The SiLago Method: Next Generation VLSI Architectures and Design Automation

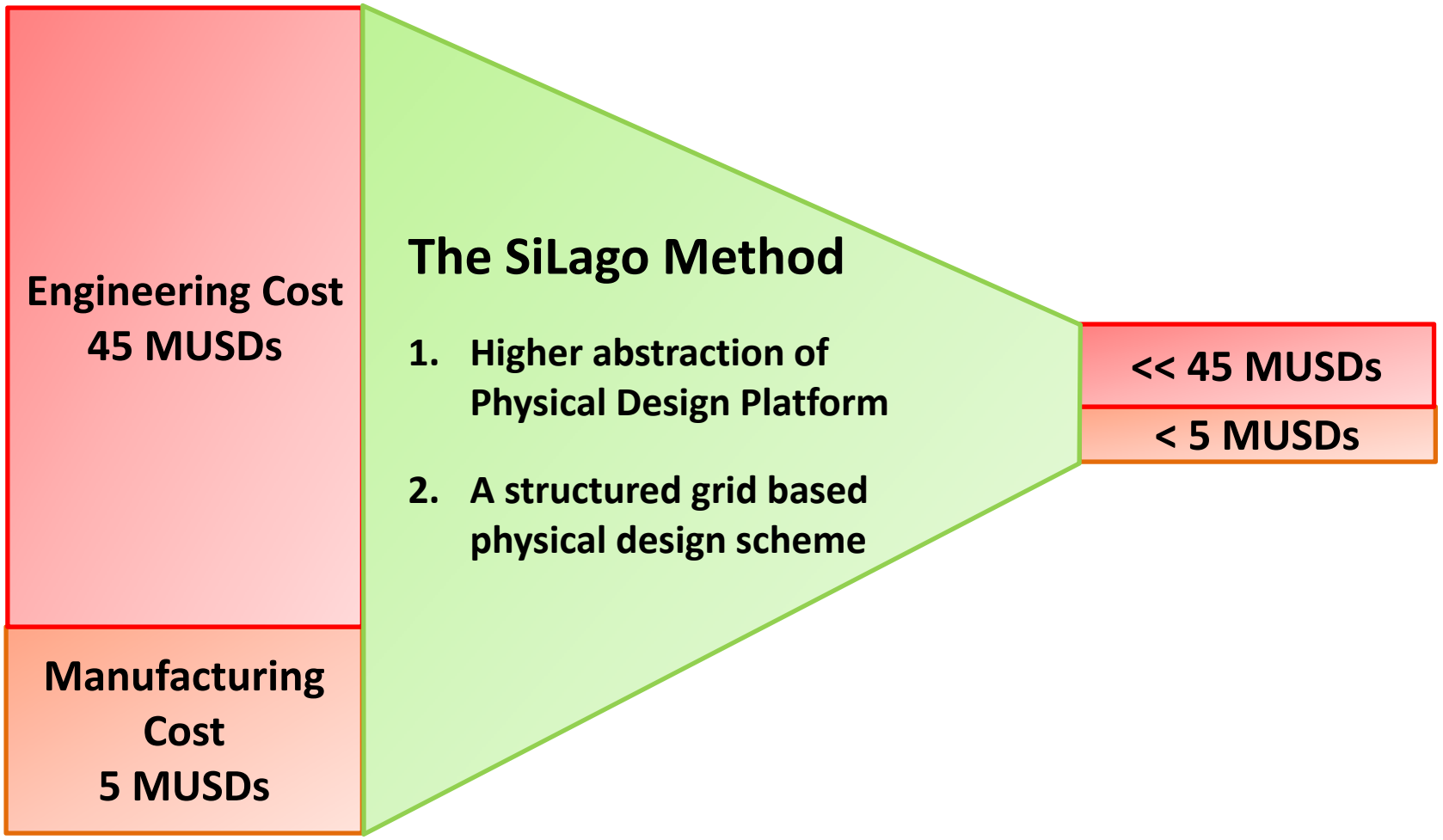
Ahmed Hemani

KTH – Dept of Electronics and Embedded Systems, School of ICT, KTH.

Acknowledgement:

Nasim Farahini, Muhammad Asad, Li Shuo, Hassan Sohofi, Muhammad Ali Shami, Adeel Tajammul, Omer Malik, Anders Lansnser, Christer Svensson

The Core Ideas behind the SiLago Method



The Large Engineering and Manufacturing Cost

Software Centric Accelerator rich Platform Based Design

Loss in Silicon and Computational Efficiencies

Blocks Application Categories Stifles Innovation

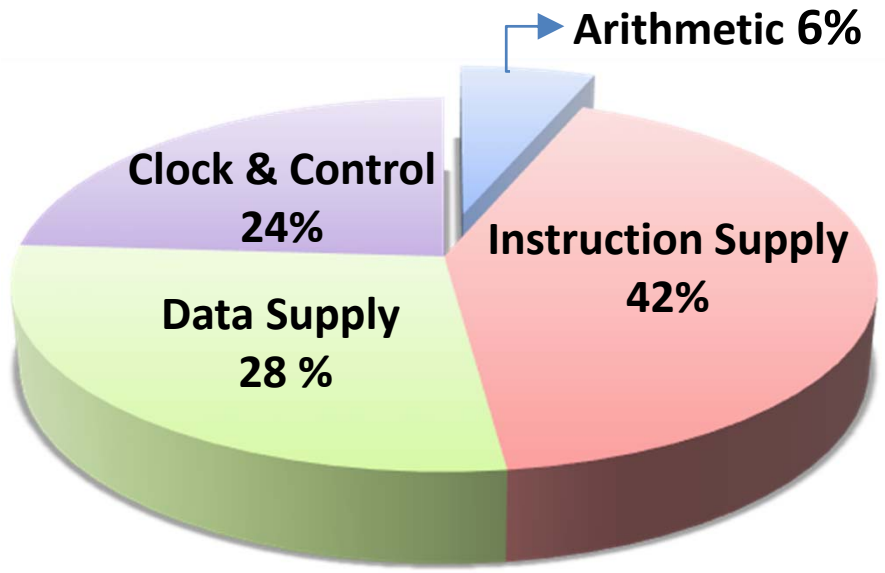
Generality vs. Customisation

Accelerator Rich Software Centric Platform Based SOC Design

The SiLago Method

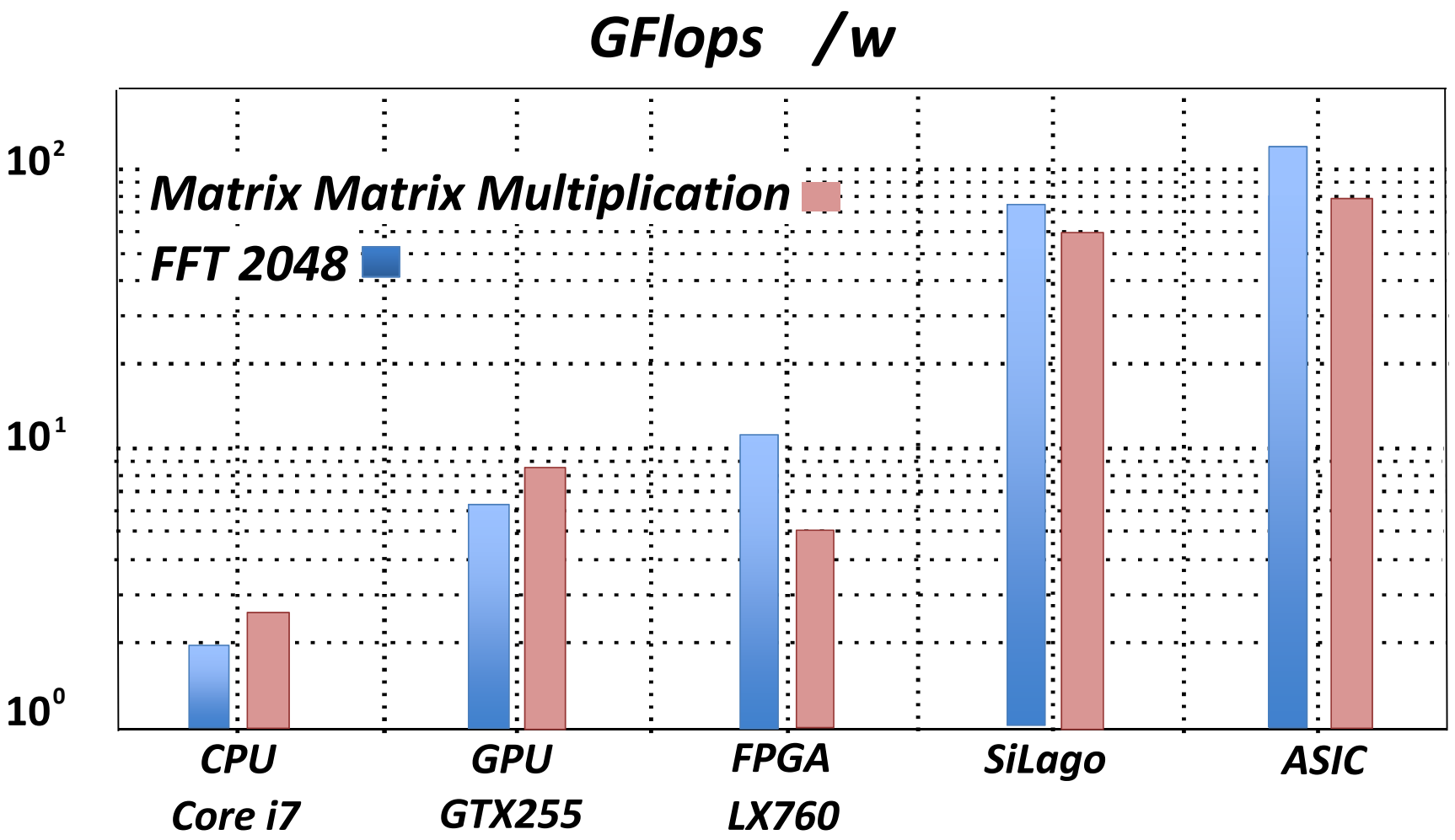
Hardware Centric Custom Design

Energy Breakdown in a GPP



William J. Dally, James Balfour, David Black-Shaffer, James Chen, R. Curtis Harting, Vishal Parikh, Jongsoo Park, and David Sheffield, Stanford University, ”Efficient Embedded Computing”. IEEE Computer July 2008

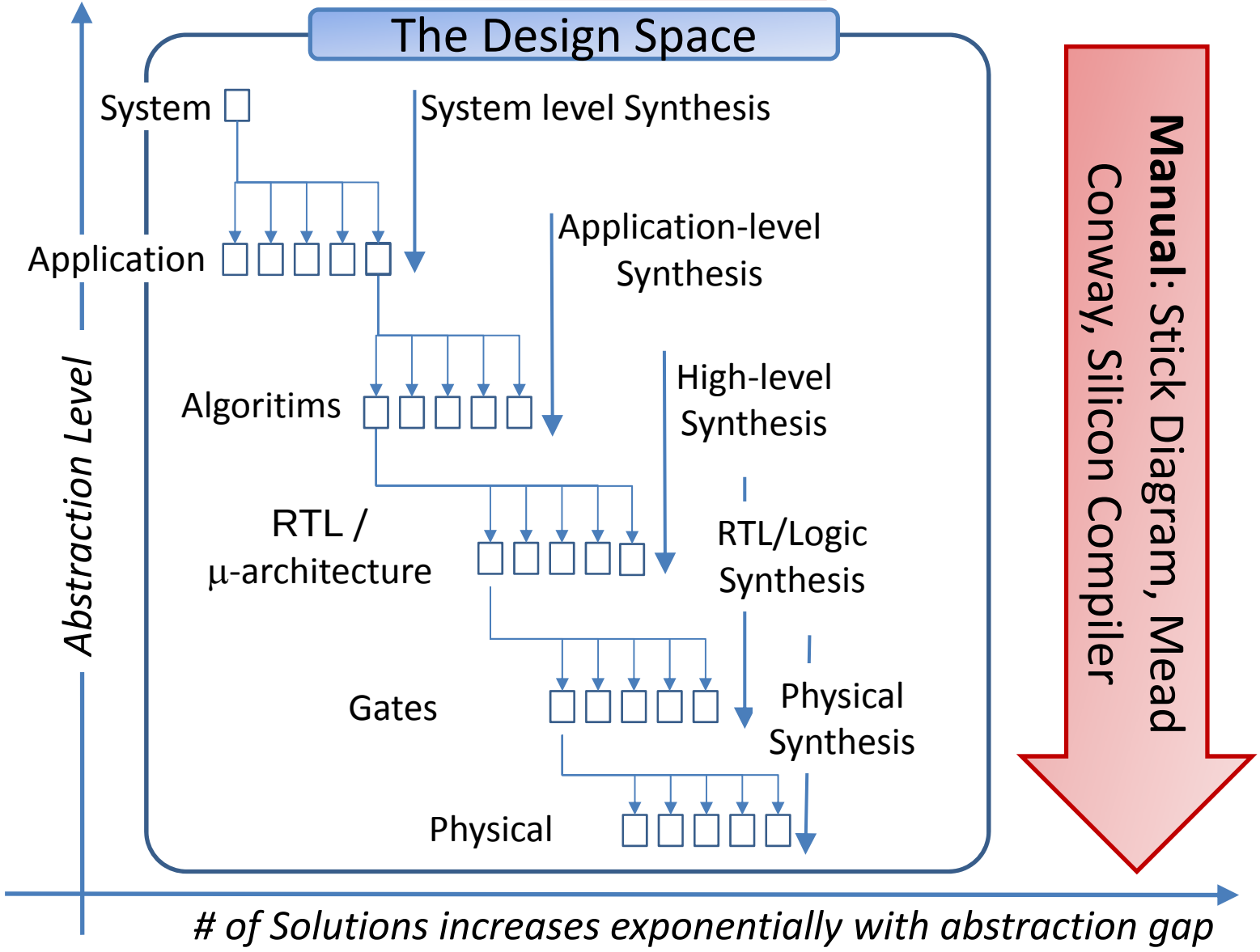
The Impact of Customization



A Brief History VLSI Design Automation

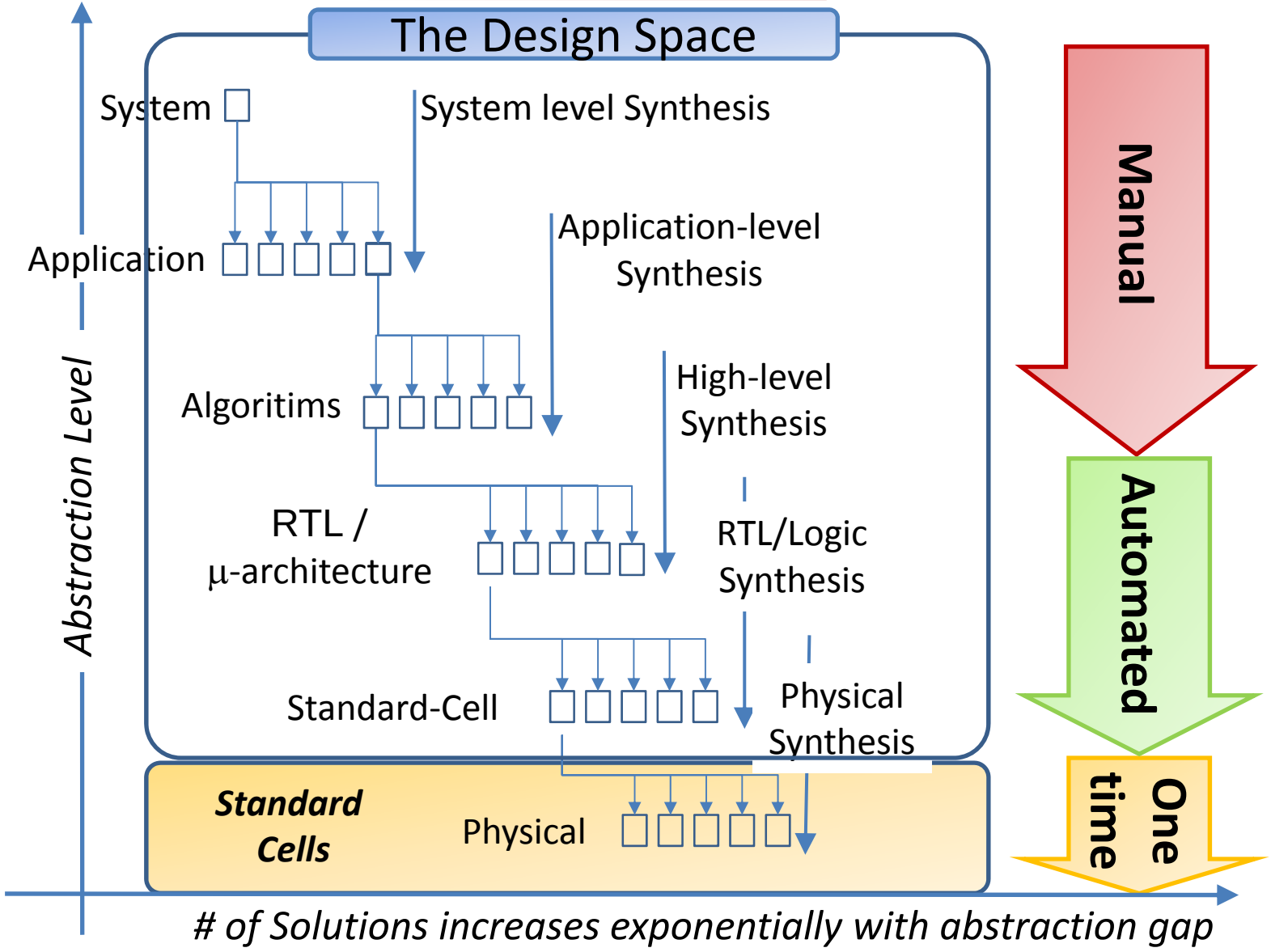
To Explain Why the Path of Customization has been abandoned

The Mead Conway Era



The Mead Conway Era Survived As long as the complexity was of the order of O(10K gates)

The Standard Cell Era



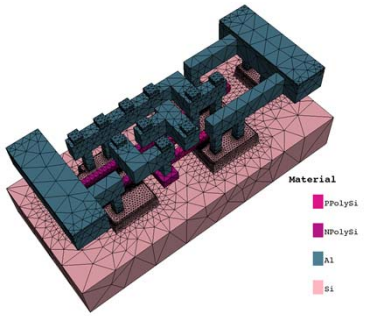
What Standard Cells Did

Abstraction

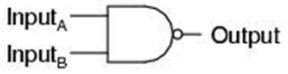
Boolean level abstraction

Hides circuit and physical design details

Enabled logic synthesis



NAND gate



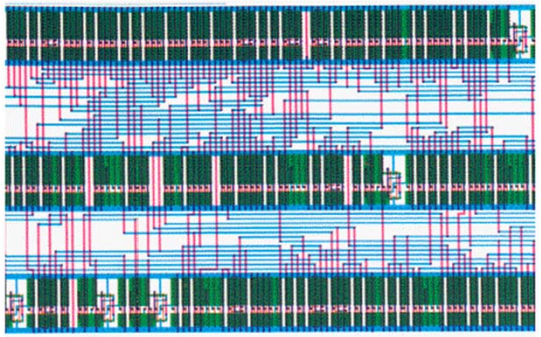
A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

Physical Design Discipline

Standard pitch and Row based layout

Enabled physical design automation

Standard Cell Layout

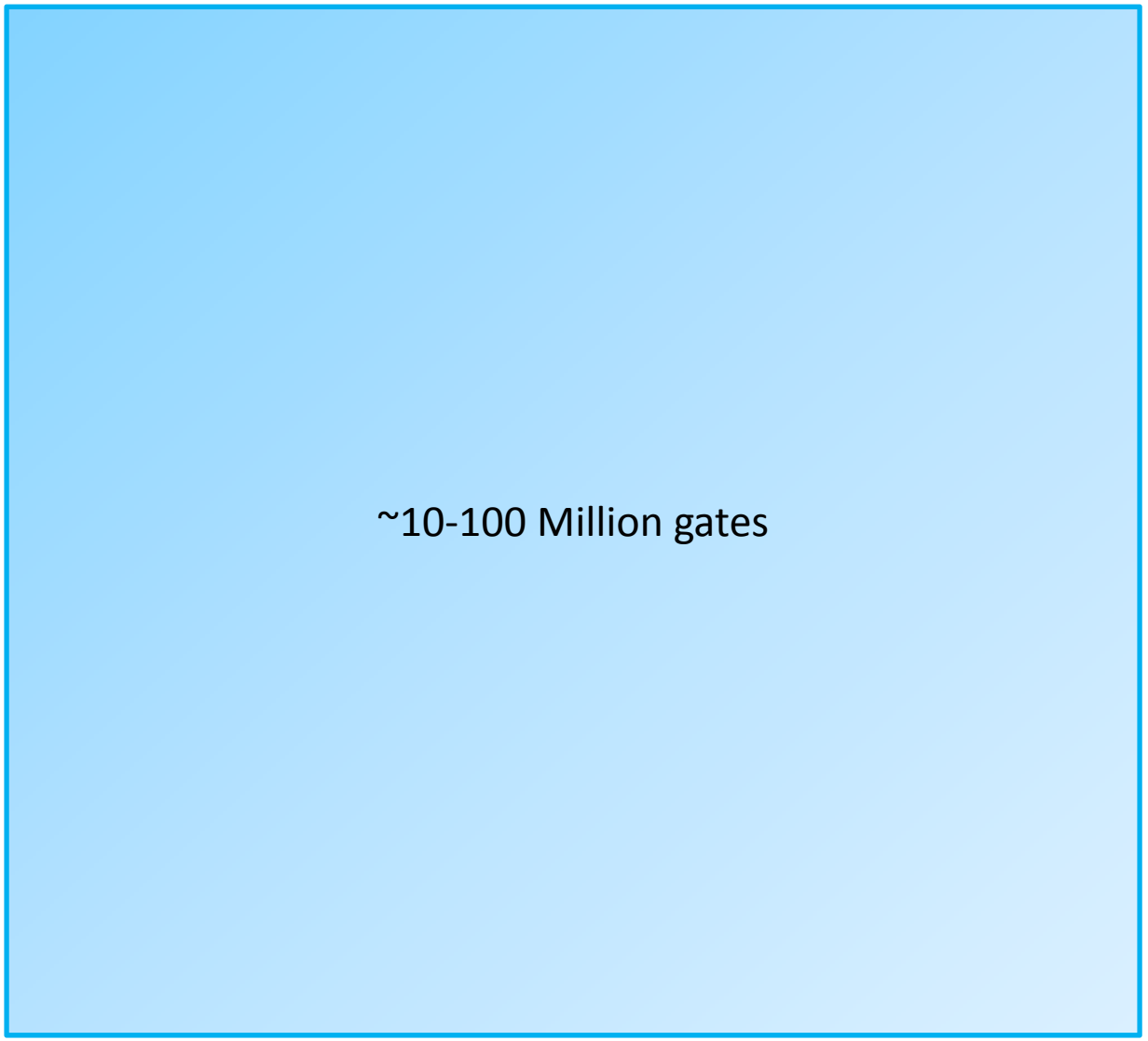


Improves efficiency of

1. **Synthesis** from RTL to GDSII
2. **Verification** at RTL
3. **System Design**

