Designing Programmable Platforms: From ASIC to ASIP

MPSoC 2005

Heinrich Meyr

CoWare Inc., San Jose
and
Integrated Signal Processing Systems (ISS),
Aachen University of Technology, Germany

Agenda

- Facts & Conclusions

- Heterogeneous MPSoC
  - Energy Efficiency vs.Flexibility
  - How to explore the Design Space?

Agenda

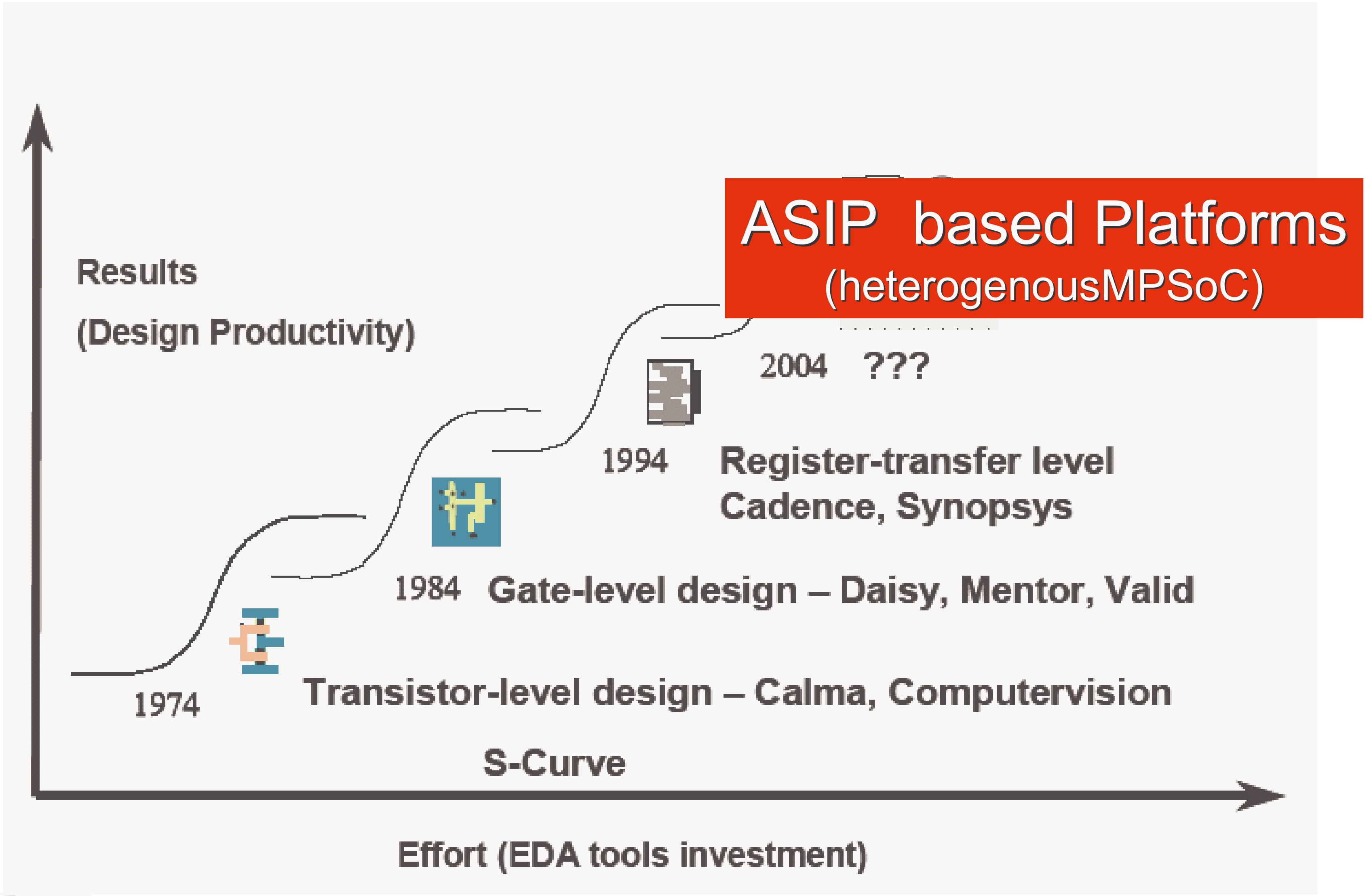
- ASIP Design

- Economics of SoC Development

- Conclusions

Facts & Conclusion

Core Proposition


ASIP based Platforms
(heterogenousMPSoC)
Results
(Design Productivity)
2004 ???
1994 Register-transfer level
Cadence, Synopsys
1984 Gate-level design – Daisy, Mentor, Valid
1974 Transistor-level design – Calma, Computervision
S-Curve
Effort (EDA tools investment)

Guiding Principle

„Form follows Function“

Mies van der Rohe

Agenda

- Facts & Conclusions
- Heterogeneous MPSoC
  - » Energy Efficiency vs.Flexibility
  - » How to explore the Design Space?

- ASIP Design
- Economics of SoC Development
- Conclusions

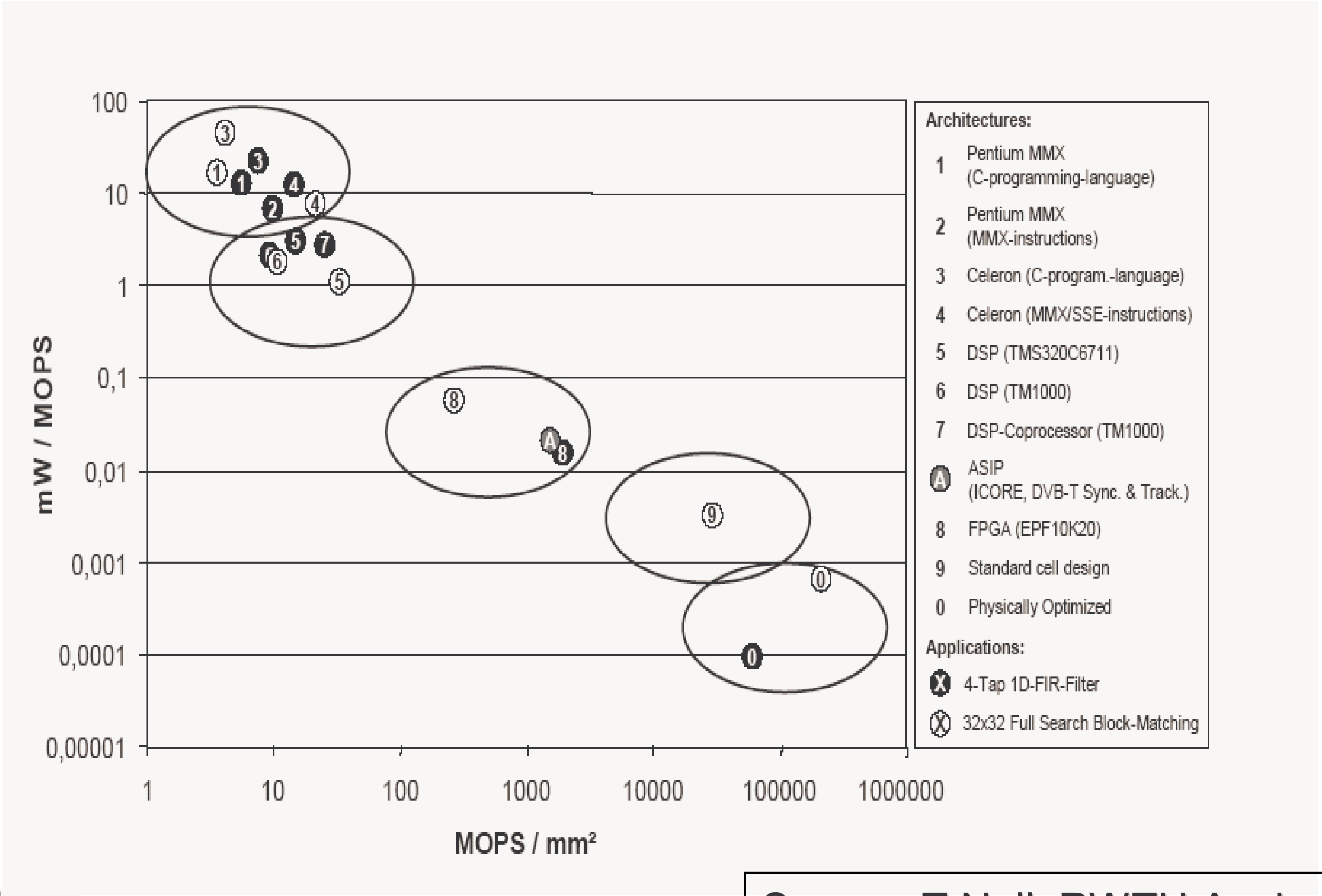
Trade-off between Flexibility and Energy -Efficiency

Architectural Objectives

Need more MOPS/Watt and MOPS/mm² to minimize the global performance measure for battery driven devices

Energy / decoded Bit = (Joule/Bit)

