

A Vision for Neuromorphic and Analog Computing Tools

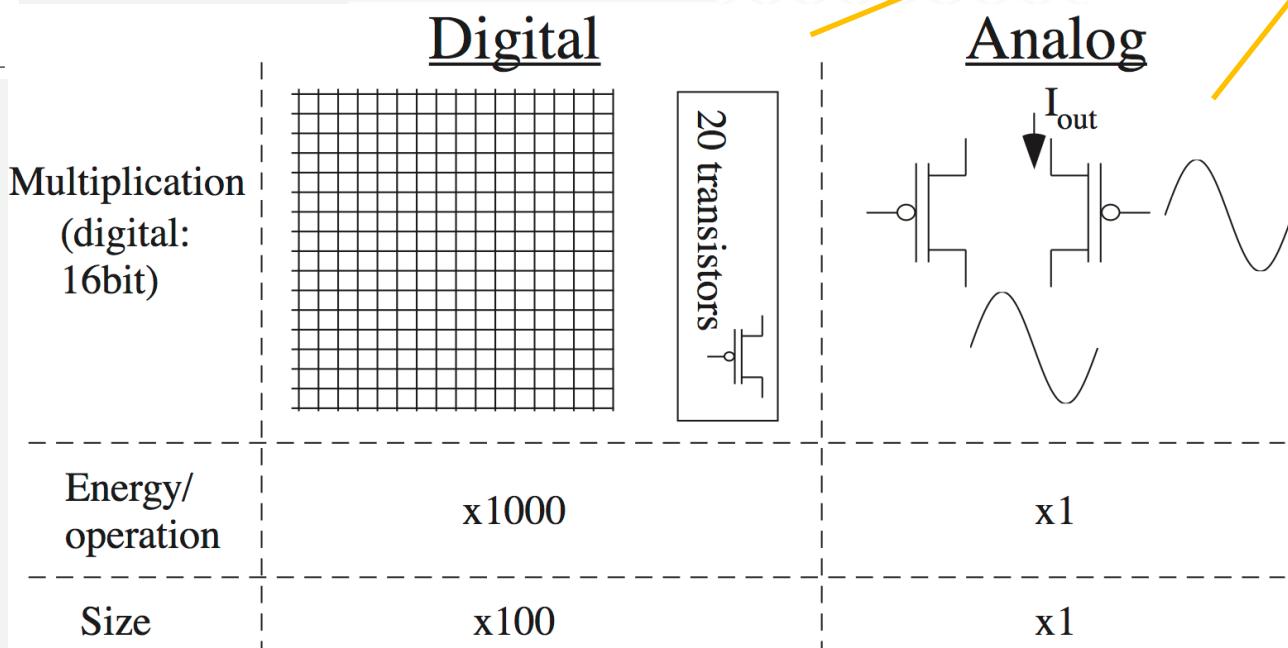
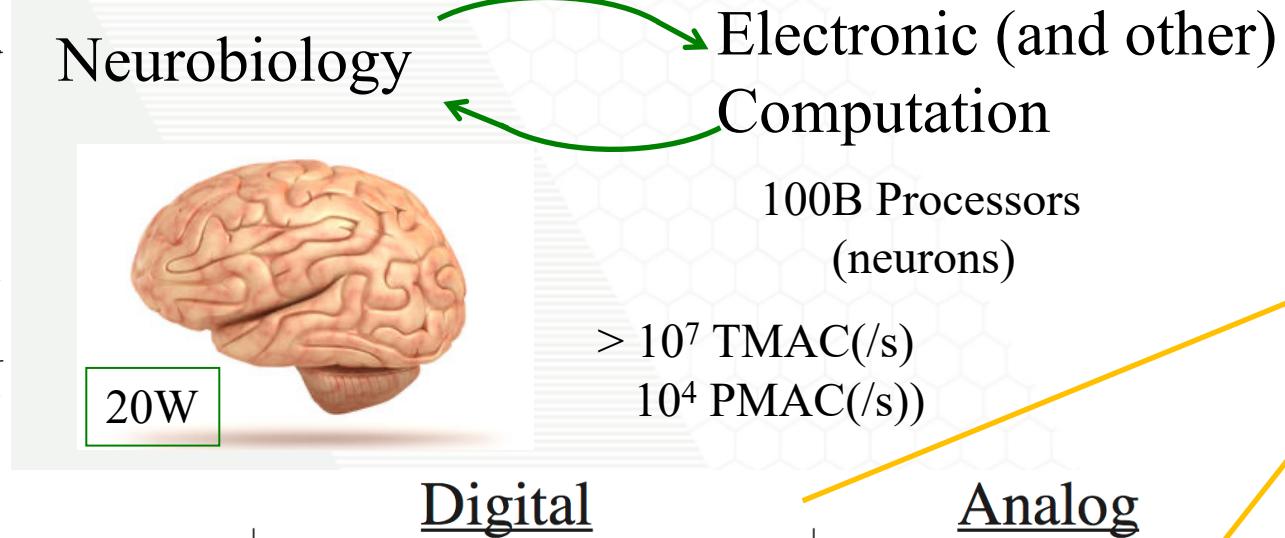
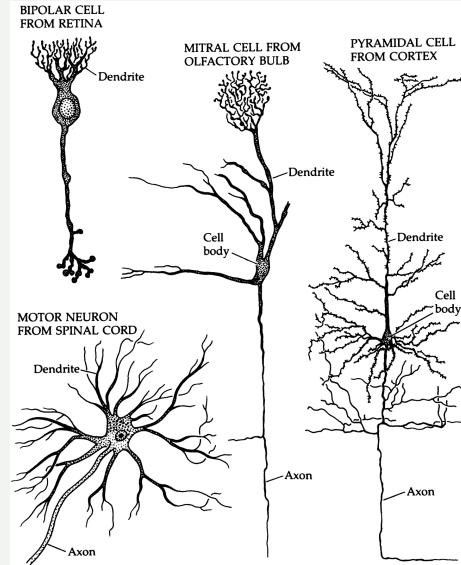


Georgia Institute of Technology
Jennifer Hasler



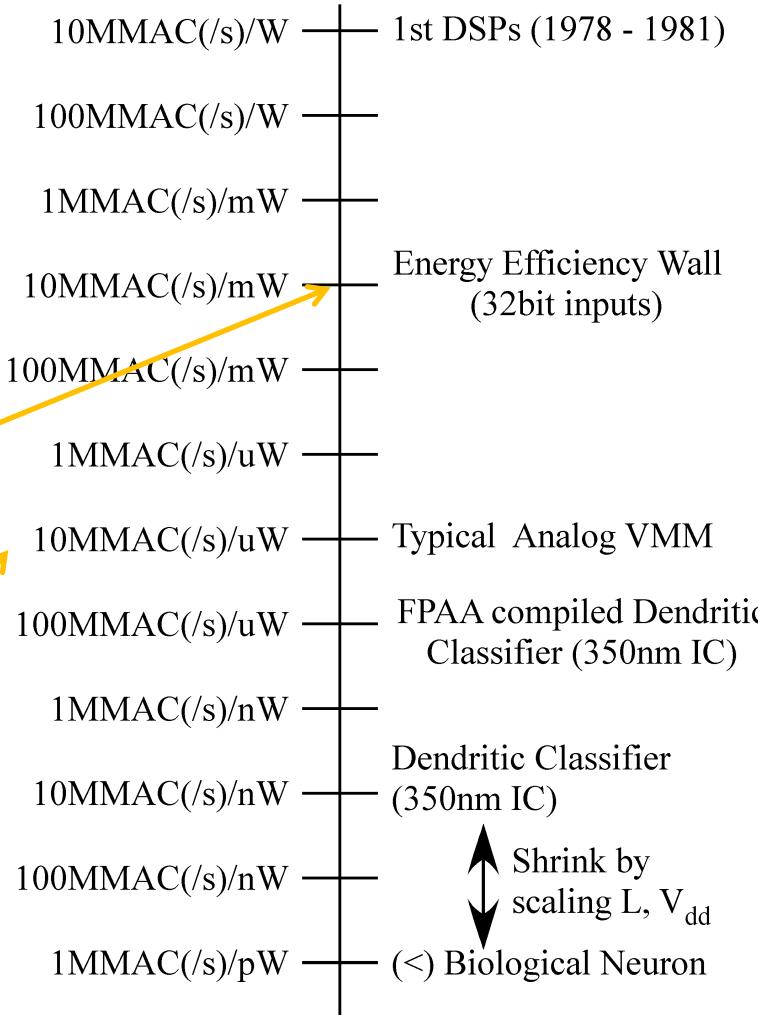
Georgia Tech College of Engineering
School of Electrical
and Computer Engineering

NEUROMORPHIC COMPUTING



C. Mead, IEEE Proceedings, 1990

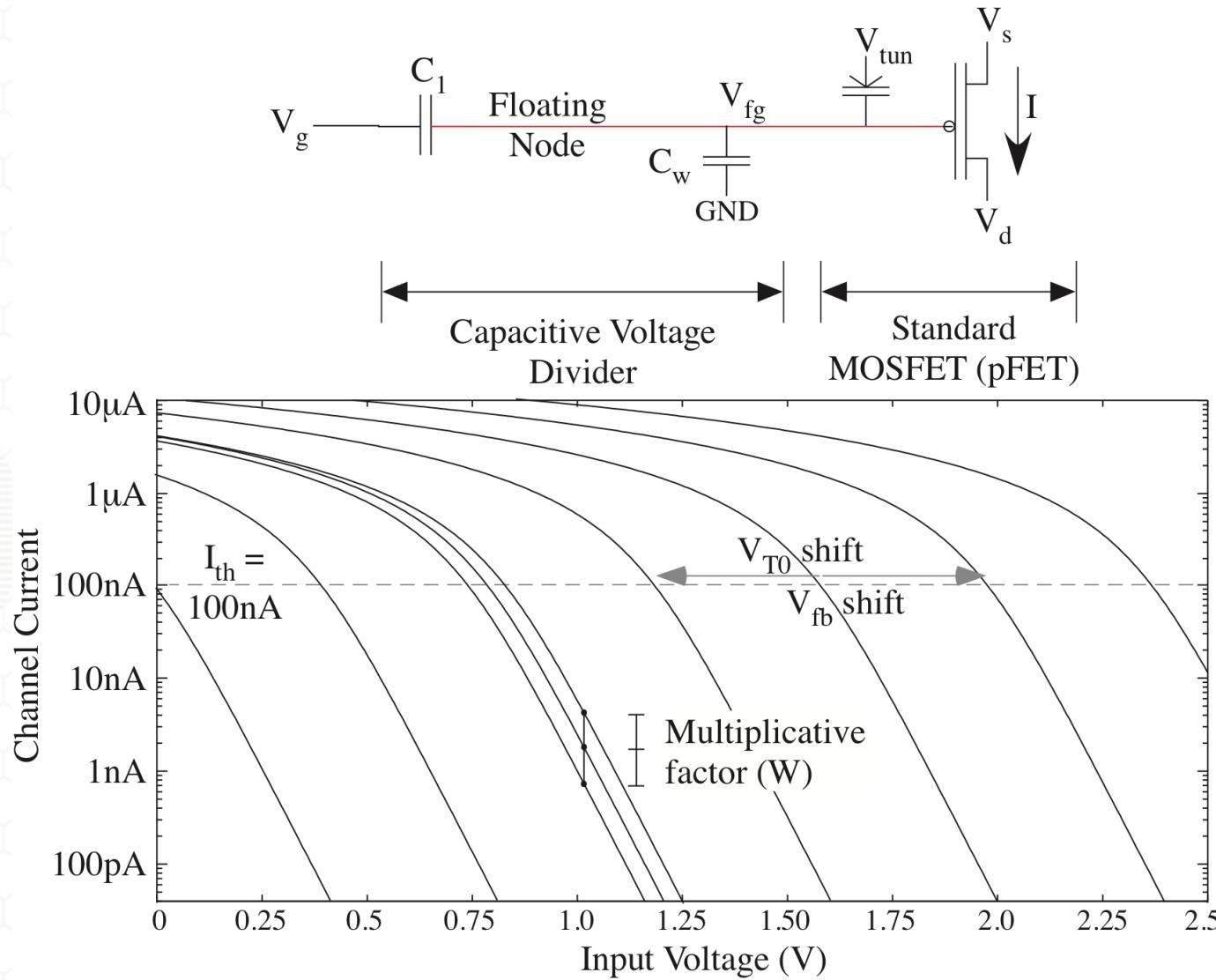
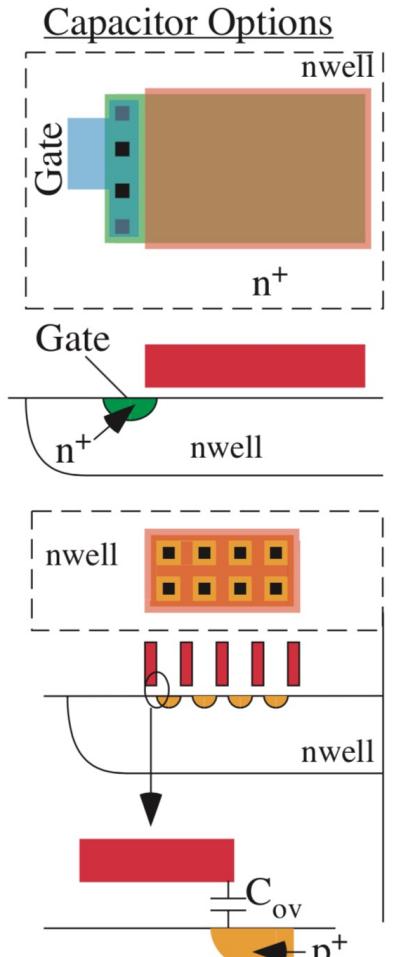
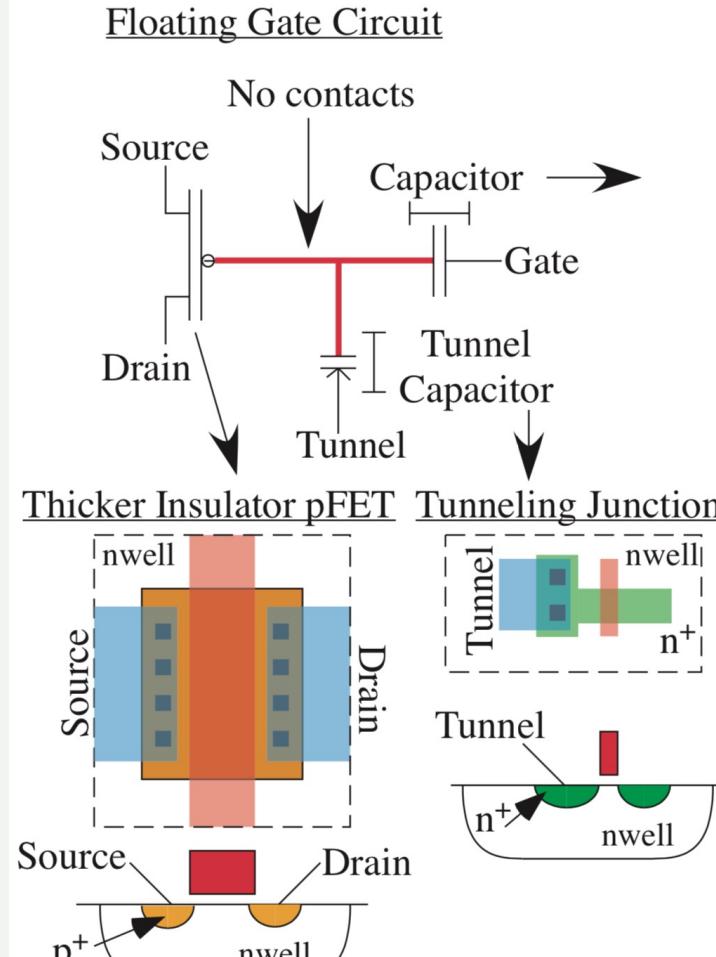
Power Efficiency Scaling



