CUstom Built HEterogeneous Multi-Core ArCHitectures (CUBEMACH): Breaking the Conventions

Nagarajan Venkateswaran
Director, Waran Research Foundation

Karthikeyan Palavedu Saravanan - Nachiappan Chidambaram Nachiappan
Research Trainees (2008 - 2010), Waran Research Foundation

Aravind Vasudevan - Balaji Subramaniam - Ravindhiran Mukundarajan
Former Research Trainees (2007 - 2009), Waran Research Foundation

Motivation : Heterogeneity Redefined

- Cost Effective High Performance Custom Built Heterogeneous Multi-Core Node Design for wider class applications
  - Inter and Intra core heterogeneity
- Breaking the Conventions
  - Multiple User Multiple Application without Space-Time sharing in a Cluster : Cost sharing across users
  - Single User Multiple Application without Space-Timer Sharing (non-multiprogramming) : Cost sharing across applications

Overview

- Custom Built Heterogeneous Multi-Core Architectures (CUBEMACH)
- Design Space
  - – Architectural Space
  - – Optimization Space
    - – Customer Vendor Interaction
  - – Simulation Space
- CUBEMACH Design and Simulation Tool Framework
- Conclusion

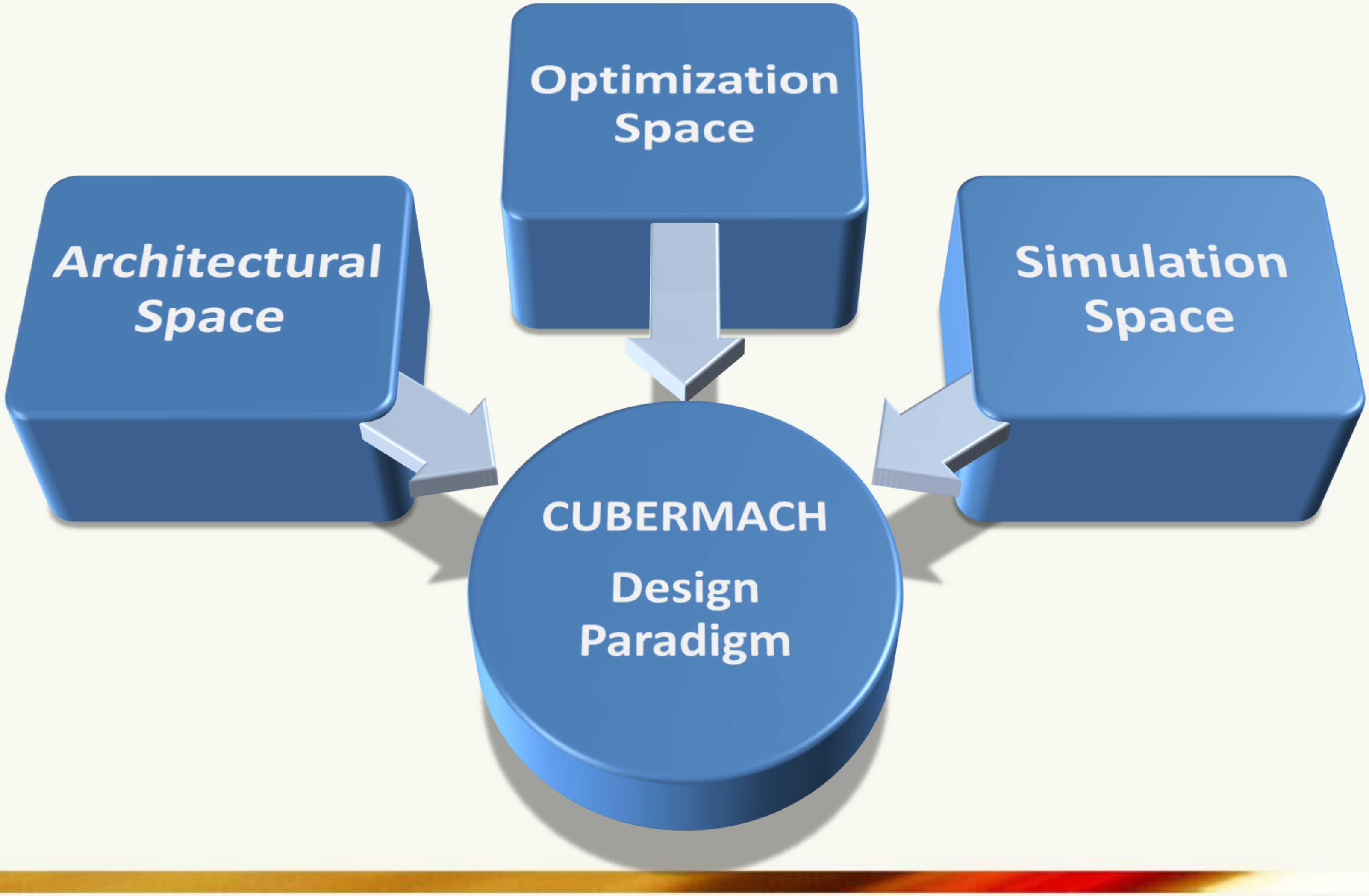
Custom Built Heterogeneous Multi-Core Architectures (CUBEMACH)

- CUBEMACH promises
  - – Increased Resource Utilization
  - – Multiple Application Flavored Architectures
  - – Elimination of Space Time Sharing at the Quantum Level during Multiple Application Execution
  - – Manufacturing and Operational Cost reduction

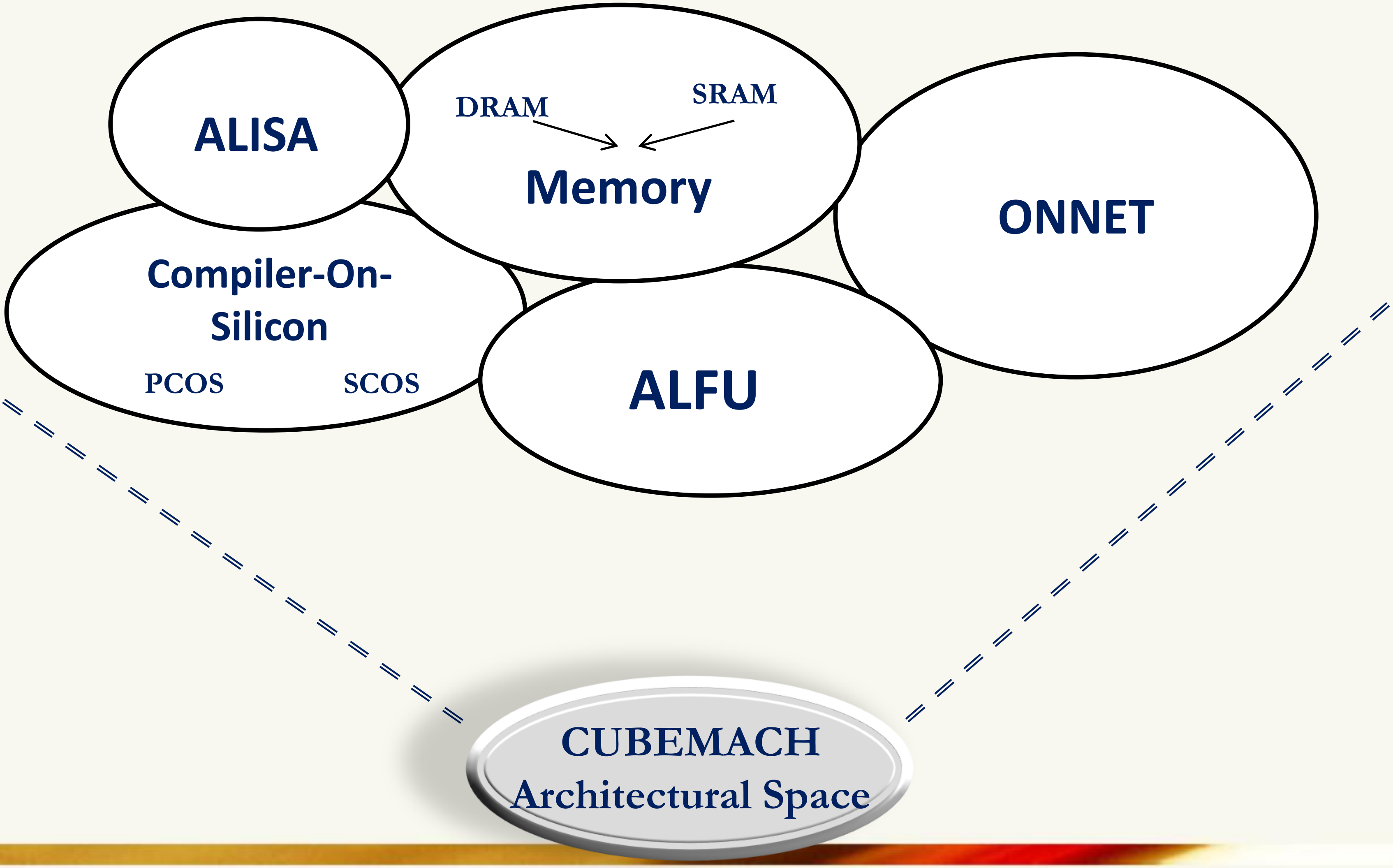
Overview

- Custom Built Heterogeneous Multi-Core Architectures (CUBEMACH)
- Design Space
  - Architectural Space
  - Optimization Space
    - Customer Vendor Interaction
  - Simulation Space
- CUBEMACH Design and Simulation Tool Framework
- Conclusion

