Rethinking Automated Synthesis of MPSoC Architectures

Brett Meyer

Don Thomas

ECE Department

NSF-NGS Workshop 2007

Carnegie Mellon

MESH

Modeling Environment for Software and Hardware

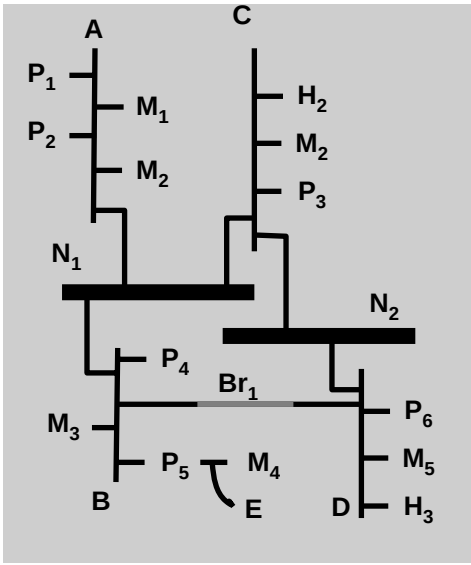
Acknowledgments

- Funding
  - NSF, General Motors Research, ST Microelectronics, Semiconductor Research Corporation (SRC)
- People involved in larger project
  - JoAnn Paul — now at Virginia Tech
  - Students — Alex Bobrek, Josh Pieper, Jeff Nelson

Single Chip Multi-core Embedded Systems

MESH

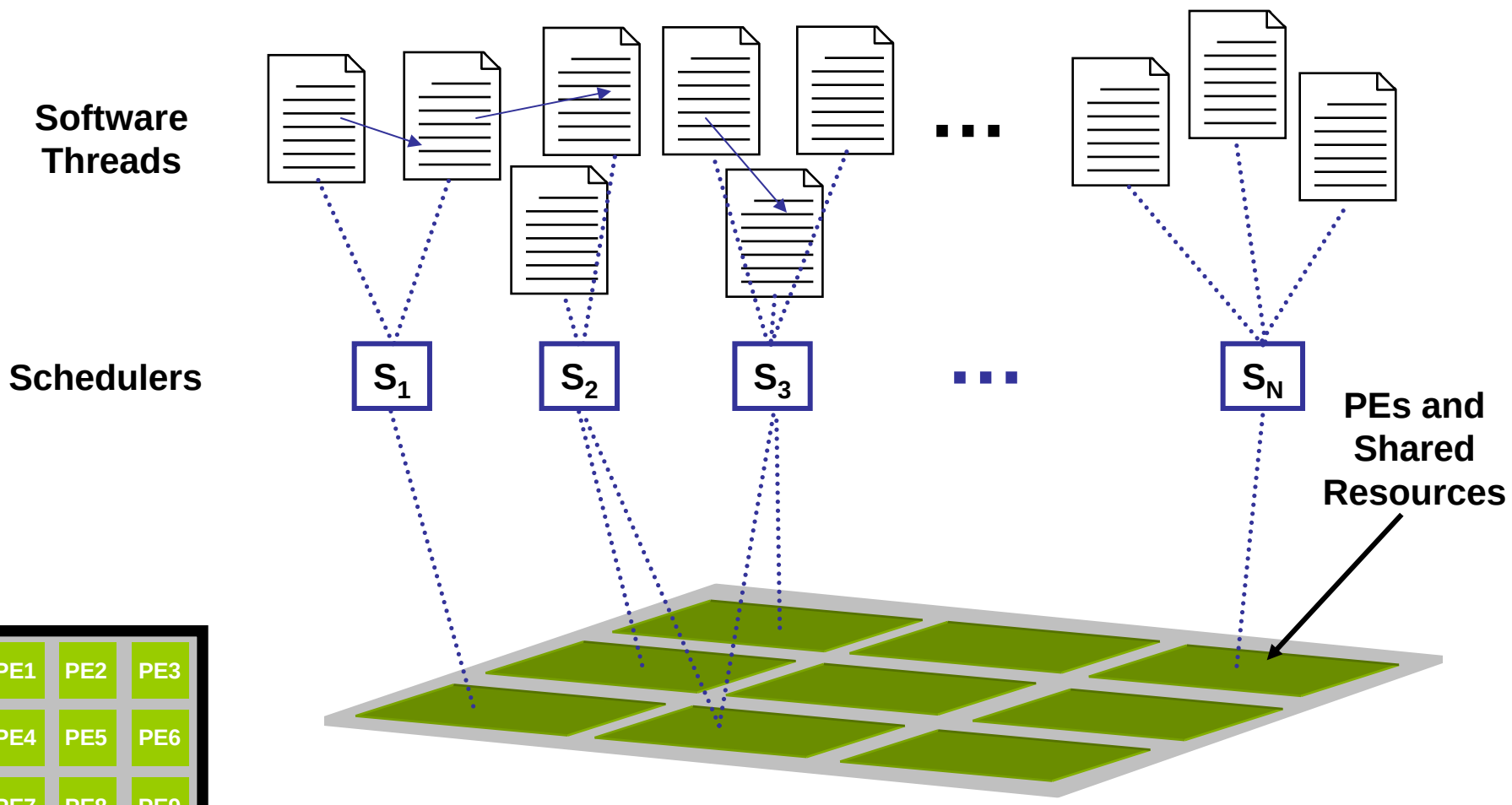
- The MESH Project
  - Modeling Environment for Software and Hardware
- Embedded Systems — custom design scenario
  - Design of single-chip heterogeneous multiprocessors
  - Tens of processing elements
  - Asking what is the best multiprocessor architecture for the system functionality
  - Leveraging knowledge of the application and their datasets for optimization
- Design elements — customize by selecting
  - Off-the-shelf processor cores for the PEs
  - Pre-designed (synthesizable) on-chip buses/networks
  - Off-the-shelf memories (shared)
  - Special hardware IP
  - Schedulers for PEs or groupings of PEs
  - Off-the-shelf and custom software
    - Real-time and best effort



