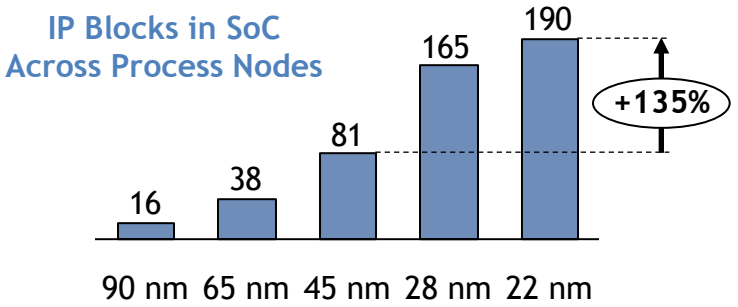
NetSpeed SYSTEMS

NetSpeed ORION:
A New Approach to Design On-chip Interconnects

August 26th, 2013

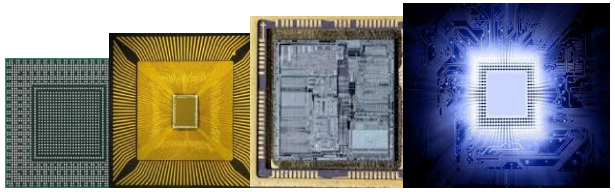
Interconnects Becoming Increasingly Important

- Growing number of IP cores
  - Average SoCs today have 100+ IPs
  - Mixing and matching of IP cores

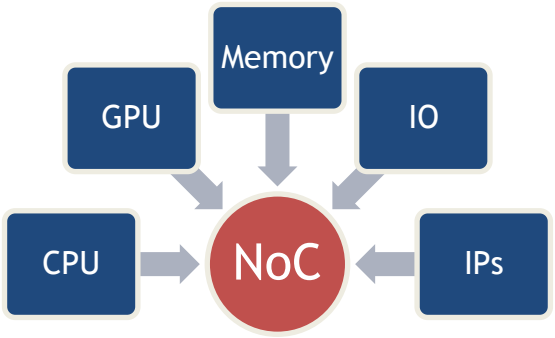


Source: International Business Strategies, 2012

- Increasing design complexity
  - Higher performance
  - Complex traffic, QoS and deadlock
  - Power Management



- Increasing off-the-shelf IPs use
  - Most SoC components available as standard IPs
  - Designing SoC mean designing the interconnect
  - Interconnect affects schedule 50% of the time



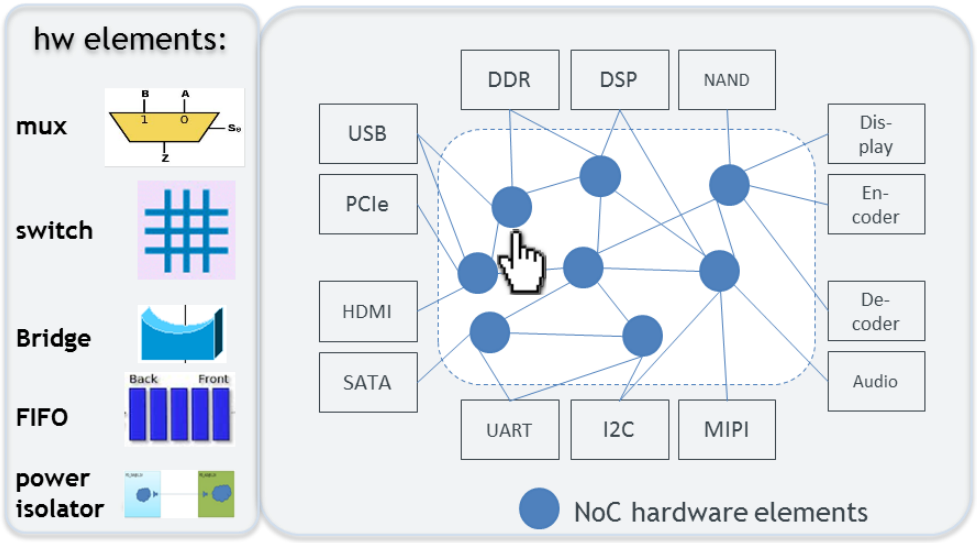
NoC Design Solutions

- Key NoC design elements and challenges are:

1. Architecture	2. Algorithms	3. Hardware
• Topol • Traffic • Route	• QoS • Deadlock • Load b	• Switch, channels • Protocol bridge • Power/clock support

- Existing commercial NoC solutions: Focused on hardware

Visio like GUI and NoC design



Key Design Challenges Not Addressed

1. What is the right NoC topology?
2. How many NoC layers and channels are needed to satisfy bandwidth?
3. Is the NoC deadlock free?
4. Are end-to-end QoS requirements satisfied?
5. Is the NoC efficient? Are NoC channels and buffers optimized?

