The 2nd R-CCS Symposium,
Future Co-design Session 13:30 to 15:30
Day 2, Feb 18, 2020, Kobe, Japan

Synchoricity as the basis for going Beyond Moore

Not a Typo

Ahmed Hemani

Professor, Dept. Of Electronics, School of EECS, KTH, Stockholm Sweden

Email: hemani@kth.se

Going Beyond Moore !

Solutions to go beyond Moore

1. **Squeeze more out of CMOS**
   a. ASICs like custom functional hardware
   b. Delivers 2-4 orders better energy-delay product compared to GPUs, FPGAs and Multi-cores

2. **Complement CMOS with emerging technologies**
   a. 2.5D and 3D Integration (DRAM)
   b. Computation in memory using Memristors
   c. Plasmonics

Make it easy to use the solution

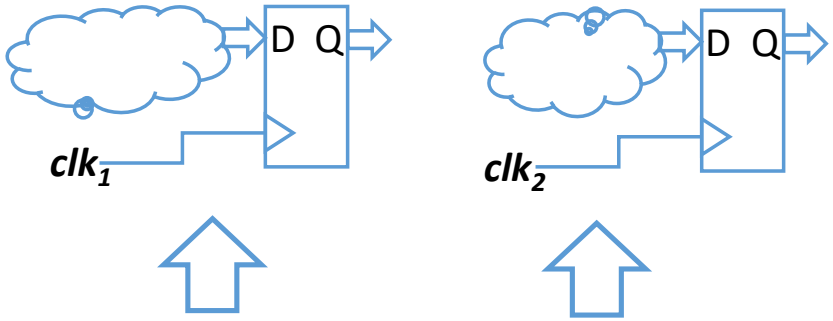
"Science makes progress, not when you find a solution, but when you make it easy to use the solution"

-- Venki Ramakrishnan, Nobel Laureate

Synchronicity

Synchoricity

Time is discretized using clock ticks



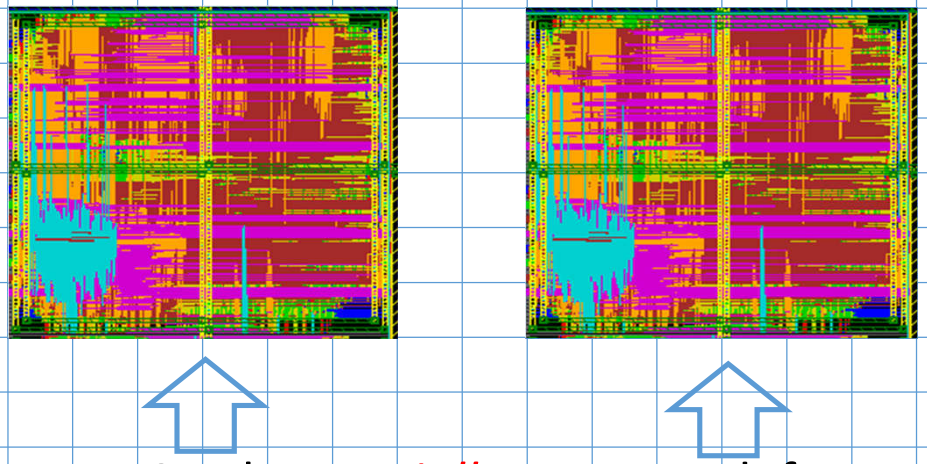
Can be *temporally* composed If

$clk_1 = clk_2$

&

The two clocks are skew aligned

Space is discretized using a virtual grid



Can be *spatially* composed If

If the number of grid cells

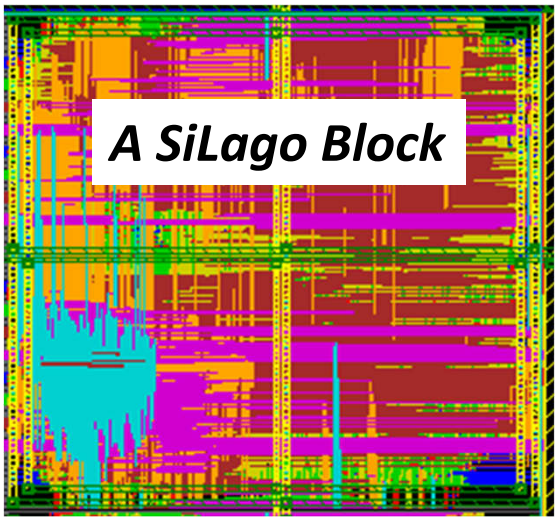
in each dimension are equal

&

Their interconnect edges are abuttable

SiLago (Silicon Lego) Blocks

SiLago Blocks are the new standard cells

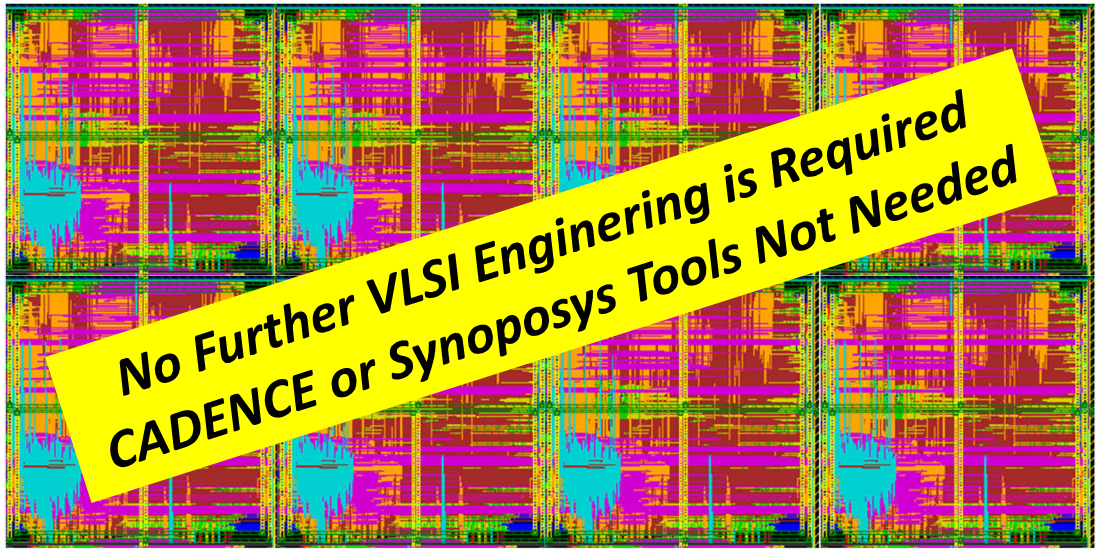


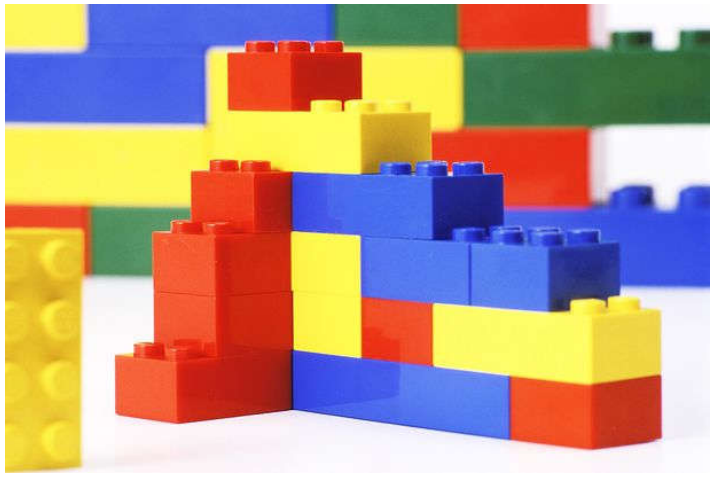
RTL & Coarse Grain Reconfigurable
4-5 orders larger than Standard Cells

Characterized with postlayout data
Empowers Synthesis from Higher Abstractions

Inter SiLago Block Wires bought to periphery at right place and right metal layer to enable compositoin by abutment

VLSI Designs are Composed by Abutting SiLago Blocks





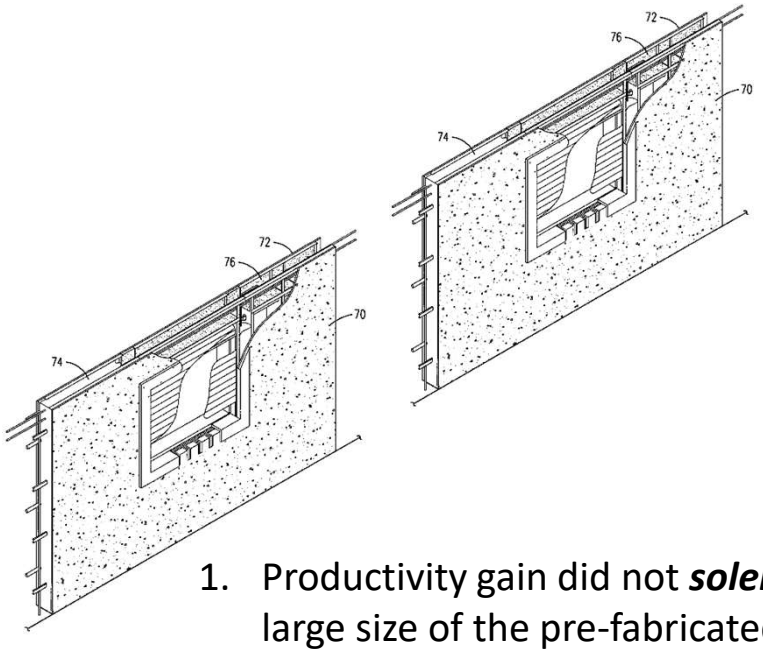
***All Wires** – functional and infrastructural (reset, clocks and power grid) are created as a result of abutment*

