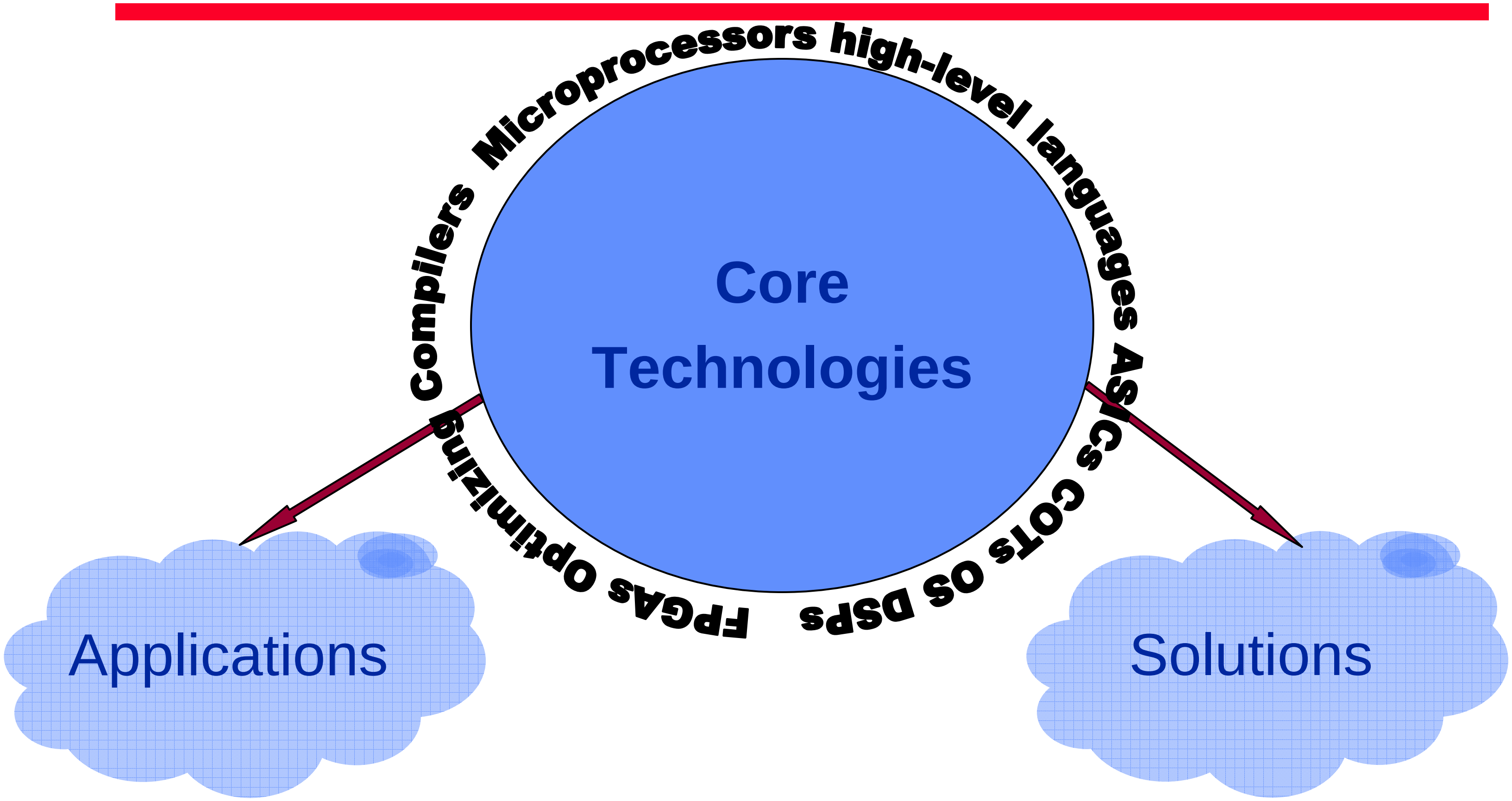
RICE

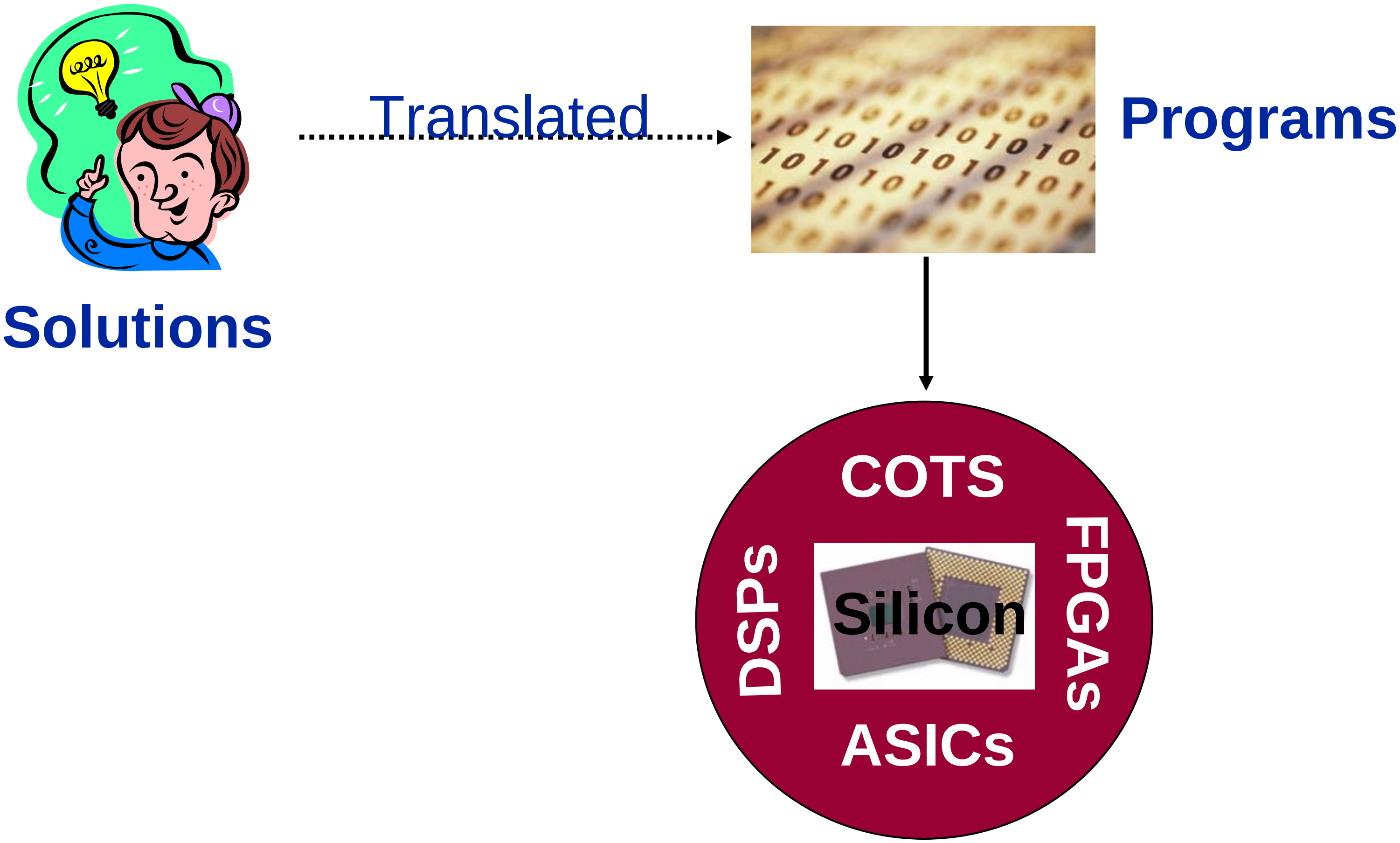
CS 514/ ECE 518: Designing Embedded Computing Environments (Overview)

Philosophy of this Course


Microprocessors high-level languages ASICs COTS OS DSPs FPGAs Optimizing Compilers
Core Technologies
Applications
Solutions

- The evolution of core technologies enrich the class of applications and solutions
- We will be studying the key technologies that will be driving the growth of embedded systems in the next decade

What is computing?



