Other Rogues - Analog, Reconfigurable, and Neuromorphic Accelerators and Related Tools

FPAA Content provided by Dr. Jennifer Hasler, ECE and CRNCH faculty

CREATING THE NEXT MOORE'S LAW

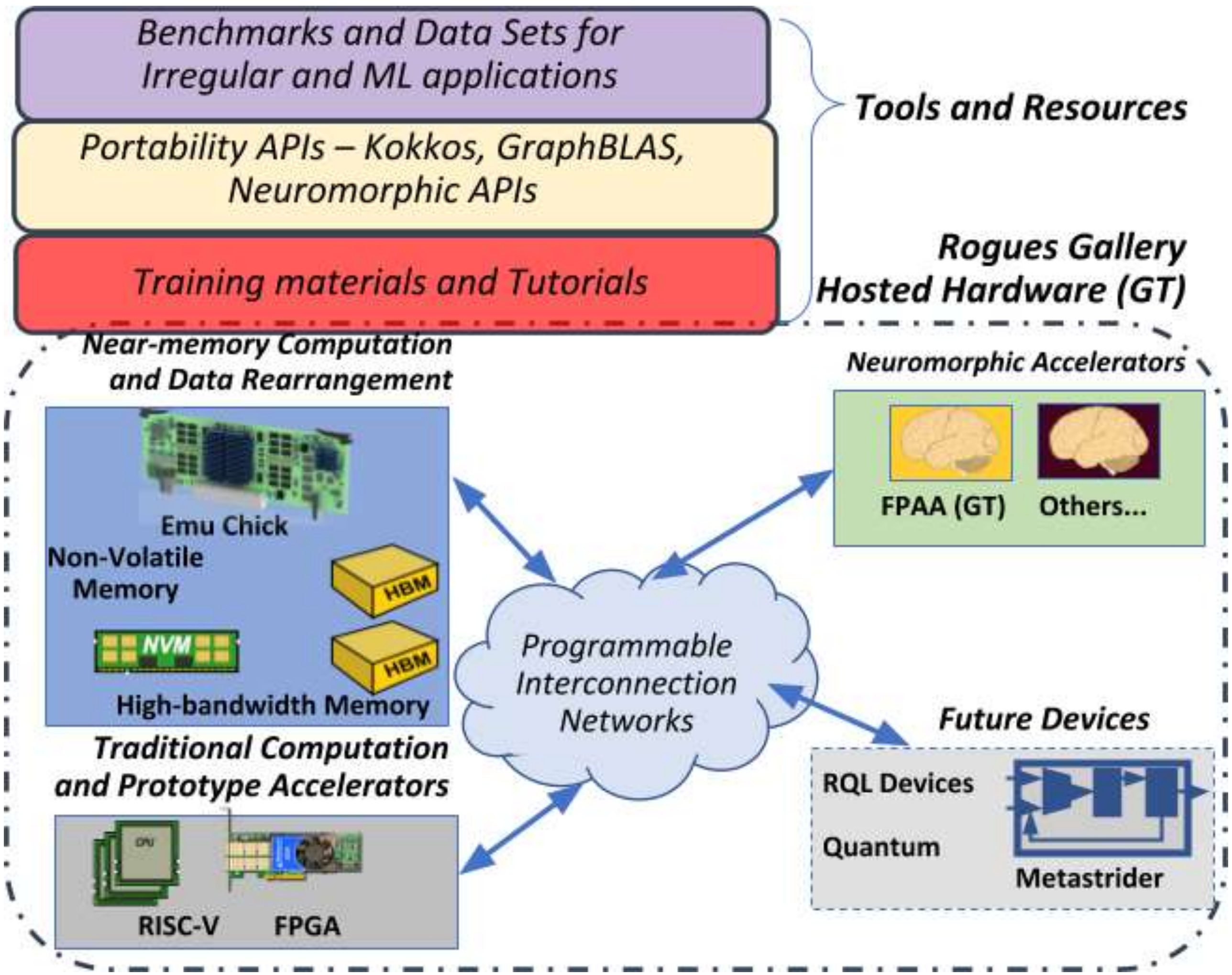
Georgia Tech | Center for Research into Novel Computing Hierarchies

For more information, visit our website at **www.crnch.gatech.edu** or send email to **crnch@gatech.edu**

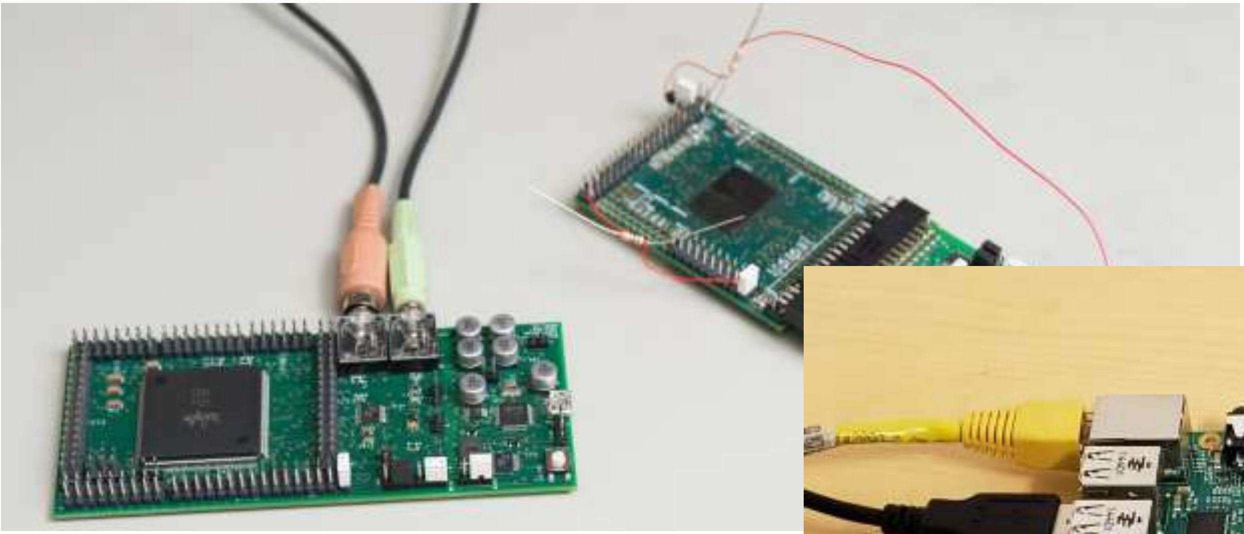
Outline

- Neuromorphic Computing
  - Field Programmable Analog Arrays
  - Other future platforms
- Reconfigurable Computing
- Benchmarking and Tools

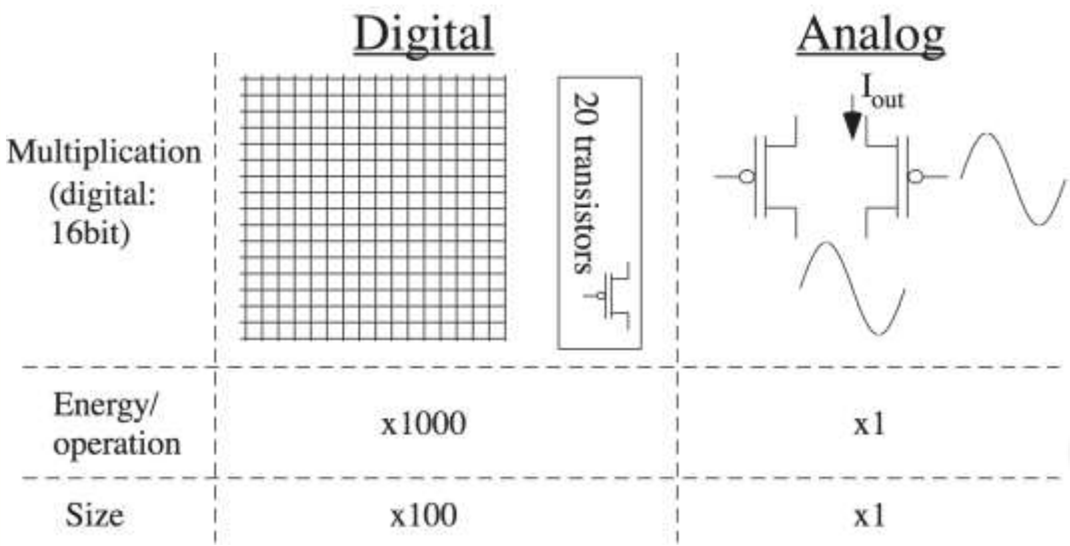
Rogues Gallery Recap



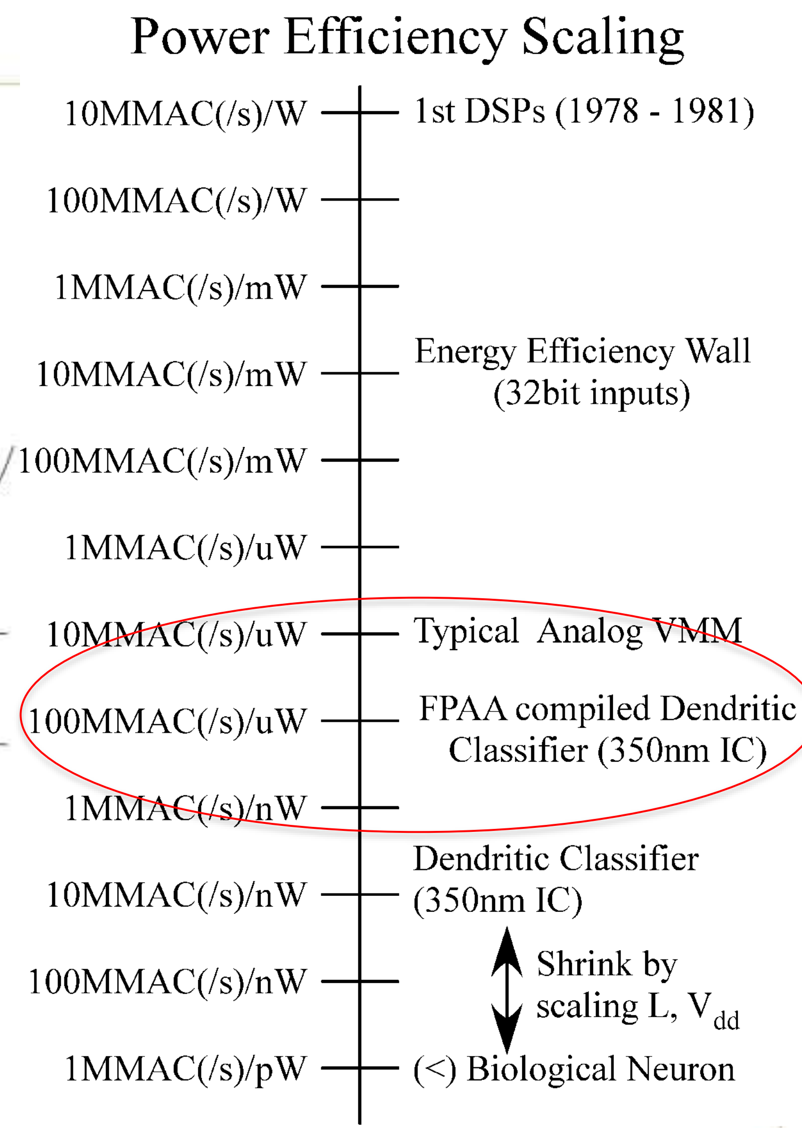
Field Programmable Analog Array (FPAA)



Why Physical / Analog Computing?



