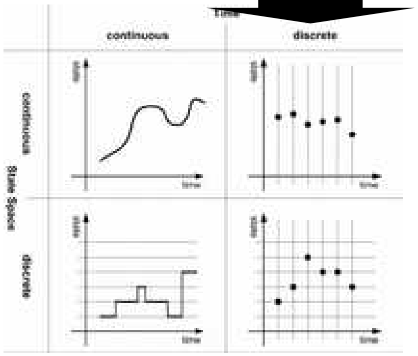
(Energy-Efficient Hybrid A/D Approx Comp. In Continous Time, IEEE Journal of Solid-State-Circuits)



Anadigm Inc. and ZRNA (open source)

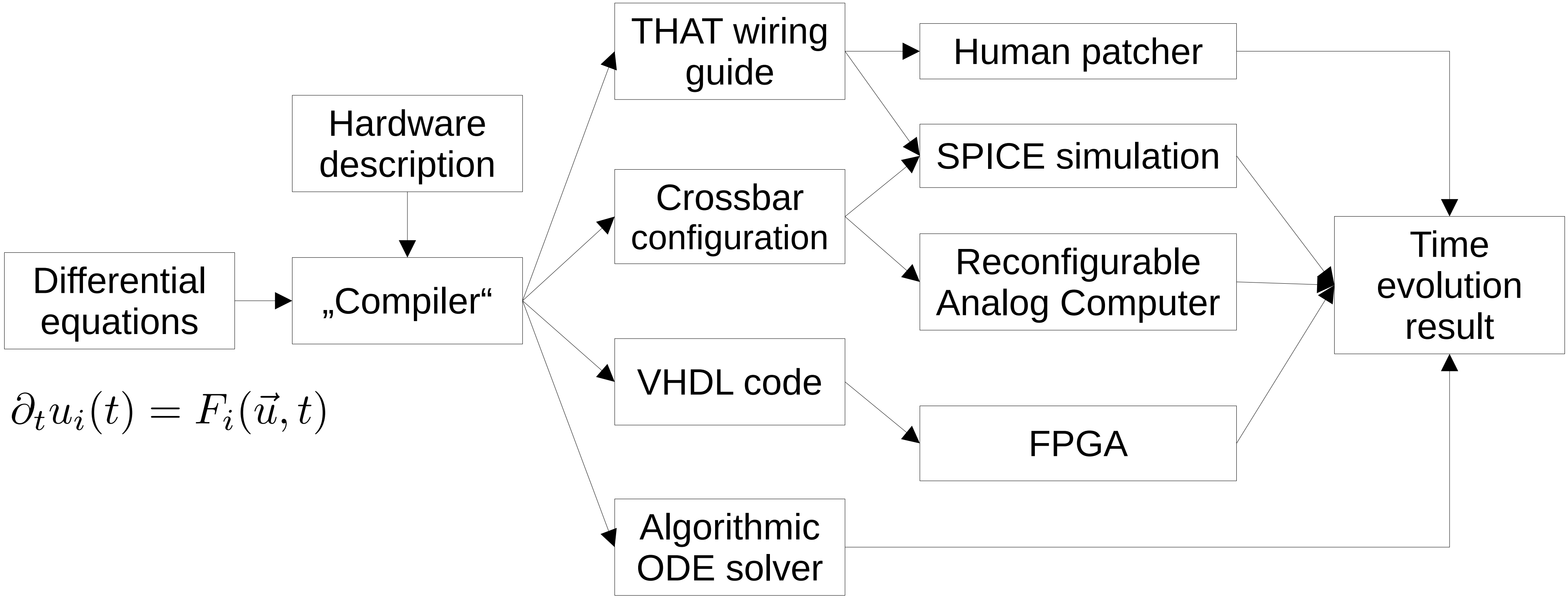
[http://zrna.org]

Switched capacitor configurable analog blocks (CABs)



