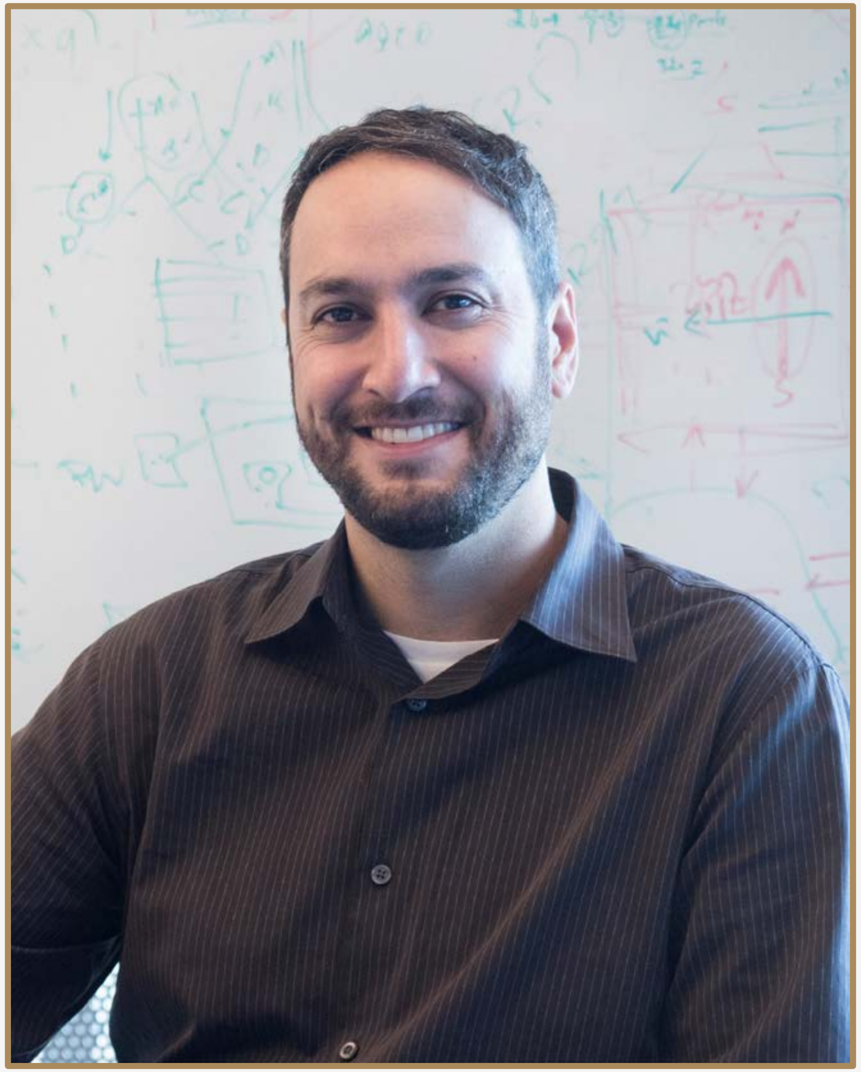


BRUCEK KHAILANY

DIRECTOR OF RESEARCH, ASIC & VLSI
NVIDIA CORPORATION

DISTRIBUTION STATEMENT A. Approved for public release

HIGH-PRODUCTIVITY IC DESIGN FOR MACHINE LEARNING ACCELERATORS

This research was, in part, developed with funding from the Defense Advanced Research Projects Agency (DARPA).
The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

DESIGN COMPLEXITY

More transistors: improved capability AND design costs

[Electronics 1965]