CUBEMACH Design Paradigm



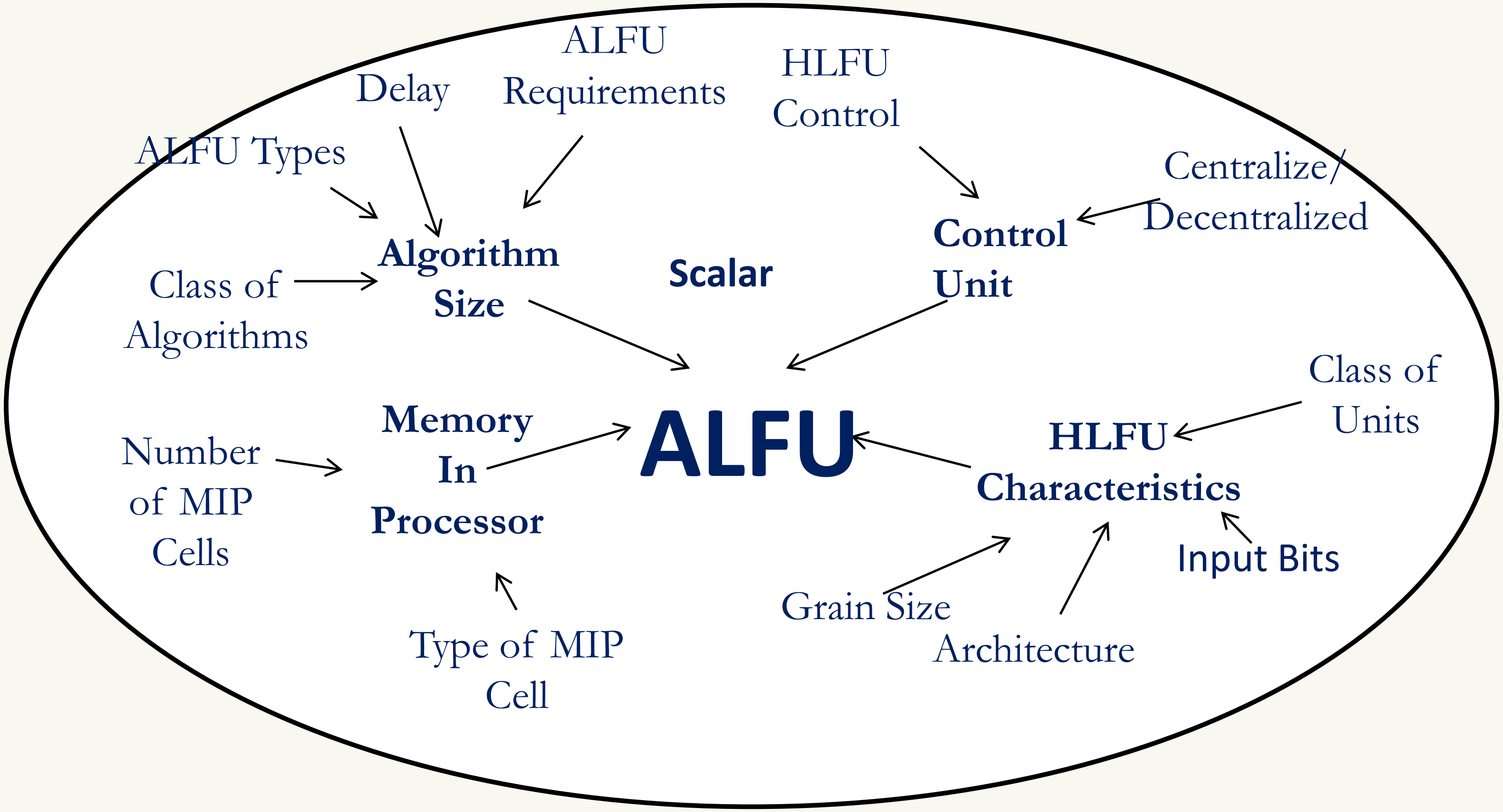
Architectural Design Space - CUBEMACH



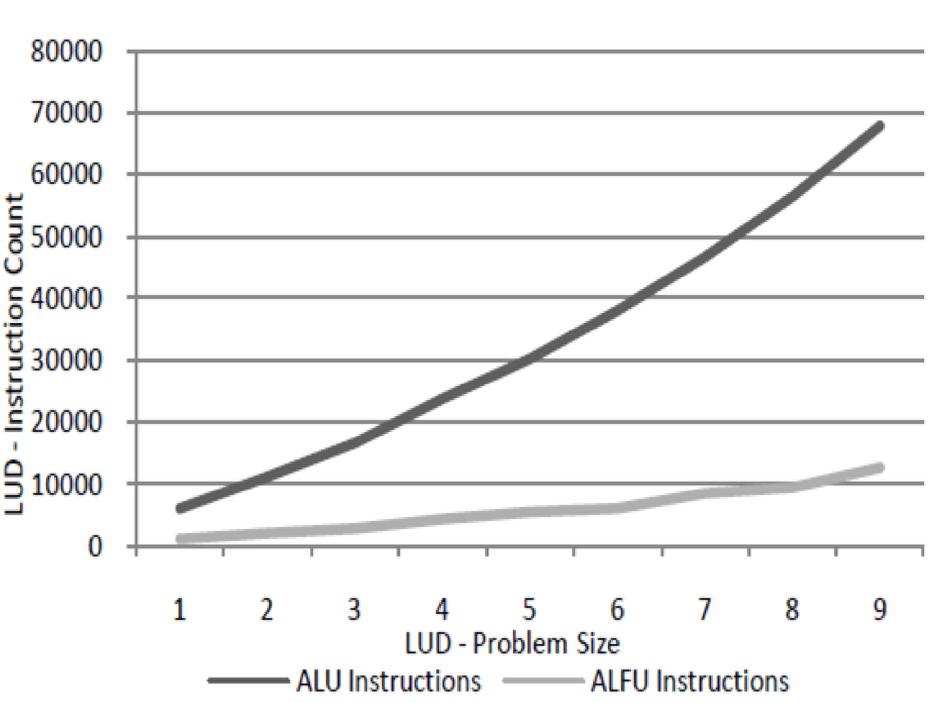
Architectural Space

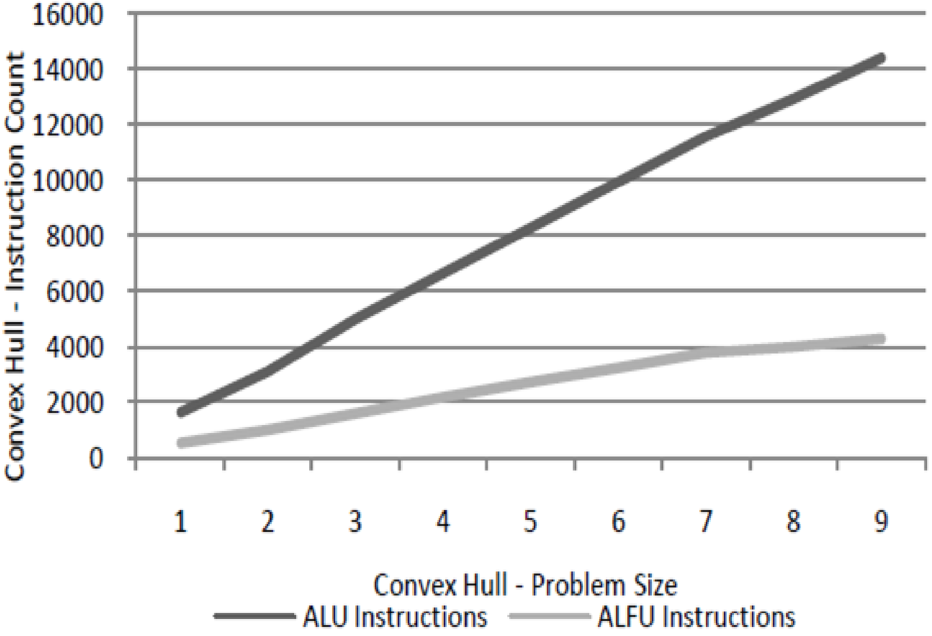
- Why ALU Why Not ALFU??
  - Hardwired units
  - Design : Homogeneously Structured
  - Reduced Instruction Generation & Fetches : Employ a Higher Level ISA
  - Reduced memory-functional unit interaction
  - Helps execute multiple applications without space & time sharing

Algorithm Level Functional Unit



ALU vs ALFU Instruction Generation Results



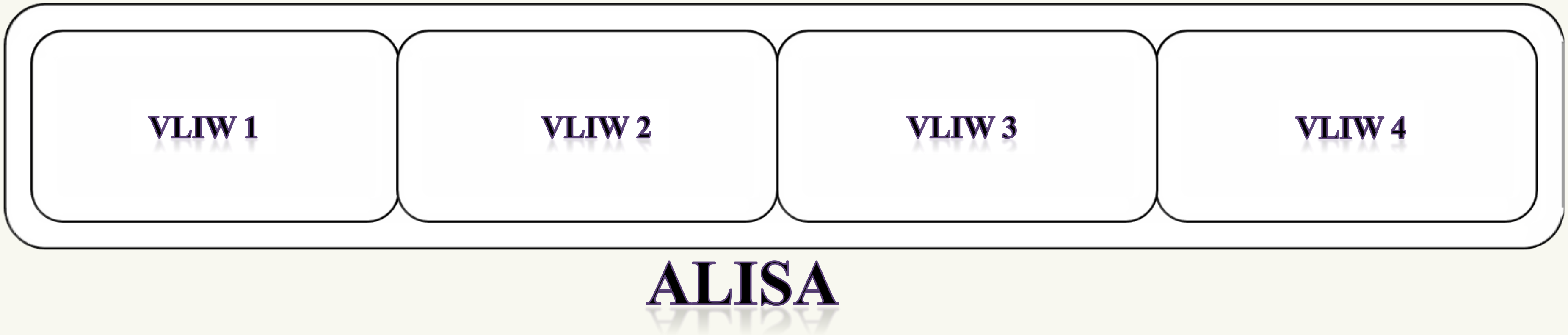


Sample Algorithm Level Functional Units

- **Scalar Units**
  - Scalar Adder / Subtractor
  - Scalar Multiplier
  - Scalar Divider
  - Comparator
  - Sorter
  - Multiple Operand Adder
  - Min / Max Finder
- **Vector Units**
  - Inner Product
- **Matrix Centric Units**
  - Matmul
  - Matadd
  - Chain Matadd
- **Graph Theoretic Units**
  - Graph Traversal Unit – BFS, DFS
  - KL Graph Partitioning