Standard Cells as building blocks are not scalable for 10-100 million gate designs

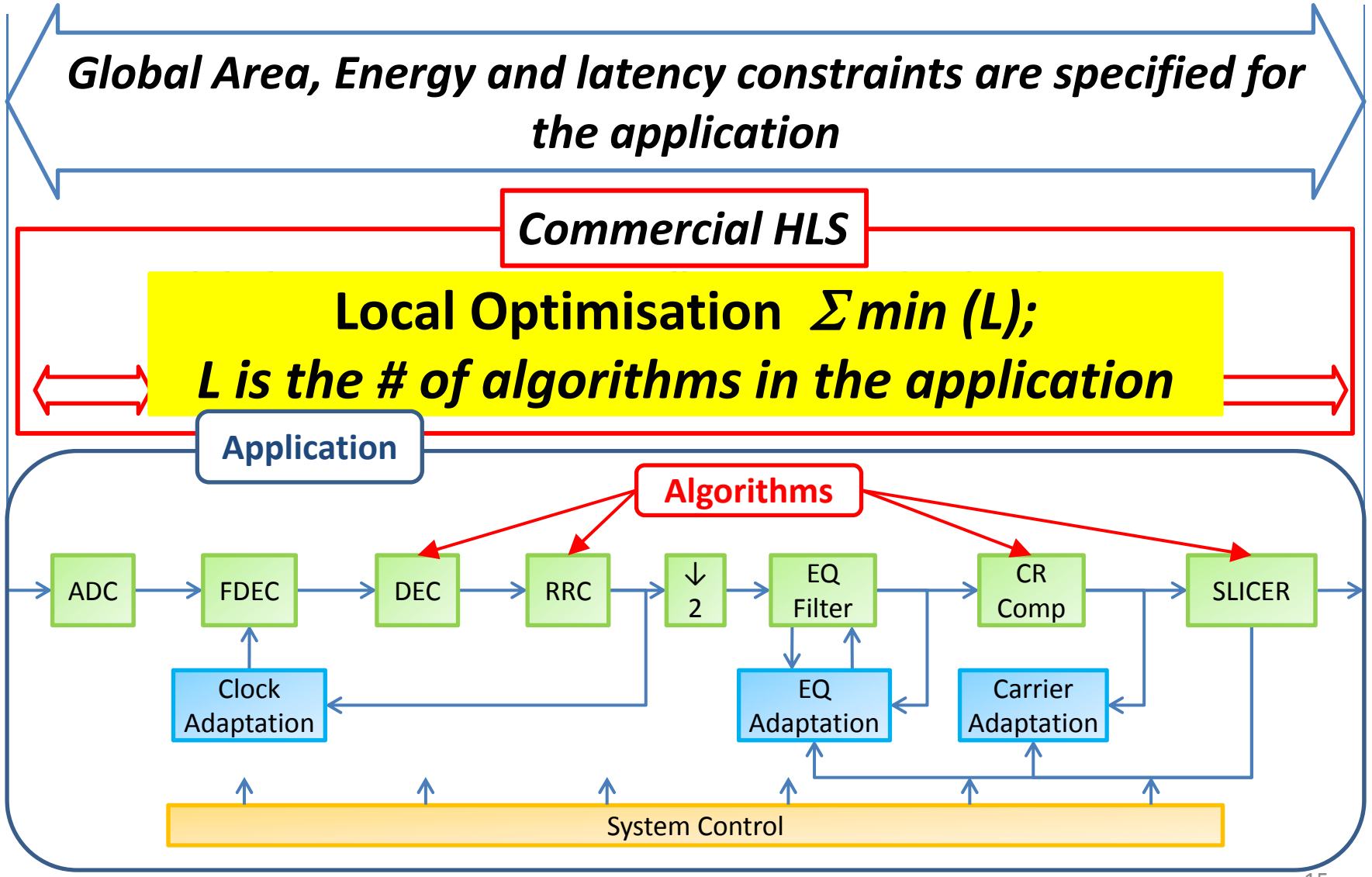
Standard Cell



An Analogy

So what happens when you try to build skyscapers with bricks

Commercial HLS achieves local optimisation

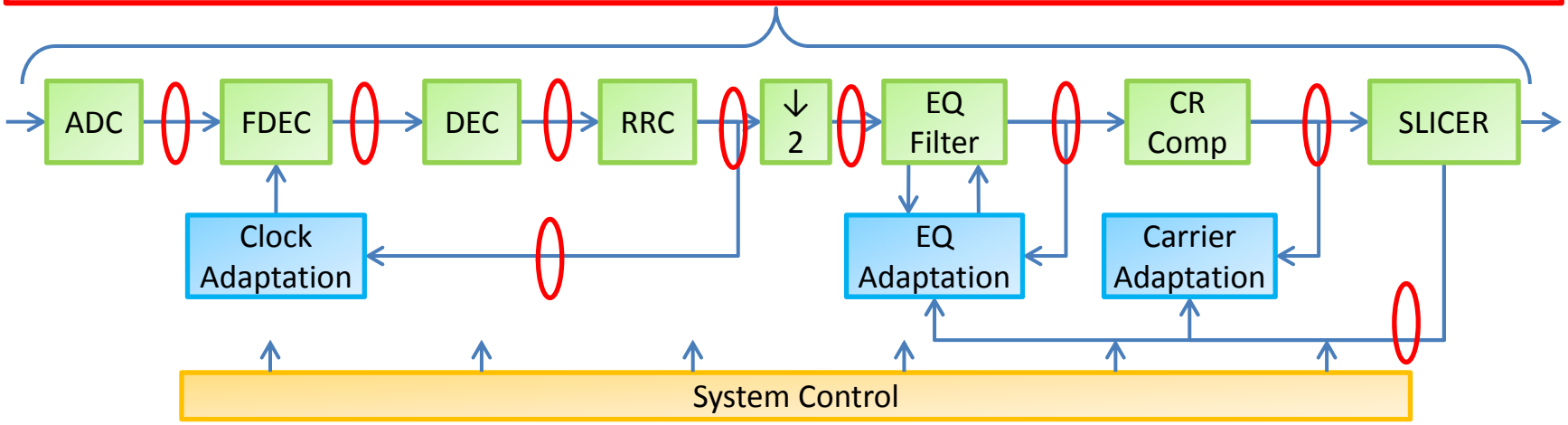


Commercial HLS: No synthesis of inter-algorithm interfaces in an Application

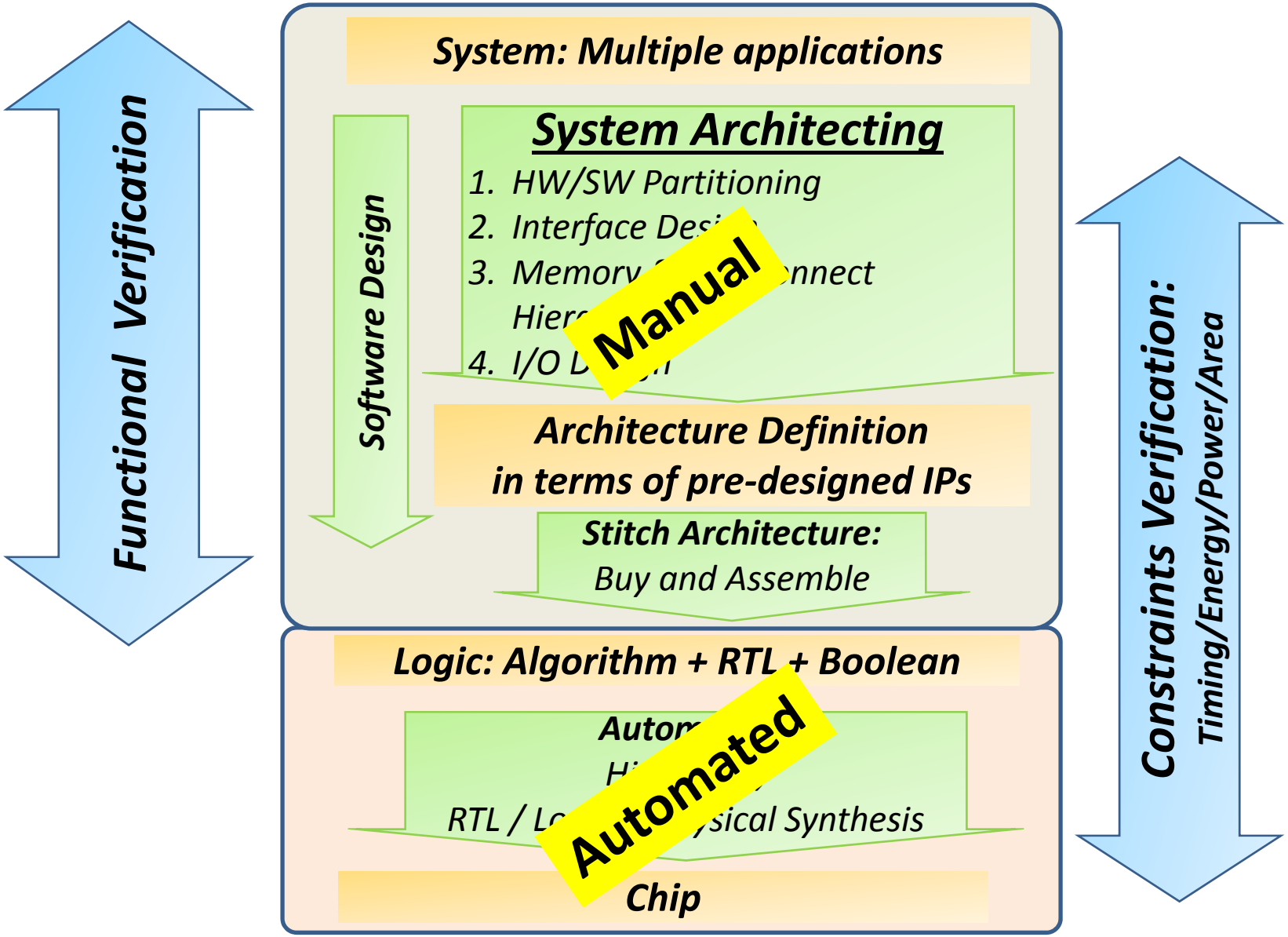
Commercial HLS

The user has to manually refine the interface between the synthesized algorithms

This manual refinement induces a functional verification step because the correct by construction contract assured by machine translation is now violated



The 45 MUSD State of the Art SOC Design Flow



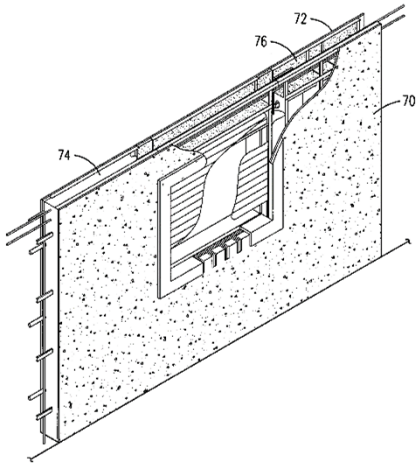
Solution:

The SiLago Method

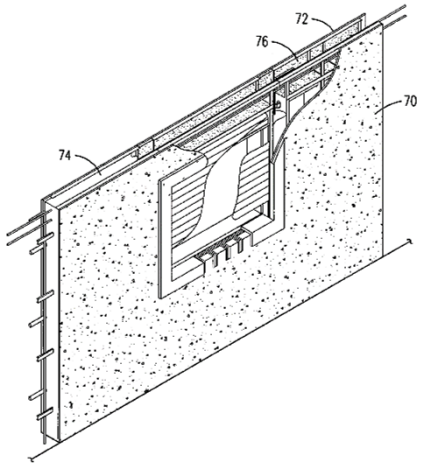
SiLago = Silicon Large Grain Object

Inspired by Lego

We shifted to pre-fabricated wall segments



The First Proposition – Raise Abstraction to μArch level



Standard Cell

Characterised boolean operations

4-5 orders larger than Sandard Cell

SiLago Blocks
Are NOT IPs – Soft or Hard

SiLago Block

(Register Files, DPUs, Switch boxes, Processors, SRAM banks etc.)

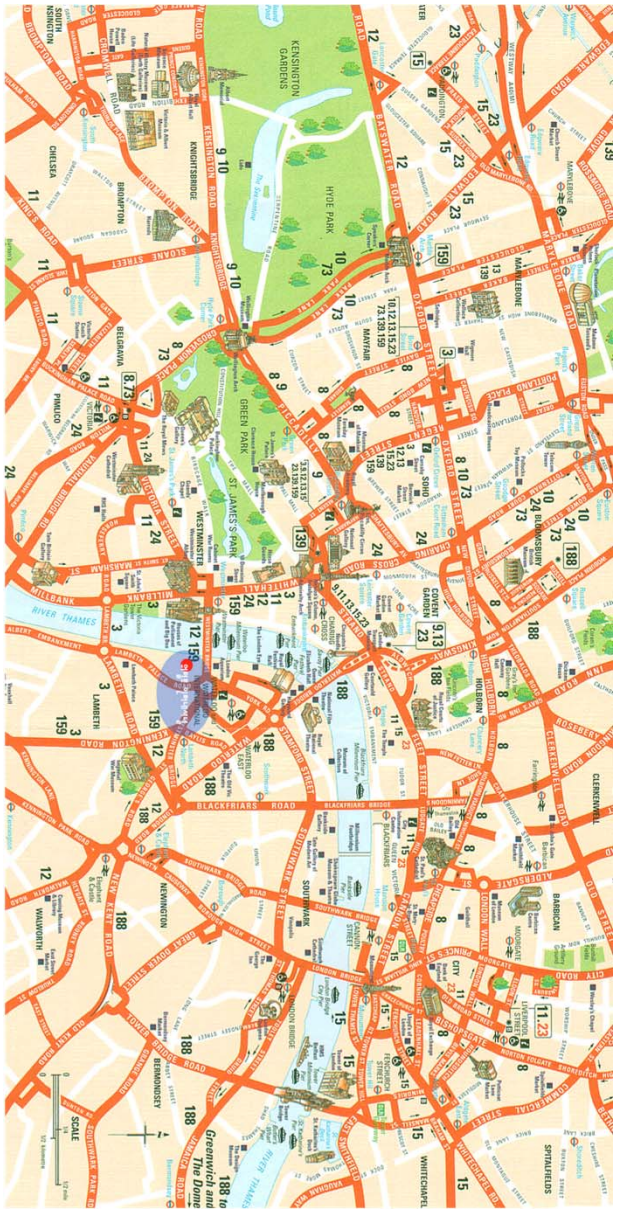
Characterised Micro-architectural operations

Solutions to VLSI Design Complexity:

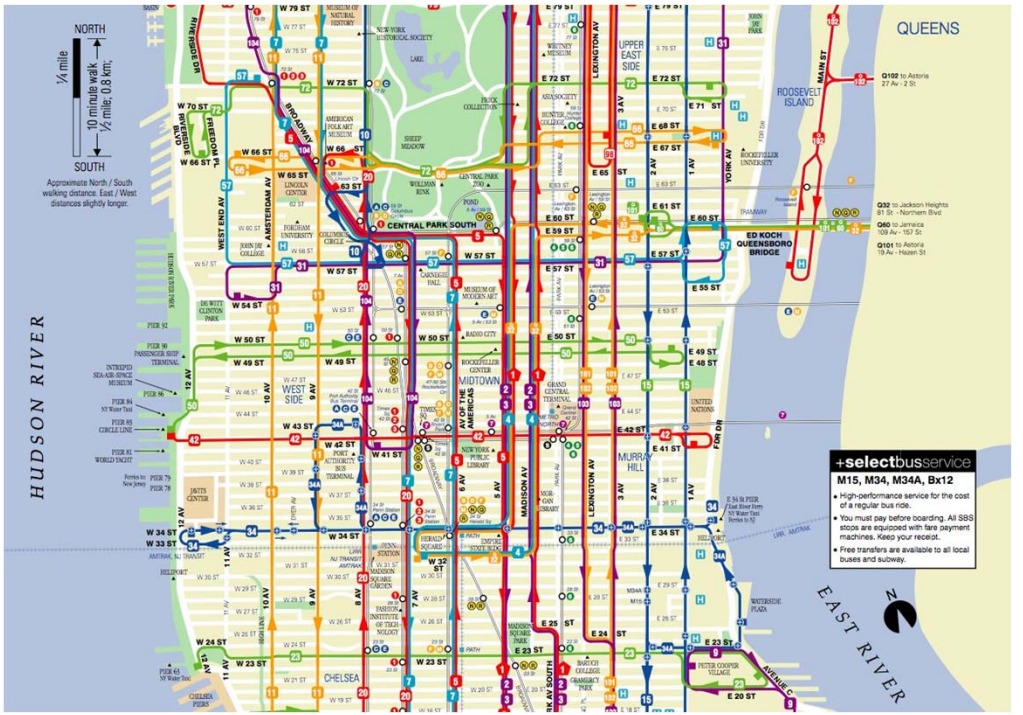
1. *Abstraction*
2. *Physical Design Discipline / Regularity*

