

Advanced Topics in Multi-/Many-core Systems

Bruce Shriver
University of Tromsø, Norway

November 2010
Óbuda University, Hungary

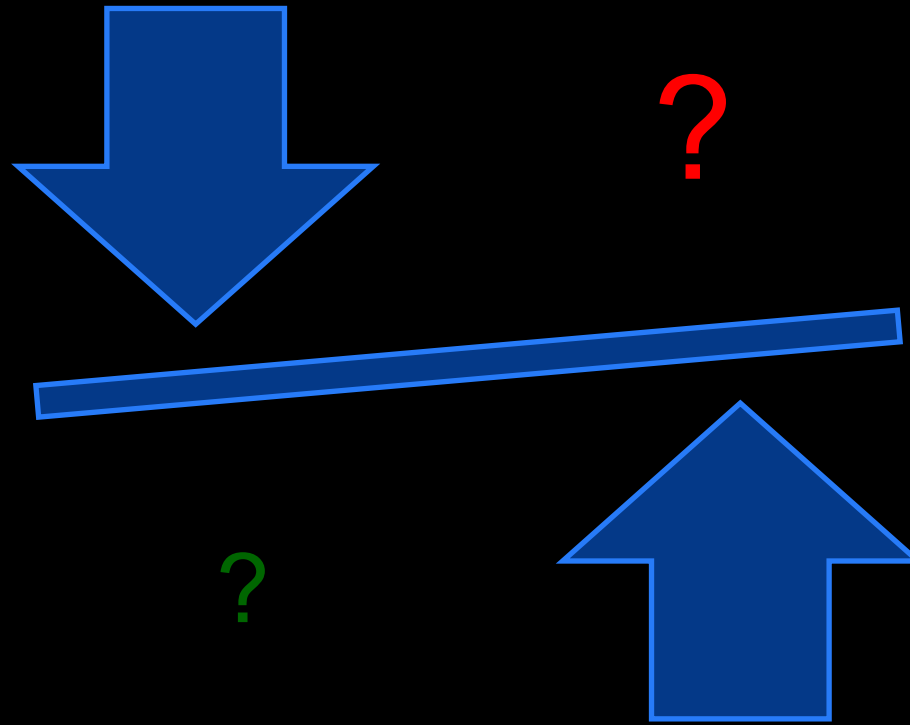
Set of Three Lectures

Reconfigurability Issues in Multi-core Systems

- The first lecture explores the thesis that reconfigurability is an integral design goal in multi-/many-core systems.

The Re-Design Imperative: Why Many-core Changes Everything

- The next two lectures explore the impact the multi-/many-core systems have on algorithms, programming language, compiler and operating system support and vice-versa.



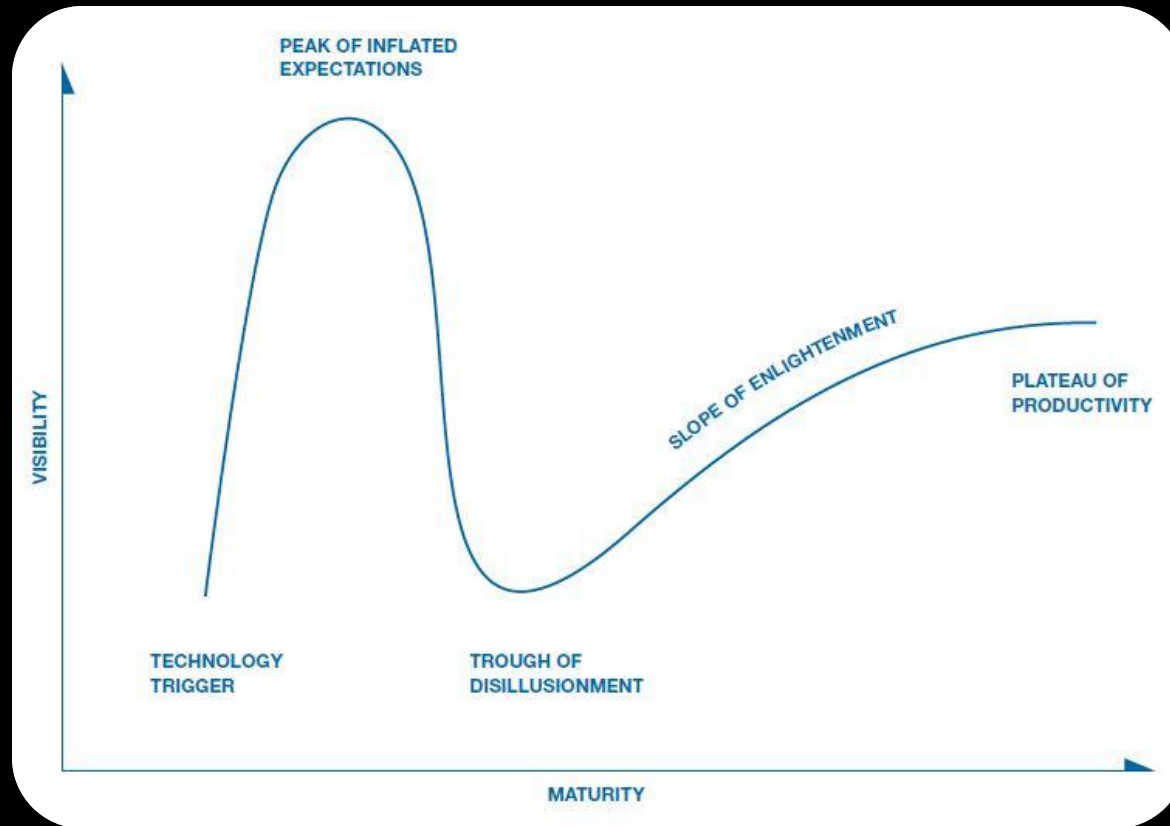
These lectures are intended to raise more questions than they answer

