

Computer Engineering Lab

Philip Leong (philip.leong@sydney.edu.au)



THE UNIVERSITY OF
SYDNEY



Computer
Engineering
Lab

CRICOS 00026A TEQSA PRV12057



Acknowledgment of Country

I acknowledge the Gadigal people of the Eora Nation as the traditional custodians of the land on which the university stands and pay respects to Elders past, present and emerging.

Computer Engineering Lab



Computer
Engineering
Lab

- CEL focuses on how to utilise parallelism to solve computing problems
 - Novel architectures, applications and design techniques

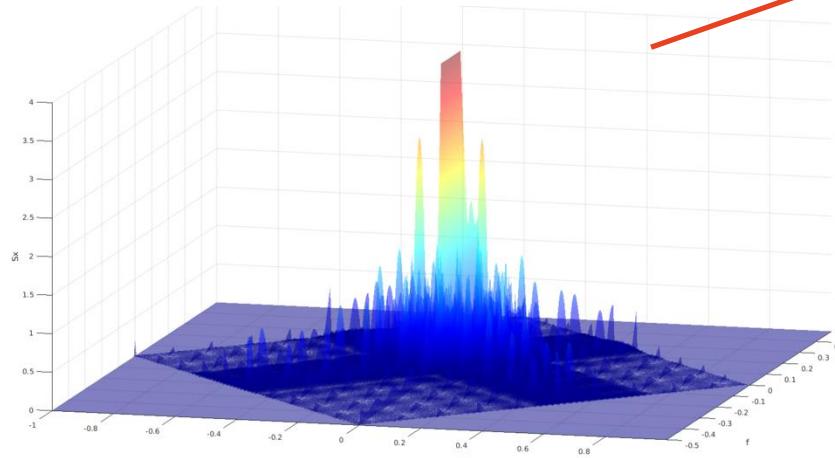
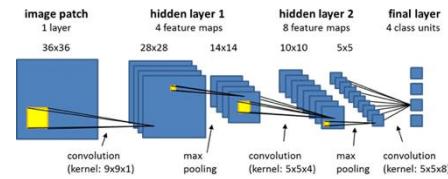
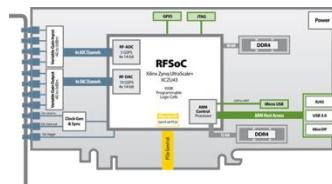


Radio Frequency Machine Learning

- Processing RF signals remains a challenge
- ML offers an opportunity to achieve RF scene understanding
- Overall research theme is to achieve high performance in RFML using FPGAs through (EPIC)
 - Exploration, Parallelism, Integration, Customisation

What are we trying to do?

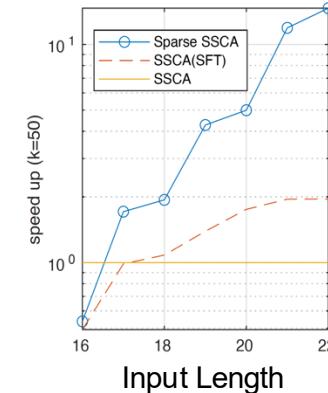
Combine real-time FPGA HW, cyclostationary analysis, ML and AR



Cyclostationary Signal Processing (CSP)