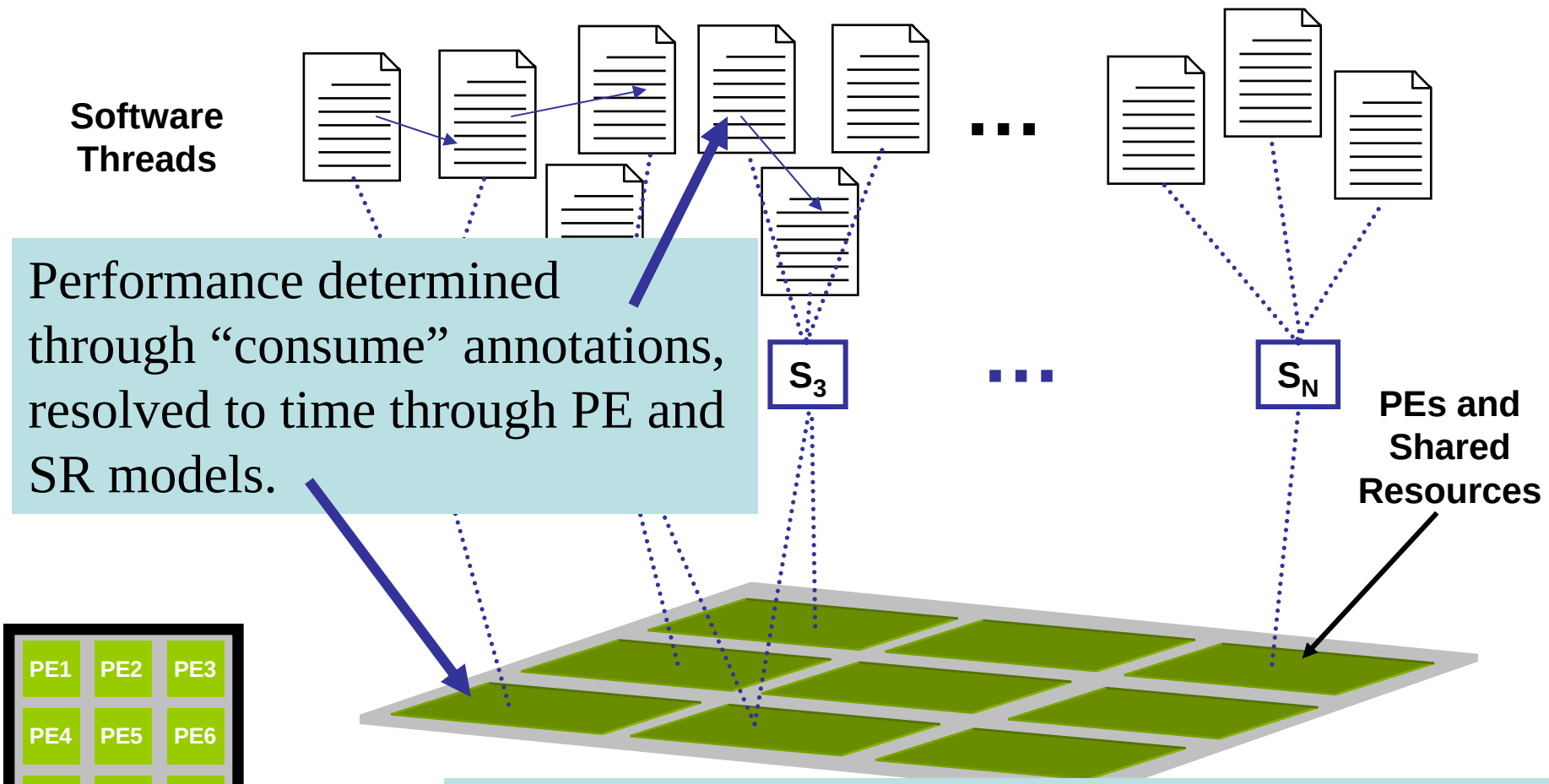
Needed: a design methodology to reduce design costs of these complex systems