We Begin With

The Lure of Reconfigurable Systems

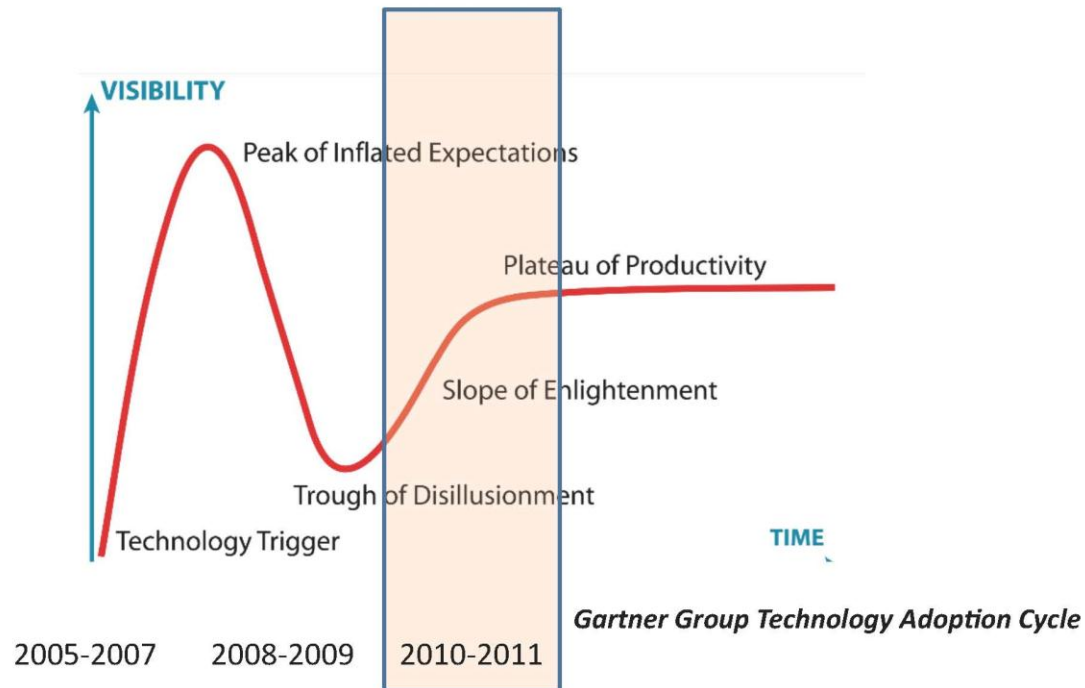
Gartner Hype Cycle

From their “Research Methodologies” [webpage](#)