Introducing NetSpeed Orion

Architecture Automation

- Automatically determine most efficient topology
- Automatically optimize NoC channels and buffers
- Automatically map SoC traffic to NoC channels using high level SoC traffic specifications
- Fine grained control to architect after optimizations

State-of-the-art Algorithms

- End-to-end QoS
- Network and Protocol level deadlock free
- Placement aware SoC IP core to NoC mapping

Efficient

- Physically aware NoC design
- Adaptable and scalable architecture
- Optimal latency and power consumption
- Correct by construction and faster time to market

Design Flow: Specify, (Optimize) & Generate

① Specify

1. Number of components
2. Traffic flow
3. Floorplan
4. Bandwidth & Latency
5. Power
6. QoS

② Optimize

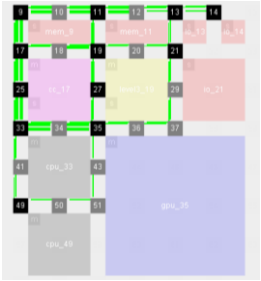
NetSpeed NocStudio

Automatically create efficient NoC designs

Architecture Exploration
Evaluate and Optimize
Power Optimization

③ Generate

Physically-aware, customized Interconnect IP

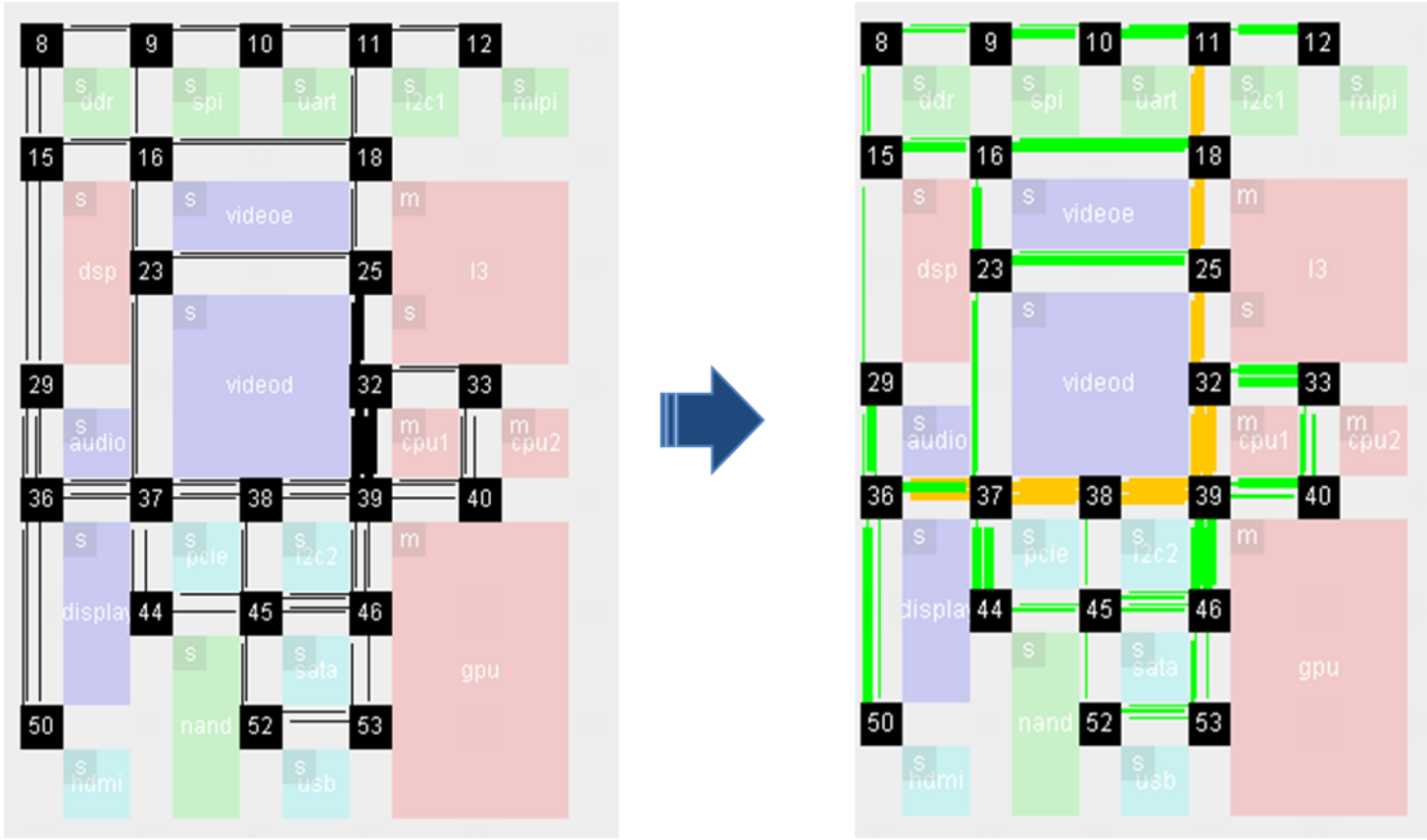


- C++ Models
- Performance Statistics
- Synthesizable RTL
- Verification Test benches