***Cost-Metrics** of the composite design becomes known with post layout accuracy*

Inspiration from Construction Industry

An Analogy

We shifted to pre-fabricated wall segments



1. Productivity gain did not ***solely*** come from the large size of the pre-fabricated wall segments

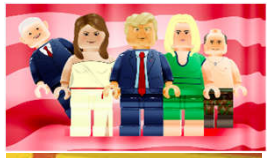
2. Productivity gain came from physical design discipline that enables composition by abutment

3. IPs in VLSI Design lack this discipline and composition by abutment

Lego Kits The Berkeley Dwarfs SiLago Regions Types








	The Berkeley Dwarfs
1	Dense Linear Algebra
2	Sparse Linera Algebra
3	Spectral Methods
4	N Body Methods
5	Structured Grids
6	Unstructured Grids
7	MapReduce
8	Combinational Logic
9	Graph Traversal
10	Dynamic Programming
11	Back-track and Branch n Bound
12	Graphical Models
13	Finite State Machine

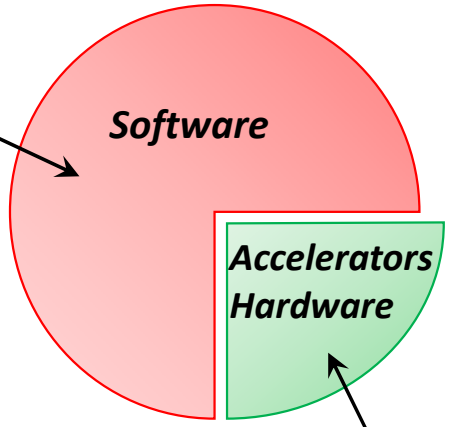
Region Types – SiLago Block Types

Functional	***Infrastructural***
Graph Theory	NOCs
Outer Modem	Scratch Pad Memory
Inner Modem	PLL + CGU
Protocol Processing	Power Management
Spectral Methods	Memory Controller
Dense Linear Algebra	FIFO, FIFO Controller
Sparse Linear Algebra	*RISC Processors – RISC-V*
Dynamic Programming	DMA
State Machines	Memory Consistency

Hardware Centric vs. Software Centric Accelerators vs. Flexilators

Software Centric Platform Based Design

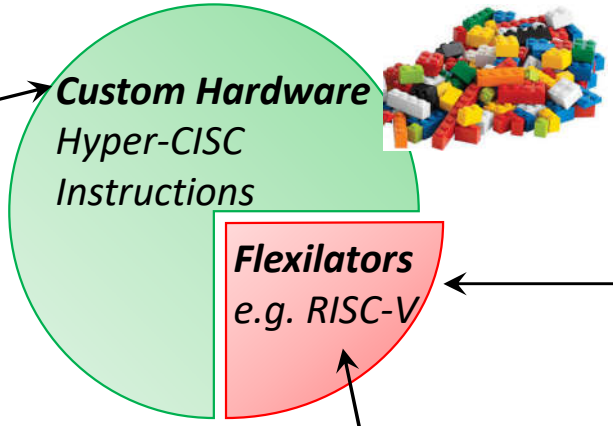
By default Functionalities are mapped as software



Only power and performance critical functionalities are mapped as hardware accelerators

Hardware Centric Synchoros VLSI Design

By default Functionalities are mapped as custom functional hardware

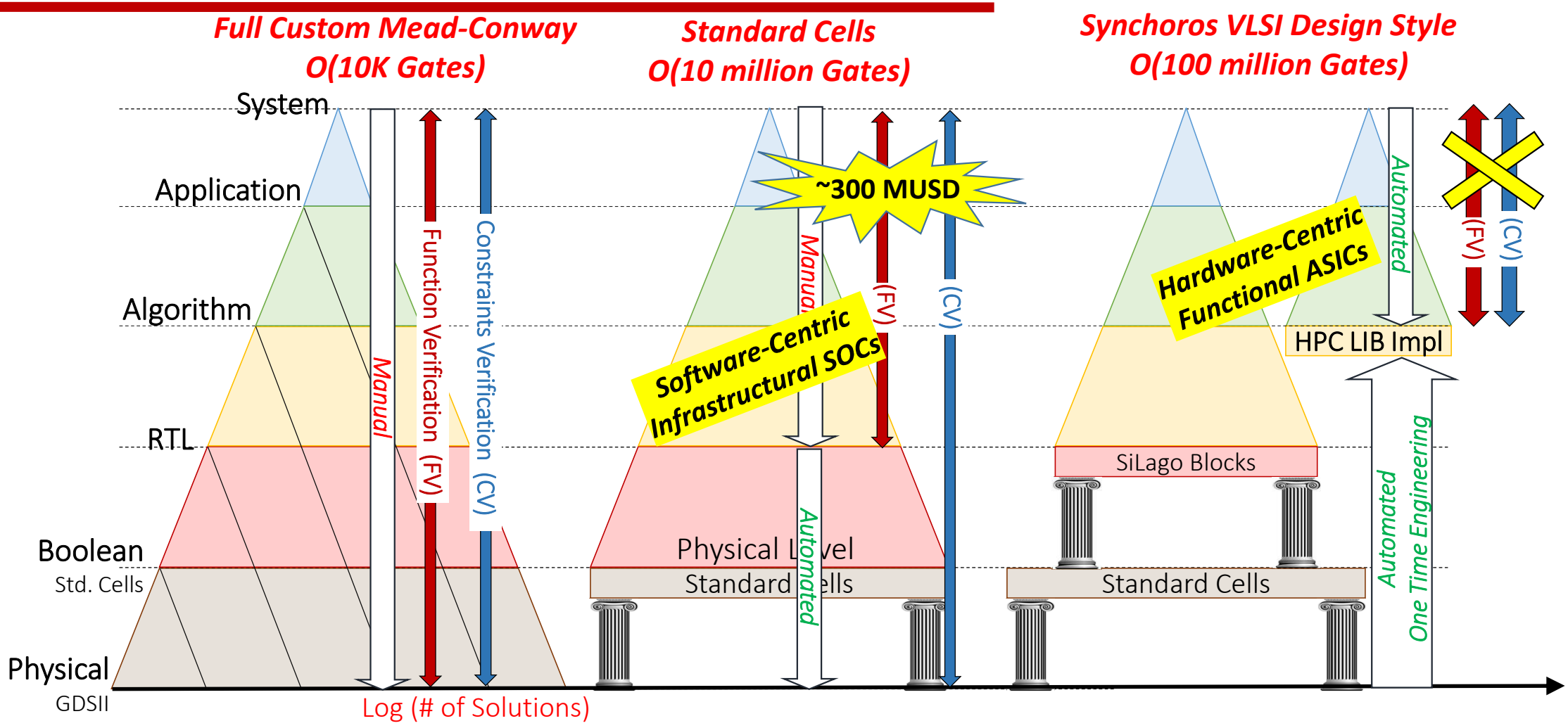


Only flexibility critical, dynamic and non-deterministic functionalities are mapped to

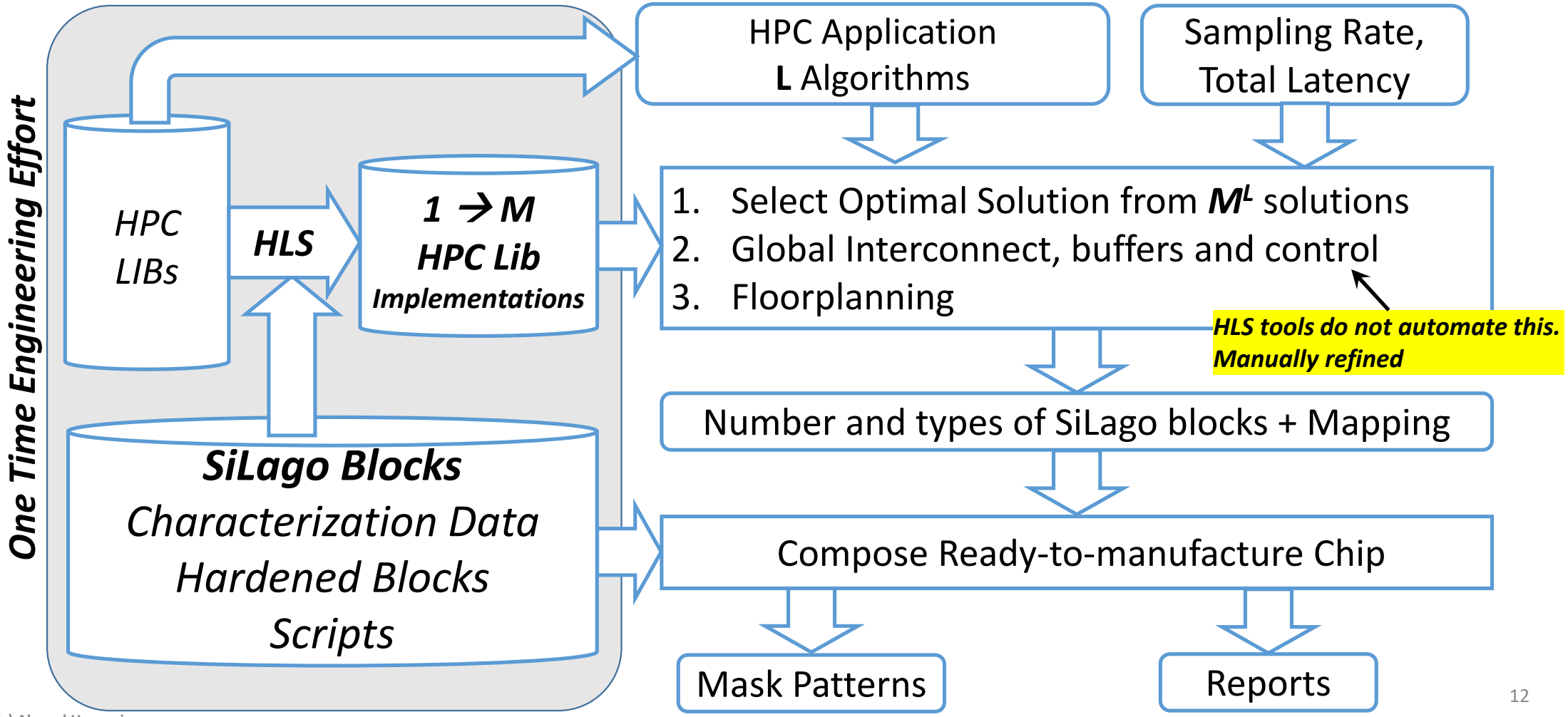
SiLago Flexilators: RISC-V, FSMs, FIFOs, Arbiters, Schedulers, NOCs etc.