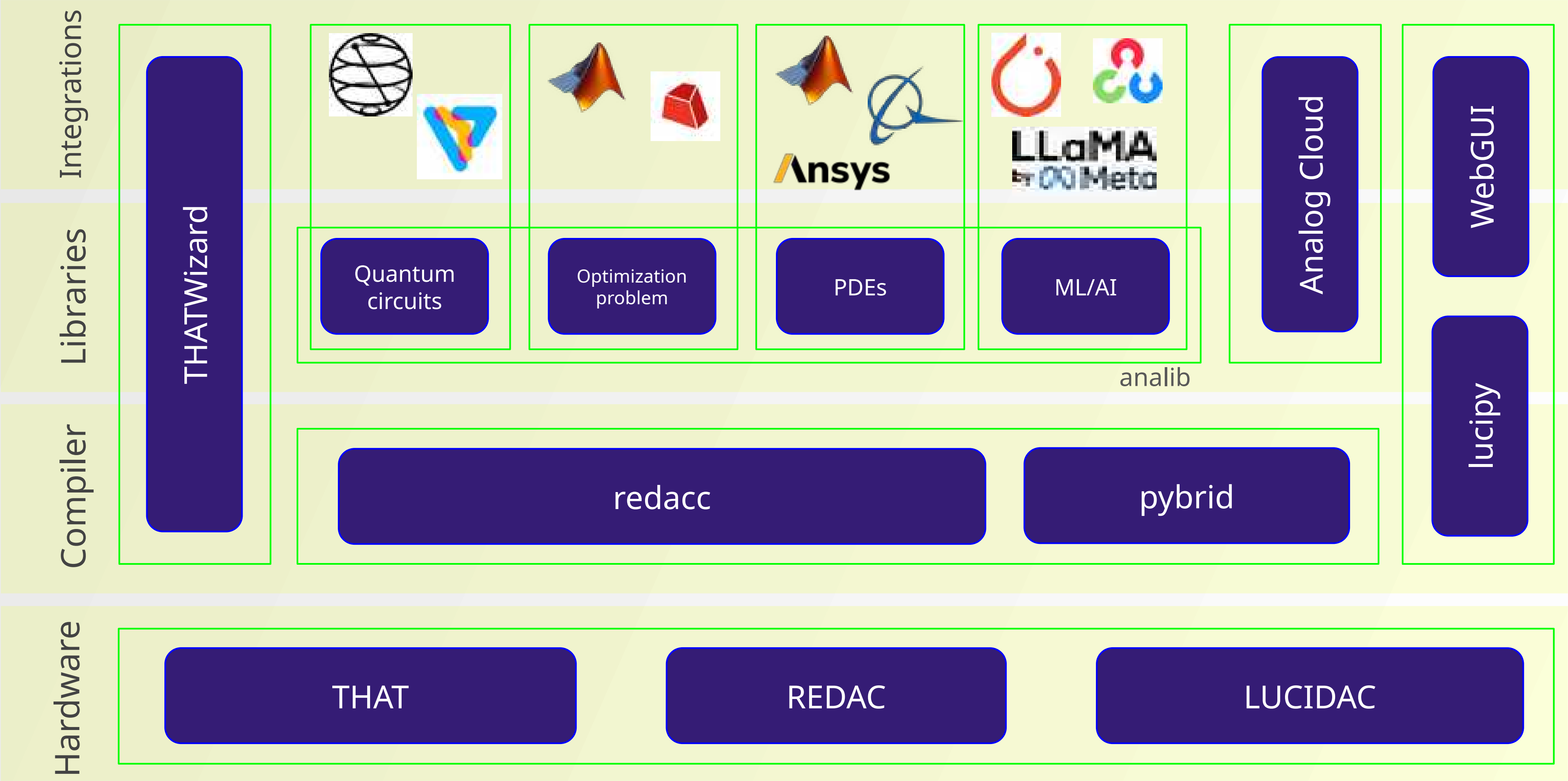
[10.14279/tuj.eceasst.27.385]