ML on FPGA

Apple Vision Pro



10x Acceleration
of CSP in SW



RF Fingerprinting
on FPGA

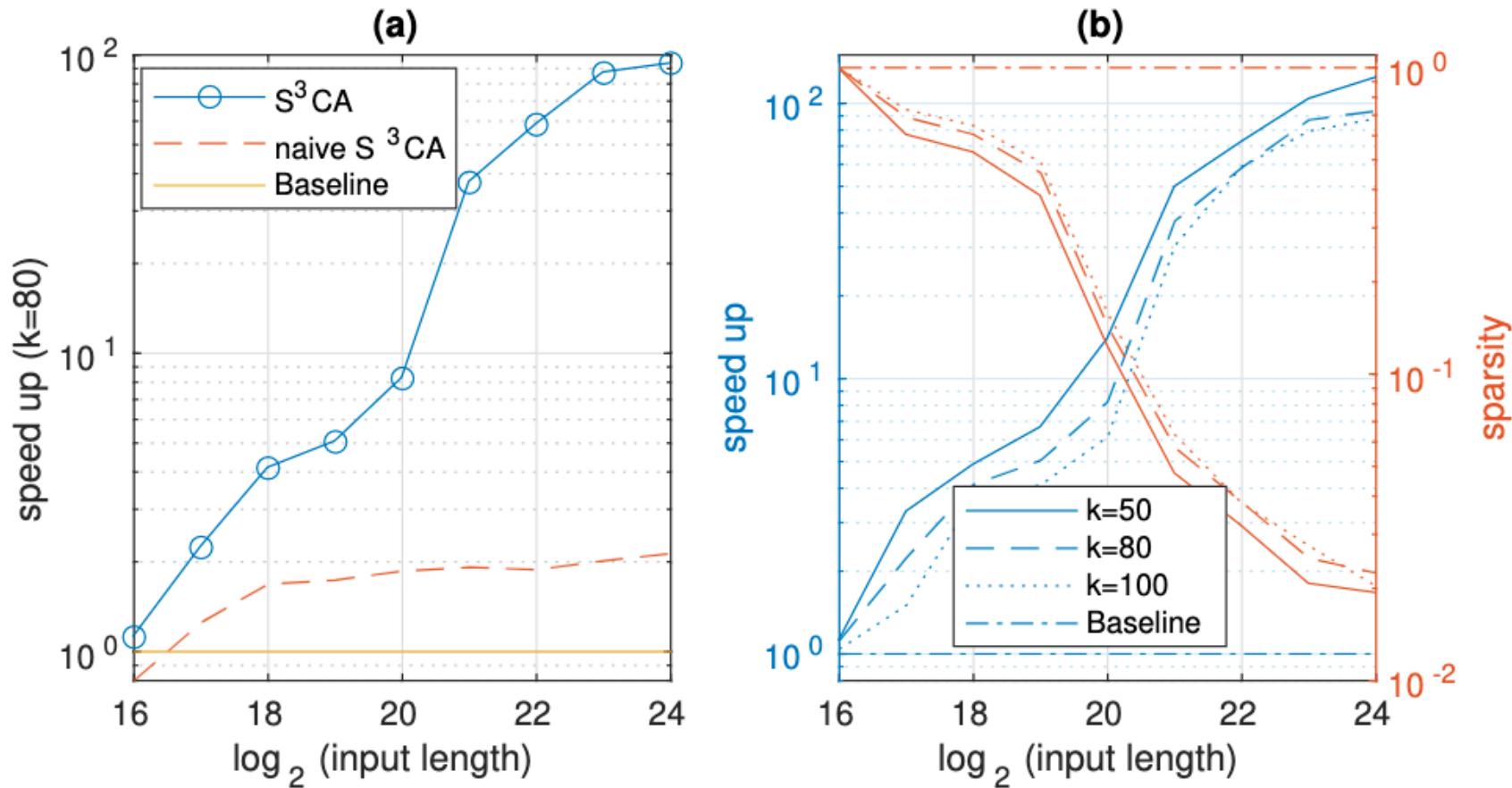
Addressing Research Gaps in Radio Frequency Machine Learning

- Requirements: High throughput, low latency, good accuracy, integration
- Some research gaps (ALERT)
 - Algorithms
 - Latency
 - Edge
 - Representations (features)
 - Training (to adapt to changing conditions)

Some Ways to Bridge Gaps

- **Algorithms**: lower complexity techniques with minimal accuracy reduction
- **Latency**: highly pipelined architectures, quantisation, hardware-friendly DNN techniques
- **Edge**: single-FPGA solutions that reduce off-chip communications and utilise massive on-chip bandwidth, integration, memory footprint reduction
- **Representations**: techniques that match properties of signals, e.g. complex numbers, man-made modulation schemes
- **Training**: systems that can adapt to changing conditions