Lego Flexilators

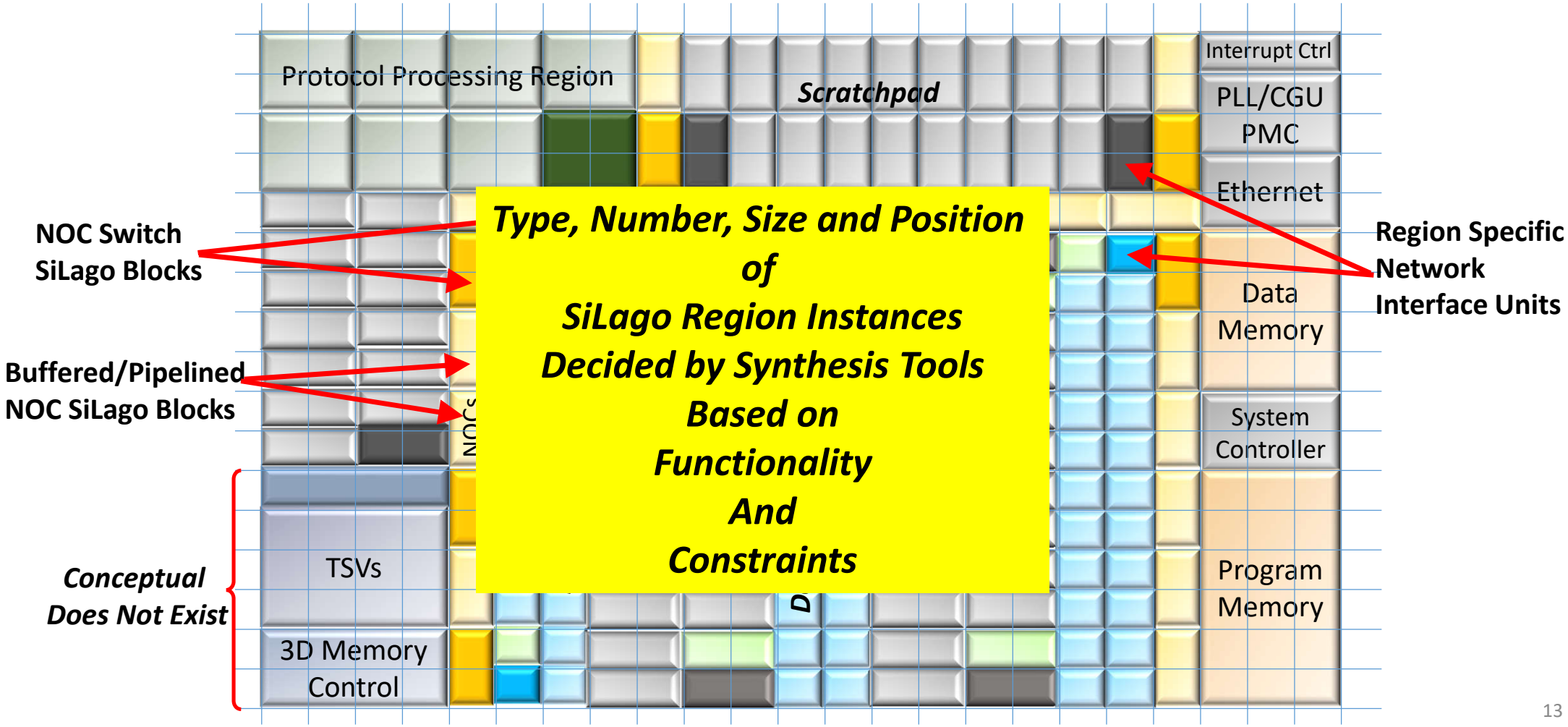
Why does Synchoros VLSI Design Work ?