ALISA – Algorithm Level Instruction Set Architecture

- Algorithm Level Instructions
- Triggers ALFUS
- ALISA ⟶ Multiple VLIWs
- ALISA for heterogeneous multi-cores



VLIW 1

VLIW 2

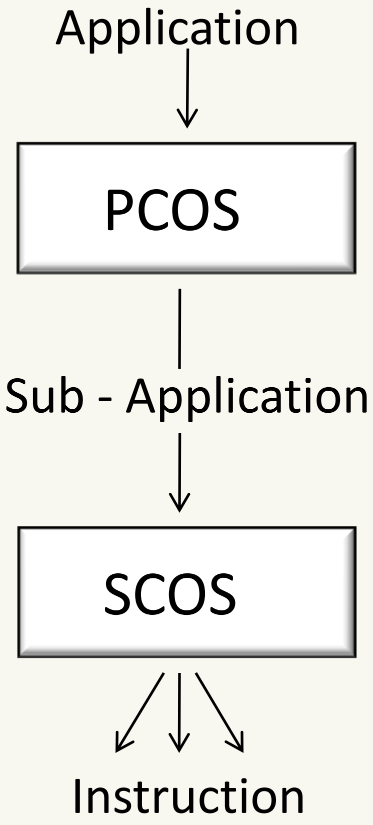
VLIW 3

VLIW 4

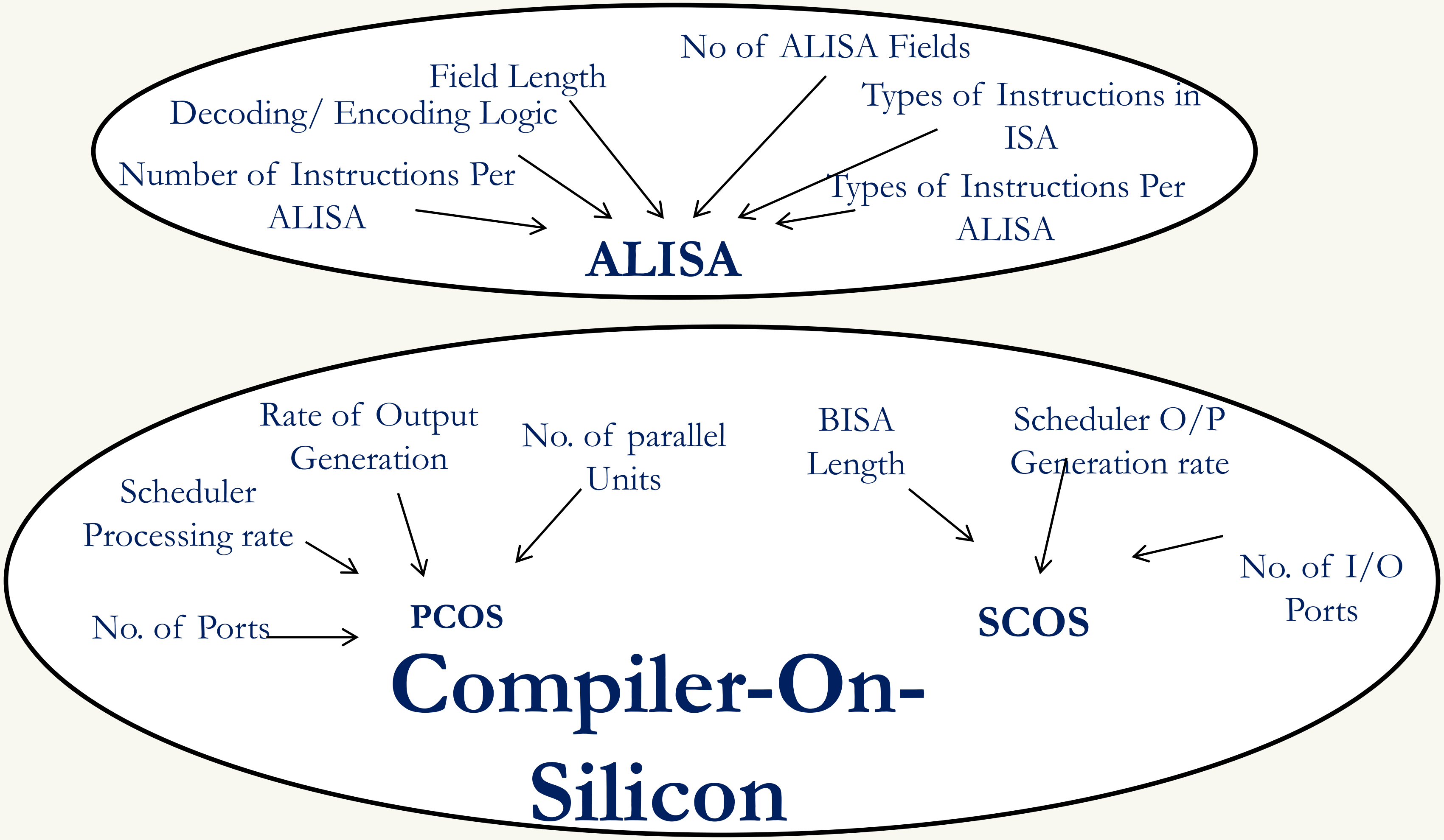
ALISA

Hierarchical Compilation Scheme

- PCOS Partitions A Problem Into Sub-Problems – Level 1
- SCOS Partitions The Sub-Problems Into ALFU Level Instruction – Level 2