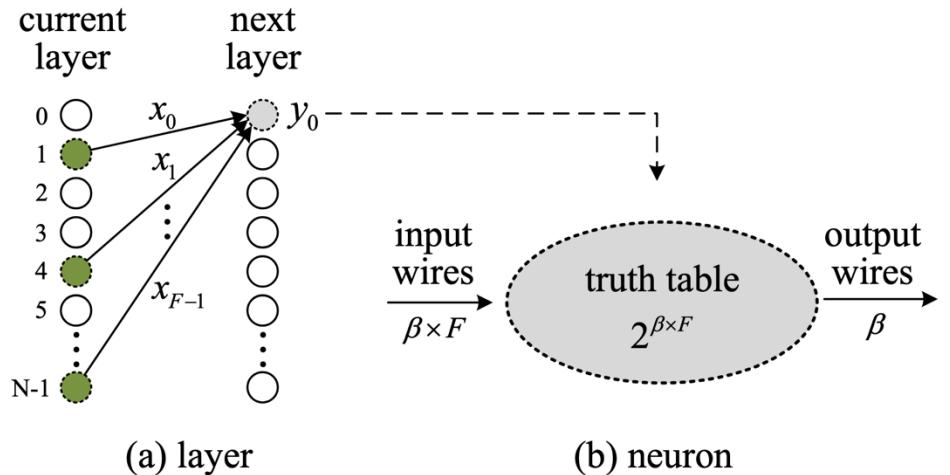
ALERT Algorithms: Sparse FFT



https://phwl.org/assets/papers/s3ca_spl24.pdf

ALERT Latency: PolyLUT-Add

Each neuron implemented in a single LUT

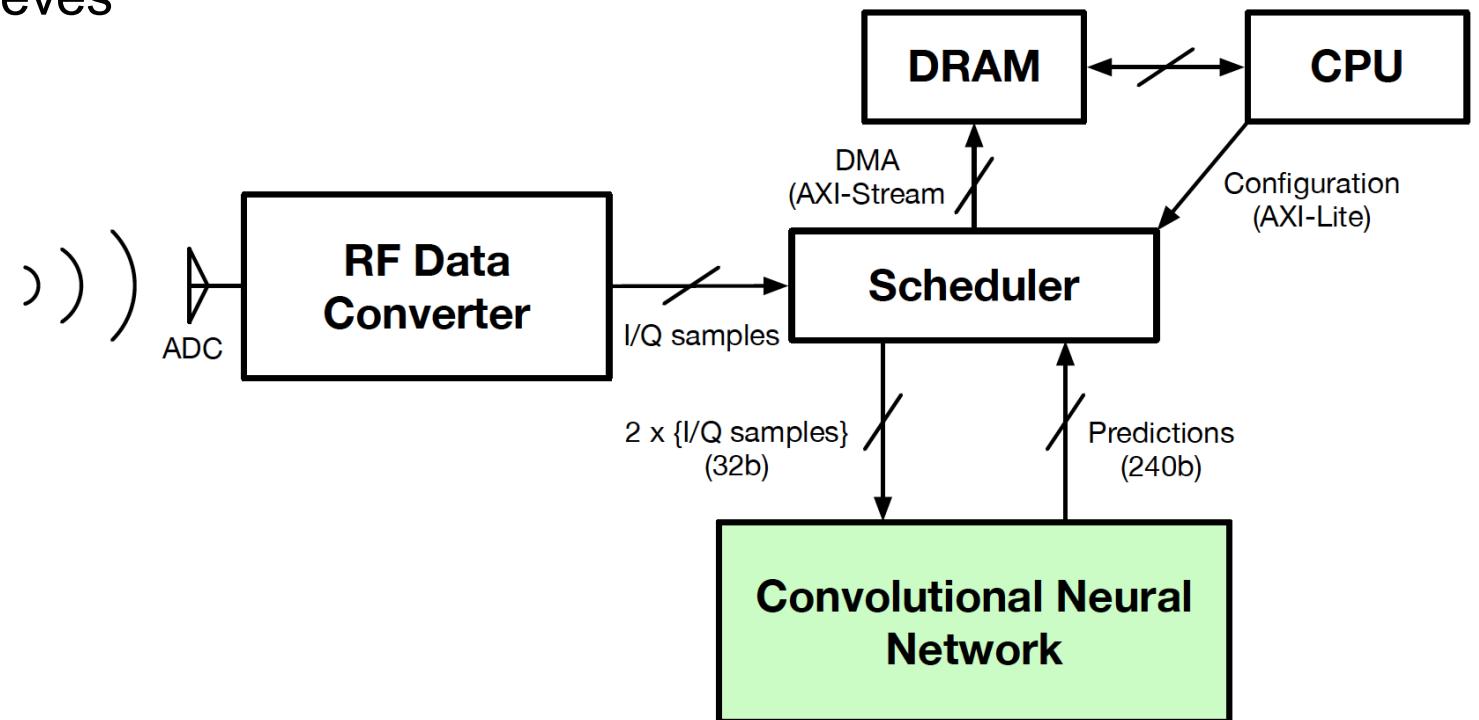


Dataset	Model	Accuracy↑	LUT	FF	DSP	BRAM	$F_max(\text{MHz})\uparrow$	Latency(ns)↓
MNIST	PolyLUT-Add (HDR-Add2, $D=3$)	96%	15272	2880	0	0	833	7
	PolyLUT (HDR, $D=4$) [8]	96%	70673	4681	0	0	378	16
	FINN [20]	96%	91131	-	0	5	200	310
	hls4ml [21]	95%	260092	165513	0	0	200	190
Jet Substructure	PolyLUT-Add (JSC-XL-Add2, $D=3$)	75%	47639	1712	0	0	400	13