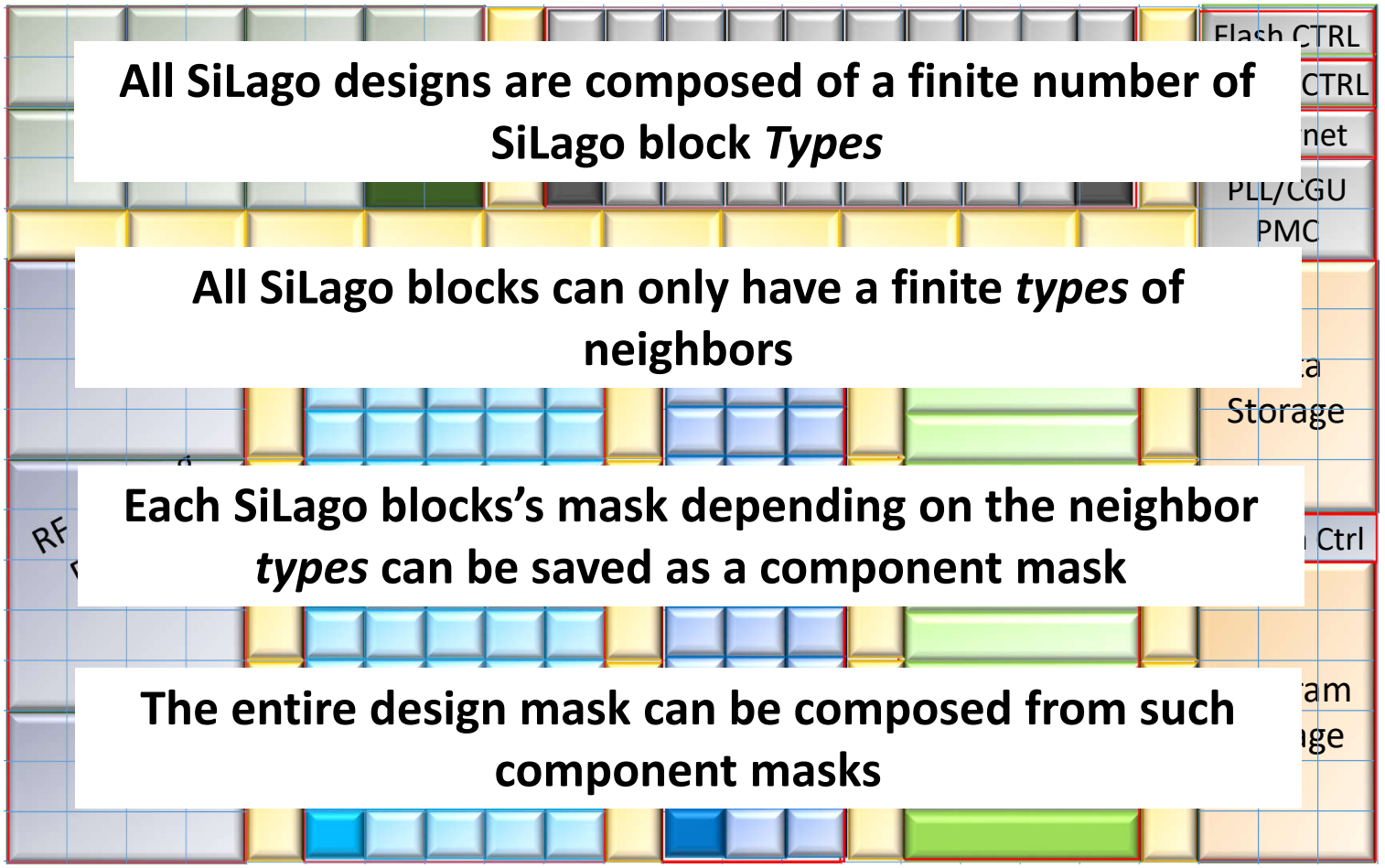
SiLago Application Level Synthesis



SiLago Design Instances = Σ Region Instances



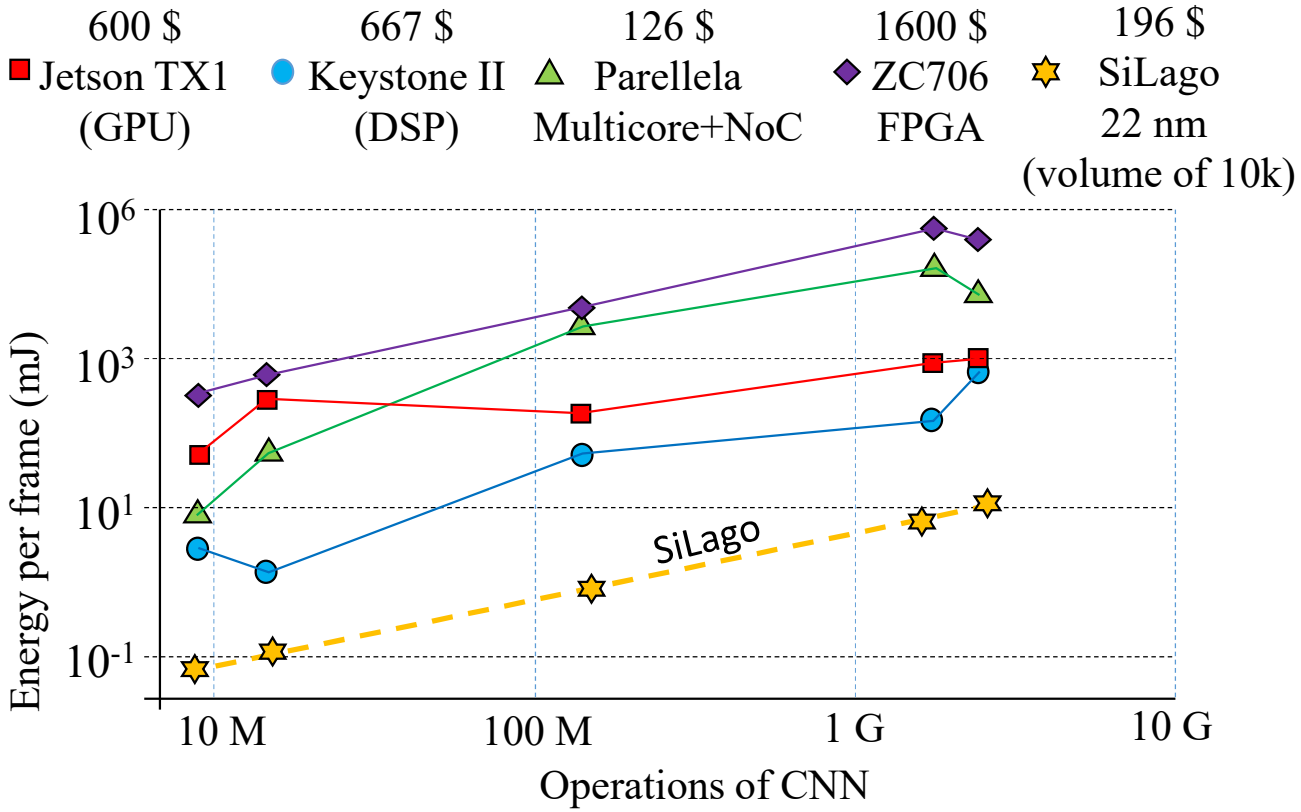
SiLago can also potentially reduce the manufacturing cost



The DFT Cost can also be factored out

The DFT can be made much more efficient reducing time spent on ATE

What becomes possible



Data on GPU, DSP, Parallela and FPGA adapted from
G. Hegde, S. Siddhartha, and N. Kapre, "CaffePresso: Accelerating convolutional networks on embedded SoCs," ACM Transactions on Embedded Computing System, vol. 17, 2017.

Going Beyond Moore !

Solutions to go beyond Moore

1. **Squeeze more out of CMOS**
   a. ASICs like custom functional hardware
   b. Delivers 2-4 orders better energy-delay product compared to GPUs, FPGAs and Multi-cores

2. **Complement CMOS with emerging technologies**
   a. 2.5D and 3D Integration (DRAM)
   b. Computation in memory using Memristors
   c. Plasmonics

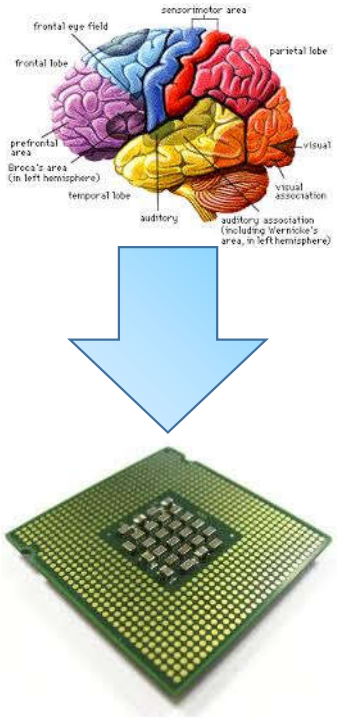
Make it easy to use the solution

"Science makes progress, not when you find a solution, but when you make it easy to use the solution"