Stahlberg's modification of the **GHC** at **MRSC 2010**

What's Ahead for Reconfigurable Computing



22-MAR-2010

MRSC 2010



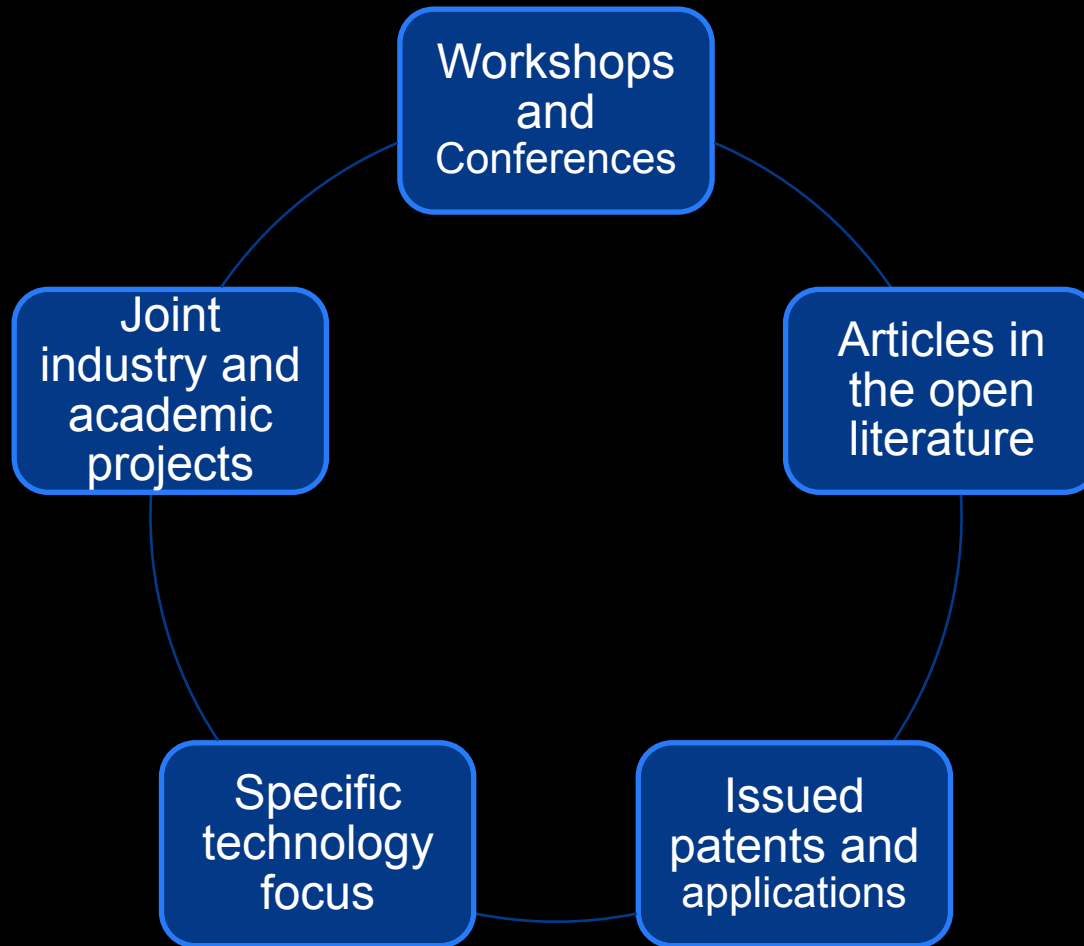
Reconfigurable Computing

Gerry Estrin's seminal 1960 paper
(see Origins paper)

The basic issues remain the same

- Granularity
- Partitioning/clustering/assigning
- Frequency of configuring/reconfiguring
- Routing/Interconnects
- Tools and tool flow – requirements, specifications, models, simulations, prototyping, testing, etc.

Over 50 Years of Interest in Reconfigurable Systems



Some 2010 Conferences

HiPEAC Workshop on Reconfigurable Computing (January 2010)

Many-Core and Reconfigurable Supercomputing Conference (MRSC, March 2010)

Engineering of Reconfigurable Systems and Algorithms (ERSA, May 2010)

International Symposium on Applied Reconfigurable Computing (March 2010)

International Conference on ReConFigurable Computing and FPGAs (ReConFig, December 2010)

Benefits of Reconfigurable Systems



Reliability and availability

Various levels of granularity

Performance 100s/1000s X conventional CPU-based implementations for specific applications

Development time less than application-specific circuits

Very low energy budgets

Performance and Power Consumption

Algorithm	Speedup	FPGA	CPU
DES Encryption [3]	24	Garp 133 MHz	SPARC 167 MHz
Number Factoring [4]	6.8	Xilinx XC4085 16 MHz	UltraSPARC 200 MHz
Intrusion Detection [5]	27.8	Xilinx Virtex2 303 MHz	Pentium 4 1.7 GHz
Numerical Simulation [6]	5.69	Xilinx Virtex4 50 MHz	Intel P4 3.0Ghz
Genome Sequencing [7]	100	Xilinx Virtex4 125 MHz	AMD Opteron 2.2 GHz