Solutions

Translated

Programs

COTS

DSPs

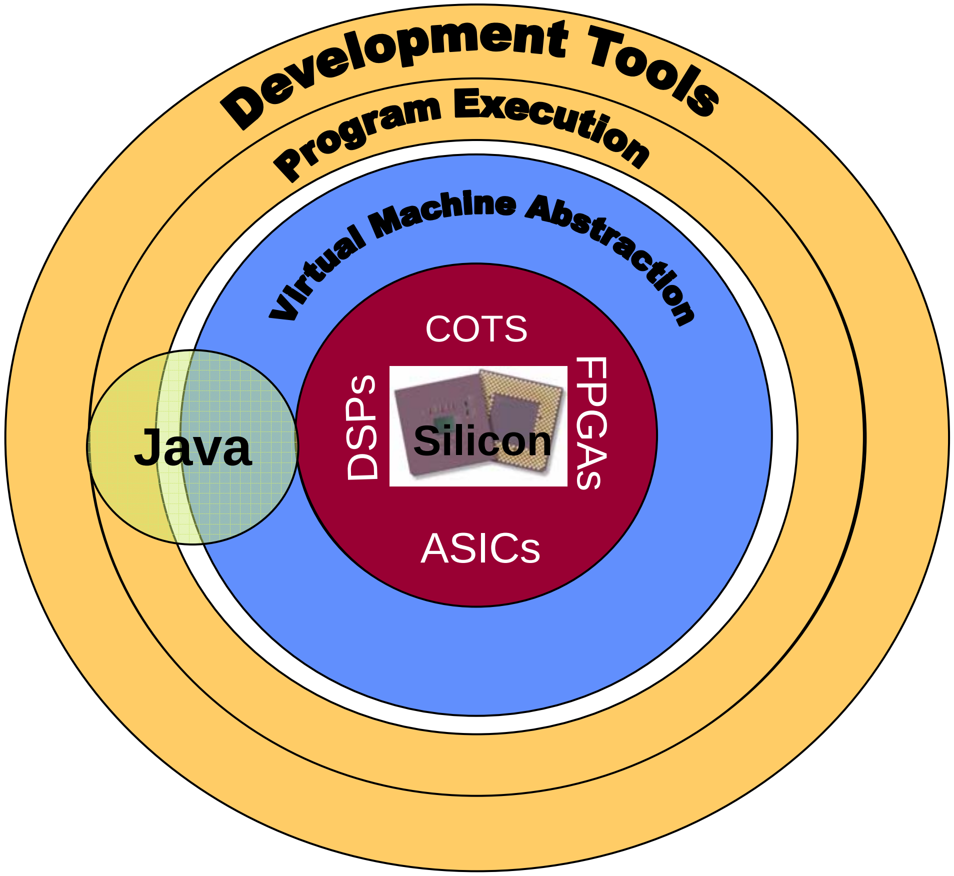
Silicon

FPGAS

ASICs

- In the abstract, computing is realizing solutions to problems onto silicon

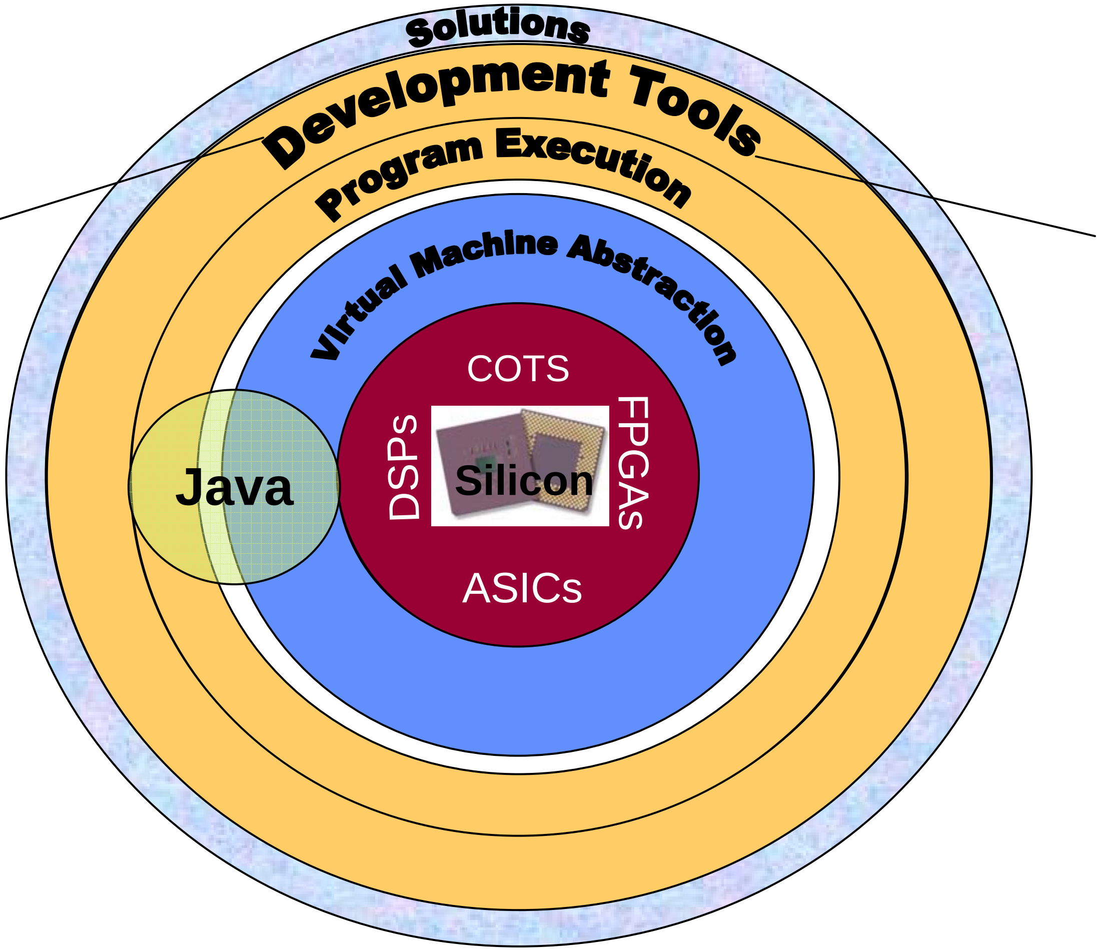
The Virtual Machine



- The virtual machine provides an abstraction of the silicon to support the software layers

The Development Tools

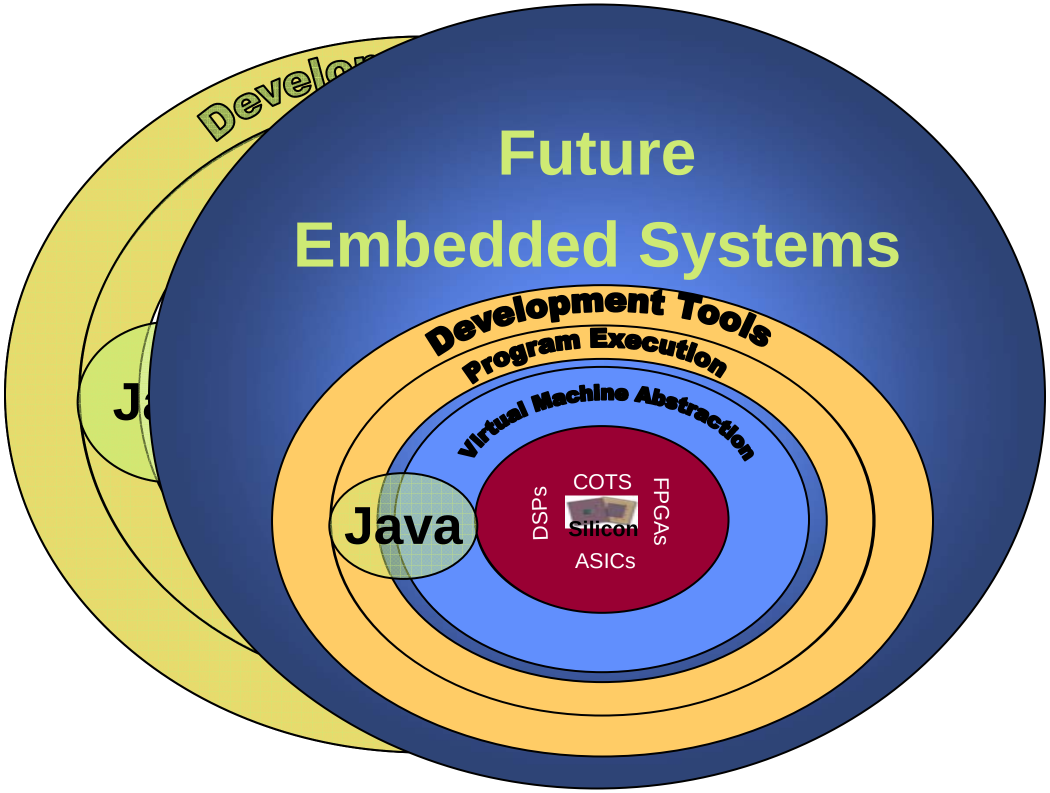
- High-level languages
- Compilers



- CAD
- VLSI

- The development tools provide designers the vehicle for implementing their solutions

The State-of-the-Art for Tomorrow



- The embedded systems of tomorrow will leverage the evolving hardware and software technologies

Course Goals

- Provide a broad overview of the embedded computing space
  - Hardware options
  - Software solutions
  - Putting it all together
- Provide an introduction to the challenges faced by the computing and engineering side of this field
- Provide a core set of knowledge that allows students to be active participants in developing these new computing engines.

Embedded Computing

Why ?

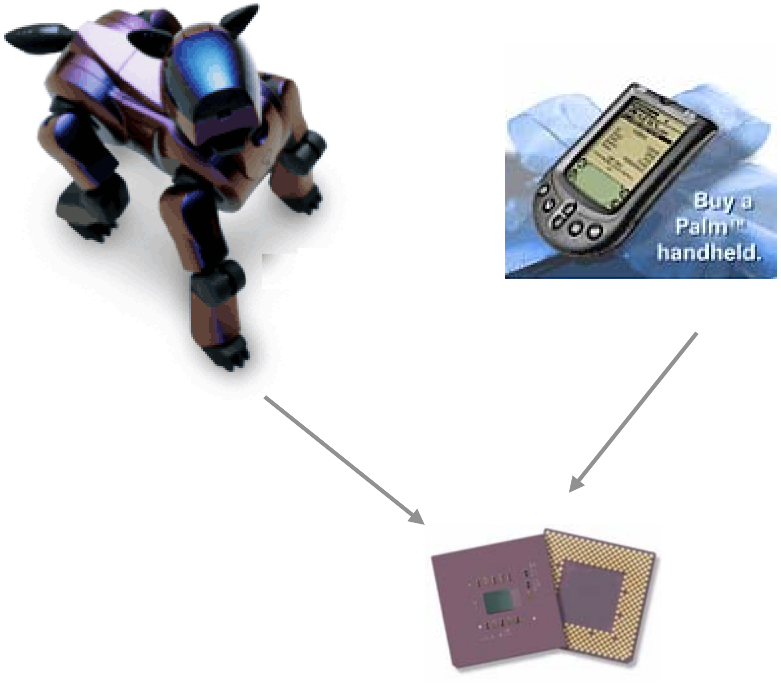
What ?

How ?

The Nature of Embedded Systems

Visible Computing

View is of end application



(Hidden computing element)

Favorable Trends

- Supported by Moore's (second) law
  - Computing power doubles every eighteen months

  Corollary: cost per unit of computing halves every eighteen months

- From hundreds of millions to billions of units

- Projected by market research firms (VDC) to be a 50 billion+ space over the next five years

- High volume, relatively low per unit $ margin

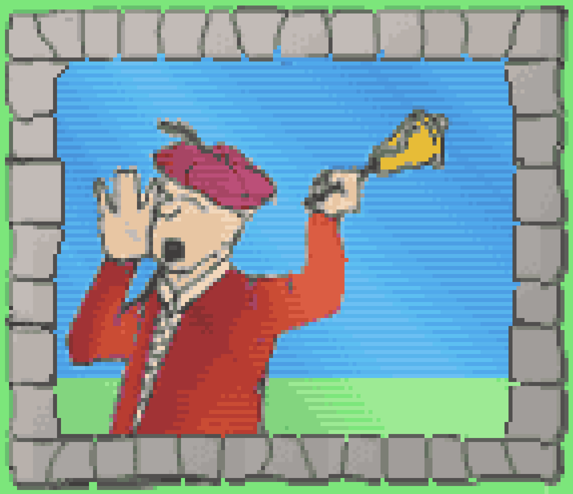
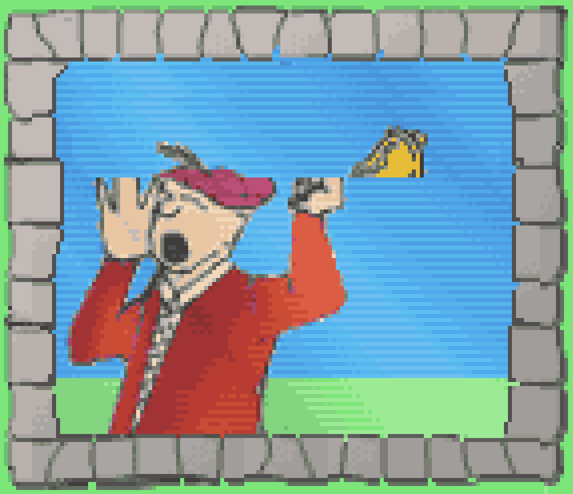
Embedded Systems Desiderata