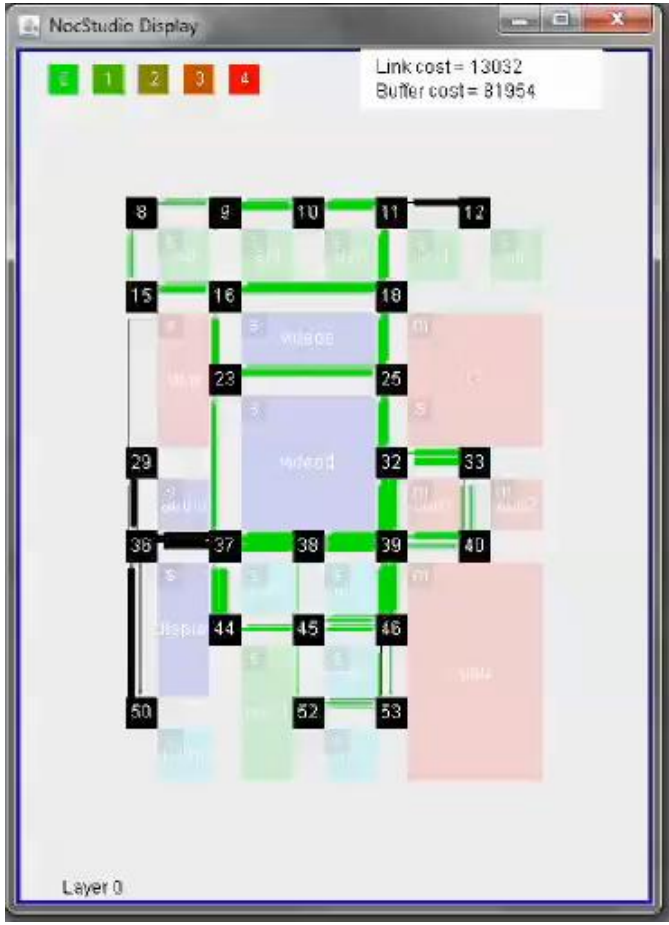
SPECIFY → OPTIMIZE → GENERATE

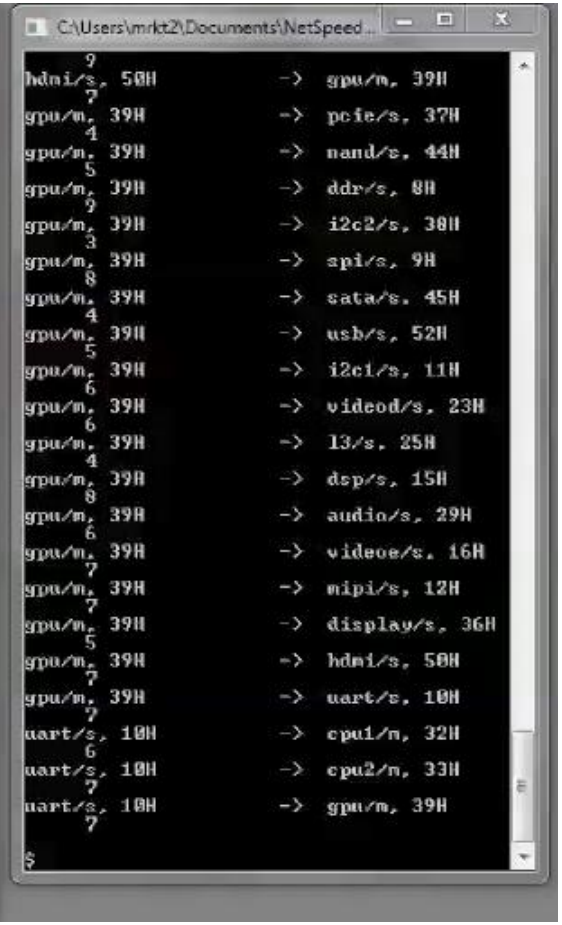
- Reference 2D grid is used to define the IP cores in SoC
  - Enables our solution to create physically aware interconnect
  - Automated and visual design and analysis of the interconnect
- Add SoC IPs
  - Rough SoC floorplan description
  - Position, shape, size, ports & protocols
- Add Traffic between components – creates baseline NoC design

- Algorithmic optimizations to meet system specifications (Bandwidth, latency, QoS)
  - Links & Channels: optimize bandwidth and QoS
  - Floorplan: Optimize placement of IP blocks
  - Topology: Final NoC can morph into bus, tree, mesh, other
  - Routes: Leverage path diversity between blocks