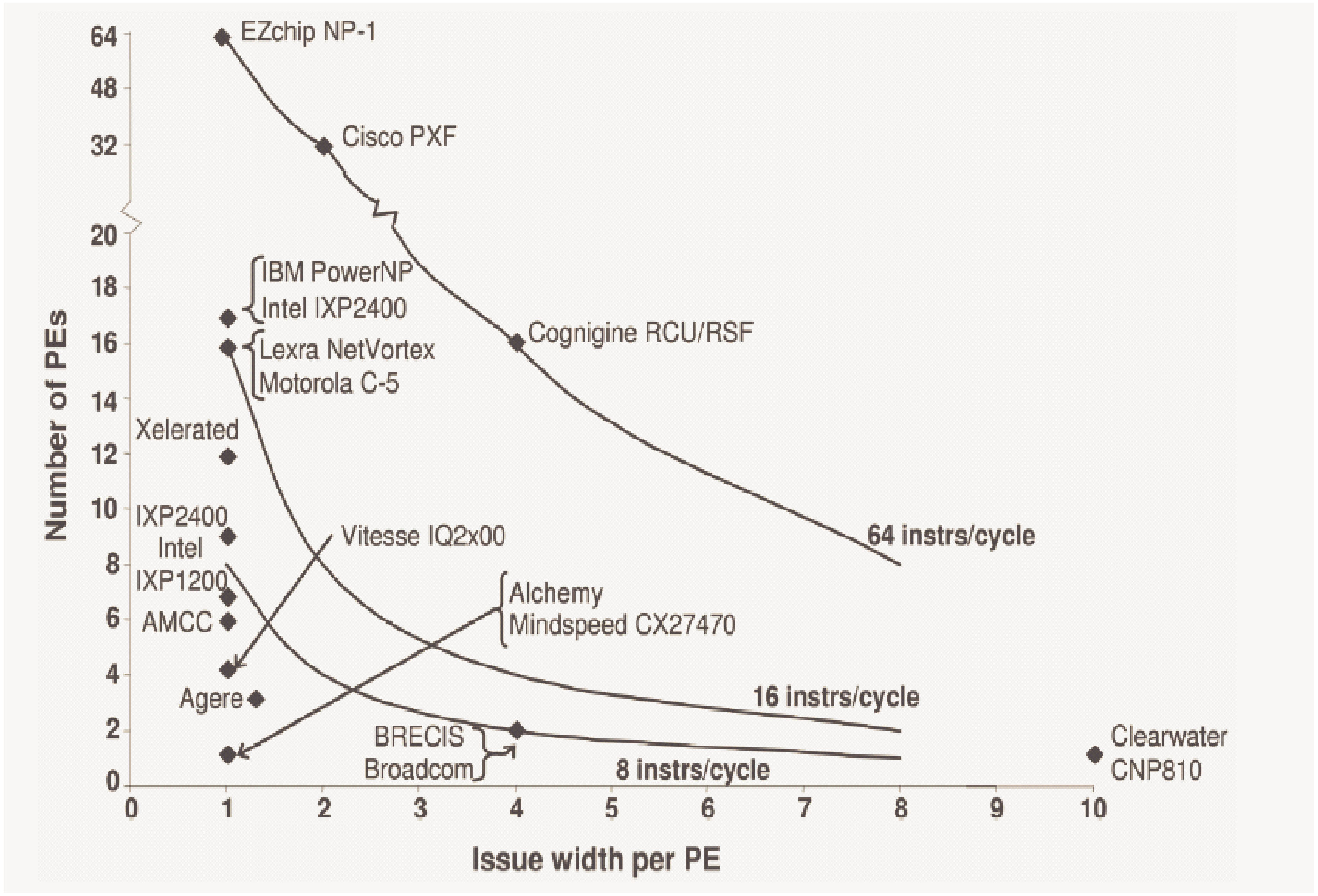
Computational Effiency vs. Flexibility



Source: T.Noll, RWTH Aachen

How to Explore the Design Space and design MPSoC´s?

Diversity of Network Procesors



Source: K.Keutzer, M.Griess, G.Martin, H.Meyr: „Designing ASIP: The MESCAL Methodology“, Springer 2005

The five elements of the MESCAL Methodology

1. Judiously apply benchmarking
2. Inclusively identify the architectural space
3. Efficiently describe and evaluate the ASIPs
4. Comprehensibly explore the design space
5. Sucessfully deploy the ASIP

Properties of the Task

- The signal/information processing task can be naturally partitioned
  - » Decoders
  - » Filters
  - » Channel estimator

MESCAL 1:
Judiously apply benchmarking

Use A Priori Knowledge of the Task

- The building blocks are loosely coupled
- The signal processing task is (mostly) cyclostationary

From Function to Algorithm Classes

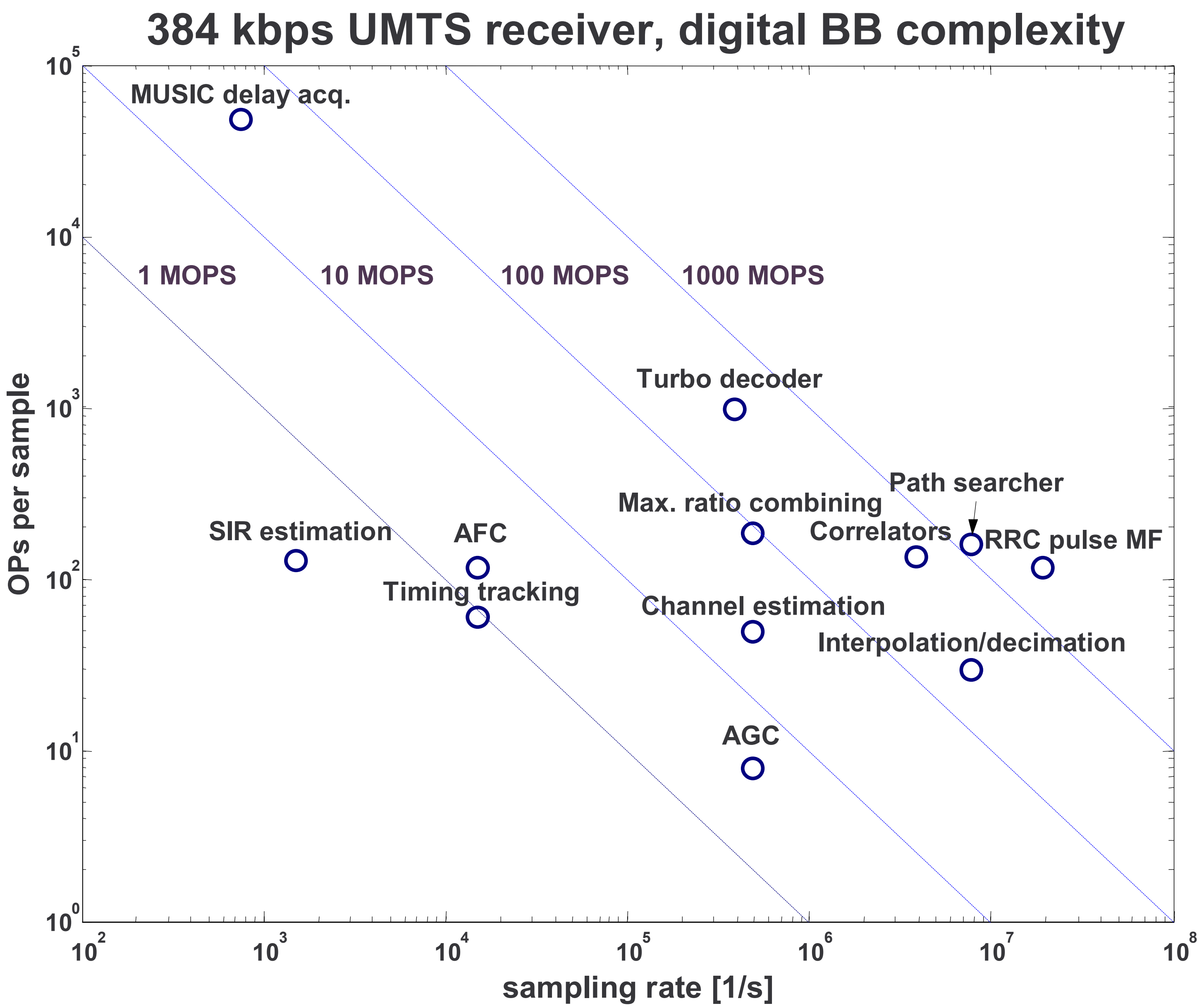
- Butterfly unit
  - Viterbi & MAP decoder
  - MLSE equalizer
- Eigenvalue decomposition (EVD)
  - Delay acquisition (CDMA)
  - MIMO Tx processing
- Matrix-Matrix & Matrix-Vector Multiplication
  - MIMO processing (Rx & Tx)
  - LMMSE channel estimation (OFDM & MIMO)
- CORDIC
  - Frequency offset estimation (e.g. AFC)
  - OFDM post-FFT synchronization (sampling clock, fine frequency)
- FFT & IFFT (spectral processing)
  - OFDM
  - Speech post processing (noise suppression)
  - Image processing (not FFT but DCT)