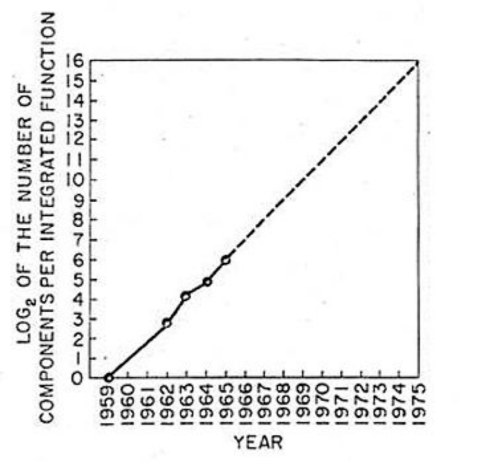
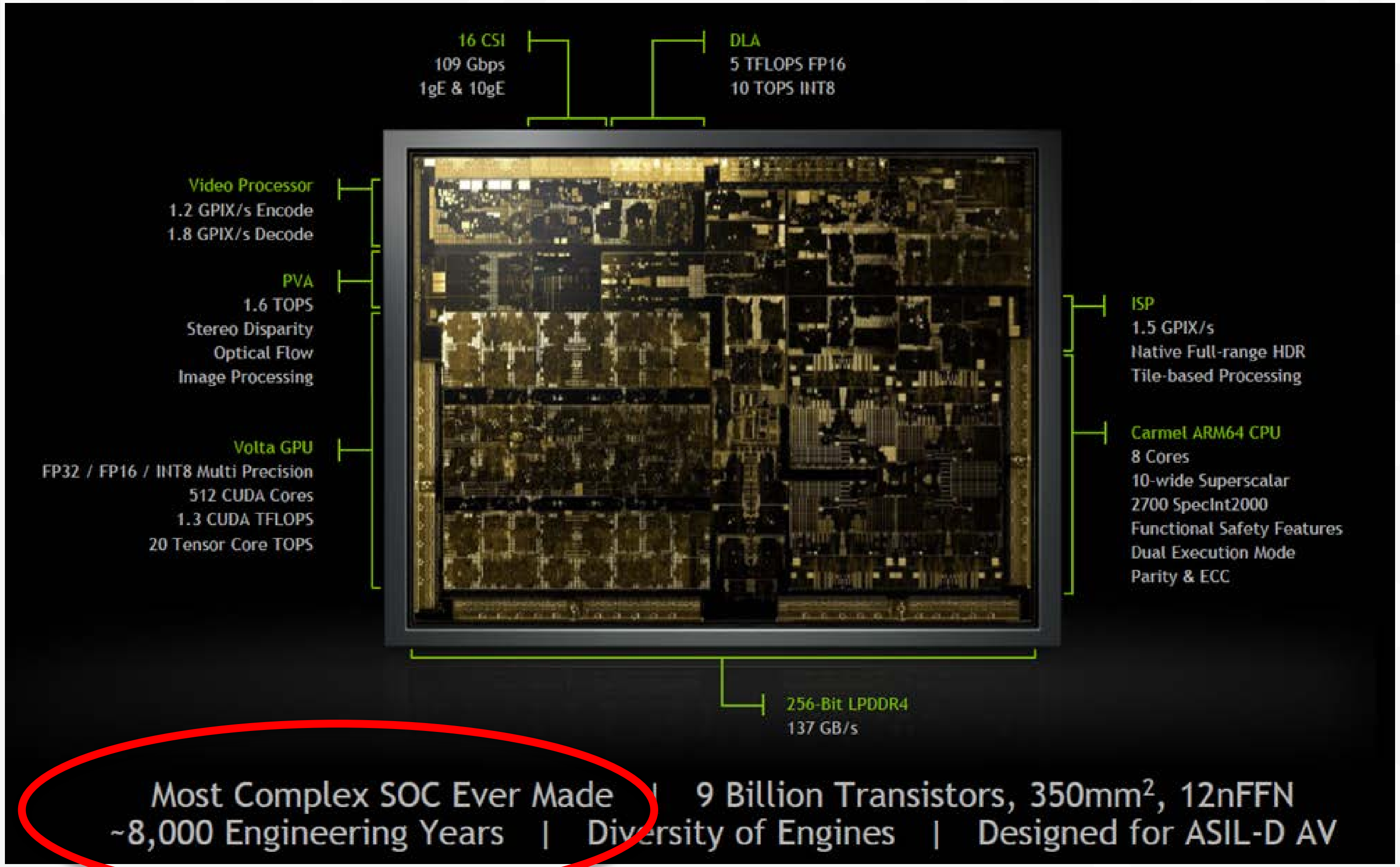