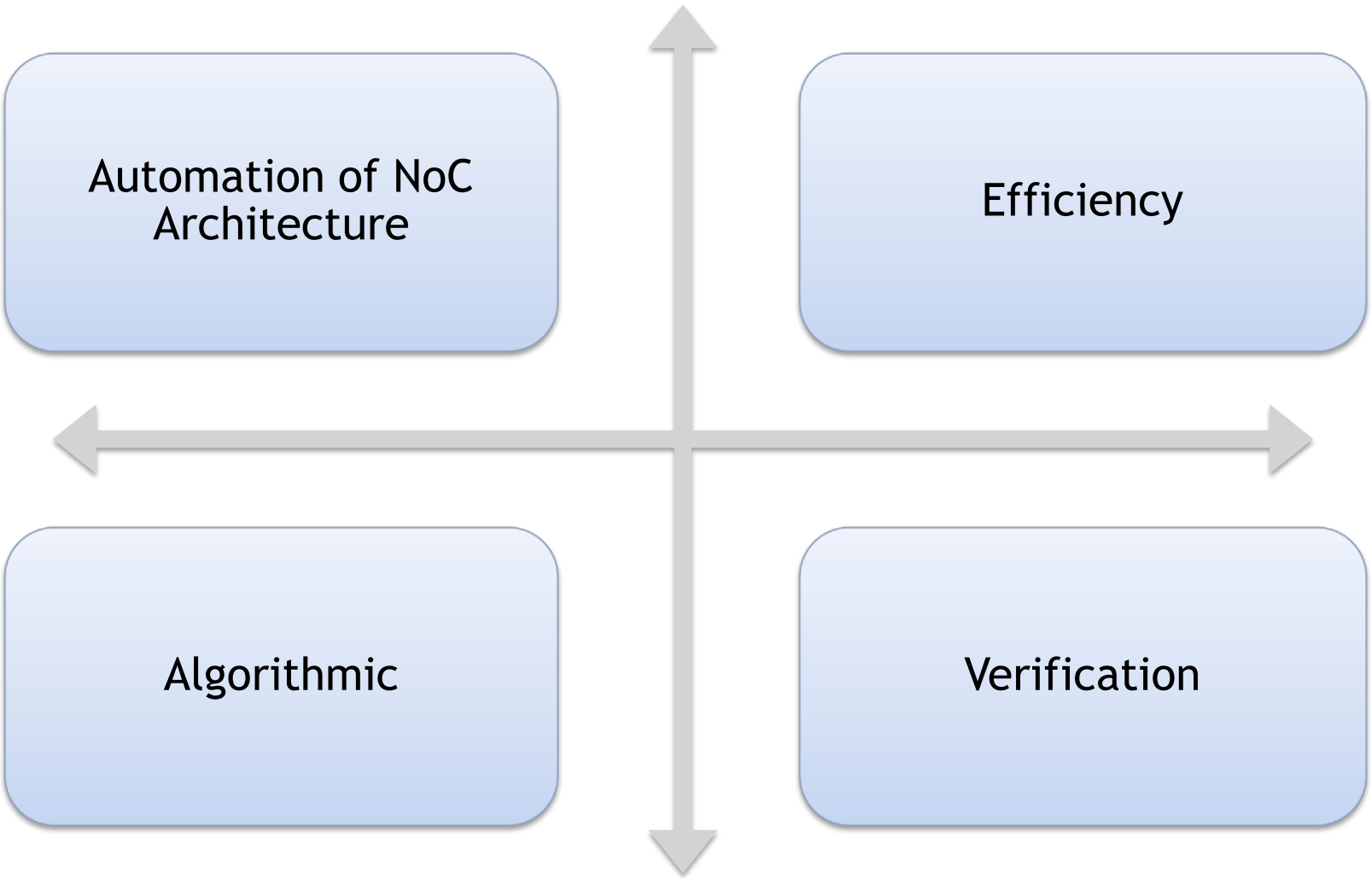


- Simulator to characterize performance of NoC
  - Performance details
    - Visual display of traffic congestion
    - Performance probes
    - Detailed statistics
- Synthesizable RTL
- Functional C++ model
- Verification test bench





Key Challenges and NetSpeed Innovation



Automated Design and Topology

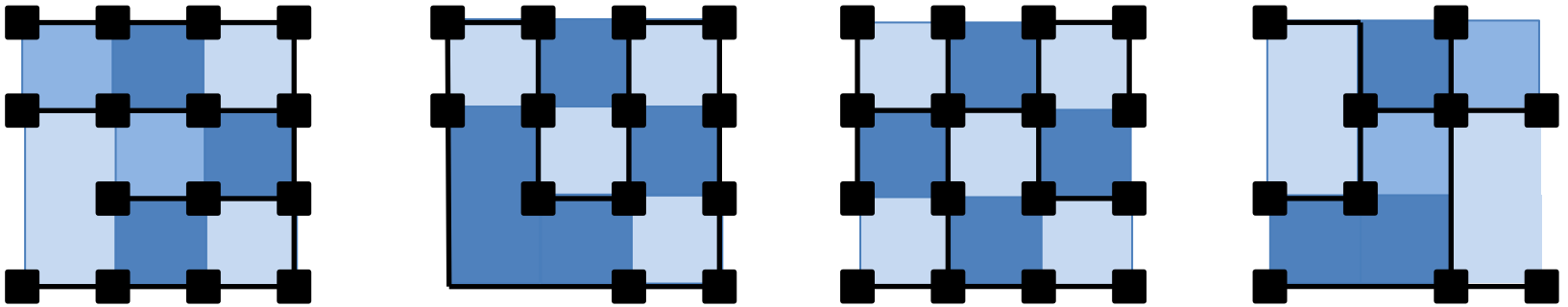
Which Topology

- A variety of standard topologies: Ring, Bus, Fat tree, Mesh
- What is the most optimal topology for a SoC family?
- Do we always have to use a standard textbook NoC topology?

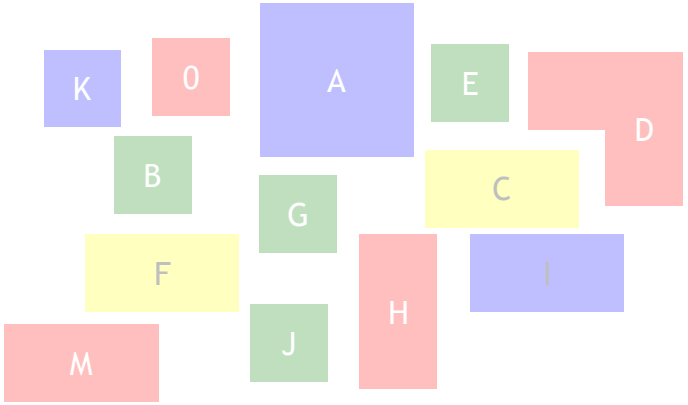
Is This Topology Efficient for my SoC

- Standard topologies may be inefficient for a SoC
- Would a topology satisfy the bandwidth
- Would it provide optimal latency

- Machine learning algorithms to automatically determine most optimal topology
  - Adaptive topology for any given connectivity
  - Fully heterogeneous in channel/buffer sizes
  - SoC Floorplan aware