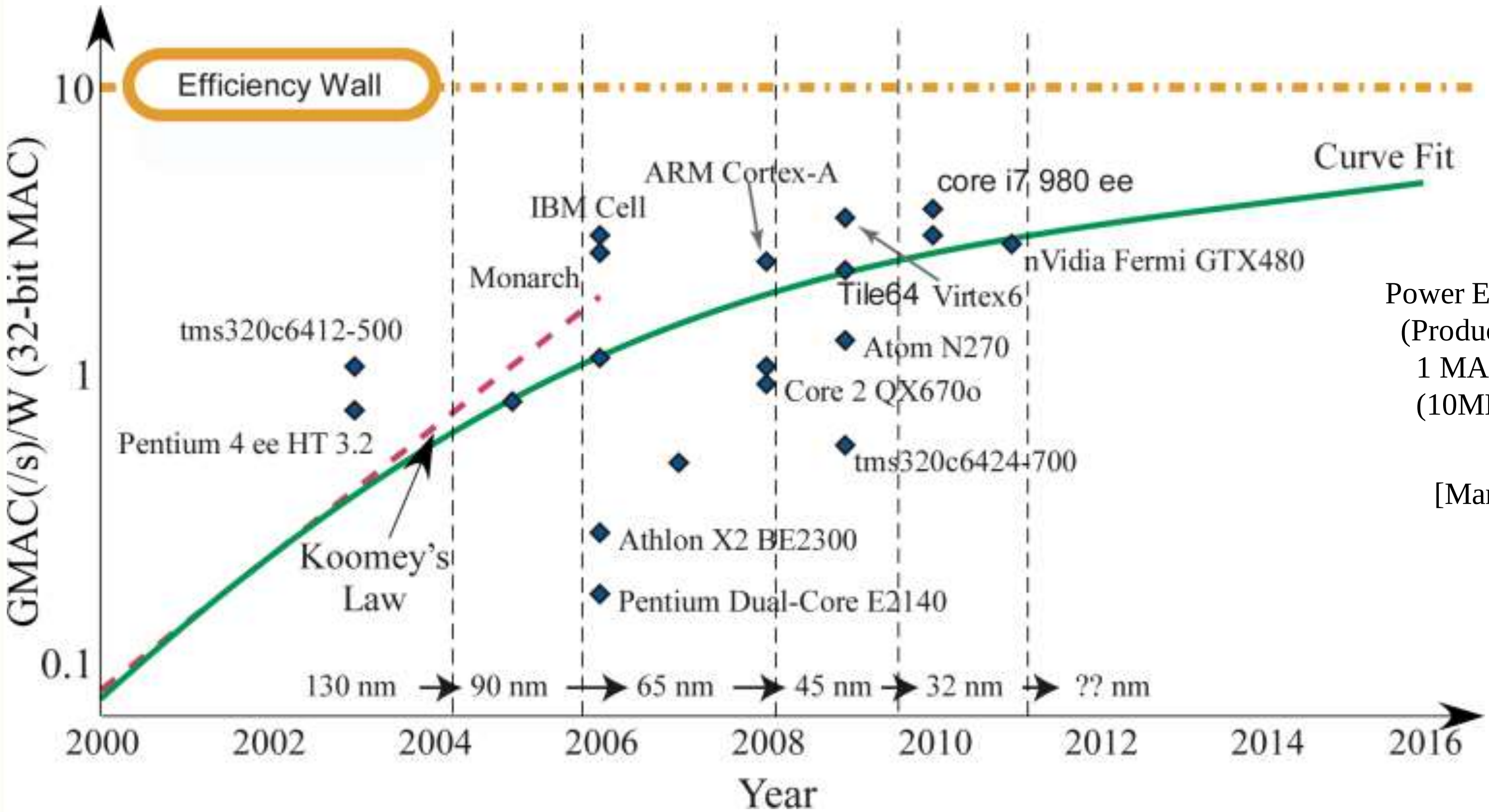
Expansion of C. Mead's original hypothesis, 1990



[Hasler and Marr 2013]

Why Physical / Analog Computing?

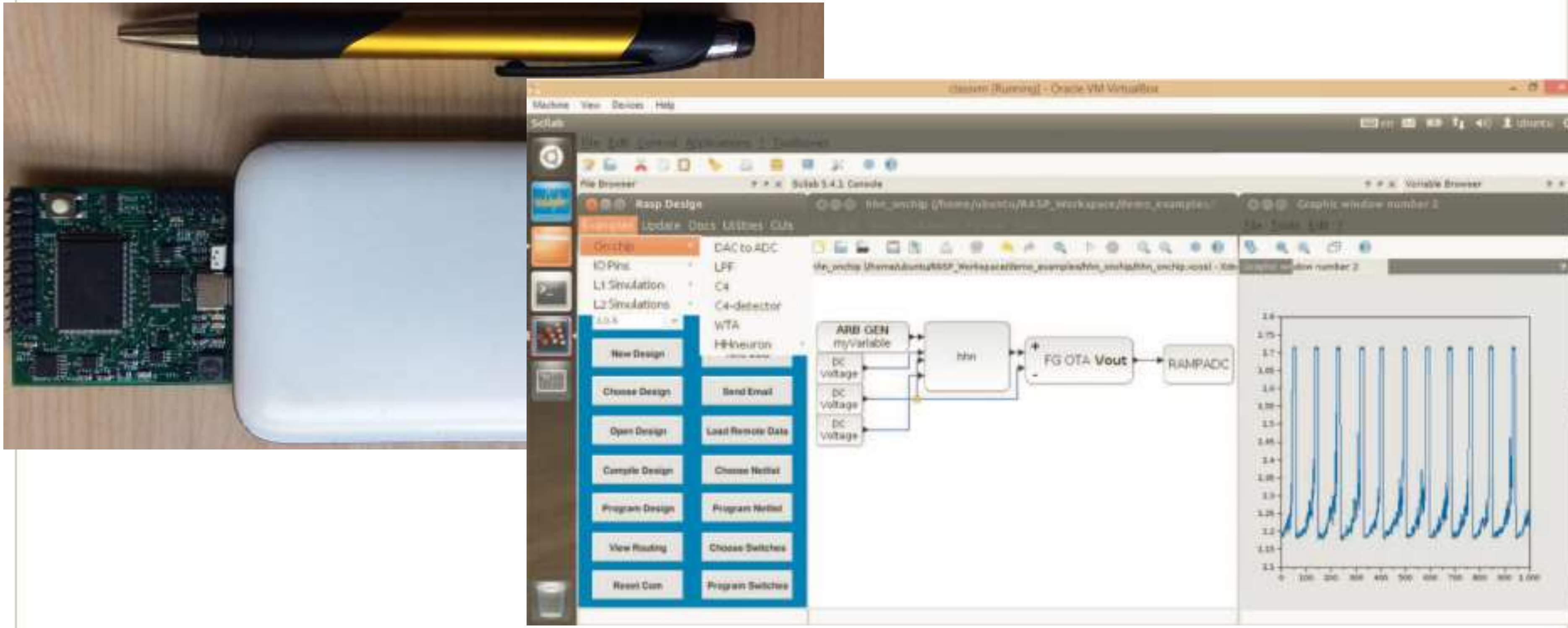
Digital Hitting Limits of Power Efficiency – Koomey's Law



Power Efficiency wall
(Production ICs)
1 MAC in 100pJ
(10MMAC/mW)

[Marr, et. al, 2013]

FPAAs enable Ultra-low Power Computing

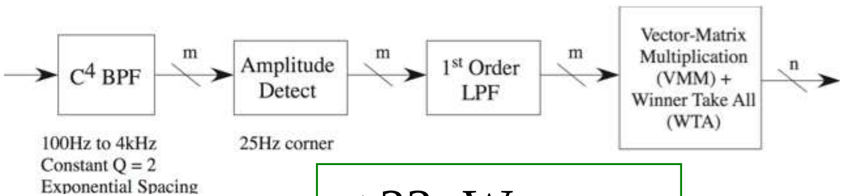


The FPAA provides a mixed analog/digital platform with a general and configurable design that can be used create secure and legacy resistant mixed-signal devices

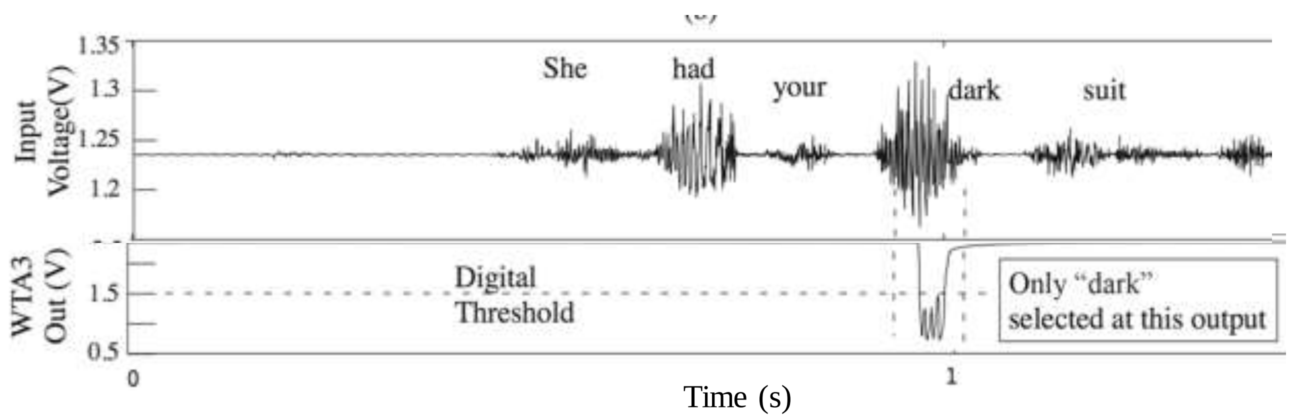
FPAA Classification Task

Accoustic Classification

Ultra-low energy computation



< 23µW power



Embedded learning & classification: 20-30µW on full, 1s Nzero database (GOMAC 2016)

FPAA Classification Examples

1. Built on Analog Vector-Matrix Multiplication (VMM), including in routing, Winner Take All (WTA), etc.

2. Initial Command-Word Classification (Hand-tuned Weights)   (2015)

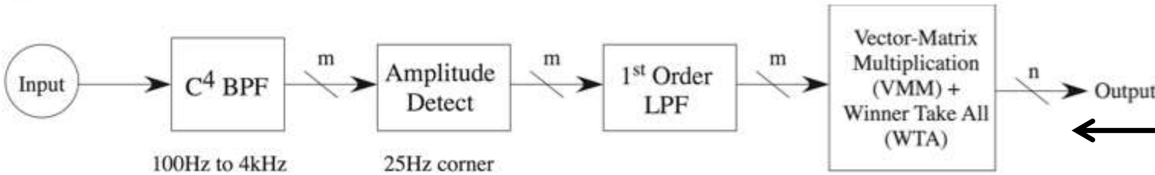
3. Biomedical classifier (Hand-tuned weights) and computational measures (Knee sounds, Heart monitoring)
4. Speech – No-Speech Detector (Hand tuned weights)

5. Acoustic Classification and Learning:
- Developed Theory and Training Algorithm for VMM+WTA Classification
- Trained using modified Lincoln Lab's acoustic data set for Nzero program.
- Yielded correct detection on all sets

All of these classification tasks can be performed on a device that can fit in your pocket!