- Low Power
  - High battery life
- Small size or footprint
- Real-time constraints

Performance comparable to or surpassing leading edge COTS technology

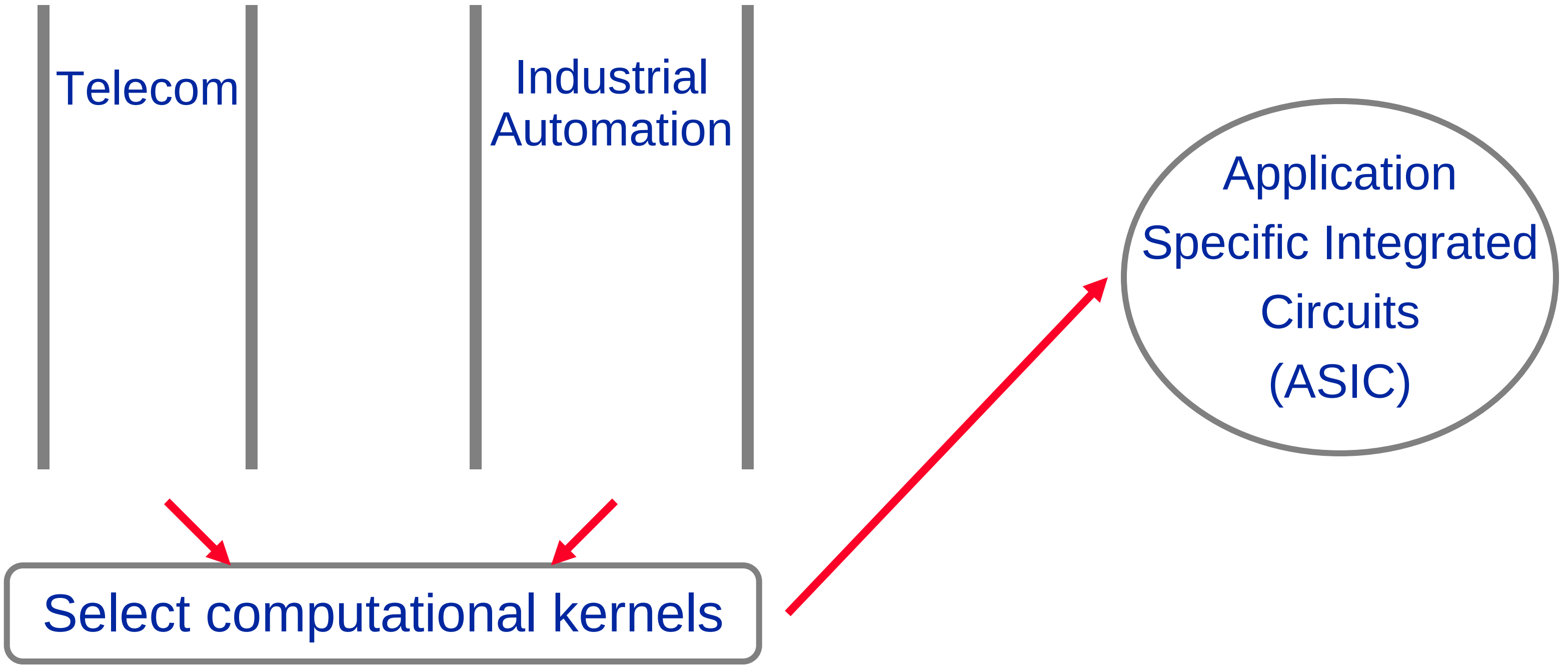
Rapid time-to-market

Timing Example

<table>
<tr><th colspan="2">Video-On-Demand</th></tr>
<tr><td><b>Predictable Timing Behavior</b></td><td><b>Unpredictable Timing Behavior</b></td></tr>
<tr><td></td><td></td></tr>
</table>

Current Art

Vertical application domains



Telecom

Industrial Automation

Select computational kernels

Application Specific Integrated Circuits (ASIC)

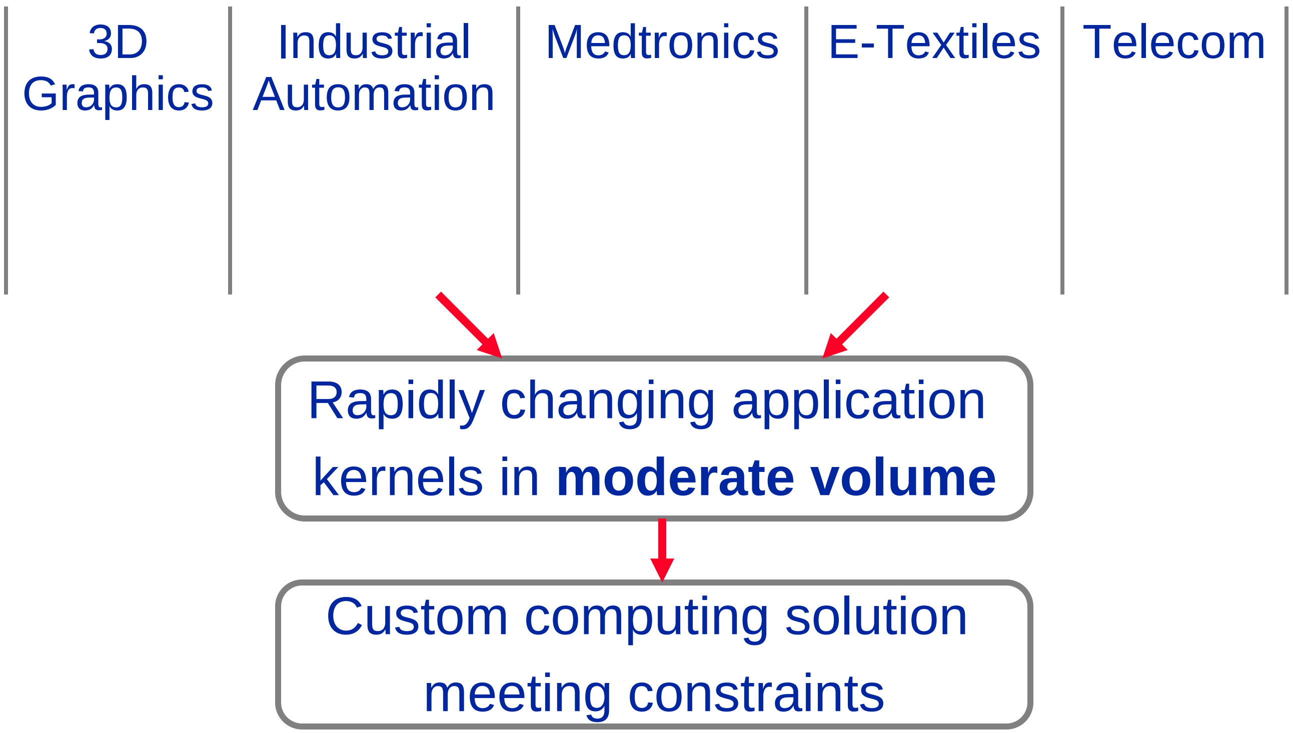
- Meet desiderata while overcoming Non-Recurring Engineering (NRE) cost hurdles through volume
- High migration inertia across applications
- Long time to market

Subtle but Sure Hurdles

- For Moore’s corollary to be true
  - Non-recurring engineering (NRE) cost must be amortized over high-volume
  - Else prohibitively high per unit costs
- Implies “uniform designs” over large workload classes
  - (Eg). Numerical, integer, signal processing
- Demands of embedded systems
  - “Non uniform” or application specific designs
  - Per application volume might not be high
  - High NRE costs ↑ infeasible cost/unit
  - Time to market pressure

The Embedded Systems Challenge

Multiple application domains



3D Graphics

Industrial Automation

Medtronics

E-Textiles

Telecom

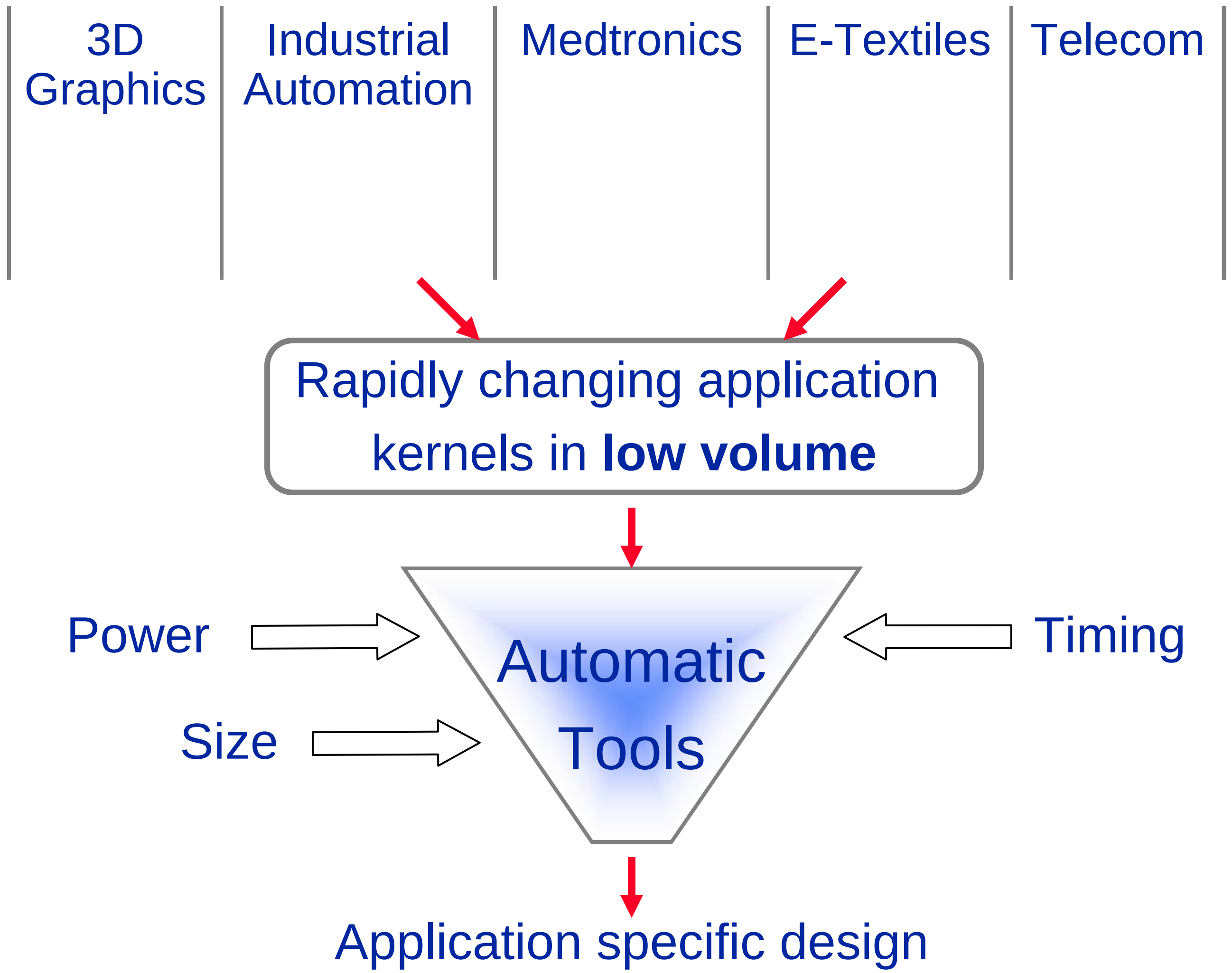
Rapidly changing application kernels in **moderate volume**