	PolyLUT (JSC-XL, $D=4$) [8]	75%	236541	2775	0	0	235	21
	Duarte <i>et al.</i> [2]	75%		88797*	954	0	200	75
	Fahim <i>et al.</i> [17]	76%	63251	4394	38	0	200	45

<https://github.com/bingleilou/PolyLUT-Add>

ALERT Edge: Modulation Classification @ 500 Msps

- Combine quantisation, sparsity and CSE to reduce computation 86%
- Fully pipelined design achieves
 - 488K classifications/s
 - 8 uS latency

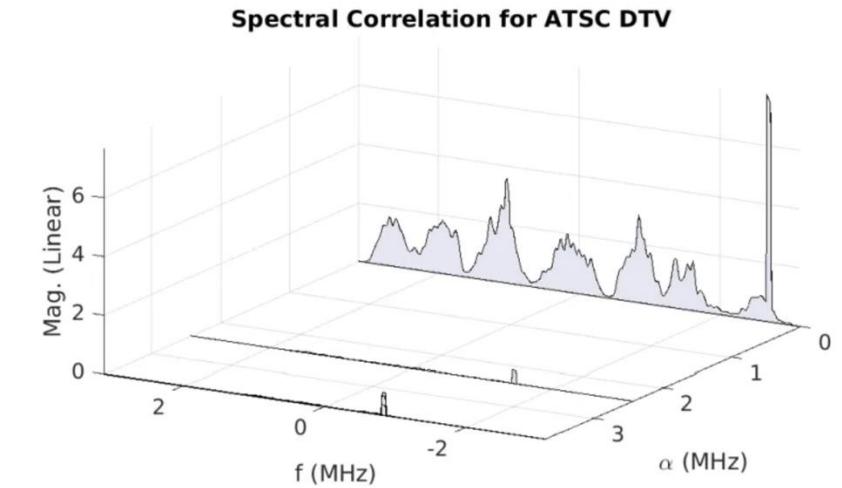
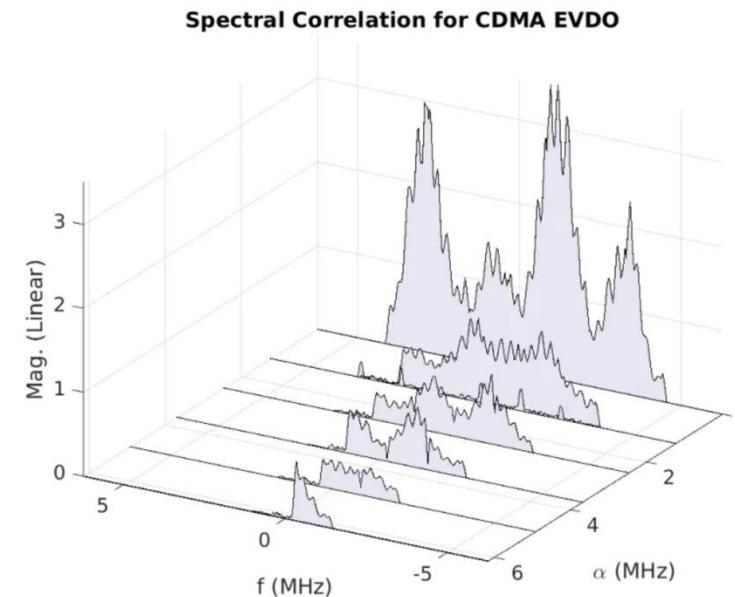
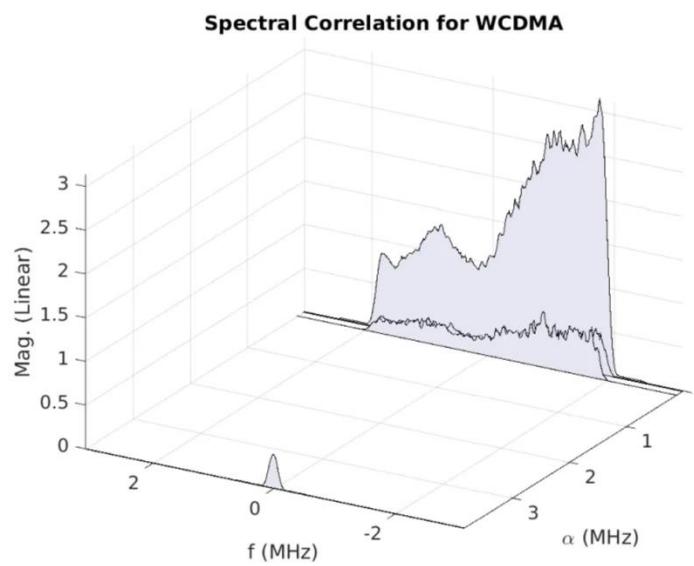