ALISA & Compiler On Silicon



No of ALISA Fields

Field Length

Decoding/ Encoding Logic

Types of Instructions in ISA

Number of Instructions Per ALISA

Types of Instructions Per ALISA

ALISA

Rate of Output Generation

No. of parallel Units

BISA Length

Scheduler O/P Generation rate

Scheduler Processing rate

No. of I/O Ports

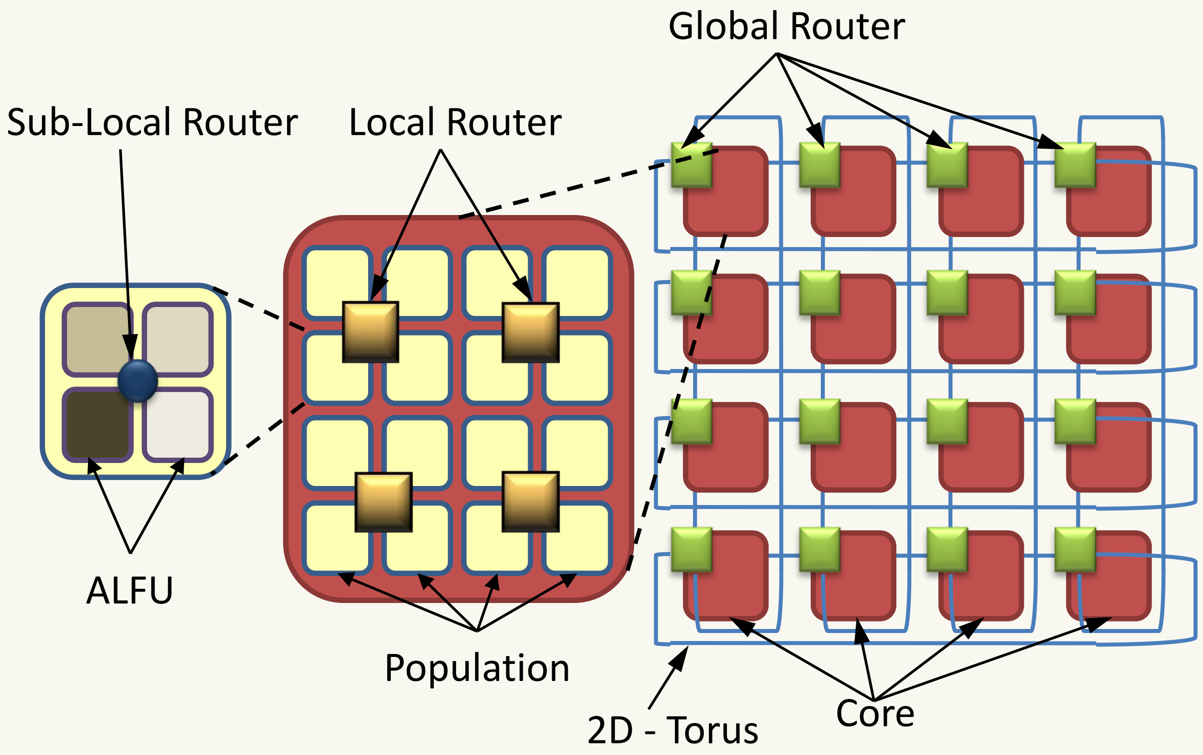
No. of Ports

PCOS

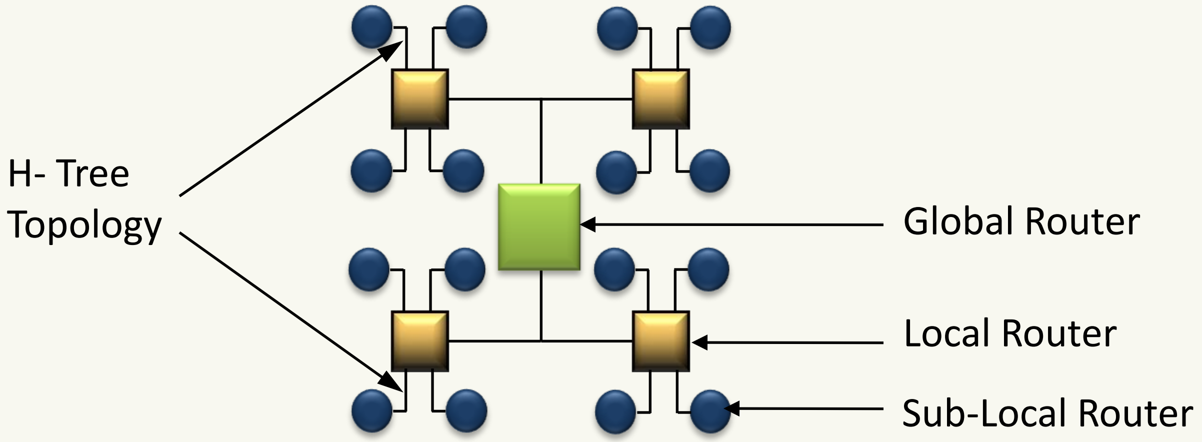
SCOS

Compiler-On-Silicon

ON-Node-Network Architecture



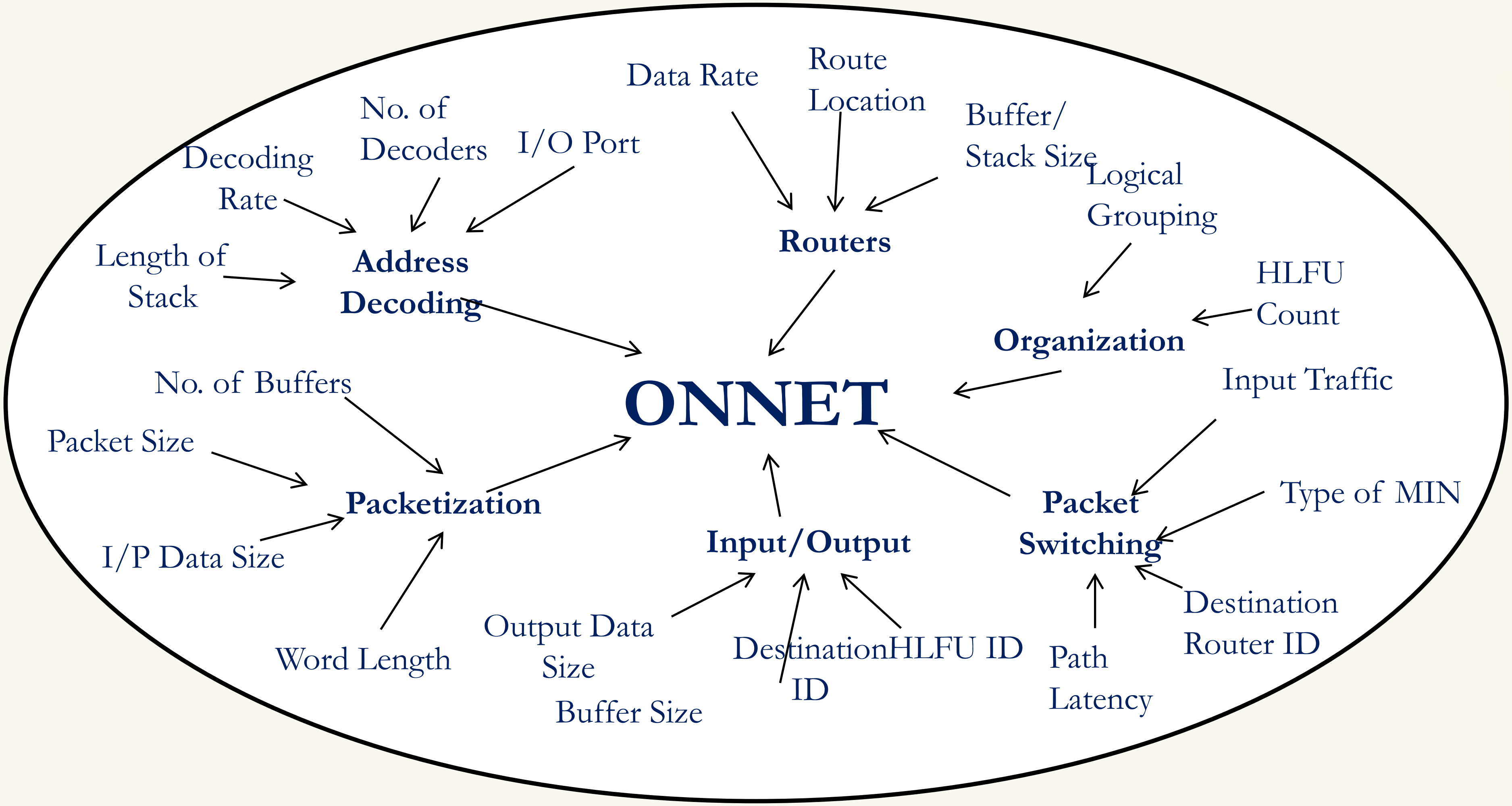
ON-Node-Network Architecture



Comparison of Conventional NOCs with ONNET

	ONNET	Conventional NOCs
Type of Switch	MIN	Crossbar
Number of Routers	$N* \log_{2} (N)$	N^2
Hierarchy	Yes	No
Switching Latency	Log_2(Number of Inputs) * Switch Delay	Number of Inputs * Switch Delay

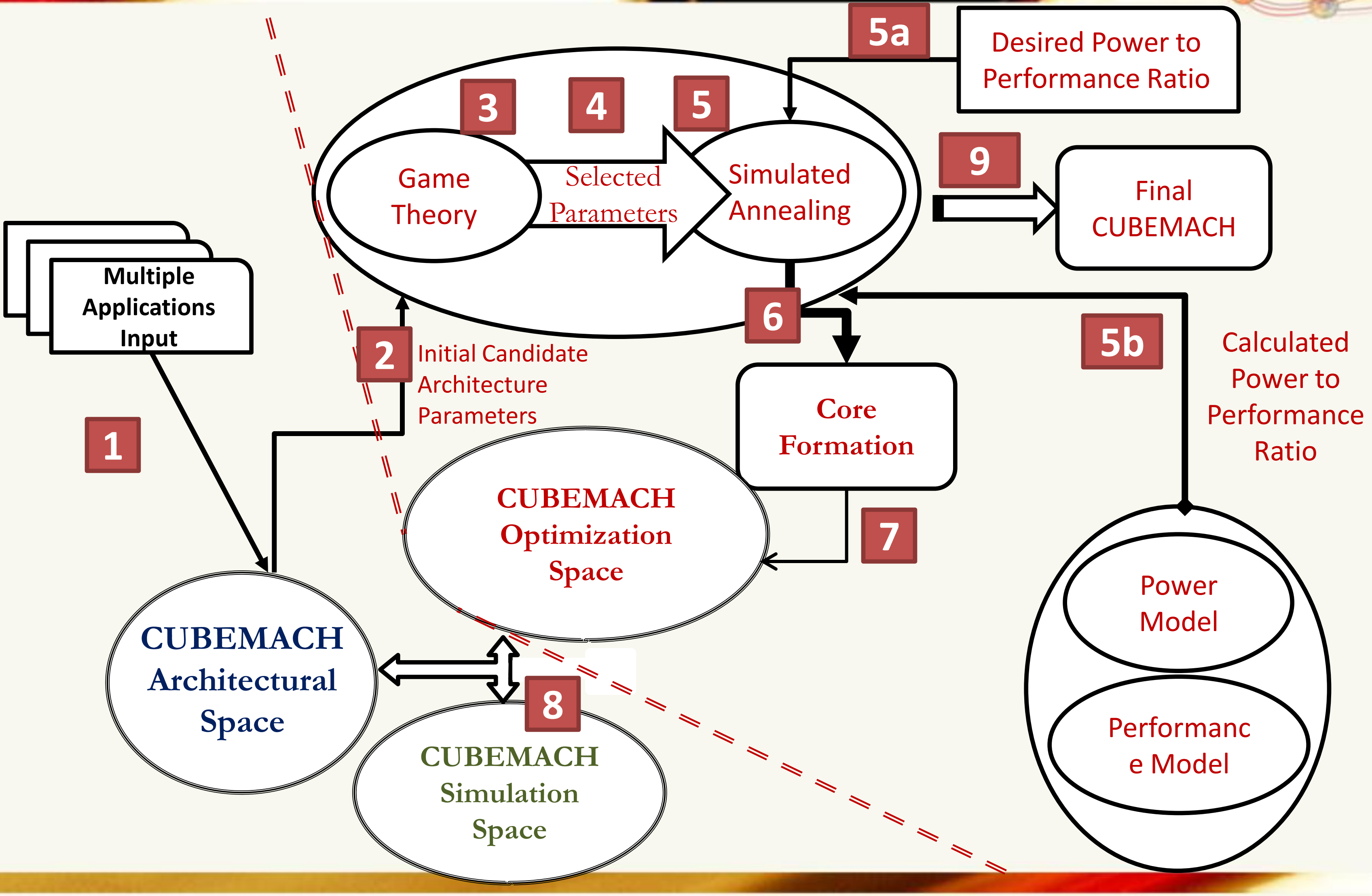
On Node Network Architecture



Overview

- Motivation
- Custom Built Heterogeneous Multi-Core Architectures (CUBEMACH)
- Design Space
  - Architectural Space
  - Optimization Space
    - Customer Vendor Interaction
  - Simulation Space
- CUBEMACH Design and Simulation Tool Framework
- Conclusion