[Hasler and Marr 2013]

IEEE Spectrum,
June 2017

ANALOG PROGRAMMABILITY → FLOATING-GATE (FG) CIRCUITS



Program FG charge once and holds for part lifetime
(i.e. 10 year lifetime = 10-100 μ V)

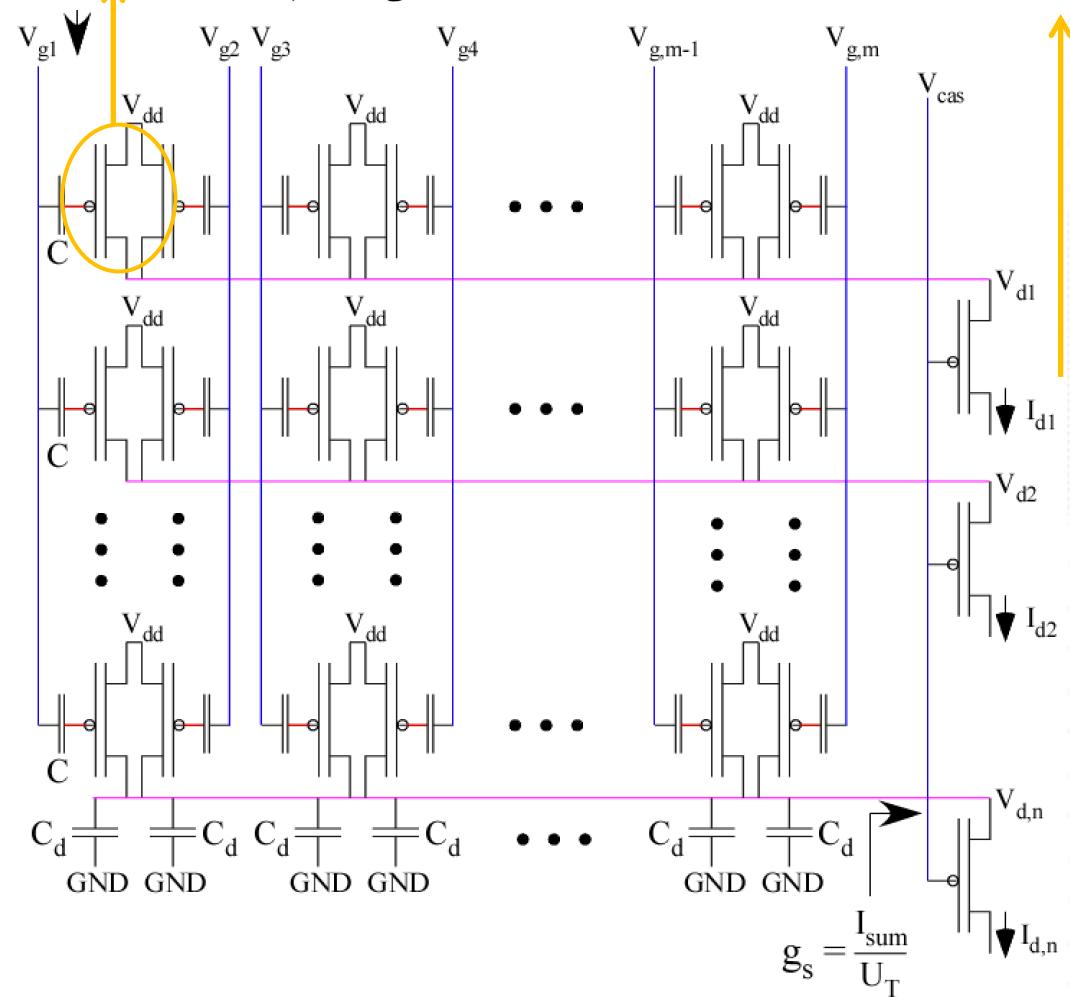
Programmed to 14bit accuracy: e.g. 60 μ V on 1V supply

Floating-Gate (FG) circuit techniques enables
direct solution for mismatch (V_{T0}) (Standard CMOS)

PROGRAMMABILITY → ANALOG VMM → SIGNAL PROCESSING

$$I \approx g_{m1} V_{g1}$$

$$= g_m W_{1,1} V_{g1}$$



$$I_{d,1} = I_{sum,1} \approx g_m \sum_{k=1}^m g_m W_{1,k} V_{g,k}$$

Single Transistor Learning Synapse (1994-1995)

Computing in Memory (2001)

[ARVLSI 2001]

Experimental verification (VMM) of
Mead's Hypothesis (2004)

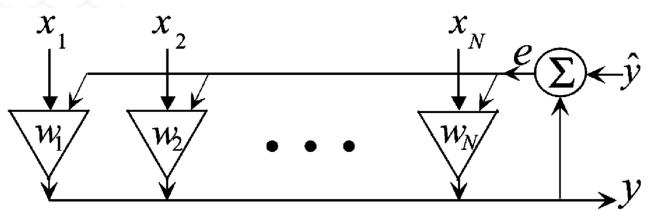
[Chawla, et. al, CICC 2004]

$$\mathbf{i}_d = g_m \mathbf{W} \mathbf{v}_g$$

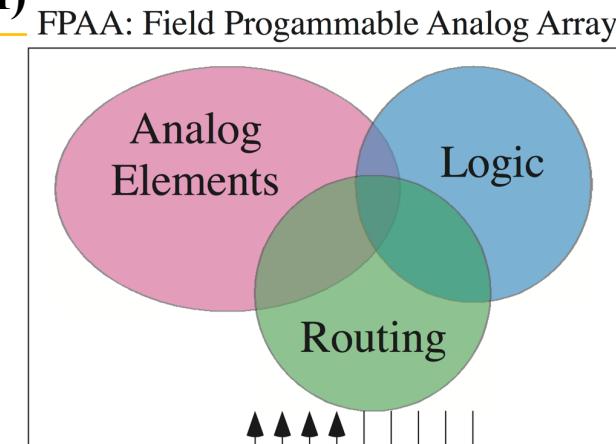
Vector Matrix Vector

Vector-Matrix Multiplication (VMM)

Adaptive Filters (2003)
→ Continuous Learning NN



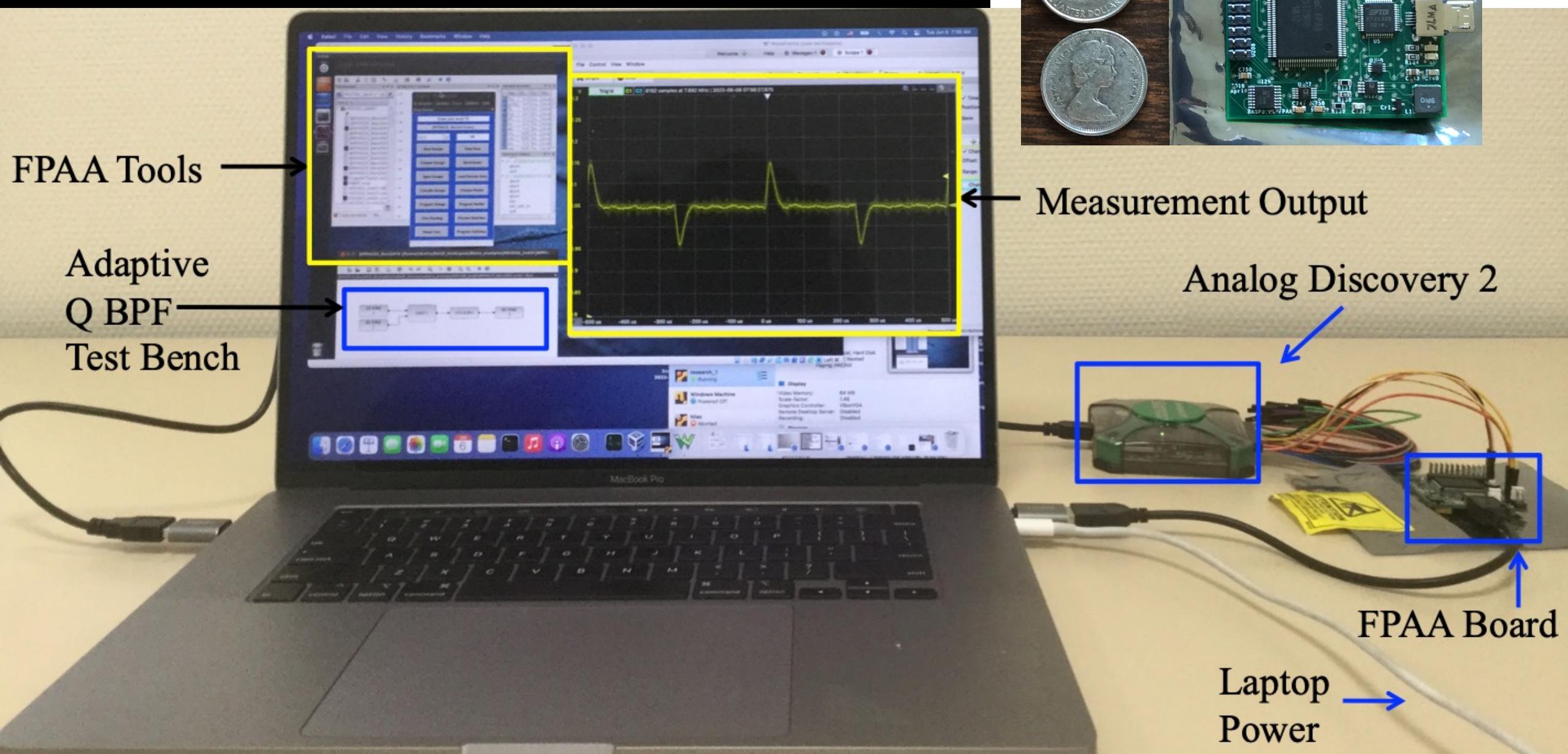
[Dugger & Hasler, 2003-2005]