-- Venki Ramakrishnan

BCPNN
Bayesian Confidence Propagation Neural Network
Professor Anders Lansner

BCPNN Requirements

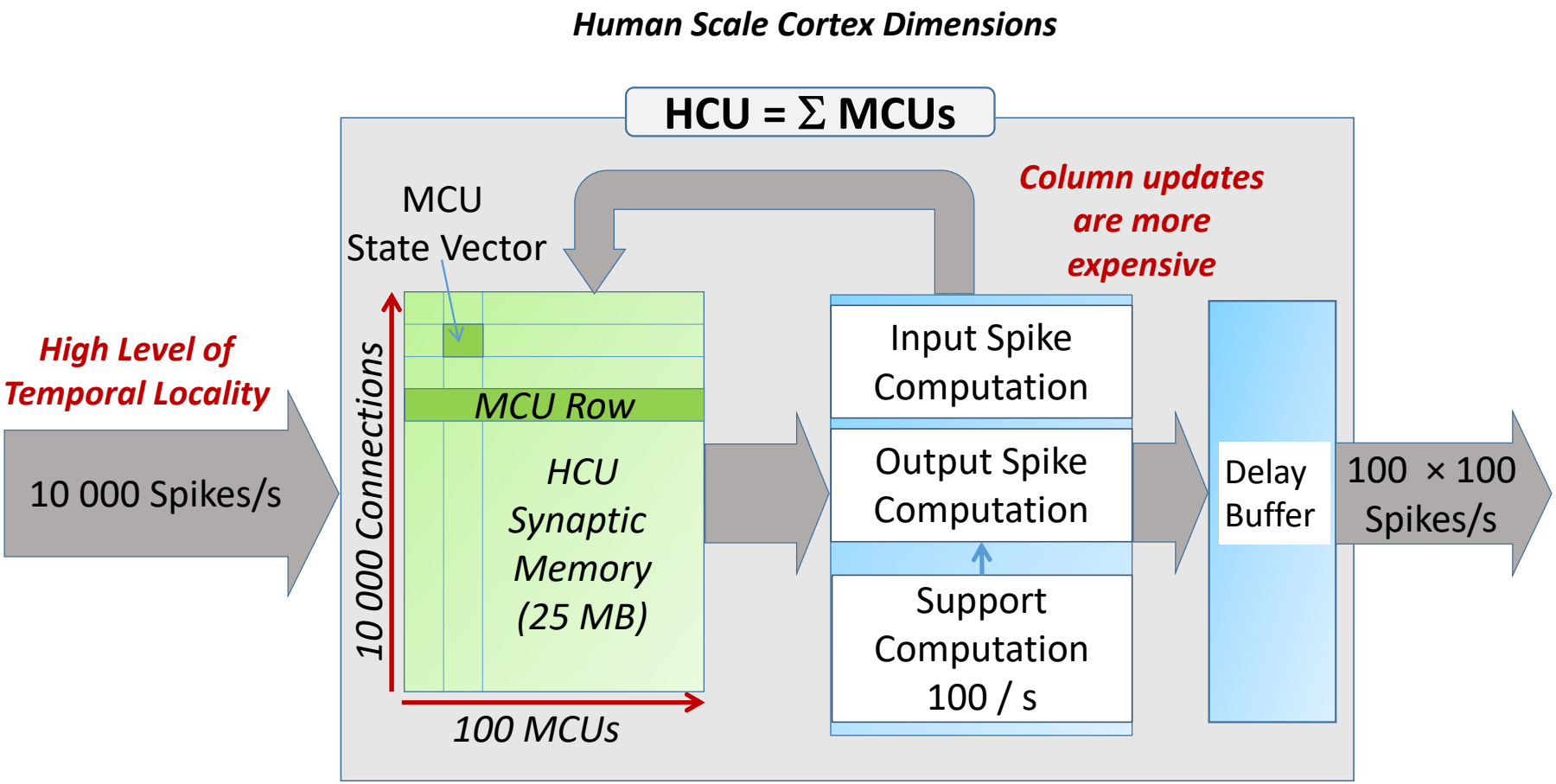


Functional Requirements: Human Scale - Realtime

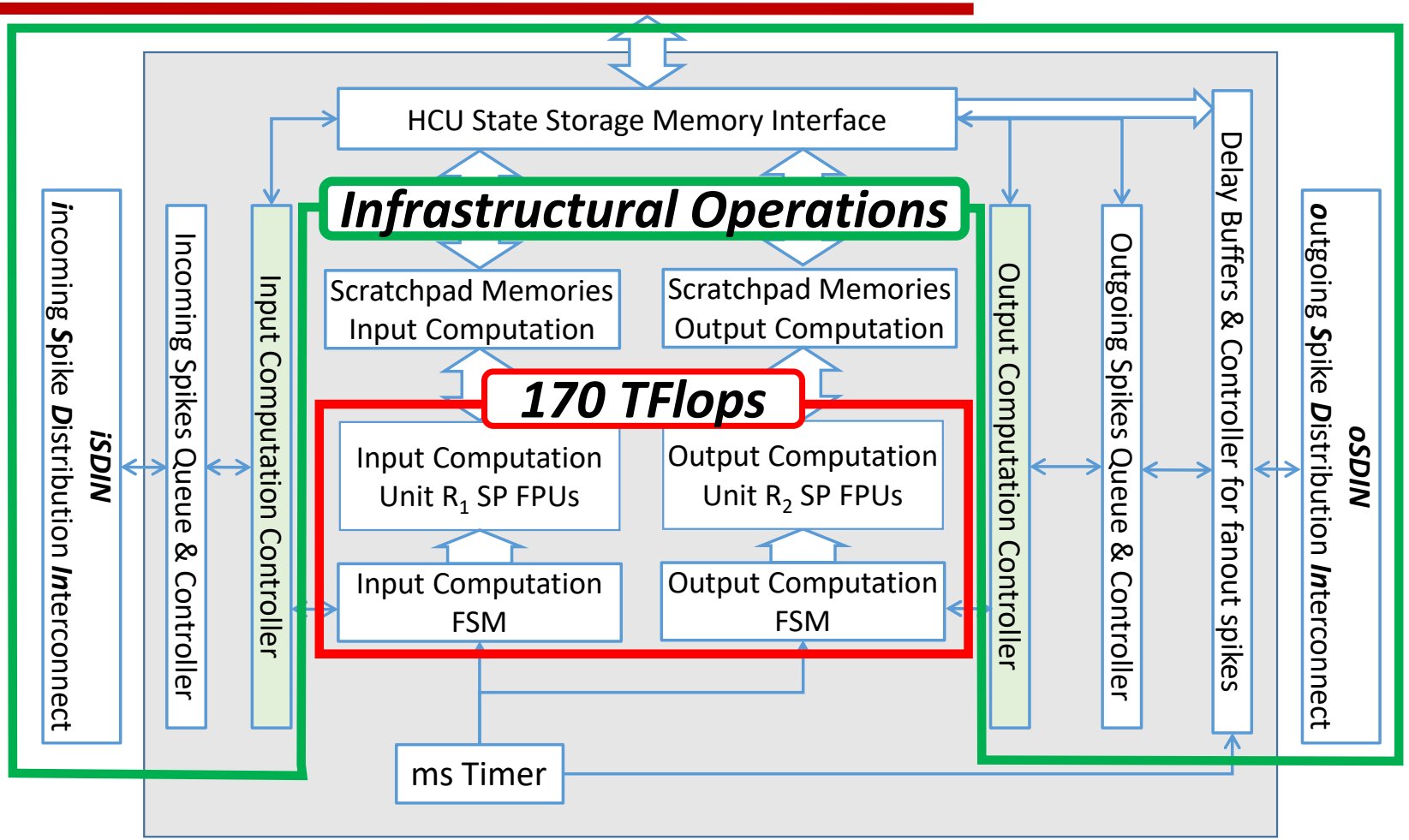
1. ***Realtime simulation***
2. ***2 Million HCUs – non-deterministically concurrent***
3. ***170 TFlops/s – BCPNN Computation***
4. ***50 TBs – Synaptic Weight Storage***
5. ***200 TBs / s – Bandwidth for synaptic storage***
6. ***250 GBs / s – Spiking Bandwidth***