Table 1: Hardware to software speedup

This figure is taken from Chandy and Singaraju's 2009 paper, "Hardware parallelism vs. Software parallelism"

FPGA implementations of various algorithms provided substantial speedups at lower clock speeds

"implementations of various algorithms" = use of the FPGA reconfigurable fabric to eliminate context switching and fetch costs, and increasing pipelining and instruction parallelism

“Systems that, by means of redundancy, monitoring, and learning capacity, have the ability either to correct or compensate for internal error.” Dorrough

Systems that are capable of adapting their behavior and resources based on changing environmental conditions and demands. MIT Report

Systems that can be adapted to varied user requirements and varied environments

Self-Repairing Systems

Systems where a large number of simple entities interact to produce complex global behavior

Self Organizing Systems



Self-Aware Systems

Core reconfigurable concepts?

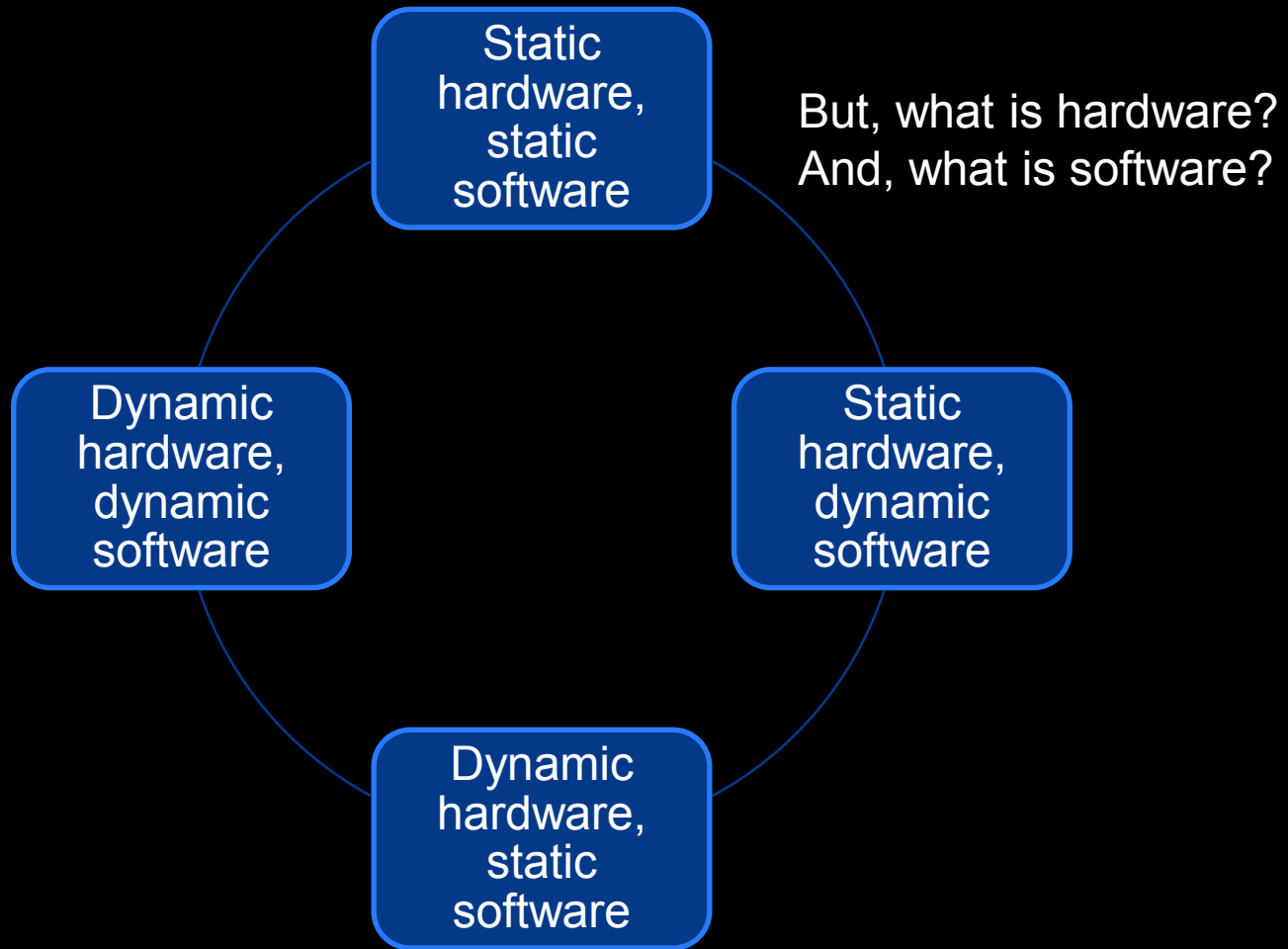
Self-Modifying and Self-Replicating Systems