Analog Computing Toolchain as we built it today



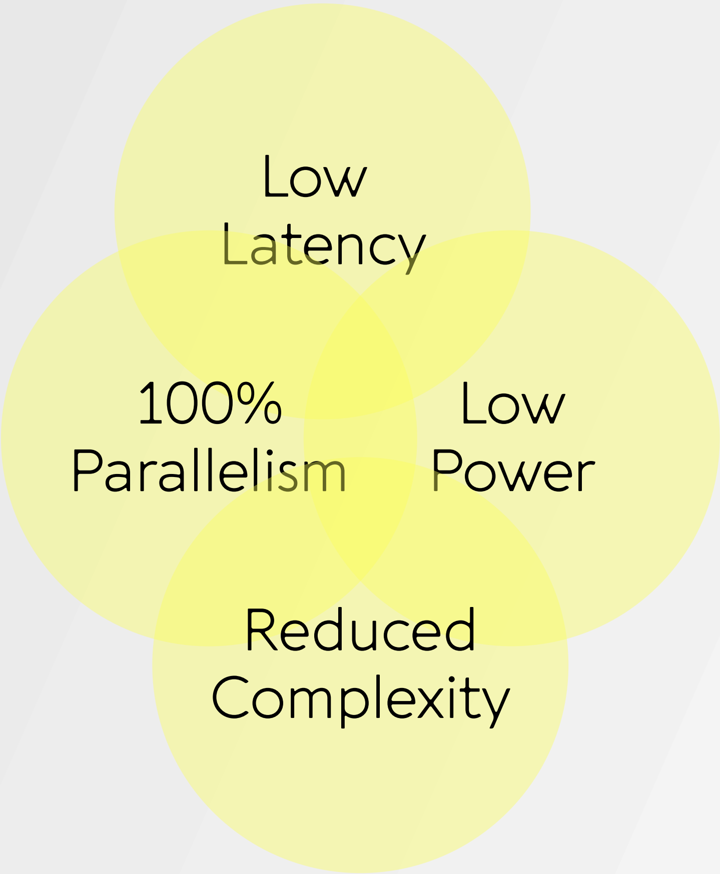
$$\partial_t u_i(t) = F_i(\vec{u}, t)$$

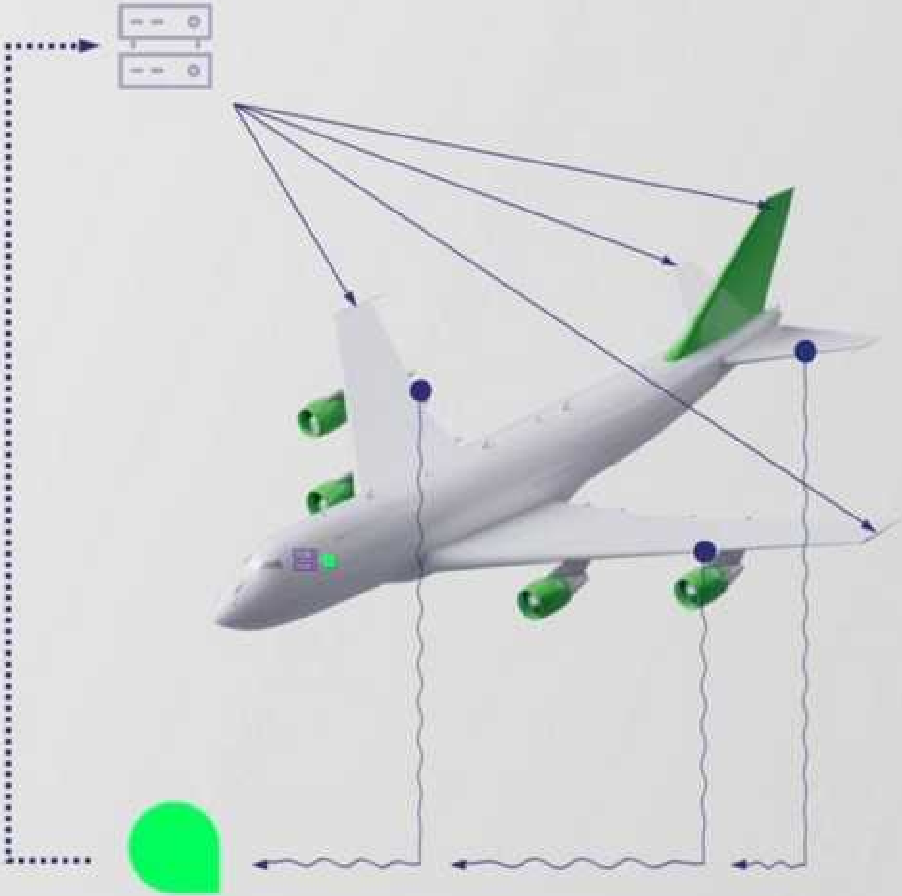
anabrid tech stack (industry audience)