The VLSI community has largely forgotten the second component

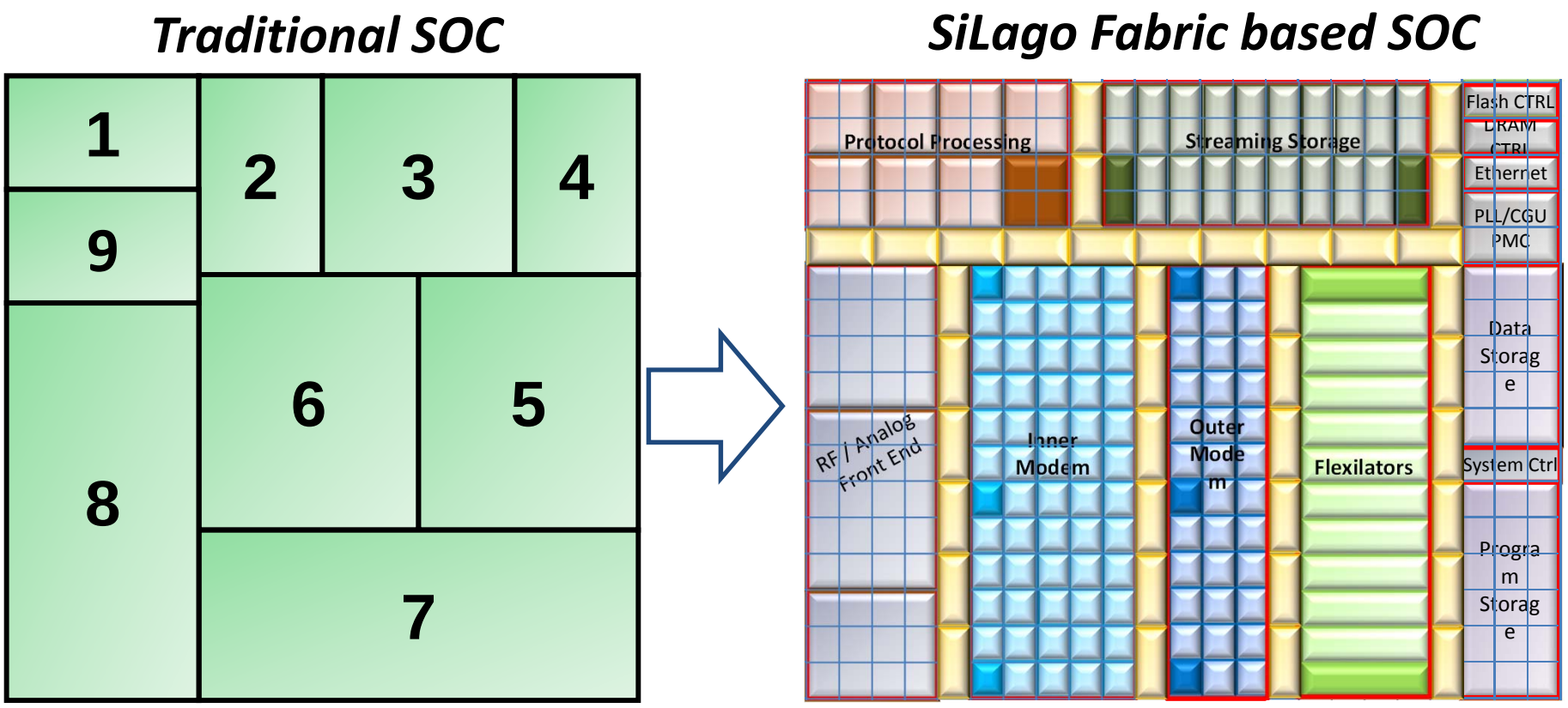
London



Manhattan



A grid based structured layout scheme

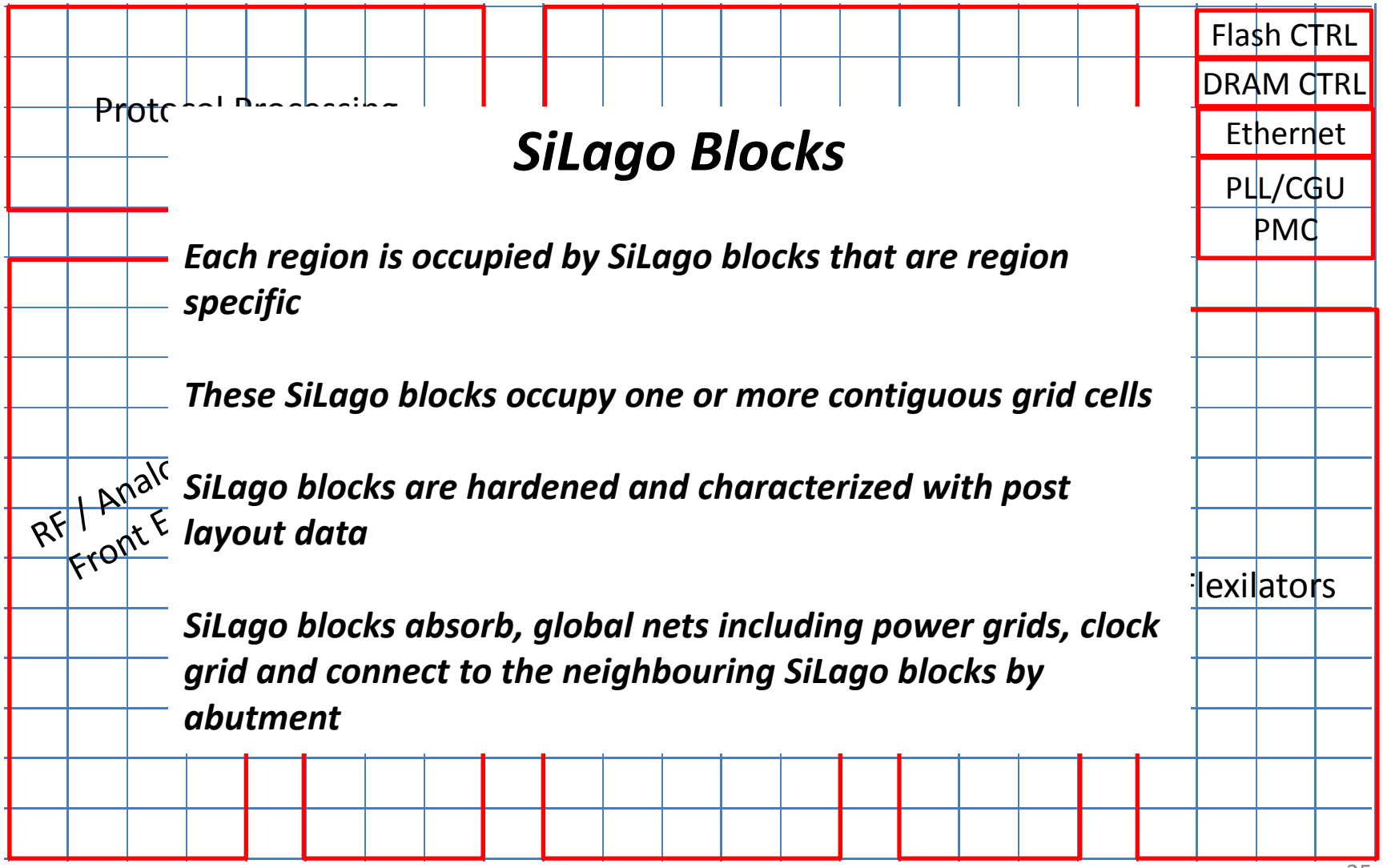


Physical Design Regularity is the sword that can slay the demons of VLSI Design Complexity

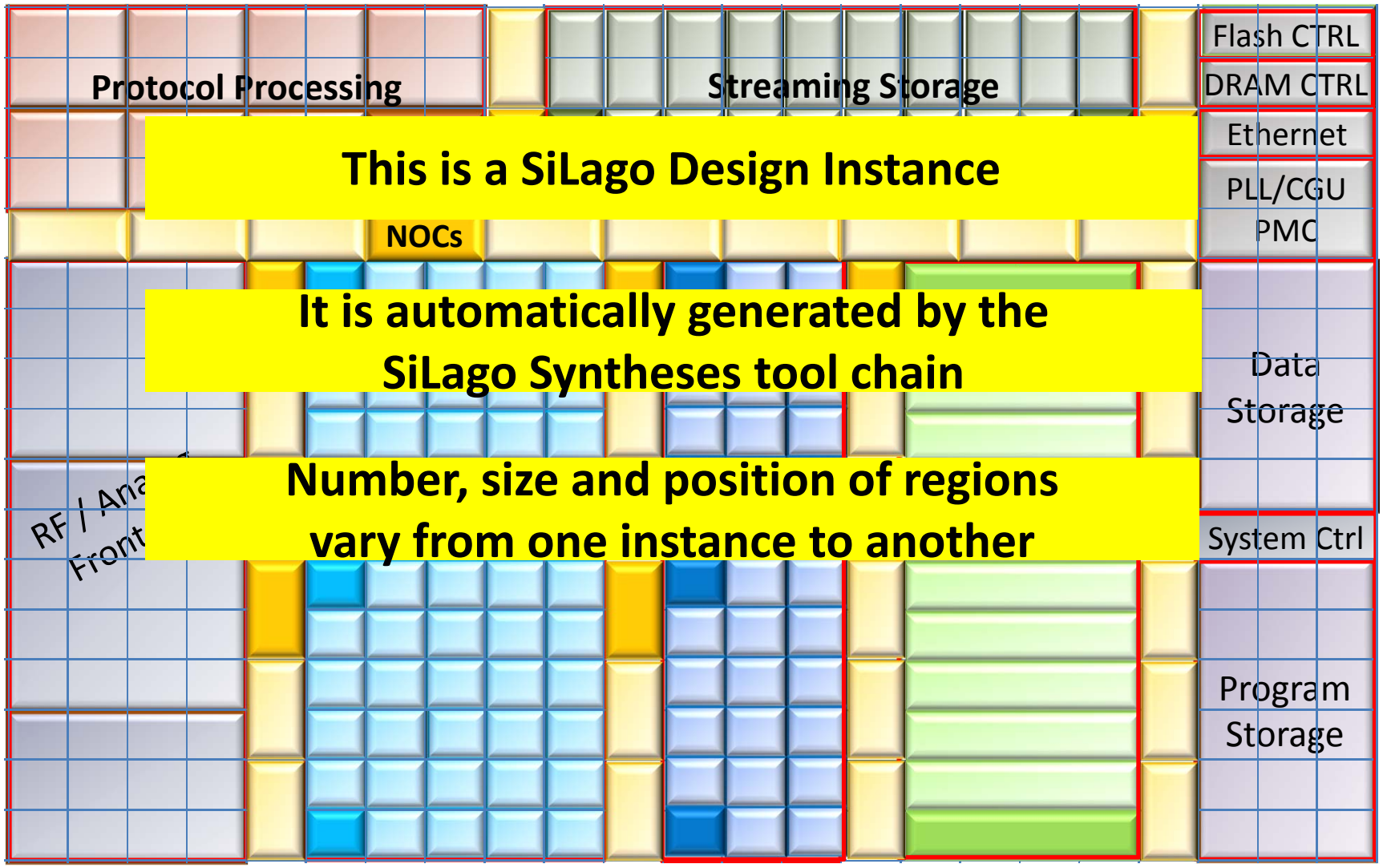
The SiLago Method

Ahmed Hemani, Nasim Farahini, Syed M.A.H. Jafri, Hassan Sohofi, Shuo Li and Kolin Paul, ”The SiLago Solution: Architecture and Design Methods for a Heterogeneous Dark Silicon Aware Coarse Grain Reconfigurable Fabric”, Chapter 3 in the book “The Dark Side of the Silicon” Springer, DOI 10.1007/978-3-319-31596-6

The SiLago Concepts



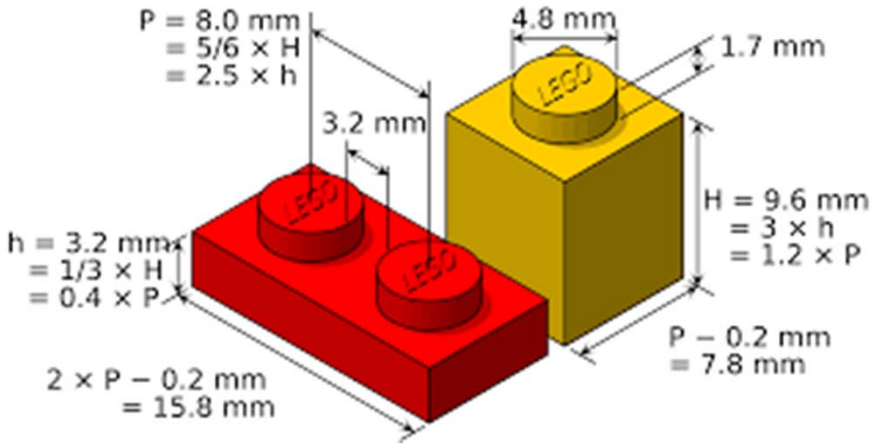
The SiLago Concepts



SiLago Interconnects are also hardened

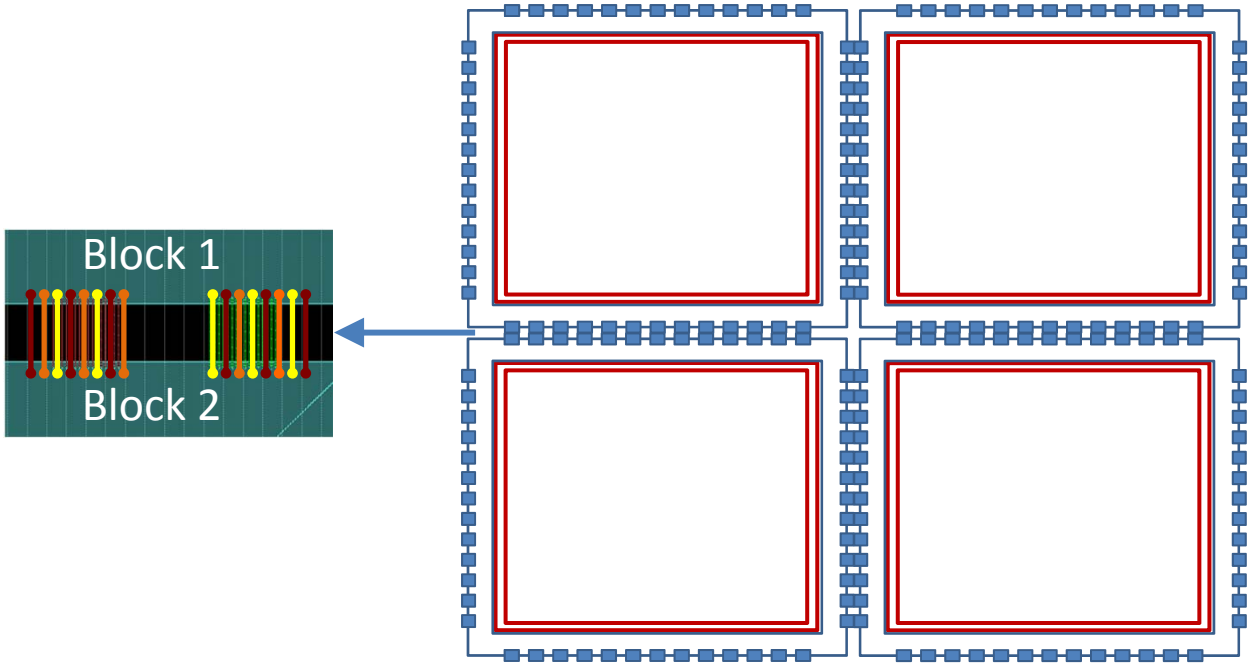
The SiLago interconnects are not just logical interconnect, i.e., soft.

They are physical and electrical objects in a templatized or parametric manner



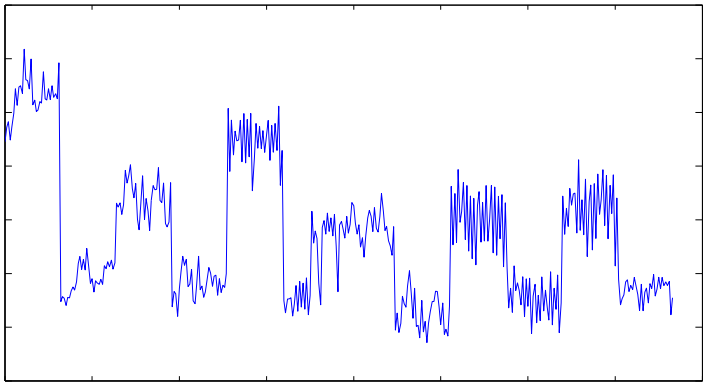
SiLago fabrics are composed by abutment

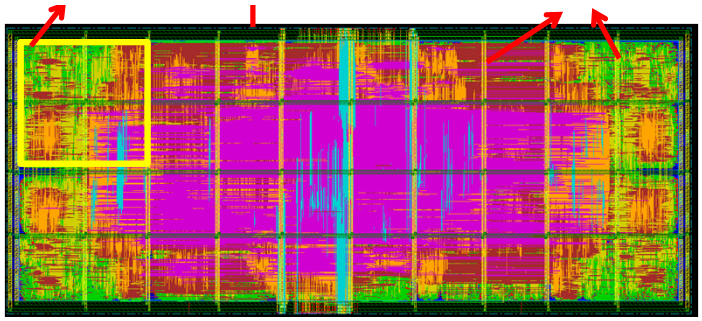
1. SiLago blocks absorbs
   a) Clock Tree & Power Ring
   b) Absorbs regional and global interconnect
   c) Pins on the periphery at right positions
2. Fabric Composition by abutment



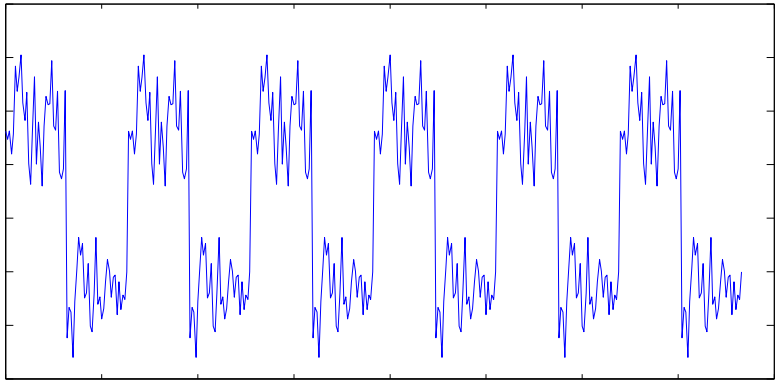
SiLago Platform Cost Metrics are Space Invariant

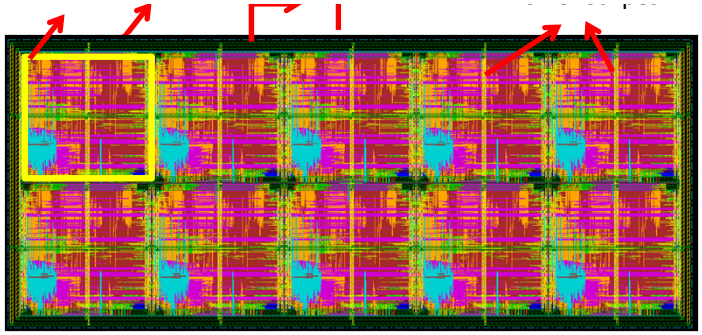
1. 16 global wires in each cell varies by about 70% from cell to cell
2. This variation is a proof that even if it is hierarchical design, the cost metrics would vary





1. The SiLago physical design discipline ensures that all wires are of exact same length





Clocking & STA

- Clock
  - Three levels of clocking: local, regional and global
  - Local

    Each SiLago block is hardened to be timing clean and synthesized ***with a certain margin for skew and latency***

    The Local Clock is synthesized using standard EDA flow
  - Regional

    Each Region is a synchronous region and the regional clock is manually synthesized to have sufficient buffers to maintain good edge and the delays balanced to keep the skew and latency within the margins of the local clock
  - Global

    Regions communicate with each other on latency insensitive basis using a previously developed GRLS scheme.

    For more details see

    http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6507330&tag=1
- STA – Static Timing Analysis
  - ILMs are created for each SiLago blocks
  - Once the regional clocks are synthesized and inserted back into the data base, a hierarchcial STA script is run to ensure that the entire design is timing clean.

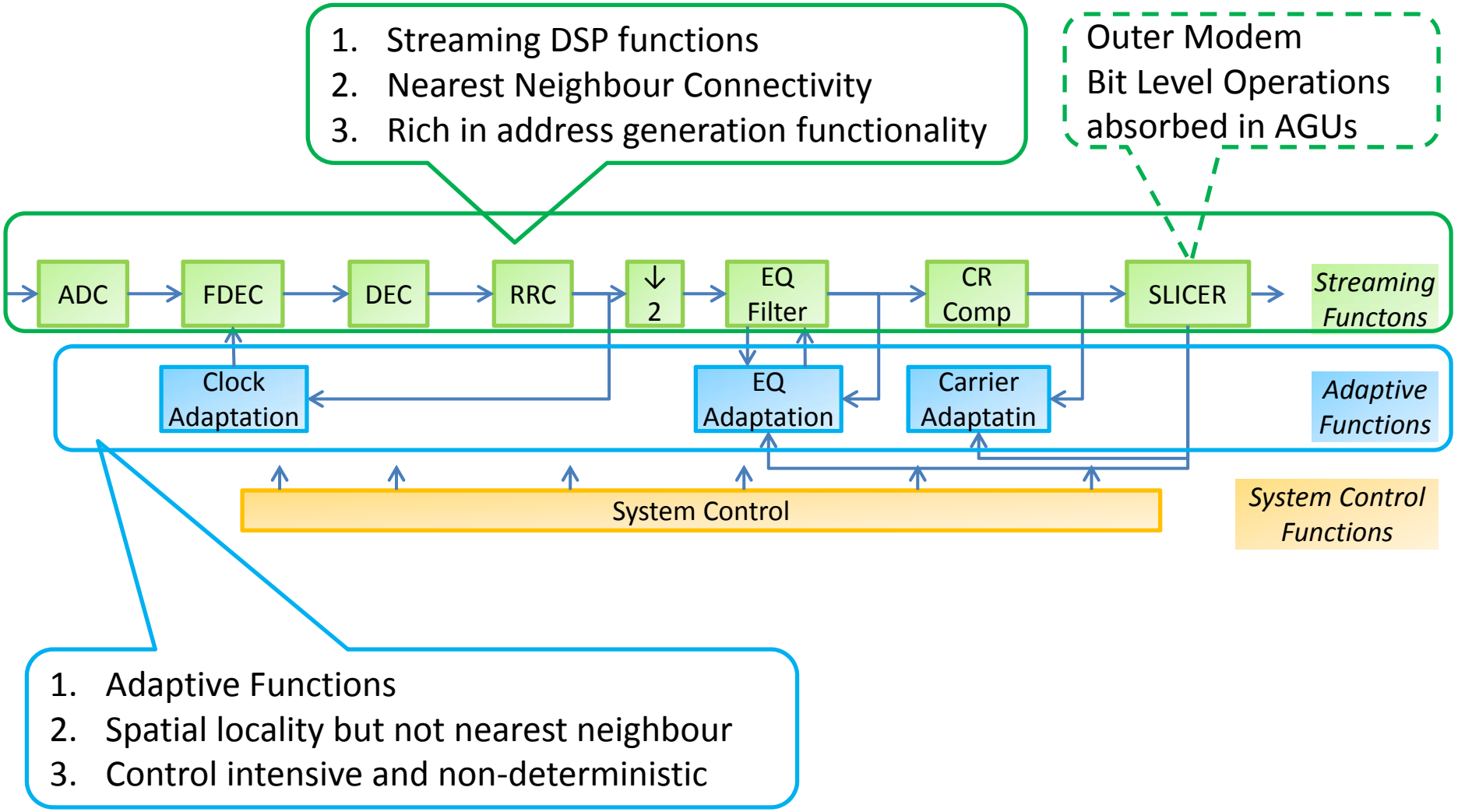
Characterization

1. Each SiLago block is hardened
2. Sufficiently exhaustive simulation is performed for molecules of SiLago blocks at gate level with post layout data back annotated
   - The SiLago blocks cannot be too large and complex
   - The same pipeline cannot be used for multiple operations
3. Concurrent operations within and neighbouring SiLago blocks weakly couple and we model this coupling
4. The NOCs are parameterically hardened

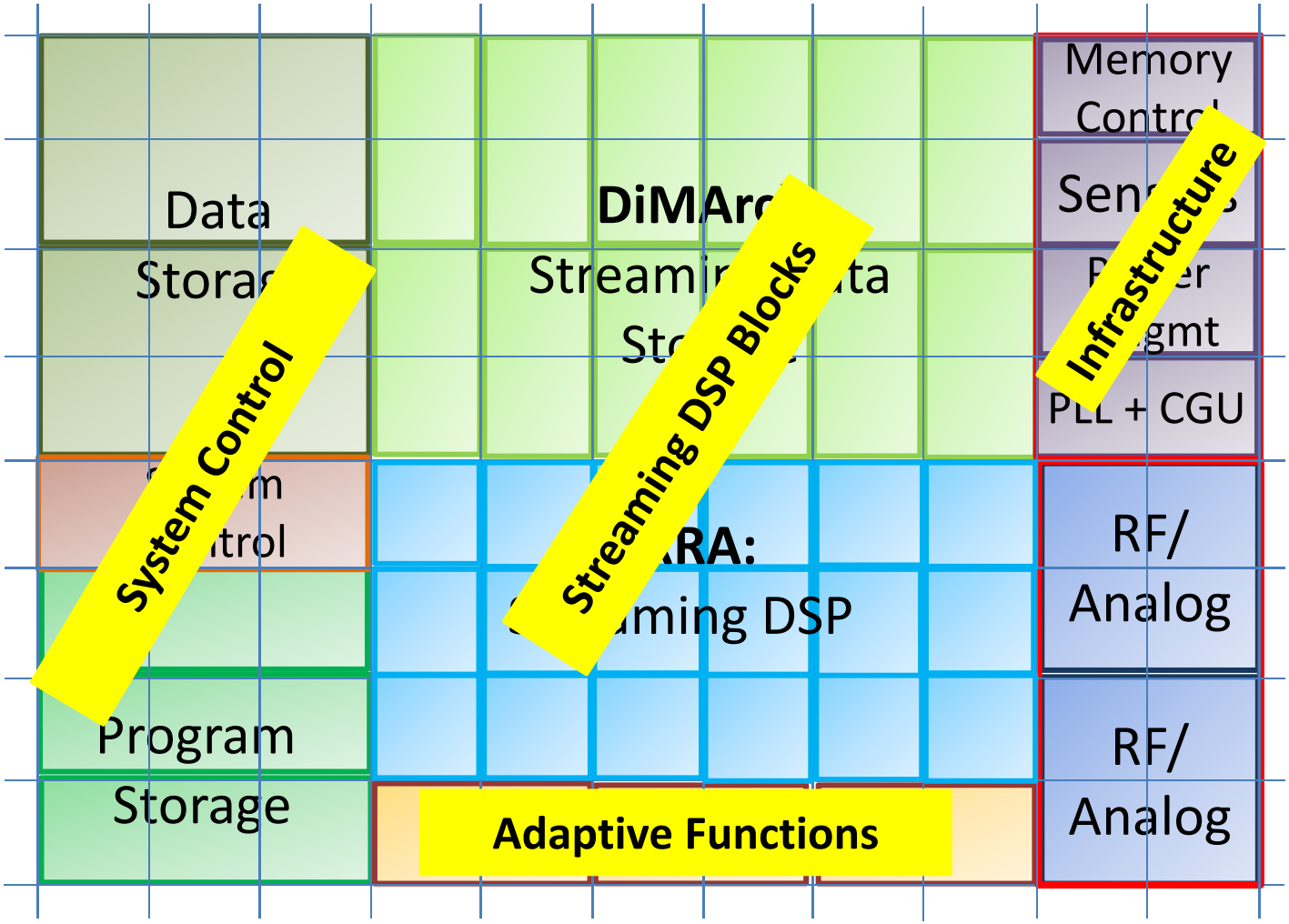
The SiLago Proof of Concept

How are the μ-architectural design decisions made ?