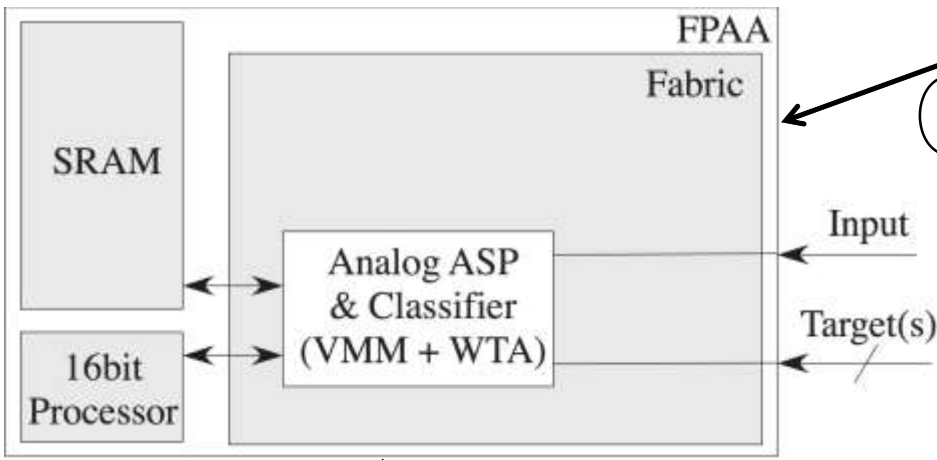
More Analog Classifier (VMM+WTA)

Classification + On-Line Training



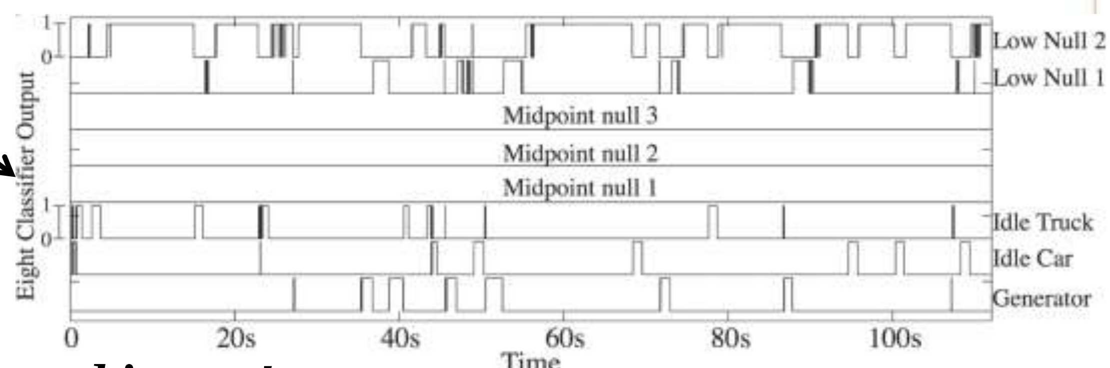
1

An ML-inspired acoustic classifier is 1) defined in macro block form, 2) compiled to the FPAA fabric and 3) used for audo classification on MIT-LL's Nzero dataset



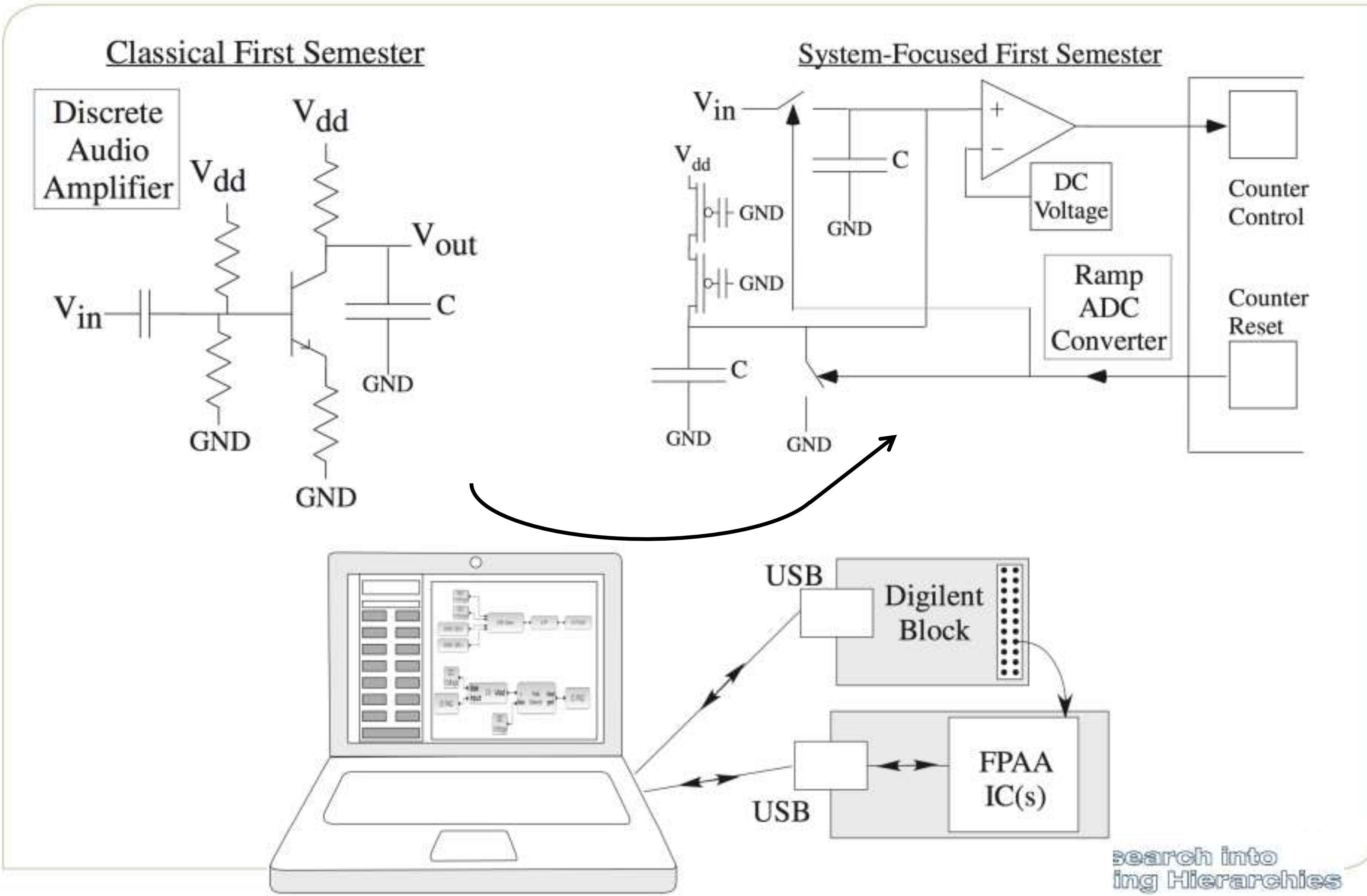
2

3

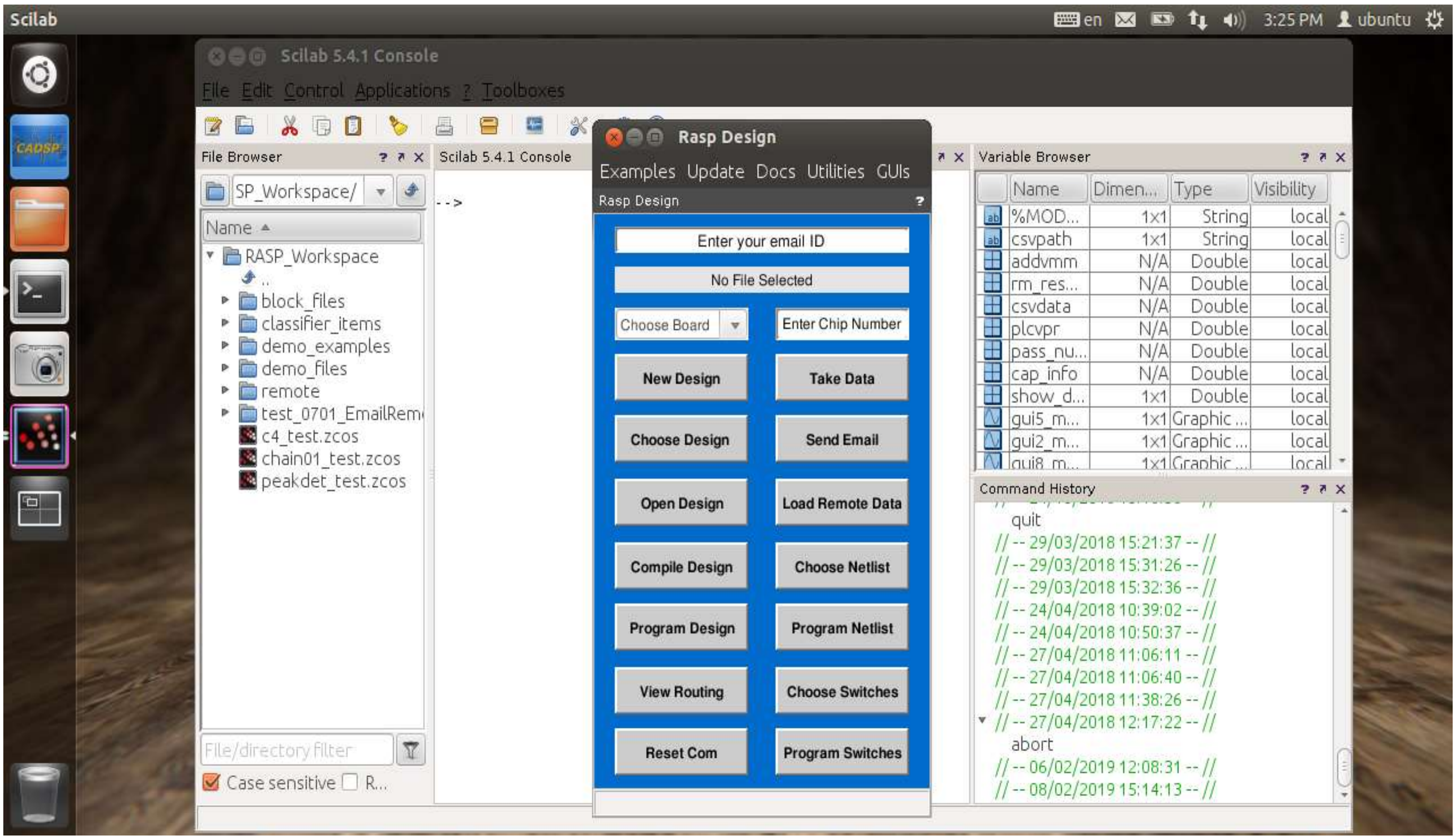


100% classification rate for tested inputs!