Infrastructural Requirements

The BCPNN Computation Model

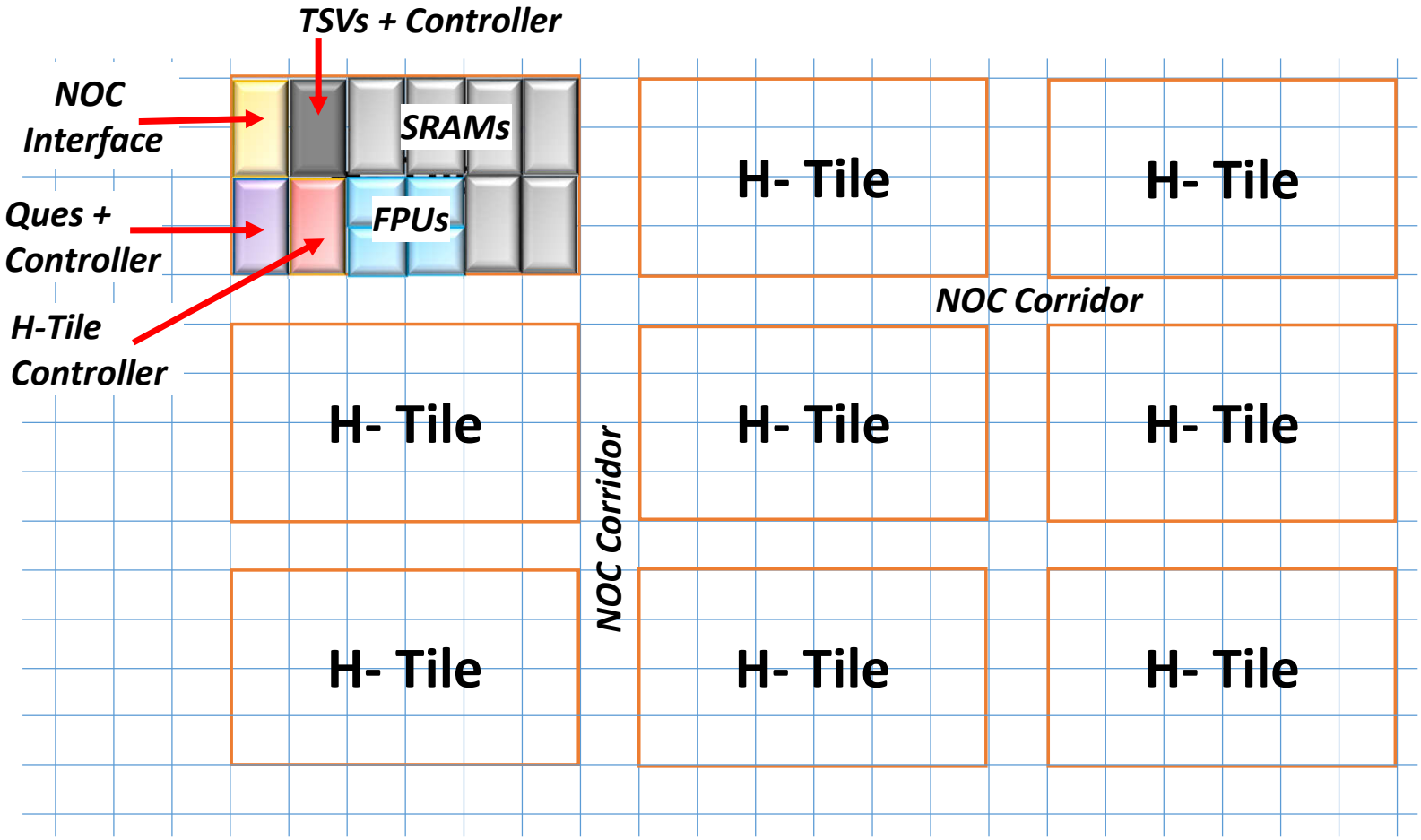


Infrastructural Operations are Significant

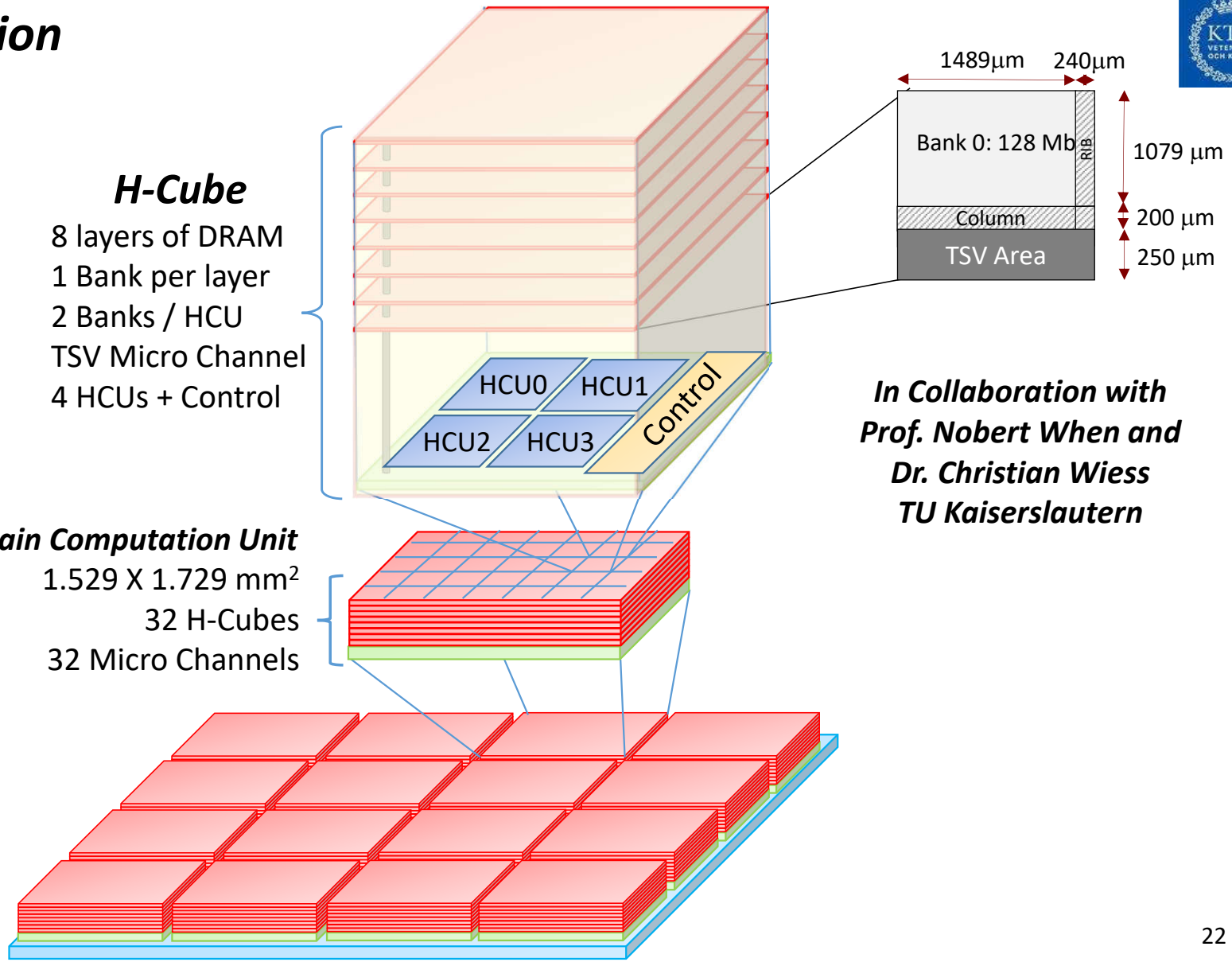


The Silicon Lego Bricks for Method Applied to BCPNN

A Structured Physical Design Scheme to enable System-level synthesis



BCPNN Implementation



H-Cube

8 layers of DRAM
1 Bank per layer
2 Banks / HCU
TSV Micro Channel
4 HCUs + Control

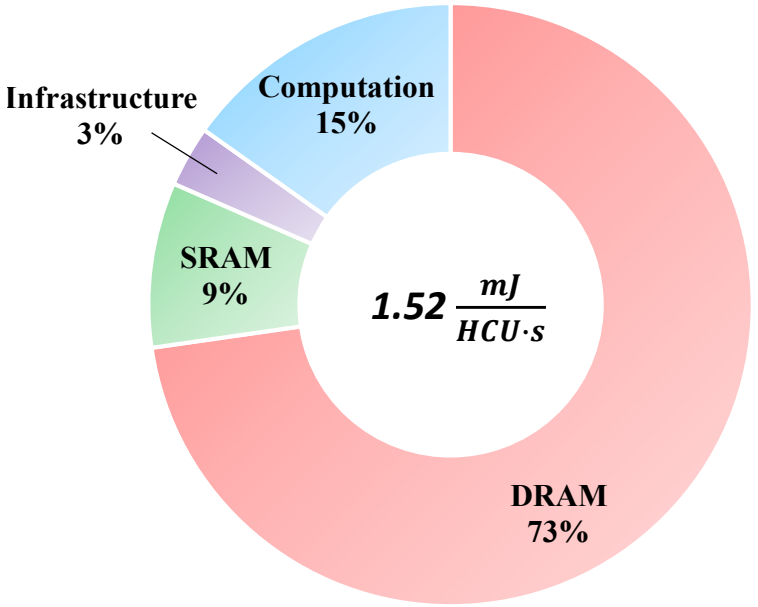
BCU: Brain Computation Unit

1.529 X 1.729 mm^2
32 H-Cubes
32 Micro Channels

In Collaboration with Prof. Nobert When and Dr. Christian Wiess TU Kaiserslautern

BCPNN: ASIC vs GPUs

(a) Energy Breakdown ASIC

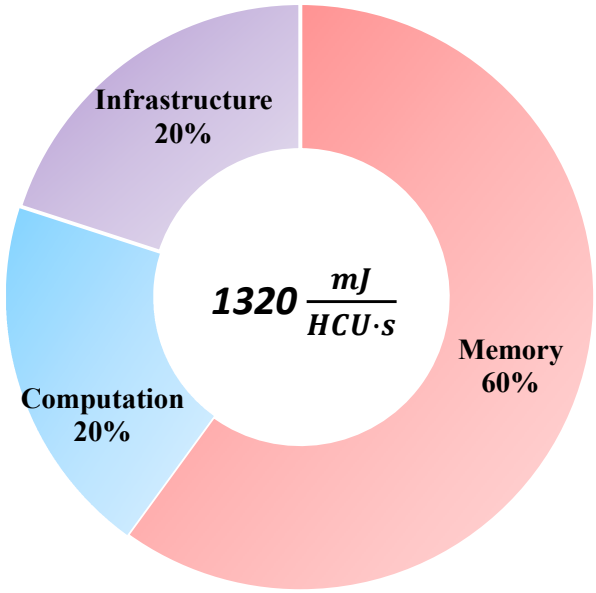


ASIC: 3.0 kW

GPUs: 2.6 MW

SpiNNaker-2 comparable to GPUs

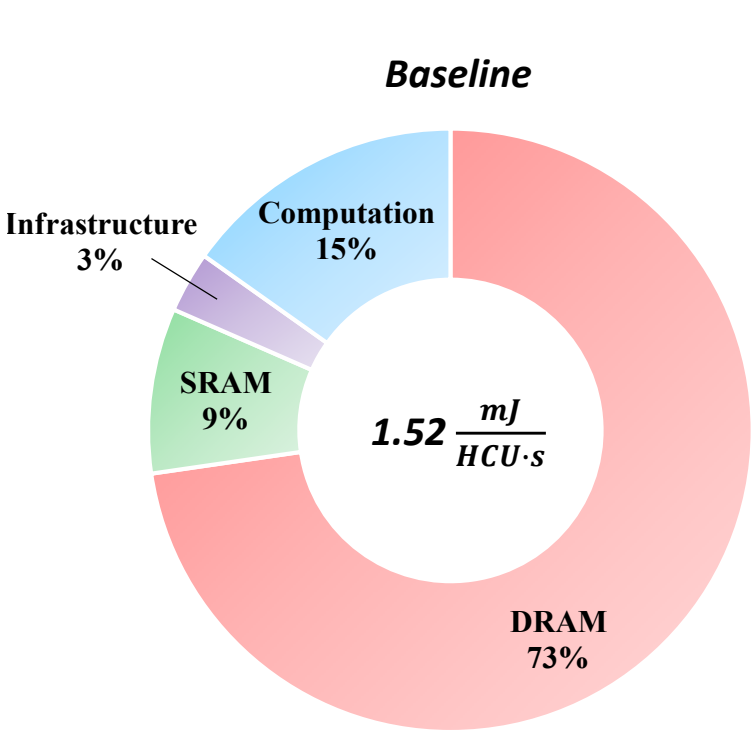
(b) Energy Breakdown GPUs



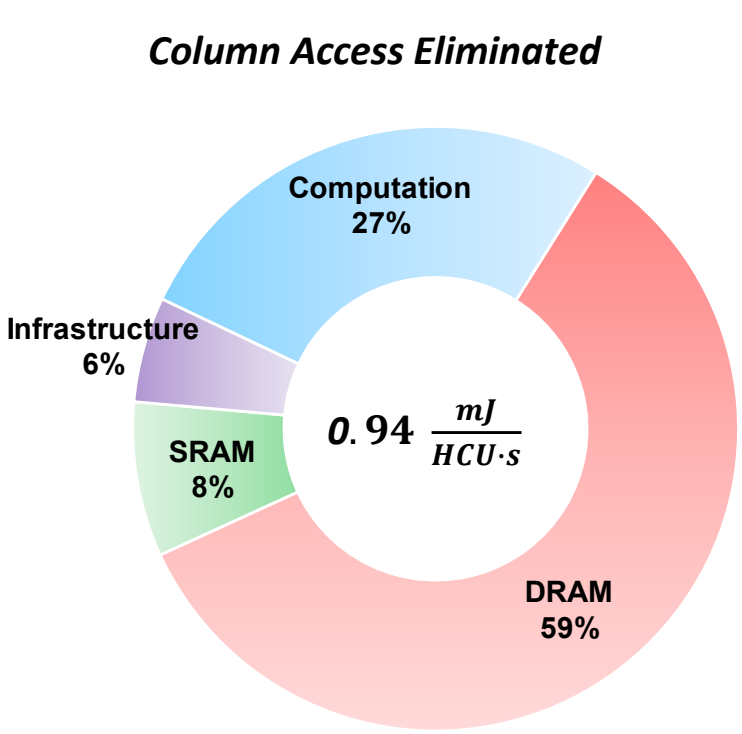
Energy Delay Product: 3.06 kJ · s

# of GK210 cores : 5000
Energy : 563.1 kJ
1s Realtime : 4.69 s simulated time
Energy Delay Product : 2642 kJ · s

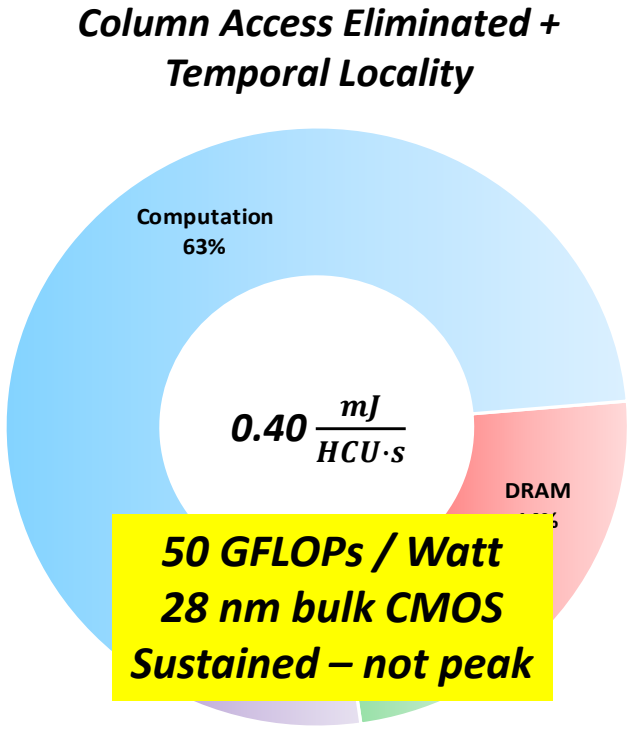
The Impact of Column Access Elimination + Exploiting Temporal Locality