https://phwl.org/assets/papers/amc_raw20.pdf

ALERT Representations: Cyclostationary Features

Cyclostationary: probability distribution changes periodically over time

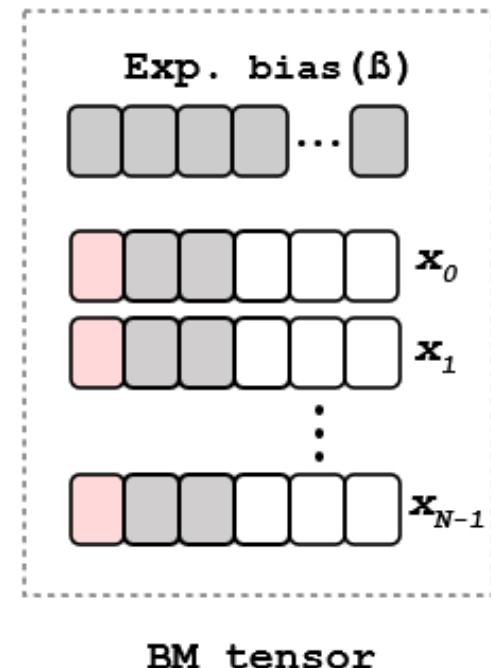
- Excellent at low SNR (but computationally expensive)