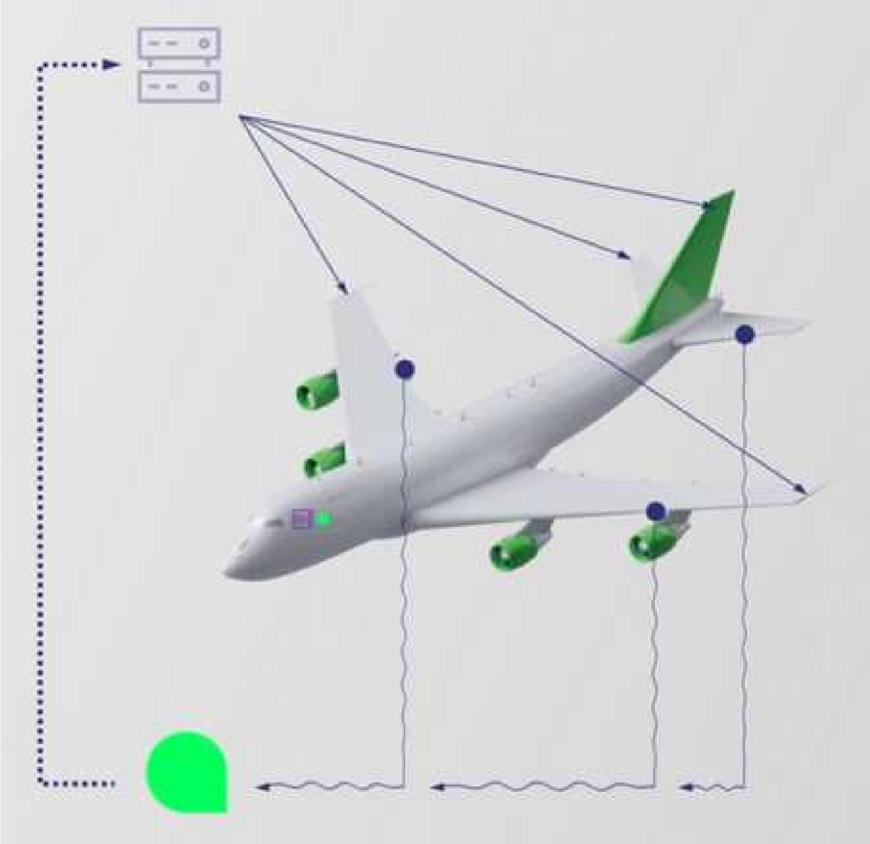
Talking to industry

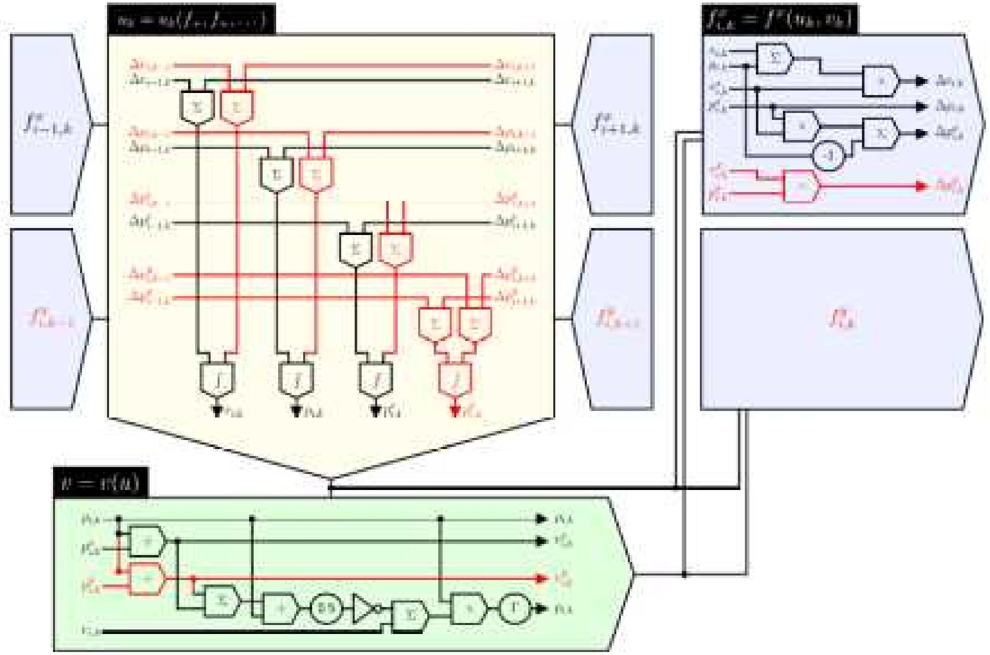
Reasons why to go analog





Solving industry problems with Analog CMOS





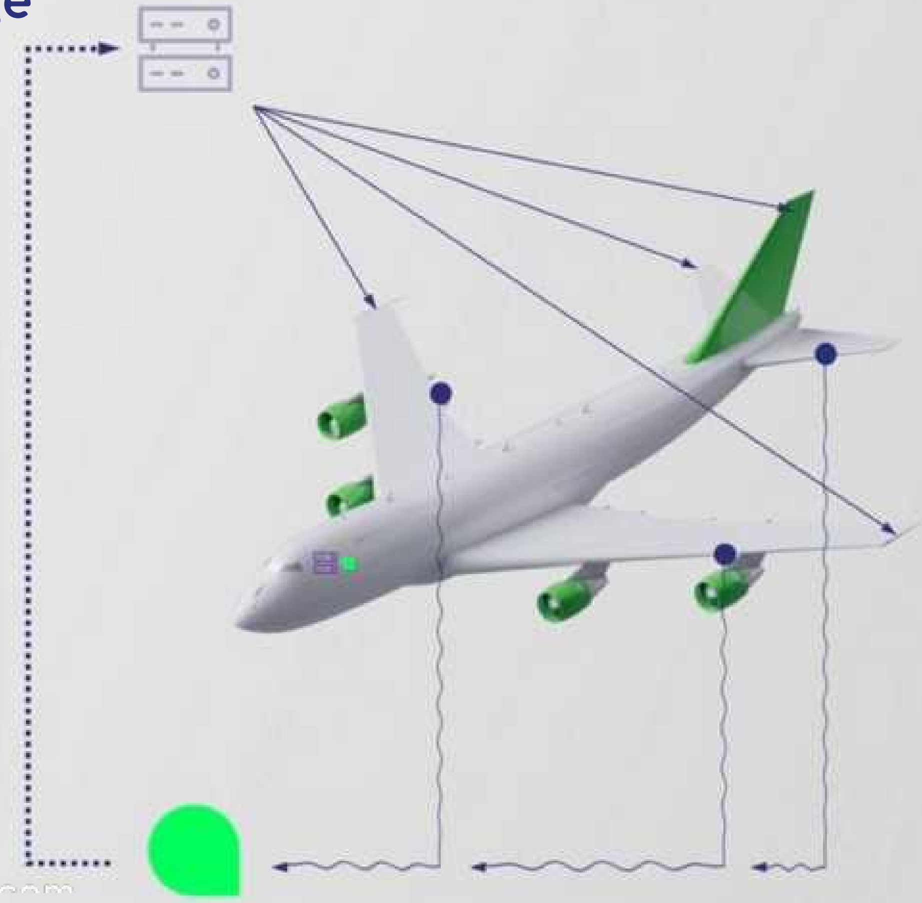
[Sarpeshkar, Neural Computation, 10, 1601-1638 (1998)] [Köppel et al. Advances in Radio Science 2020 / Arxiv:2102.07268] [Bournez, Pouly 2016]

Anabrids Hardware Description Language

Implementing APIs

Circuit.analog

```
solve_ode( @(t,u)
   diff(rho,t) = div(p),
   diff(p_i, t) = div(p_i*v_j - p*dela_ij),
   diff(e, t) = div(v_i*(e+p)),

   p = rho*eps*(Gamma-1)
)
```