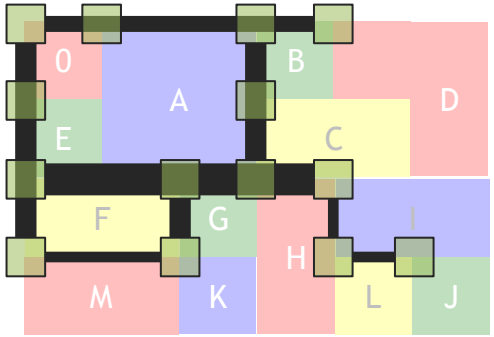


Physically Aware, Efficient, Correct by Construction



Optimize floorplan
Create Optimal NoC

Predictable area, frequency
Meet floorplan constraints

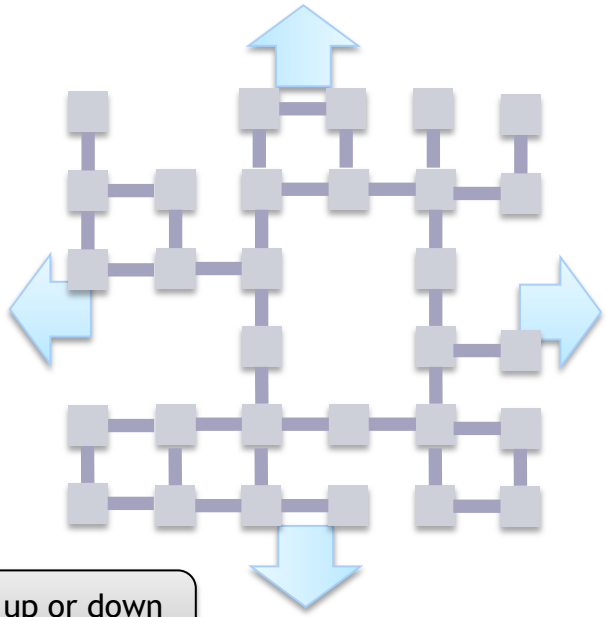


1. What is the most optimal floorplan?
2. Given a Floorplan, Design NoC that is correct by construction

- Use heuristic algorithms – simulated annealing
- Use machine learning and graph theory for efficient design

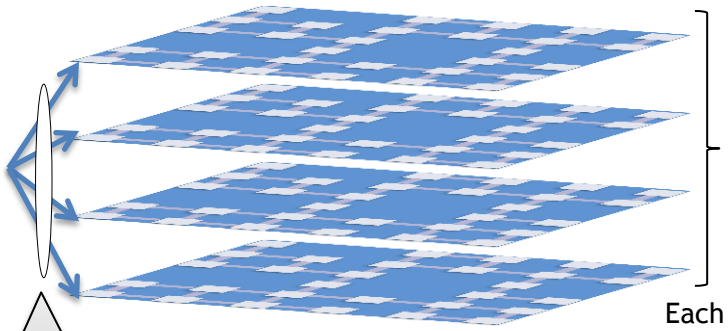
Automated And Efficient Scaling

- In addition to Flexibility, heterogeneity and correctness
- Scales efficiently
  - Scale with single NoC layer
  - Scale with multiple NoC layers