Target Application Domain: Modems & Codecs



Proof of Concept SiLago Platform

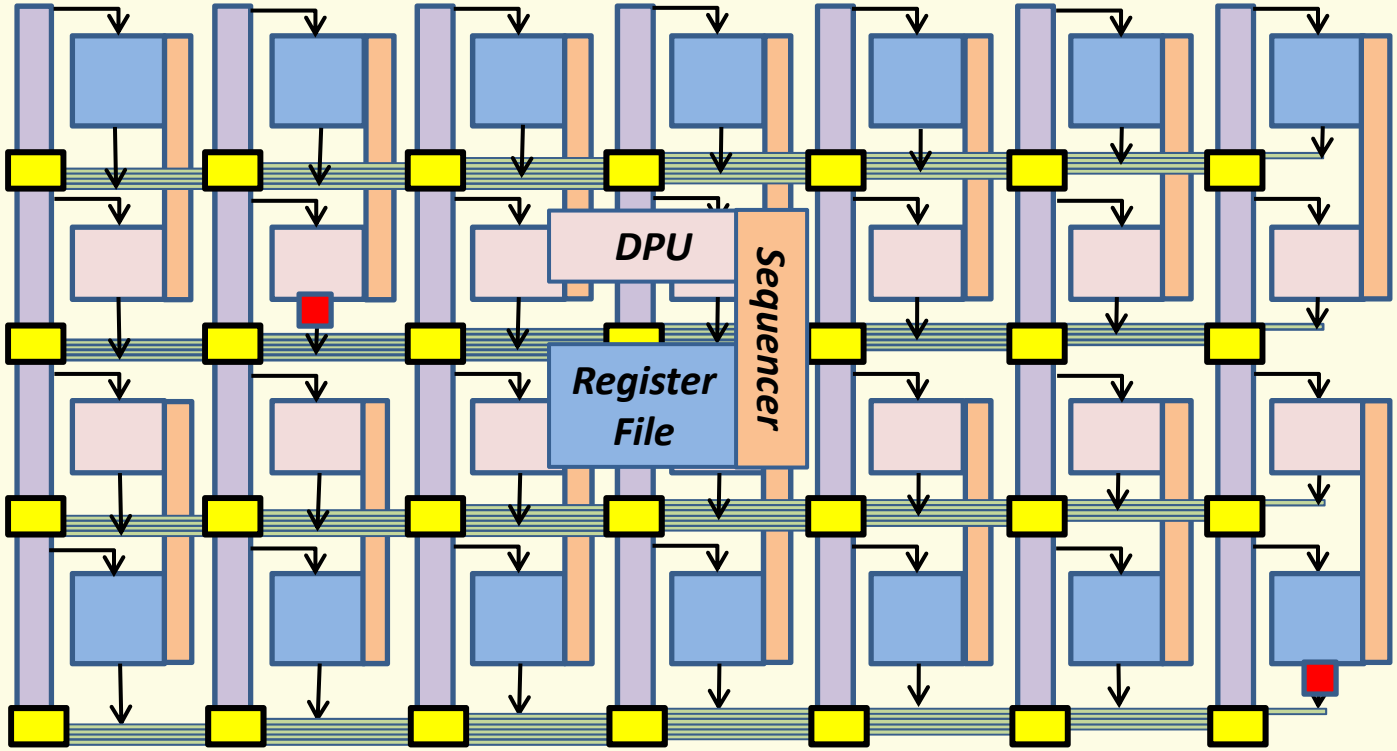


DRRA – Computational Fabric

Dynamically Reconfigurable Resource Array

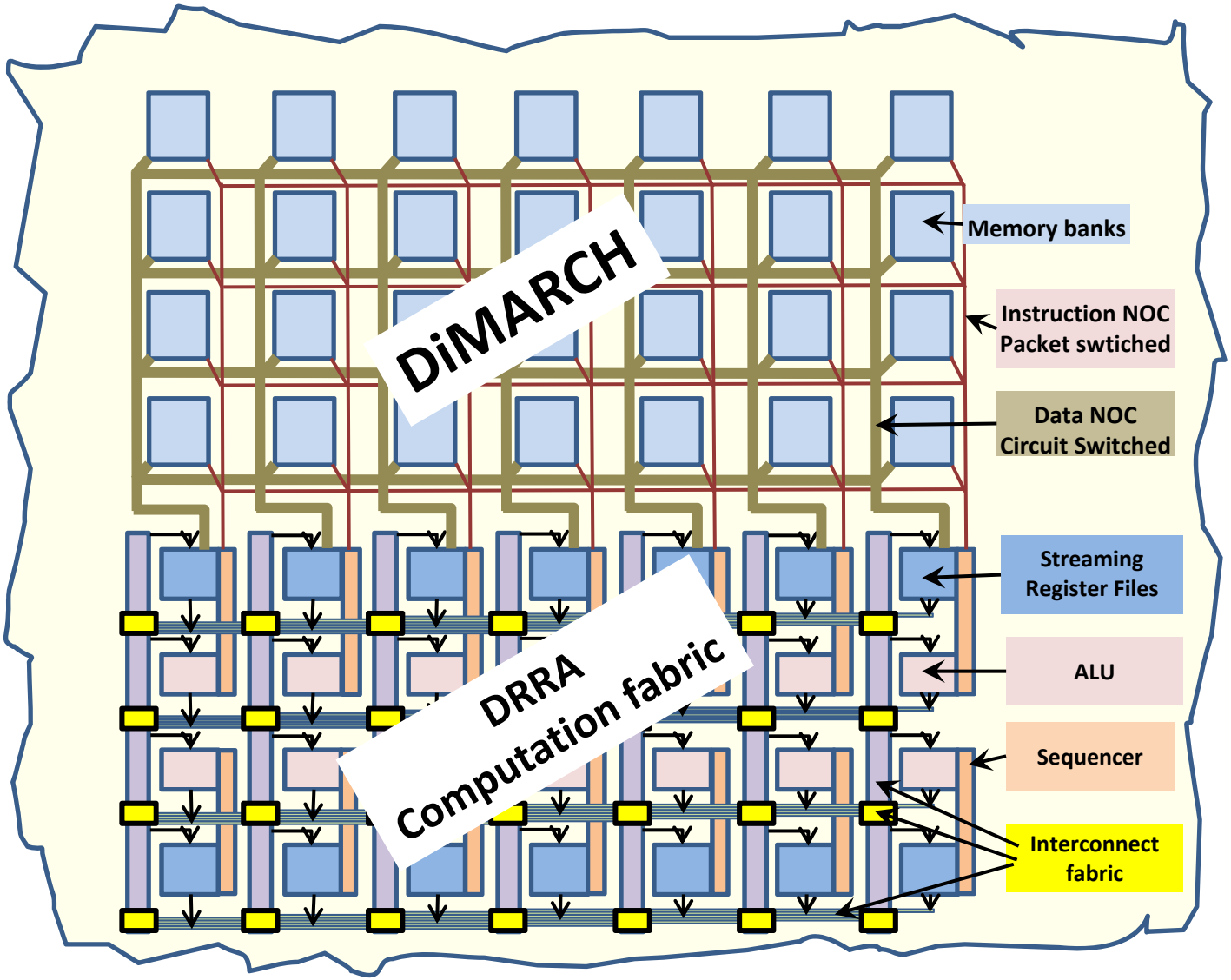
This is only a fragment

22 nm, 100 mm²
10 000 DRRA Cells

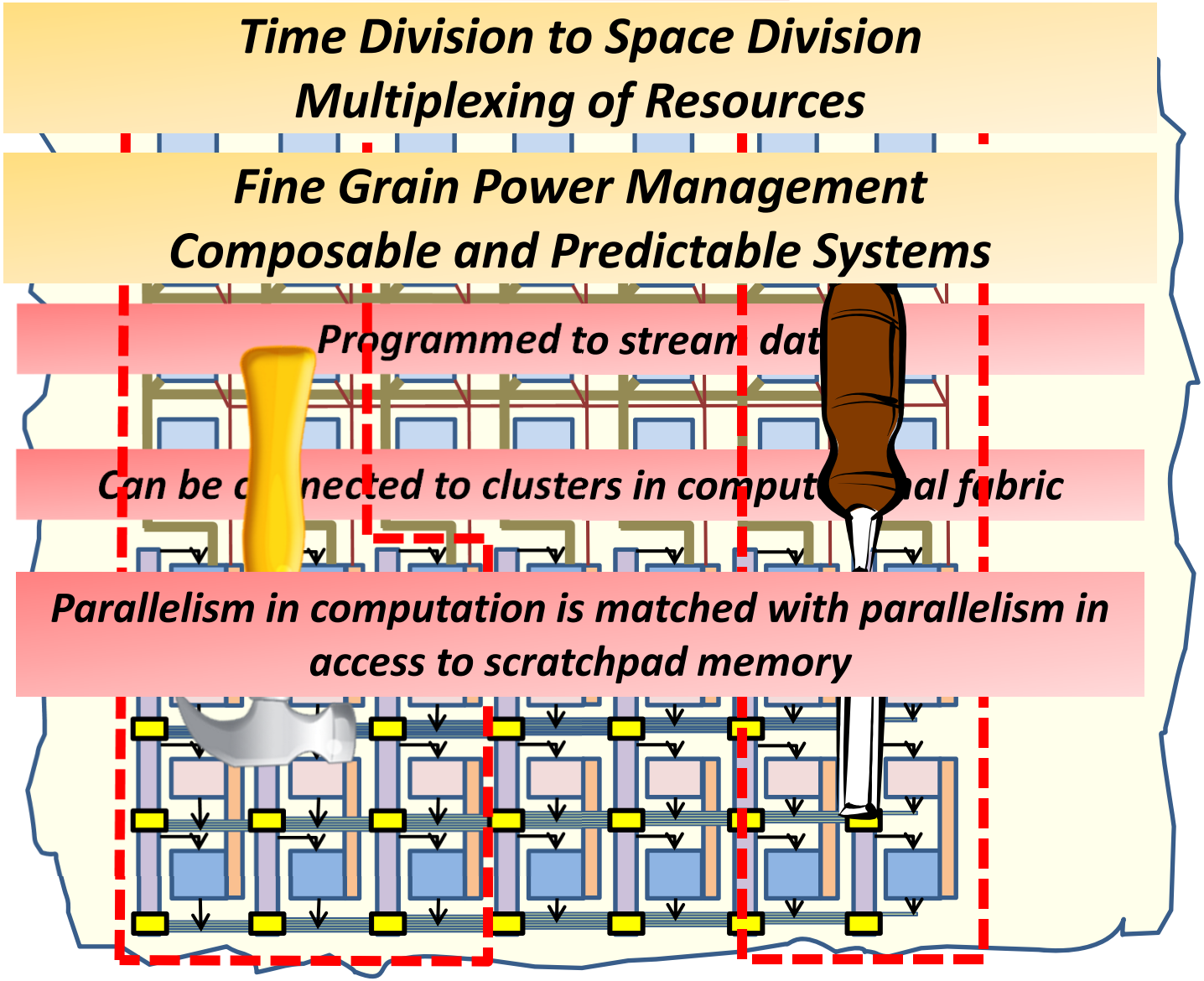


DPU & Register File Outputs
3 Columns to the Left and and to the Right
And this 3 column window slides

Distributed Memory Fabric – DiMARCH



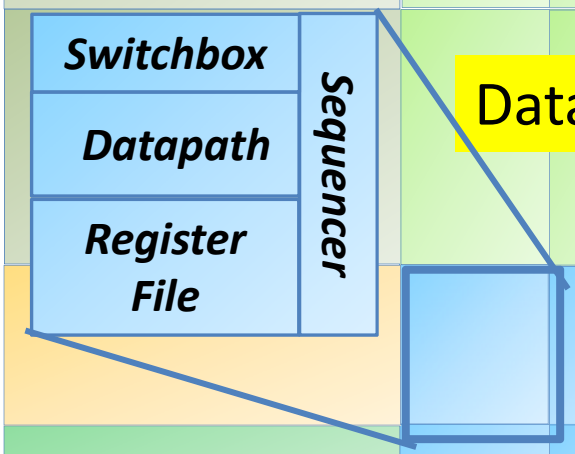
Private Execution Partitions



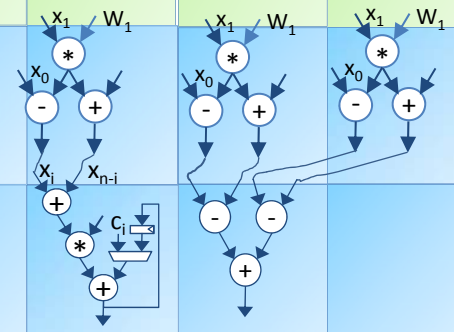
1. M.A. Shami, A. Hemani, Address generation scheme for a coarse grain reconfigurable architecture, in 2011 IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP) (2011), pp. 17–24
2. N. Farahini, A. Hemani, K. Paul, Distributed runtime computation of constraints for multiple inner loops, in 2013 Euromicro Conference on Digital System Design (DSD) (2013)
3. M.A. Shami, A. Hemani, Classification of massively parallel computer architectures, in 2012IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW) (2012), pp. 344–351
4. N. Farahini, A. Hemani, Atomic stream computation unit based on micro-thread level parallelism, in 2015 IEEE 26th International Conference on Application-specific Systems, Architectures and Processors (ASAP) (2015), pp. 25–29
5. N. Farahini, A. Hemani, H. Sohofi, S.M.A.H. Jafri, M.A. Tajammul, K. Paul, Parallel distributed scalable runtime address generation scheme for a coarse grain reconfigurable computation and storage fabric. Microprocess. Microsyst. 38, 788–802 (2014)
6. M.A. Shami, A. Hemani, Morphable DPU: smart and efficient data path for signal processing applications, in IEEE Workshop on Signal Processing Systems, 2009 (SiPS 2009) (2009), pp. 167–172
7. M.A. Shami, A. Hemani, Control scheme for a CGRA, in 2010 22nd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD) (2010), pp. 17–24
8. M.A. Shami, A. Hemani, An improved self-reconfigurable interconnection scheme for a coarse grain reconfigurable architecture, in NORCHIP, 2010 (2010), pp. 1–6
9. M. A. Shami, A. Hemani, Partially reconfigurable interconnection network for dynamically reprogrammable resource array, in IEEE 8th International Conference on ASIC, 2009. ASICON'09 (2009), pp. 122–125
10. M.A. Tajammul, M.A. Shami, A. Hemani, S. Moorthi, NoC based distributed partitionable memory system for a coarse grain reconfigurable architecture, in 2011 24th International Conference on VLSI Design (VLSI Design) (2011), pp. 232–237

Clustering SiLago blocks is clustering Standard Cells or LUTs in FPGAs

Variations in Function, Capacity and Parallelism created by clustering micro-architectural operations in SiLago blocks



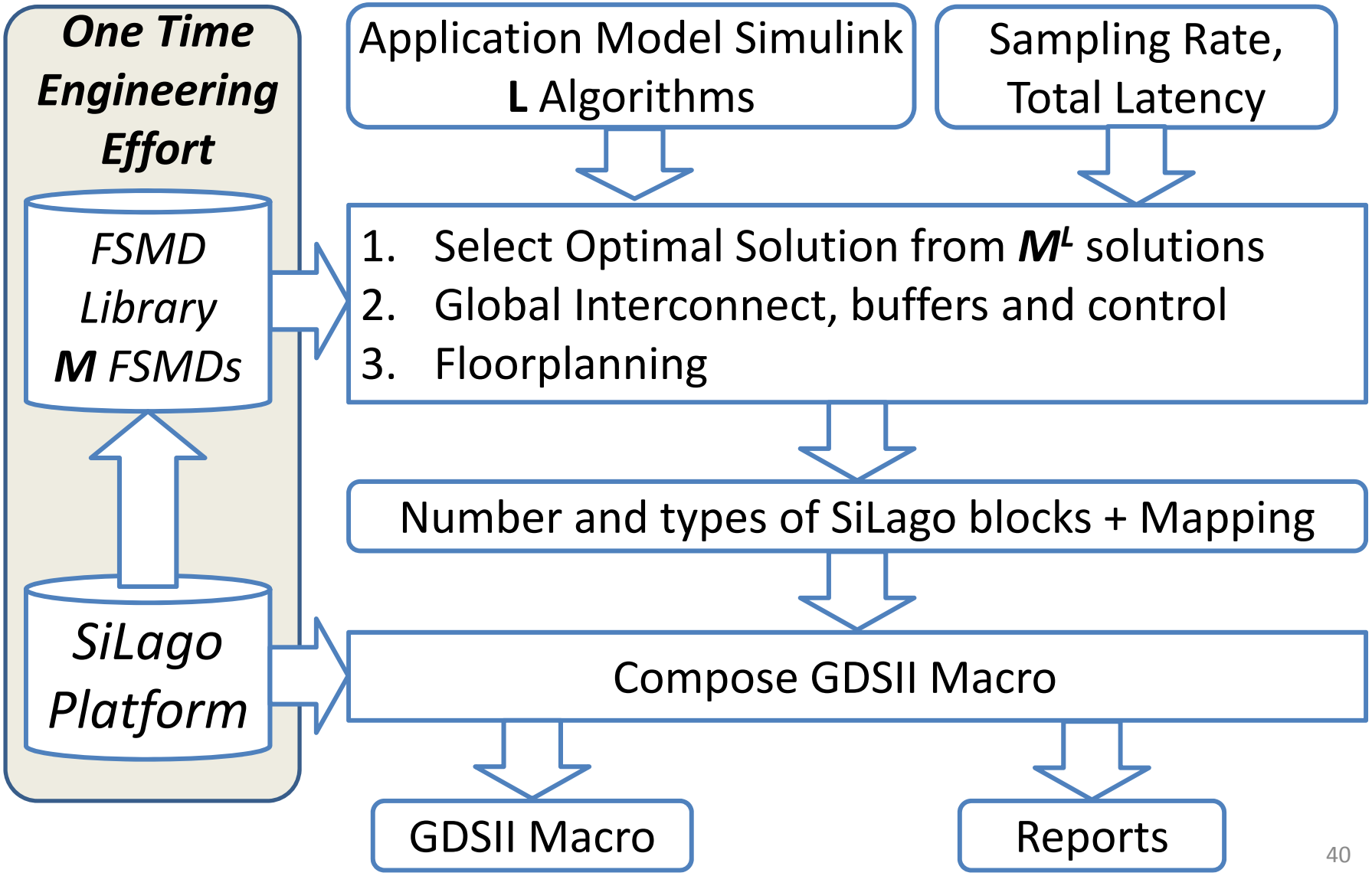
Datapath can be clustered to create arbitrary DFG