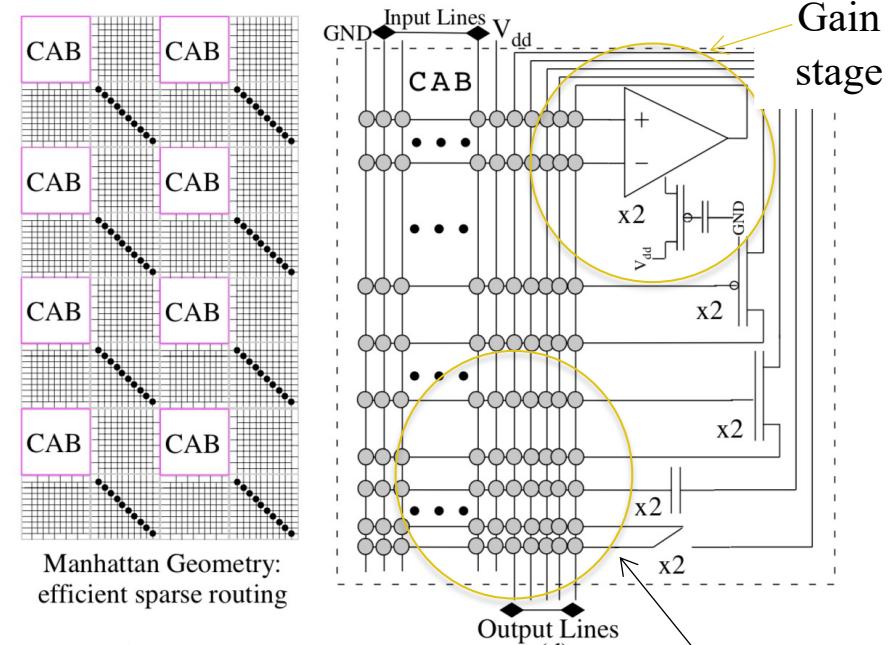
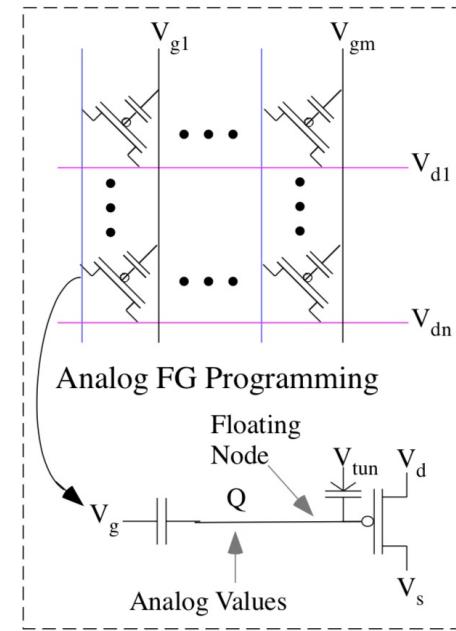
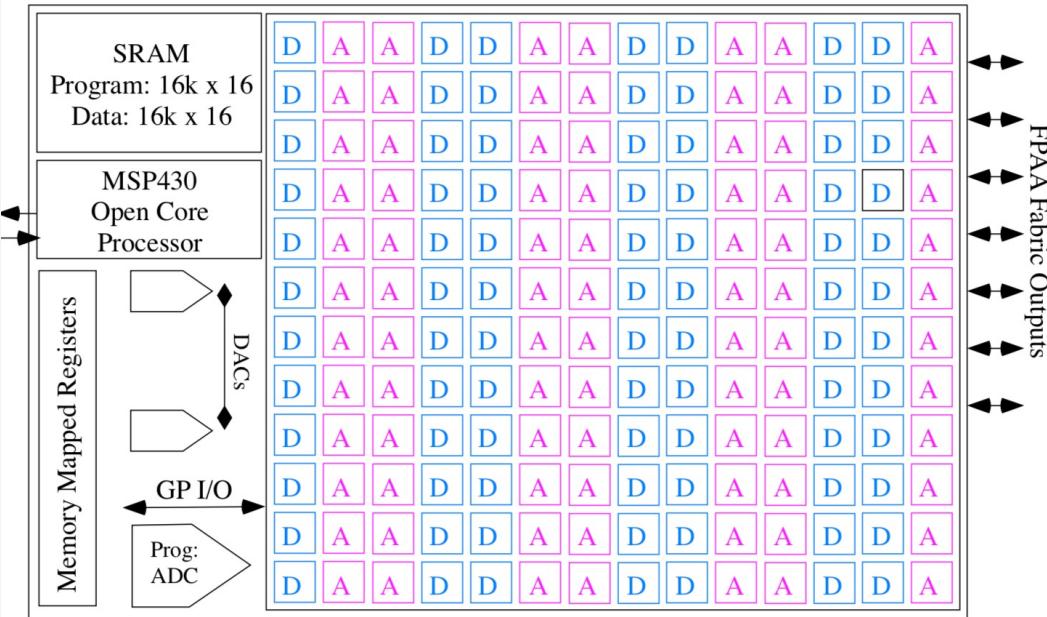
Inputs / Outputs (Analog or Digital)
(2002)

Computing in Routing (2007)

TYPICAL FPAA EXPERIMENTAL SETUP



LARGE SCALE FIELD PROGRAMMABLE ANALOG ARRAYS (FPAA)



SoC FPAA:
350nm CMOS

Partial List of Demonstrated FPAA Algorithms

Vector-Matrix Multiplication (VMM)
Acoustic and Bio sensor processing
Optimal Path Planning
Delay lines and linear phase filters

Analog Computing (e.g. ODE, $Ax=b$)
Neural interfacing and processing
Neural architectures
Spatiotemporal Beamforming

Embedded machine learning
Acoustic Inference and training
Compressed Sensing Reconstruction
IC security / Noise Generators

"Switches are not
Dead Weight"
(2007)

Want to read more on FPAs?



Large-Scale Field-Programmable Analog Arrays

J. Hasler, "Large-Scale Field-Programmable Analog Arrays," *Proceedings of IEEE*, 2020.

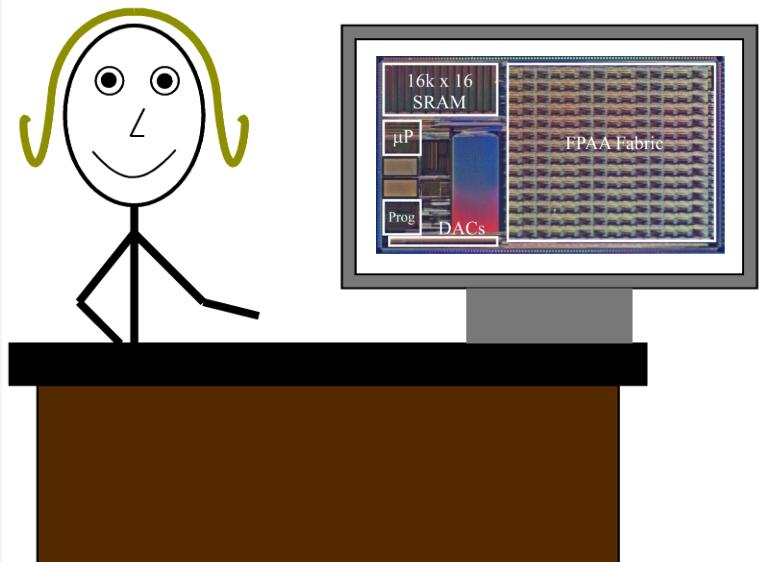
By JENNIFER HASLER^{ID}, Senior Member IEEE

ABSTRACT | Large-scale field-programmable analog array (FPAA) devices could enable ubiquitous analog or mixed-signal low-power sensor to processing devices similar to the ubiquitous implementation of the existing field-programmable gate array (FPGA) devices. Design tools enable high-level synthesis to gate/transistor design targeting today's FPGA devices and the opportunity for analog or mixed-signal applications with FPAA devices. This discussion will illustrate the FPAA concepts and FPAA history. The development of FPAs enables the development of multiple potential metrics, and these metrics illustrate future FPAA device directions. The system-on-chip (SoC) FPAA devices illustrate the IC capabilities, computation, tools, and resulting hardware infrastructure. SoC FPAA device generation has enabled analog computing with levels of abstraction for application design.

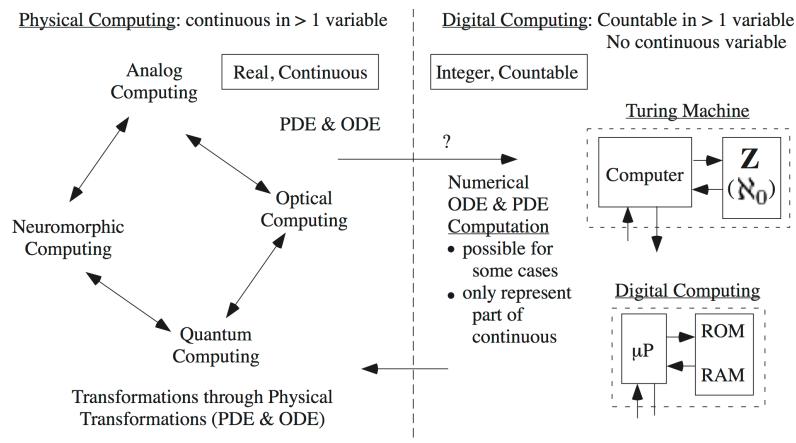
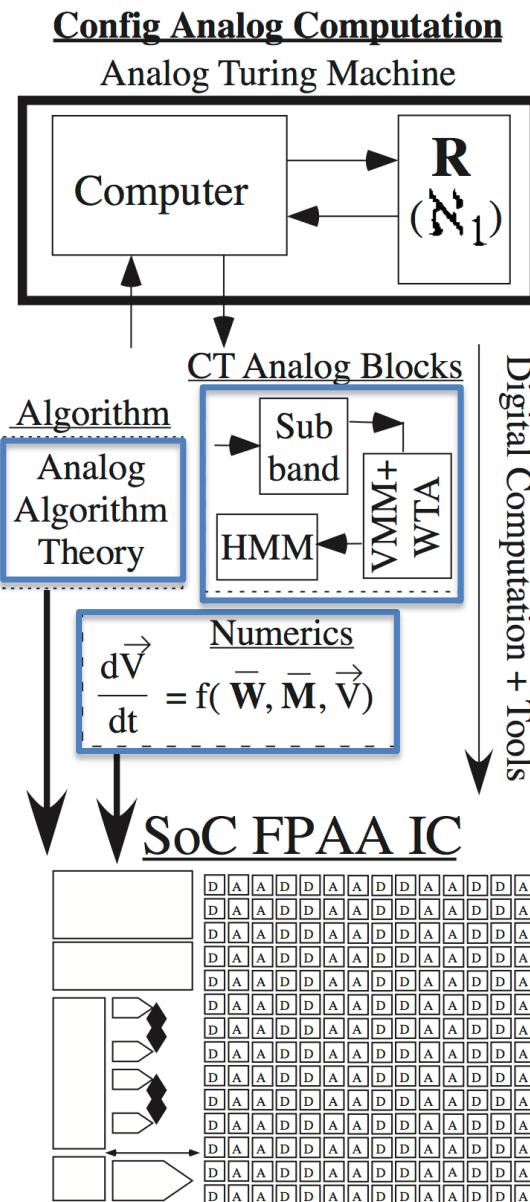
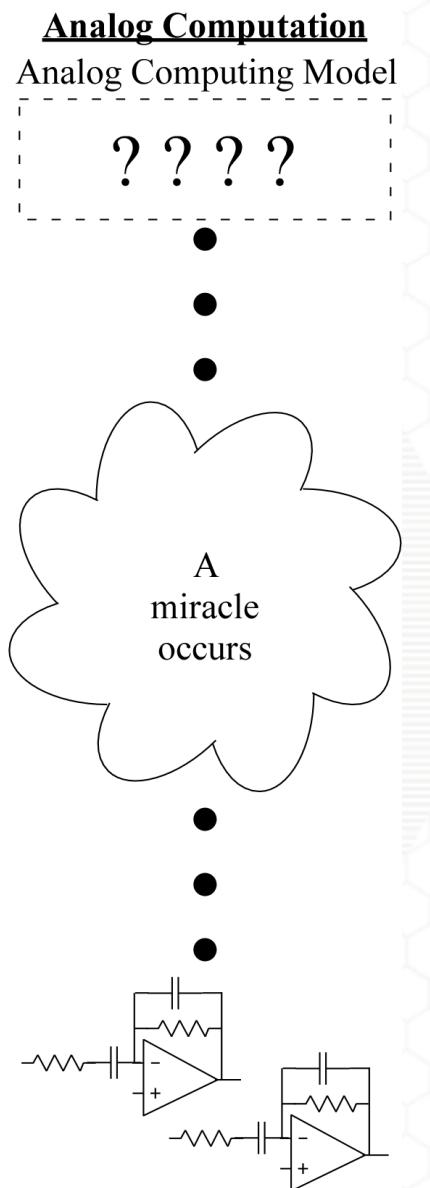
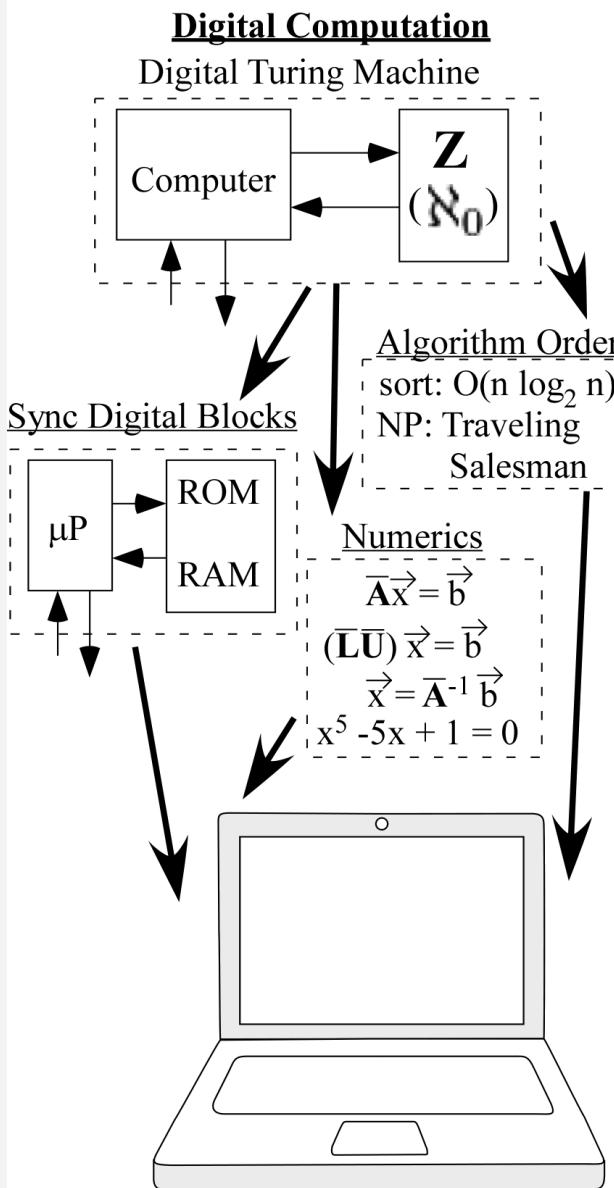
KEYWORDS | Analog-digital integrated circuits, analog integrated circuits, CMOS integrated circuits, field-programmable analog arrays (FPAs), field-programmable gate arrays (FPGAs)