Scale up or down the network size

Multiple Physical and Virtual networks

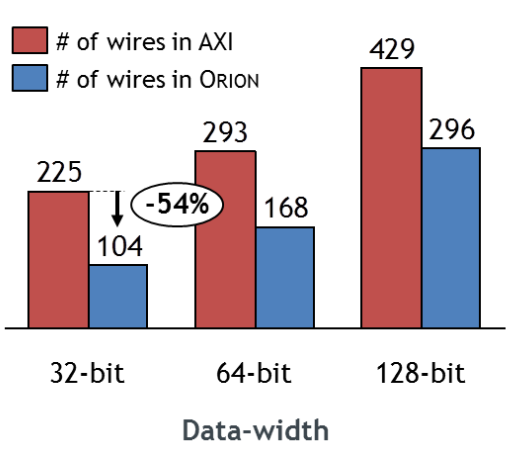


Each layer is independently optimized

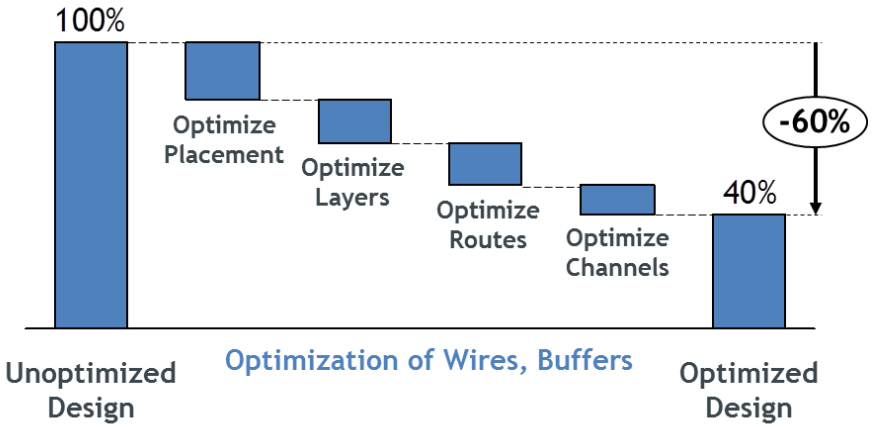
Optimize bi-section bandwidth

Quantifying Efficiency

Baseline NoC



Power and Area Optimizations

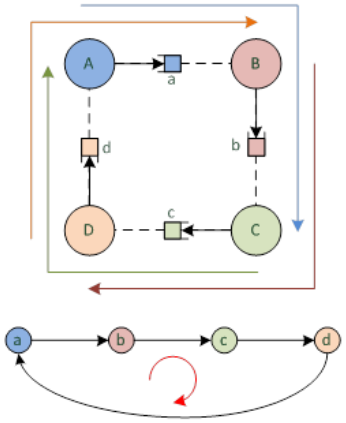


Synthesis Results

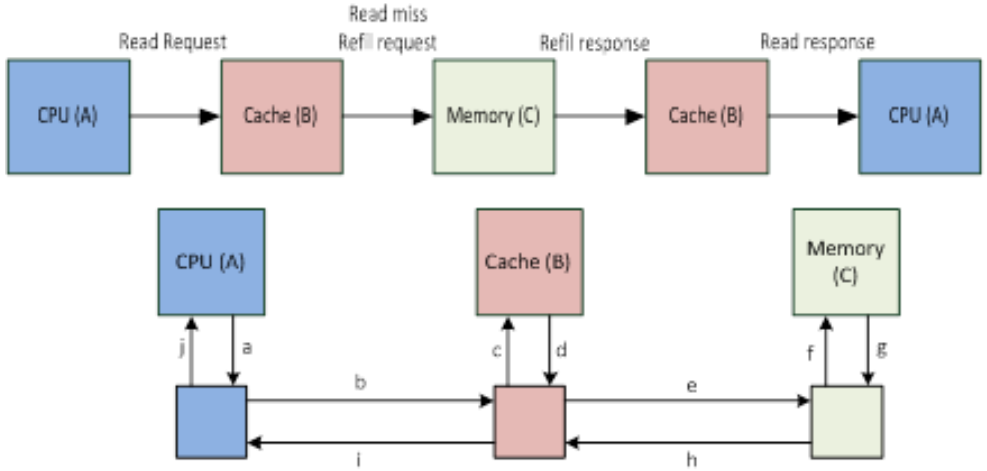
Router configuration	Area (sq. um)	Freq. (GHz)	Leakage Power (mW)
32bit data, 4 VC, 5 ports	59952	1.984	0.126
64bit data, 2 VC, 5 ports	41201	1.984	0.1718
64bit data, 4 VC, 5 ports	90252	1.888	0.183
128bit data, 4 VC, 5 ports	142576	1.872	0.3
256bit data, 2 VC, 5 ports	120075	1.984	0.54
256bit data, 4 VC, 5 ports	162890	1.728	0.5647

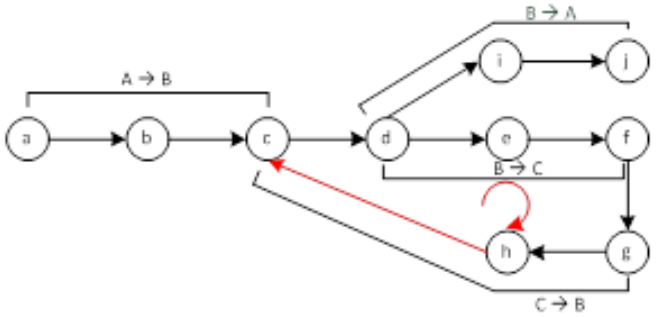
ALGORITHMIC: DEADLOCK AVOIDANCE

Network-Level Deadlock



Protocol-Level Deadlock : Memory Subsystem





- Challenges:
  - Irregular topology
  - Complex system traffic
    - Inter-dependent messages
    - Multiple protocols in an SoC
    - Complex ordering requirements
- Solution
  - NetSpeed ORION uses graph theory algorithms and formal techniques
    - Correct by construction
    - NoC design remains efficient
    - robust (can handle complex irregular topologies and routing, etc.)