Carnegie Mellon

Our Layered System Model



Performance Modeling

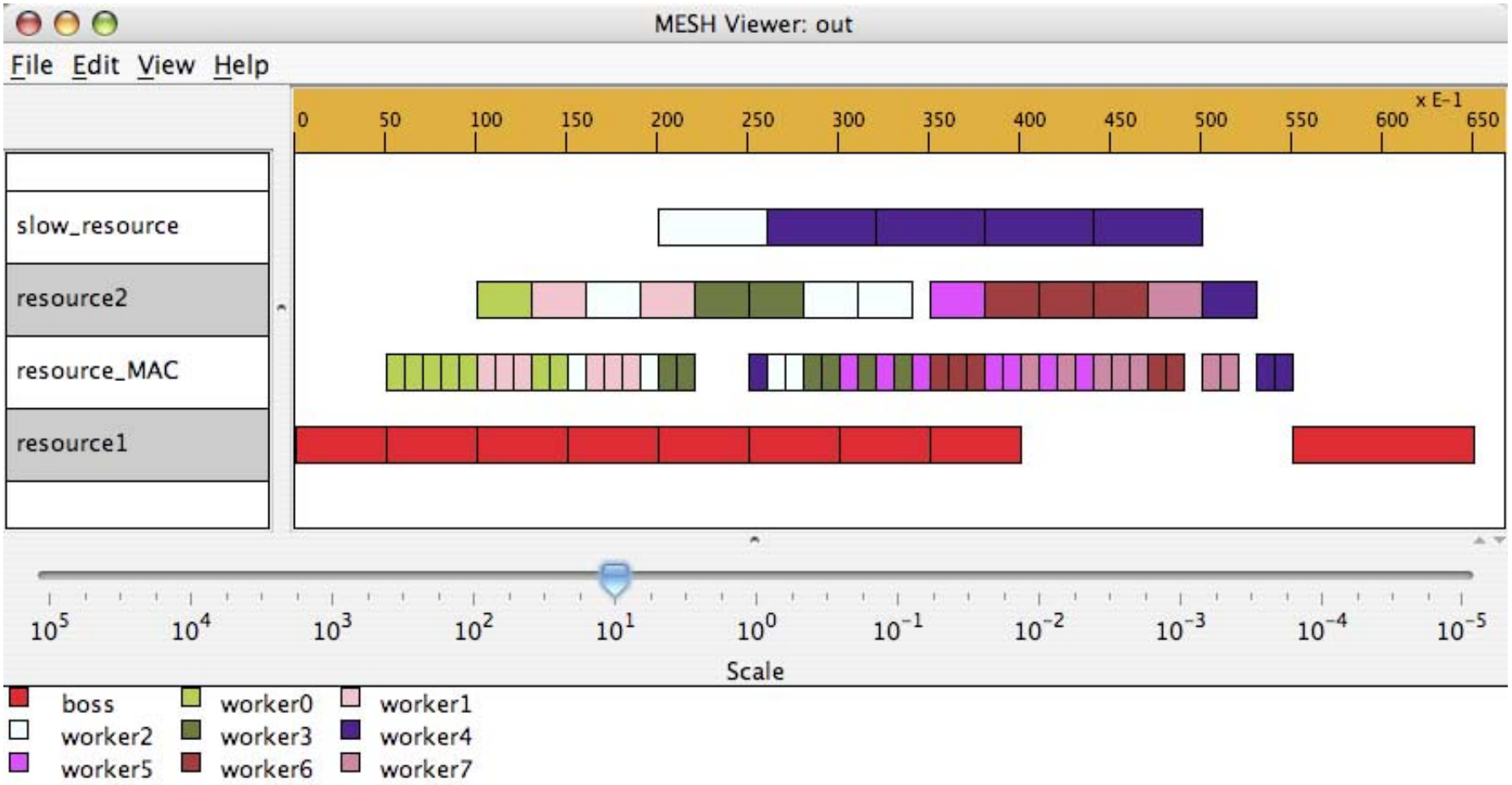


Performance determined through “consume” annotations, resolved to time through PE and SR models.

Annotation regions of ~300K cycles with 1-2% accuracy wrt cycle accurate simulation.

Carnegie Mellon