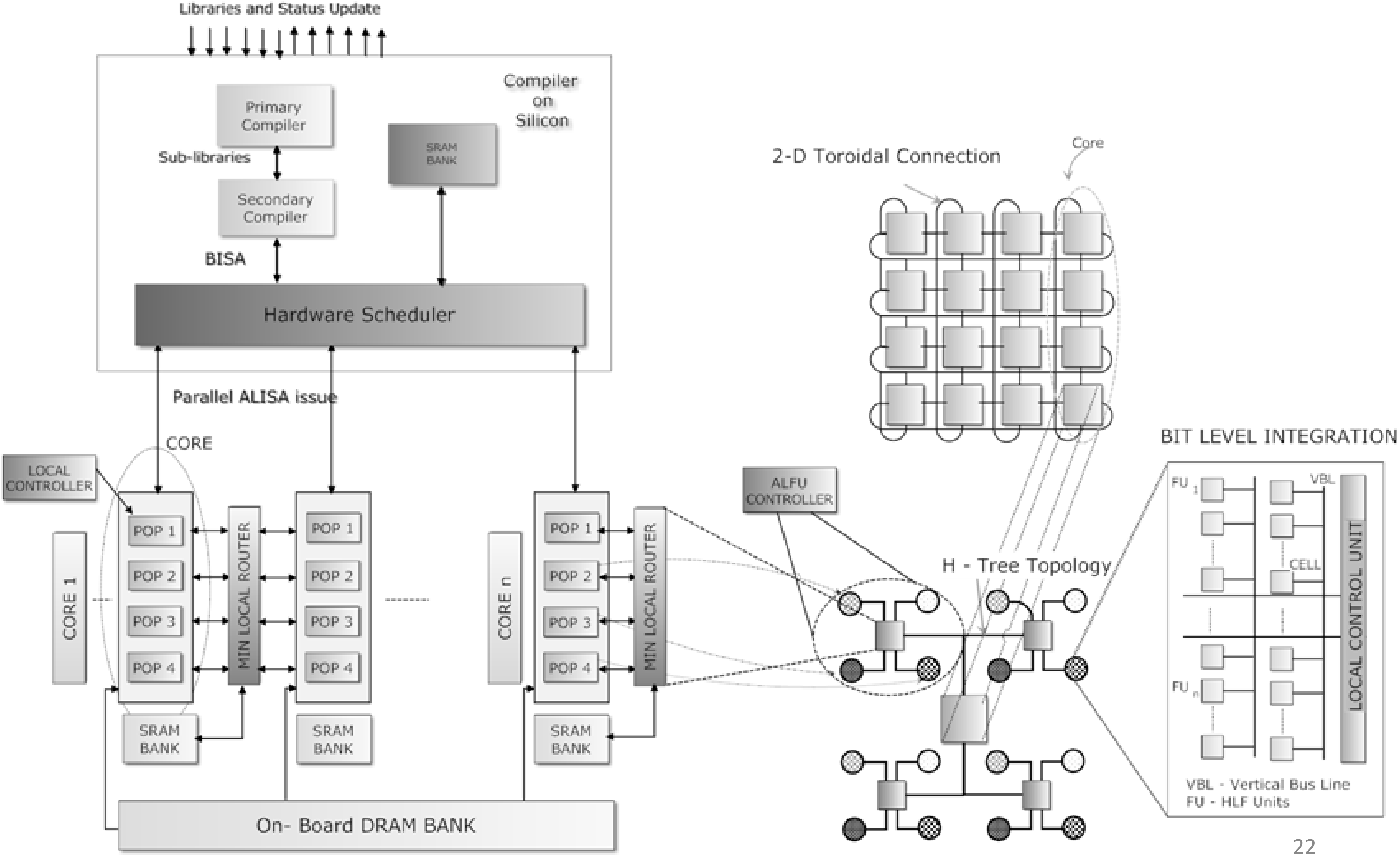
Optimization Space


5a
Desired Power to Performance Ratio
3
4
5
Game Theory
Selected Parameters
Simulated Annealing
9
Final CUBEMACH
Multiple Applications Input
6
2
Initial Candidate Architecture Parameters
5b
Calculated Power to Performance Ratio
1
Core Formation
CUBEMACH Optimization Space
7
CUBEMACH Architectural Space
8
CUBEMACH Simulation Space
Power Model
Performanc e Model

Optimization Space

- Generates Optimized CUBEMACH for input specifications such as,
  - Power – Performance – Cost
  - Initial Architecture
- Power and Performance Model
- Uses GT and SA for optimization of Power and performance
- Uses KL For Core Grouping

Sample CUBEMACH Architecture

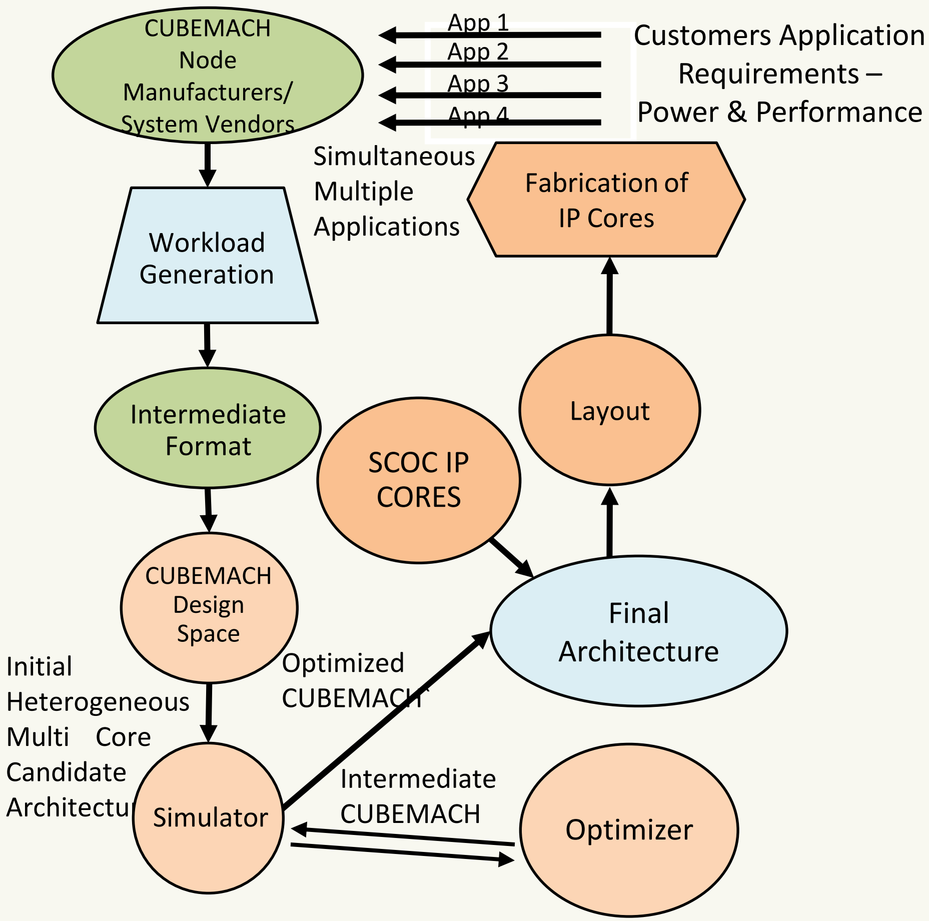


CUBEMACH Design Implementation : Supercomputer On Chip (SCOC) IP Cores

SCOC IP Cores

- ALFUs designed as SCOC IP Cores
- Soft IP Core
- Coarse-grained Reusable Soft IP Cores
- Scalable IP Cores

Customer Vendor Interaction



Overview

- Motivation
- Custom Built Heterogeneous Multi-Core Architectures (CUBEMACH)
- Design Space
  - – Architectural Space
  - – Optimization Space
    - –Customer Vendor Interaction
  - – Simulation Space
- CUBEMACH Design and Simulation Tool Framework
- Conclusion