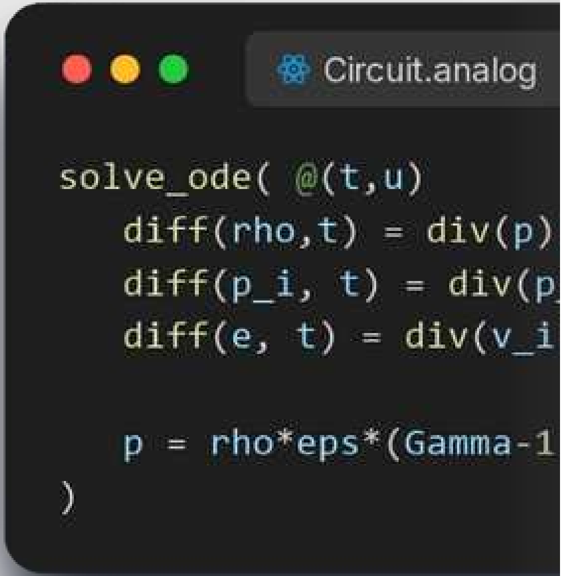
anabrid

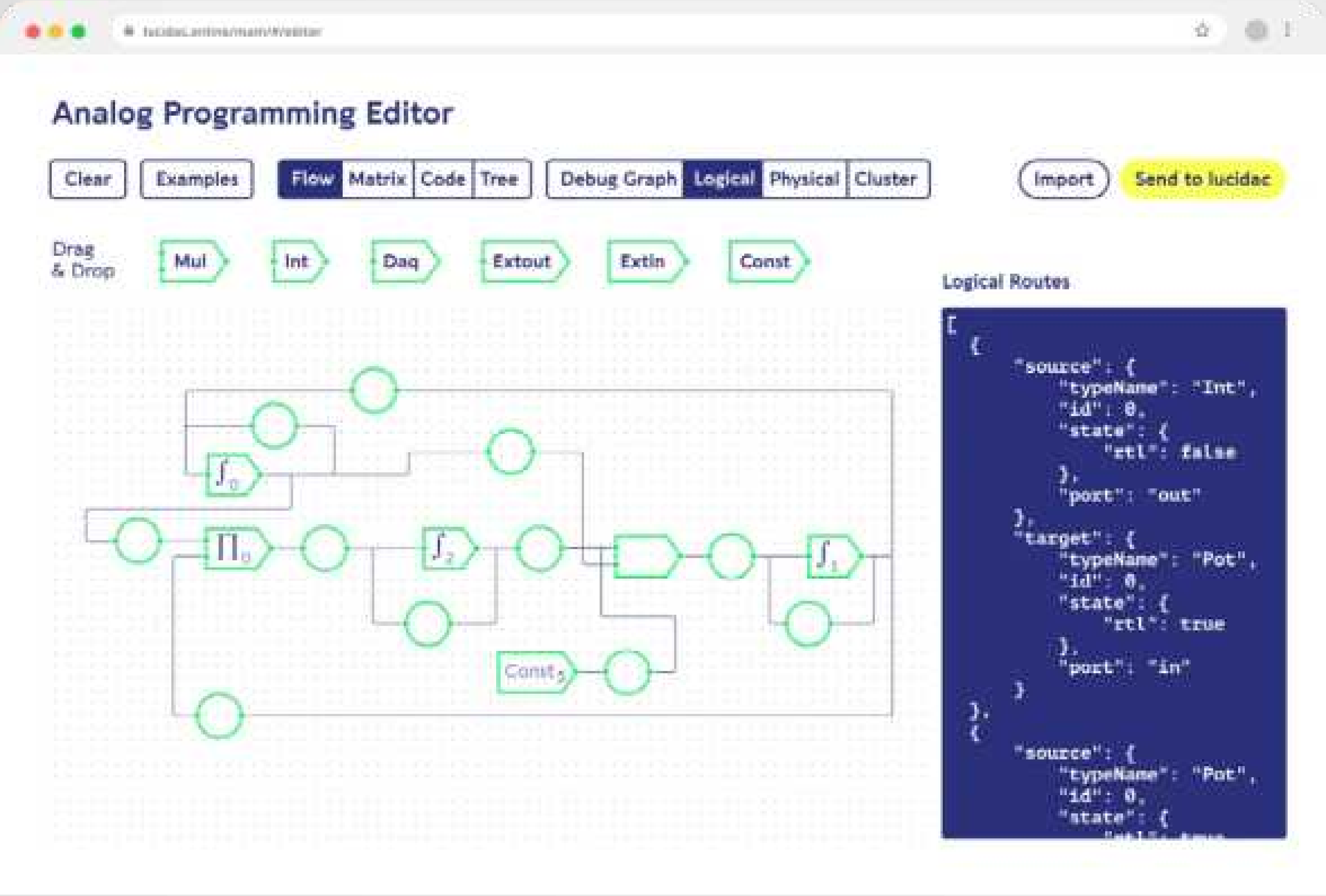
BOEING

[Köppel et al. Advances in Radio Science 2020 / Arxiv:2102.07268]