Algorithmic Descriptors

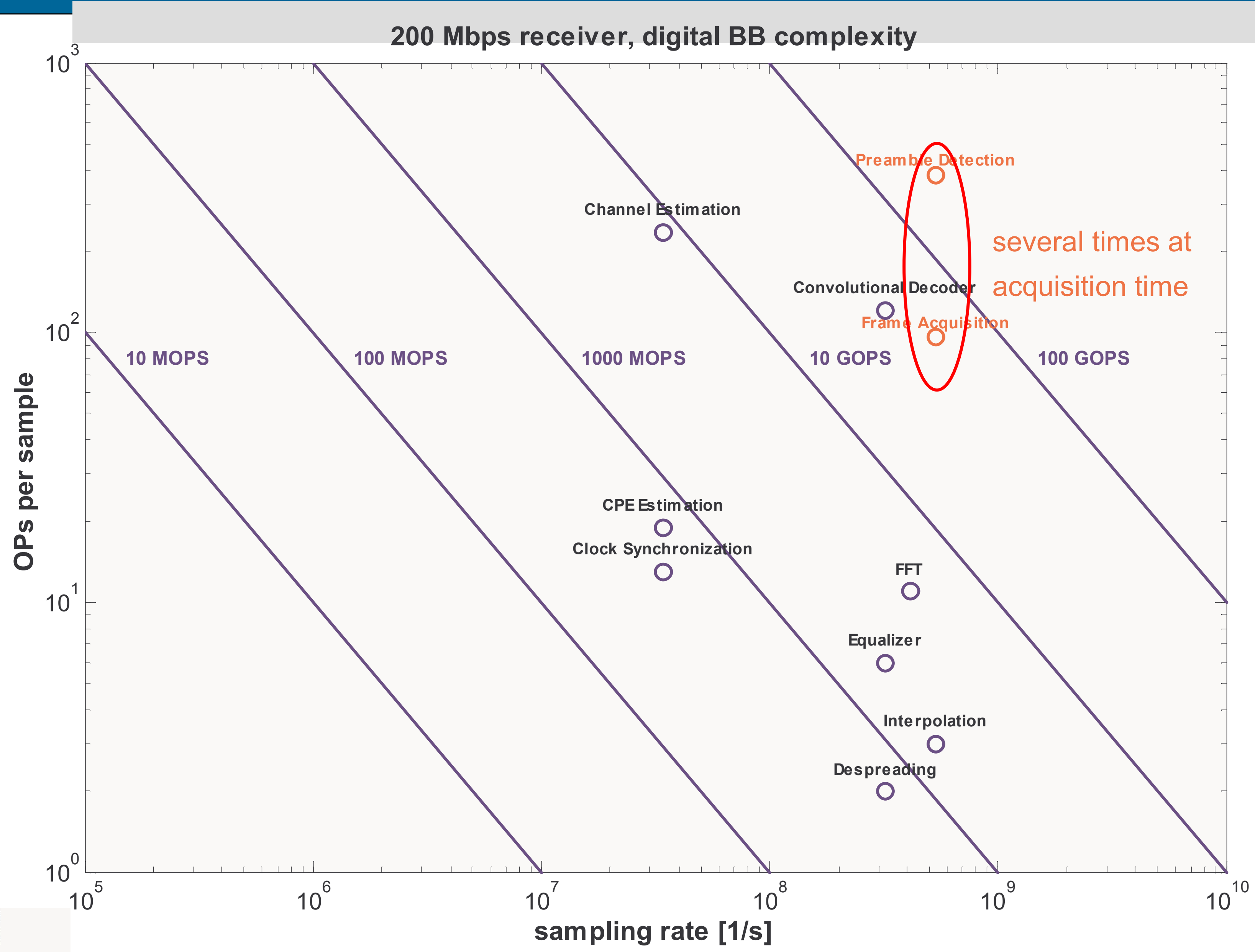
- Clock rate of processing elements (1/Tc)
- Sampling rate of the signal (1/Ts)
- Algorithm characteristic
  - Complexity (MOPS/sample)
  - Computational characteristic
    - Data flow
      - Data locality
      - Data storage
      - Parallelism
    - Control flow
- Connectivity of algorithms
  - Spatial
  - Temporal

Basic Blocks: Algorithm Types

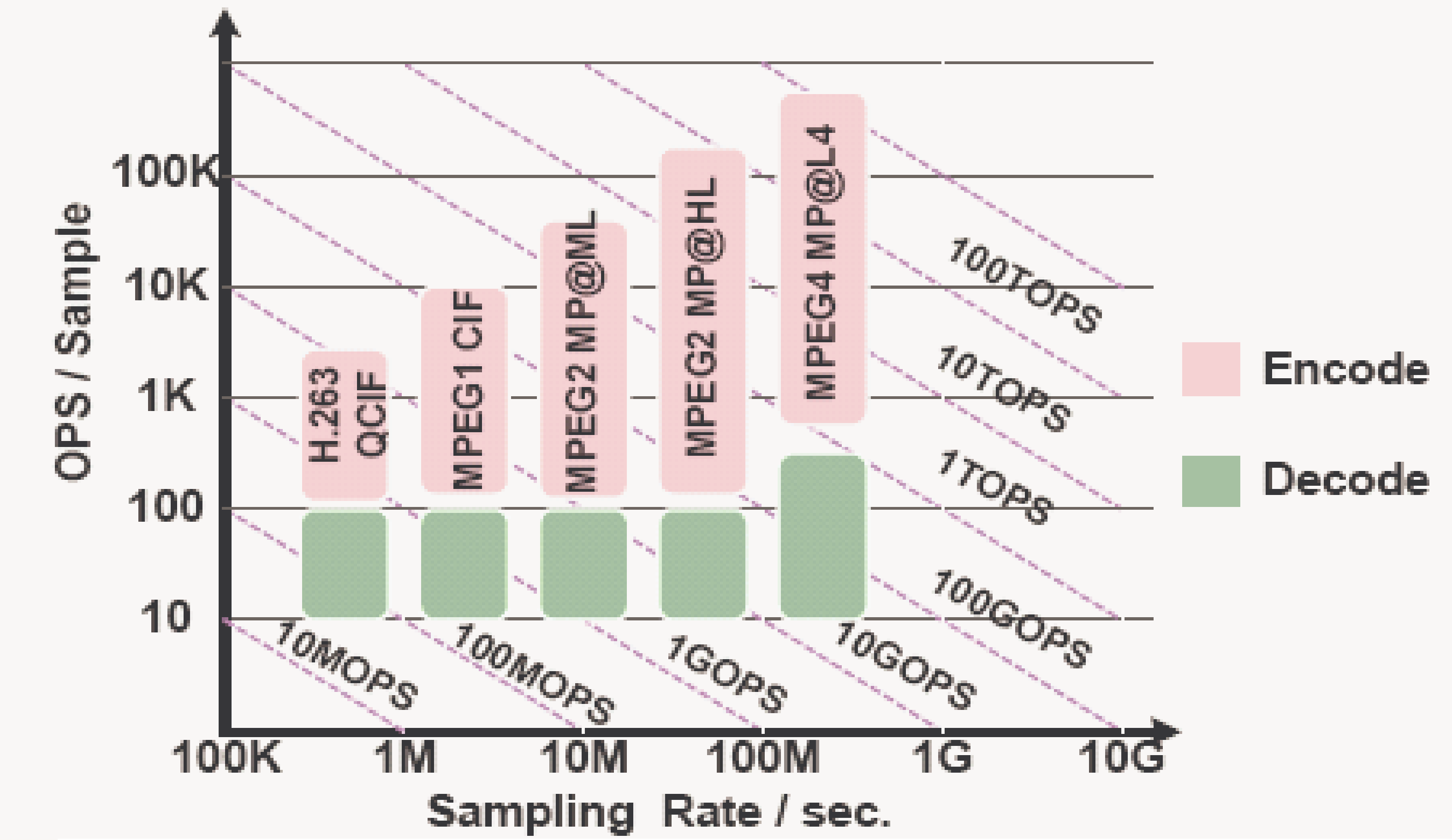
384 kbps UMTS Receiver BB Complexity



200 Mbps UWB Receiver BB Complexity I



Computational complexity of previous and recent video coding standards



Source: L.G Chen, SIPS 2003

System Functional Requirements for Handheld

Massive Parallelism required in the foreseeable future

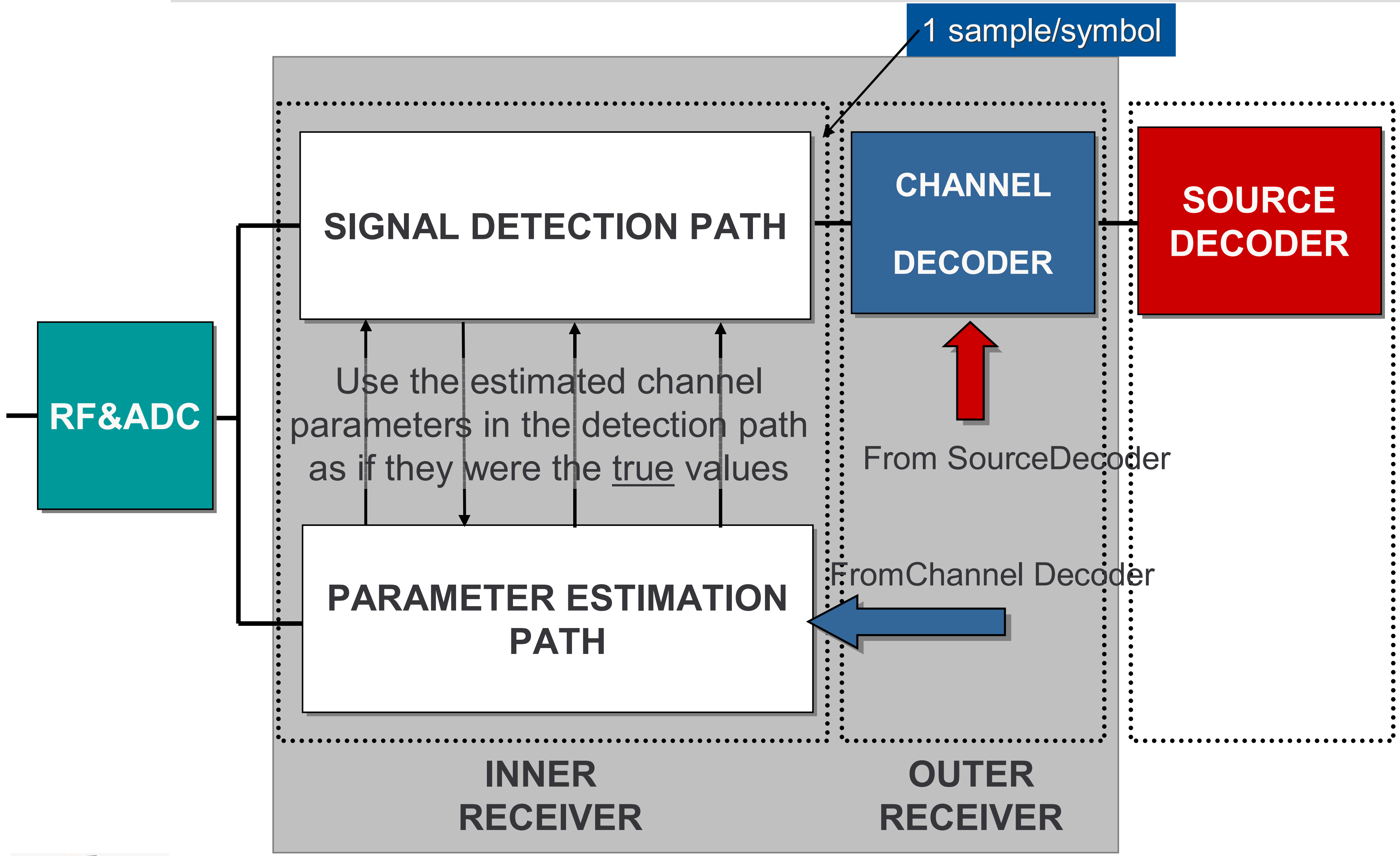
	2003	2009	2013
Frequency (MHz)	300	600	1500
Giga Operations	0,3	14	2458
Operations per Cycle	1	23	1638