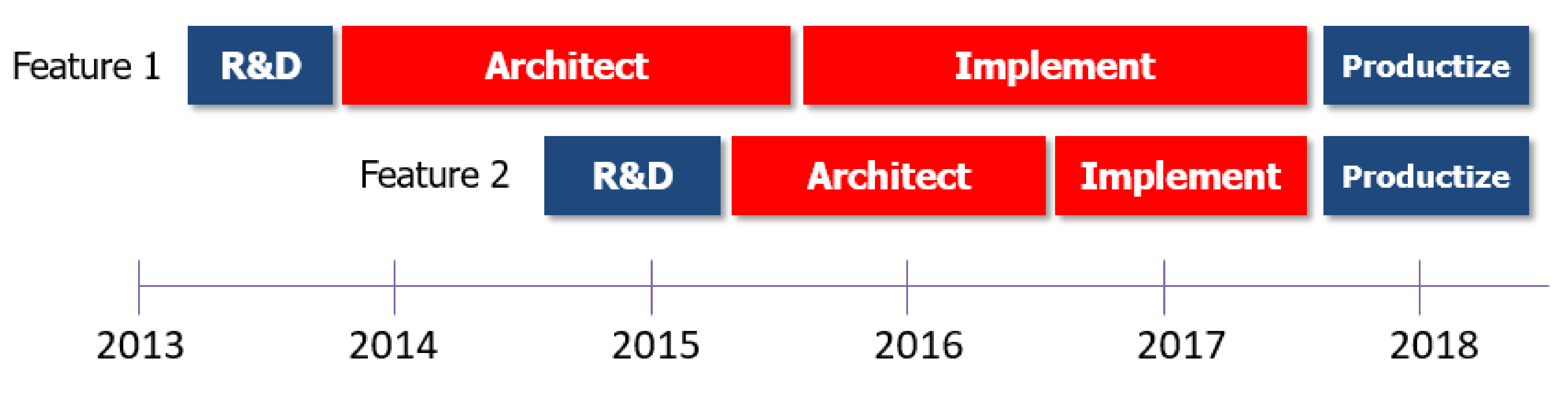
Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

NVIDIA Xavier SoC [CES 2018]



MOTIVATION – LOWER DESIGN EFFORT

- Typical development timeline: 3-5 years from R&D to product:

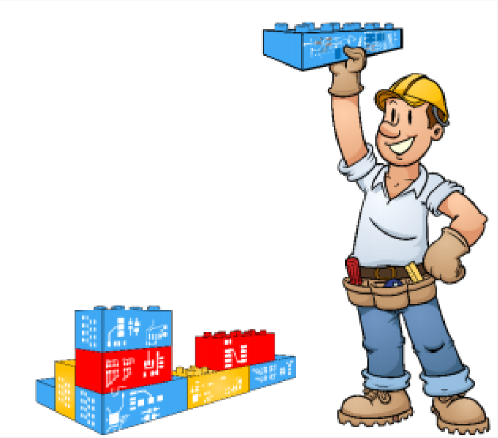


- Design and especially *verification* dominates implementation effort
  - Majority of all digital IC design effort at NVIDIA
  - Prohibits which features make it into each SoC
- How commercial companies would benefit from 10x lower design effort
  - Overlap architect & implement phases for faster time-to-market
  - Get more features into each SoC

RESEARCH APPROACH UNDER CRAFT

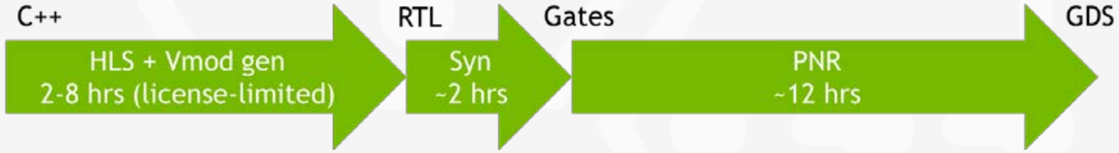
RAISE HARDWARE DESIGN LEVEL OF ABSTRACTION

- Use higher-level languages
  - e.g. C++ instead of Verilog
- Use tools
  - e.g. High-Level Synthesis (HLS)
- Use libraries / generators
  - MatchLib