3 kW



1.88 kW



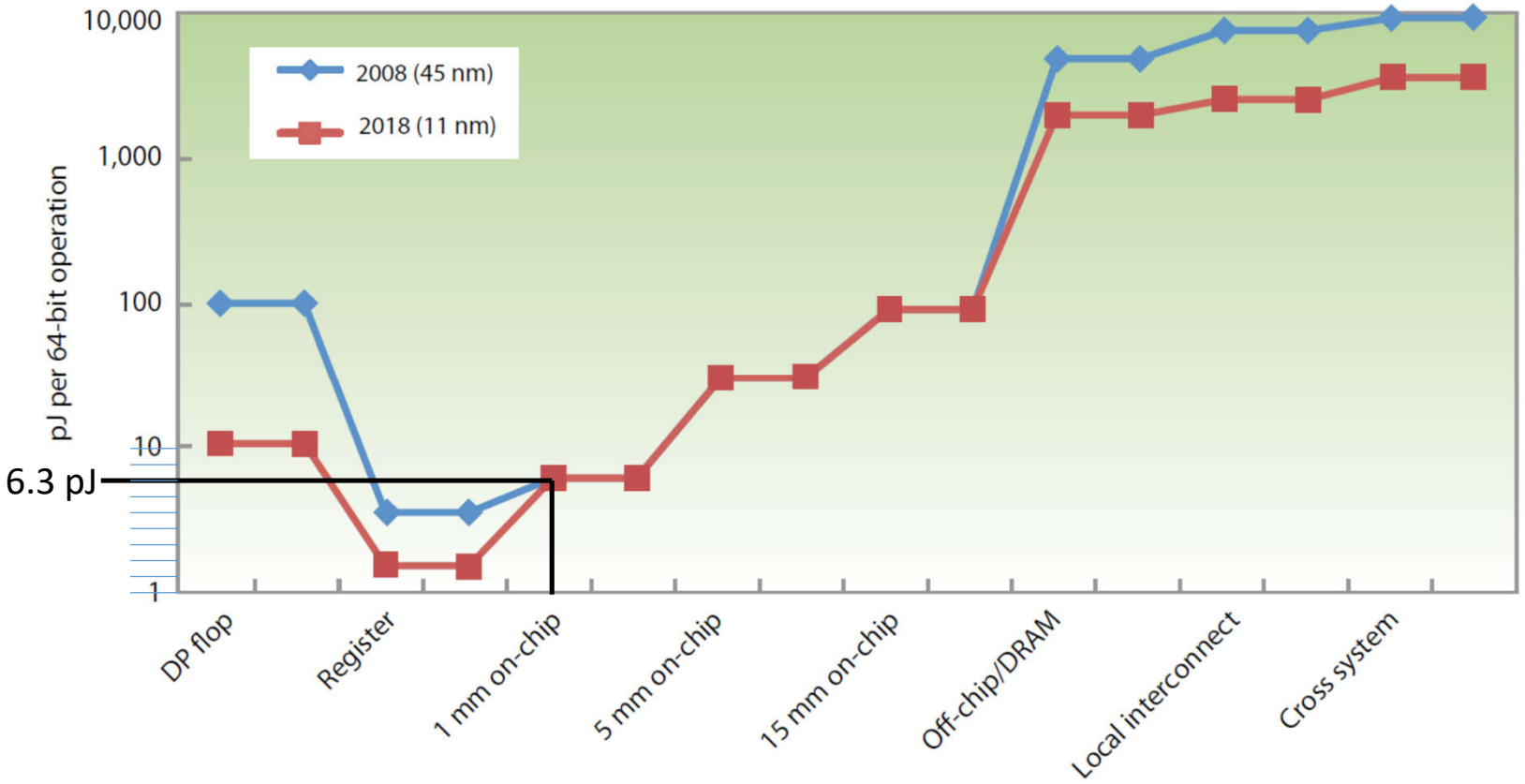
800 W

50 GFLOPs / Watt
28 nm bulk CMOS
Sustained – not peak

Such optimizations can be automatically inferred from Simulations

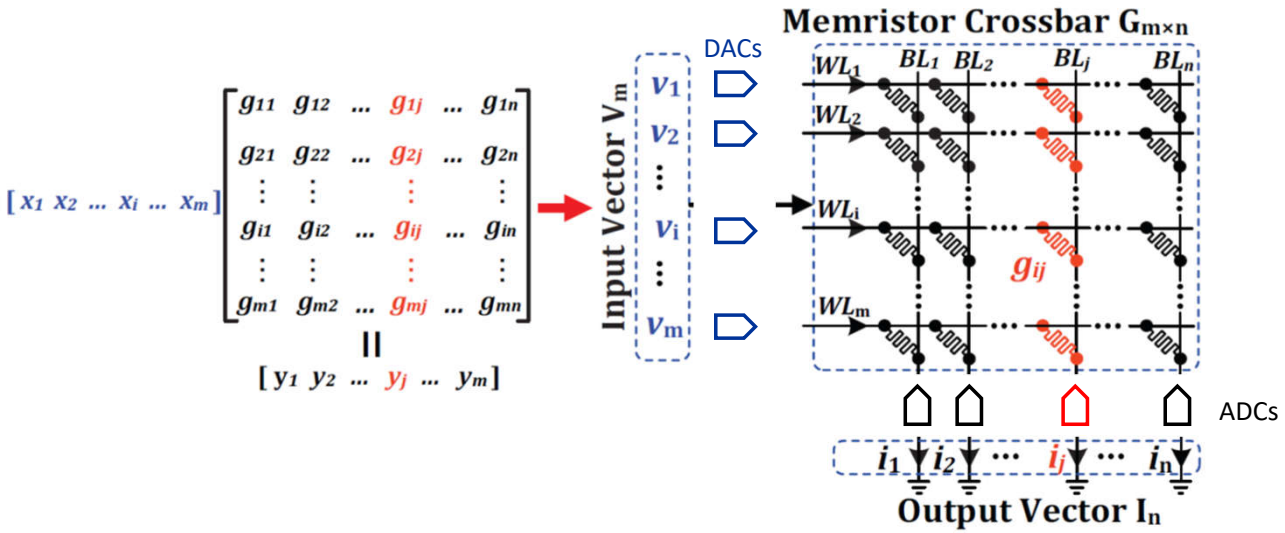
Interconnect and Storage are Expensive

3.2 pJ = 32 bit Data 1 mm ~= 32-bit FLOP > accessing 1 bit in 3D integrated DRAM



P. Kogge and J. Shalf, "Exascale computing trends: Adjusting to the 'new normal' for computer architecture," *Comput. Sci. Eng.*, vol. 15, no. 6, 2013.

Computation in Memory using Memristors



Reminiscent of Analog Computation

Benefits:

1. Single cycle dot product
2. Can be extended to do addition, multiplication, element wise multiplication, matrix inversion
3. No need to fetch, decode and execute instructions → addresses wire problem
4. In some application instances, initialization of matrix would be a one-time event

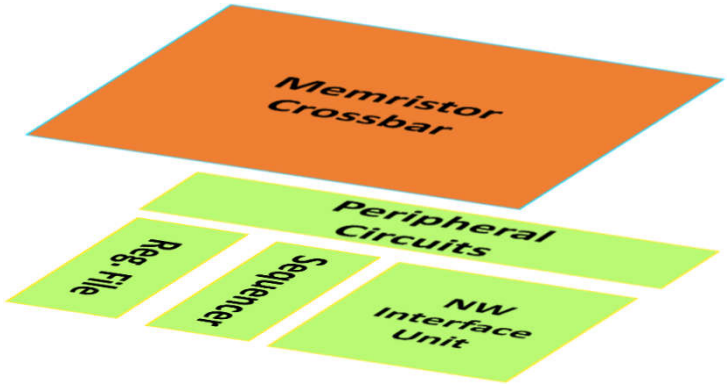
Challenges

1. Large matrices will need to be fragmented resulting in movement of data. Need complimentary control circuitry
2. ADC's consume significant power and inject latency
3. Accuracy
4. Experimental solutions reported. Not part of mainstream design flow

Source of Diagram Above: Chenchen Liu, Qing Yang, Bonan Yan, Xiaocong Du, Hai (Helen) Li, "A Memristor Crossbar Based Computing Engine Optimized for High Speed and Accuracy", ISVLSI 2016

Memristor based CIM in the SiLago Framework

Region Types – SiLago Block Types	
Functional	***Infrastructural***
Graph Theory	NOCs
Outer Modem	Scratch Pad Memory
Inner Modem	PLL + CGU
Protocol Processing	Power Management
Spectral Methods	Memory Controller
Dense Linear Algebra	FIFO, FIFO Controller
Sparse Linear Algebra	*RISC Processors – RISC-V*
Dynamic Programming	NVM
State Machines	DRAM Vaults
Memristor CIM	



1. A Memristor CIM in a range of dimensions
2. Characterized with post-layout data and circuit level simulations and validated with test chips
3. Exports, functional matrix operations and infrastructural operations like initializing crossbar, NIU operations, reg file operations etc.
4. Higher abstraction synthesis tools can refine in terms of CIM SiLago blocks and know its performance, energy and area.