ALGORITHMIC: QOS

ALGORITHMIC

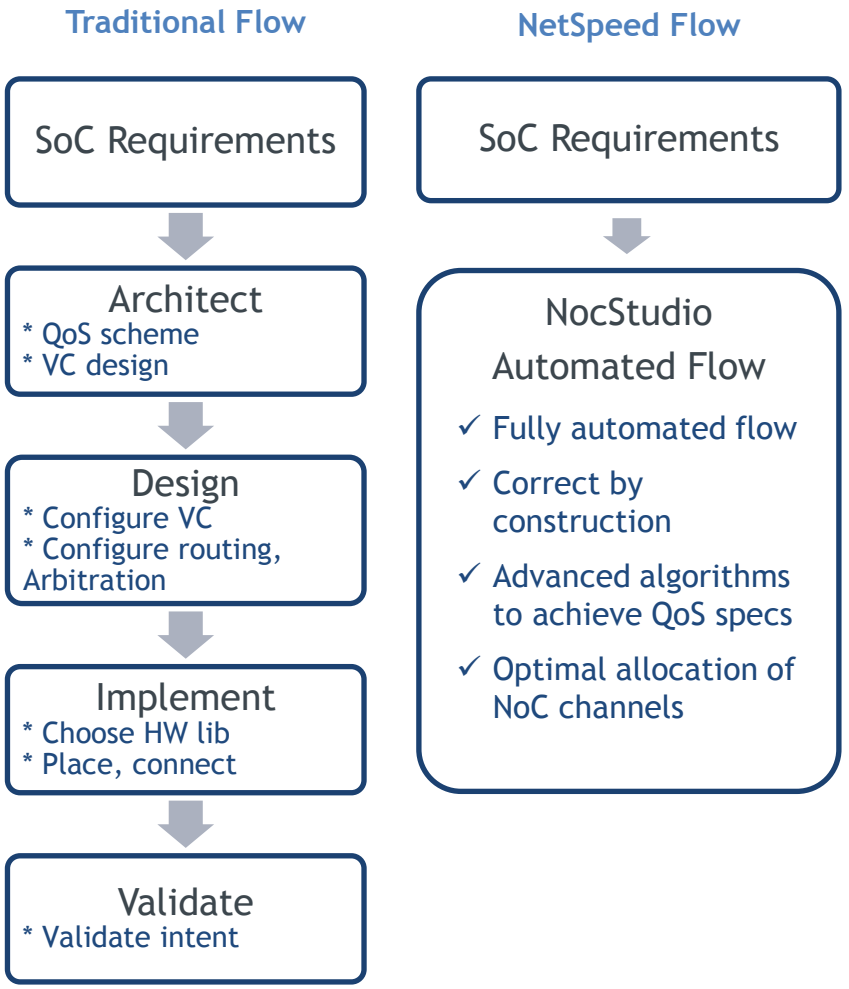
- Ideal QoS scheme characteristics:
  - Fairness (Strict & weighted)
  - Dynamic (Adjusts allocation among active agents)
  - Work-conserving (High agents utilization)
  - Low-cost (HW cost of implementation)

Existing QoS schemes	Limitation
Limit outstanding messages	Wastes resource bandwidth
Rate limit sources	Wastes resource bandwidth
Weight-based arbitration	Unfair in deep networks
Token passing	Unfair and slow response
Age-based arbitration	Unfair in deep networks

- **NetSpeed Solution:**
  - **Distributed QoS algorithms for robust, end-to-end QoS with a low cost implementation**

Sophisticated, Efficient and Fair

AUTOMATED



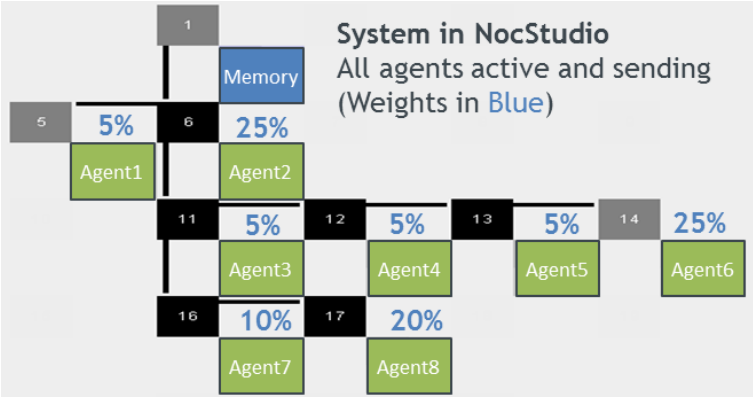
QoS: NoC RTL Simulation Results

NetSpeed SYSTEMS

Different Weights

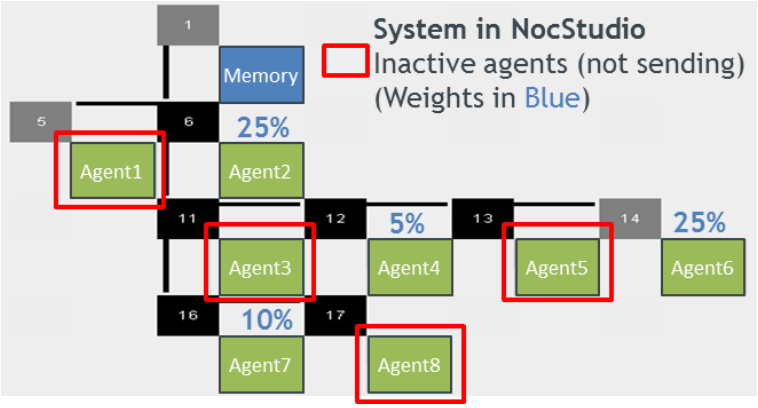
Performance and Fairness Summary:

Agents	1	2	3	4	5	6	7	8
% BW Target	5	25	5	5	5	25	10	20
% BW Actual	5.1	25.8	5.1	4.9	4.9	24.2	10.0	19.8



Different Weights (Inactive Agents)

Agents	1	2	3	4	5	6	7	8
% BW Target	5	25	5	5	5	25	10	20
% BW Target (Normalized)	-	38.5	-	7.7	-	38.5	15.3	-
% BW Actual		39		7		38	15	

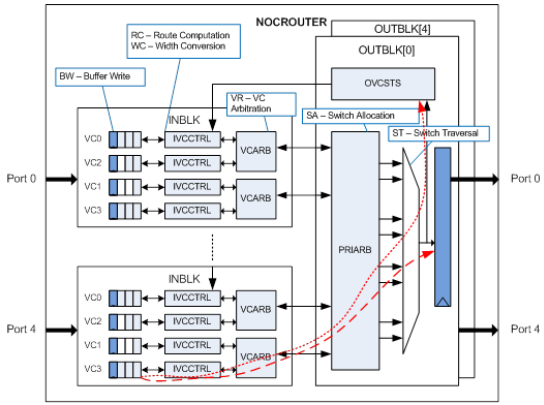


VERIFICATION: CHALLENGES & SOLUTION

- The challenge: How to verify given such flexibility?
- Solution: 3-tier approach to verification