Source: International Technology Roadmap for Semiconductors (ITRS, TX 2003)

Lessons Learned

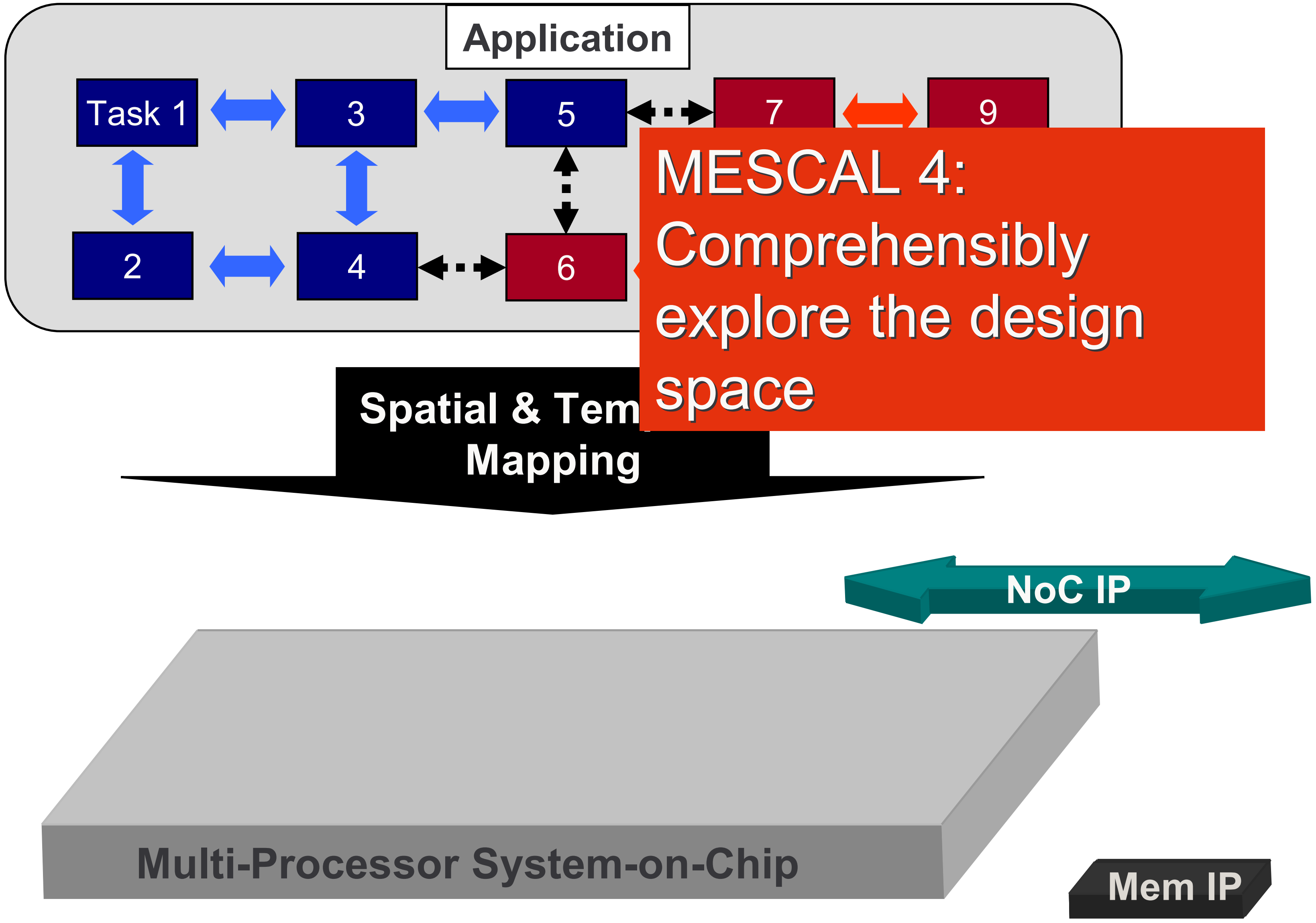
- Virtual Prototype (Product) of utmost importance
  - » Early customer interaction
  - » Debugging
  - » Verification&Validation
- Product Differentiator
  - » 80% of Area and Power Consumption in the inner receiver (Algorithm and Architecture Design)
  - » 10-15% of Area and Power Consumption in Decoder (Architecture Design)
  - » 5% of Area and Power Consumption in the ARM (But major portion of cost is SW/Protocol implementation)

Canonical Receiver Model



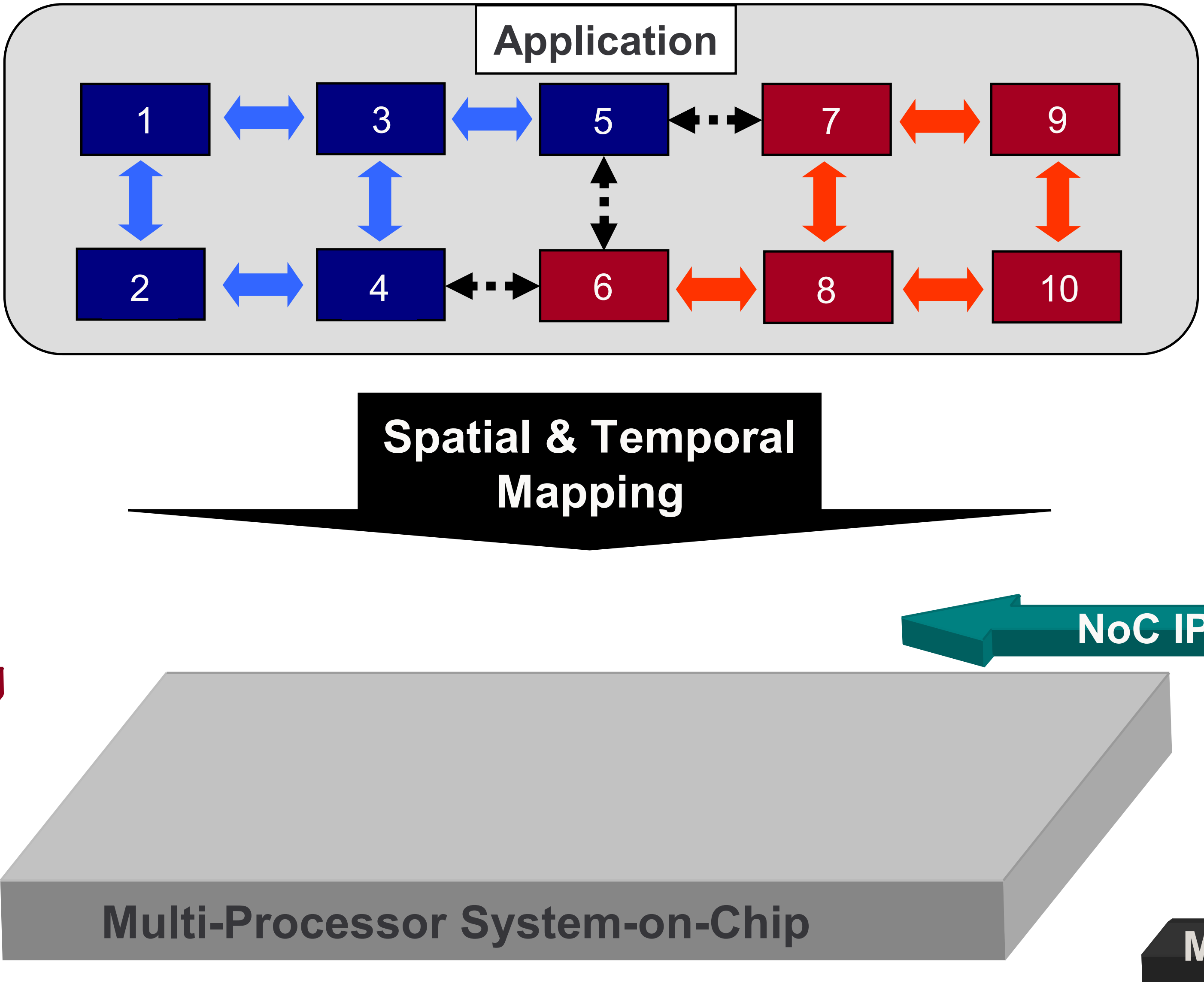
H.Meyr et al., " Digital Communication Receiver", J.Wiley 1998