25 Watt Biologically Plausible Human Scale Brain

Single Precision Floating Point

Move to 16/32 bit Integer Arithmetic

1. **Synaptic Storage/Access will reduce by ~50%**
2. **Computation Energy will reduce by ~75%**

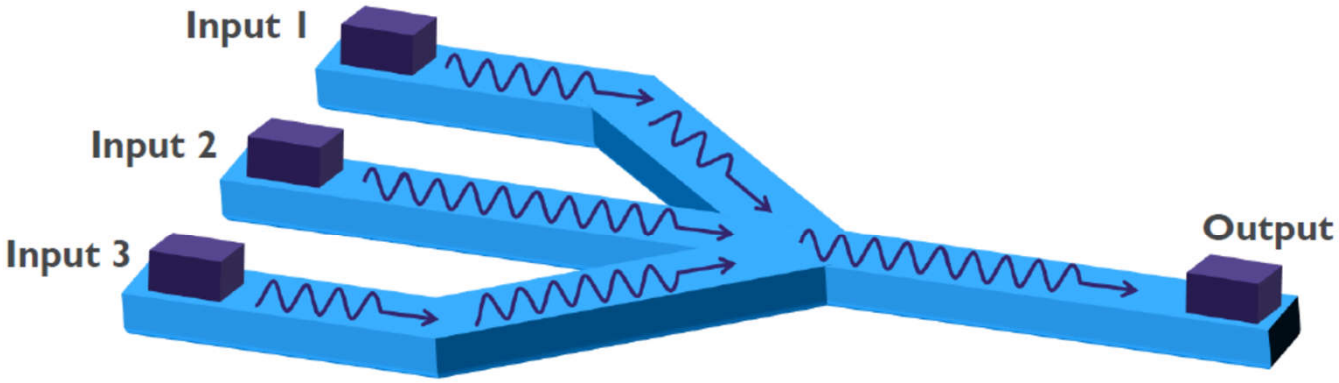
Caveat:
Based on best effort estimates and not on actual implementation

~800 Watts → **~250 Watts** → **ReRAM Computation in Memory** → **~25 Watts**

~ 2 TOPs/watt
28 nm bulk CMOS

Wave Based Computing using Plasmons

1. *Logic values encoded as phase of the waves*
2. *Interference of waves interpreted as majority gate computation*



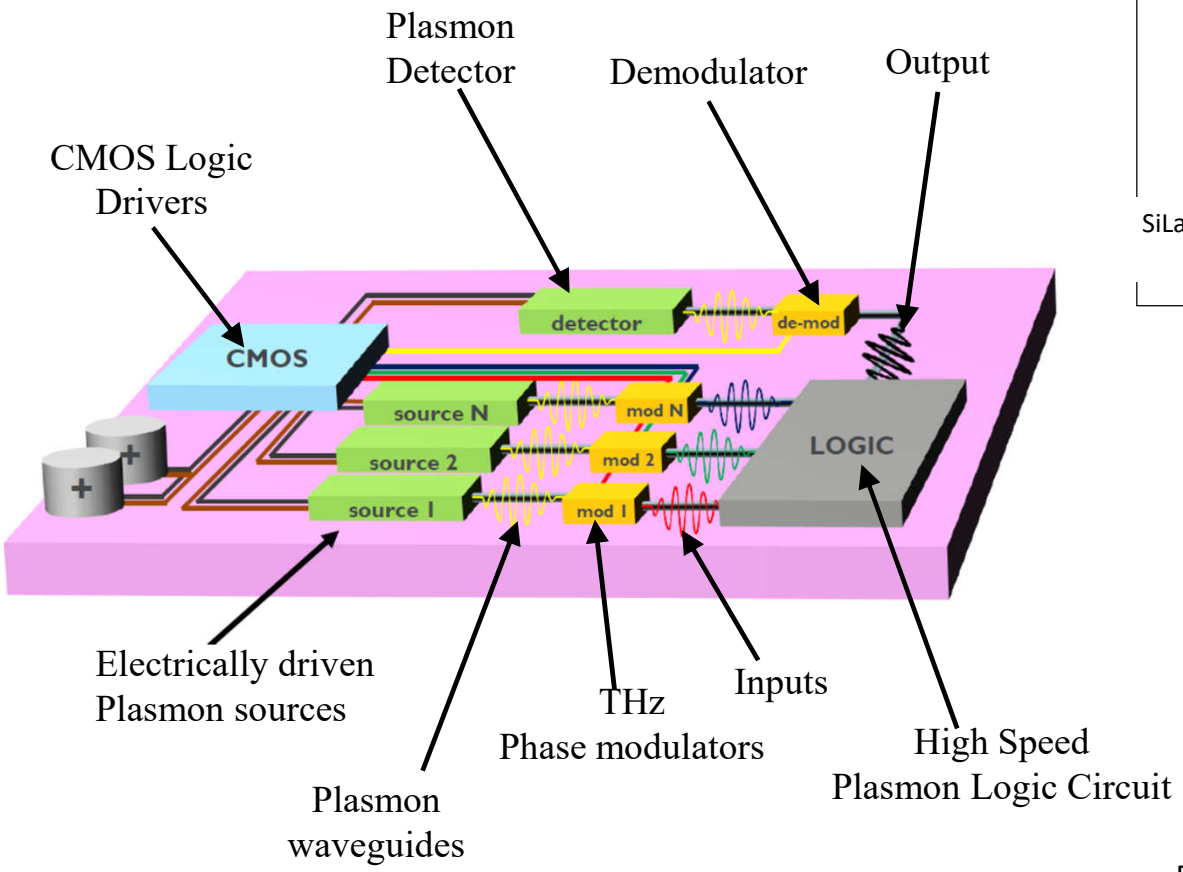
In wave computing, information is coded in the phase or the amplitude of the wave.

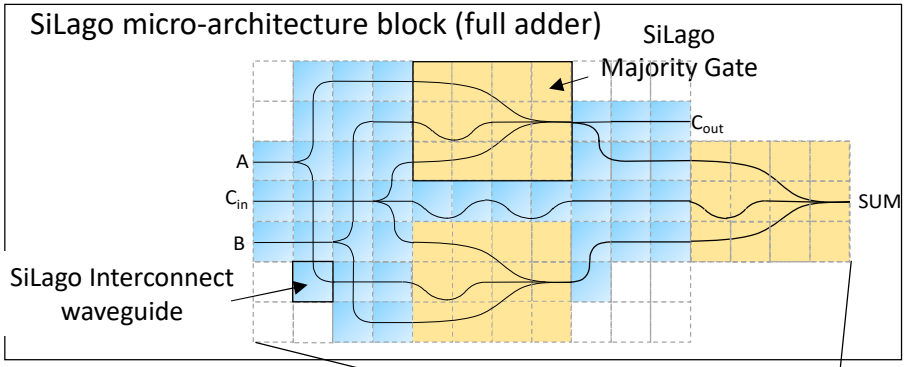
Computation by interference
Majority logic gate

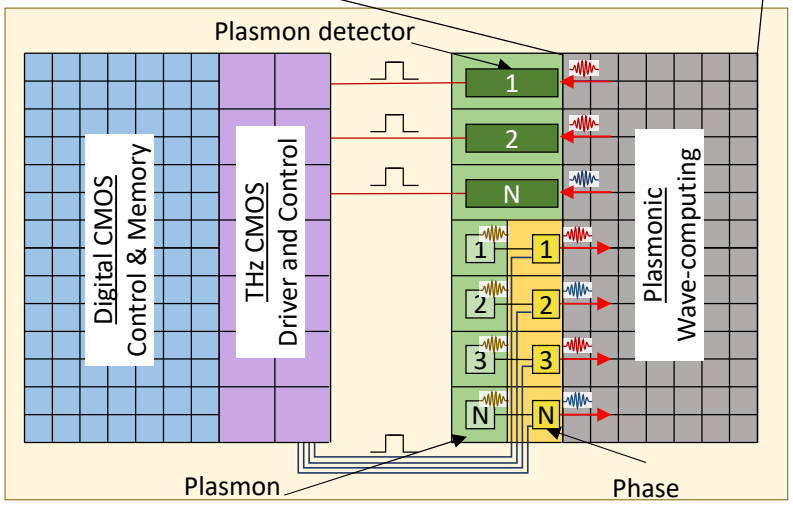
I1	I2	I3	O
0	0	0	0
0	0	1	0
0	1	1	1
1	1	1	1

Output phase after interference is equal to the majority of input phases.

Plasmonics + CMOS Computing using SiLago blocks







PlasmonExa – Plasmonics based Exa-scale computing design.
Synthesized in terms of Silicon Lego (SiLago) blocks.

Impact

1. 1000 X Power Density
2. More Affordable

Software Centric / GPU + Based Computing

1. 1000 X Power Density
2. More Affordable

Hardware Centric SiLago Based Computing