AGILE VLSI DESIGN

- Small teams, jointly working on architecture, implementation, VLSI
- Continuous integration with automated tool flows
- Agile project management techniques
- 24-hour spins from C++-to-layout



OBJECT-ORIENTED HLS-BASED DESIGN

- **MatchLib** – **M**odular **A**pproach **T**o **C**ircuits and **H**ardware **Lib**rary
  - HLS-compatible
  - Highly-parameterizable, high QoR implementations
  - Open-sourced at https://github.com/NVlabs/matchlib

What is MatchLib?

MatchLib is a library of reusable modules & functions for common HW structures, somewhat analogous to STL in programming world or DesignWare in ASIC world. Key motivations are:

- Encapsulate verified functionality
- Encapsulate QoR-optimized implementation
- Heavy use of templates and other C++ features for parameterization

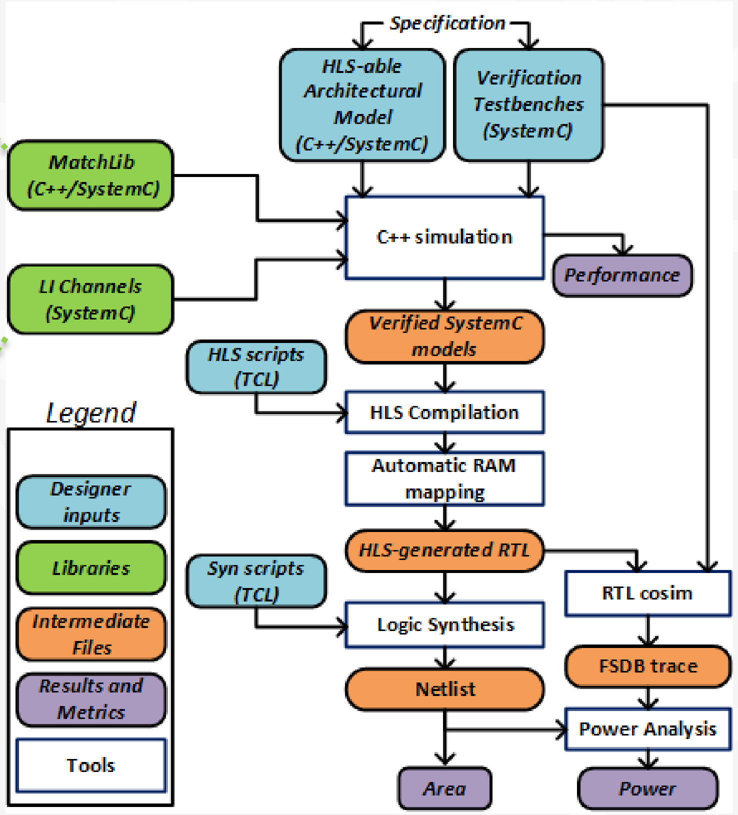
Components for common HW structures can be in one of the three forms:

- C++ functions : datapath description
- C++ classes : state updating methods
- SystemC modules : self-contained modules

In addition, there is a collection of auxiliary non-synthesizable components useful for building testbench and debug infrastructure of developed HW.