Custom computing solution meeting constraints

- Sustain Moore's corollary
  - Keep NRE costs down

Responding Via Automation

Multiple application domains



3D Graphics

Industrial Automation

Medtronics

E-Textiles

Telecom

Rapidly changing application kernels in **low volume**

Power

Size

Automatic Tools

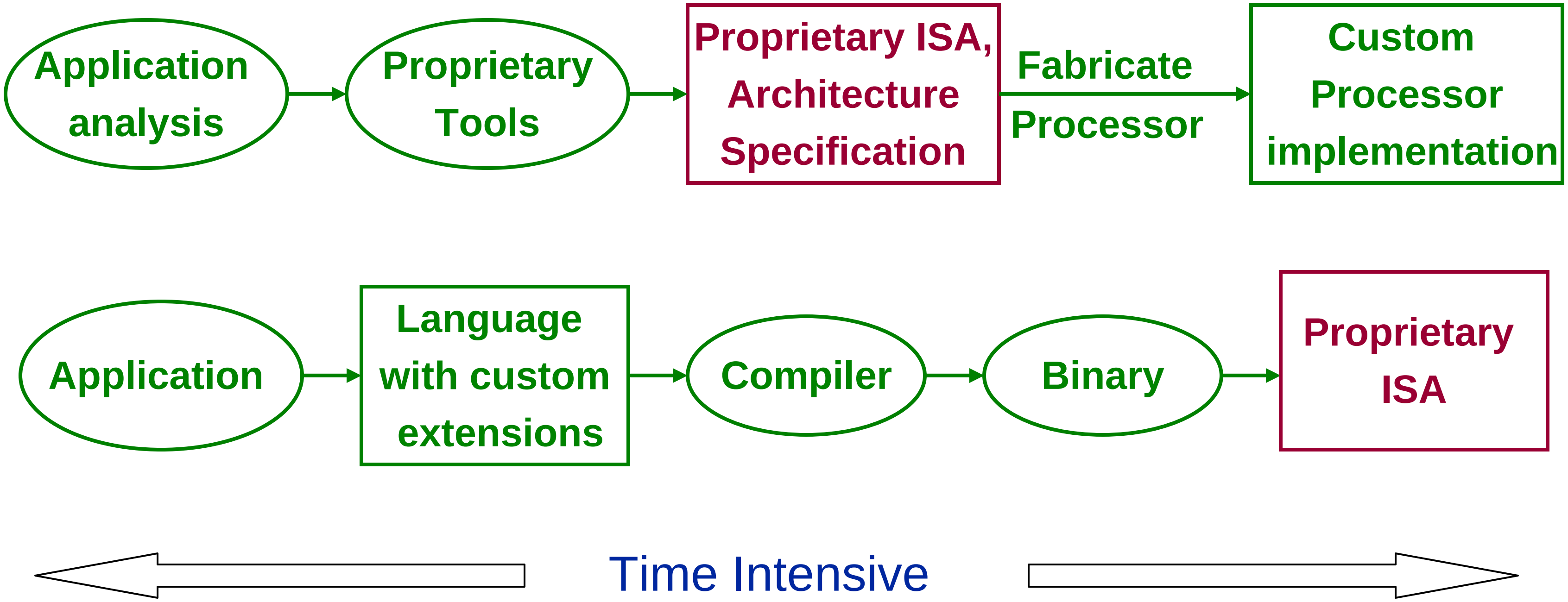
Timing

Application specific design

Three Active Approaches

- Custom microprocessors
- Architecture exploration and synthesis
- Architecture assembly for reconfigurable computing

Custom Processor Implementation



Application analysis

Proprietary Tools

Proprietary ISA, Architecture Specification

Fabricate Processor

Custom Processor implementation

Application

Language with custom extensions

Compiler

Binary

Proprietary ISA

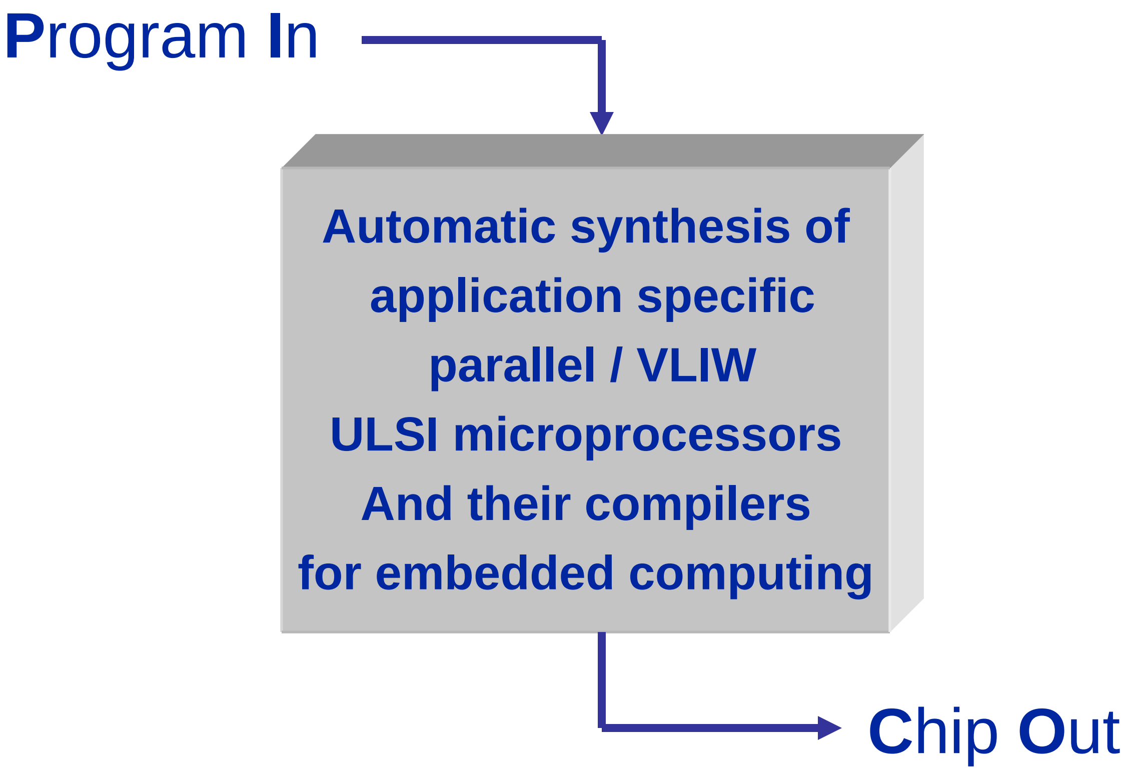
Time Intensive

- High performance implementation
- Customized in silicon for particular application domain
- O(months) of design time
- Once designed, programmable like standard processors

Tensilica, HP-ST Microelectronics approach

Architecture Exploration and Synthesis

The PICO Vision



Program In

Automatic synthesis of application specific parallel / VLIW ULSI microprocessors And their compilers for embedded computing

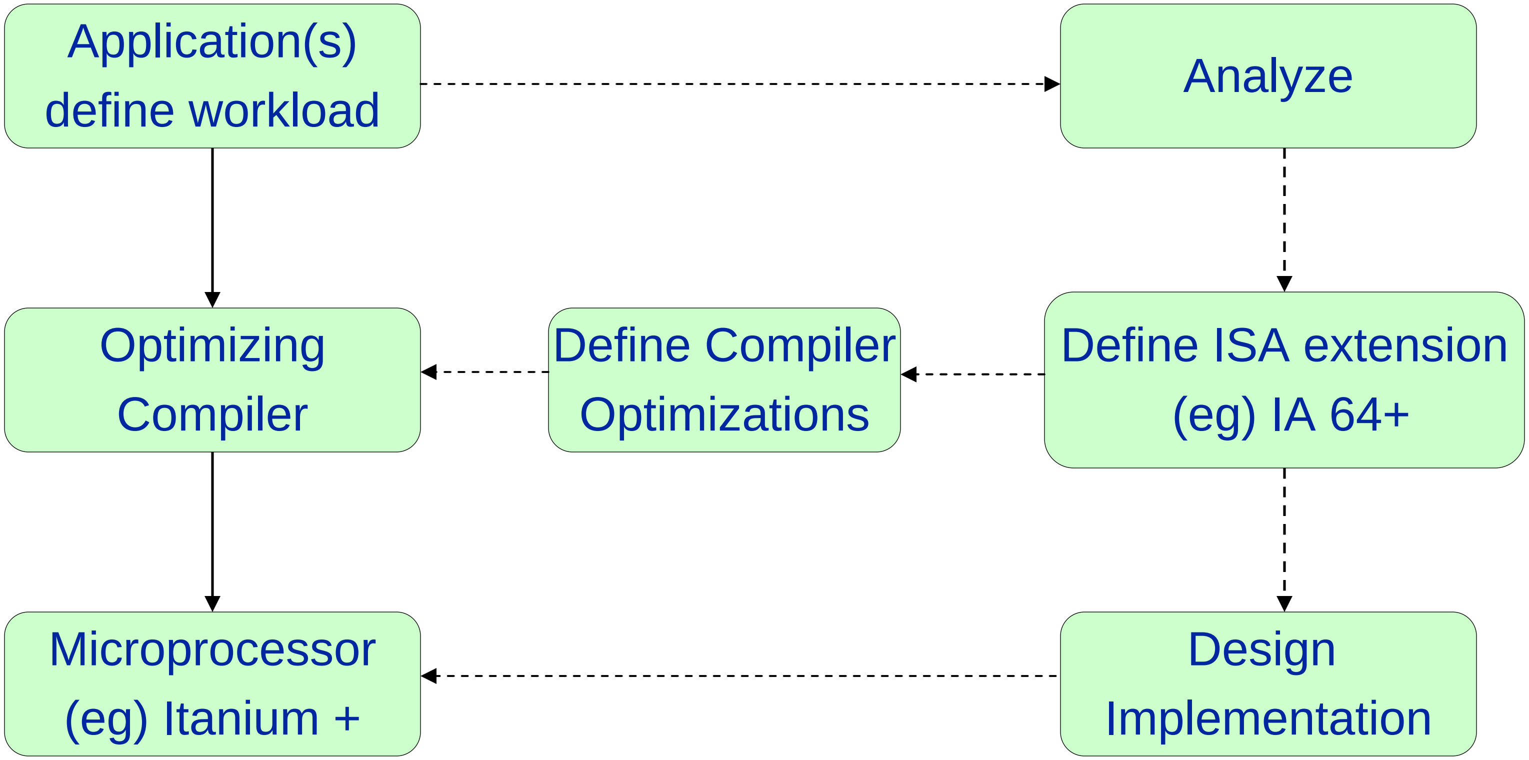
Chip Out

“Computer design for the masses”