A collection of entities that adapt their individual or collective behavior based on experience

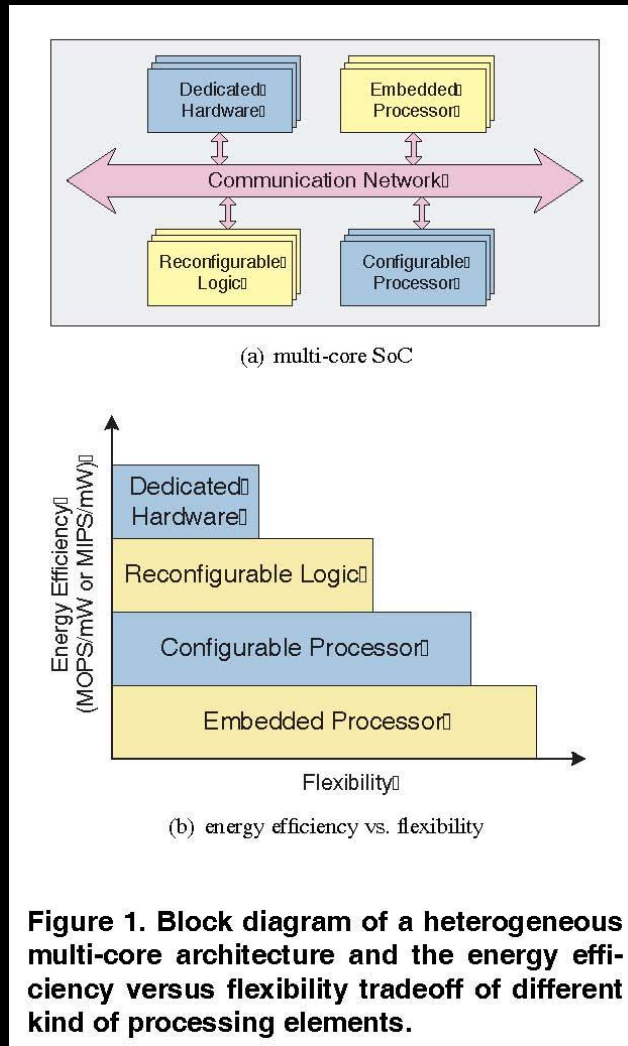
Adaptive Systems

Biological inspired systems that can modify/reproduce themselves. PREPLEXUS

What can be Reconfigured



Kwok's Heterogeneous Multi-Core SoC Model



- Figure taken from 2008, Kwok and Kwok, On the Design, Control, and Use of a Reconfigurable Heterogeneous Multi-Core System-on-a-Chip
 - Dedicated hardware = DPSs, cryptographic cores, etc.
 - Embedded Processor = conventional CPU cores
 - Reconfigurable logic = FPGA to off load computation intensive algorithms
 - Configurable processors = customizable for specific tasks

Bindings

When should reconfiguration occur

- Statically, dynamically, mixed

What resources should be configurable/reconfigurable?

- Hardware: mechanism, policy, both
- Software: mechanism, policy, both

Reconfigurable Instruction Cell Array (RICA)

RICA: An array of customizable instruction cells; dynamic reconfigurable, enabling the mapping of dependent and independent instructions; with software support analogous to a coarse-grained FPGA implementation