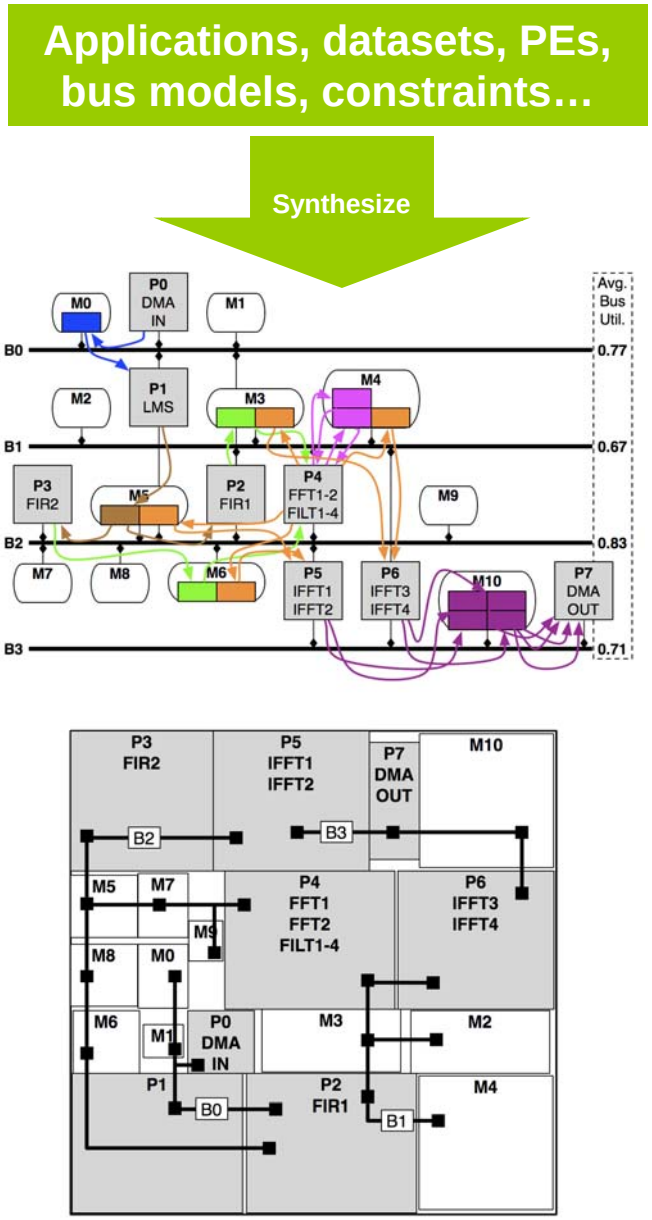
Simulation Results



Mesh Research Directions

MESH

- Model, simulate and synthesize
- Research topics
  - Performance simulation
  - Synthesis through to floorplanning
  - Designing robust systems in light of deep submicron wearout mechanisms
  - What should the language be