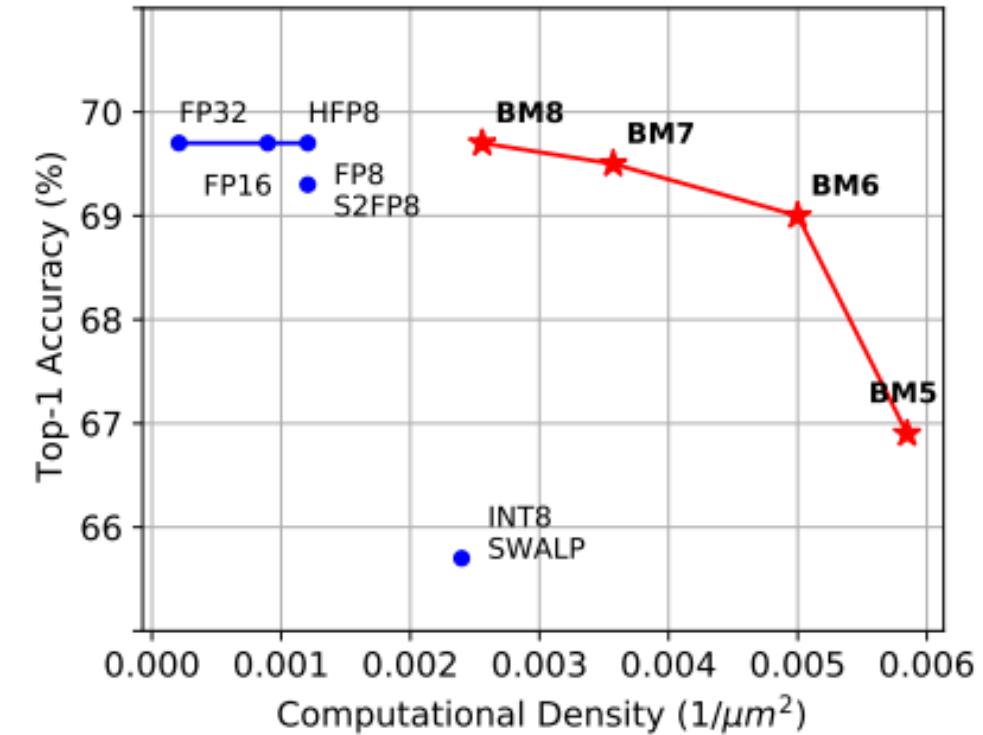
https://phwl.org/assets/papers/famssca_fpl25.pdf