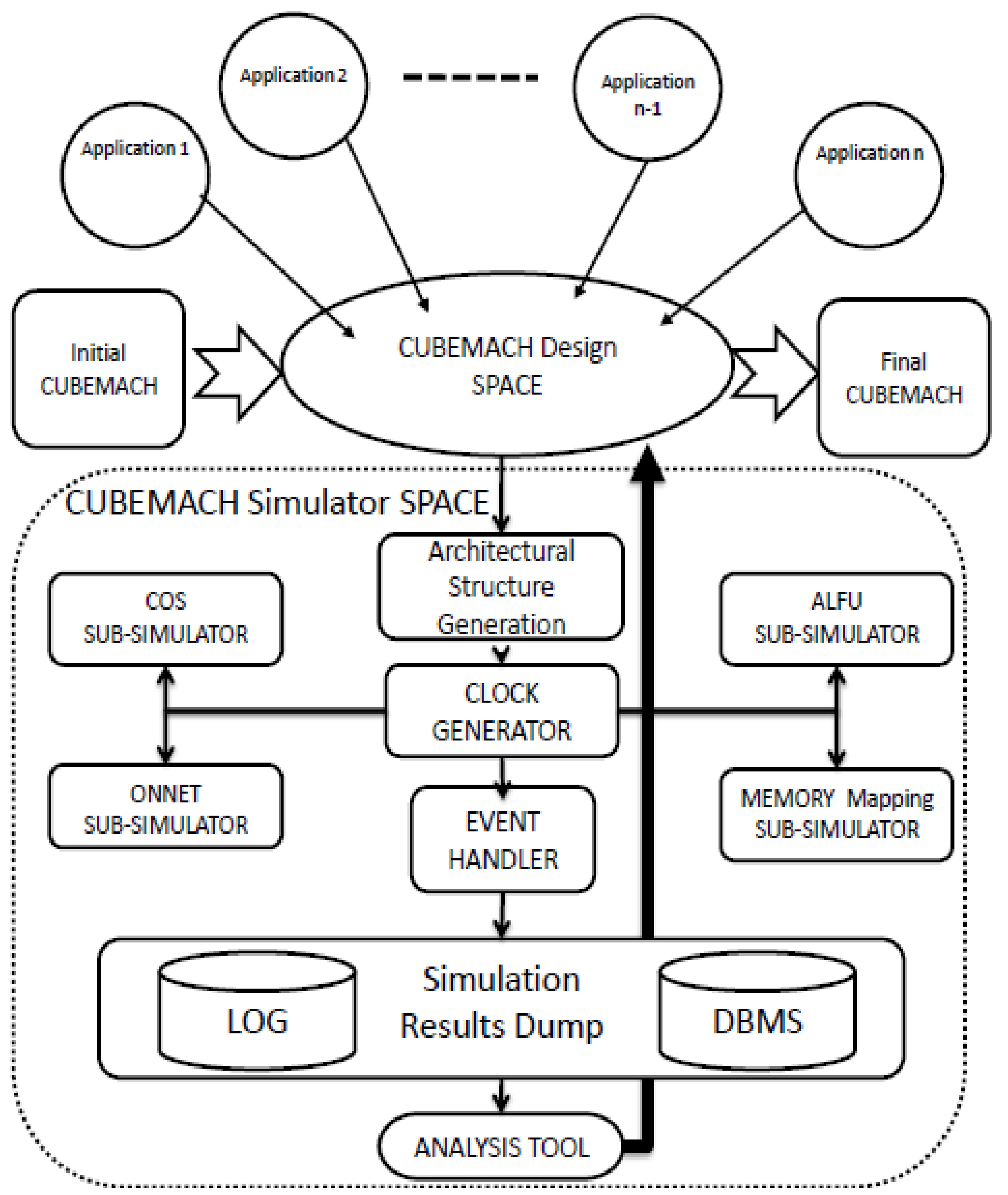
CUBEMACH Simulator

- pThread based Simulator
- Evaluates candidate CUBEMACH Architecture
- Feed results to CUBEMACH Optimizer
- CUBEMACH Optimization Engine (COE) produces Optimized Architecture
- Simulation & Optimization : An iterative process
- Consists of

ALFU Sub-Simulator COS Sub-Simulator

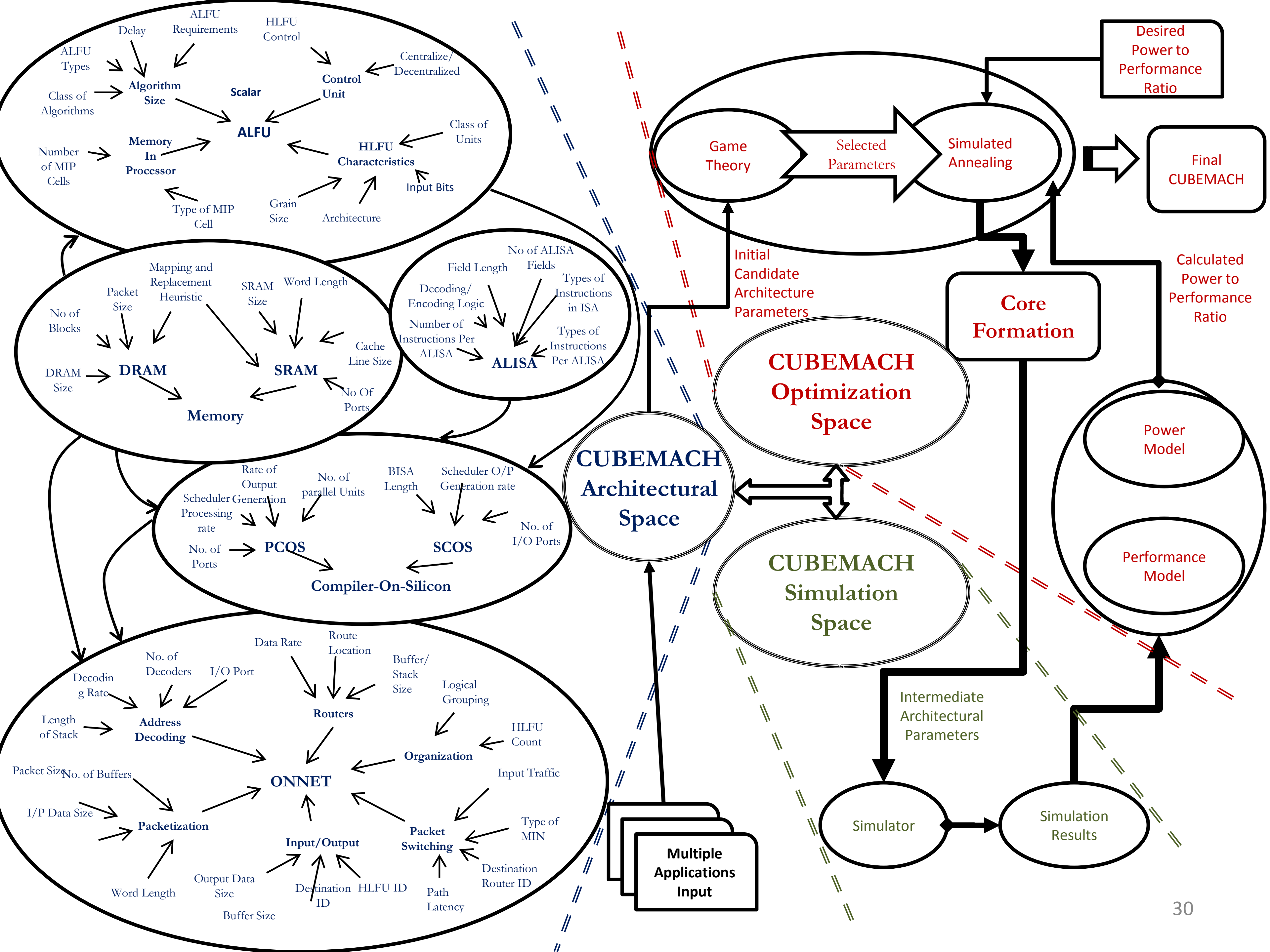
ONNET Sub-Simulator Memory Sub-Simulator

CUBEMACH Simulator



What we have seen . . .

Integrated CUBEMACH Design Paradigm …



ALFU Types
Delay
ALFU Requirements
HLFU Control
Centralize/ Decentralized
Class of Algorithms
Algorithm Size
Scalar
Control Unit
ALFU
Number of MIP Cells
Memory In Processor
HLFU Characteristics
Class of Units
Input Bits
Type of MIP Cell
Grain Size
Architecture

Packet Size
Mapping and Replacement Heuristic
SRAM Size
Word Length
No of Blocks
Cache Line Size
DRAM Size
DRAM
SRAM
No Of Ports
Memory

Field Length
No of ALISA Fields
Decoding/ Encoding Logic
Types of Instructions in ISA
Number of Instructions Per ALISA
Types of Instructions Per ALISA
ALISA

Rate of Output Generation
Scheduler Processing rate
No. of parallel Units
BISA Length
Scheduler O/P Generation rate
No. of I/O Ports
No. of Ports
PCQS
SCOS
Compiler-On-Silicon

Data Rate
Route Location
Buffer/ Stack Size
No. of Decoders
I/O Port
Decodin g Rate
Logical Grouping
Routers
Length of Stack
Address Decoding
HLFU Count
Organization
ONNET
Packet Size
No. of Buffers
Input Traffic
I/P Data Size
Packetization
Type of MIN
Packet Switching
Input/Output
Word Length
Output Data Size
Destination ID
HLFU ID
Path Latency
Destination Router ID
Buffer Size

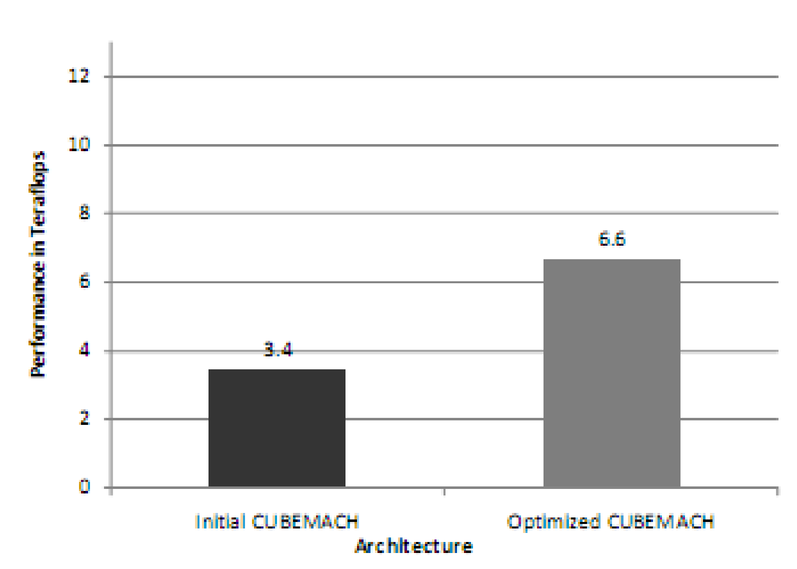
CUBEMACH Architectural Space

Multiple Applications Input

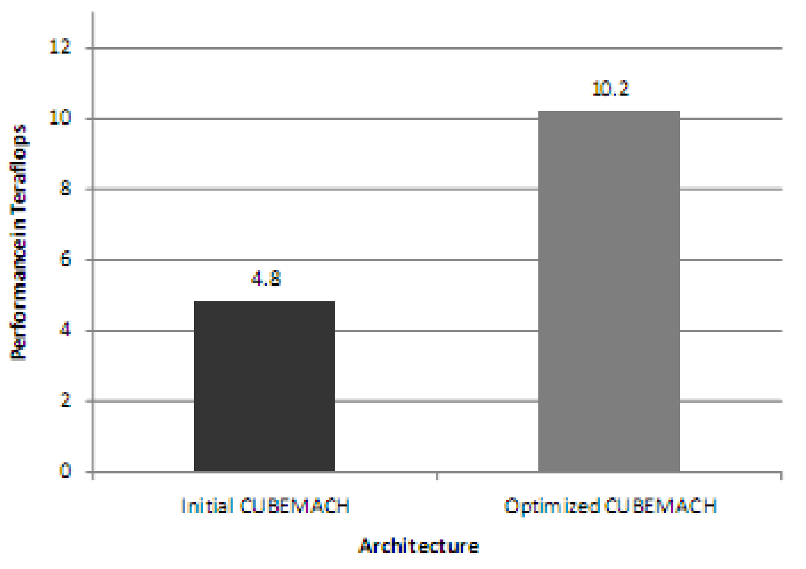
Desired Power to Performance Ratio
Game Theory
Selected Parameters
Simulated Annealing
Final CUBEMACH
Initial Candidate Architecture Parameters
Core Formation
Calculated Power to Performance Ratio
CUBEMACH Optimization Space
Power Model
Performance Model
CUBEMACH Simulation Space
Intermediate Architectural Parameters
Simulator
Simulation Results