Anabrids Hardware Description Language

Implementing





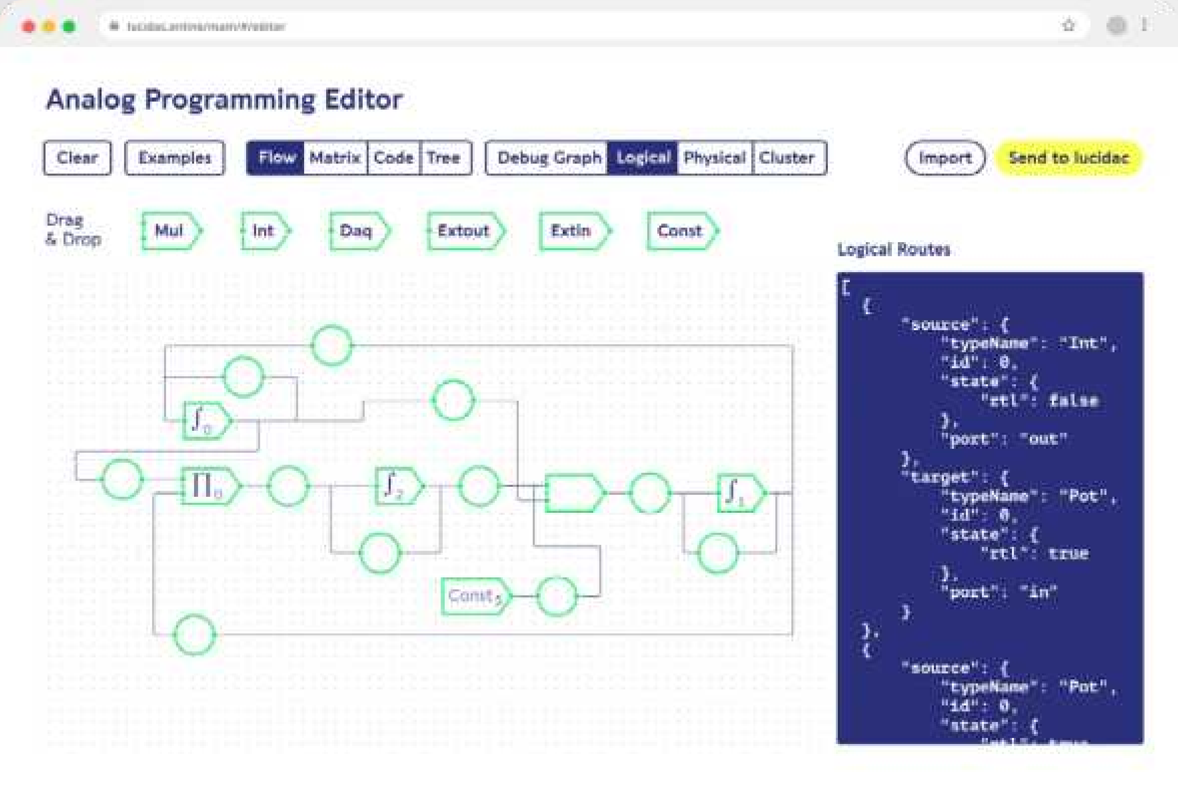
anabrid

Configuring op-amp circuits

Data flow programming

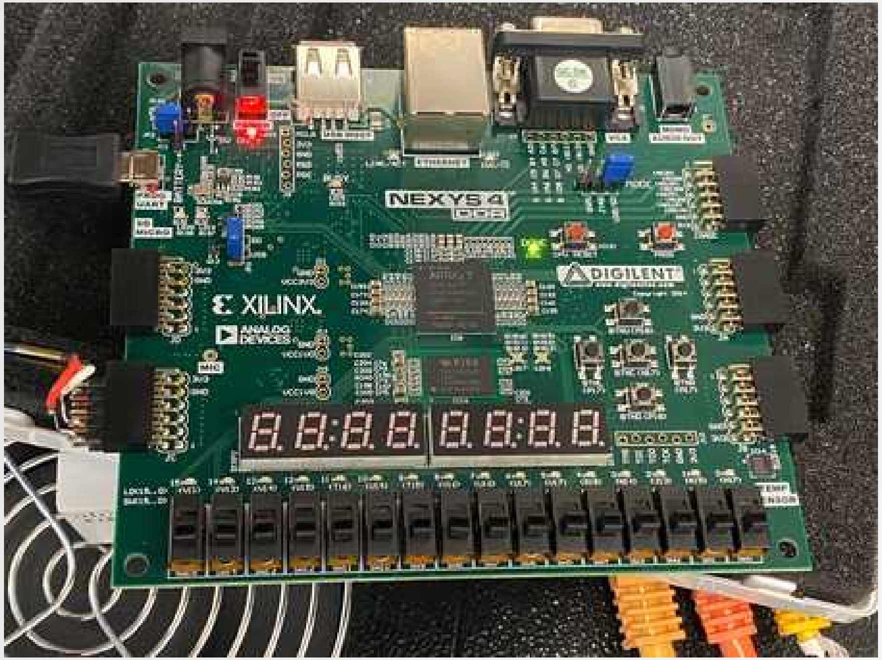
Example listing: **lorenz.py**

```python
from lucipy import LUCIDAC, Circuit

a = 1.0
b = 2.8
c = 2.666 / 10

l = Circuit()          # Create a circuit

mx = l.int(ic = 0.1) # Allocate an integrator with initial condition
my = l.int()           # two more integrators
mz = l.int()
xz = l.mul()           # Allocate two multipliers
xy = l.mul()

l.connect(mx, xz.a)  # Product -x * -z = xz
l.connect(mz, xz.b)

l.connect(mx, xy.a)  # Product -x * -y = xy
l.connect(my, xy.b)

l.connect(my, mx, weight = -a)
l.connect(mx, mx, weight = a)

l.connect(mx, my, weight = -b)
l.connect(xz, my, weight = -5)
l.connect(my, my, weight = 0.1)

l.connect(xy, mz, weight = 5)
l.connect(mz, mz, weight = c)

l.probe(mx, front_port=0) # Connect a DSO to
l.probe(my, front_port=1) # ports 0, 1 or 2
l.probe(mz, front_port=2) # to see the output!

hc = LUCIDAC()
hc.set_circuit(l)

hc.set_op_time(unlimited=True)
hc.start_run()
```