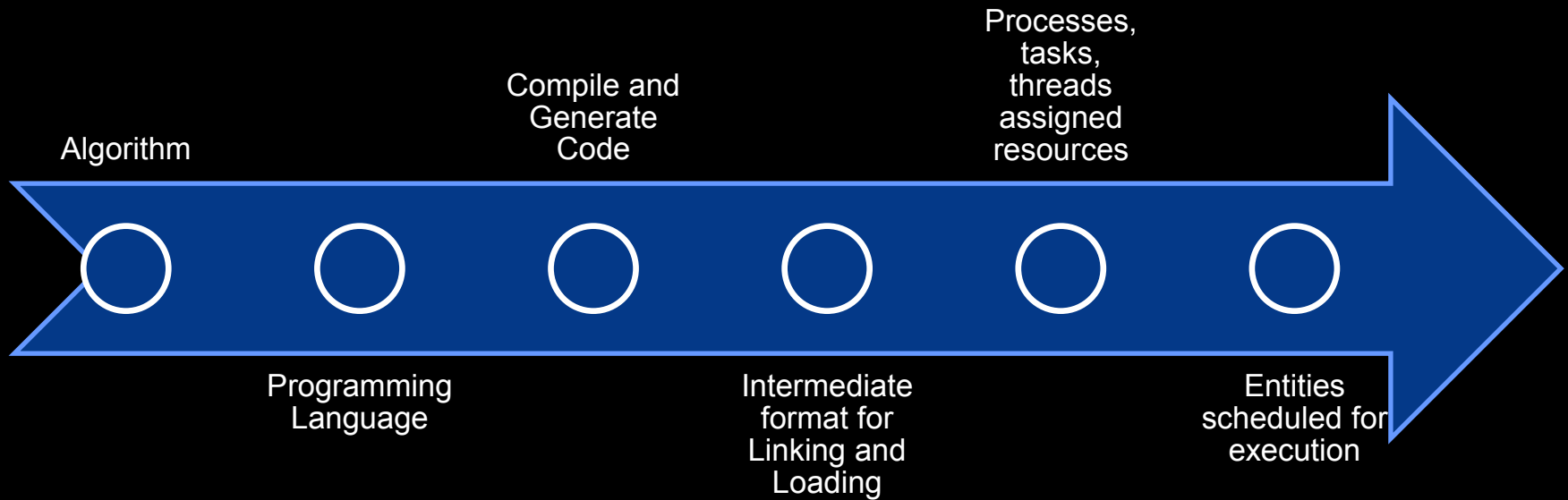
2008, Khawam et al, The reconfigurable instruction cell array

Patent Application and Commercialization

2008 El-Rayis et al, Addressing Future Space Challenges using Reconfigurable Instruction Cell Based Architectures

2010, Han, Multi-core Architectures with Coarse-grained Dynamically Reconfigurable Processors for Broadband Wireless Access Technologies

Implications



Some Examples

Implementation Technology Constraints

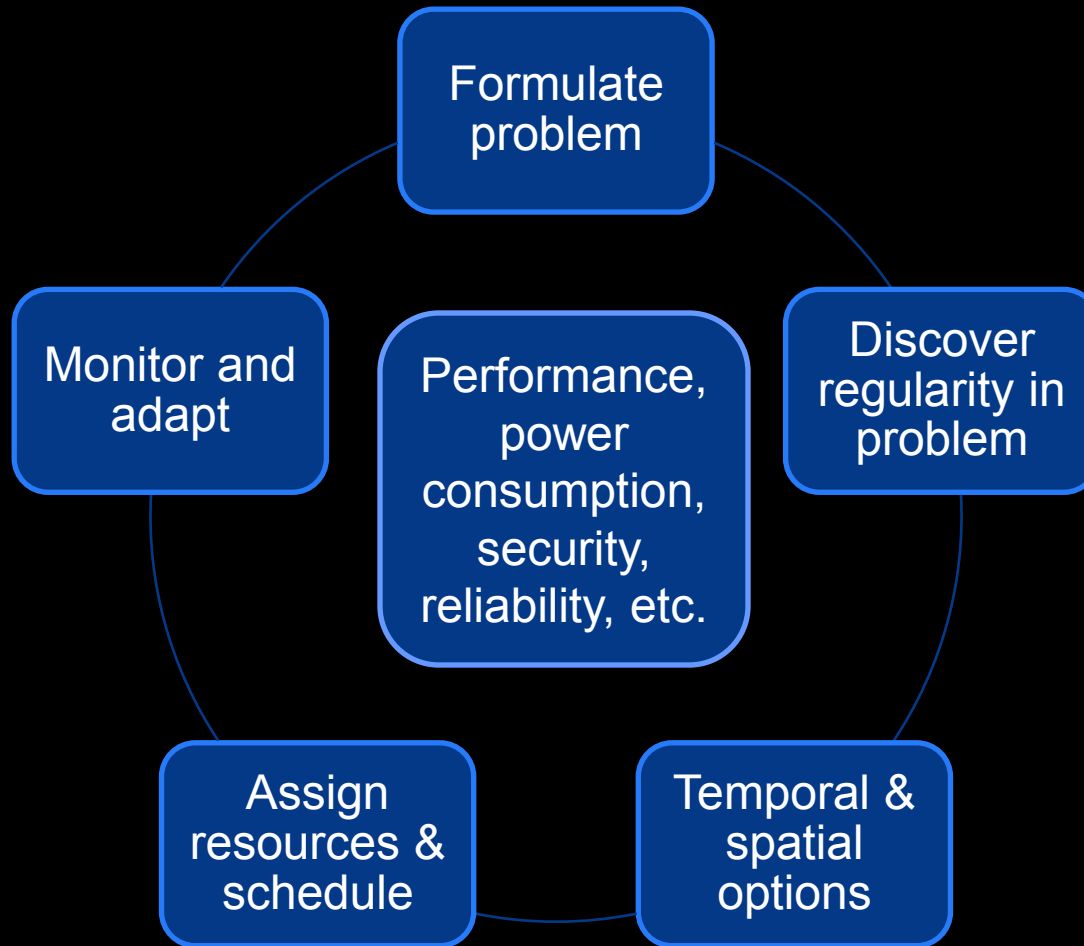
Partial or full configuration or reconfiguration