Sample CUBEMACH Architecture :

Simulation Results



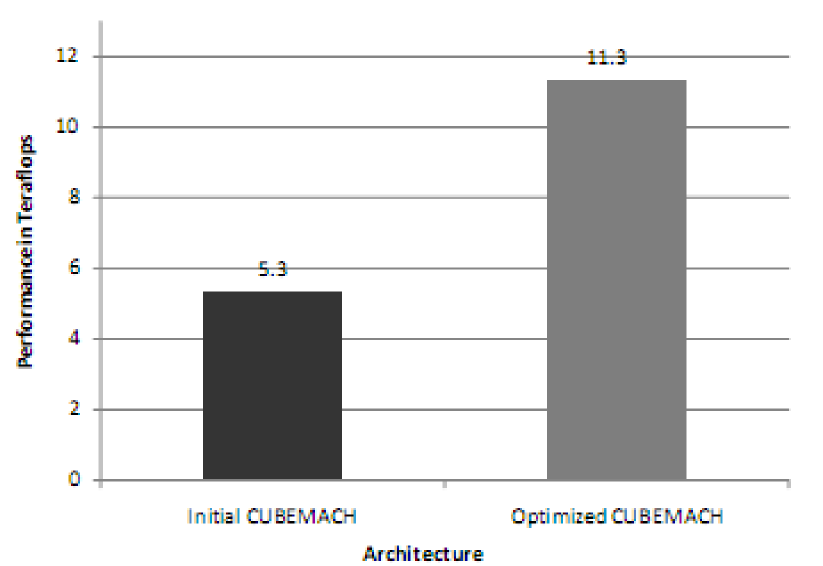
Matrix Based Algorithms



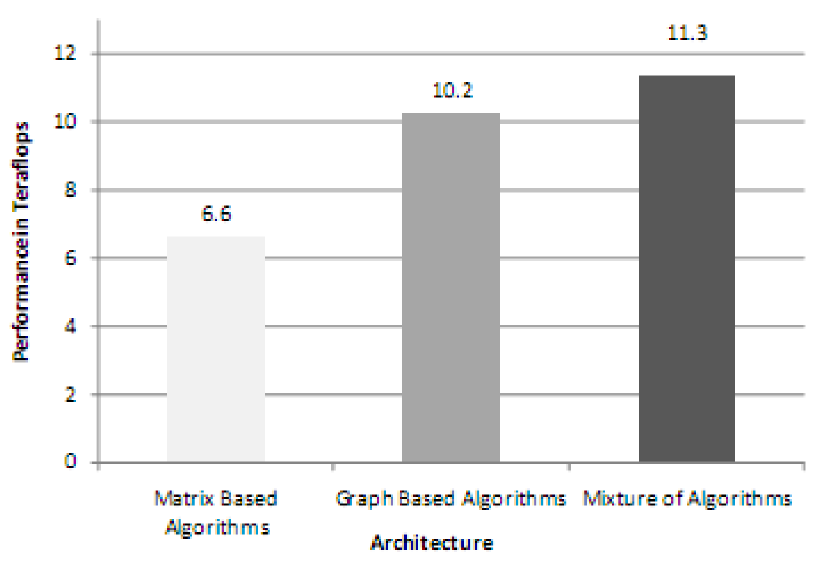
Graph Based Algorithms

Sample CUBEMACH Architecture :

Simulation Results



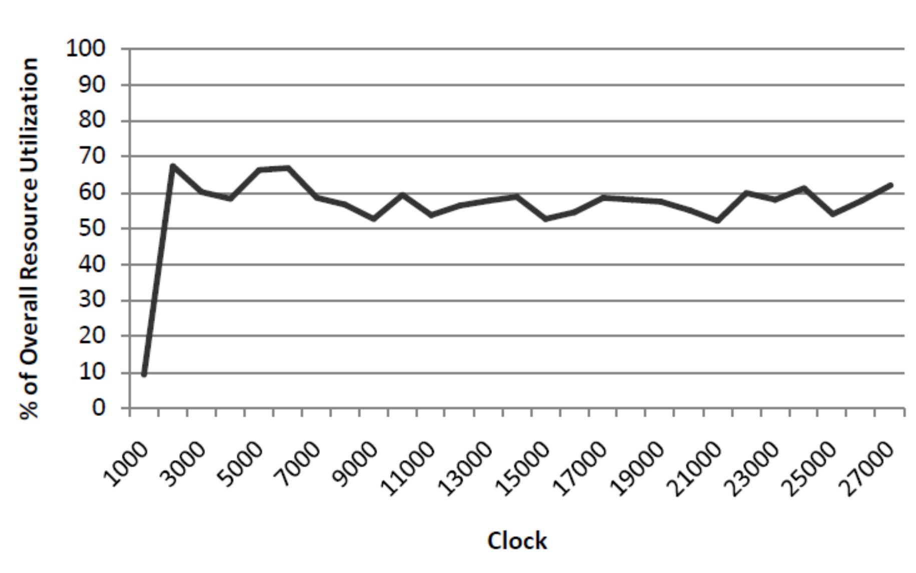
Mixture of Algorithms

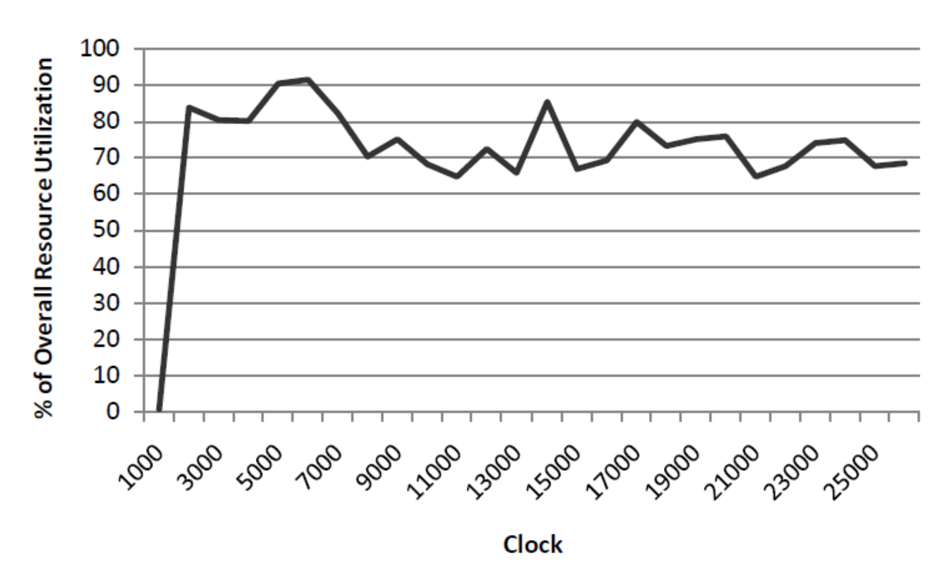


Comparison of Performance delivered by Optimized Architectures for corresponding types of Algorithms

Sample CUBEMACH Architecture :

Simulation Results



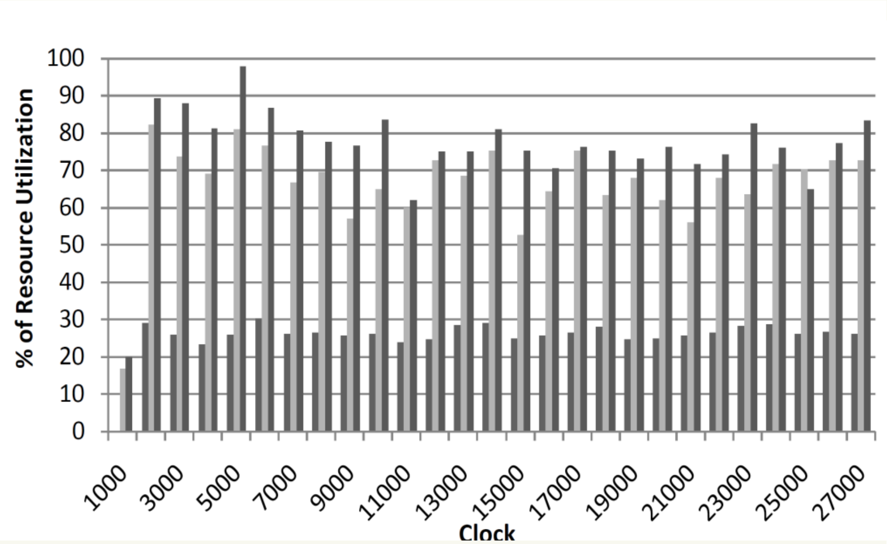


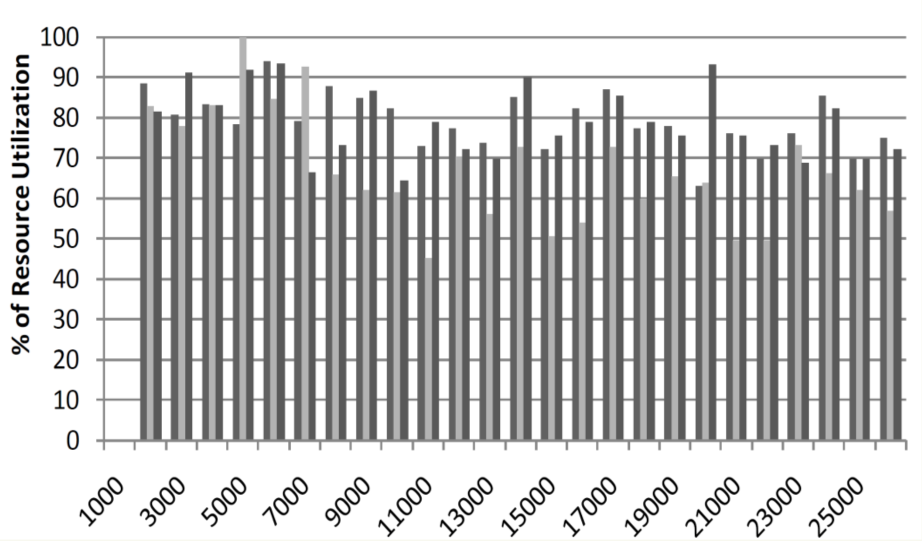
Overall Resource Utilization of :