- Goal: "STL/Boost" for HW
- Working closely with Mentor on HLS support, technology transfer

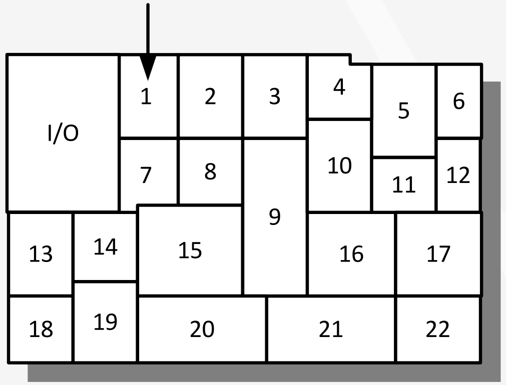
- "Push-button" C++-to-gates flow
- Target: 10x productivity of manual RTL coding



["A Modular Digital VLSI Flow for High-Productivity SoC Design", Khailany et al., DAC 2018]

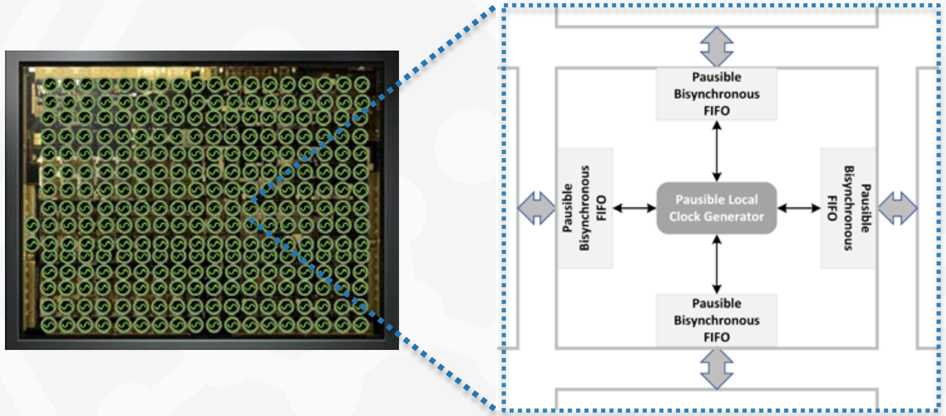
AGILE VLSI DESIGN TECHNIQUES

SMALL, ABUTTING PARTITIONS FOR FAST PLACE-AND-ROUTE ITERATIONS



- Daily iterations through P&R tools
  - Always working on “top of tree” RTL
- Agile, incremental approach to design closure during march-to-tapeout phase
  - Tweak floorplans, timing constraints
  - RTL design bugs, performance, VLSI constraints, and tool settings all converge together

GLOBALLY ASYNCHRONOUS LOCALLY SYNCHRONOUS PAUSABLE ADAPTIVE CLOCKS

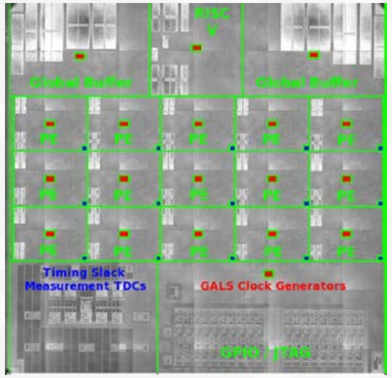


- 100s of local adaptive clock generators
  - Fast, error-free clock domain crossings
  - “Correct by construction” top-level timing closure
  - Reduced margin for power-supply noise

[“A Fine-Grained GALS SoC with Pausible Adaptive Clocking in 16 nm FinFET”, Fojtik et al., ASYNC 2019 (Best Paper)]

3 SOC TESTCHIP DEMONSTRATIONS

Programmable ML Inference Accelerator in TSMC 16nm FinFET (Methodology Demonstration)



RC17 Testchip

2016

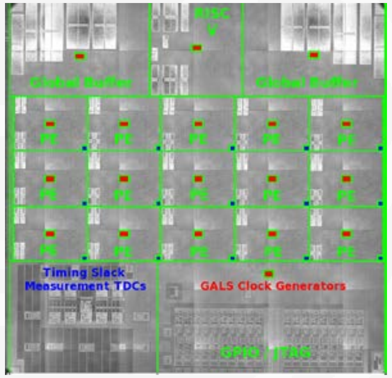
2017

2018

Chip	Purpose	Team	RTL + Verif Effort
RC17	Develop Methodology	8-10 people for arch, RTL, verif, VLSI	20 eng-months

3 SOC TESTCHIP DEMONSTRATIONS

Programmable ML Inference Accelerator in TSMC 16nm FinFET (Methodology Demonstration)