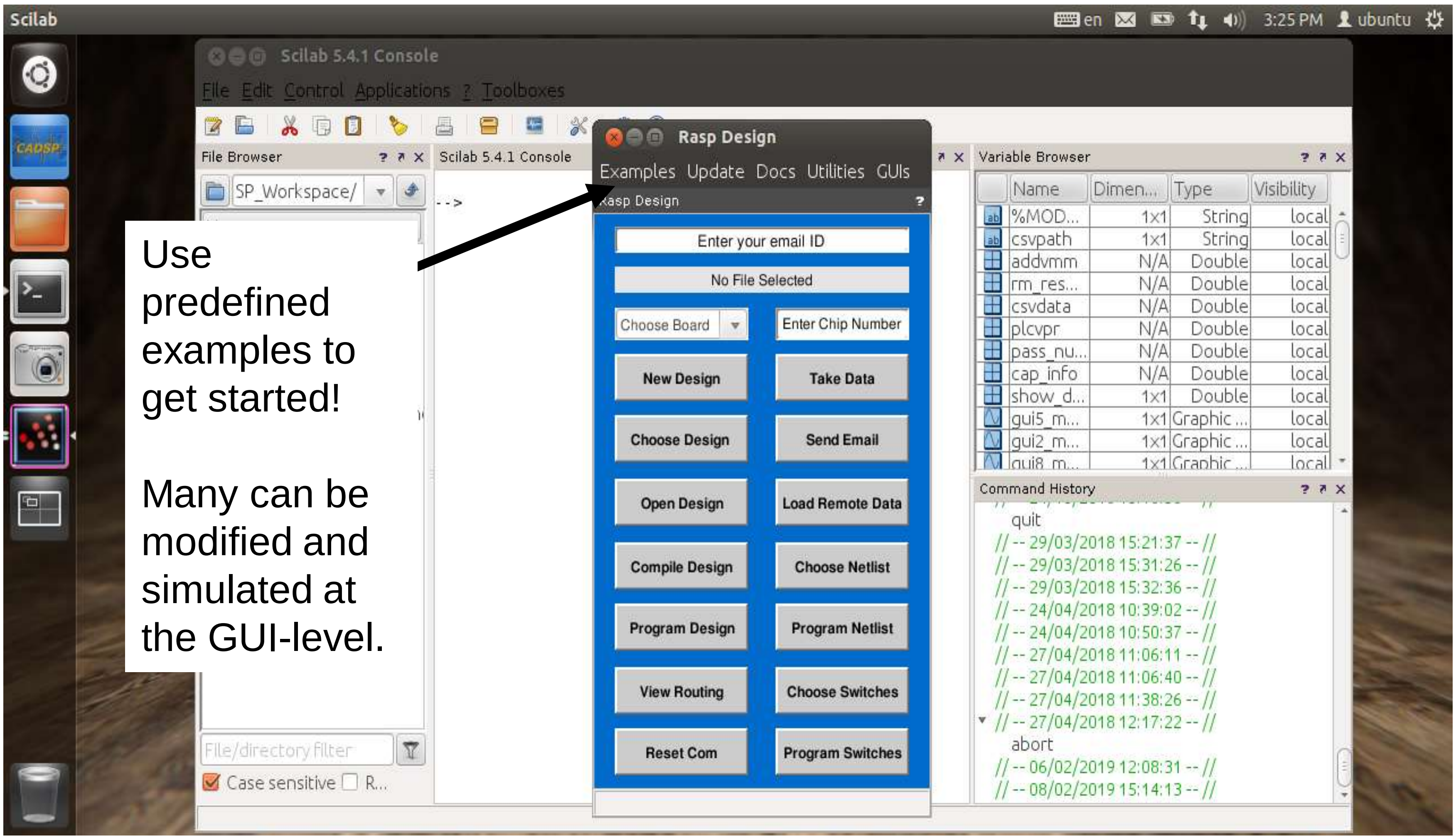
SoC FPAAs in the Classroom



FPAA Toolflow and Examples



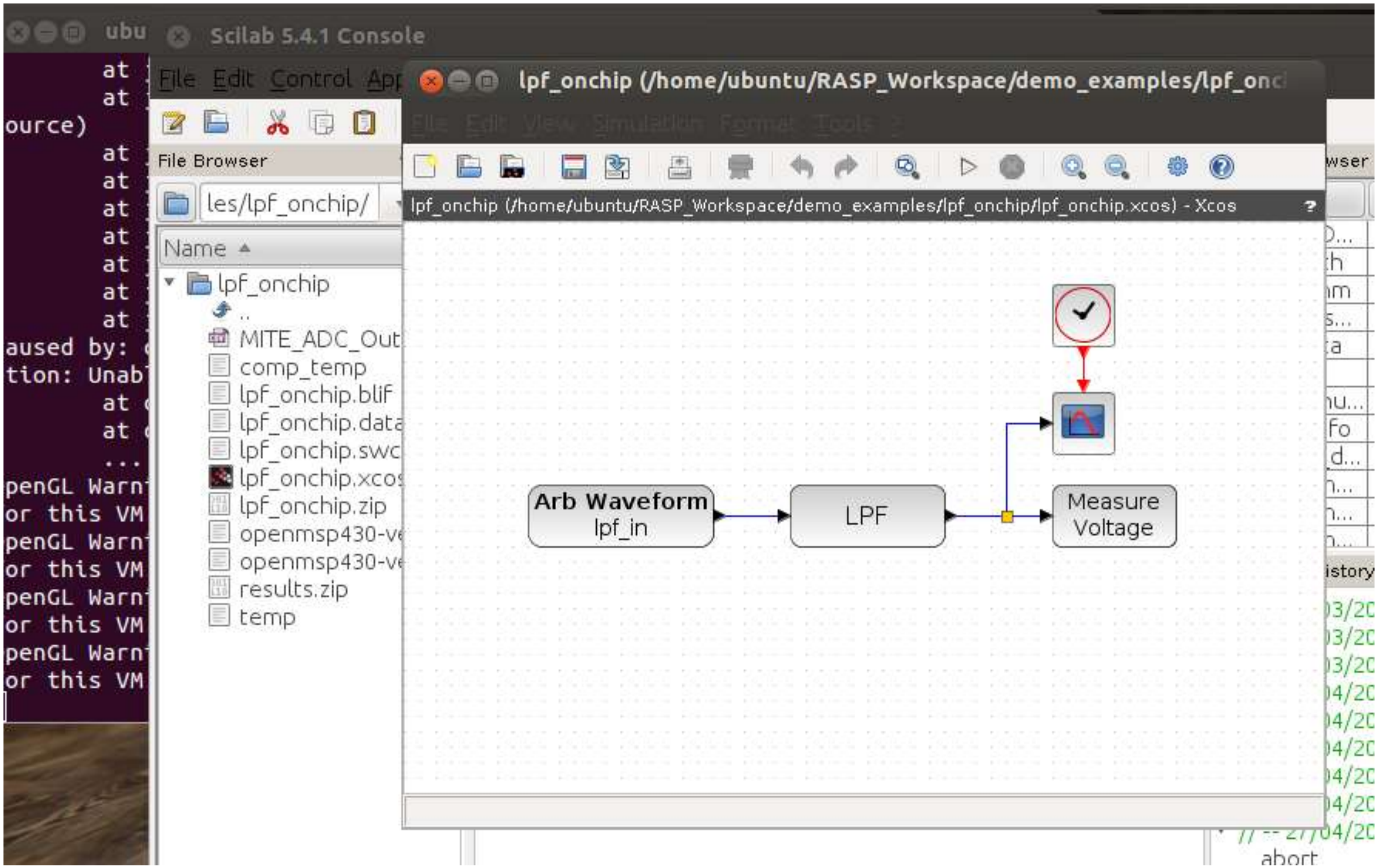
FPAA Toolflow and Examples



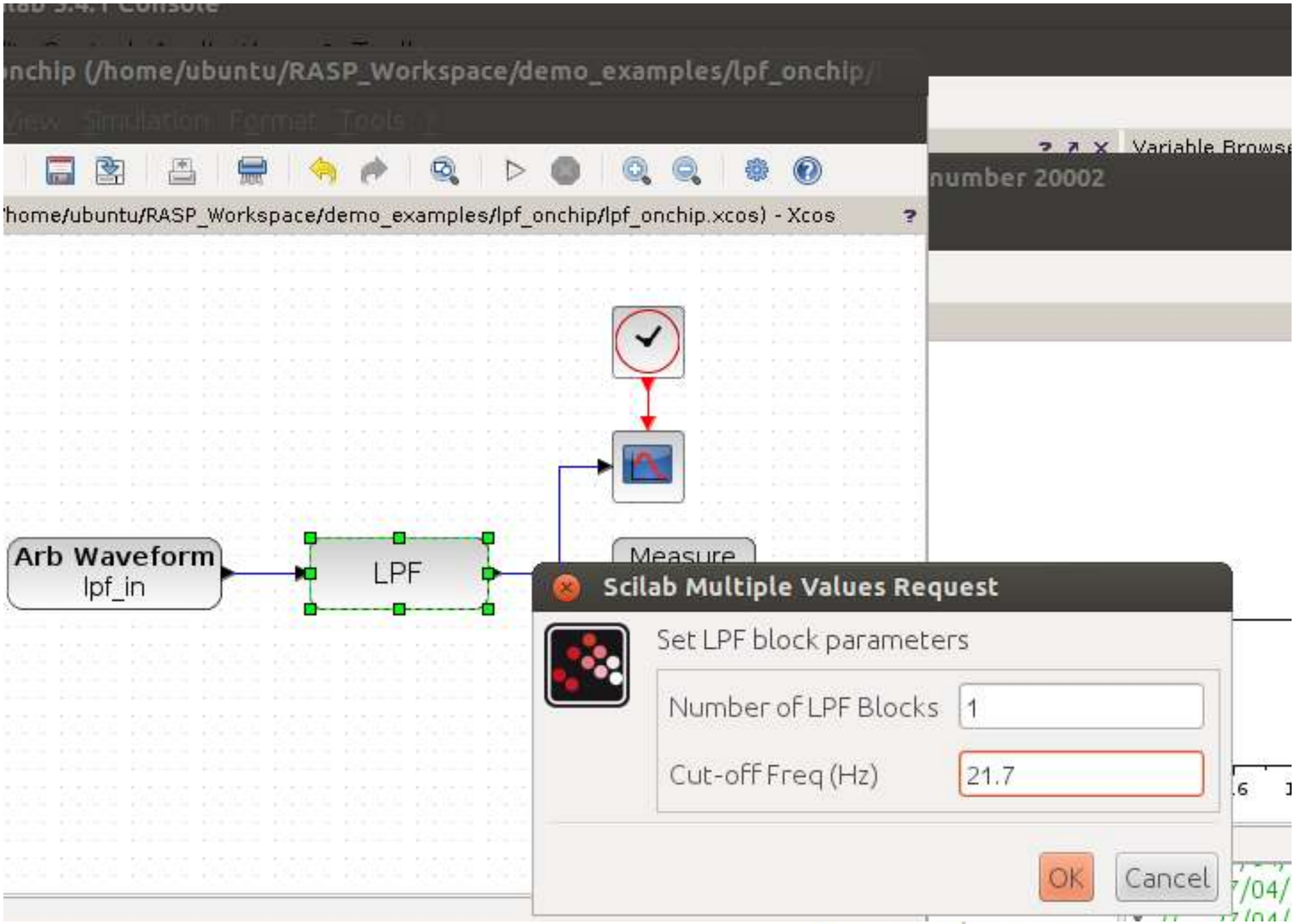
Use predefined examples to get started!

Many can be modified and simulated at the GUI-level.