Design Task: Spatial and Temporal Mapping



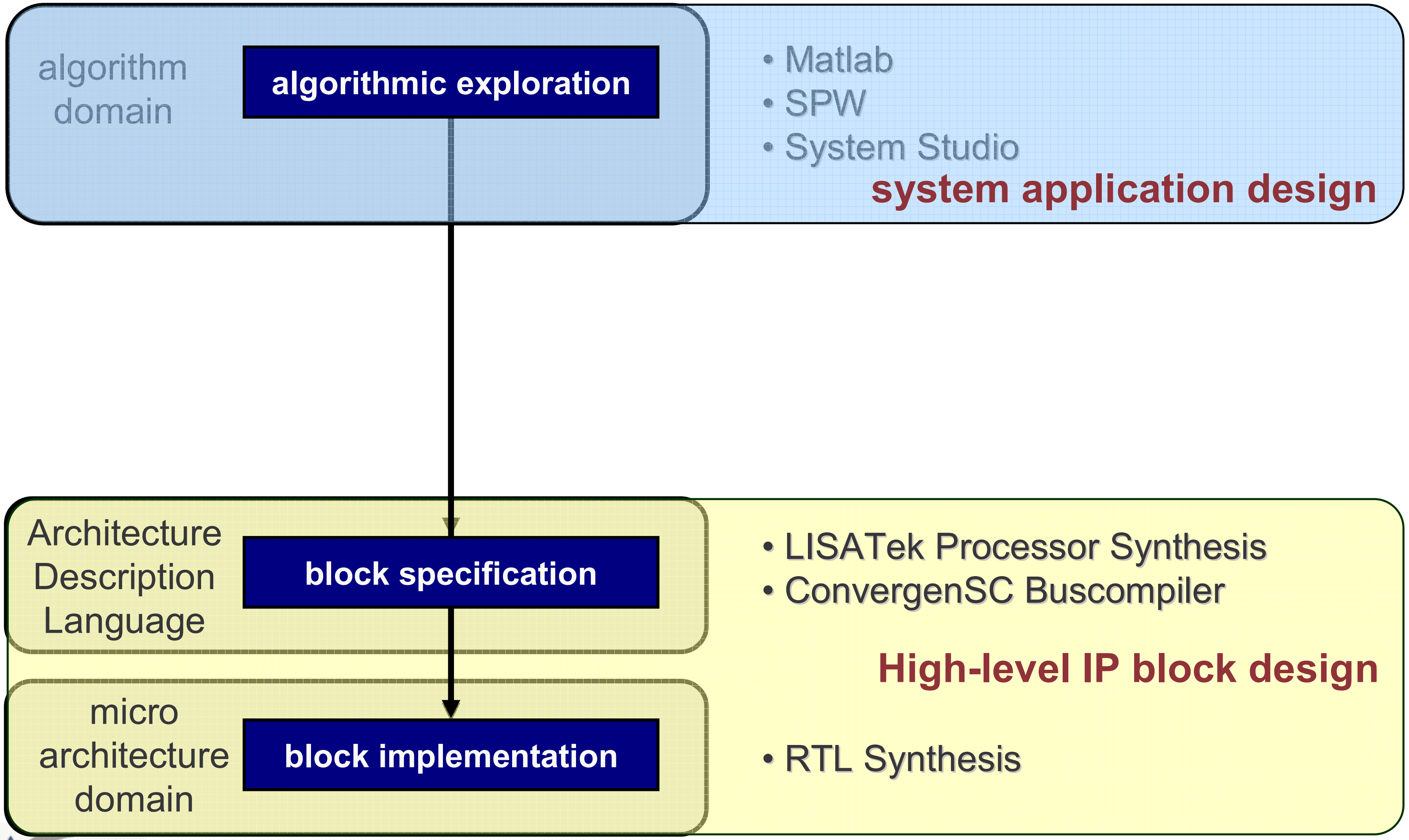
MESCAL 4: Comprehensibly explore the design space

Design Task: Spatial and Temporal Mapping



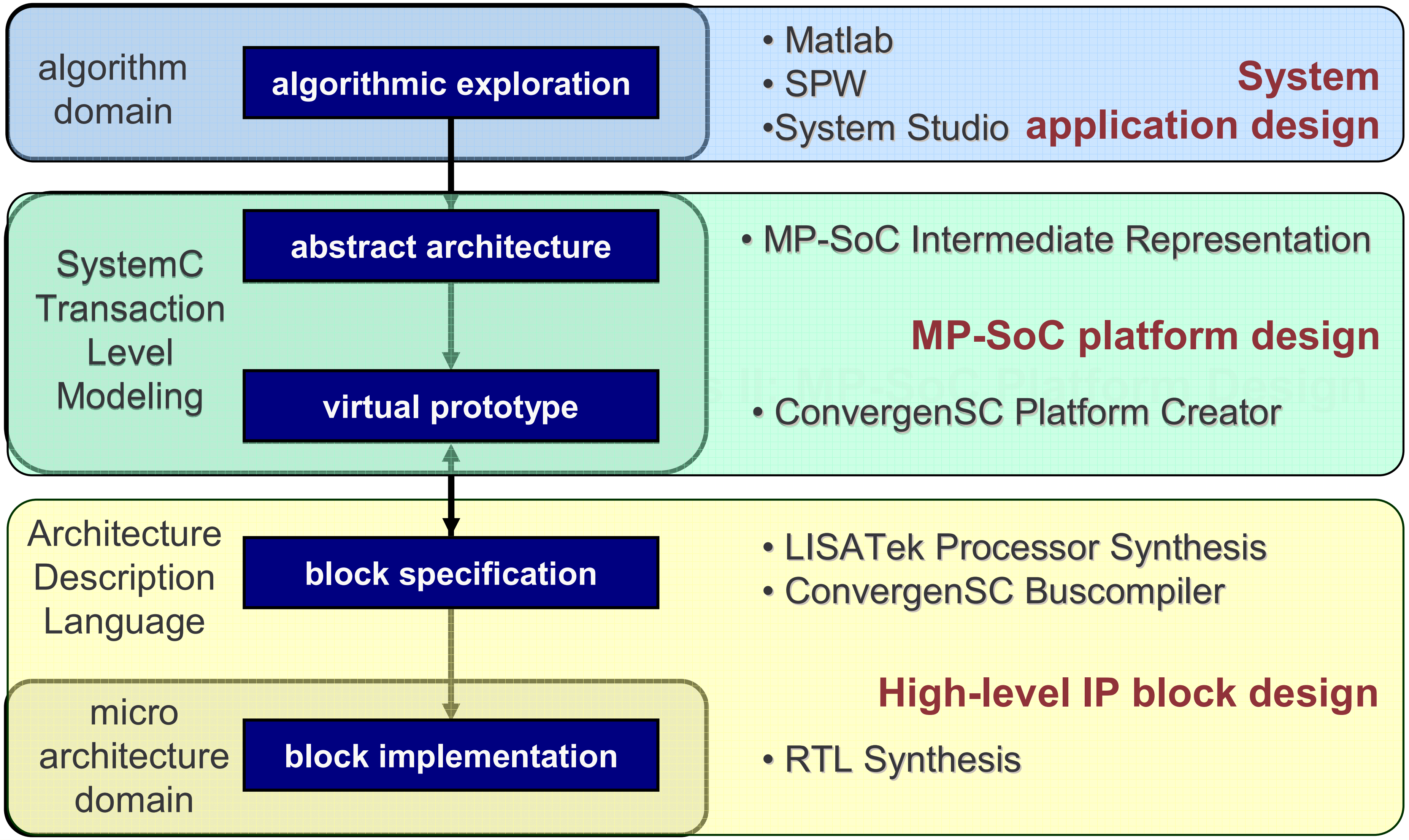
Enabling MP-SoC Design

System Level Tools I: Application & IP Creation



algorithm domain
algorithmic exploration
• Matlab
• SPW
• System Studio
system application design
Architecture Description Language
block specification
micro architecture domain
block implementation
• LISATek Processor Synthesis
• ConvergenSC Buscompiler
High-level IP block design
• RTL Synthesis