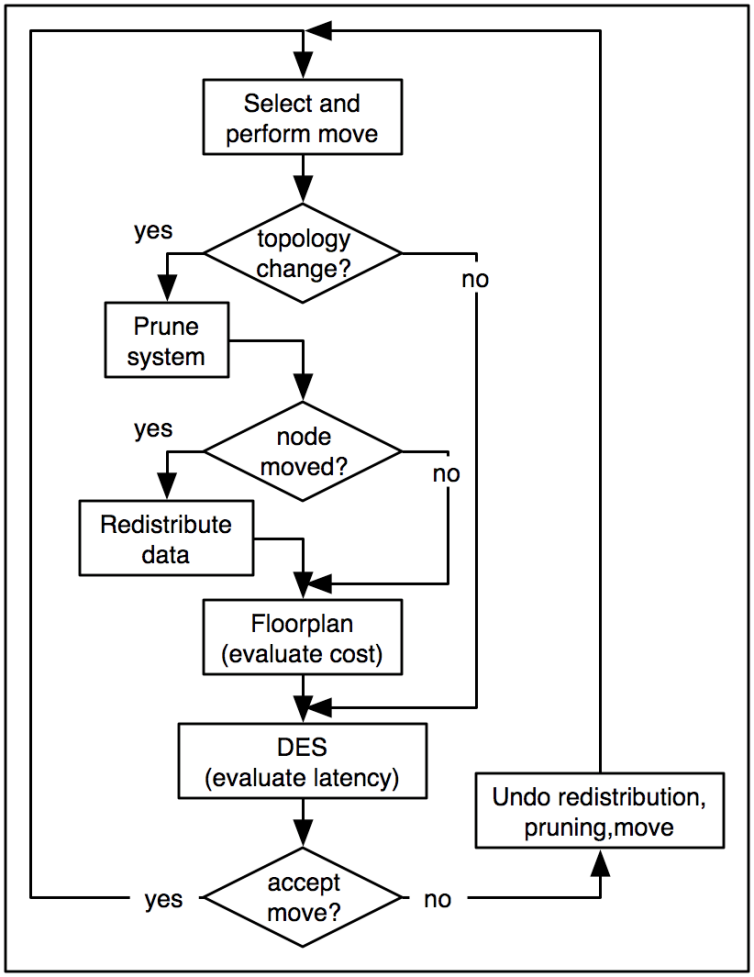


Carnegie Mellon

Synthesis Methodology

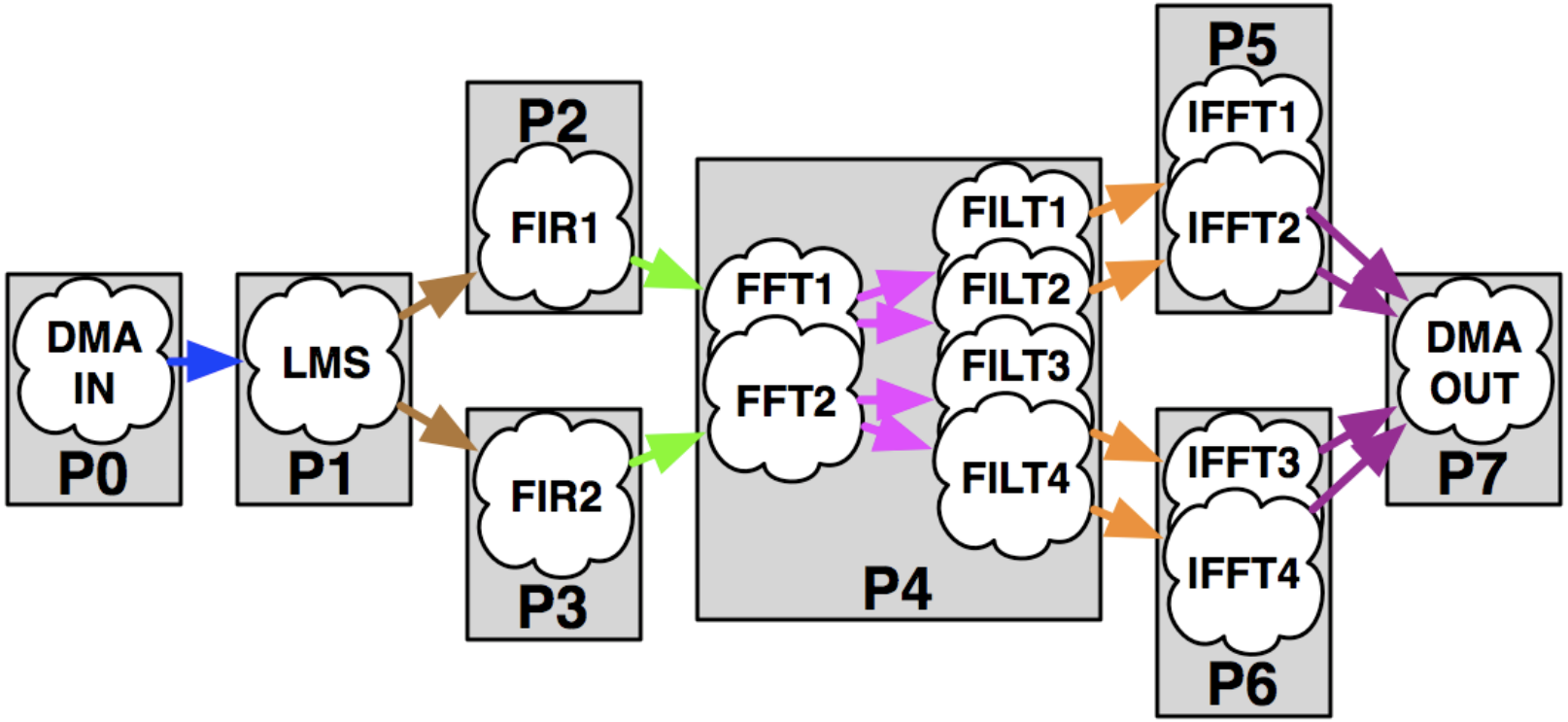
MESH

- System synthesis
  - Memory allocation
    - Number and location
  - Data-array mapping to memories
  - Bus interconnect
    - Number and connections
- Augmented simulated annealing
- Guided by full system evaluation in loop
  - Performance
  - Floorplan