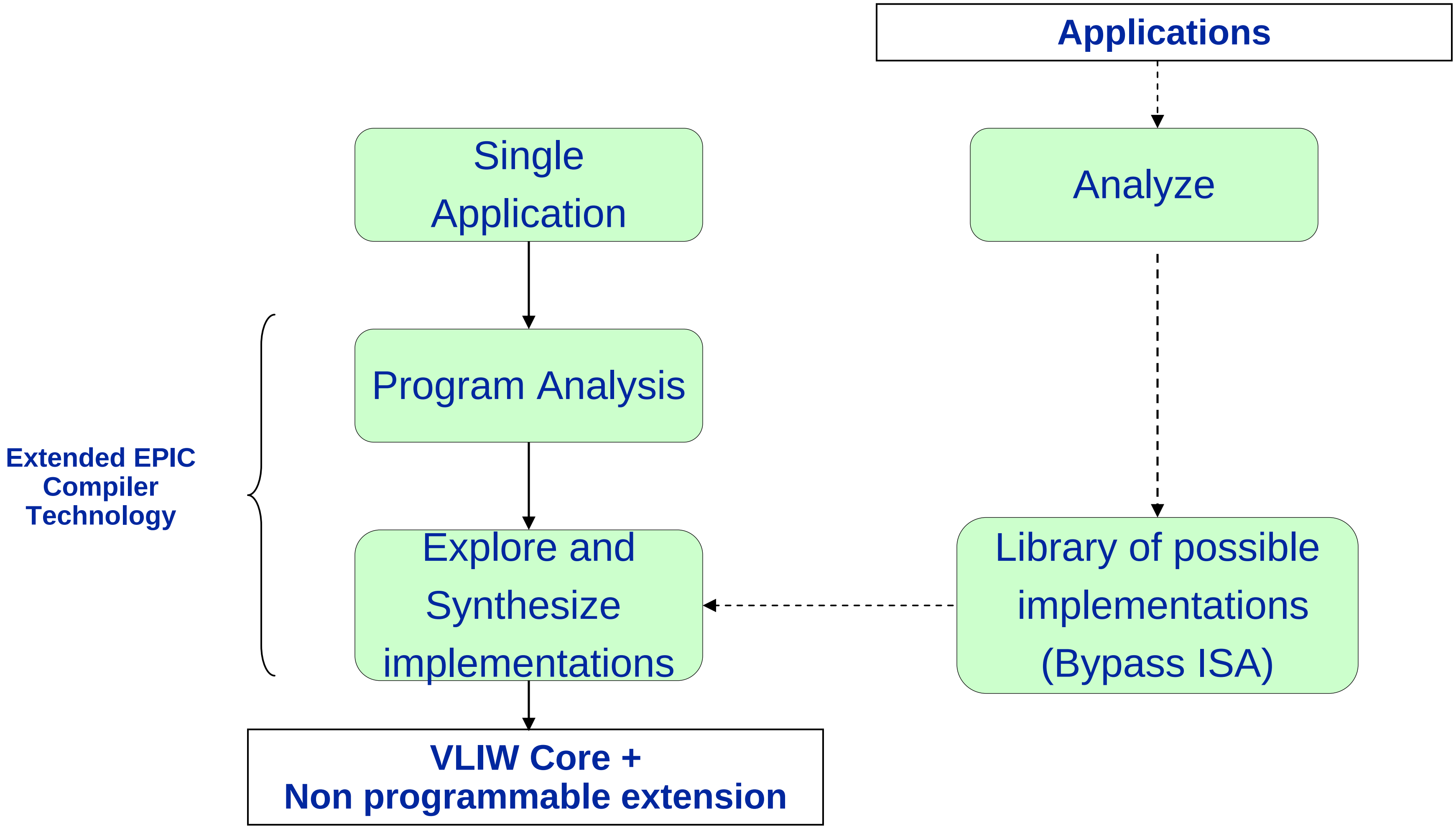
“A custom system architecture in 1 week tape-out in 4 weeks”

B Ramakrishna Rau “The Era of Embedded Computing”, Invited talk, CASES 2000.(from HP Labs Tech report HPL-2000-115)

Custom Microprocessor Design



Application Specific Design



Applications

Single Application

Analyze

Program Analysis

Extended EPIC Compiler Technology

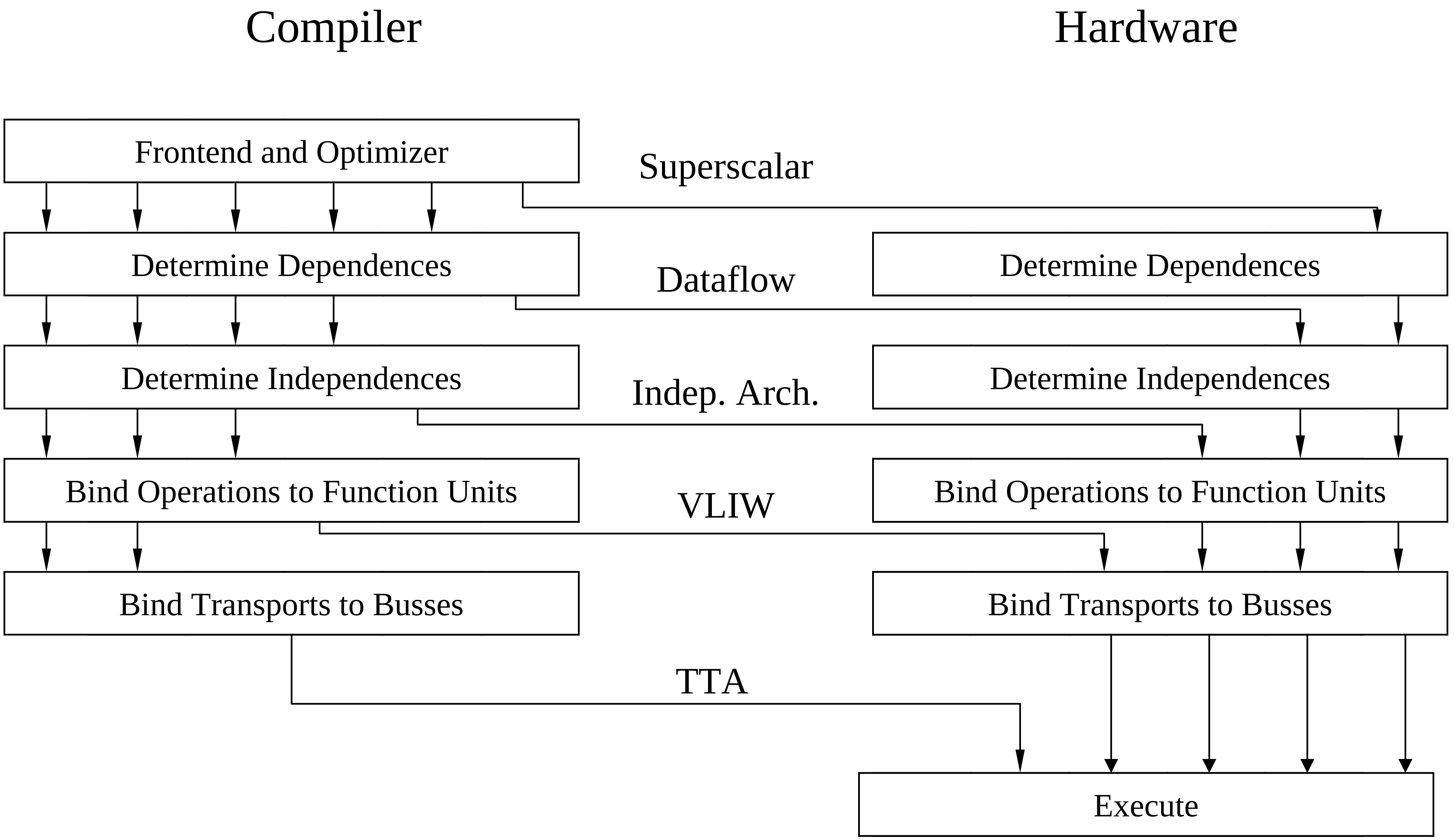
Explore and Synthesize implementations

Library of possible implementations (Bypass ISA)

VLIW Core +
Non programmable extension

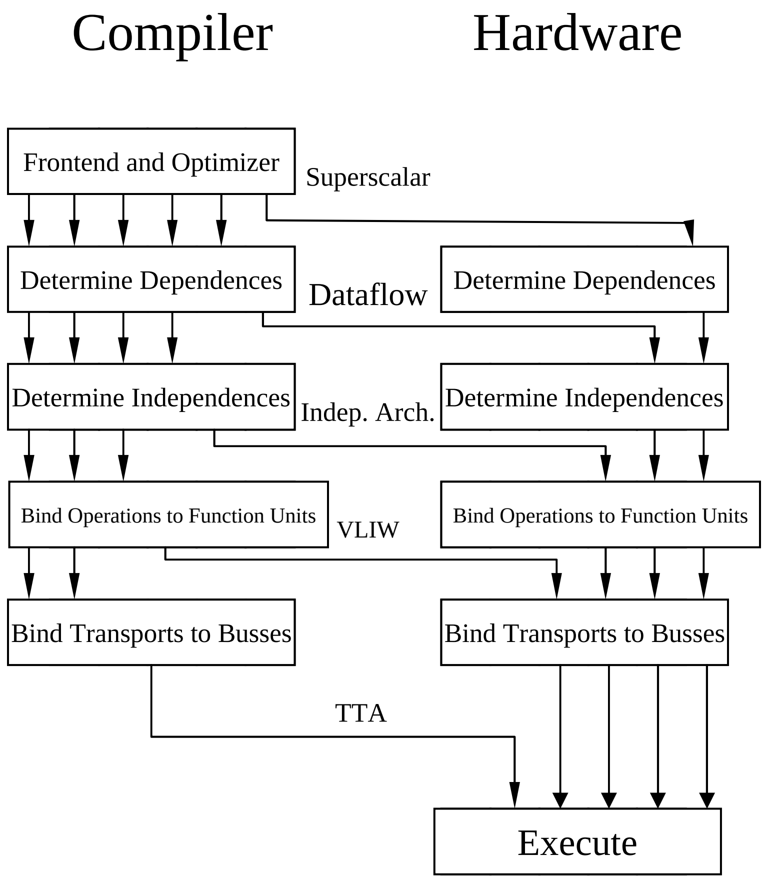
Application specific processor runs single application

The Compiler Optimization Trajectory



B. Ramakrishna Rau and Joseph A. Fisher. Instruction-level parallel: History overview, and perspective. The Journal of Supercomputing, 7(1-2):9-50, May 1993.