System Level Tools I: Application & IP Creation

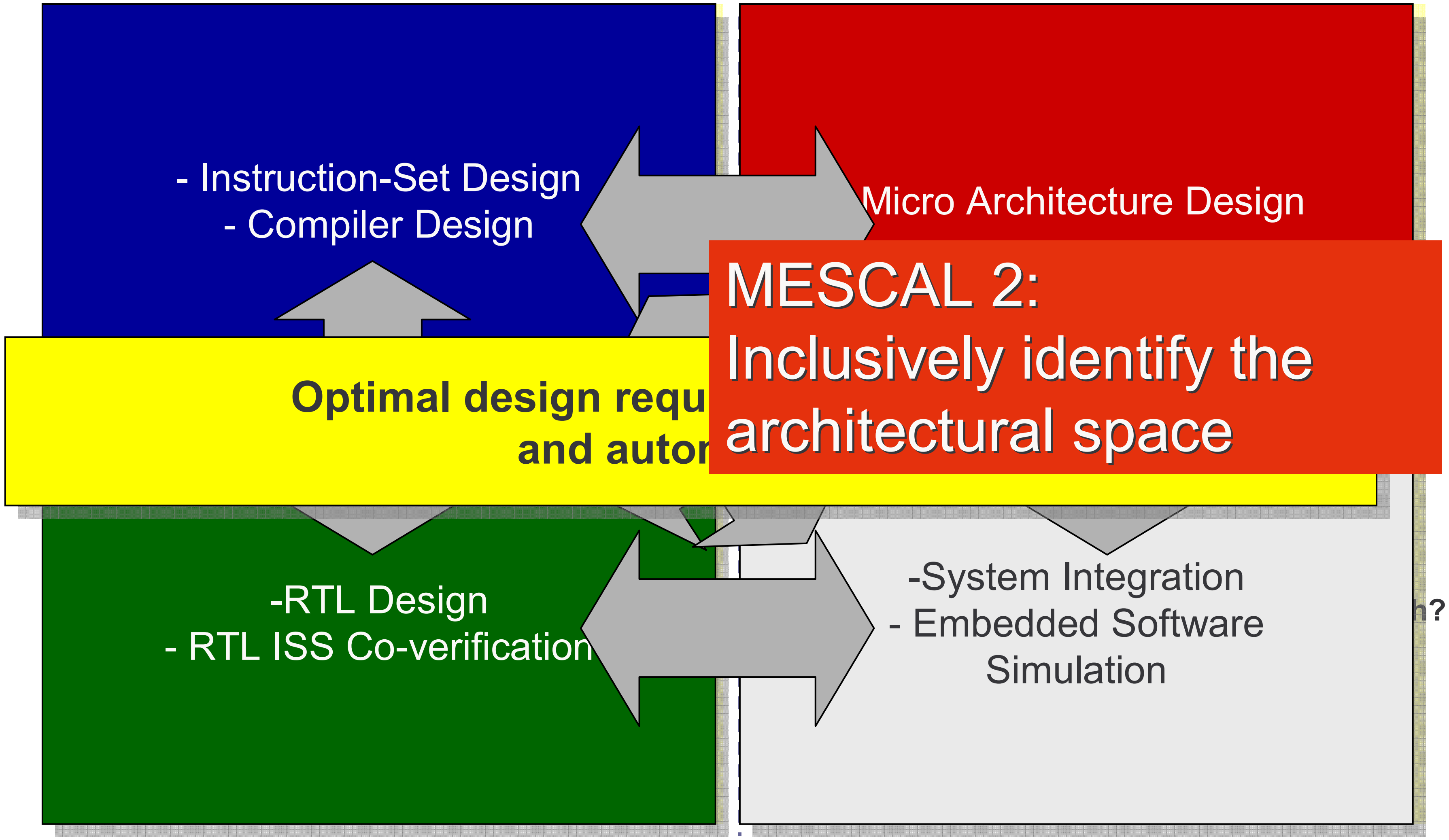


algorithm domain
algorithmic exploration
• Matlab
• SPW
•System Studio
System application design
SystemC Transaction Level Modeling
abstract architecture
virtual prototype
• MP-SoC Intermediate Representation
MP-SoC platform design
• ConvergenSC Platform Creator
Architecture Description Language
block specification
micro architecture domain
block implementation
• LISATek Processor Synthesis
• ConvergenSC Buscompiler
High-level IP block design
• RTL Synthesis

Agenda

- Facts & Conclusions
- Heterogeneous MPSoC
  - » Energy Efficiency vs.Flexibility
  - » How to explore the Design Space?
- **ASIP Design**
- Economics of SoC Development
- Conclusions

Processor Design Space


- Instruction-Set Design
- Compiler Design
Micro Architecture Design
-RTL Design
- RTL ISS Co-verification
-System Integration
- Embedded Software Simulation
h?

Optimal design requ

and autor

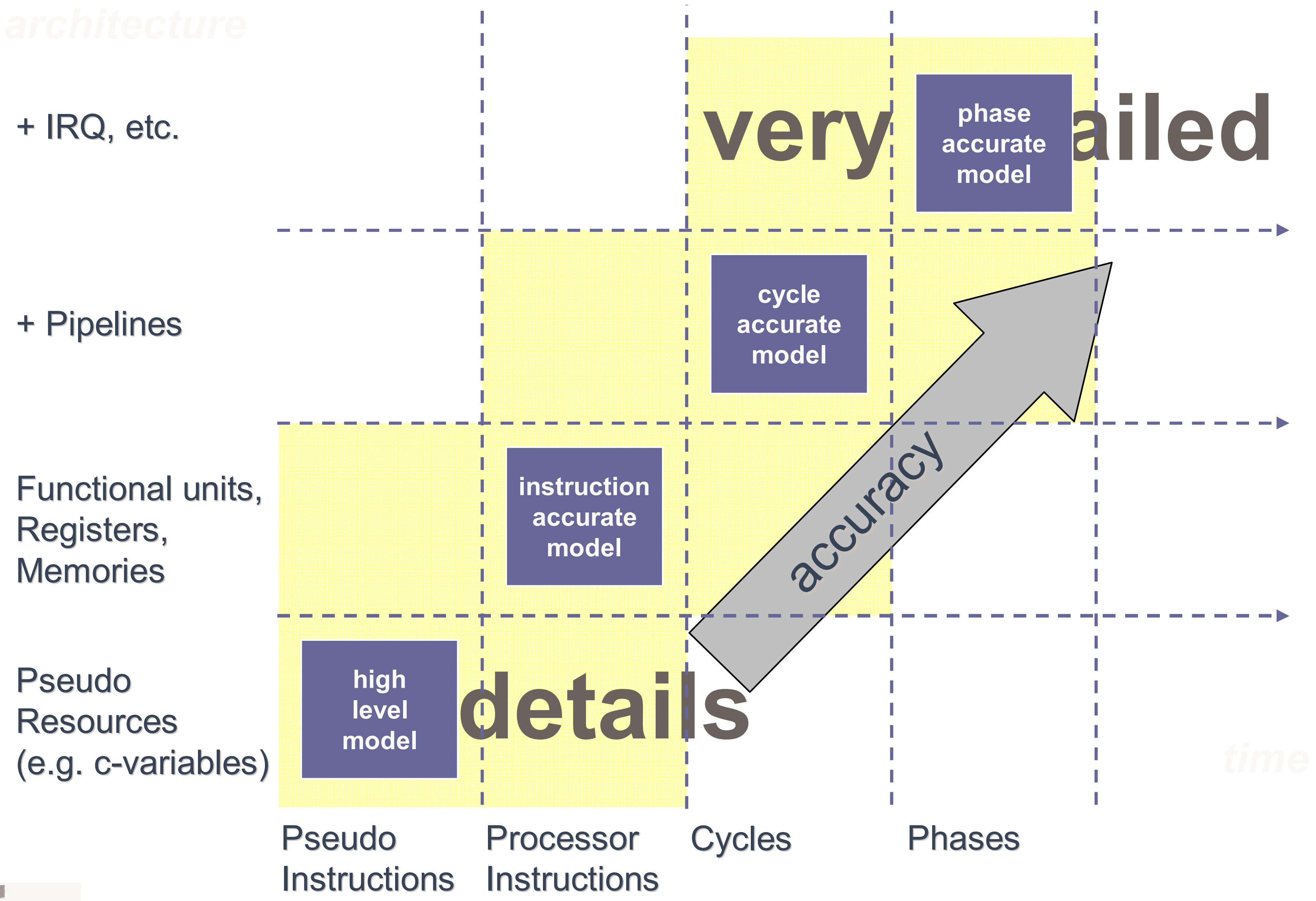
MESCAL 2:
Inclusively identify the architectural space

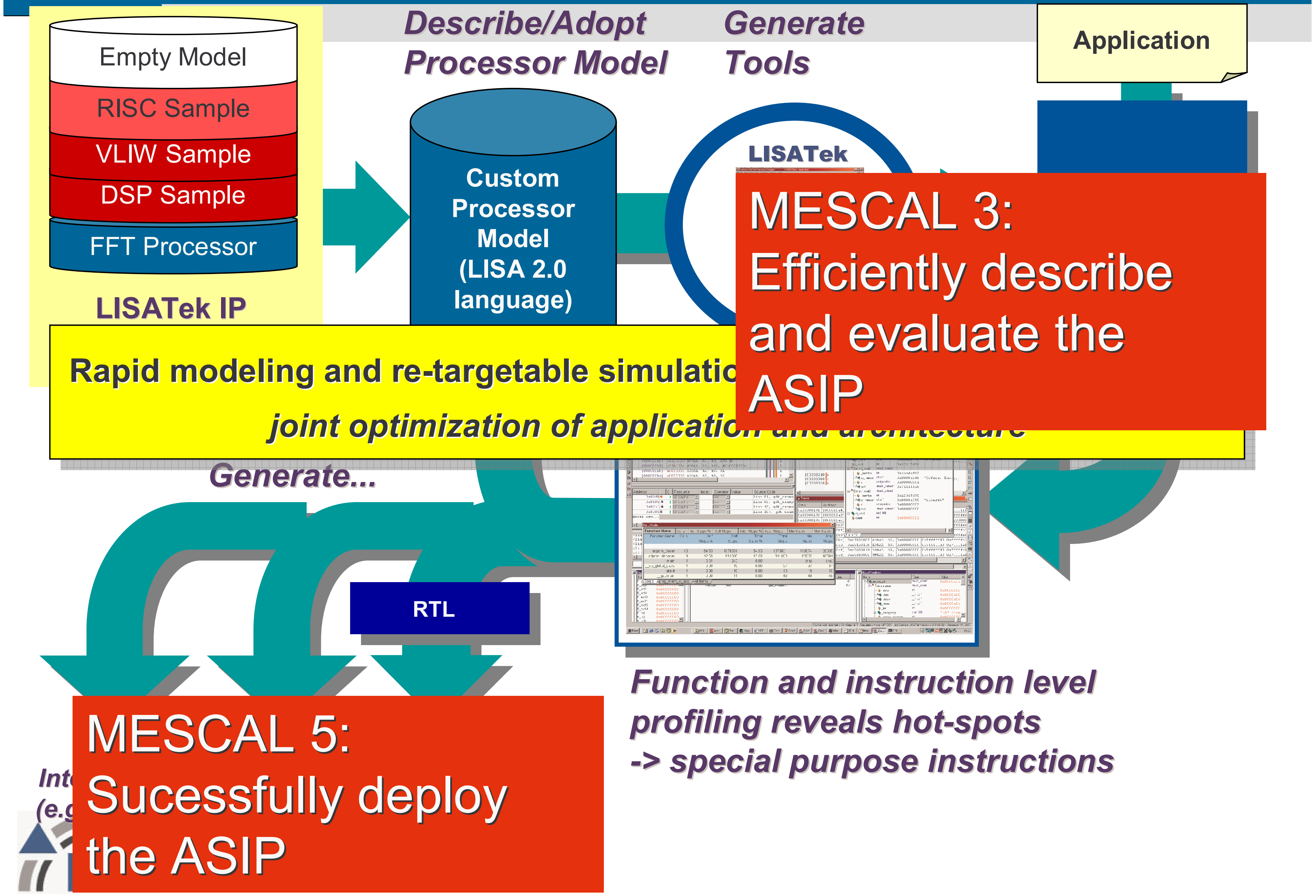
Architecture Description Language based Processor Design