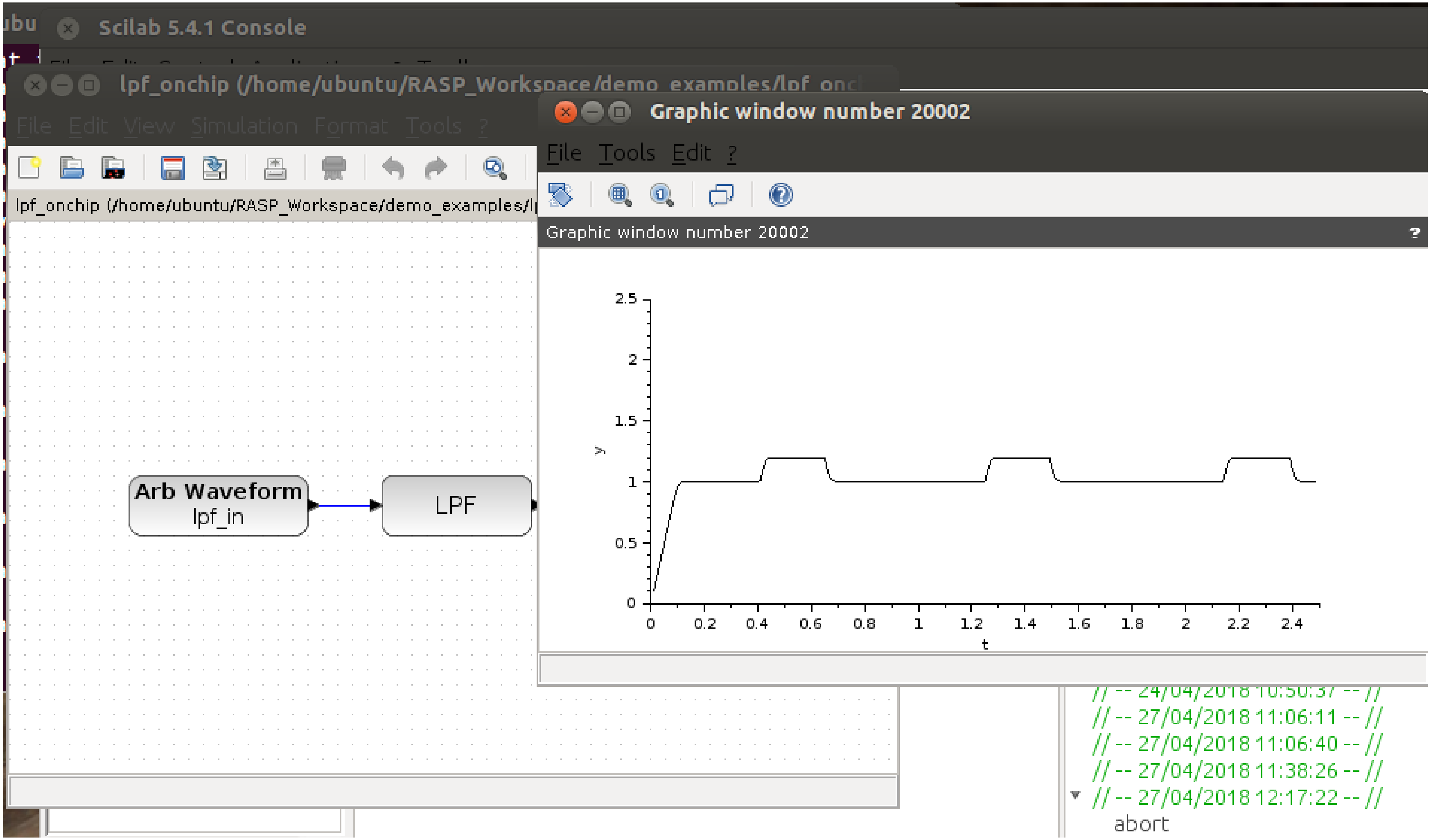
FPAA Toolflow and Examples – LPF example



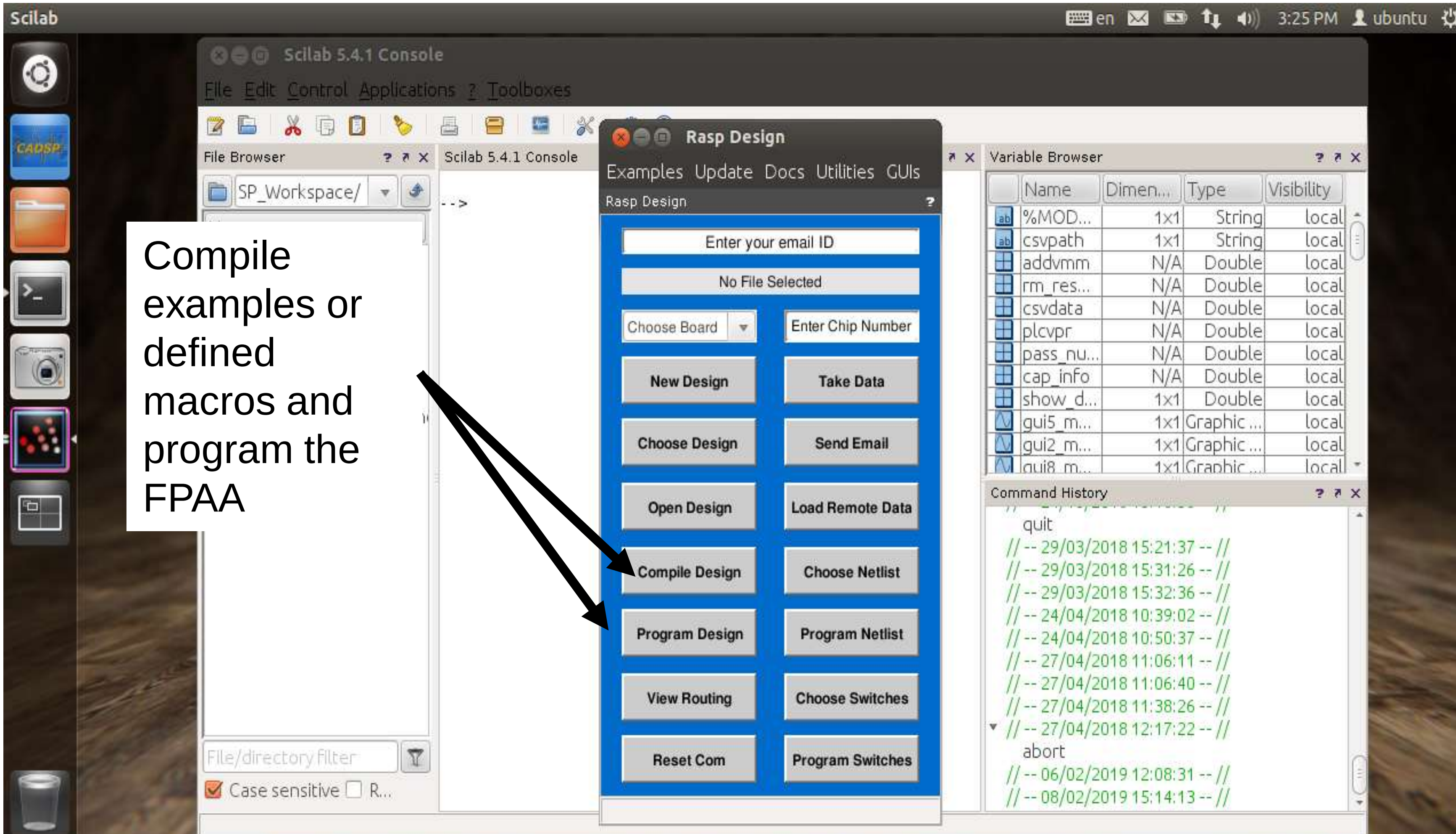
FPAA Toolflow and Examples – Macro block inputs