ALERT Training: Training of DNNs at the Edge

- Block minifloats which can train with 8 and 6-bit precision



Dataset: ImageNet
Model: ResNet-18

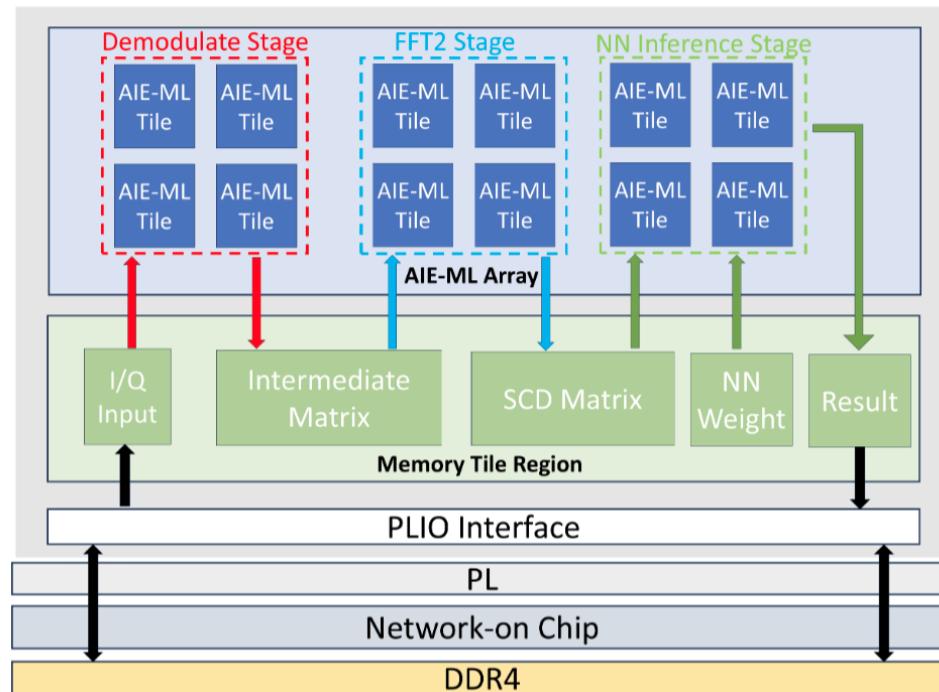


<https://openreview.net/forum?id=6zaTwpNSsQ2>

First Real-time Cyclostationary + AI RFML System



Real-time FPGA HW, cyclostationary analysis and AI

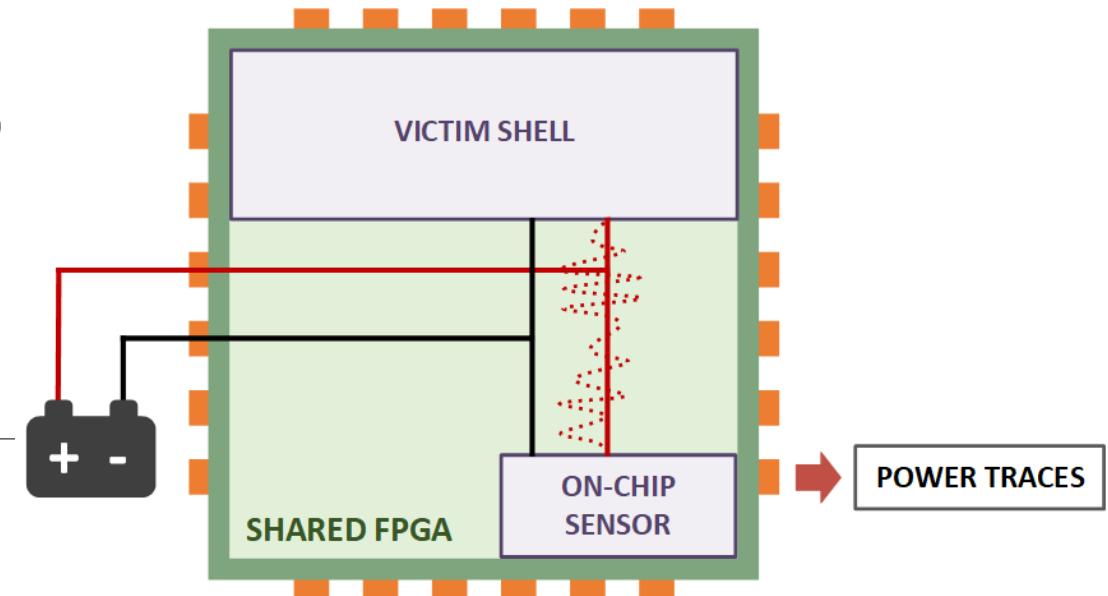
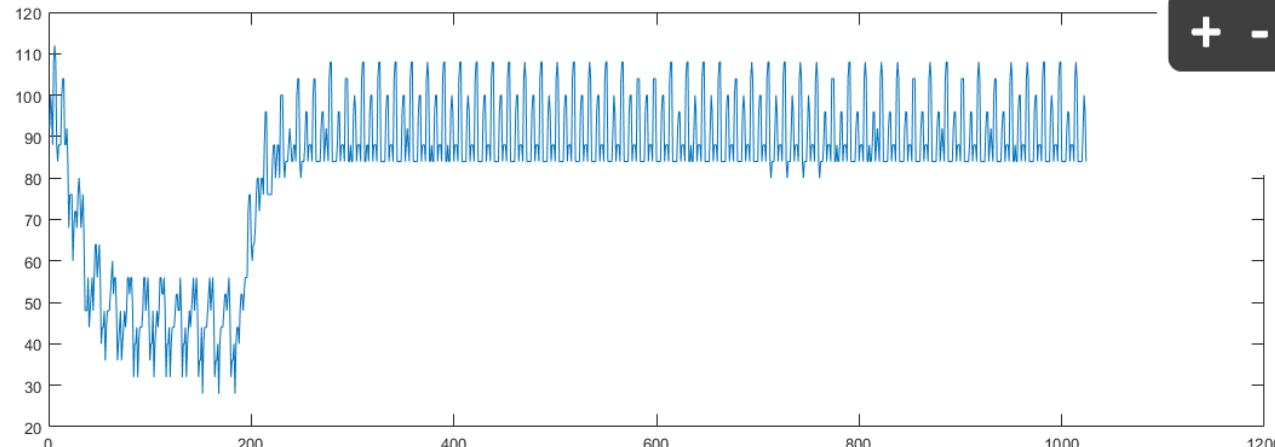