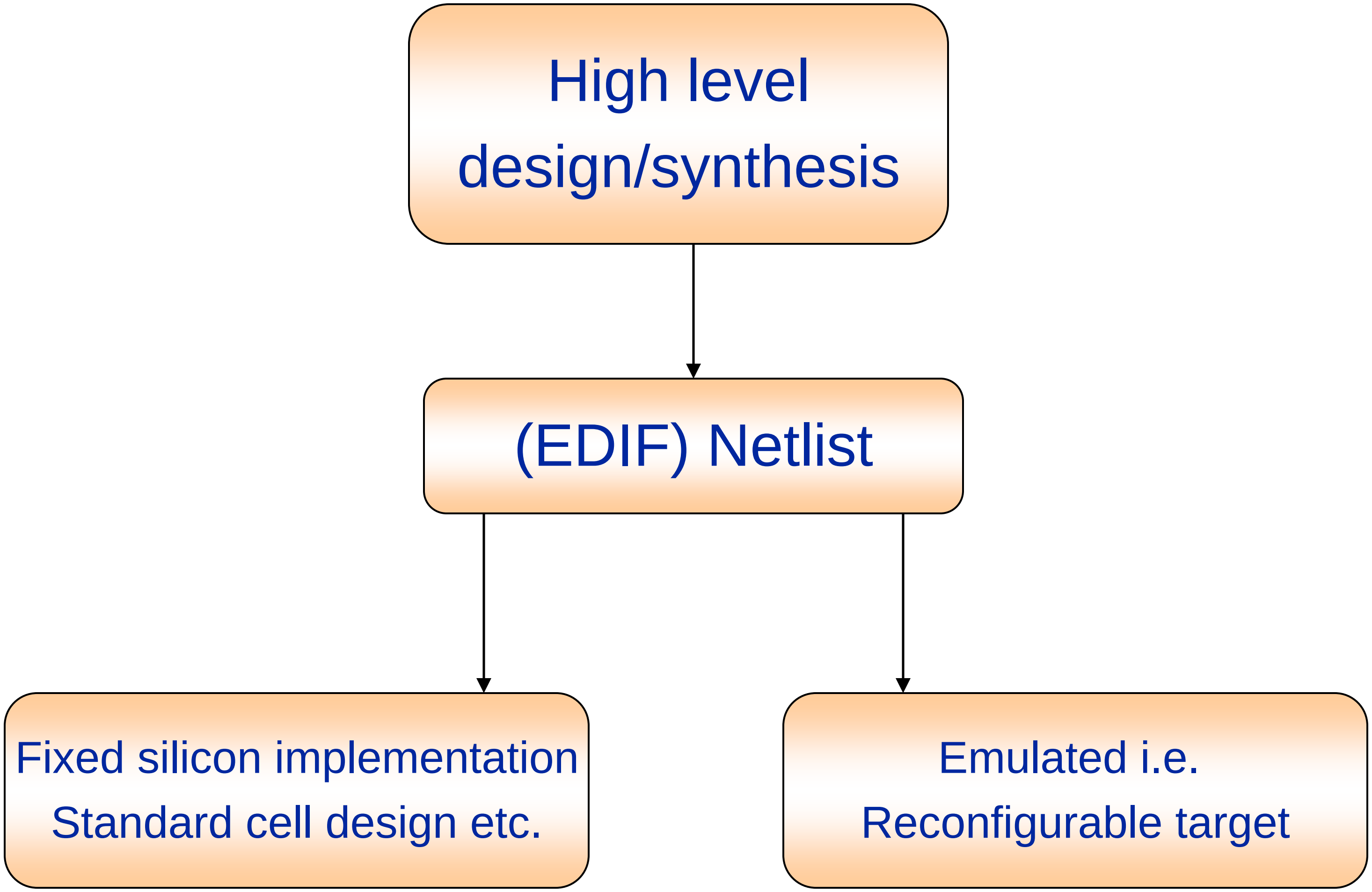
What Is the Compiler’s Target ``ISA’’?



B. Ramakrishna Rau and Joseph A. Fisher. Instruction-level parallel: History overview, and perspective. The Journal of Supercomputing, 7(1-2):9-50, May 1993.

- Target is a range of architectures and their building blocks
- Compiler reaches into a constrained space of silicon
- Explores architectural implementations
- O(days – weeks) of design time
- Exploration sensitive to application specific hardware modules
- Fixed function silicon is the result
- Verification NRE costs still there
- One approach to overcoming time to market

Choices of Silicon



High level design/synthesis

(EDIF) Netlist

Fixed silicon implementation Standard cell design etc.

Emulated i.e. Reconfigurable target

Reconfigurable Computing

FPGAs As an Alternative Choice for Customization

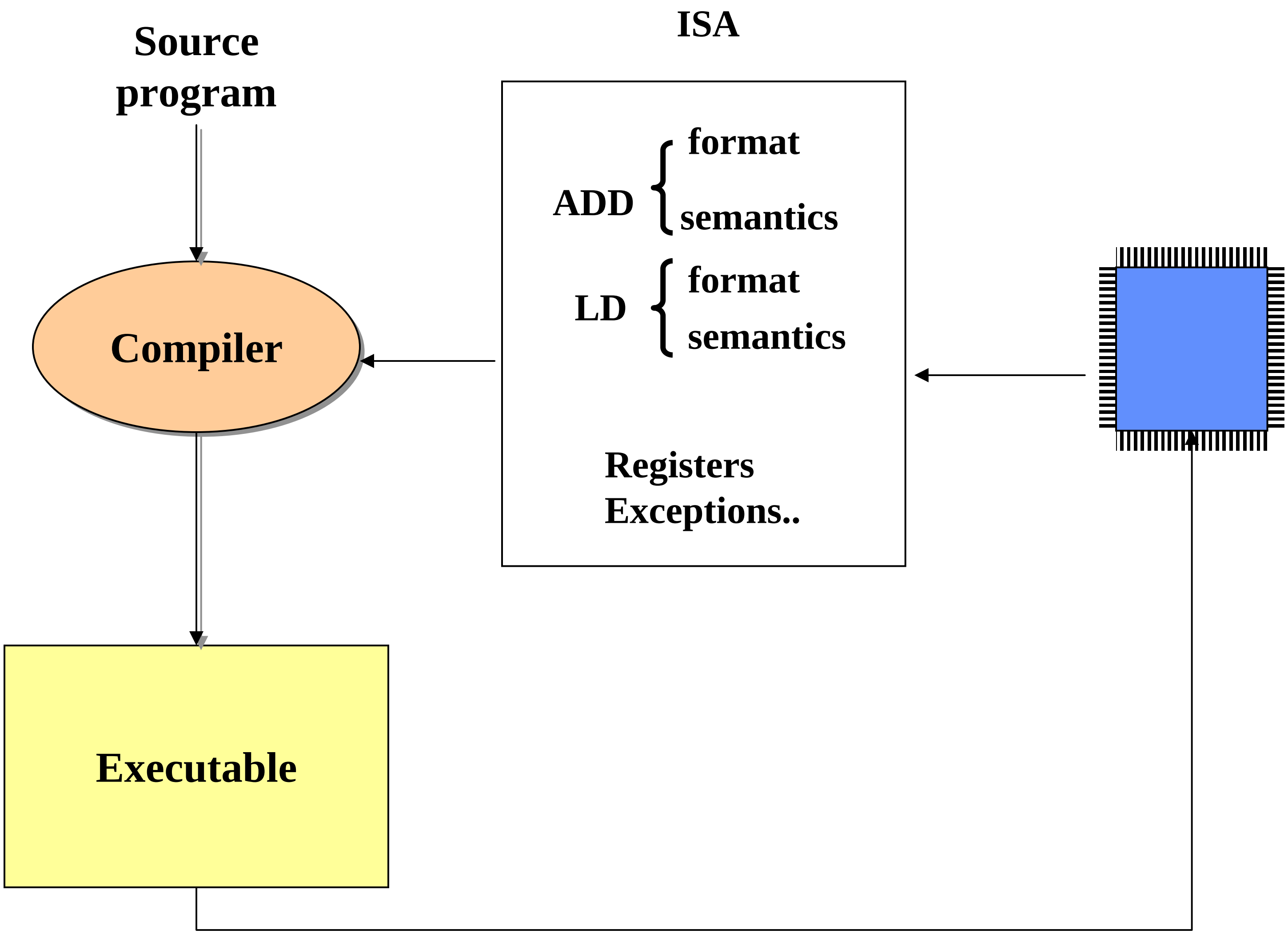
- Frequent (re)configuration and hence frequent recustomization
- Fabrication process is steadily improving
- Gate densities are going up
- Performance levels are acceptable
- **Amortize large NRE investments by using COTS platform**