FPAA Toolflow and Examples - Simulation of mixed digital/analog blocks

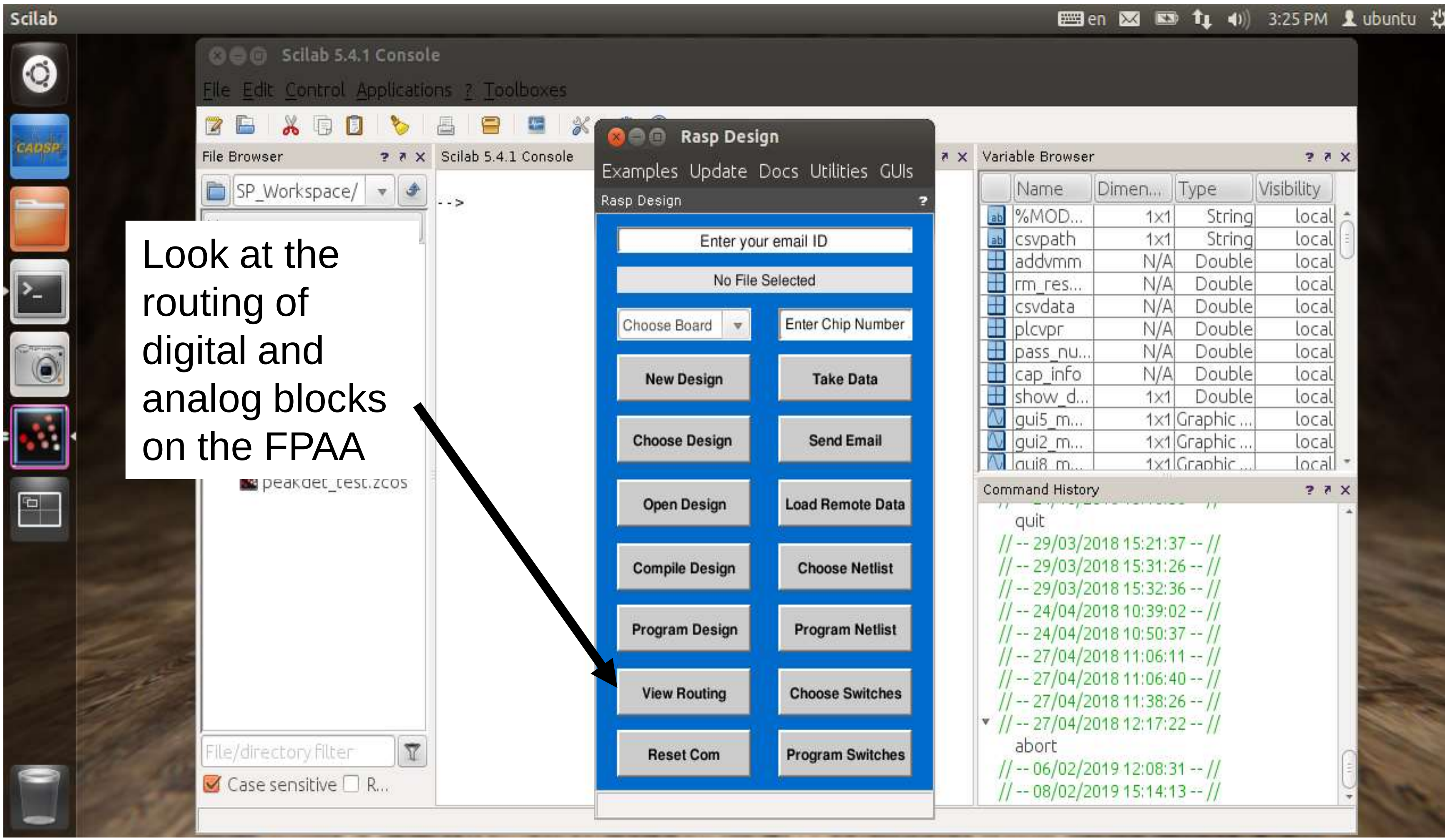


FPAA Toolflow and Examples



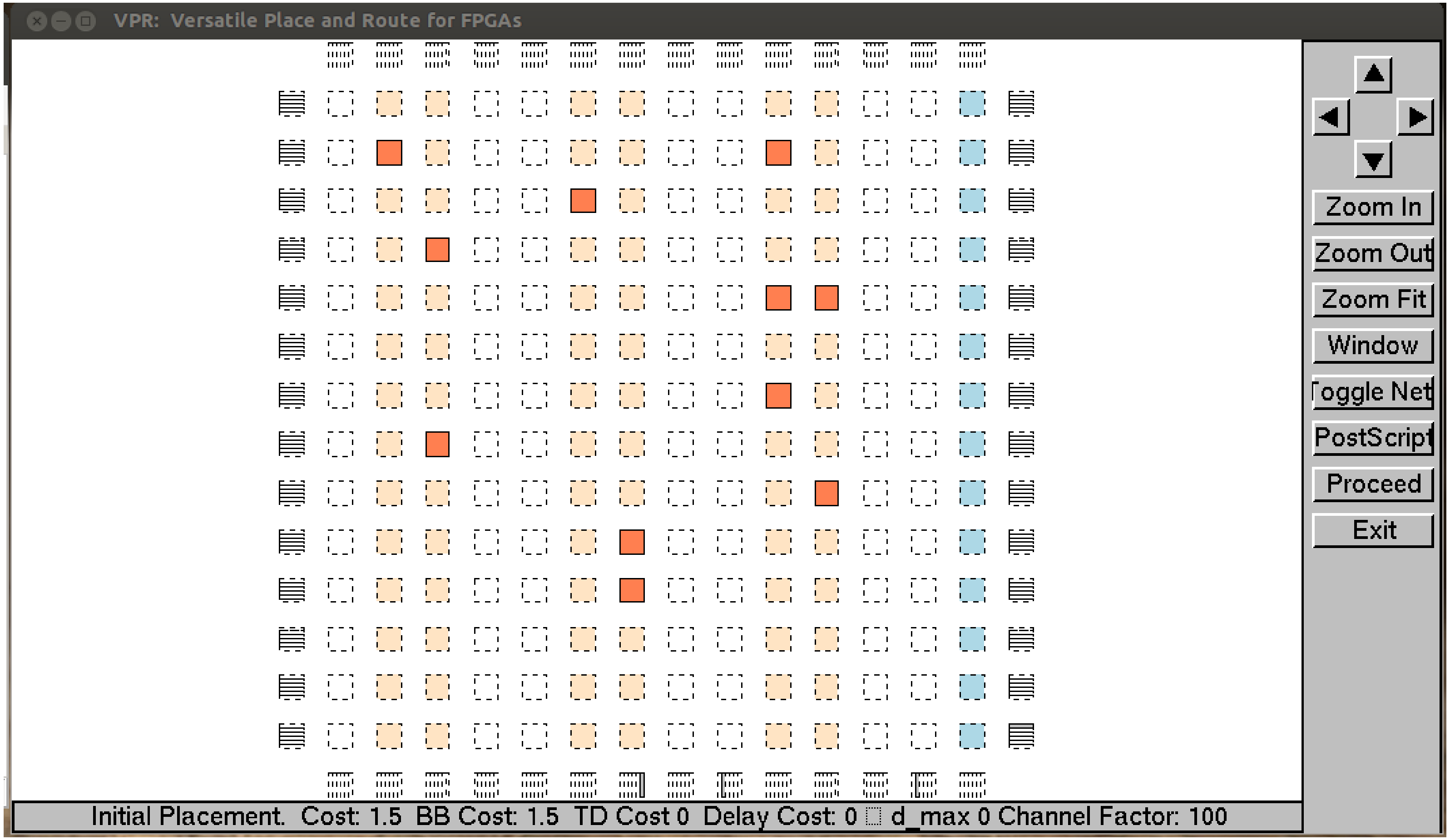
Compile examples or defined macros and program the FPAA

FPAA Toolflow and Examples



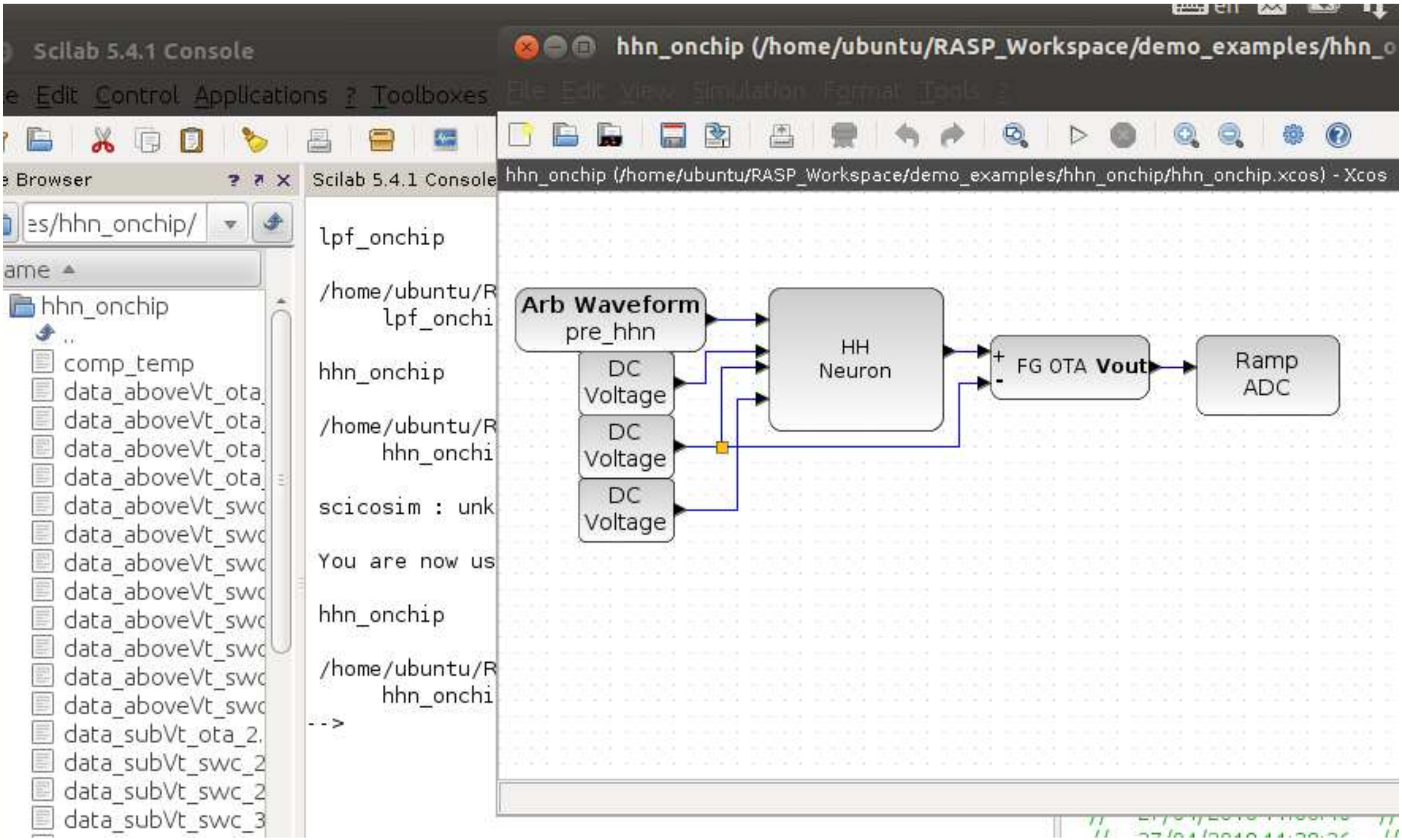
Look at the routing of digital and analog blocks on the FPAA

FPAA Toolflow and Examples

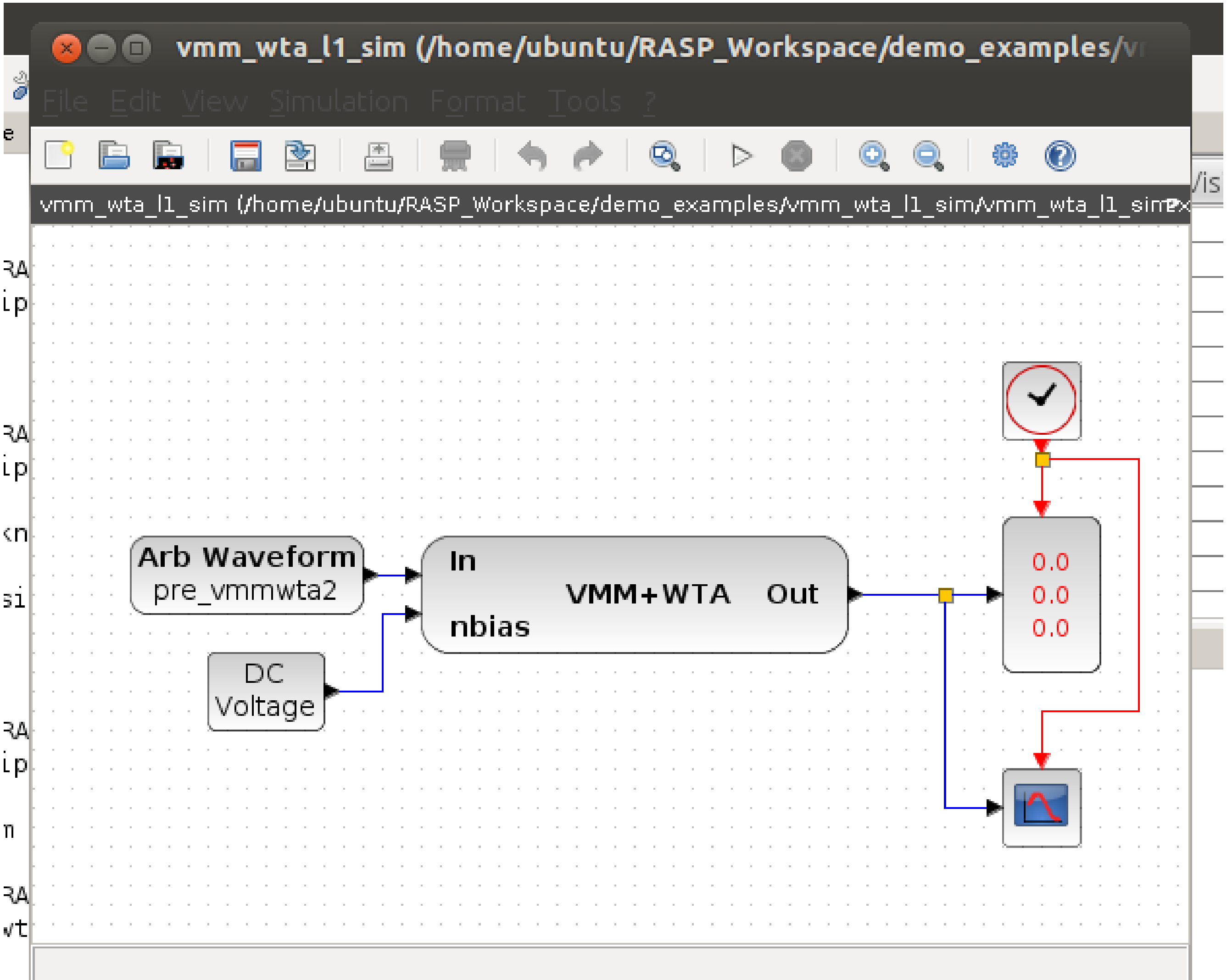


Routing diagram shows two linked hhNeuron models mapped onto the analog elements of the FPAA