- The purpose of an **architecture description language** (e.g LISA) is:
  - To allow for an iterative design to efficiently explore architecture alternatives
  - To jointly design "hip communication
  - To automatically implementation)
  - To automatically generate tools
    - Assembler ,Linker, Compiler, Simulator, co-simulation interfaces
- From a **single** model at various level of temporal and spatial abstraction

MESCAL 3:
Efficiently describe the ASIP

LISA 2.0 - Abstraction Levels





Rapid modeling and re-targetable simulatio

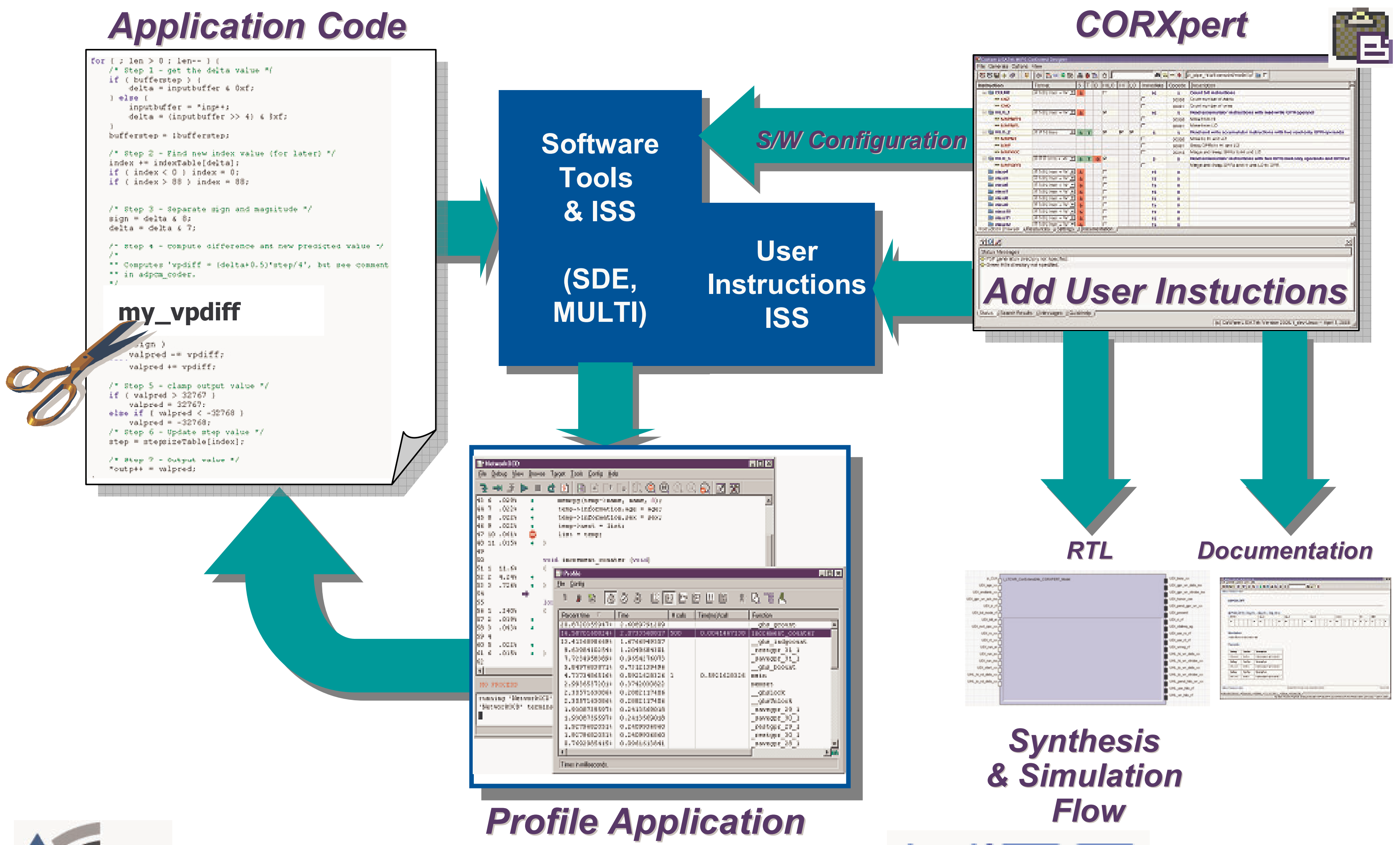
joint optimization of applicatio

MESCAL 3:
Efficiently describe and evaluate the ASIP

Function and instruction level profiling reveals hot-spots -> special purpose instructions

MESCAL 5:
Sucessfully deploy the ASIP

CORXpert™ - Automating MIPS CorExtend™



```
for ( ; len > 0 ; len-- ) {
    /* Step 1 - get the delta value */
    if ( bufferstep ) {
        delta = inputbuffer & 0xf;
    } else {
        inputbuffer = *inp++;
        delta = (inputbuffer >> 4) & 0xf;
    }
    bufferstep = !bufferstep;

    /* Step 2 - Find new index value (for later) */
    index += indexTable[delta];
    if ( index < 0 ) index = 0;
    if ( index > 88 ) index = 88;


    /* Step 3 - Separate sign and magnitude */
    sign = delta & 8;
    delta = delta & 7;

    /* Step 4 - Compute difference and new predicted value */
    /*
    ** Computes 'vpdiff = (delta+0.5)*step/4', but see comment
    ** in adpcm_coder.
    */
```

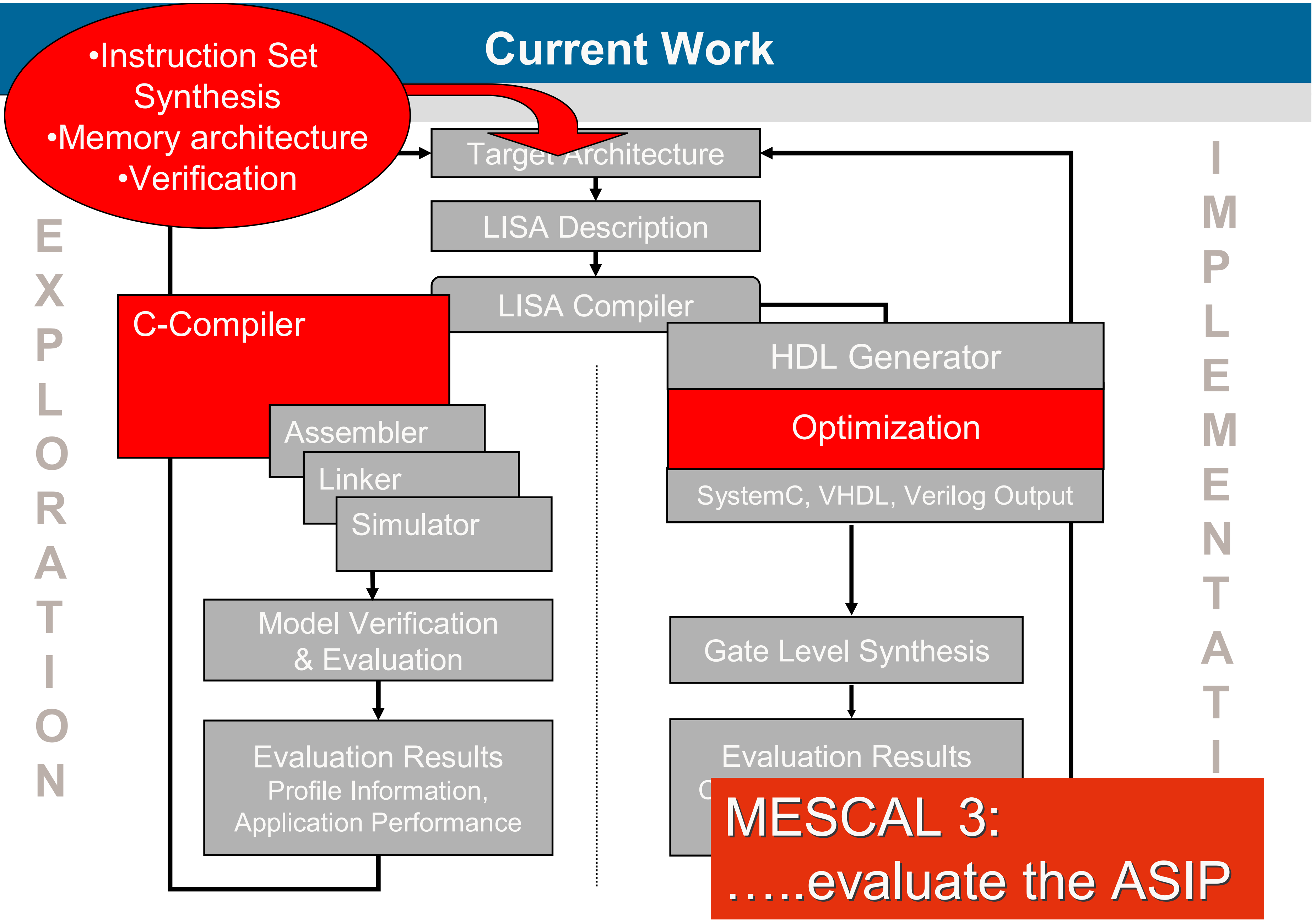
my_vpdiff

```
      sign )
        valpred -= vpdiff;
        valpred += vpdiff;

    /* Step 5 - clamp output value */
    if ( valpred > 32767 )
        valpred = 32767;
    else if ( valpred < -32768 )
        valpred = -32768;
    /* Step 6 - Update step value */
    step = stepsizeTable[index];

    /* Step 7 - Output value */
    *outp++ = valpred;
```