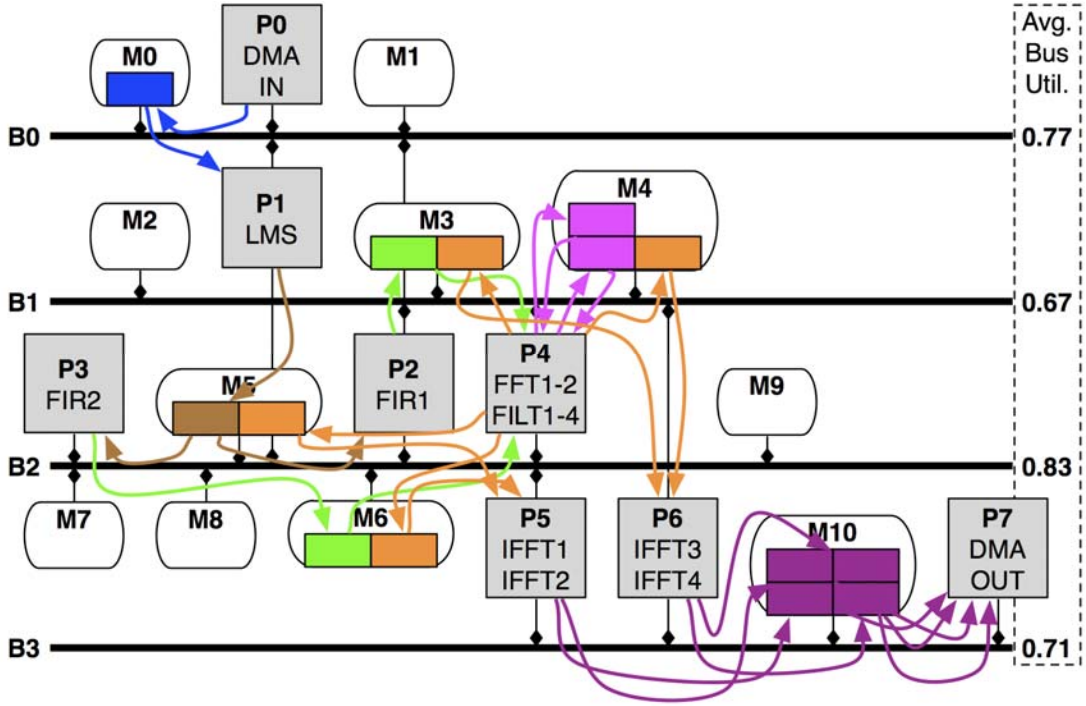
Carnegie Mellon

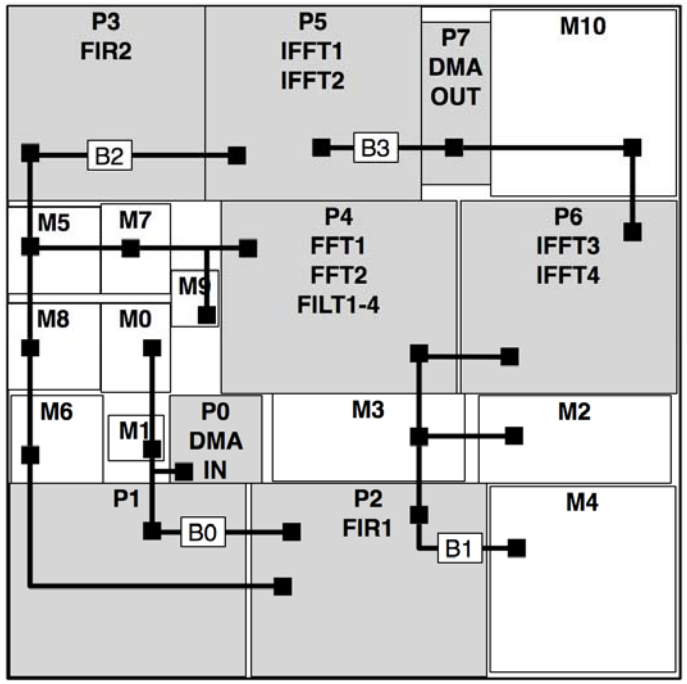
An Illustration



- ARM7s, DMA engines, SRAM, one bus type
- EXP 1: α to balance performance and cost
- EXP 2: Multiple α to find pareto-optimal points