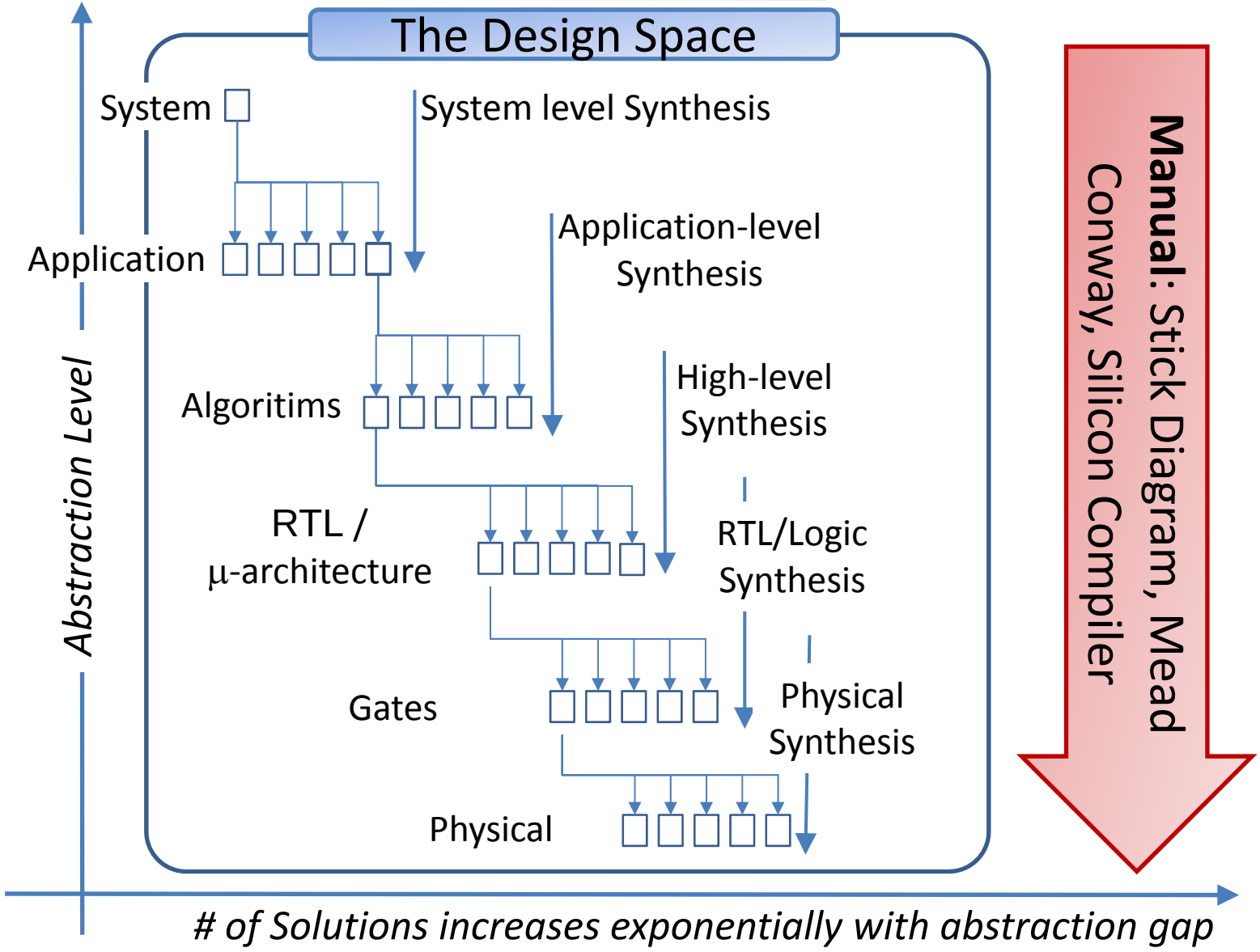
Register files/SRAMs can be also clustered to create larger and/or more parallel data storage

Sequencers can be organized to create hierarchical FSMs

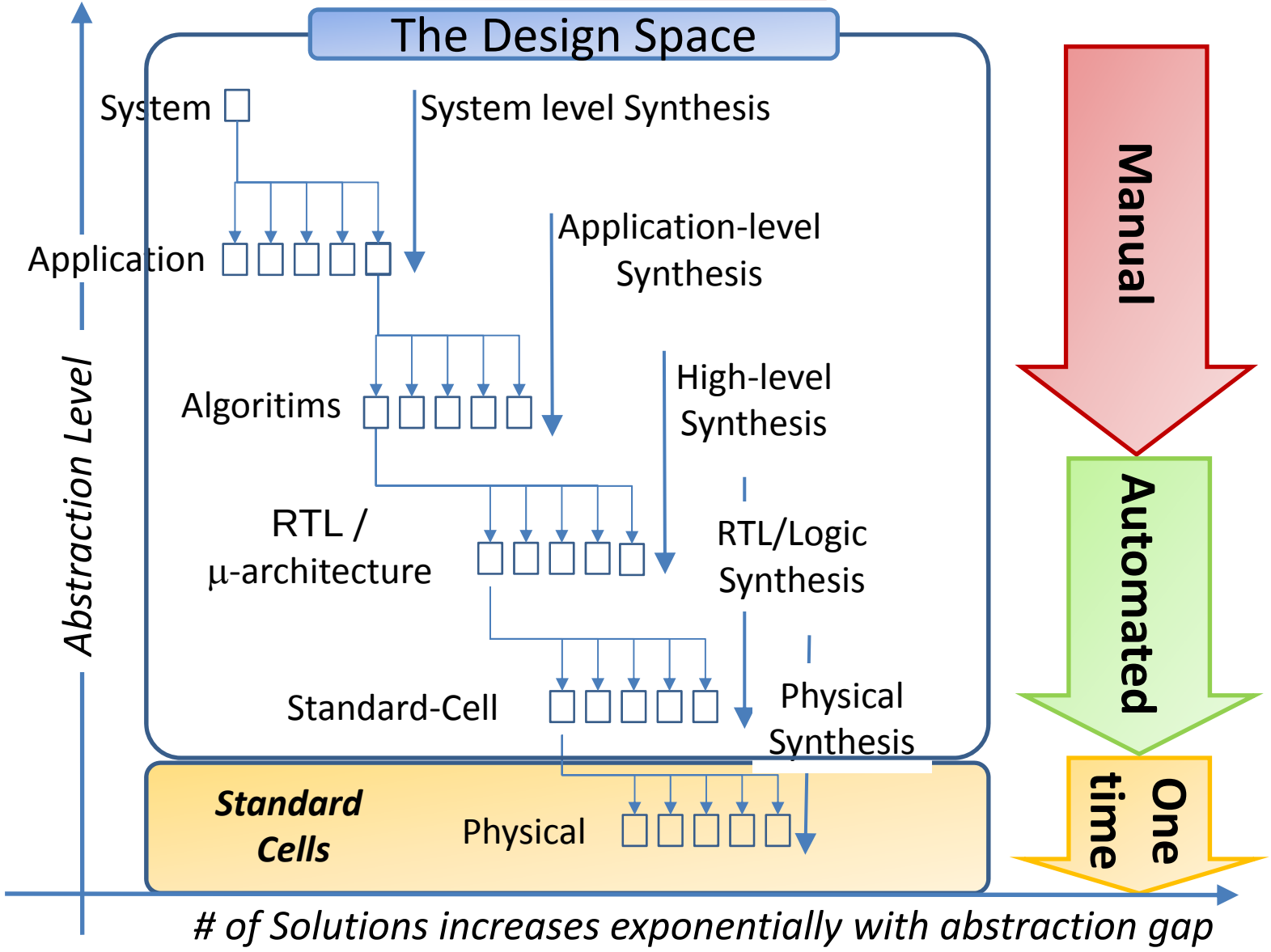
SiLago Design Flow



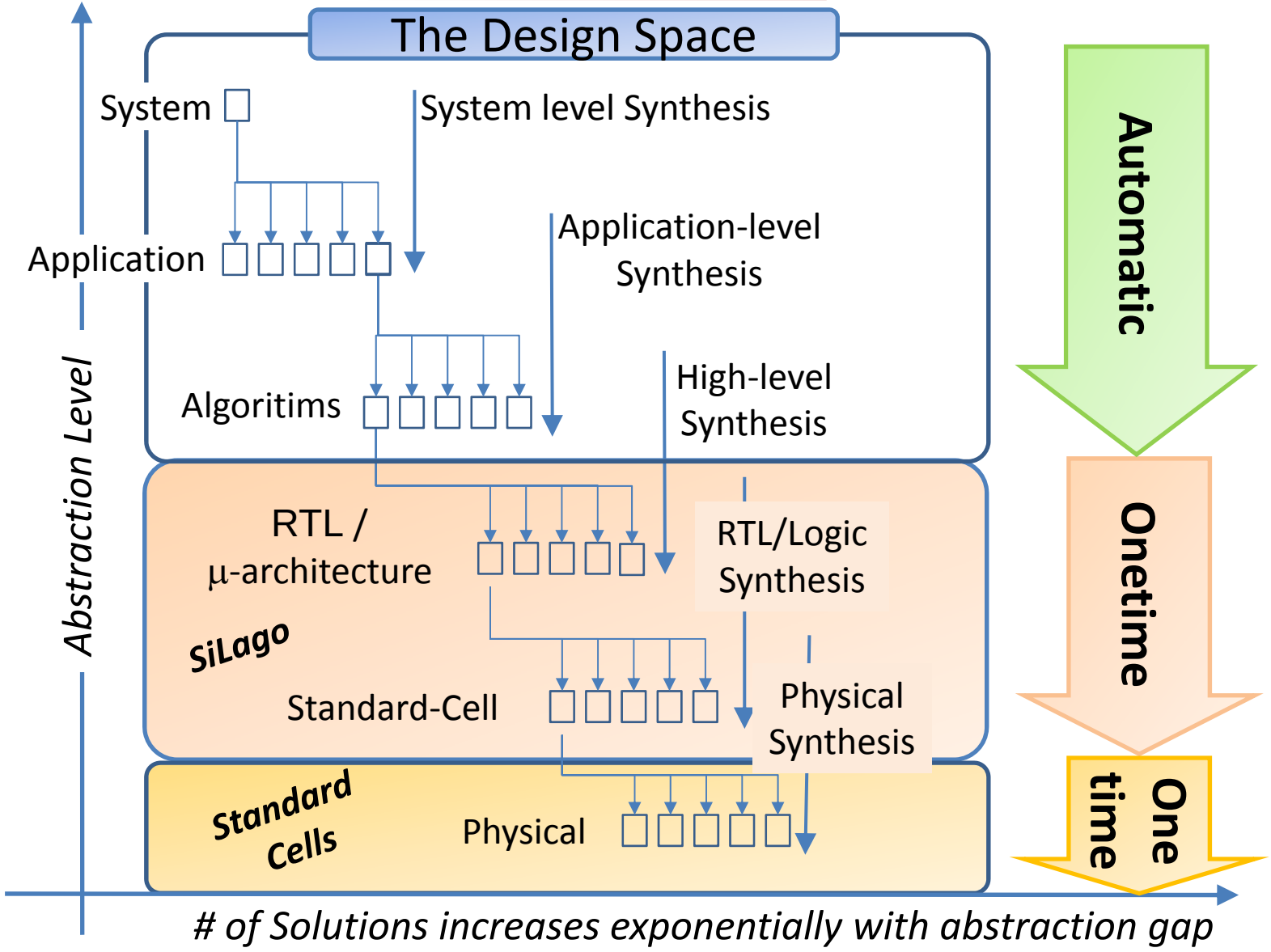
The Mead Conway Era



The Standard Cell Era



The SiLago Era



SiLago achieves Global Optimisation

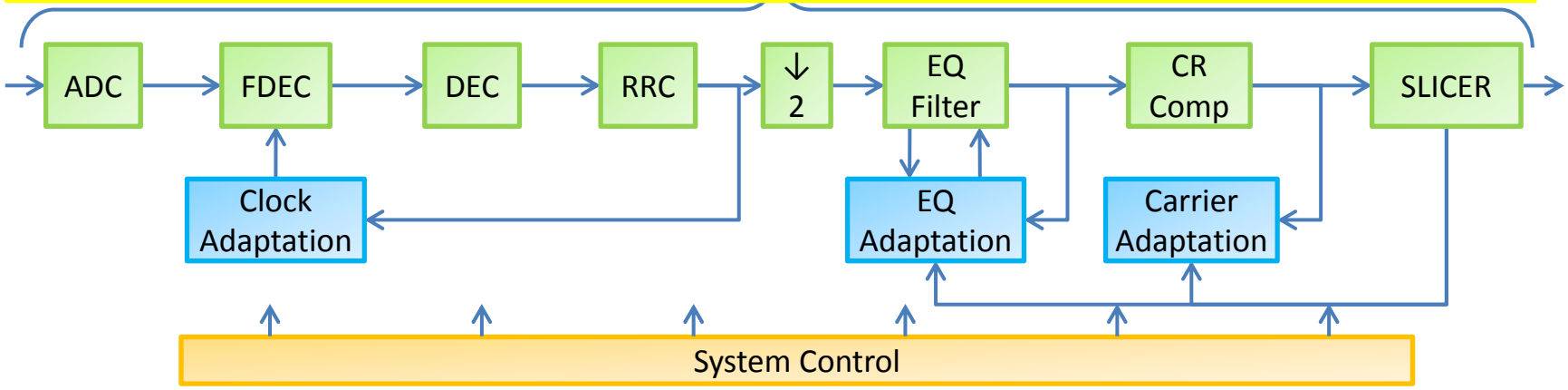
Global Area, Energy and latency constraints are specified for the application

L: Algorithms in Application

M: Number of ways of implementing each algorithm

SiLago: Global Optimisation - *Min (M^L)*

Commercial HLS : Local Optimization - *Σ min(L)*

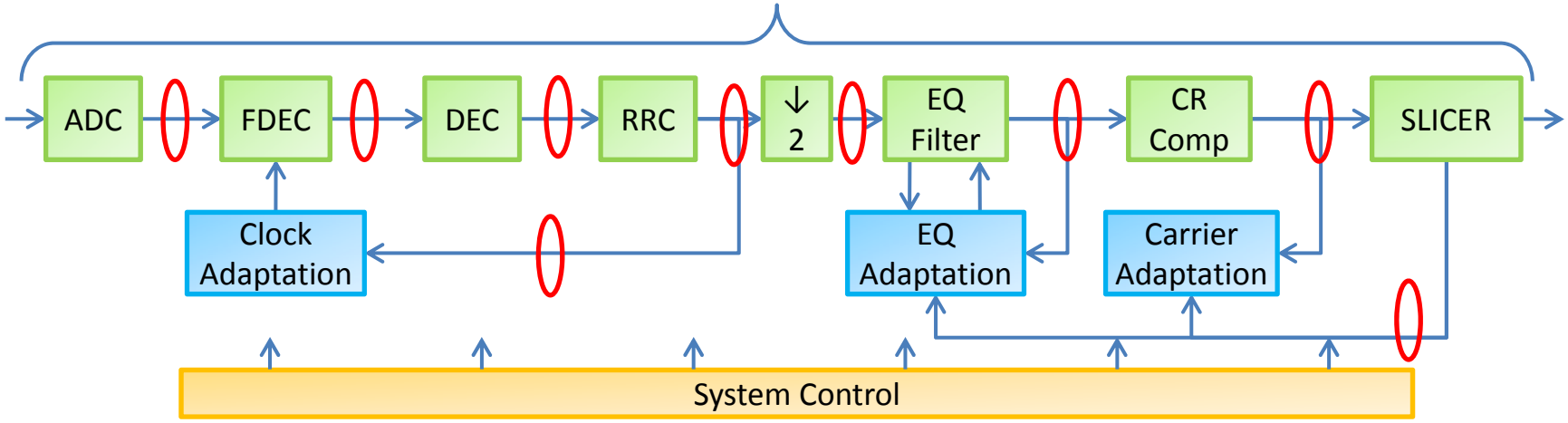


SiLago also automates Interface Synthesis

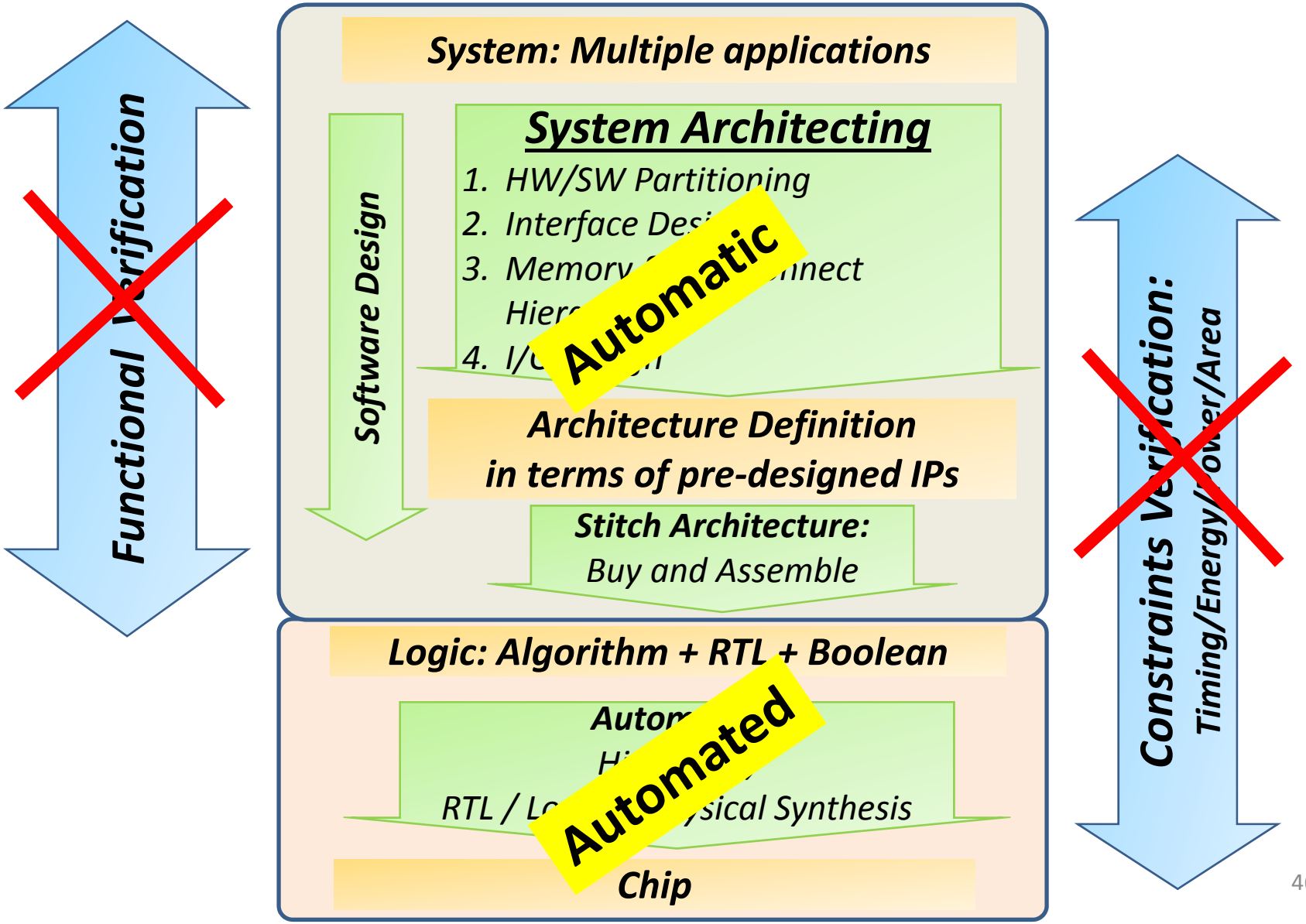
SiLago Application Level Synthesis

The interfaces are automatically synthesized depending on the chosen degree of parallelism of algorithms.

Machine translation ensures correct by construction guarantee

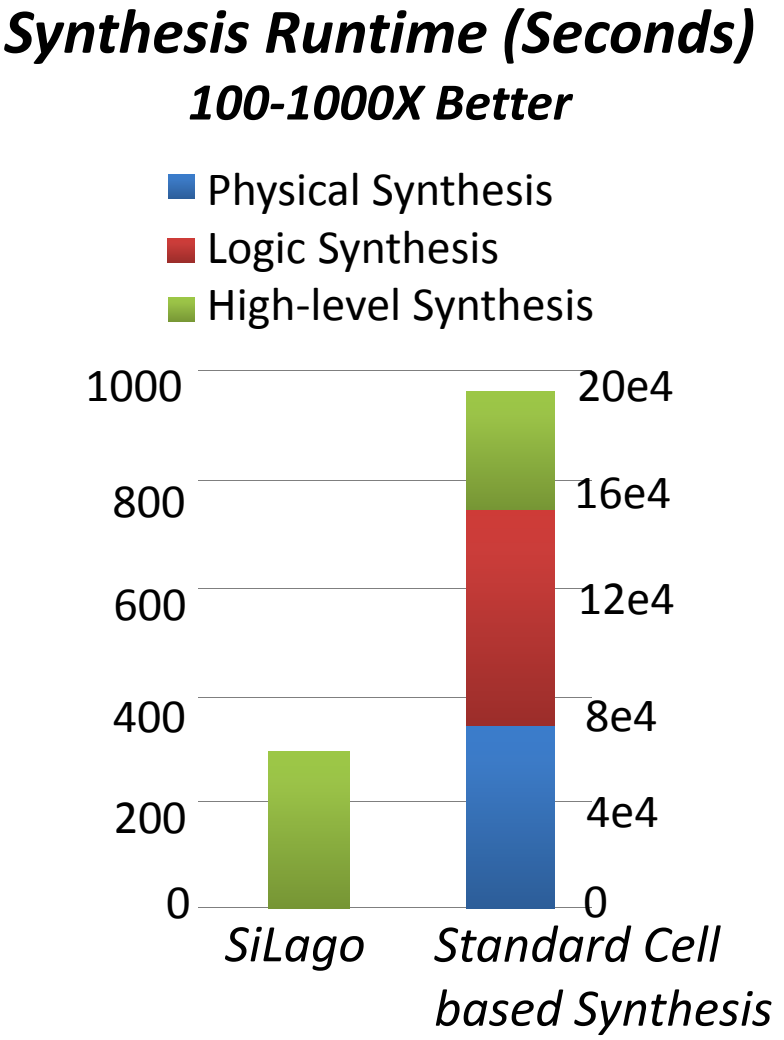


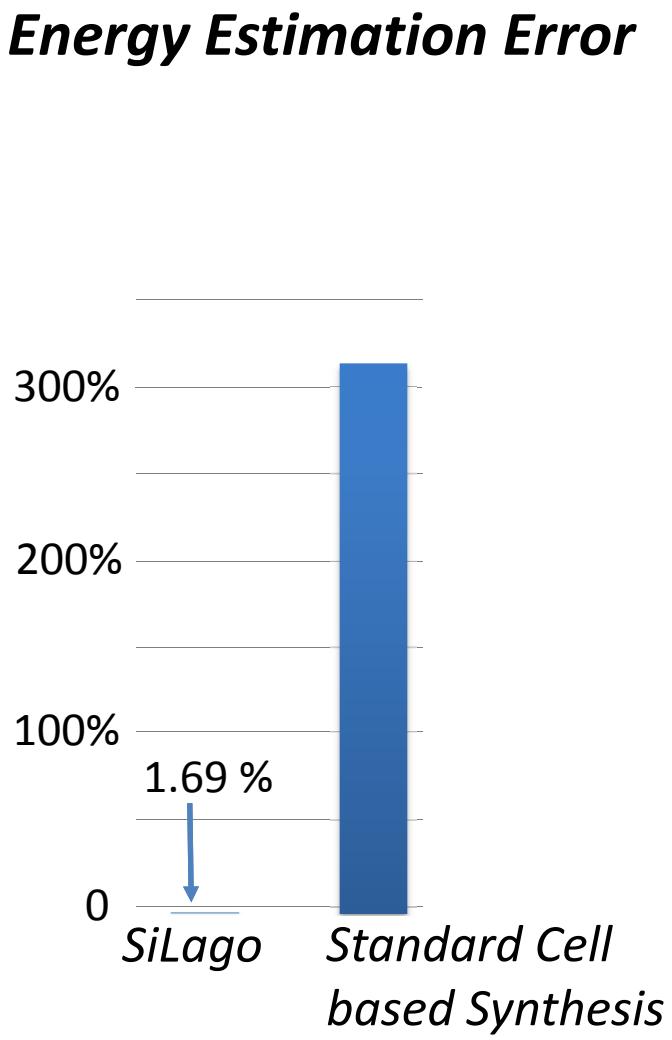
What the SiLago Method promises to achieve ?