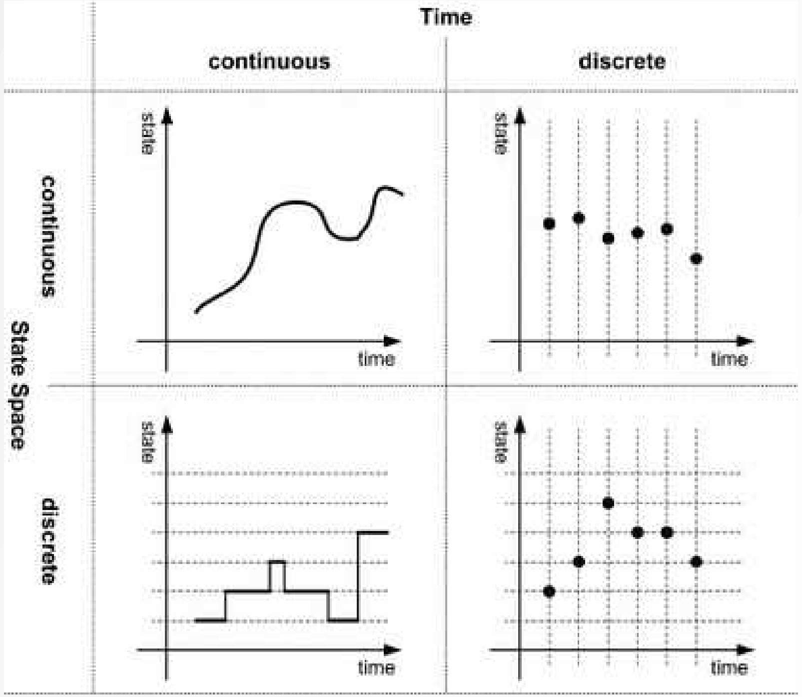


Digital Differential Analysers

Analog Computing on FPGAs



Old Nexys 4 DDR FPGA board, 100Mhz



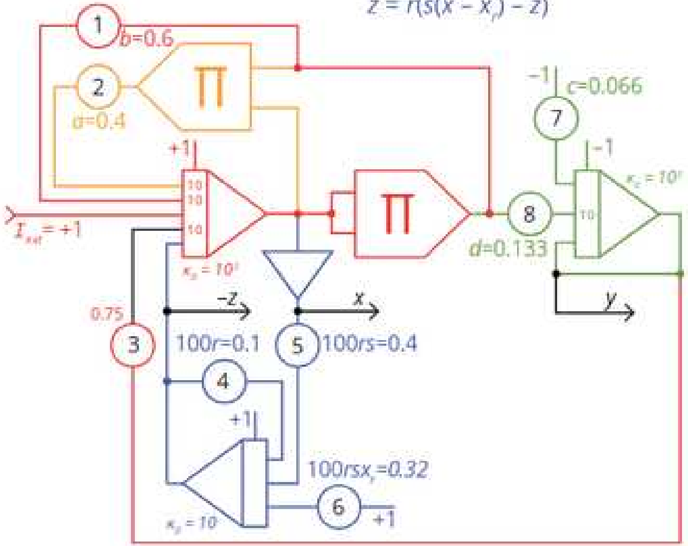
- Exploit massive amount of Multiply-Add Units
- 32bit fixed point (16bit integer, 16bit fractional)
- Toy problems: Rössler attractor, Hindmarsh-Rose Neuron.
- Utilization about 11%

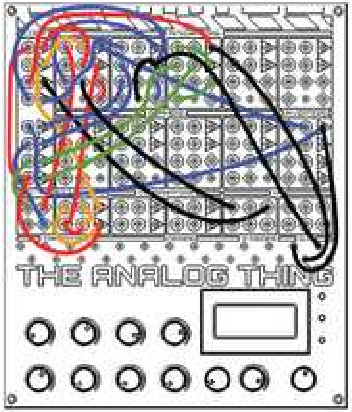
9.4 NEURONAL BURSTING

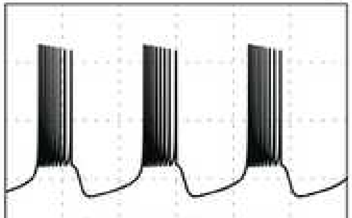
Neurons in the central nervous system receive nerve impulses from other "upstream" neurons and, if such inputs exceed their firing thresholds, give off nerve impulses to "downstream" neurons. Computationally speaking, the aggregation of incoming nerve impulses (also referred to as summation) is a process of integration, albeit within a limited time window, such that incoming impulses soon expire, and only the most recent ones carry weight in the integration as time passes. A few sporadic impulses are usually not enough to exceed the firing threshold. Instead, it takes either multiple upstream neurons to fire together at least near-simultaneously, or a single upstream neuron to send multiple impulses in short bursts. James L. Hindmarsh and R. Malcolm Rose proposed a model for this neuronal bursting in 1984. The model, consisting of the three first-order differential equations shown below, responds to inputs to I_{ext}. The variables x, y, and z correspond to the neuron's (bursting) output potential, the transport of sodium and potassium through fast ion channels, and the transport of other ions through slow channels, respectively.

To add a 10-weighted input to the first (red) integrator, a resistor network is connected to its summing junction (SJ) jack. To operate the second (blue) integrator more slowly than the other two, its output is patched to its SLOW jack. Set the coefficient potentiometers to the values shown in the patching diagram. Connect x (the output of the inverter) to OUT X and connect your oscilloscope or other display system to RCA Out X on the back of THAT. If more channels are available on your visual display system, connect y and $-z$ accordingly. Run the patch in OP mode and observe the image on the display system while connecting and disconnecting I_{ext} to and from +1.

$$\dot{x} = -ax^3 + bx^2 + y - z + I_{ext}$$
$$\dot{y} = -dx^2 + c - y$$
$$\dot{z} = r(s(x - x_r) - z)$$







Output: neuronal burst pattern

[THAT Booklet]