lution [1] enabled further separation of roles to address the increasing complexity resulting from Moore's law scaling [2]–[4]. Digital microprocessors (μ P) are ubiquitous from embedded applications to general-purpose (GP) computing. Programmability enables changing parameters or coefficients in a particular algorithm. Changing the stored matrix of weights for a vector–matrix multiplication (VMM) is an example of programmability. Configurability enables changing the data flow, topology, as well as the order or operations. Changing the program for an μ P is an example of configurability. Field-programmable gate array (FPGA) devices, programmable and configurable gate-level digital devices, enabled digital designers' design capabilities from gate- to system-level designs. FPGAs are ubiquitous digital computing devices found everywhere over the last two decades, arising from their initial conception (1980) and commercialization (mid-1980s) [5].

Modifying the parameters or control flow requires significant changes, such as soldering new components



FRAMEWORK FOR ANALOG / PHYSICAL COMPUTING



Analog Numerical Analysis: “Pristine Digital Computation verses Noisy Analog”

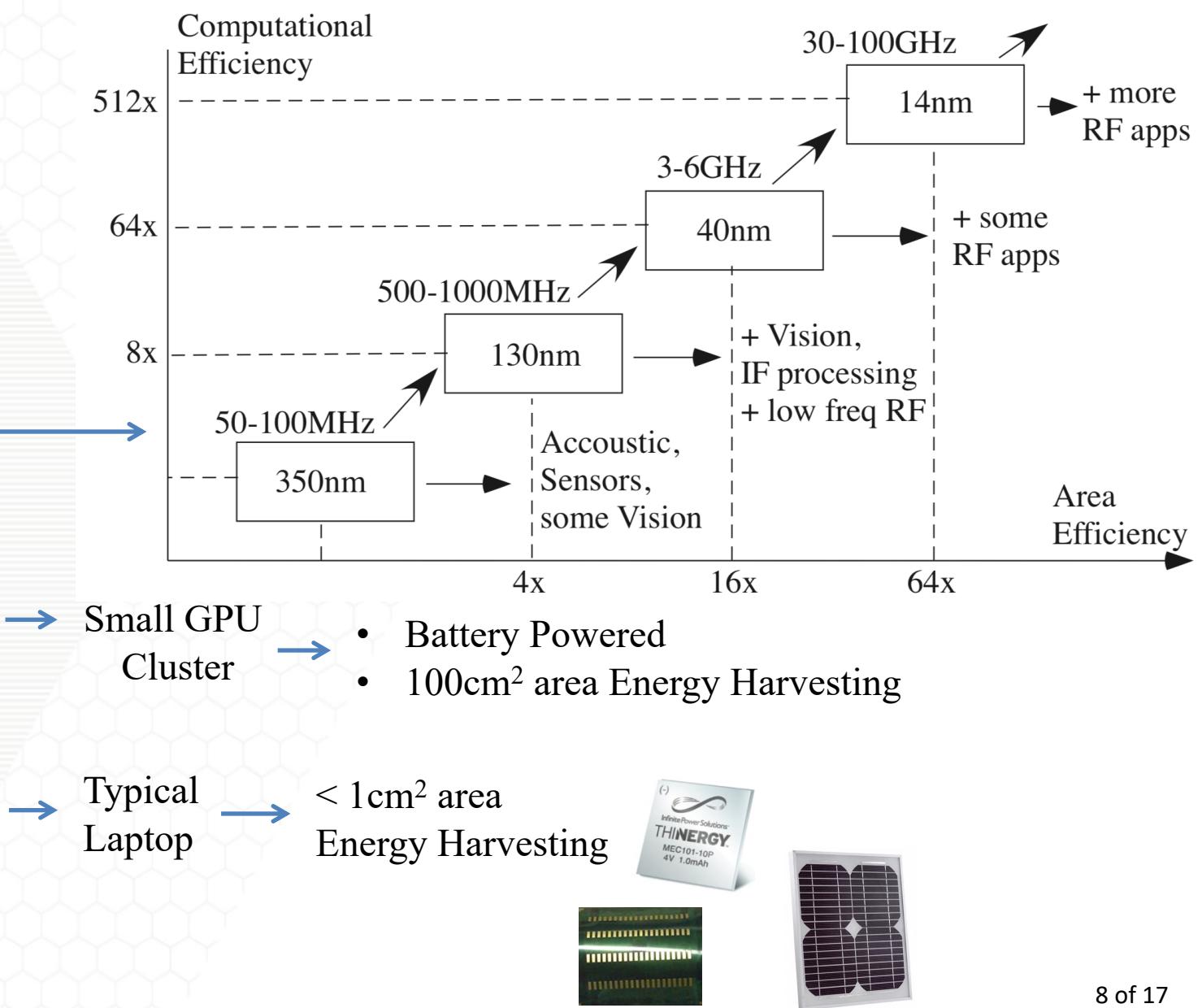
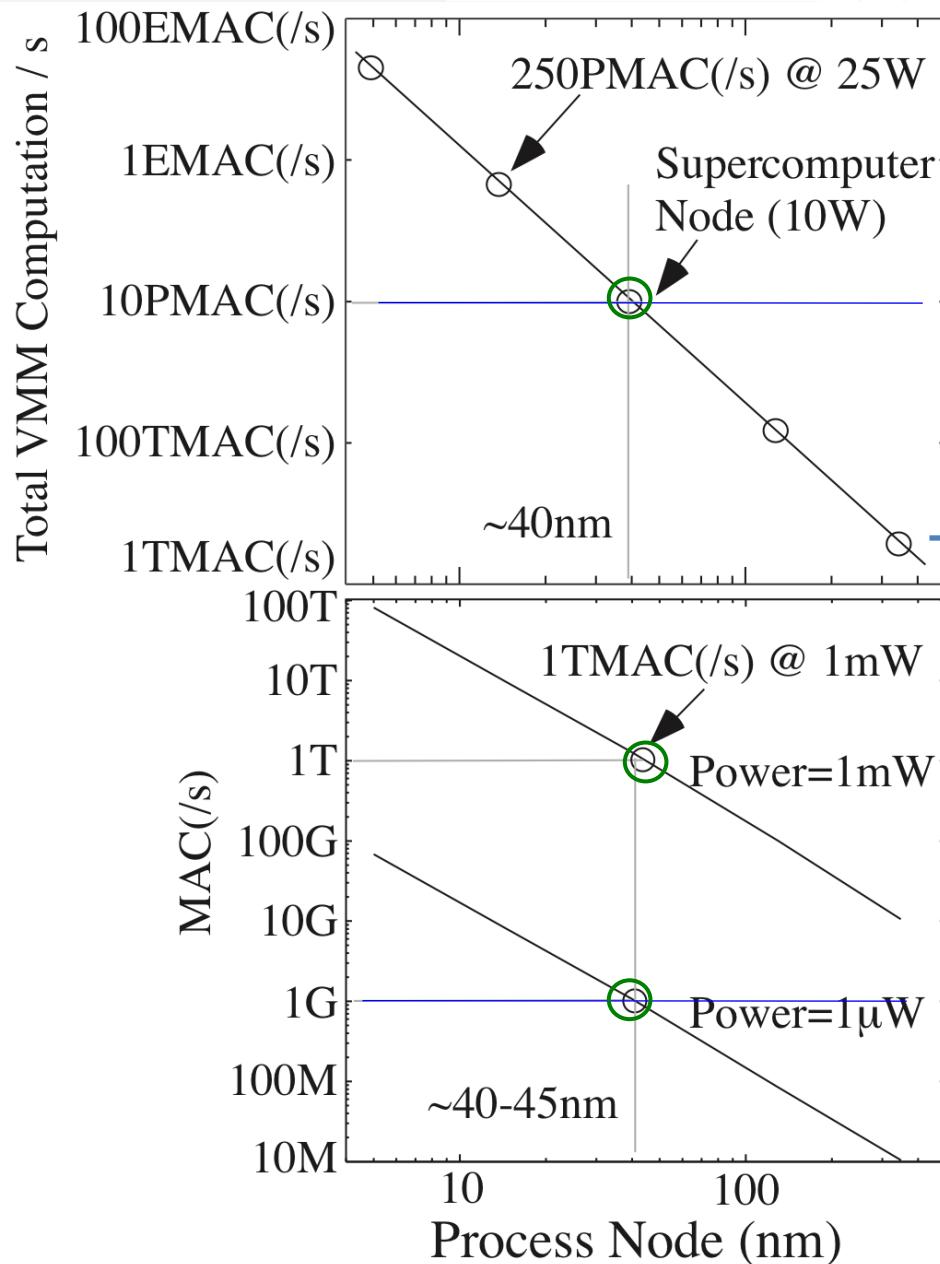
Digital: good starting precision → $Ax = b$