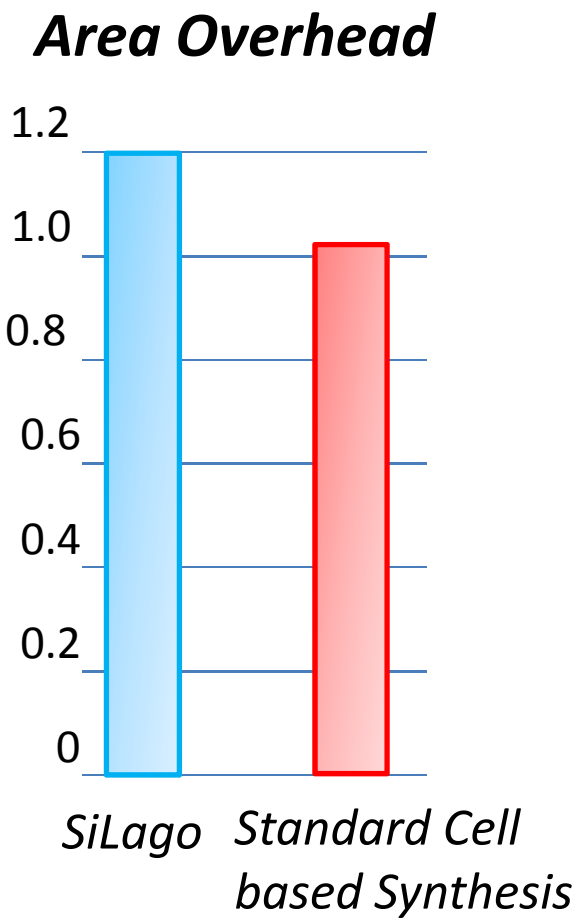
Experimental Proof that the proposed Solution Works

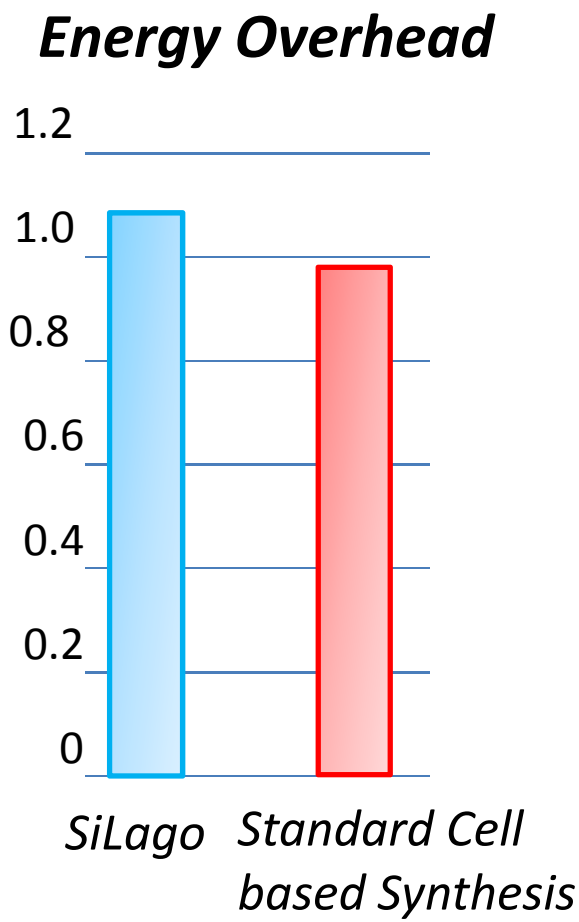
SiLago FSMD Library Development Efficiency



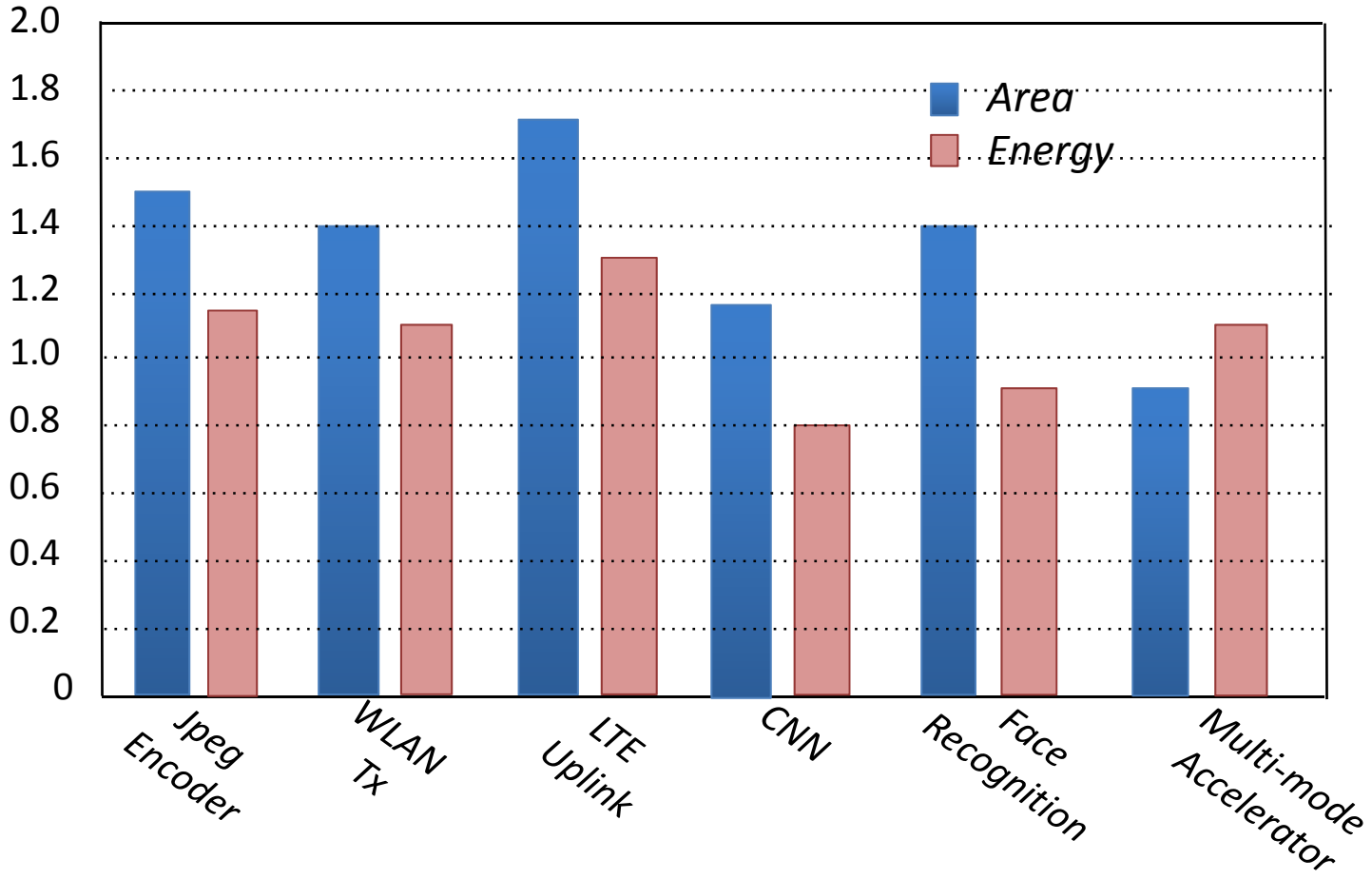


And what do we pay for it ?

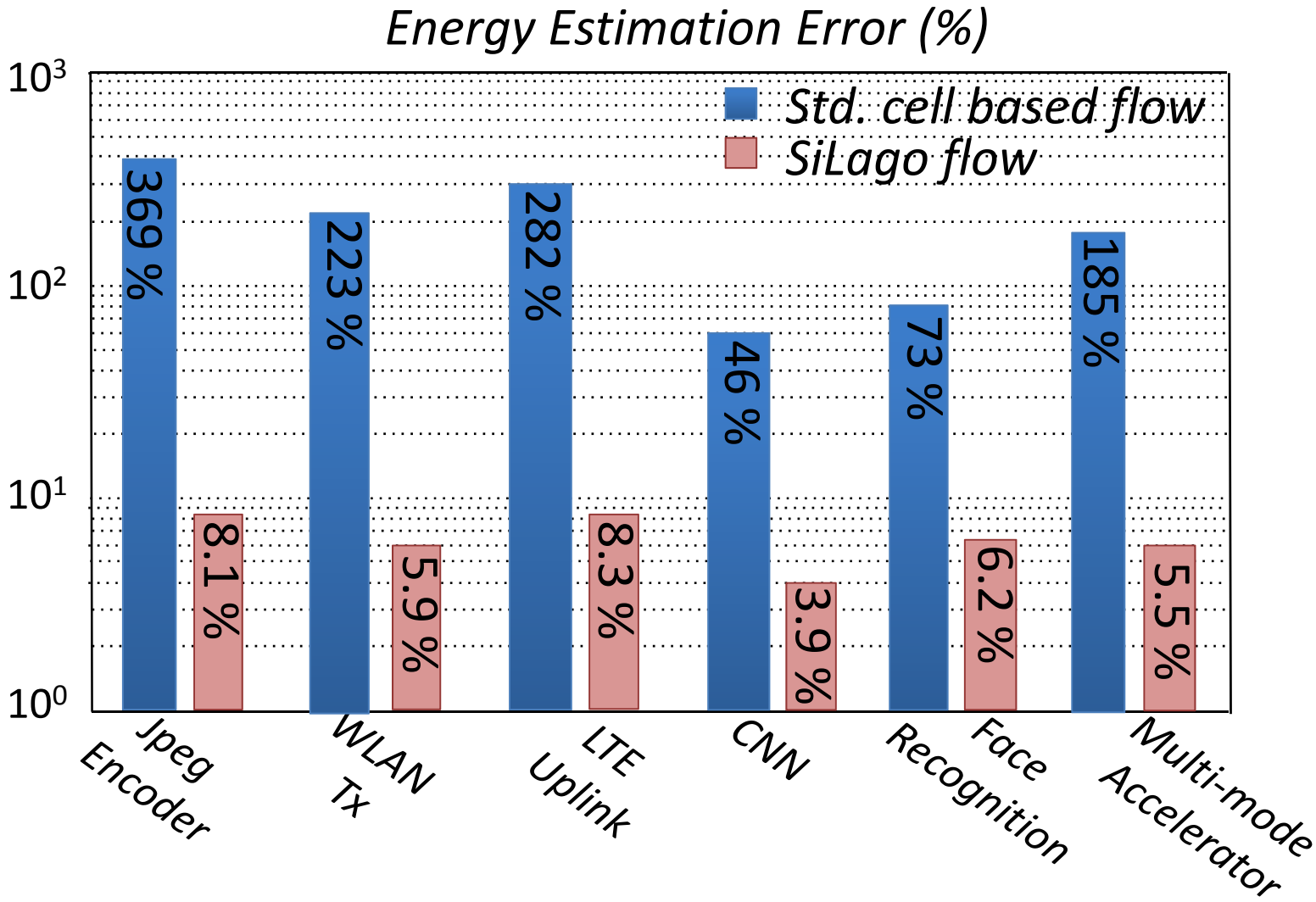




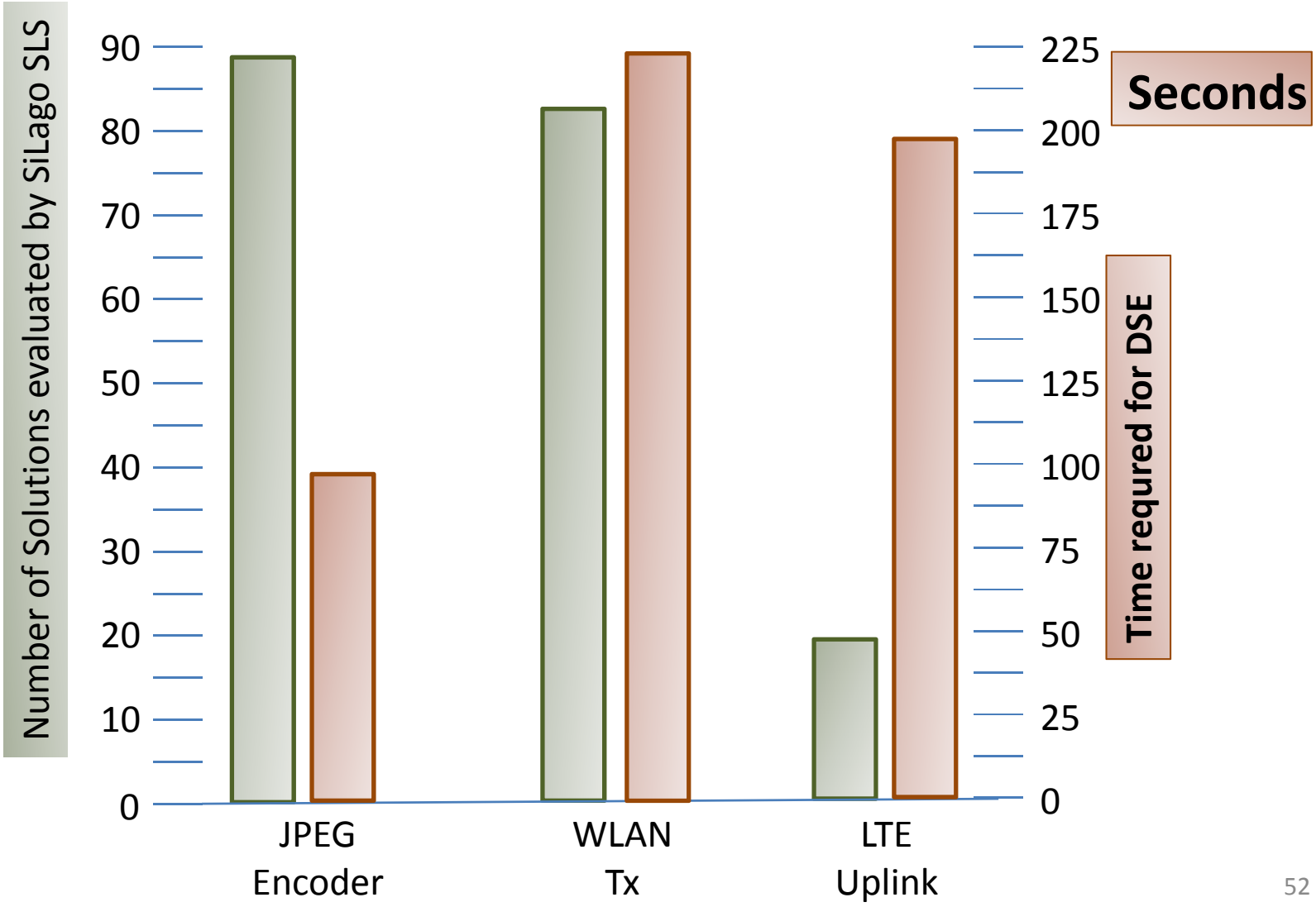
Normalized Energy and Area overhead of the Systems generated by the SiLago Design Flow



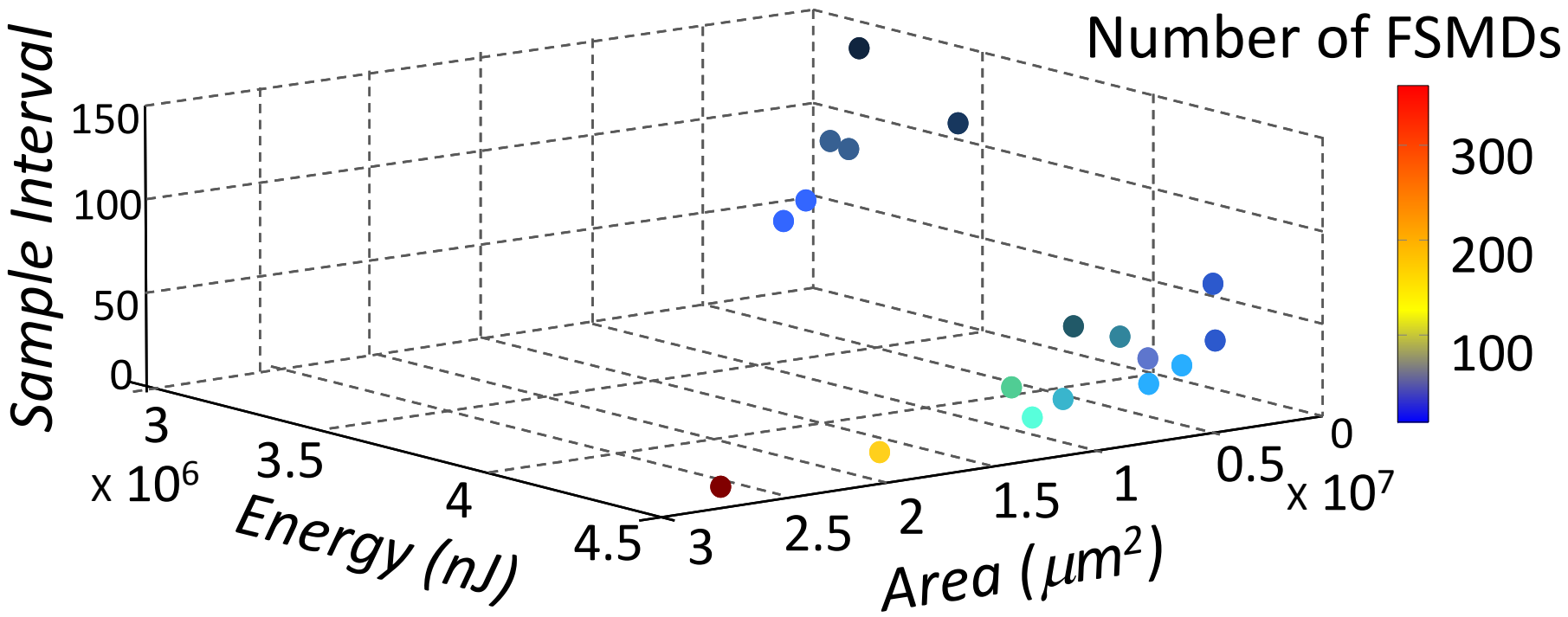
SiLago provides significant improvement in Predictability