An Example Design Point

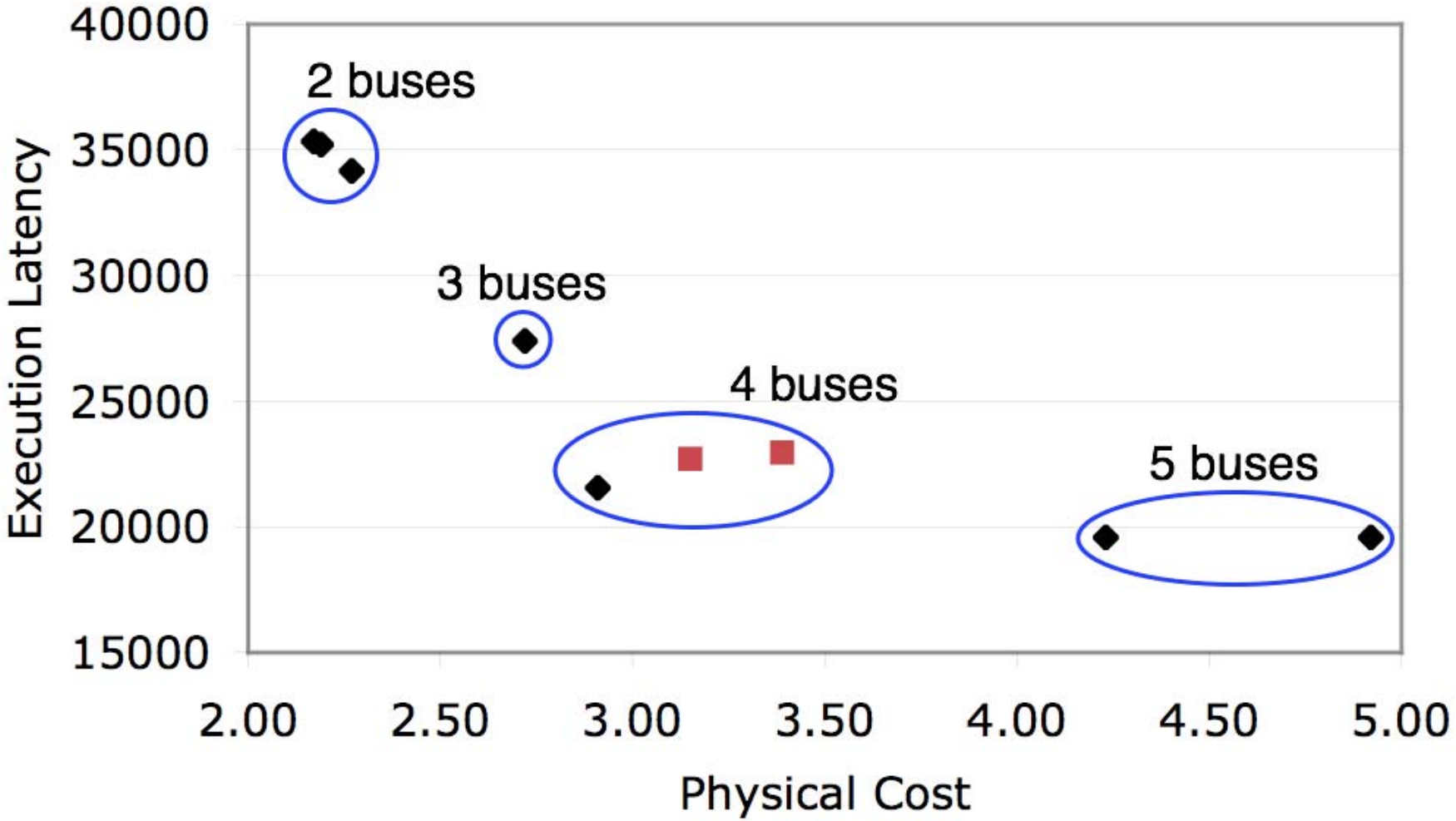




Balancing cost and performance balances partitioning and sharing

Pareto-Optimal Architectures

MESH



Carnegie Mellon

Insight into Design Methodology

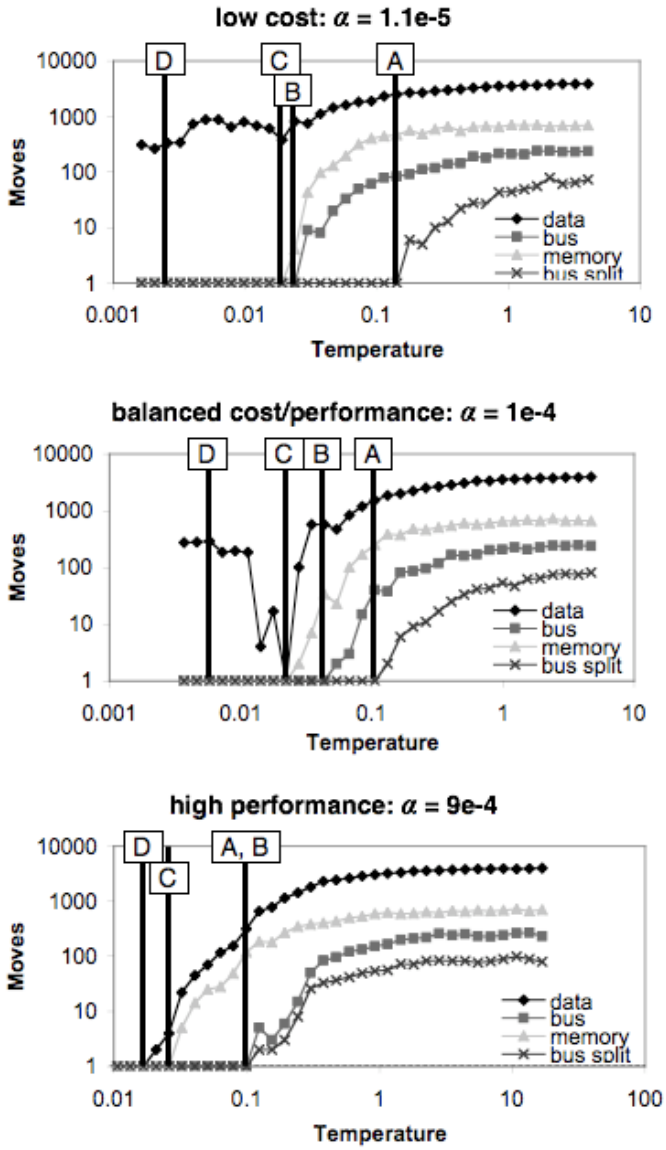
MESH

- We want insight into MPSoC design principles to better automate exploration
- We examine three design points
  - Low cost
  - Balanced cost/performance
  - High performance
- When optimizing multiple design aspects, is there a best practice?
  - Should some aspects of the design be fixed before others?

Move Acceptance Trends

MESH

- Key
  - A: bus splits end
  - B: bus moves end
  - C: memory moves end
  - D: system freezes
- Same progression occurs for all design points
  - Bus topology fixed first
  - Memory allocation fixed next
  - Data mapping fixed last



Carnegie Mellon

Discussion: Our Approach

MESH

- Compared to other approches, our's fixed bus topology ***first***, not last
  - Allocation is optimized for a given bus topology
- Optimization doesn't end with buses
  - Buses are not the only source of cost
  - System cost improves 24% after bus topology is fixed
- Experiment suggests an approach contrary to liturature

Future Work: Reliability

MESH

- Device reliability is a growing concern as devices shrink
  - Cost of manufacturing defects
  - Safe operation in the field
  - Adding a measure of robustness to our design process