Adaptive EPIC

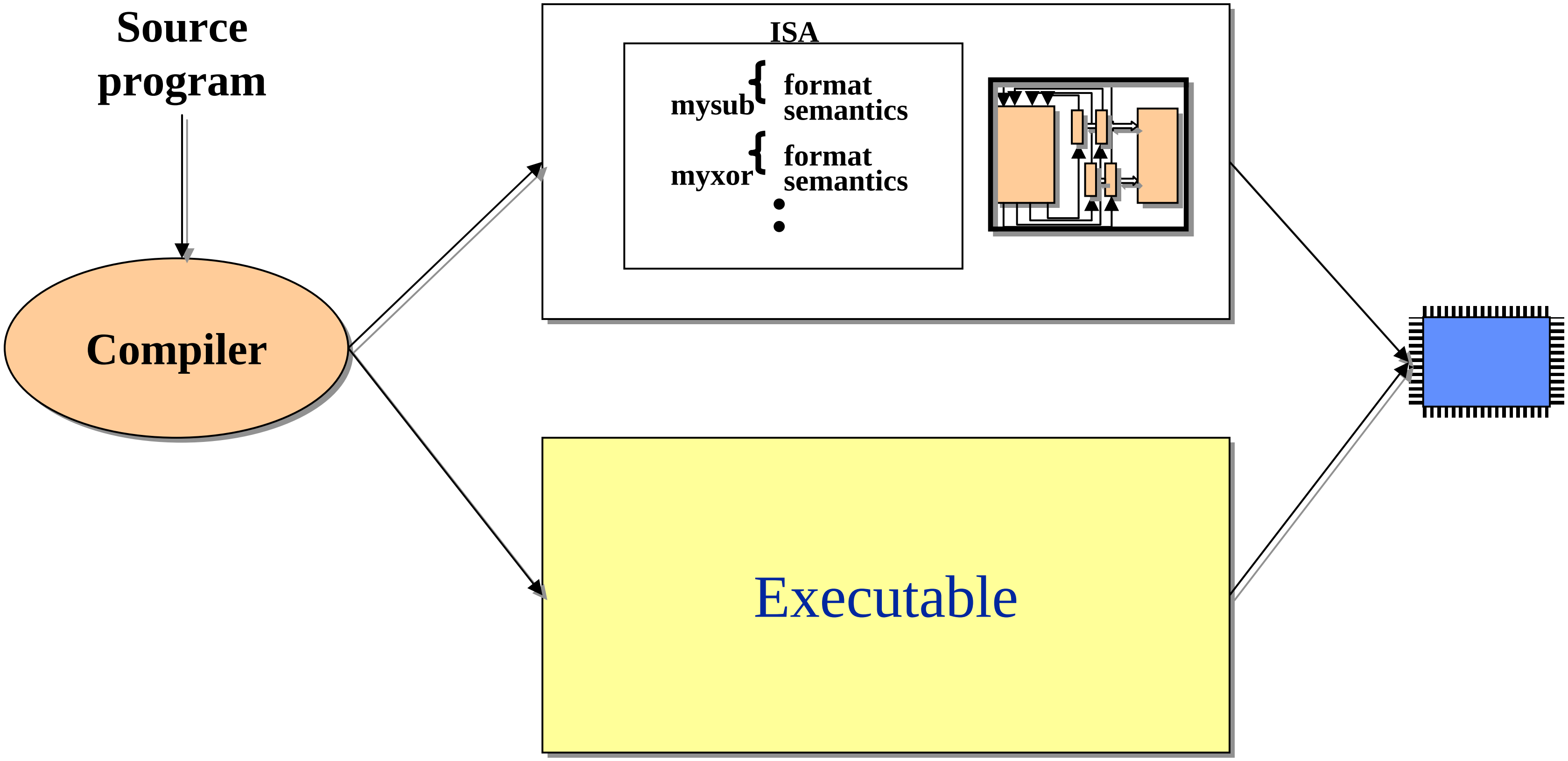
Adaptive Explicitly Parallel Instruction Computing
Krishna V. Palem and Surendranath Talla, Courant Institute of Mathematical Sciences; Patrick W. Devaney, Panasonic AVC American Laboratories Inc.
Proceedings of the 4th Australasian Computer Architecture Conference, Auckland, NZ. January 1999

Adaptive Explicitly Parallel Instruction Computing
Surendranath Talla
Department of Computer Science, New York University
PhD Thesis, May 2001
Janet Fabri award for outstanding dissertation.

Compiler-Processor Interface



Redefining Processor-Compiler Interface



Source program

Compiler

ISA

mysub { format semantics

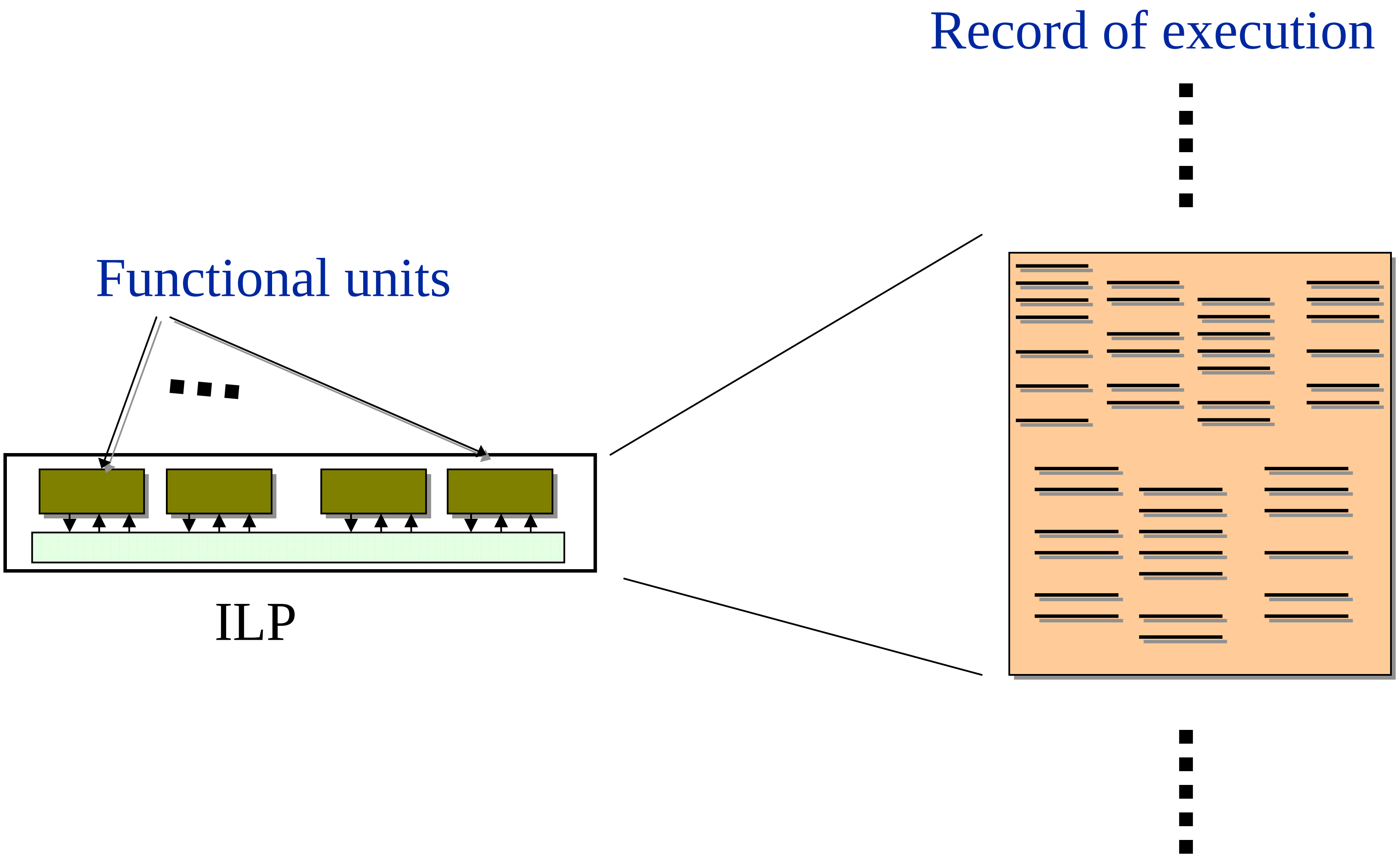
myxor { format semantics

:

Executable

Let compiler determine the instruction sets (and their realization on chip)