DSL ODE input

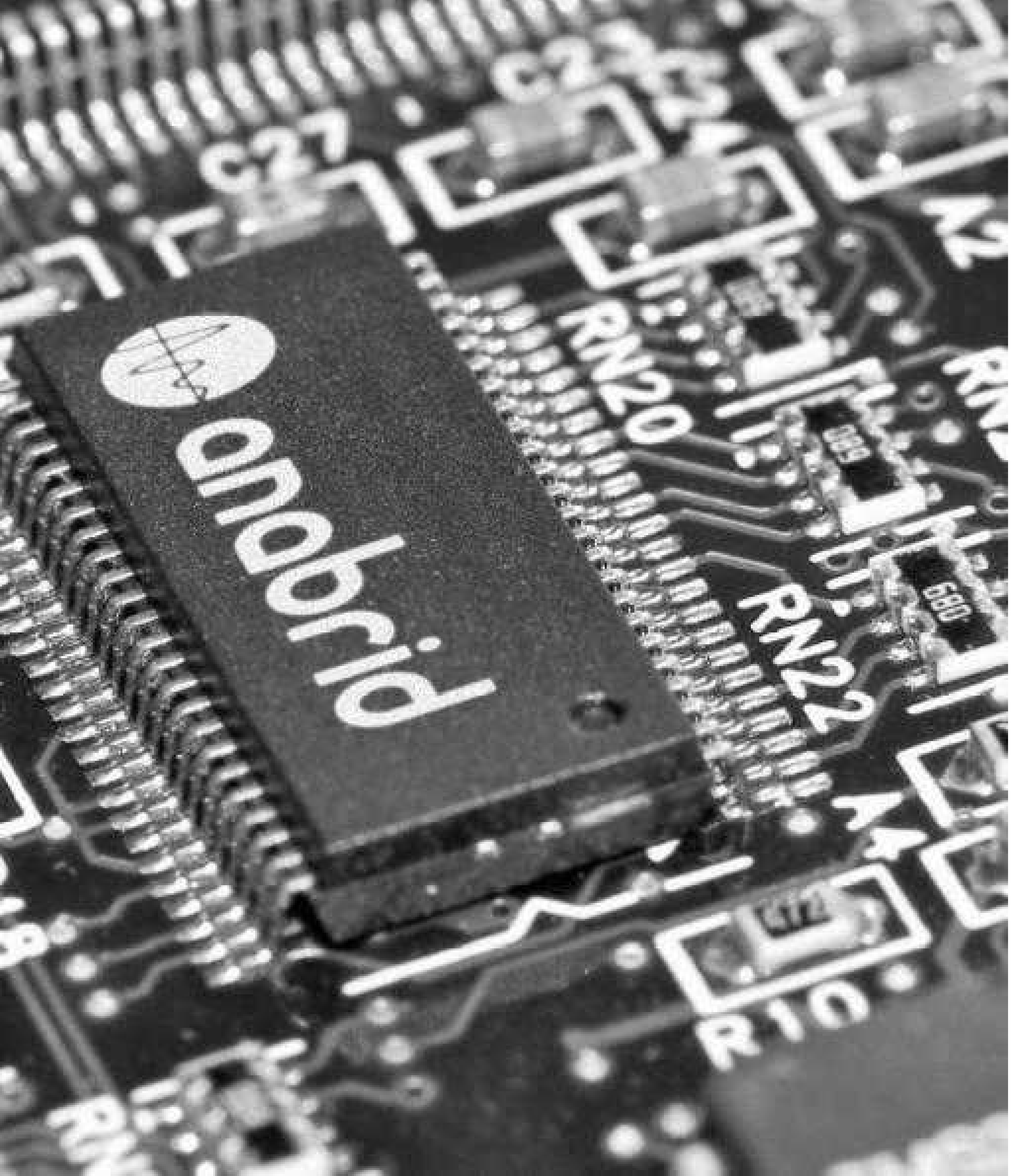
```
{
  "name": "hindmarsh-rose",
  "functions": [
    "X(t)",
    "Z(t)",
    "Y(t)"
  ],
  "parameters": [],
  "equations": {
    "a" : "1",
    "b" : "3",
    "c" : "1",
    "d" : "5",
    "r" : "0.001",
    "s" : "4",
    "i" : "1",
    "diff[X, t]": "-4 * X ** 3 + 6 * X ** 2 + 7
    "diff[Y, t]": "-1.332 * X ** 2 + 0.0666 - Y
    "diff[Z, t]": "0.004 * X + 0.0032 - 0.001 *
  },
  "conditions": {
    "X(t: 0)": 1.0,
    "Y(t: 0)": 0.0,
    "Z(t: 0)": 1.0
  },
  "outputs":[
    "X", "Y"
  ],
  "displays": [
    "plot(x: t * 0.001 - 1.0, y: X)",
    "plot(x: t * 0.001 - 1.0, y: Y)"
  ]
}
```

Verilog output

```
wire [31:0] lit_26_sig;
wire [31:0] arith_27_sig;
wire [31:0] arith_28_sig;
wire [31:0] lit_29_sig;
wire [31:0] arith_30_sig;
wire [31:0] lit_31_sig;
assign lit_2_sig = real_to_fixed(1);
assign lit_7_sig = real_to_fixed(-4);
mul mul_inst_0(
  .clk(clk),
  .reset(reset),
  .update(update),
  .ovf(ovfs[7]),
  .lhs(itor_0_sig),
  .rhs(itor_0_sig),
  .out(arith_9_sig)
);
mul mul_inst_1(
  .clk(clk),
  .reset(reset),
  .update(update),
  .ovf(ovfs[6]),
  .lhs(arith_9_sig),
  .rhs(itor_0_sig),
  .out(arith_8_sig)
);
mul mul_inst_2(
  .clk(clk),
  .reset(reset),
  .update(update),
  .ovf(ovfs[5]),
  .lhs(lit_7_sig),
  .rhs(arith_8_sig),
  .out(arith_6_sig)
);
```

PWM to DSO: „dense solution“

Micro benchmark:
FPGA: 1W
THAT: 0.4W
G-PAC: 10mW



Analog architectures on FPGAs and ASICs

Dr. Sven Köppel
koeppel@anabrid.com

Thank you for your attention!

2025-07-14, 10:30 AM
Paderborn Center for Parallel Computing (PC2)
PC2 Seminar at U Paderborn
Room X0.101, Mersingweg 5, 33098 Paderborn