RC17 Testchip

2016

2017

2018

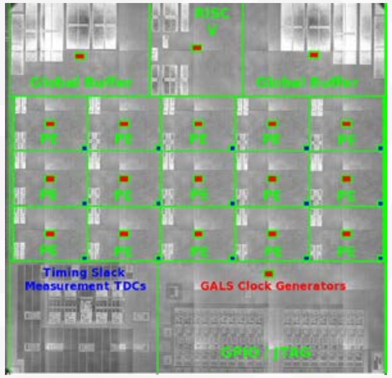
Chip	Purpose	Team	RTL + Verif Effort
RC17	Develop Methodology	8-10 people for arch, RTL, verif, VLSI	20 eng-months
RC17b	Demo portability	2 people + Cadence Design Services	5 eng-months

Port to GF 14nm

RC17b Testchip

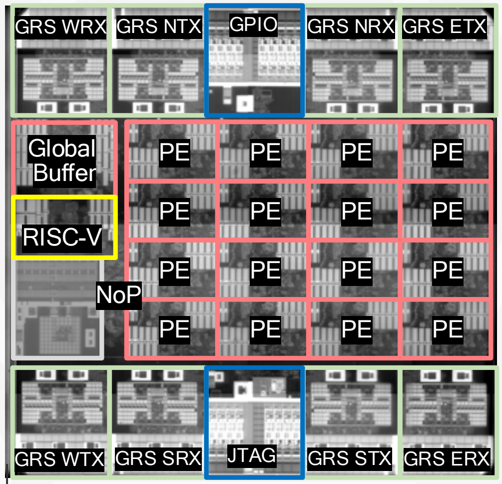
3 SOC TESTCHIP DEMONSTRATIONS

Programmable ML Inference Accelerator in TSMC 16nm FinFET (Methodology Demonstration)