Analog: good intermediate numerics → ODEs

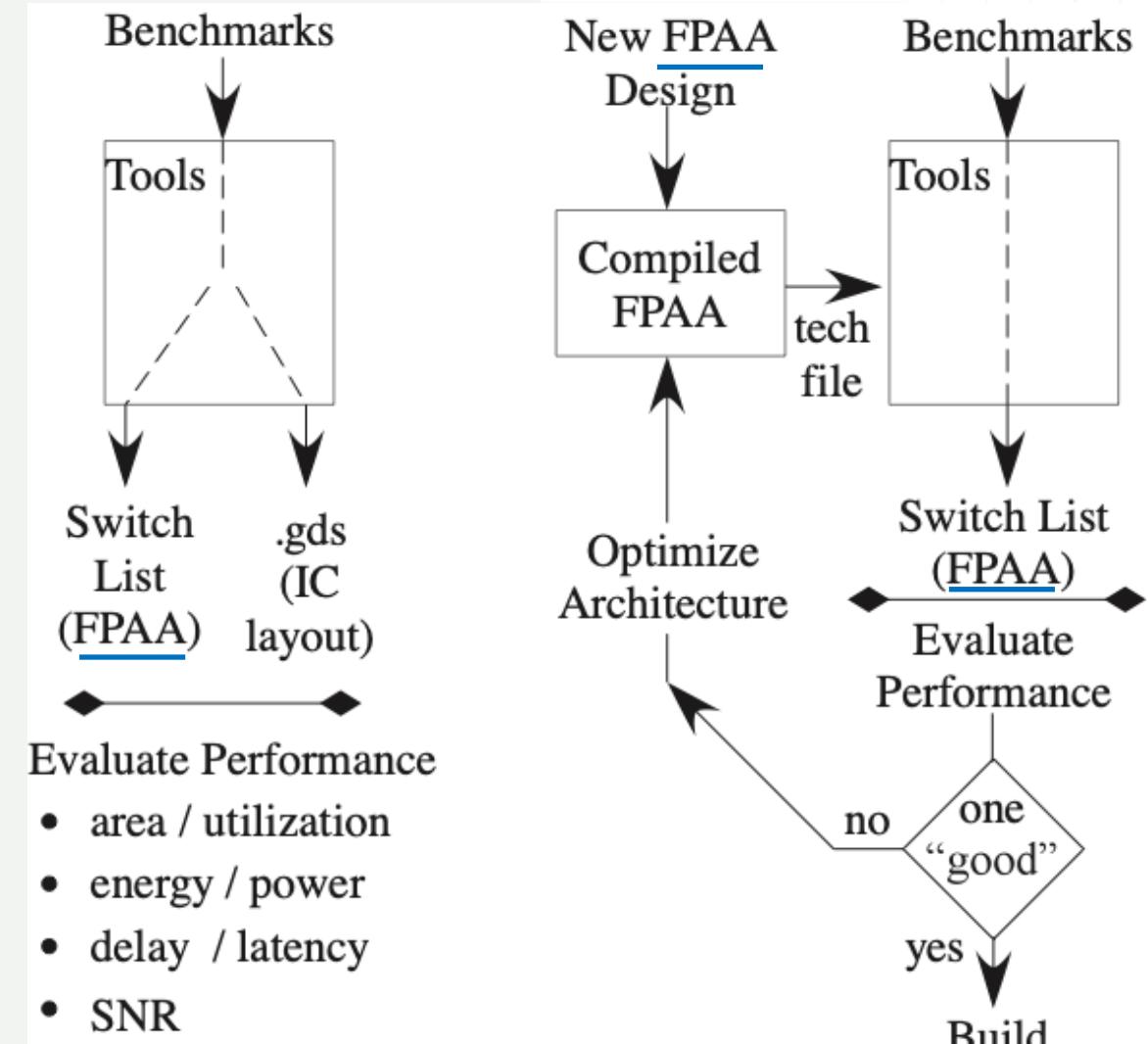
Analog Abstraction: “Digital is made from a few components, but analog is just too complicated”

Analog Architecture and Complexity: “Use classical digital for analog architectures and complexity”

FPGA Scaling further enables Energy Harvesting Applications



TOOLS FUEL THE OPPORTUNITIES



[Hasler & Hao, TRETS / FPL 2023]

Analog Computing Tools

- Developing & Developed
- Becoming part of IC Design Flow
- Analog Benchmarks
- Analog Synthesis

Programmable & Configurable Edge (& elsewhere) ML devices

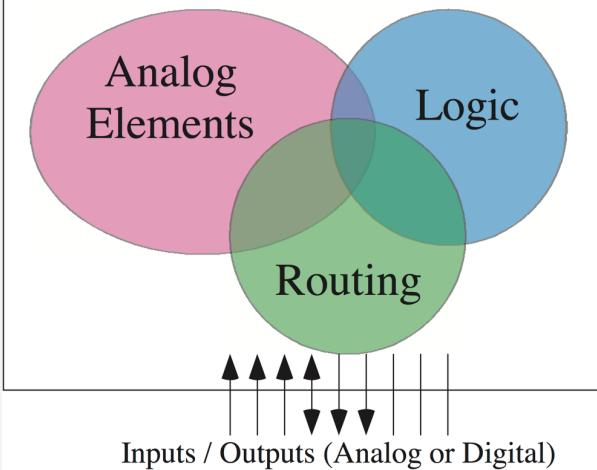
Neuromorphic Tools

- These remain an open question
- What can be borrowed from Analog Computing tools?
- What new concepts required?

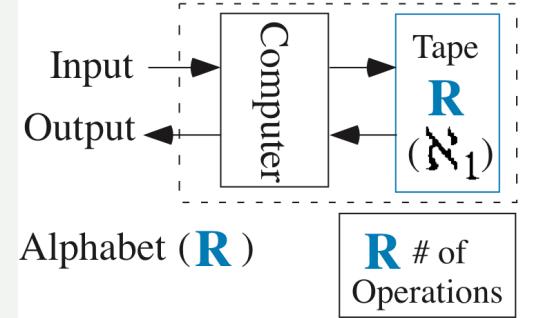
ANALOG/NEUROMORPHIC COMPUTATION

Developing Analog Tools

FPAA: Field Programmable Analog Array



Wide range of demonstrated engineering efficient algorithms



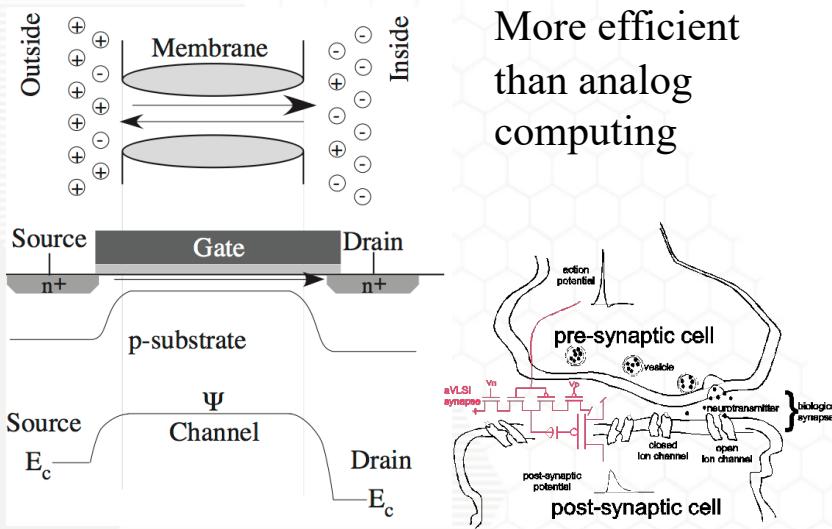
Example: Analog

$$\text{Numerical Analysis} \quad \frac{d\vec{V}}{dt} = f(\vec{V})$$

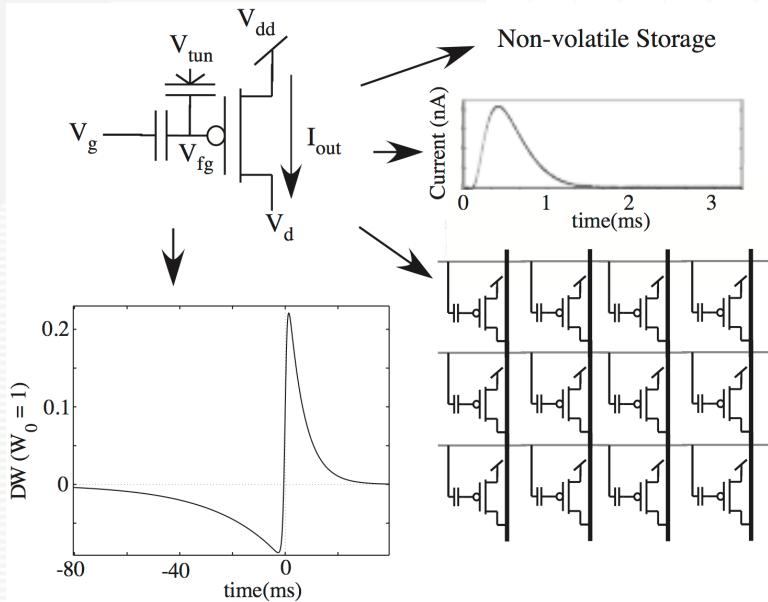
$$\text{Architecture / Algorithms} \quad \vec{A} \vec{x}$$

Abstraction

Neuromorphic Tools....

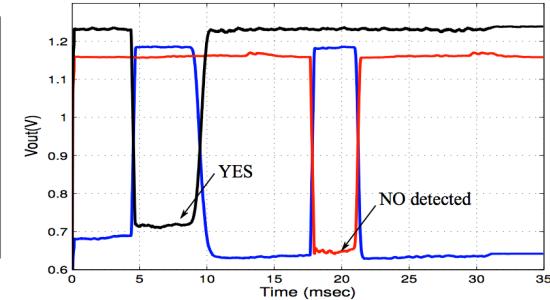
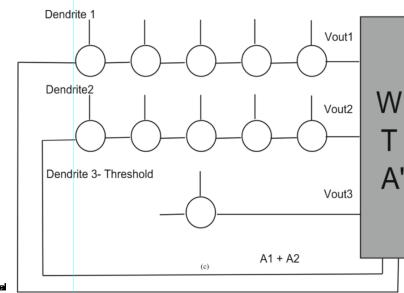


More efficient than analog computing



Very few engineering competitive Neuromorphic engineering concepts

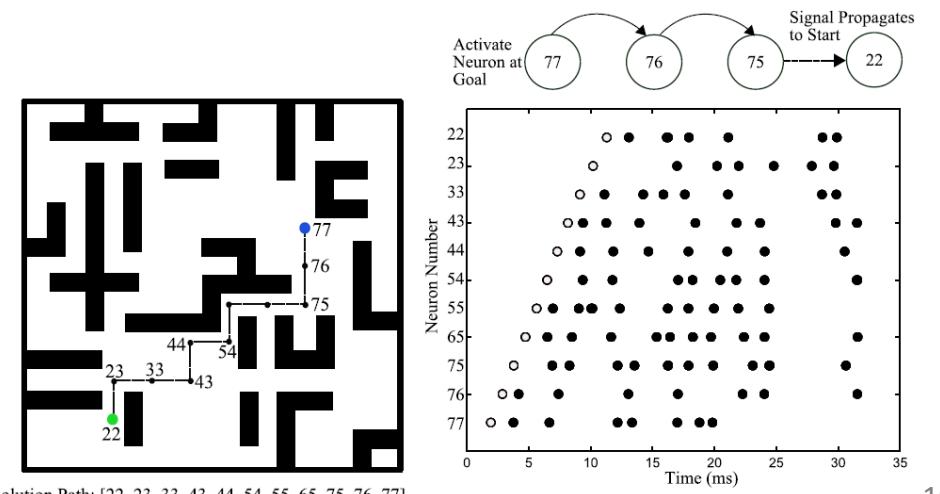
Wordspotting by Coincidence Detection



HMM classifier → Dendrite enabled Neurons

[George, et. al, 2013], FPAA data

Neuron-based Optimal Path Planning



Solution Path: [22, 23, 33, 43, 44, 54, 55, 65, 75, 76, 77]

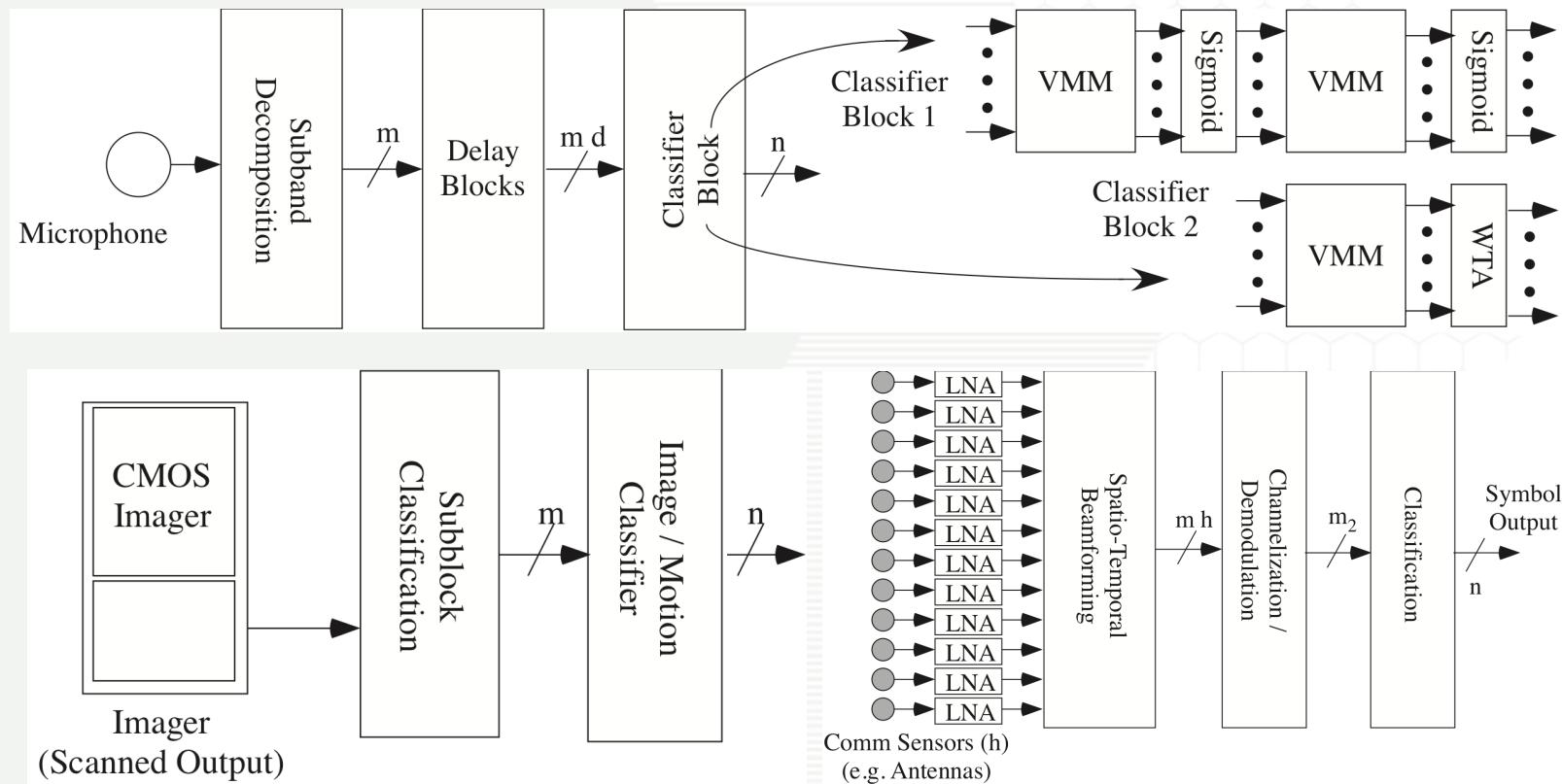
[Koziol, Brink, Hasler 2013, 2014]