- What needs to be specified statically or dynamically specify or change a configuration?
- Storage and bandwidth requirements
 - E.g., a configuration change in an FPGA can require over a megabit of data
- Power and Time (latency) implications

Request to initiate a dangerous or forbidden configuration

Detectable statically and/or dynamically?

Modification Paradigm



A Recurring Bottleneck

Interconnection technology may be one of the bottlenecks, e.g., among cores on a many-core processor chip or in a million gate FPGA

2004, Laskowski, “Program Scheduling in Look-Ahead Reconfigurable Parallel Systems with Multiple Communication Resources”

2009, Akram et al, “Workload Adaptive Shared Memory Multicore Processors with Reconfigurable Interconnects”

OS Support

Addressing OS run-time system support for applications executing on reconfigurable hardware

- 2007, So, BORPH: An Operating System for FPGA-Based Reconfigurable Computers

Addressing reconfiguration opportunities within the operating system itself

- 2006, He and Chiang, Reconfigurable and power-saving operating system design for supporting mobile nodes in sensor networks

Hardware/Software Partitioning

Identifying the parts of an application to be implemented

- On the underlying reconfigurable structure
- Within the software

2009, Ostandzadeh et al, "A Multipurpose Clustering Algorithm for Task Partitioning in Multicore Reconfigurable Systems"

Accelerator Cores

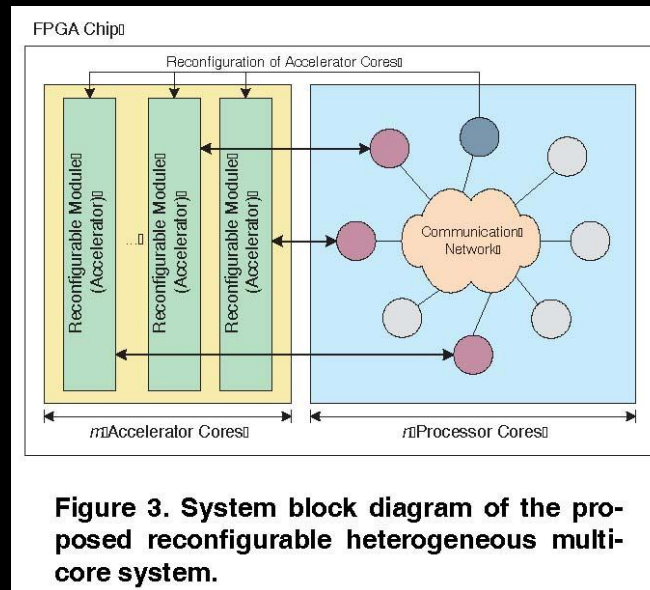


Figure 3. System block diagram of the proposed reconfigurable heterogeneous multi-core system.

Revisiting Kwok and Kwok, "On the Design, Control, and Use of a Reconfigurable Heterogeneous Multi-Core System-on-a-Chip"

- Accelerator cores are tightly coupled to processor cores
- The processor cores when need IP-cores are not available

Revisiting Dataflow in a Reconfigurable Context

2008, Bhattacharyya et al, OpenDF – A Dataflow Toolset for Reconfigurable Hardware and Multicore Systems