Design Space Exploration in SiLago Application-level Synthesis



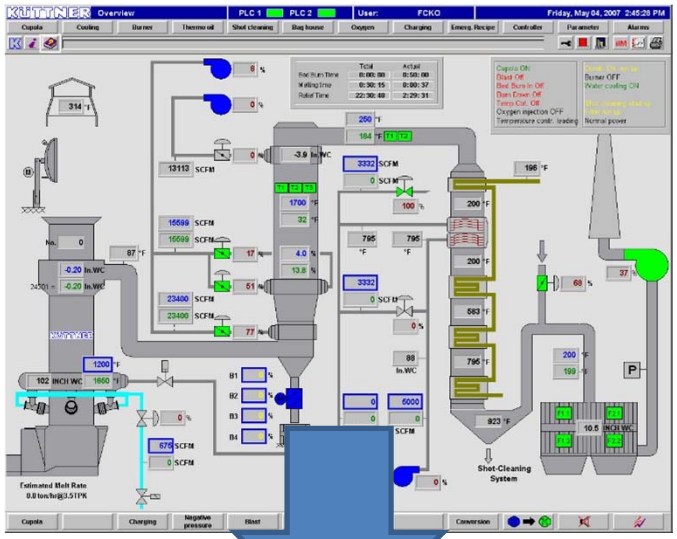
SiLego Design Space Exploration



Application of SiLago to Neuromorphic Computing

The Evolution of Embedded Systems

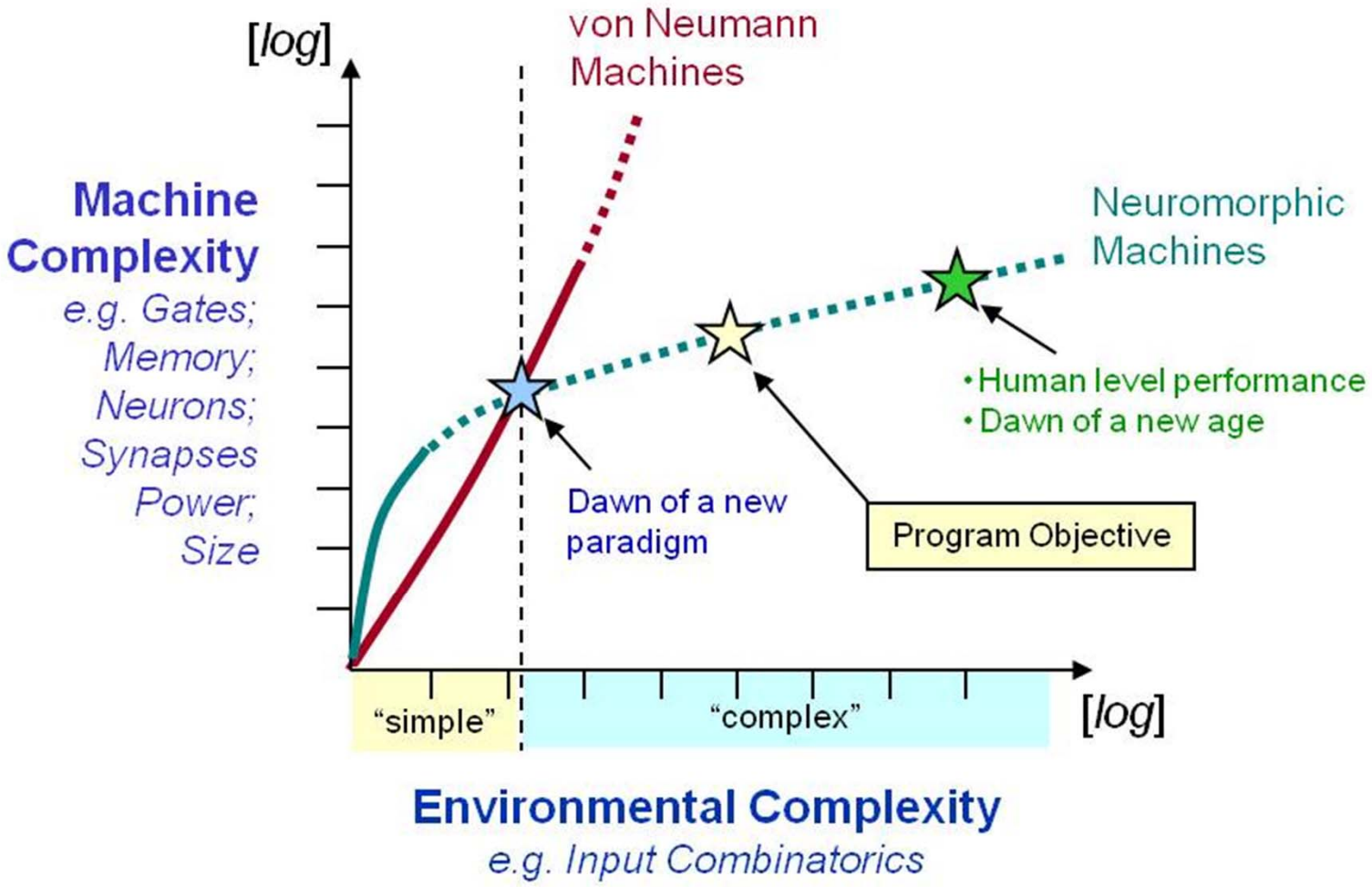
Interaction between machine and environment



Static and Finite

Dynamic and Infinite

Neuromorphic Machines are the answer



Reference: http://www.21stcentech.com/heard-synapse/

Implementing Brain in Electronics is non-trivial

Abstract model of Cortex
BCPNN → 1 PetaFlops

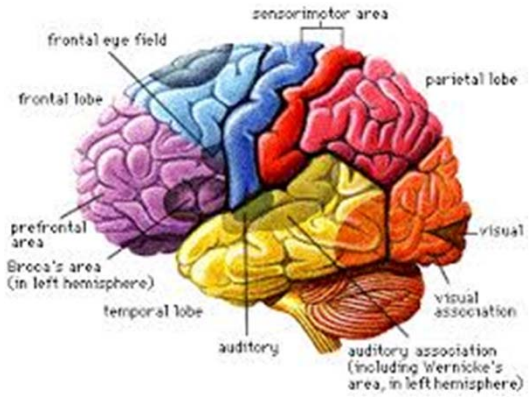
Riken - World's most efficient supercomputer
7 GFlops/watt. BCPNN → 140 kWs

20 Watts

Realistically 1 Mega Watts

What can eBrain achieve ?

20 Watts



~30 MilliWatts

The most efficient Supercomputer

~1 Mega Watts

~1000 Watts

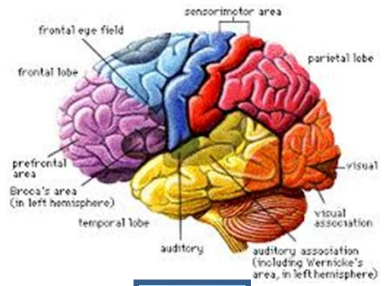
eBrain@KTH

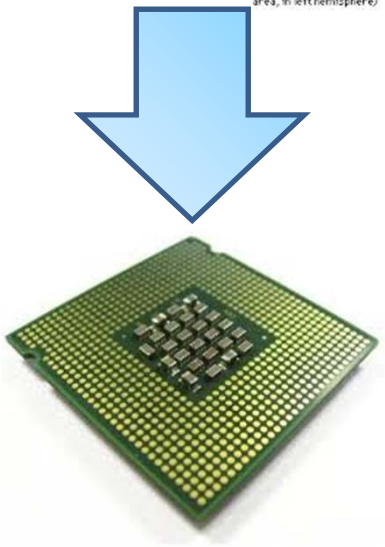
2 Kilo Watts

SiLago
Enables design automation of such high performance custom computing machines

2 Watts

BCPNN Requirements



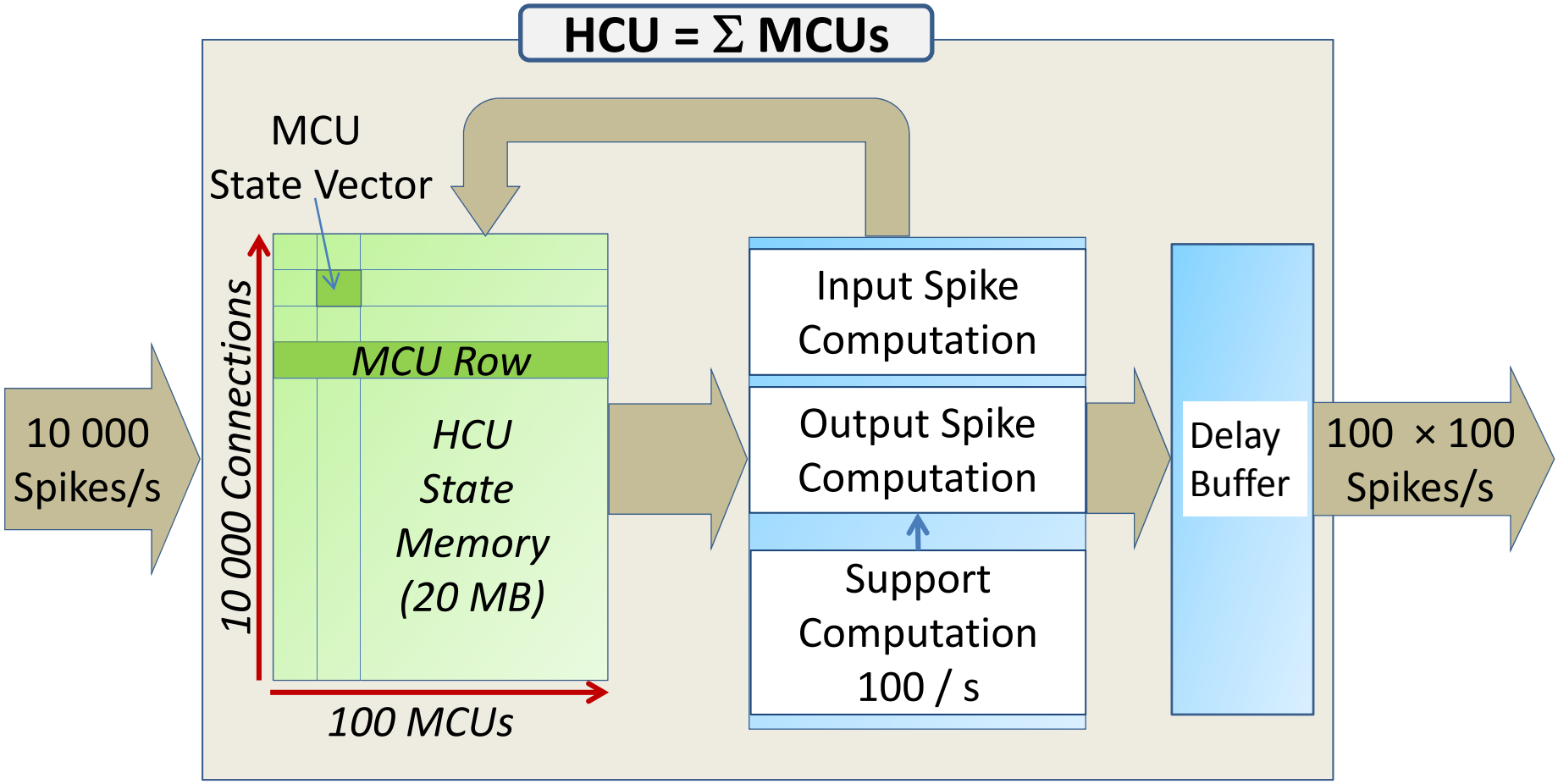


Functional Requirements

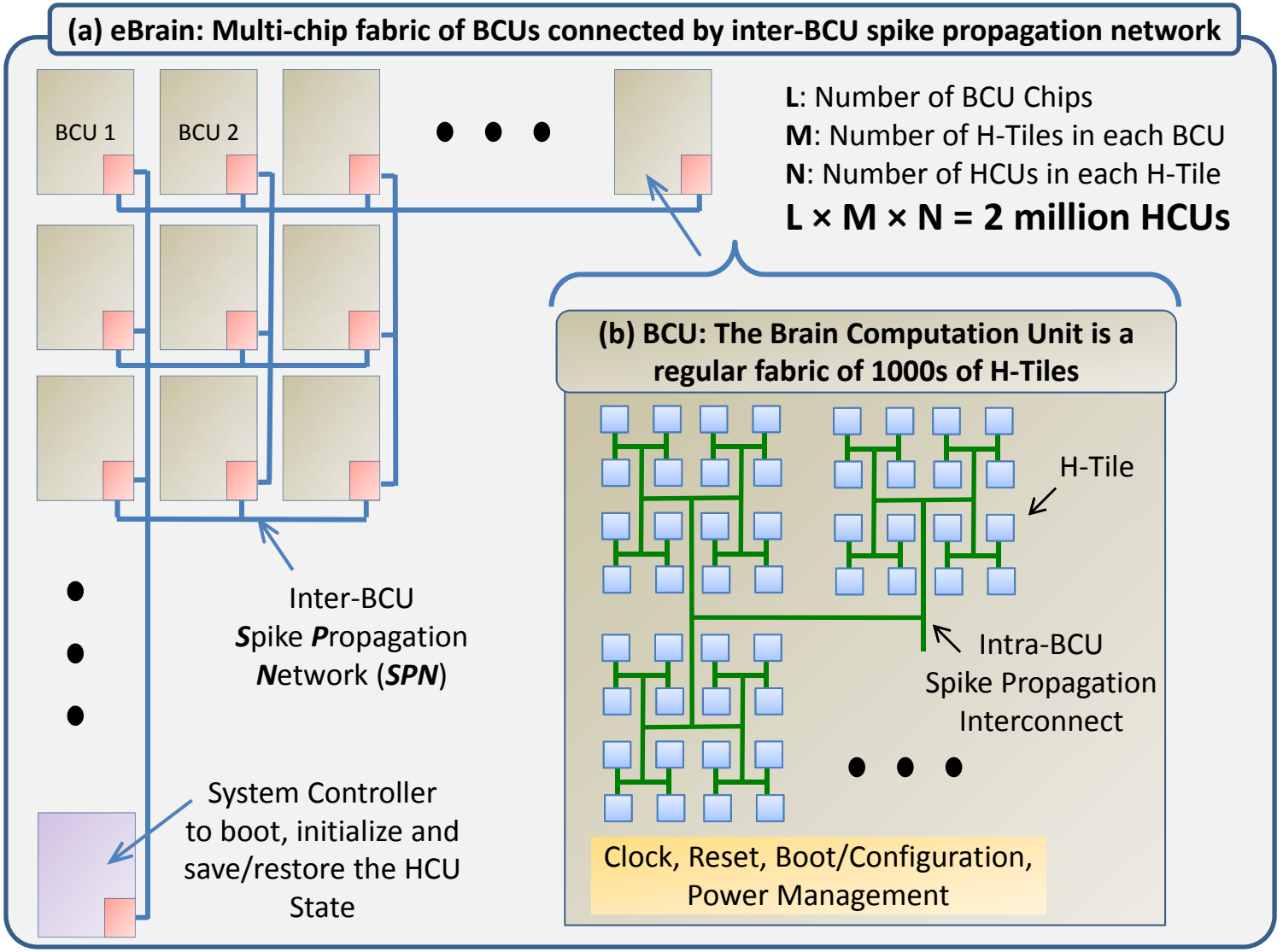
1. Realtime simulation
2. 2 Million HCUs
3. 1 Petaflops/sec – BCPNN Computation
4. 40 TBs – HCU State Storage
5. 130 TBs / s - Bandwidth
6. 20 billion spikes / s