FPAA Toolflow and Examples - hhNeuron

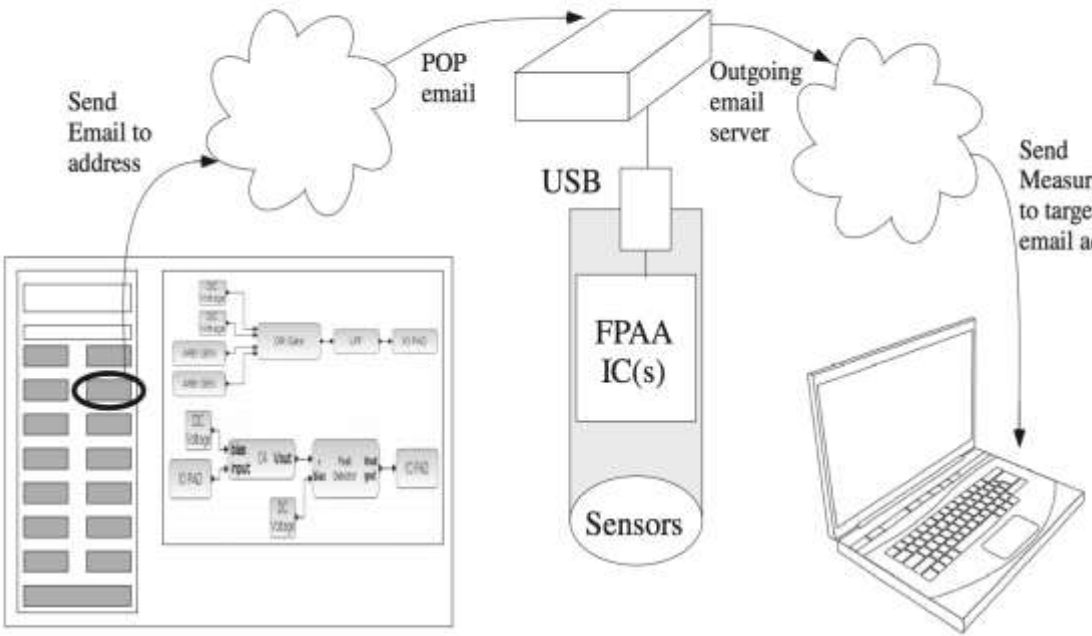


FPAA Toolflow and Examples – ML elements



FPAA Infrastructure Details

Remote FPAA System



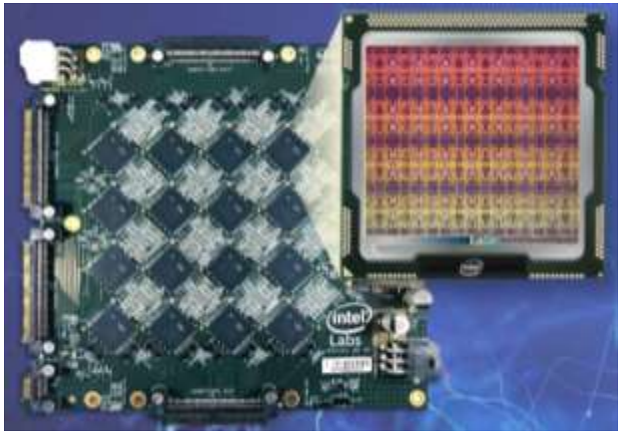
Pi-hosted FPAAs

Other Neuromorphic Platforms

Near-term work with the FPAA is looking at interfacing with other neuromorphic APIs like EONS (Evolutionary Optimized Neuromorphic Systems)

RG also hosts open-source simulation tools like Nengo and frameworks like Fugu, as they become available.

We are also starting a new project using Intel Loihi through INRC! (Cloud access, Fall 2019)

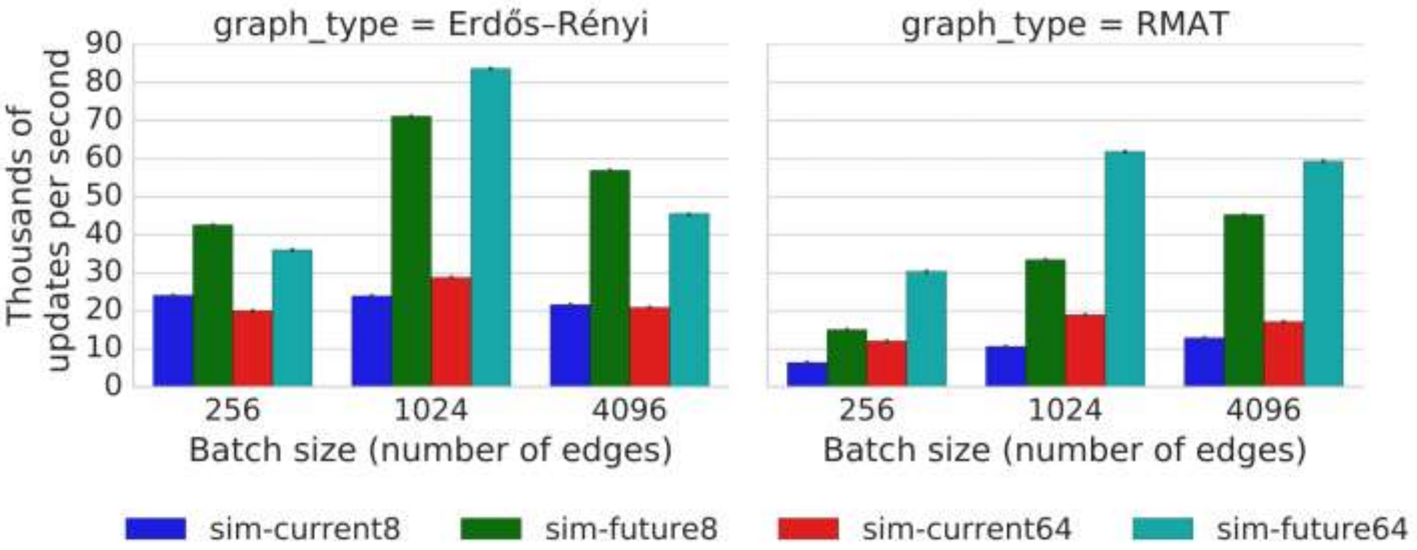


Intel Nahuku 32-chip Loihi platform, picture courtesy of Mike Davies, INRC, 2019

FPGAs as Rogue Enablers

Boards with HBM enable high bandwidth for mapping algorithms, machine learning, and neuromorphic computing (DANNA/Caspian from UTK)

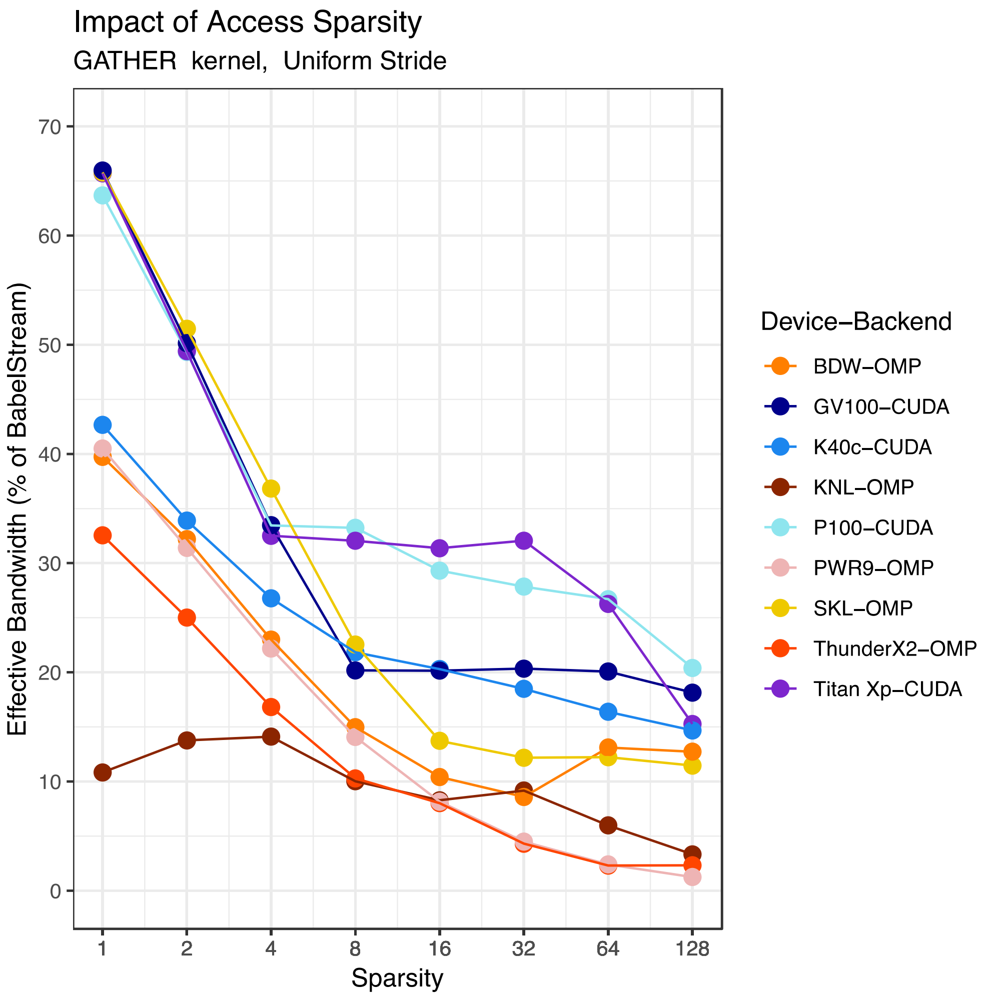
RG Benchmarking - Emu Microbench (STINGER)



Focused on Streaming BFS

Code available at https://github.com/ehein6/emu-microbench and increasingly as part of Emu toolchain releases