ANALOG & NEURMORPHIC BENCHMARKS

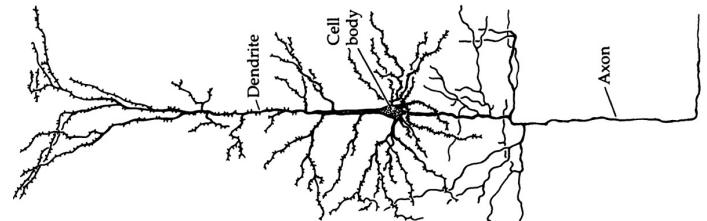
Developing Analog Tools

Benchmarks illustrate computation

- More than just picking some typical problems



Neuromorphic Tools....



Wide Open Question:

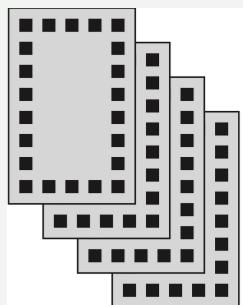
- Can make components
- Have some good sensor front-ends
- Could make regions of Cortex
(what would we do with it?)

Efficient use of event timing

- Dendrite—HMM Classifiers
(Wordspotting)
- CPG / Optimal Path Planning /
What are other efficient computations?

CREATING ANALOG BENCHMARKS

- More than just picking some typical problems
- Benchmarks ← Computing Ability



Many Analog & Mixed-Signal Computing ICs
(e.g. FPAAs)

Digital computation

High Starting Precision
&
Moderate Summation

$$\mathbf{Ax} = \mathbf{b}$$

Digital Benchmarks
• LINPACK

Analog Computation

Good Numerics
&
Moderate Starting Precision

ODEs
PDEs
Optimization
Some $\mathbf{Ax} = \mathbf{b}$

Analog Benchmarks

→ Analog & Mixed-Signal Computing Benchmarks

enables

FPGA examples
Analog Numerics
Analog Arch
Analog Abstraction
Computing Theory

Comparison,
and metric based
analog/mixed-signal
computation
design

Models of
analog
computation

	Computation
Acoustic	Command / Speech recognition from microphone signal:
Vision	Image Classification
Comm	Beamforming and Demodulation ($0.01 \times$ input freq)
Filter	10^{th} Order Programmable LPF

Algorithm
Level

Sample Acoustic Benchmark Code

`y1 = AmplitudeBPF(In)`

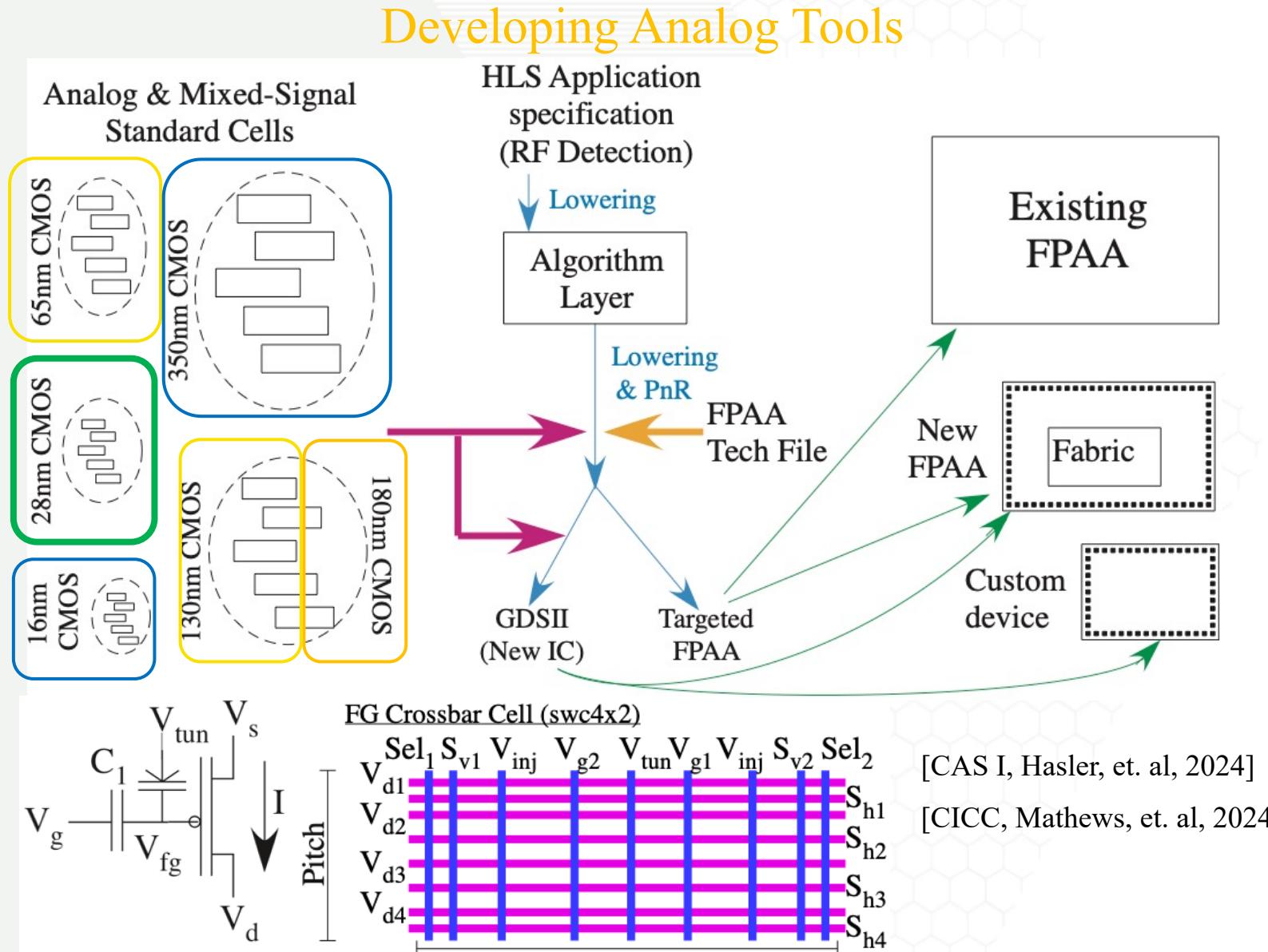
`y2 = M2V(LadderFM(input=y1,Ibias1=Ib1,Ibias2=Ib2))`

`Out = VMMWTA(input=y2,W=W1,Thresh=theta1,freq=400)`

STANDARD CELL → NEW ANALOG & NEUROMORPHIC ICS

Neuromorphic Tools....

Developing Analog Tools



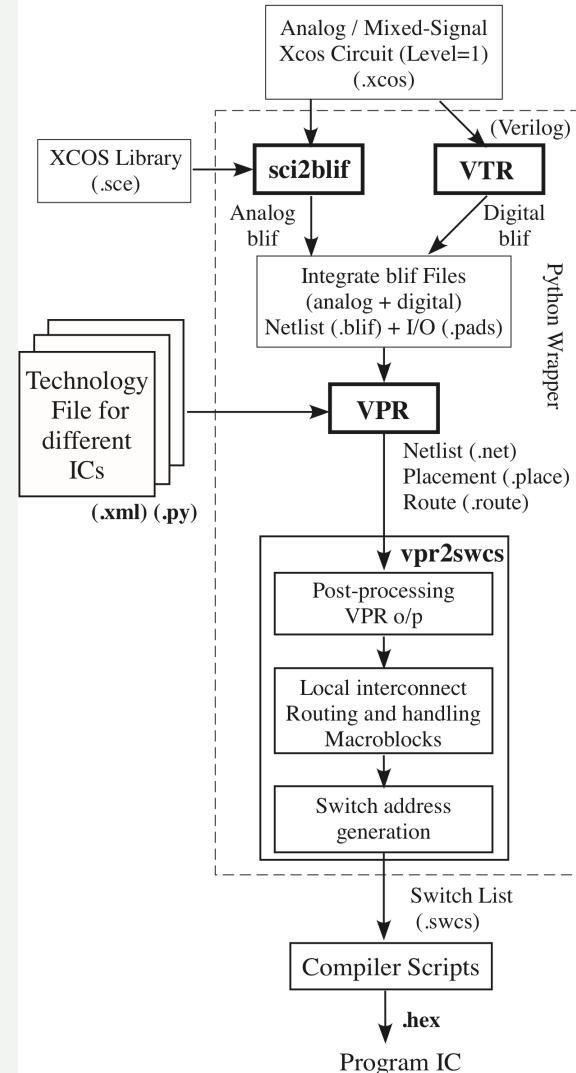
What are neuromorphic standard cells?

- Neurons (e.g. HH) and channels
- Synapses (beyond crossbar)
- Dendrites
- Further components

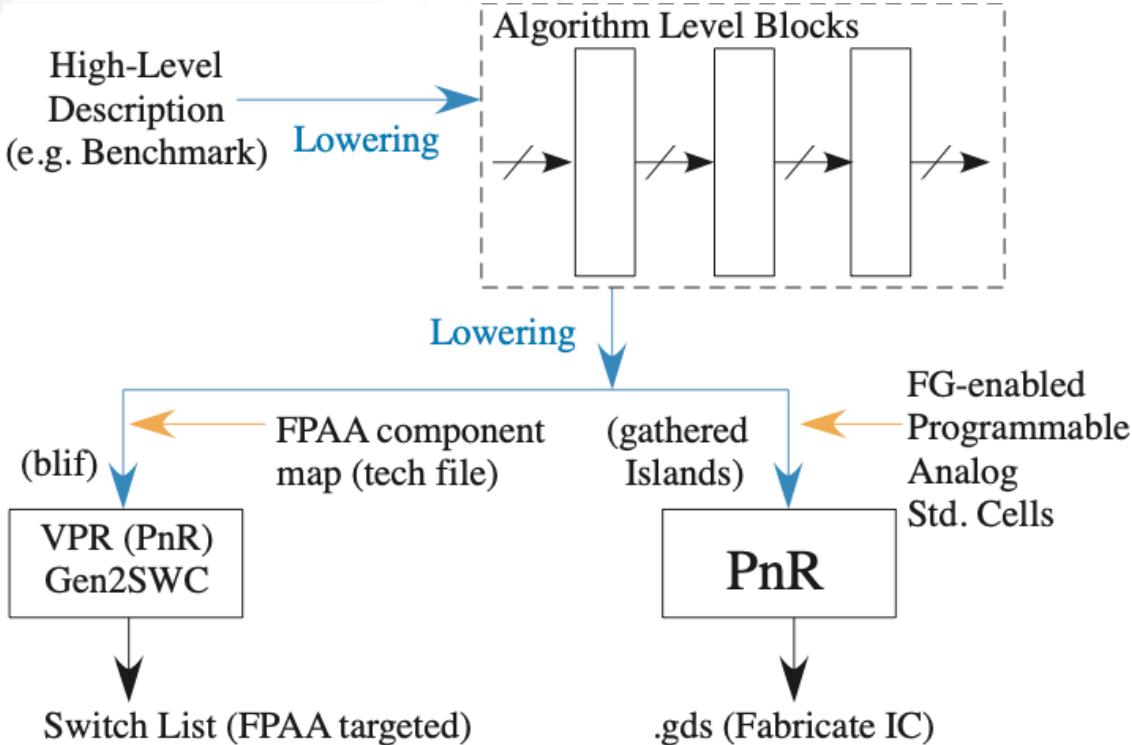
Assumes we have engineering models of neural computation....