EPIC execution model

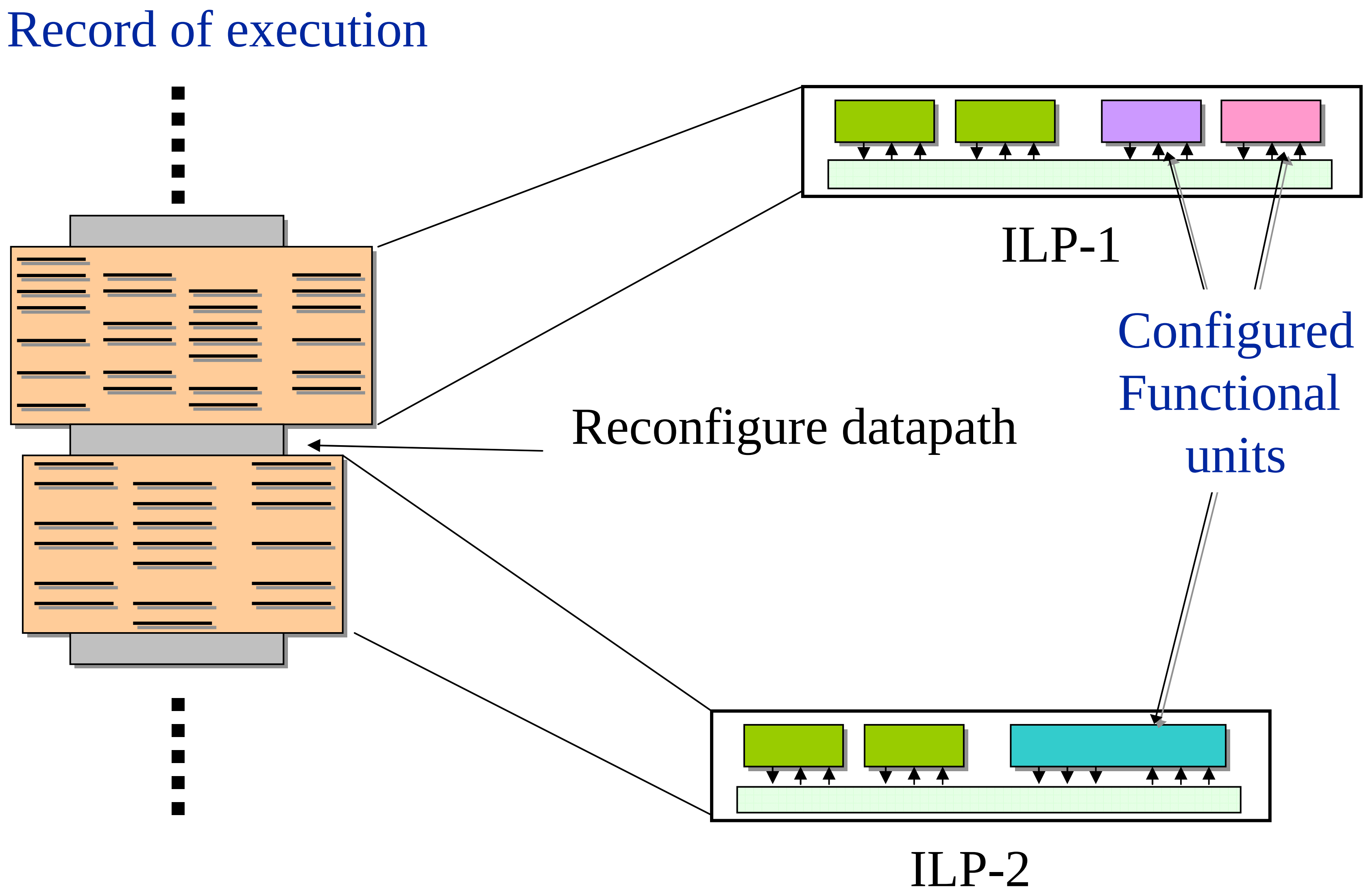


Record of execution

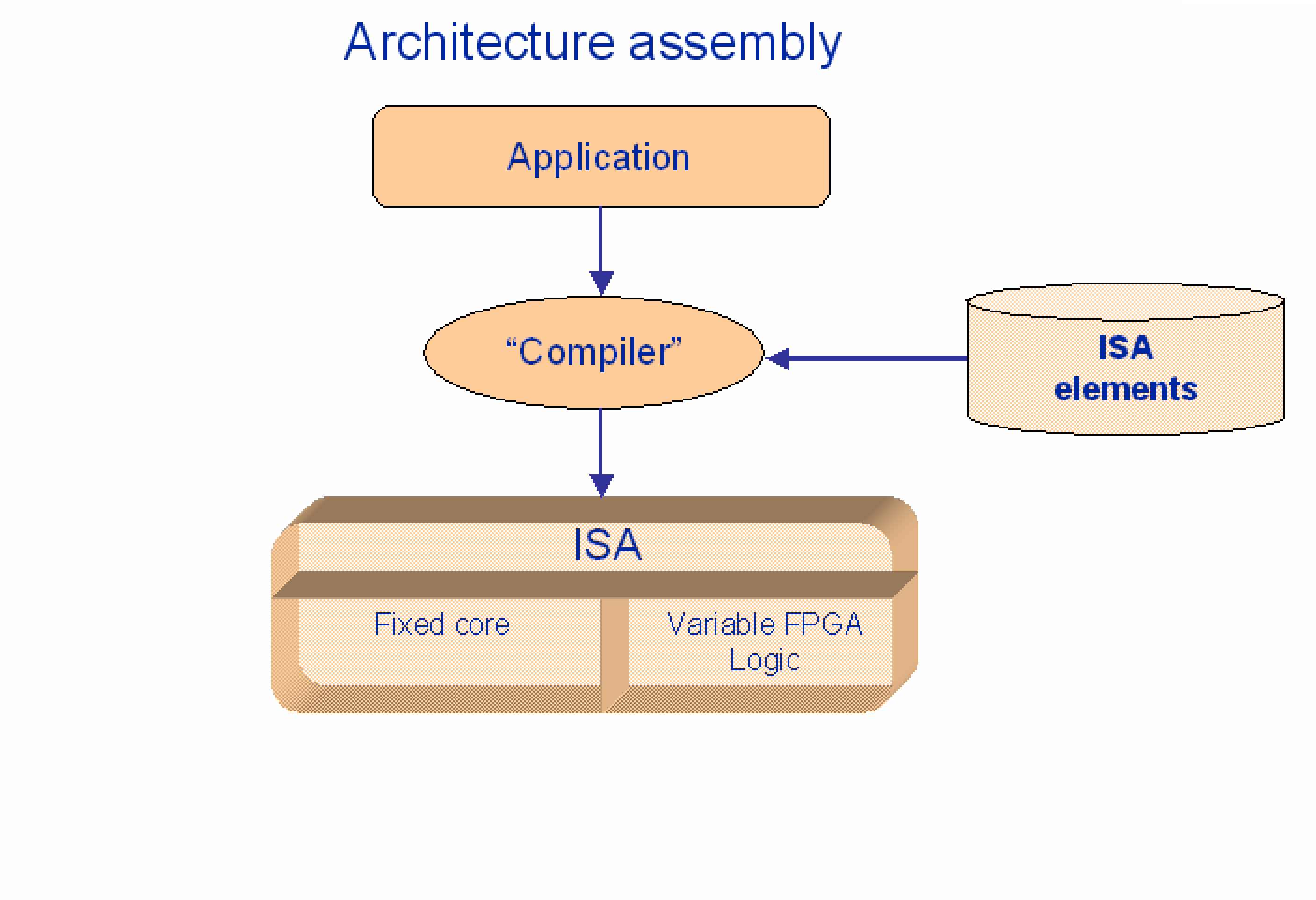
Functional units

ILP

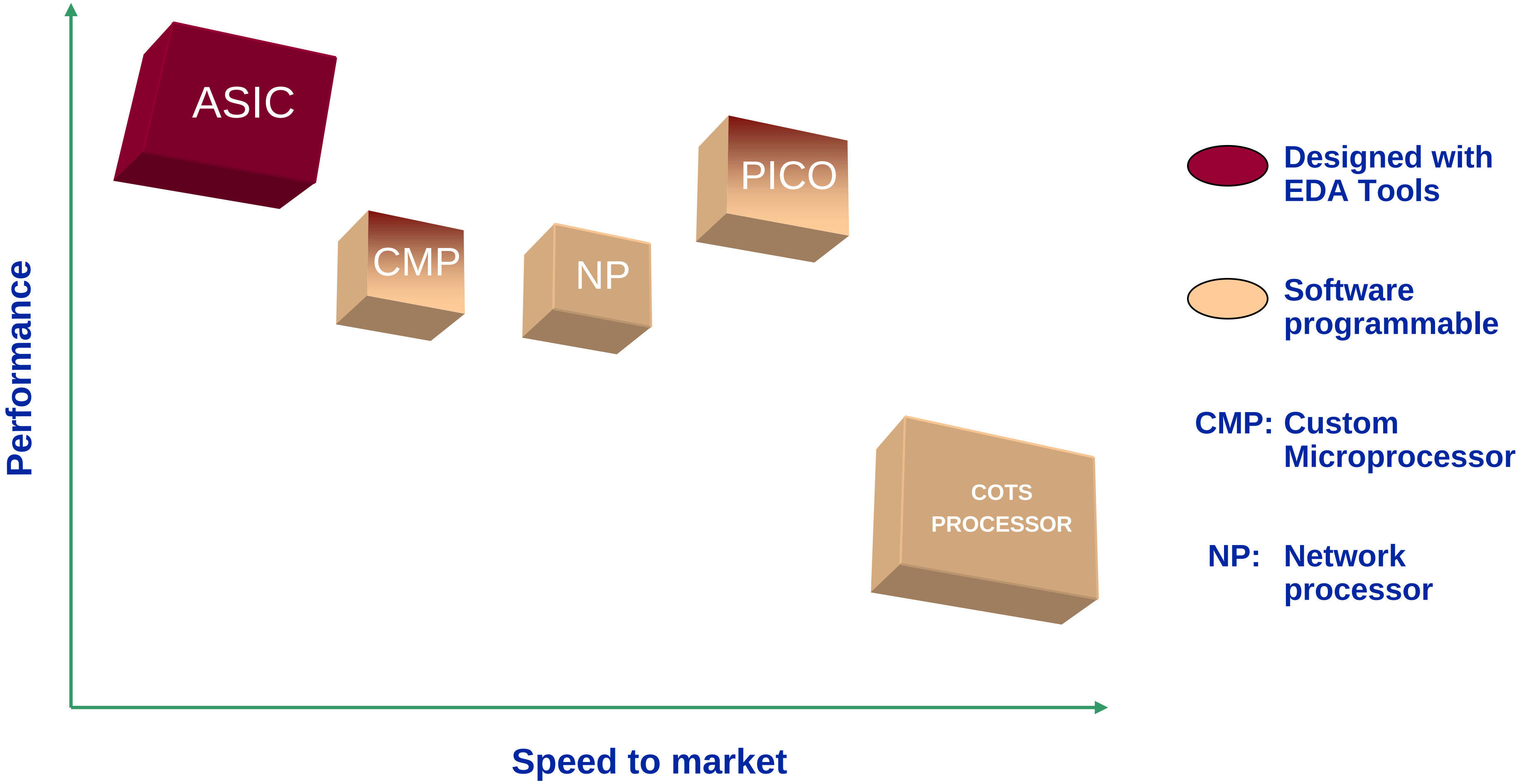
Adaptive EPIC execution model



Placing in Perspective



The Space for these Technologies



● Designed with EDA Tools

○ Software programmable

CMP: Custom Microprocessor

NP: Network processor

Krishna V. Palem, Lakshmi N. B. Chakrapani, Sudhakar Yalamanchili, “A Framework For Compiler Driven Design Space Exploration For Embedded System Customization”, *In Proceedings of the Ninth Asian Computing Science Conference, December 2004.*

Course Lecture/Tutorial

- Course Page: http://www.cs.rice.edu/~kvp1/

  follow the “Teaching” link
- Lectures will be posted weekly(prior to the week they are given)
- Lab assignments will be posted on the web page

Topics of the Course

- ISAs, Microprocessor ISAs, DSP ISAs, etc.
- EPIC Computing
- VHDL & FPGAs
- System-On-Chip
- Real-Time OS Design
- Communications and Network Solutions
- HW/SW Co-design
- Putting it all together
  - Adaptive EPIC
  - Flexible Instruction Processors
  - Architecture Synthesis
  - Architecture Assembly
- The course will provide coverage over a wide range of current technologies for the purpose of exploiting them in new & improved computing engines and will also involve guest lectures from specialists.

Textbook, Additional Reading and Supplements

- *Computers as components: Principles of Embedded Computing System Design* by *Wayne Wolf.* Morgan Kaufmann publication. ISBN 1-55860-541-X
- Engineering a Compiler by Keith Cooper and Linda Torczon, ISBN-10: 155860698X
- Embedded Computing: A VLIW Approach to Architecture, Compilers and Tools by Joseph A. Fisher, Paolo Faraboschi, Cliff Young. ISBN-10: 1558607668
- Optimizing Compilers for Modern Architectures: A Dependence-based Approach by Randy Allen and Ken Kennedy, ISBN-10: 1558602860
- Supplements will be in electronic form and posted on the web page

Lab/HW

- Homeworks will be a set of questions/problems to solve
- Labs contain a hands-on component and some high-level programming skills and will be based on the Trimaran system
  - The lab work could be completed during the lab time
- Homework is typically due a week from the day it is assigned
- All materials will be posted on the website

Grading

• Labs and homeworks:	30 %
• Term reports and Project:	50 %
• Final examination:	20 %