1. Module Level Verification

- Highly parameterized design
- Random and directed tests
- Scoreboards for verifying QoS, arbitration & routing

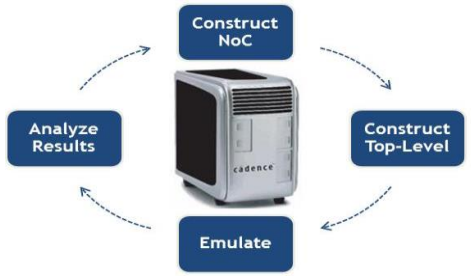


2. NoC Level Verification

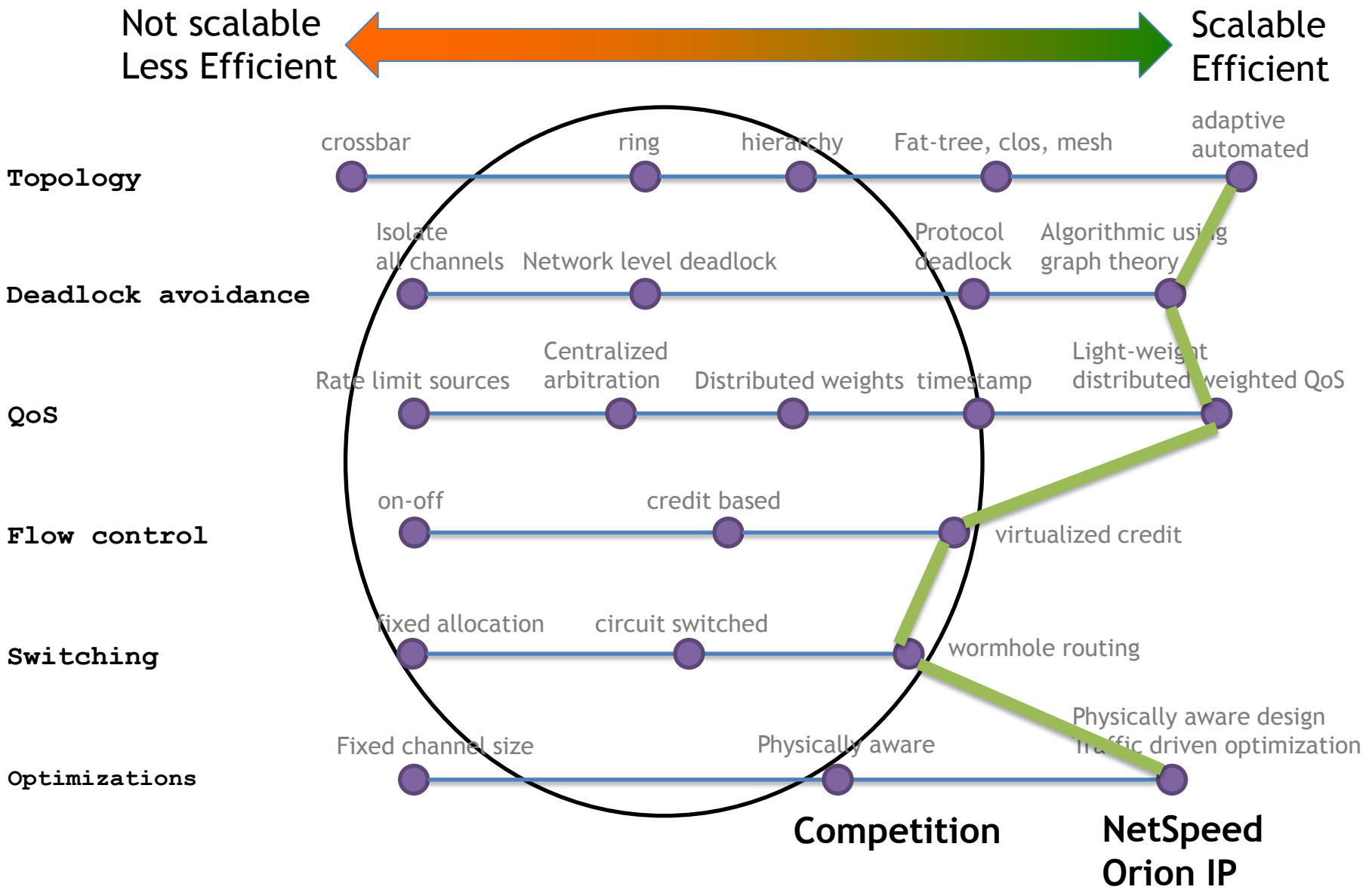
- Two parallel robust benches
- Thousands of random & directed NoC configurations
- Verification flow configurable through NocStudio
- Auto generated expect & predict UVM testbench
- Full randomization of stimulus & flow control
- Global debug monitor tracks every packet through the NoC

3. Emulation

- Successfully emulated multiple NoC configurations
- Full randomization of stimulus with end-to-end self-checking tests
- **100 Million packets through the NoC every night**



NetSpeed Orion vs. Competition



Concluding Remarks

Quantum Leap in SoC Interconnect Design

Automated

Efficient

Algorithmic

Coherent*

- Optimized Design
- Faster Time-to-market
- Higher Performance
- Lower Power

. . . Creates New Design Possibilities

* Fully coherent NoC coming in Q3 2013