ANALOG / NEUROMORPHIC SYNTHESIS TOOLS

First FPAA (Scilab) tools



Developing Analog Tools



Tools are being developed / generalized to:

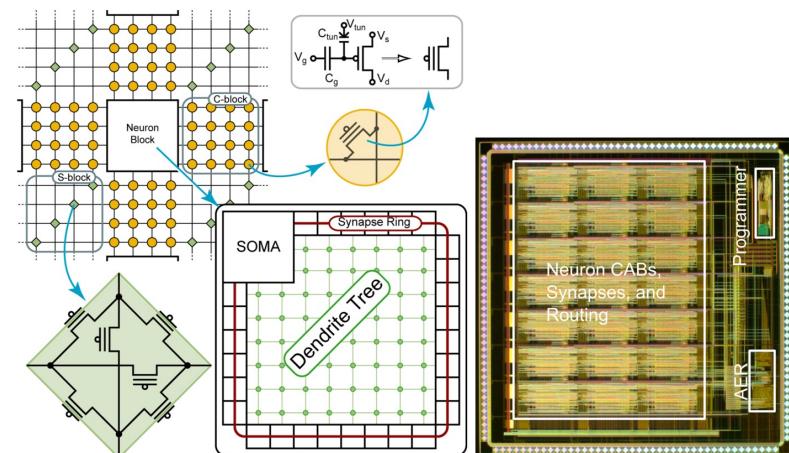
- Target an existing / defined FPAA
- Generate a custom IC
- Generate a new configurable FPAA fabric

Neuromorphic Tools....

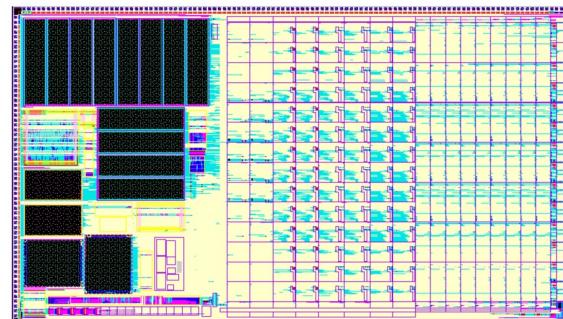
Assuming:

- Neuromorphic Computational Models
- Neuromorphic Standard Cell Library

Tool flow *should* be similar
(Neuron FPAs, 2012 & on)

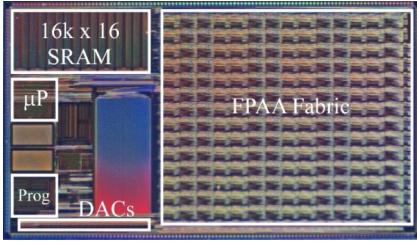
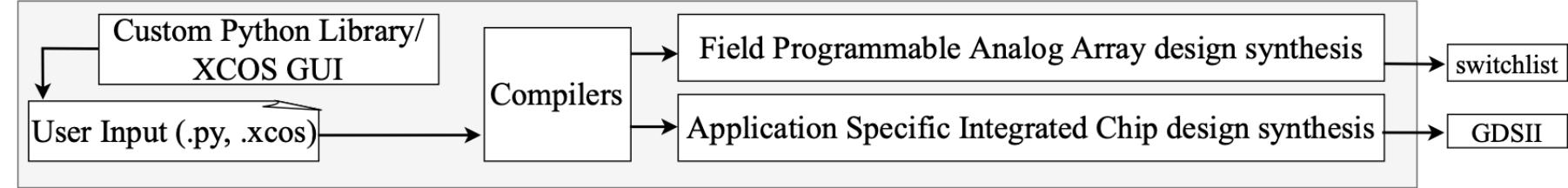


SoC FPAA with Neuron CABs

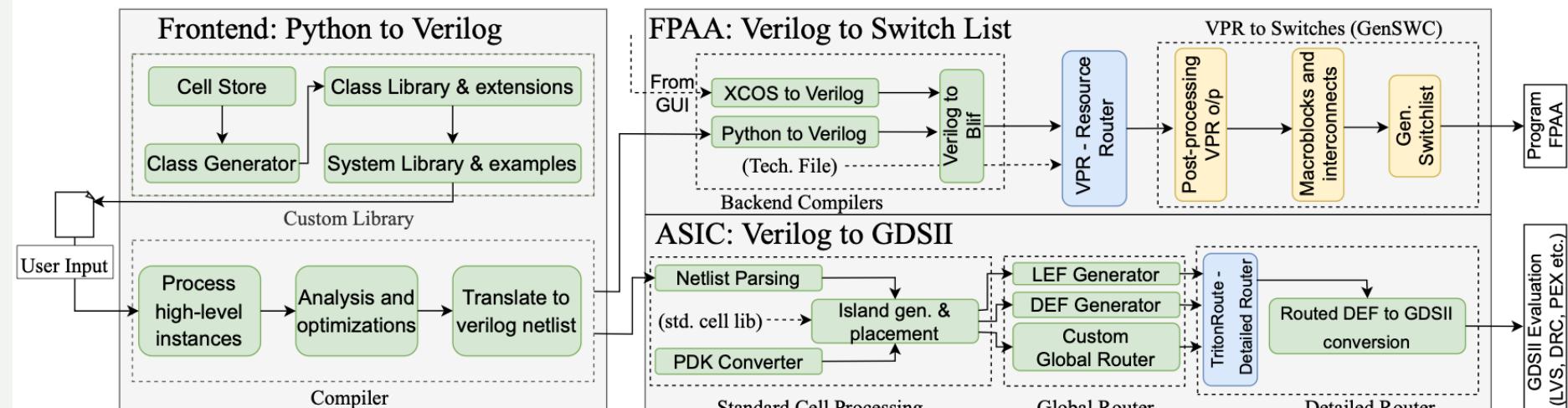
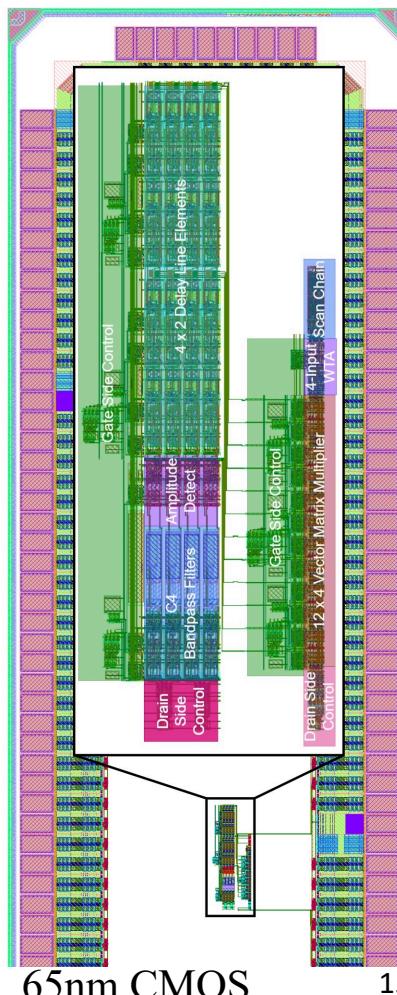


PYTHON + XCOS ANALOG / MIXED SIGNAL TOOL

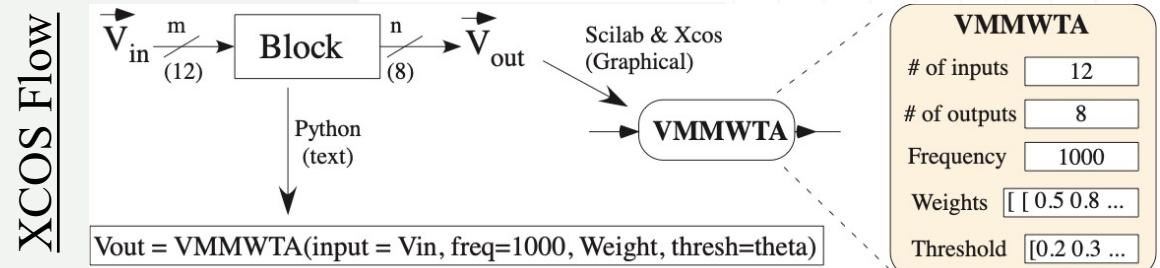
Target an existing FPAA



Generate a custom IC



Novel Contribution Our Existing FPAA Tools Pre-existing open-source tool

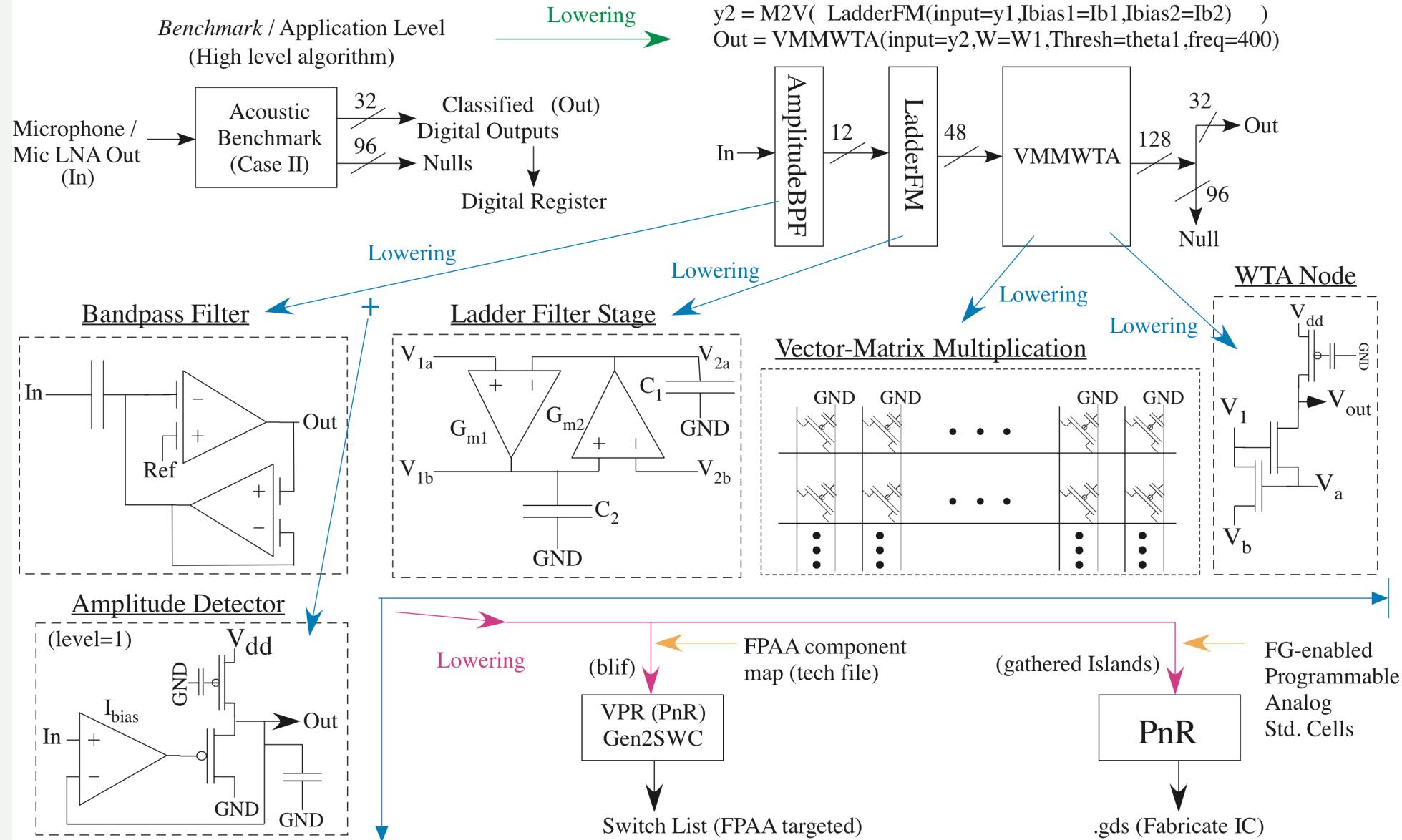


[Ige, et. al, 2023]