Platform	Speedup	Tput (GFLOPs)	Power (W)	Energy Eff (GFLOPs/W)
CPU I9-9900KF	1	9.03	14	0.65
GPU RTX3090	2	18.8	95	0.20
FPGA VEK280	7	66.3	10	6.34

https://phwl.org/assets/papers/cycloamc_fpt25.pdf

RPA attacks on FPGAs (Prof Sri Parameswaran)

- Using on chip sensors, we gather data from the victim shell (such as AES keys)
- Attacks on AWS FPGA servers
- Research into Countermeasures
- Published in CHES '22, CHES '24



RISCV Security and Enhancements (Prof Sri Parameswaran)

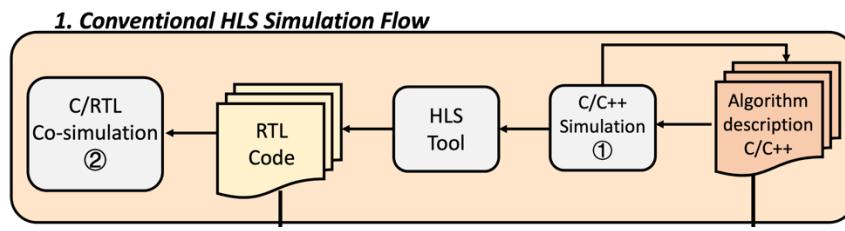
- Memory Security Extensions (DAC '21, DAC '22)
 - Implemented on FPGA
- Countermeasures for Clock Jitter-based Attacks (ICCAD '25)
 - FPGA implementation
- Efficient AI implementation
 - Implemented on FPGAs using Rocketchip with approximate multipliers for GEMMINI

Hercules Verification of Digital Designs (Dr David Boland)

If bugs are detected, we help a user find and fix them:

Snapshot controller detects potential errors and take snapshots of all hardware information

Backend recreates errors in a software simulation environment so a user can find and fix them



Instead of software simulation of hardware

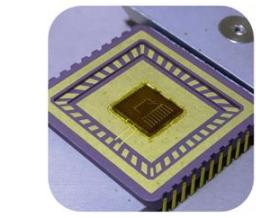
We simultaneously run on the same FPGA:

— Hardware & Golden reference software implementation

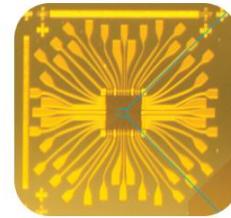
We can now detect bugs over a much longer timescale

Able to find bugs orders of magnitude faster than the state of the art

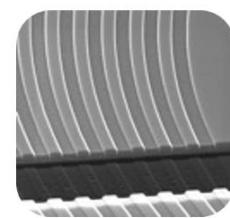
Rapid Prototype Foundry (A/Prof Steve Shu)



Semiconductor & Advanced packaging



Quantum



Photonics



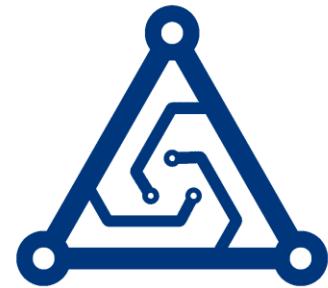
Biomedical devices

- RPF is a shared **core research facility** for micro-/nano-fabrication & metrology
 - Supported by USyd & Australian National Fabrication Facility (ANFF)
 - **700sqm+** ISO 5/6/7 cleanrooms
 - **20+** engineers/operations
 - **70+** tools, incl. litho, deposition, etching, wet processing, packaging, metrology, Fibre Braag Grating, etc.

Spinoff Companies



Real-time AI Applications
on FPGAs



TernaryNet

Fabless Semiconductor
Company

Thank you!

Philip Leong (philip.leong@sydney.edu.au)

David Boland (david.boland@sydney.edu.au)

Steve Shu (steve.shu@sydney.edu.au)

Sri Parameswaran (sri.parameswaran@sydney.edu.au)

Stephen Tridgell, Xueyuan Liu (Modulation classification)

Jingyi Li, Ruilin Wu (Cyclostationary)

Binglei Lou (LUT-based DNNs)

Wenjie Zhou (Edge training)

<http://phwl.org/talks>