RC17 Testchip

Optimized MCM DL Inference Accelerator in TSMC 16nm FinFET



2016 2017 2018

RC18 Testchip

Port to GF 14nm

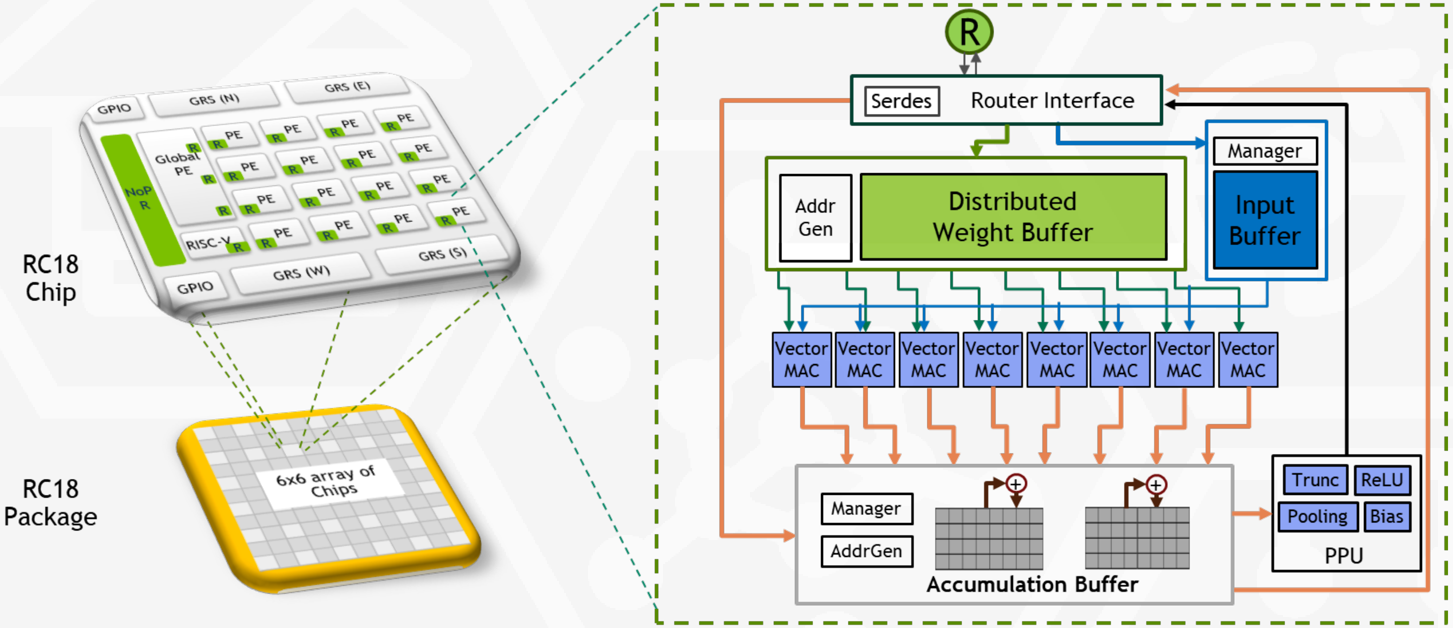
RC17b Testchip

Chip	Purpose	Team	RTL + Verif Effort
RC17	Develop Methodology	8-10 people for arch, RTL, verif, VLSI	20 eng-months
RC17b	Demo portability	2 people + Cadence Design Services	5 eng-months
RC18	Demo perf & efficiency	5-10 people for arch, RTL, verif, VLSI	20 eng-months

~10x RTL design and verification effort reduction compared to manual RTL

RC18: SCALABLE DEEP NEURAL NETWORK INFERENCE ACCELERATOR ARCHITECTURE

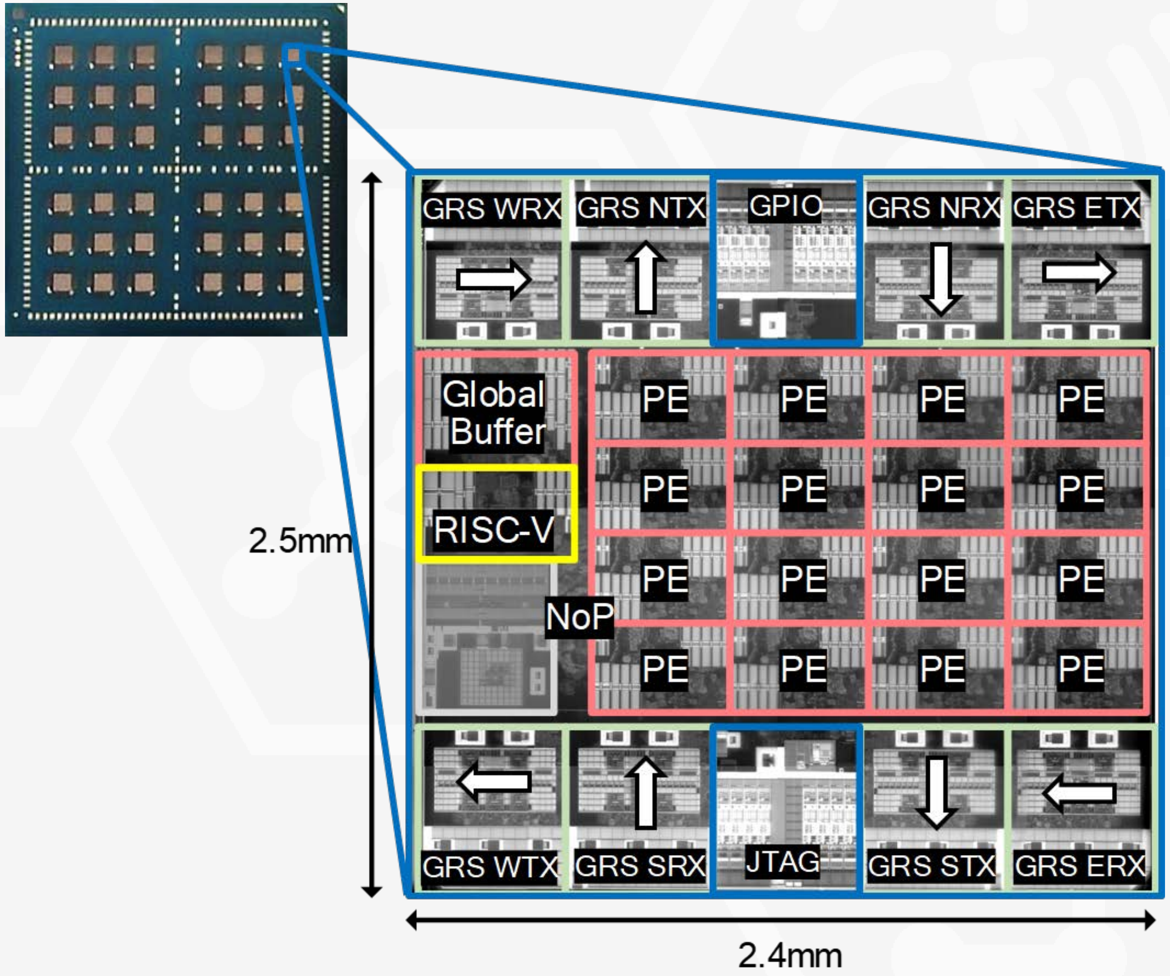
- Scalable DL inference accelerator
- 36-die MCM
- Hierarchical Network
- 128 TOPS (8b int)
- 9 TOPS/W