(i) Initial CUBEMACH Architecture : Mean = 59 %

(ii) Optimized CUBEMACH Architecture : Mean = 74 %

Sample CUBEMACH Architecture : Simulation Results





In Initial Candidate CUBEMACH Architecture,

- •Matrix ALFUS – low usage
- •Scalar ALFUS – average usage
- •Graph ALFUS – high usage

In Optimized Candidate CUBEMACH Architecture,

- •Matrix ALFUS – high usage
- •Scalar ALFUS – high usage
- •Graph ALFUS – high usage

Conclusion

- Custom Built Heterogeneous Multi-Core Architectures (CUBEMACH) promises,
  - Increased Resource Utilization
  - Multiple application flavored architectures
  - Elimination of Space Time Sharing at the Quantum Level during Multiple Application Execution (without multiprogramming)
  - Manufacturing and Running Cost reduction

Thank You

Questions??

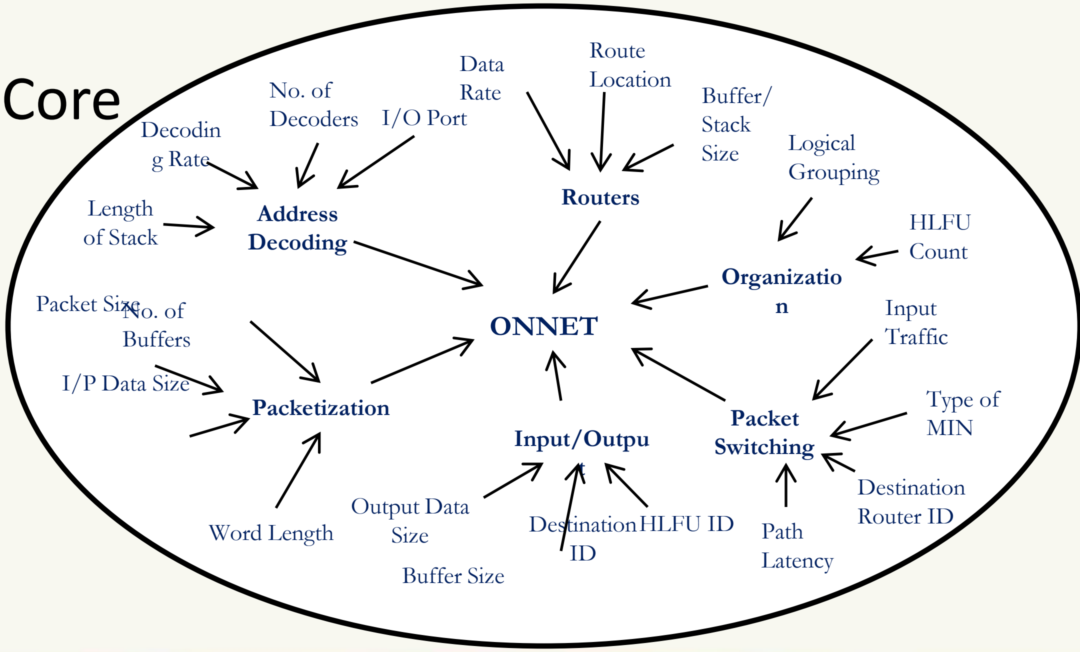
Customizable Compiler-On-Silicon

- What Compiler-On-Silicon?
- Why do we need Compiler-On-Silicon ?
- Why go for Customizable Compiler-On-Silicon ?

ONNET

Architecture uses -

- Multistage Interconnect Network
- Hardware Packetization Unit
- ONNET Design Space
  - H-Tree Structure within a Core
  - 2D Torus Across Cores
  - MIN Type



Data Rate

Route Location

No. of Decoders

I/O Port

Buffer/ Stack Size

Decodin g Rate

Logical Grouping

Routers

Address Decoding

Length of Stack

HLFU Count

Organizatio n

Packet Size

No. of Buffers

ONNET

Input Traffic

I/P Data Size

Packetization

Type of MIN

Packet Switching

Input/Outpu t

Destination Router ID

Word Length

Output Data Size

Destination ID

HLFU ID

Path Latency

Buffer Size

Architectural Design Space - CUBEMACH

- ALFU – Algorithm Level Functional Units
- BISA – Backbone Instruction Set Architecture
- COS – Compiler On Silicon
- ONNET – On Node Network
- Novel Cache Mapping Scheme
- SCOC IP Cores : Achieving cost effectiveness

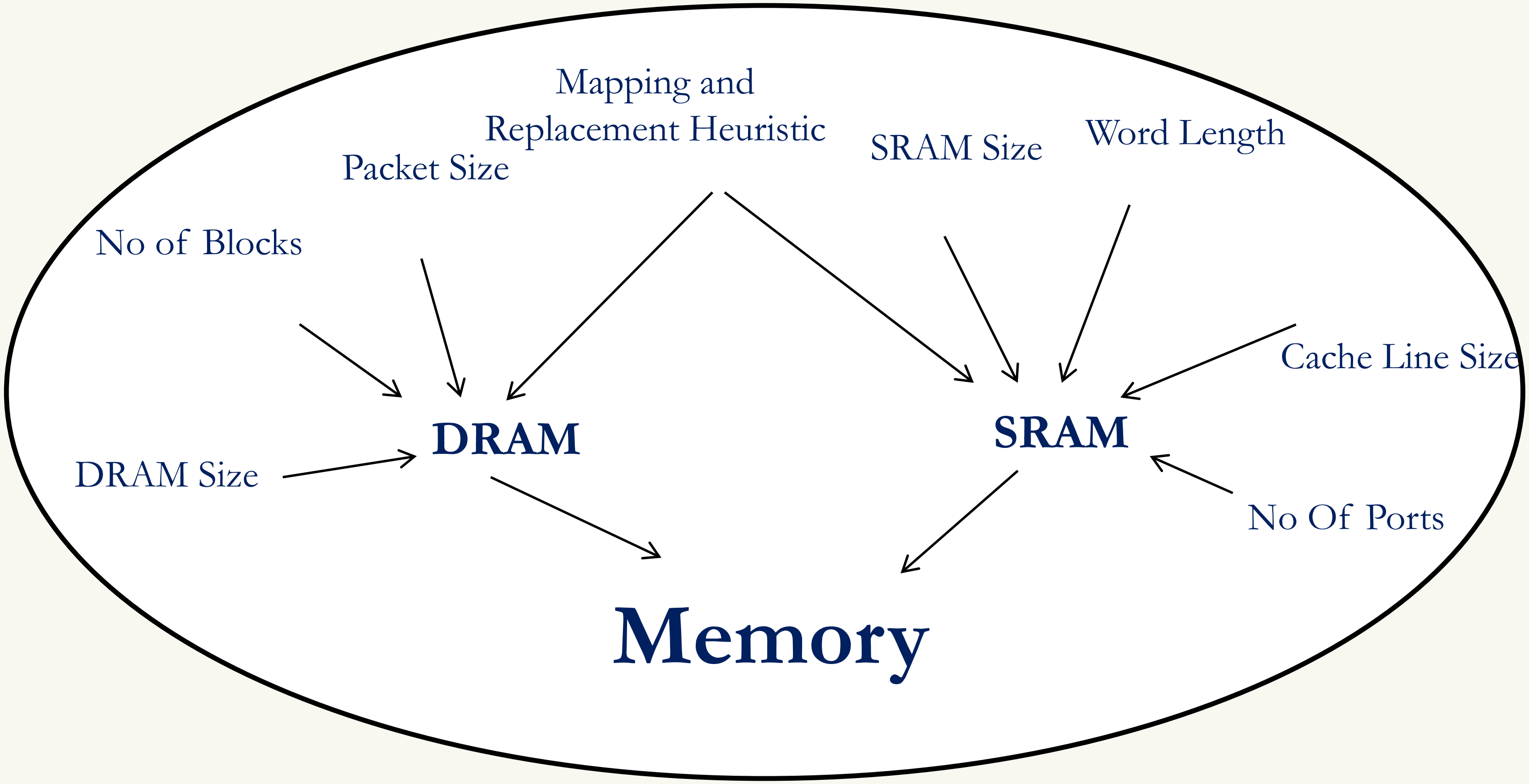
(Super Computer On Chip - IP Cores)

On Node Network Architecture

Features -

- Communication across heterogeneous multi-cores
- Data requirements of diverse ALFUs
- High bandwidth
- Scalable
- Hierarchical Network-On-Chip

Memory


Mapping and Replacement Heuristic
Packet Size
SRAM Size
Word Length
No of Blocks
Cache Line Size
DRAM
SRAM
DRAM Size
No Of Ports
Memory

Advantages of SCOC IP Cores

- Fully Customizable
- Greatly reduces Design-Turnaround-Time
- Physically Design Friendly
  - Constraints of Area, Power and Performance
- Constrained & Rigid Design Methodology