Infrastructural Requirements

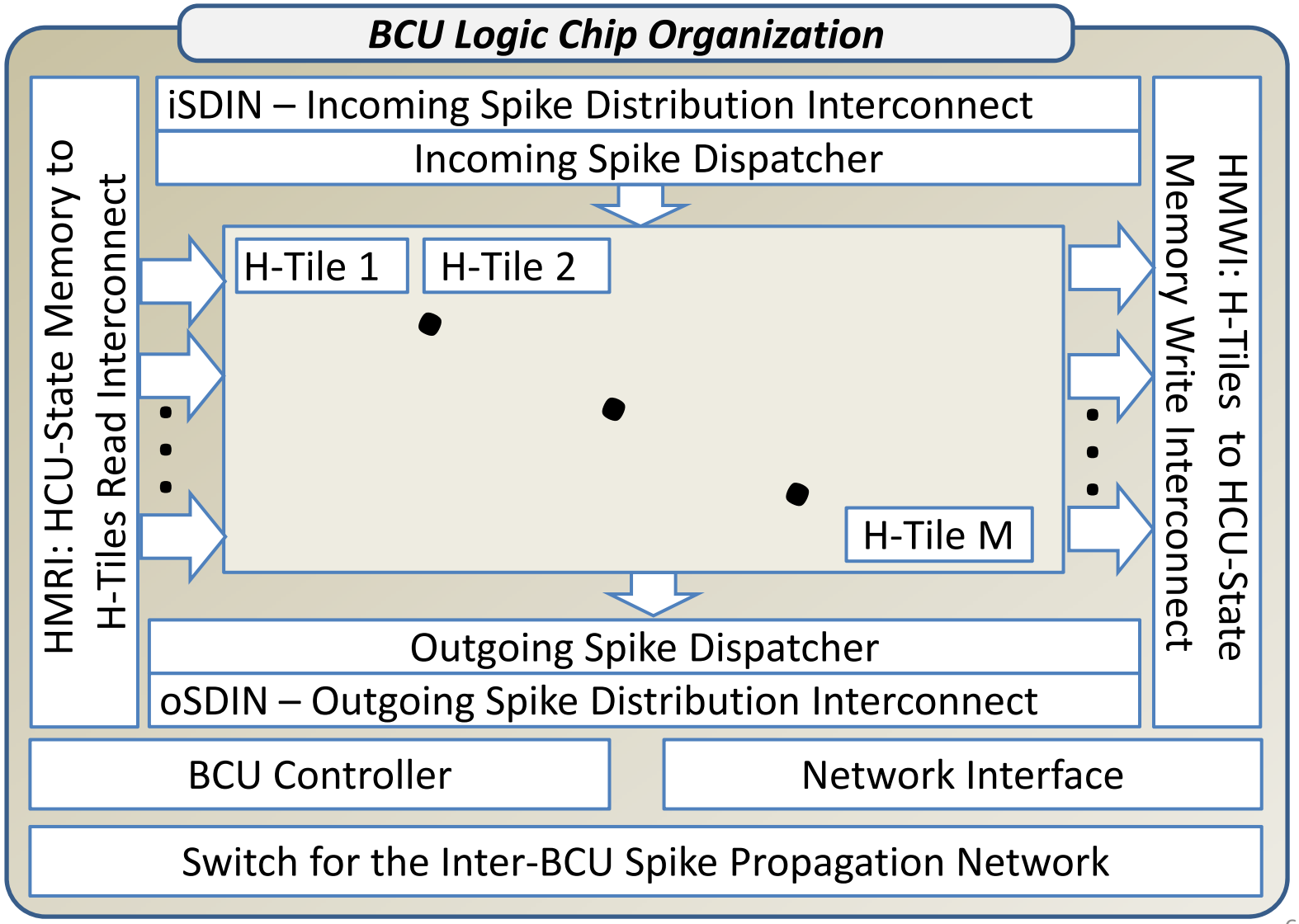
The BCPNN Computation Model



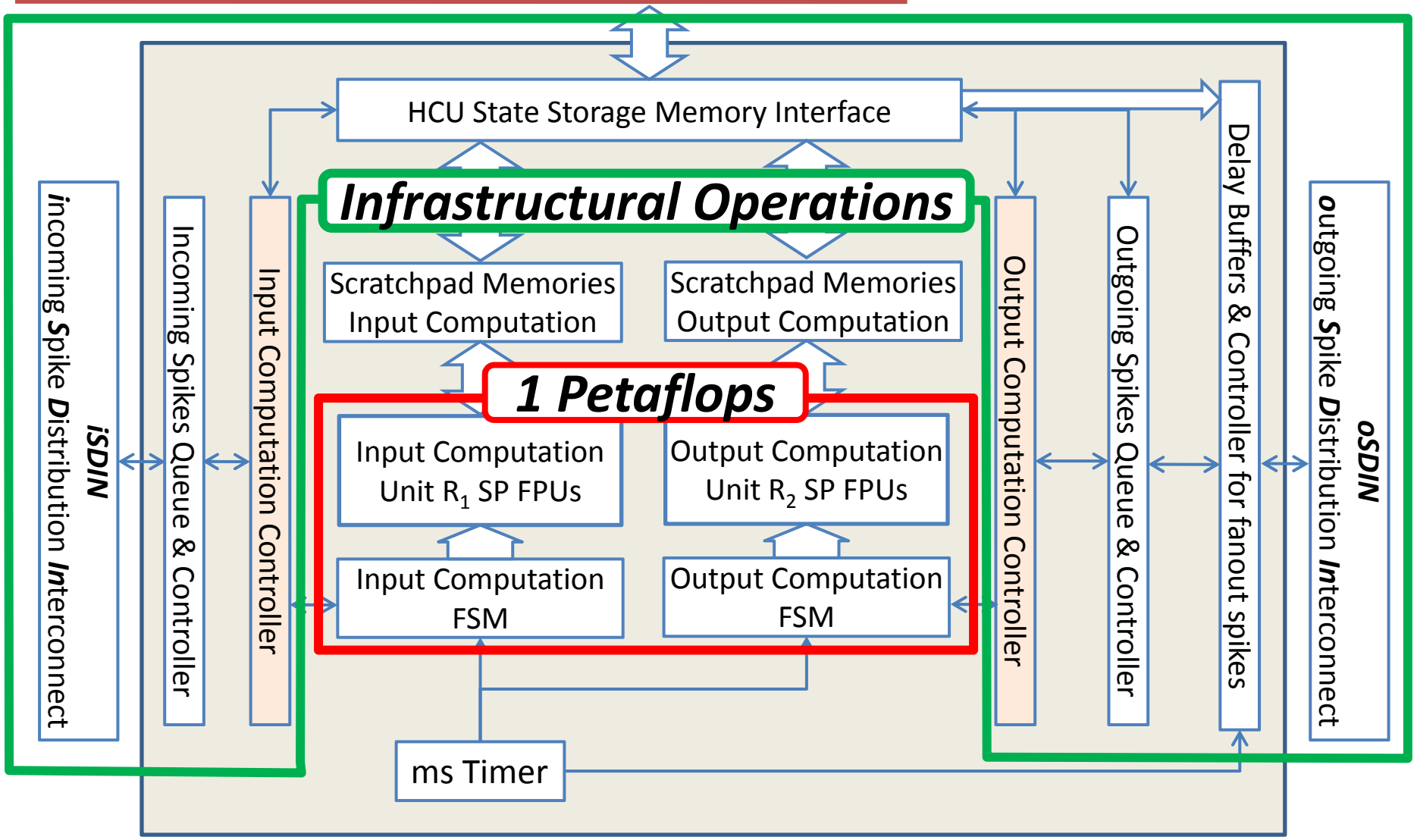
The eBrain System Concept



BCU Logic Chip - Organisation

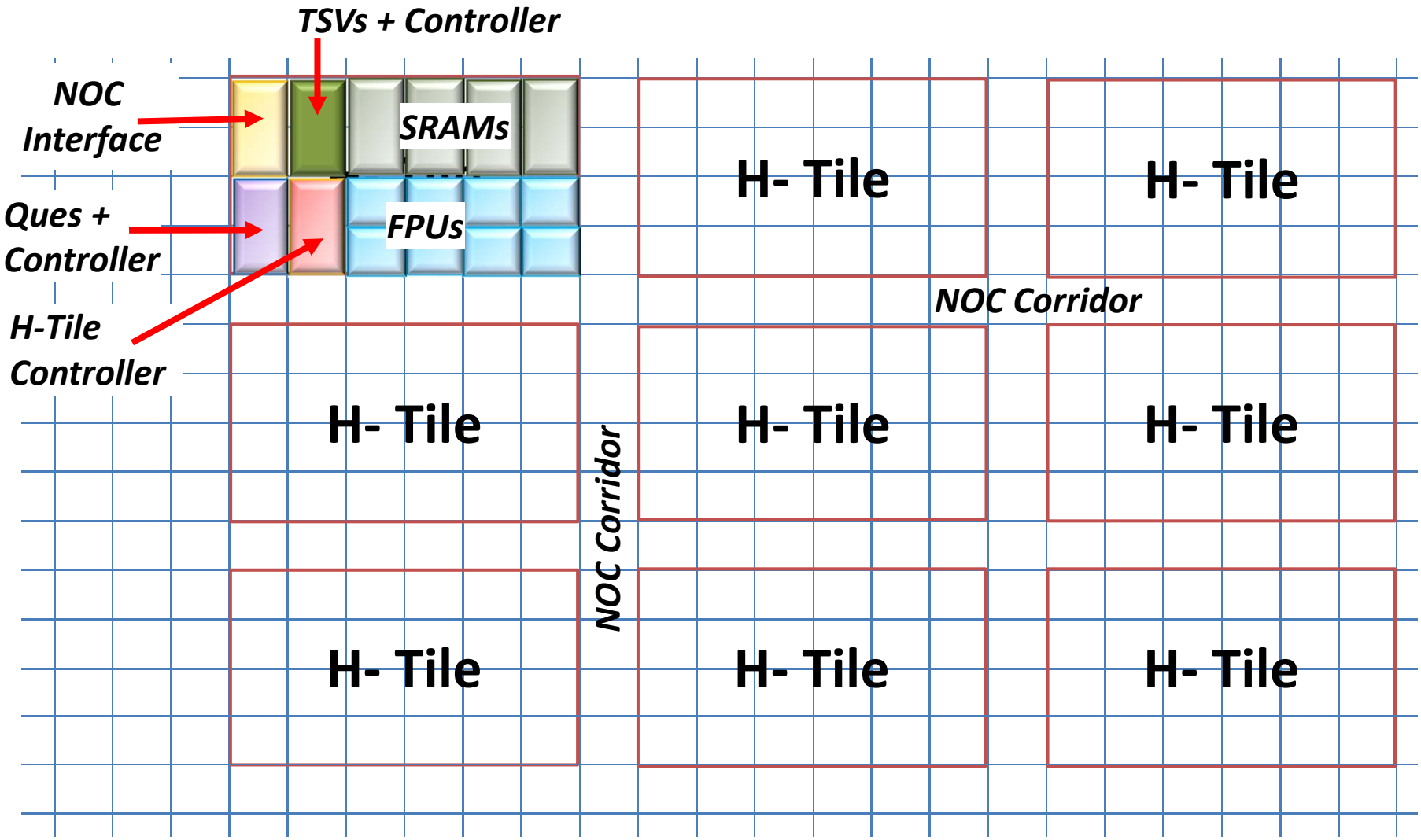


H-Tile Organisation



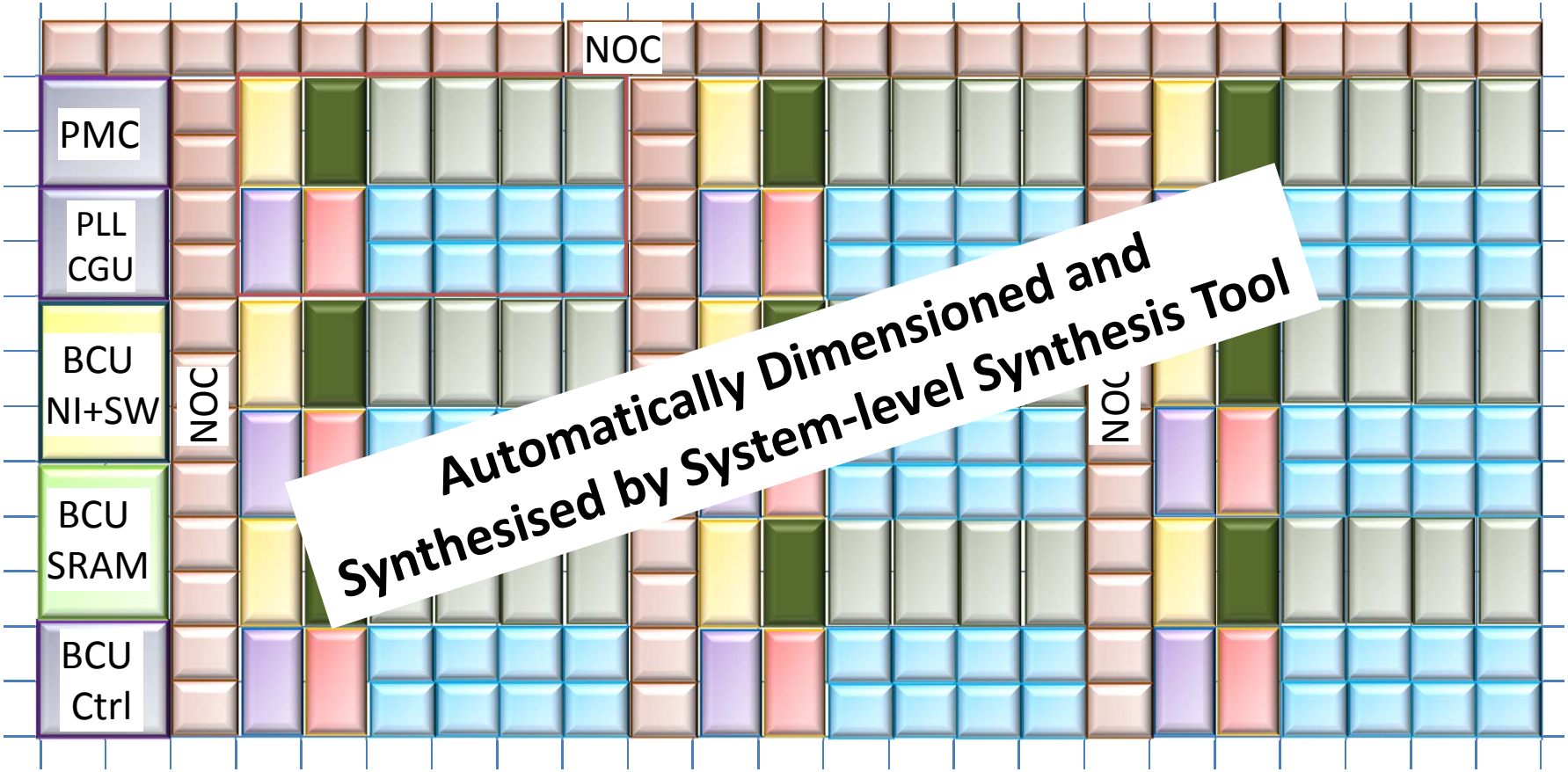
The SiLago Method

A Structured Physical Design Scheme to enable System-level synthesis



The SiLago Method

A Structured Physical Design Scheme to enable System-level synthesis



The Basis for dimensioning

Technology

- *22 nm node*
- *3D integrated custom DRAM*
- *16 X 82 mm² die integrated on an interposer*

Mouse

- *31 250 HCUs*
- *71 MCUs and 1225 connections*

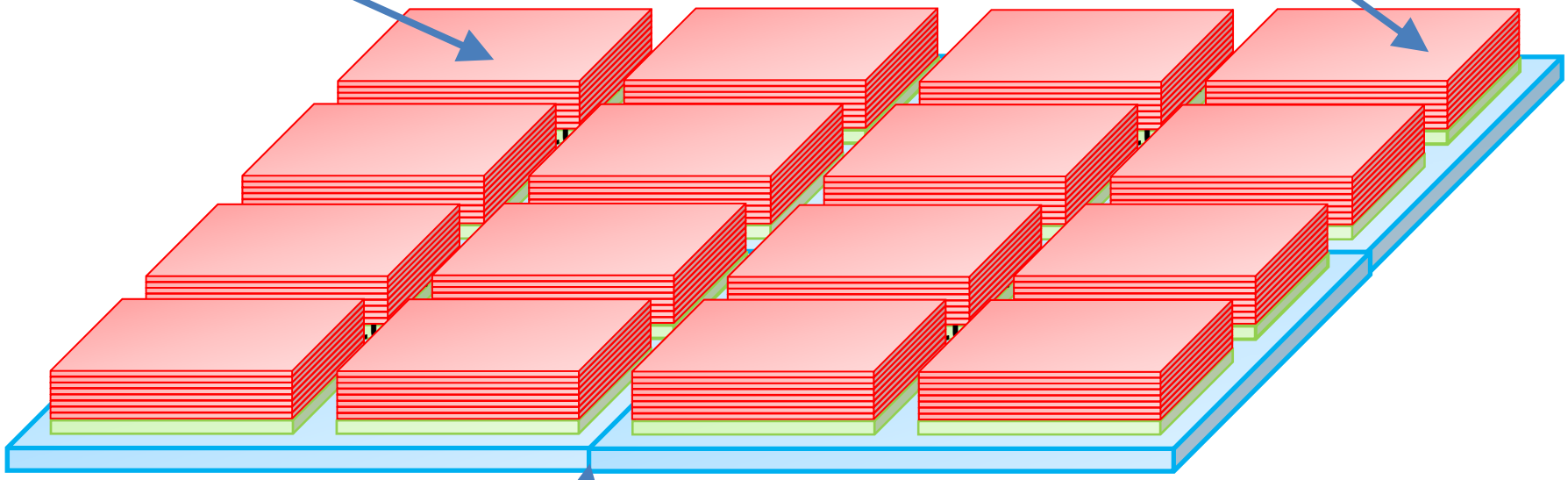
Results

- *Post layout data for Logic*
- *40 nm results conservatively scaled to 22 nm node*
- *Qualified circuit level models of 3D DRAM from TU Kaiserslautern*

Mouse eBrain
Package Level Organisation

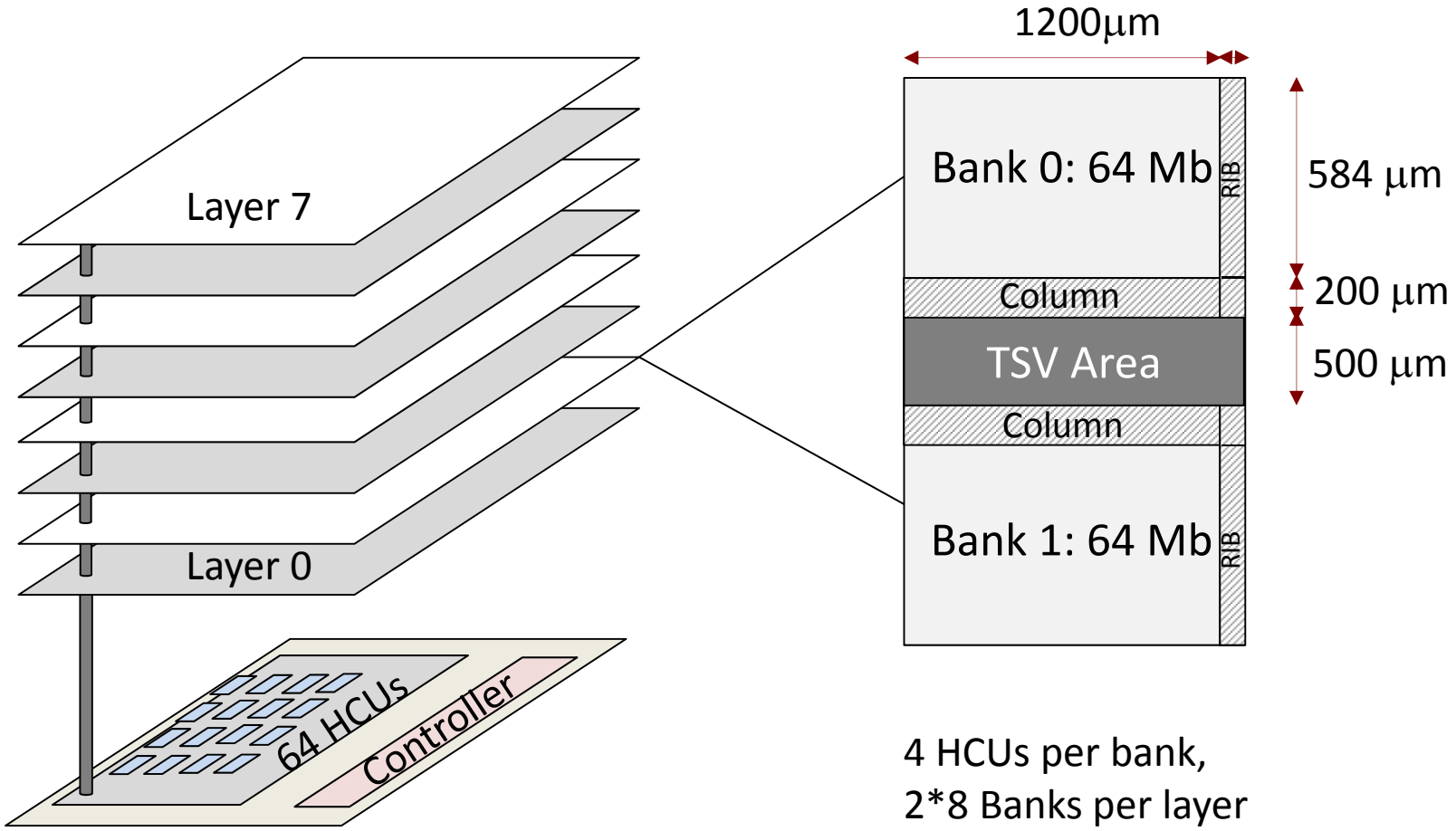
16 X 82 mm² chip with
8 layers of 3D DRAM and
32 channels per chip

32 H Tiles of 2.52 mm²
16 HCUs / H Tiles



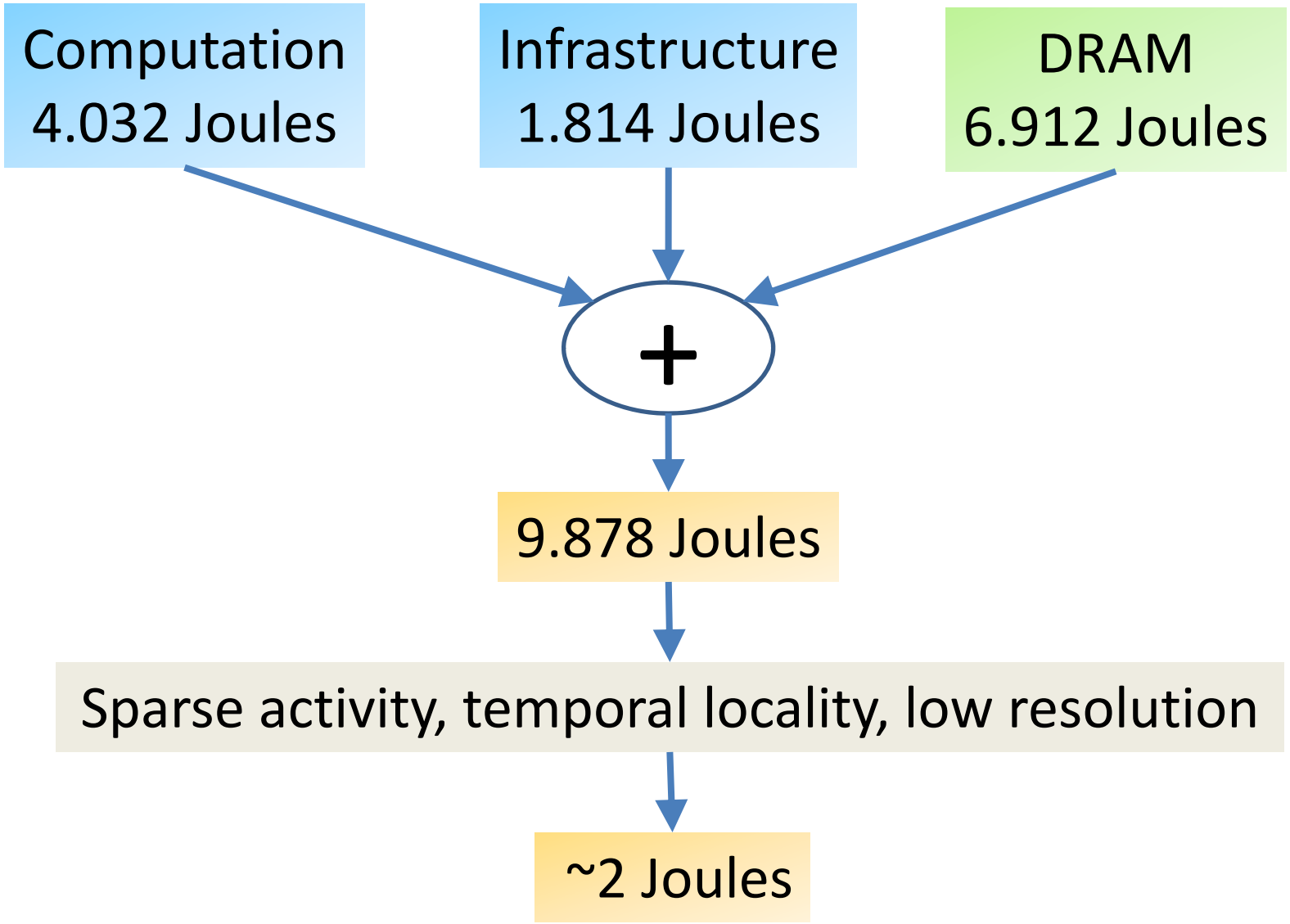
Interposer based package level inegration

Organisation and dimensions of H-Tile



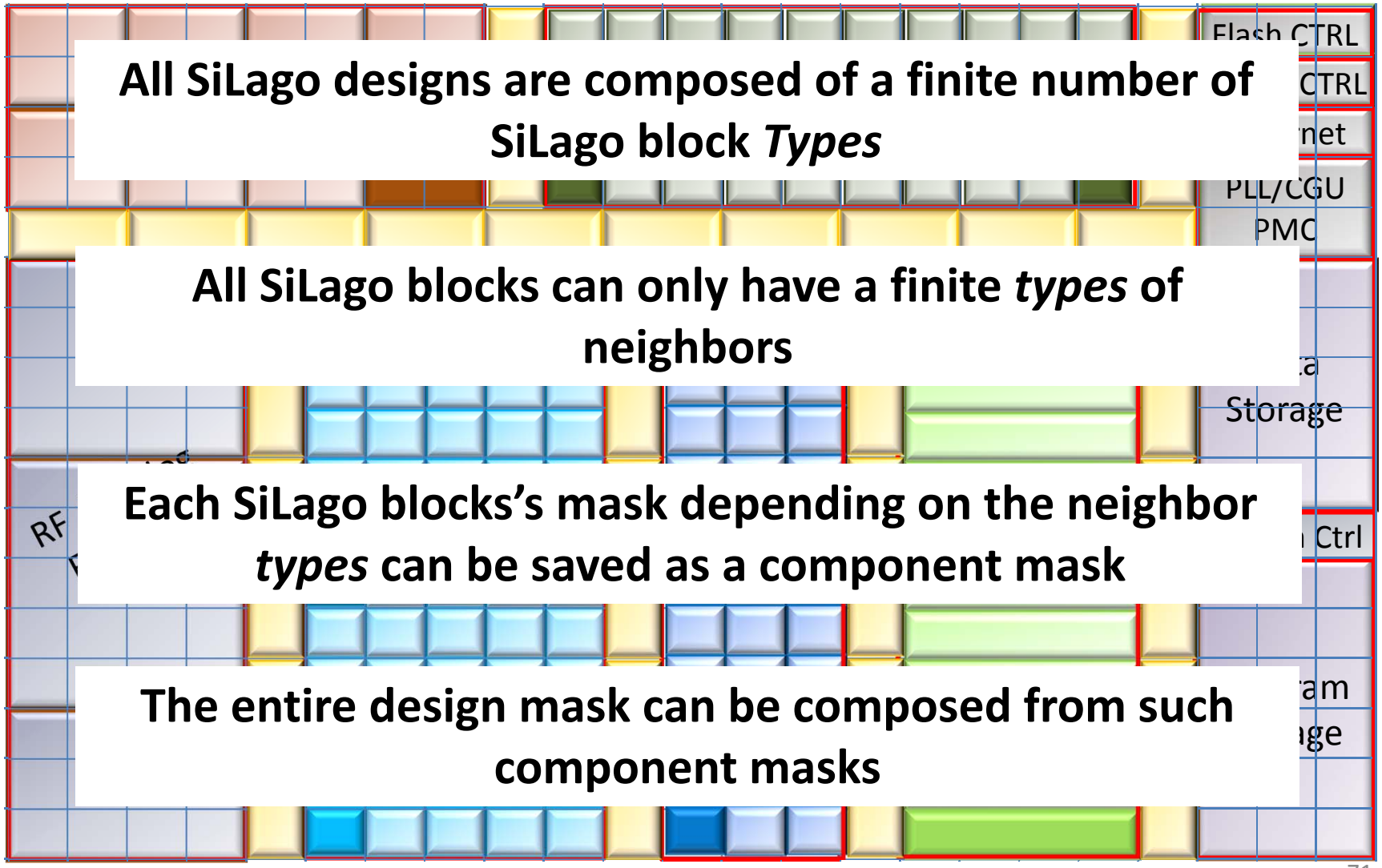
4 HCUs per bank,
2*8 Banks per layer
→ 64 HCUs per H-Tile

Energy Consumption



The SiLago Method also has the potential to lower the Manufacturing cost

SiLago can reduce the mask development cost



All SiLago designs are composed of a finite number of SiLago block *Types*

All SiLago blocks can only have a finite *types* of neighbors

Each SiLago blocks's mask depending on the neighbor *types* can be saved as a component mask

The entire design mask can be composed from such component masks

Future & Ongoing Work

SiLago Regions are being expanded to cover the 13 dwarfs of the Berkeley report on parallel computing

Extending Application Level Synthesis to System Level Synthesis
Ability to deal with non-determinism

Using SiLago Method to design

1. Complex Radio Systems – project with Catena
2. Custom Supercomputer for brain simulation and bioinformatics
3. Resilient autonomous systems based on neural networks

Extending SiLago to 3D SiLago to achieve end-to-end parallelisms

Thanks for your attention !
Questions ?