- Used CRAFT flow for design space exploration & VLSI implementation
  - 5-10 researchers, 6 months spec-to-tapeout

FABRICATED MCM-BASED ACCELERATOR

- TSMC16nm testchip
  - 16 Processing Elements (PE)
  - 100 Gbps between chips in mesh
- Wide performance range
  - 1-die: 4 TOPS (4W)
  - 36-die: 128 TOPS (106W)
- Energy-efficient compute
  - 8-bit integer datapath
  - 0.11-1.0 pJ/op (0.41V-1.2V)
- Energy-efficient chip-to-chip communication (GRS)
  - 11-25 Gbps/pin, 0.82-1.75 pJ/bit



[“A 0.11 pJ/Op, 0.32-128 TOPS, Scalable, Multi-Chip-Module-based Deep Neural Network Accelerator with Ground-Reference Signaling in 16nm”, Zimmer et al., 2019 VLSI Circuits Symposium]

SUMMARY AND FUTURE WORK

- Design methodology R&D can solve the complexity problem
- Summary
  - Raise the level of design abstraction (languages, tools, libraries)
  - Rethink design best practices to optimize for agile VLSI development
- Future work
  - Improve the maturity and adoption of libraries within the HLS community
  - GPU-accelerated and ML-assisted EDA algorithms research

ACKNOWLEDGEMENTS

- Research sponsored by DARPA under the CRAFT program (PM: Linton Salmon).
- NVIDIA Collaborators:
  - Evgeni Krimer, Rangharajan Venkatesan, Jason Clemons, Ben Keller, Matthew Fojtik, Alicia Klinefelter, Angshuman Parashar, Michael Pellauer, Nathaniel Pinckney, Mark Ren, Yakun Sophia Shao, Stephen Tell, Yanqing Zhang, Brian Zimmer, Bill Dally, Joel S. Emer, Stephen Keckler
- Harvard Collaborators:
  - David Brooks, Gu-Yeon Wei, and team
- Former NVIDIA interns/postdocs:
  - Christopher Fletcher, Davide Giri, Ziyun Li, Antonio Puglielli, Shreesha Srinath, Gopal Srinivasan, Chris Torng, Sam (Likun) Xi
- Thanks to the Mentor Graphics Catapult HLS team for discussions and support:
  - Bryan Bowyer, Stuart Clubb, Moises Garcia, Khalid Islam, and Stuart Swan.

This research was, in part, funded by the U.S. Government, under the DARPA CRAFT program. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. Distribution Statement "A" (Approved for Public Release, Distribution Unlimited).

ERI ELECTRONICS RESURGENCE INITIATIVE

SUMMIT

2019 | Detroit, MI | **July 15 - 17**