RG Benchmarking - Spatter

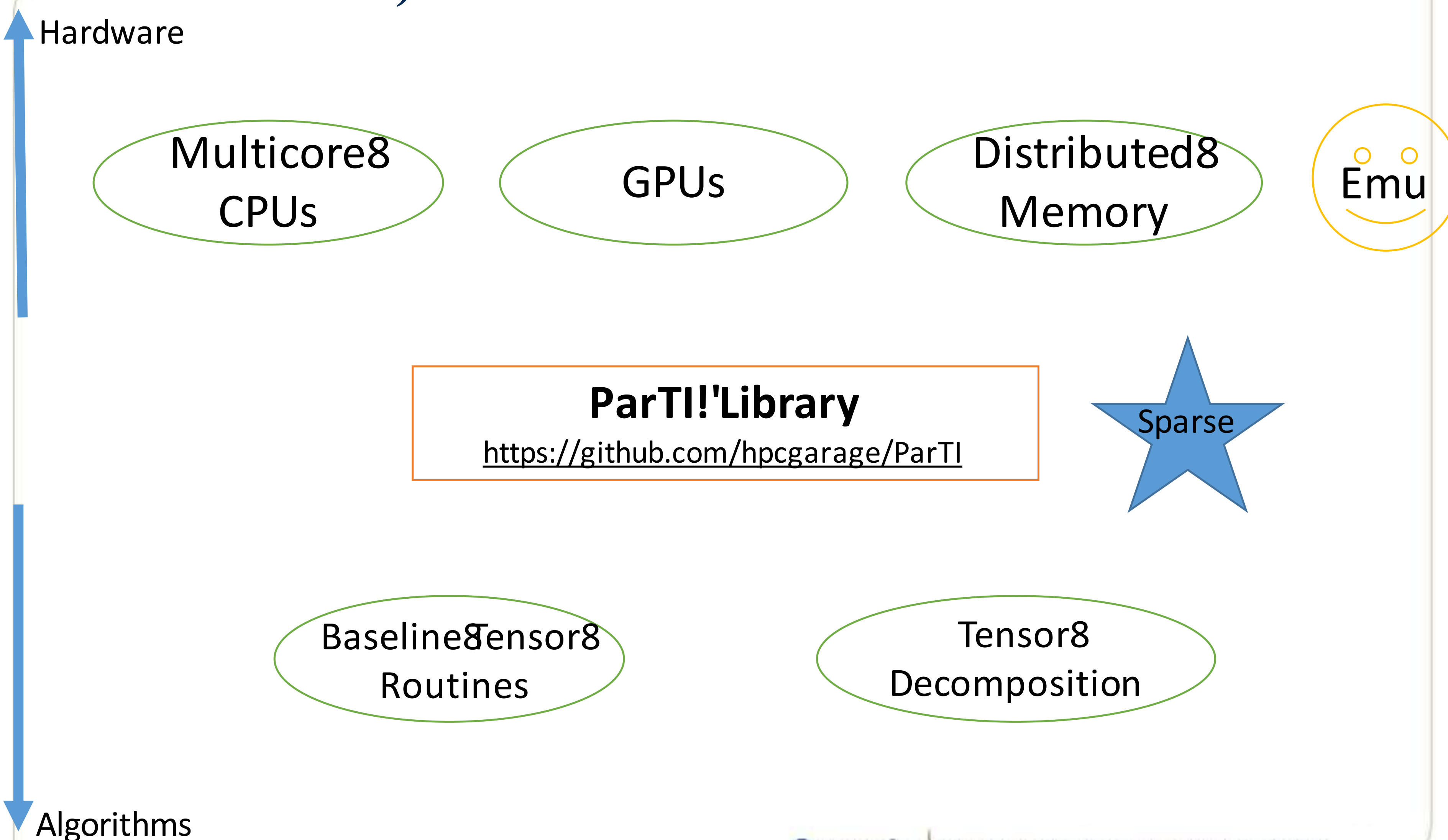


Code available at Spatter.io

Spatter provides a new benchmark to evaluate gather/scatter accesses across different CPU and GPU platforms.

Emu and FPGA backends are in progress!

RG Libraries – ParTI (Parallel Tensor Infrastructure)



RG Libraries - ParTI

Data Structures/ Platforms	Algorithms	Multicore CPUs	GPUs	Distributed Systems	Emu
COO	CP	✓	✓	✓	✓
	Tucker	✓	✓		
HiCOO	CP	✓	✓	✓	
	Tucker				

RG Tools - Kokkos

- Kokkos is a C++ library that is focused on ***performance portability*** through the mapping of parallel patterns

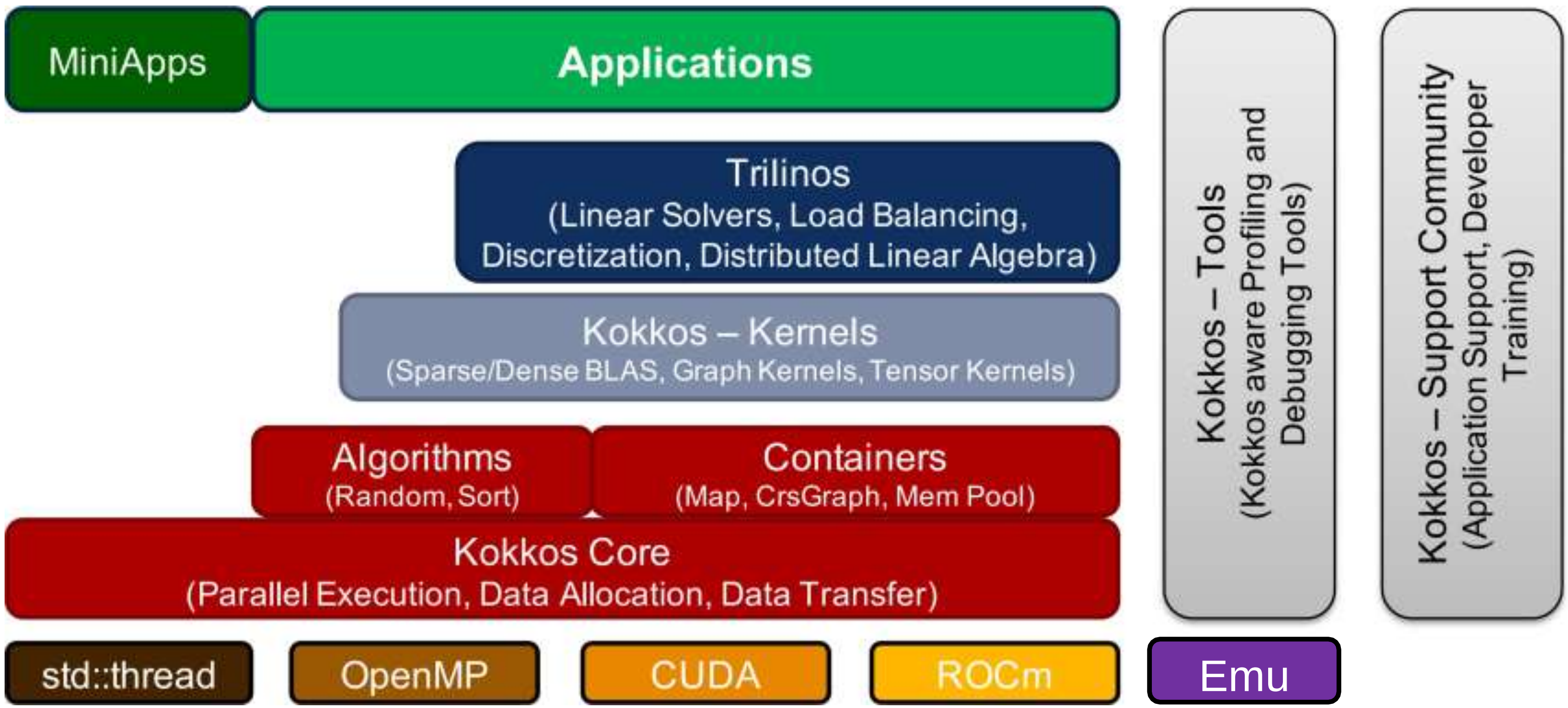
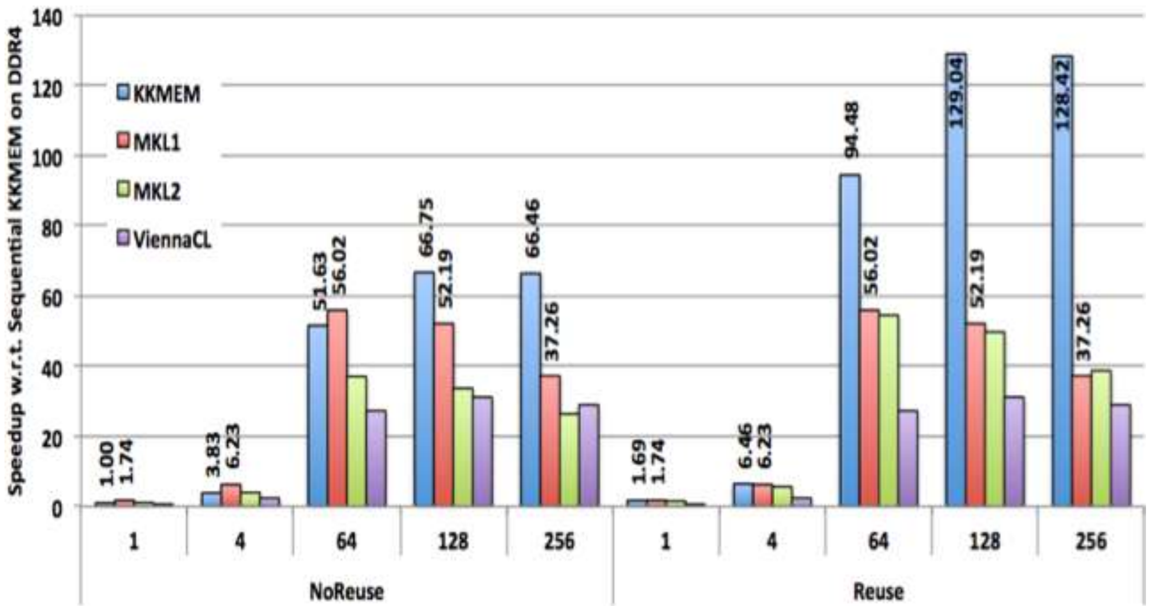


Diagram from Ellingwood, Trott, ANL 2018 Kokkos Tutorial

RG Tools - Kokkos

- Kokkos Kernels provide implementations of common sparse, dense linear algebra, and graph-related operations
  - BLAS, SPARSE, AXPY, graph coloring, tensor contraction
  - Common front-end implementations benefit from mapping to Kokkos

SPGEMM Results Comparing Kokkos Kernel implementation with MKL and ViennaCL on Knight's Landing



From Rajamanickam, et al. KokkosKernels: Compact Layouts for Batched Blas and Sparse Matrix-Matrix multiply, Batched BLAS Workshop, 2017

How to Work with the Rogues Gallery

Sign up for an account at crnch.gatech.edu – it's free!

Join our Slack or email lists and look out for CRNCH-sponsored events like tutorials and yearly summits in Atlanta, GA

Acknowledgments

- Emu – Janice McMahon and many others!
- Dr. Jennifer Hasler
- Srinivas Eswar (GT CSE)
- Dr. Eric Hein (GT ECE $\Rightarrow$ Emu)
- Patrick Lavin (GT CSE)
- Jiajia Li (GT CSE $\Rightarrow$ PNNL)
- Abdurrahman Yaşar (GT CSE)
- Dr. Ümit Çatalürek (GT CSE)
- Dr. Tom Conte (GT CS/ECE)
- Dr. Vivek Sarkar (GT CS/ECE)
- Dr. Bora Uçar (ENS Lyon CNRS)
- Dr. Rich Vuduc (GT CSE)

Code:

- crnch-rg.gitlab.io – front-end for public resources
- https://github.com/ehein6/emu-microbench
- Spatter.io – Spatter benchmark
- ParTI - https://github.com/hpcgarage/ParTI

Tutorial Feedback

- Feel free to email us:
  - Jason - jason.riedy@cc.gatech.edu
  - Jeff - jyoung9@gatech.edu
- We also have a survey for you to fill out on the event webpage at

crnch-rg.gitlab.io/pearc-2019

From the Art Institute's gardens – Check them out!