Why dataflow might actually work

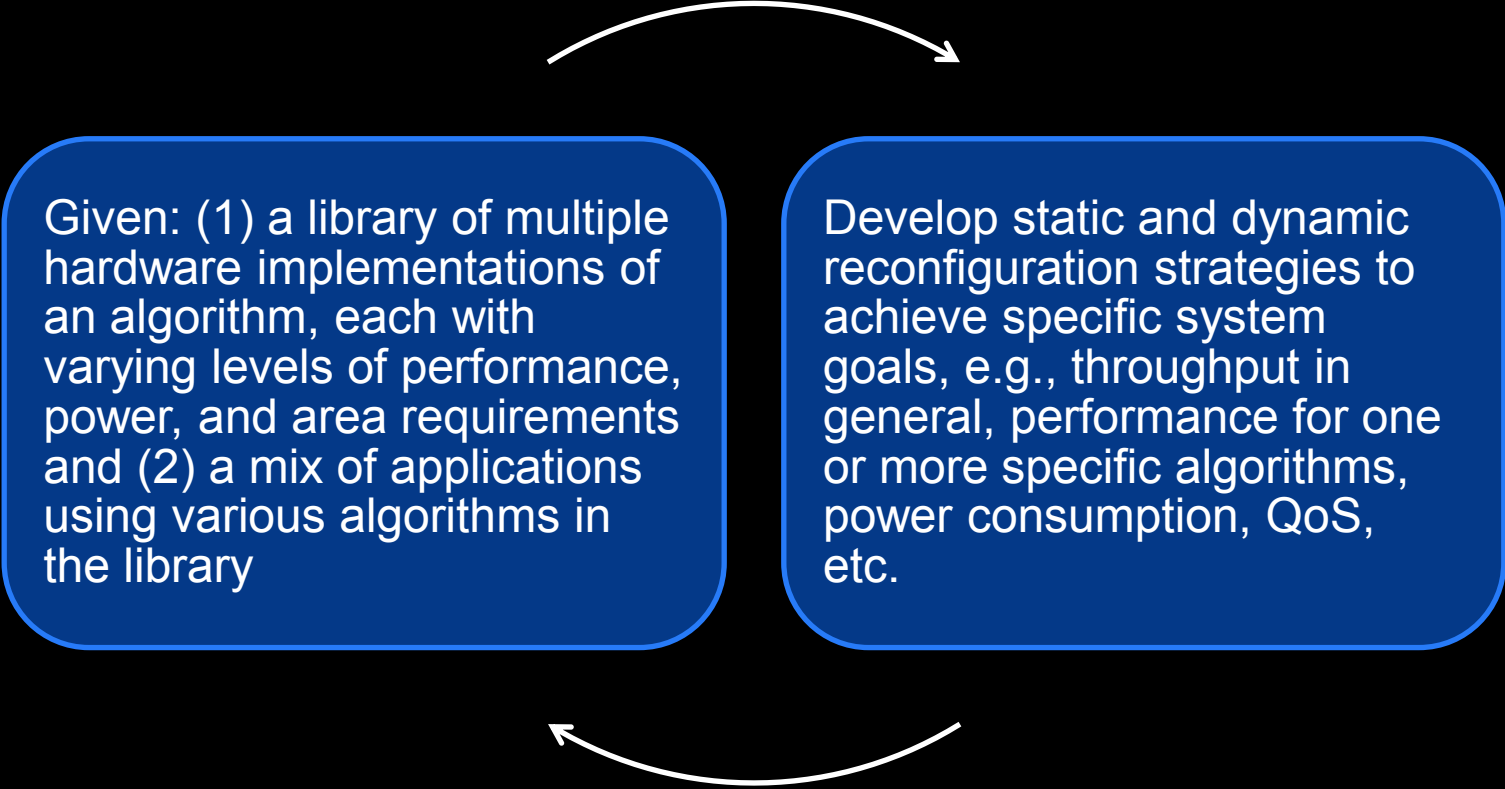
- Scalable parallelism
- Modularity and reuse
- Scheduling
- Portability
- Adaptability

Hardware Configurations

Given: (1) a library of hardware configurations, each able to achieve specific levels of performance and power consumption for specific area requirements and (2) a mix of applications executing on a particular hardware configuration

Develop static and dynamic reconfiguration strategies to achieve specific system goals, e.g., throughput in general, performance for one or more specific algorithms, power consumption, QoS, etc.

Library of Algorithms



Given: (1) a library of multiple hardware implementations of an algorithm, each with varying levels of performance, power, and area requirements and (2) a mix of applications using various algorithms in the library

Develop static and dynamic reconfiguration strategies to achieve specific system goals, e.g., throughput in general, performance for one or more specific algorithms, power consumption, QoS, etc.

Chandy/Singaraju's RHyMA

Reconfigurable Hybrid Multicore Architecture

- CPU cores, stream processing cores, reconfigurable hardware and multi-ported memory to support access from multiple cores
- Interconnect =NOC, such as a mesh

Questions