EXAMPLE SYNTHESIS: ACOUSTIC, CASE II

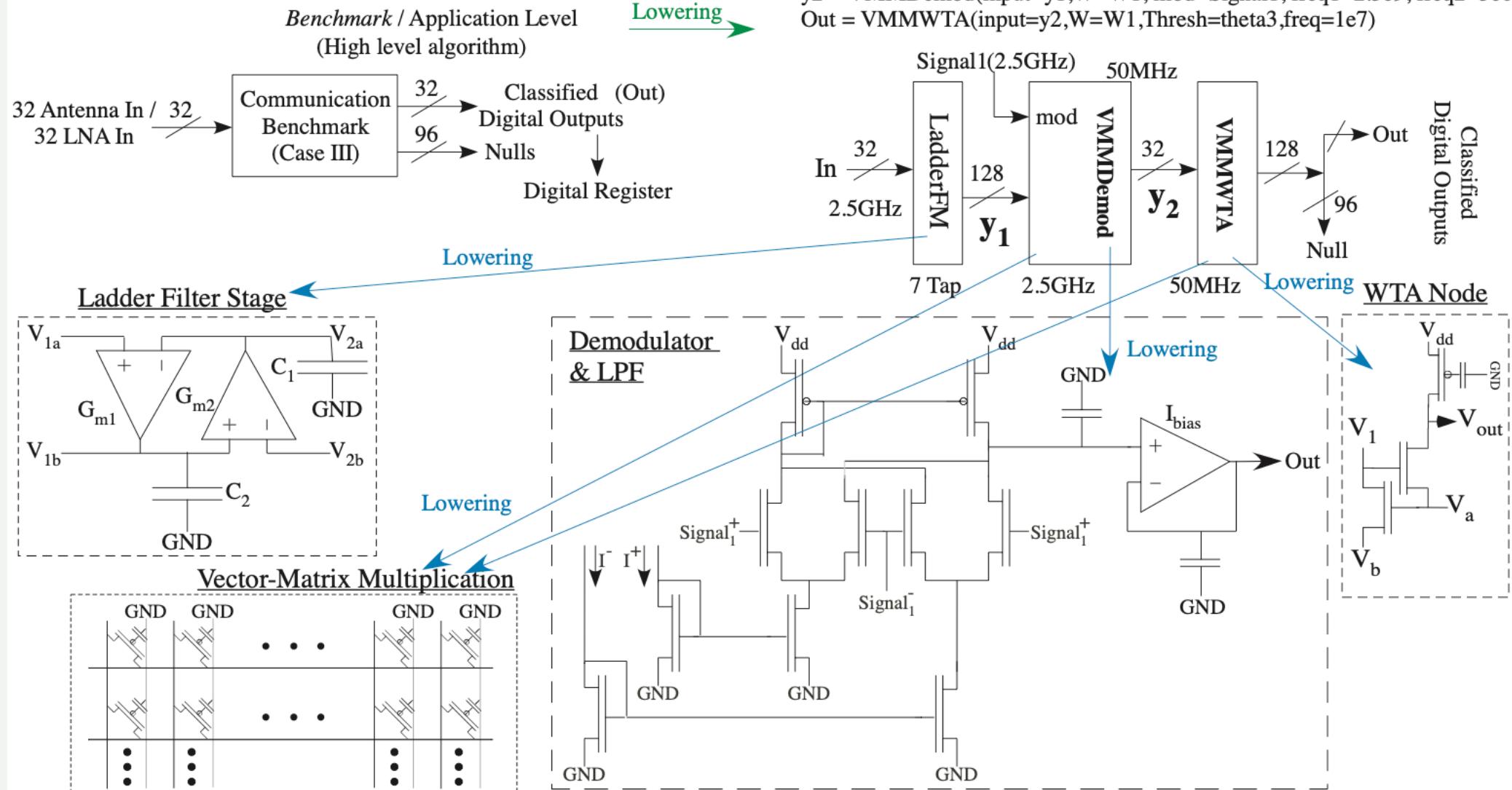
Case II Acoustic Benchmark: 32 out.

BPF filterbank, 3 delay stages, classifier, 3x nulls

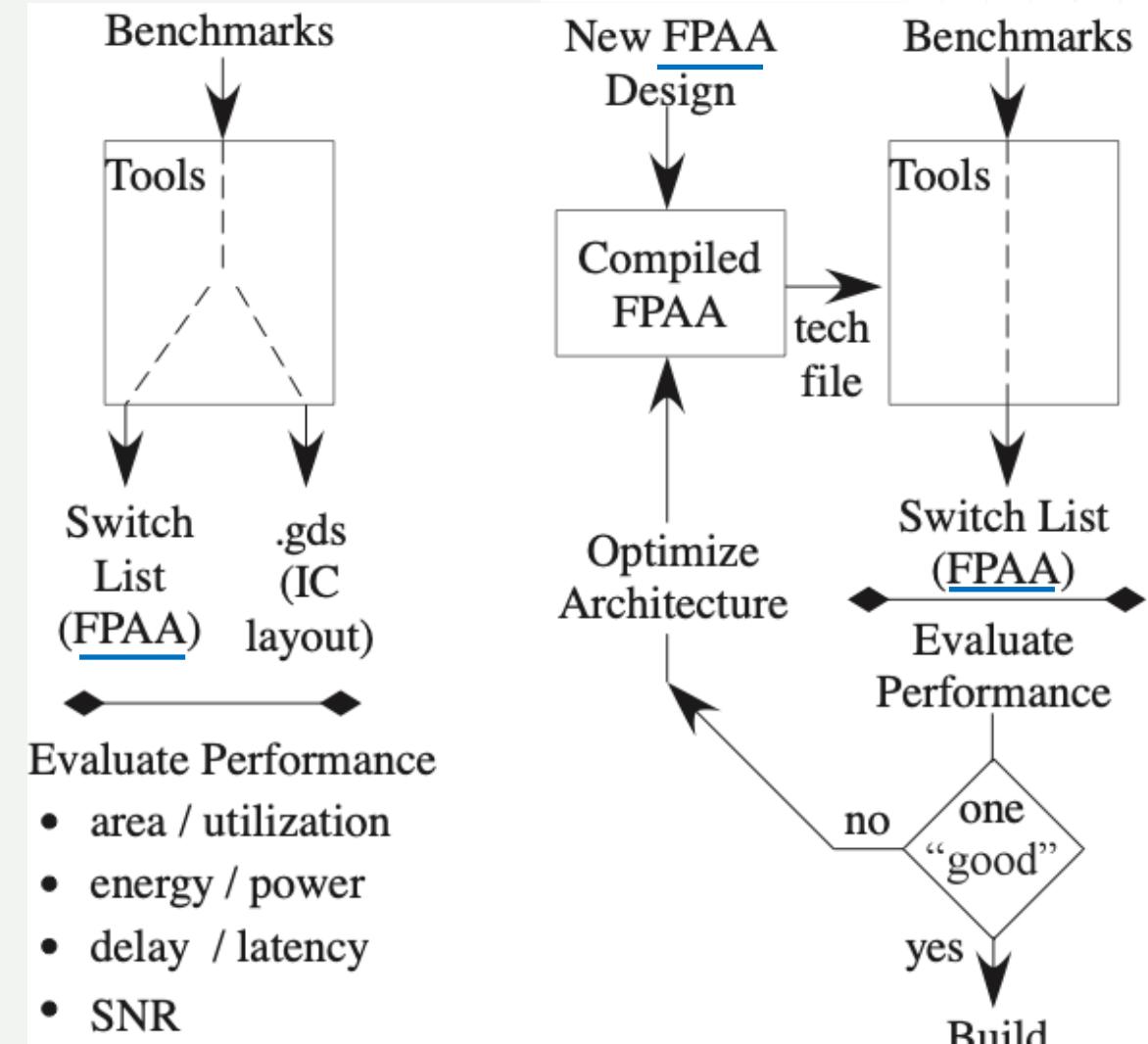


COMMUNICATION BENCHMARK (CASE III)

Case III Communication Benchmark: 32 inputs, 2GHz, 20MHz output classification bandwidth



TOOLS FUEL THE OPPORTUNITIES



[Hasler & Hao, TRETS / FPL 2023]

Analog Computing Tools

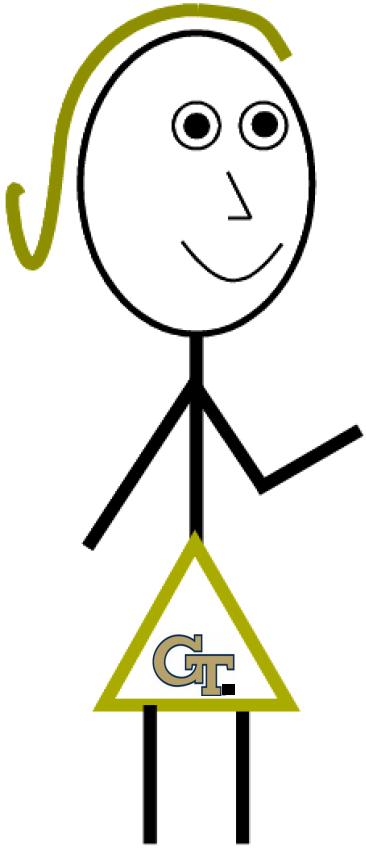
- Developing & Developed
- Becoming part of IC Design Flow
- Analog Benchmarks
- Analog Synthesis

Programmable & Configurable Edge (& elsewhere) ML devices

Neuromorphic Tools

- These remain an open question
- What can be borrowed from Analog Computing tools?
- What new concepts required?

ADDITIONAL RESOURCES

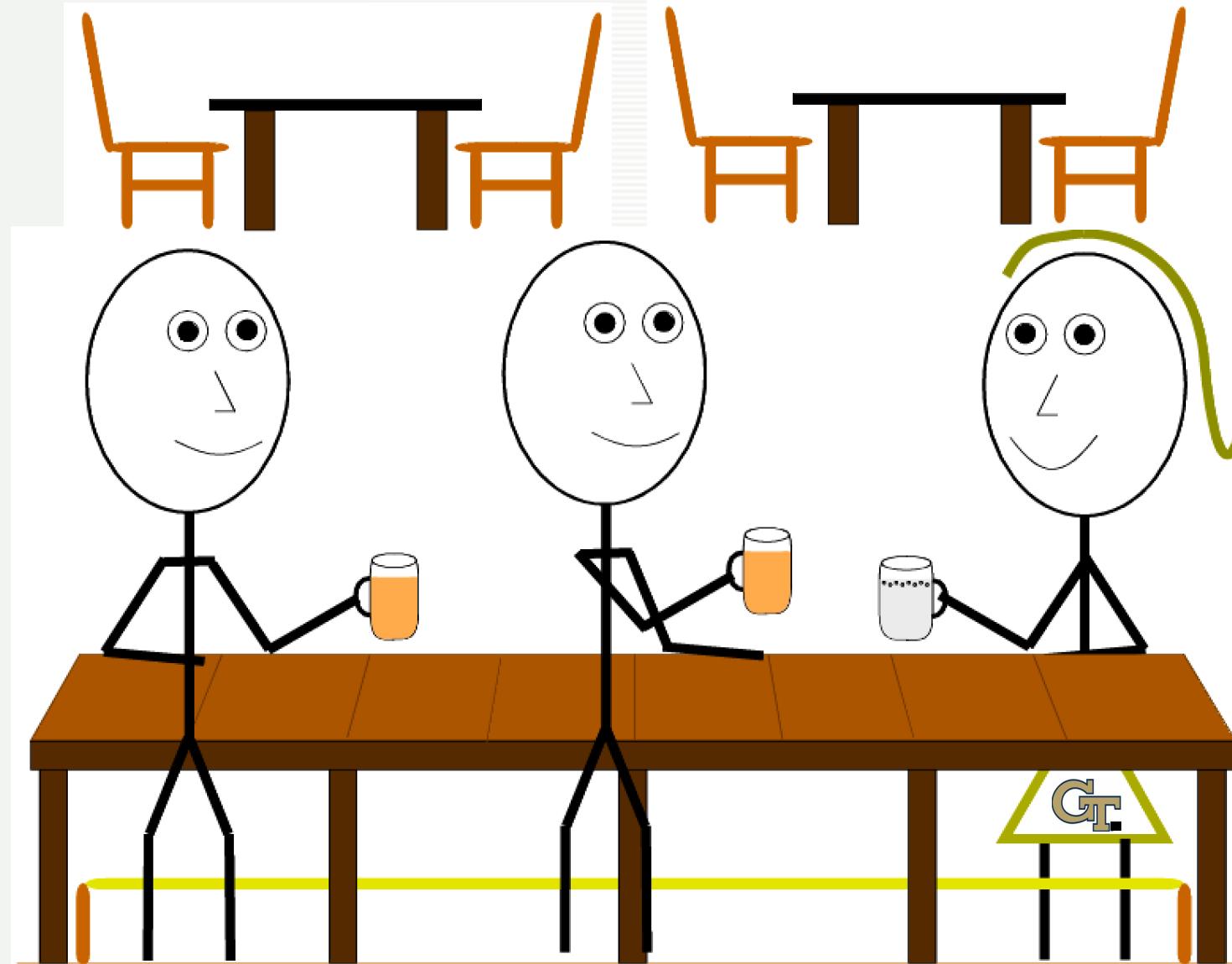


Videos:

- J. Hasler, FPAA: History, Development, Classification: <https://youtu.be/2lsz9gi8Oz8>
- J. Hasler, Future of FPAA opportunities: <https://youtu.be/rpSdb88ubfk>
- J. Hasler & A. Natarajan, Intro Open-Source FPAA Toolset: <https://www.youtube.com/8SVdhztVroc>
- J. Hasler, Historical FG Perspective, <https://youtu.be/R8iV01KZch4>
- J. Hasler, FPAA Enabling Physical Computing, <https://youtu.be/IGzinnykZIw>

FPAA on-line Workshop: <http://hasler.ece.gatech.edu/FPAAWorkshop/index.html>

PHYSICAL COMPUTING



Further questions
are definitely
welcome and
appreciated.