Current Work



MESCAL 3:
…..evaluate the ASIP

Examples

ASDSP FPGA Implementation

Myjung Sunwoo, Ajiou University,

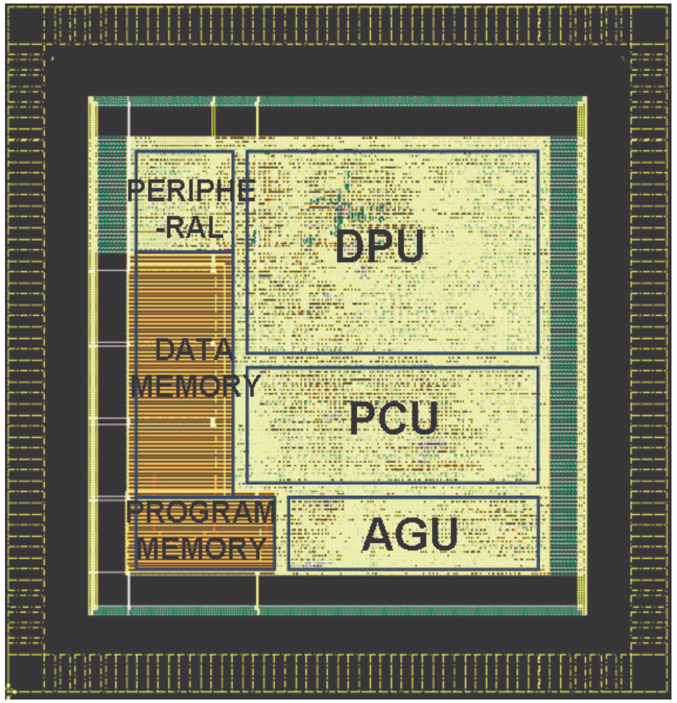
ASDSP Core Design

✓**SEC 0.18um Synthesis**

- Gate : 77,000
- Program Memory : 4 Kbyte, Data Memory : 8 Kbyte
- Frequency : 290MHz
- Power consumption : 0.87W (3mW/MHz)

FPGA Implementation

✓ iProve Xilinx xc2v6000



Support the Special Instruction Set for FFT Operation and the BMU Instruction
Improve the Performance for OFDM Communication

The ICORE

A low-power ASIP for Infineon DVB-T 2nd generation Single-Chip Receiver:

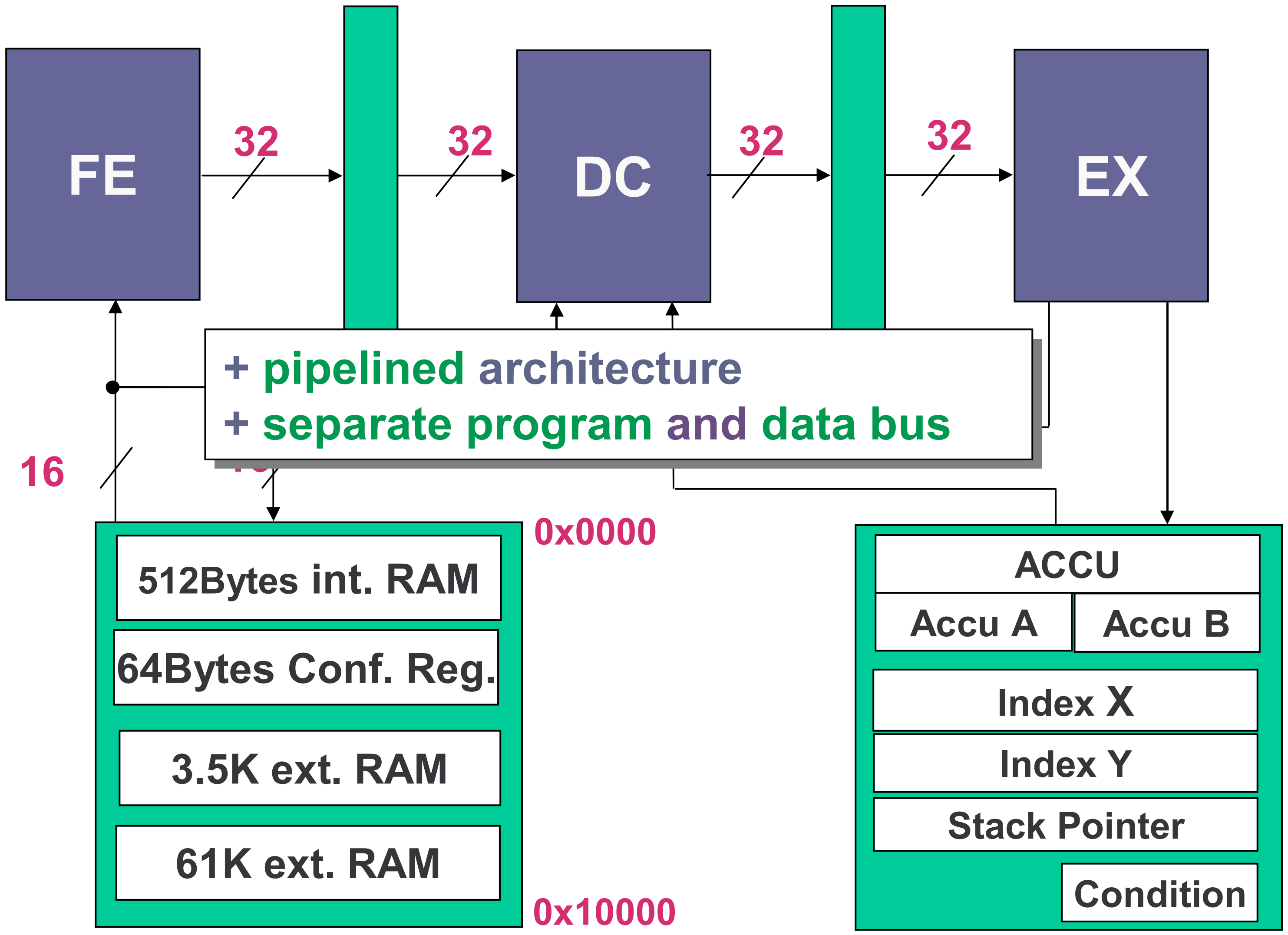
- ASIP for DVB-T acquisition and tracking algorithms (sampling-clock-synchronization, interpolation / decimation, carrier frequency offset estimation)
- Harvard Architecture
- 60 mostly RISC-like Instructions & Special Instructions for CORDIC-Algorithm
- 8x32-Bit General Purpose Registers, 4x9-Bit Address Registers
- 2048x20-Bit Instruction ROM, 512x32-Bit Data Memory
- I2C Registers and dedicated interfaces for external communication

The Motorola M68HC11 Architecture

Architecture Overview

- **M68HC11 CPU Architecture : Hot spots**
  - » 8-bit micro-controller.
    - » Harvard Architecture
  - » 7 CPU Registers.
  - » 6 different Addressing Modes.
  - » Shared data and program bus. : stalled data access
  - » Instruction width : 8,16, 24, 32, 40 : multi-cycle fetch
  - » 8-bit opcode : 181 instructions
  - » Clock speed : ~200 MHz
  - » Performance : : non-pipelined
  - » Area : 15K to 30K (DesignWare® Library)

Architecture Development with LISA



+ pipelined architecture
+ separate program and data bus

Results

•Area

< 23k gates

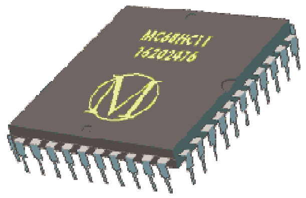
•Clock speed

~ 200 MHz

•Execution time speed up

62 % for spanning tree application

•Mapped onto Xilinx FPGA



Architecture Development with LISA

•Studying the architecture	4 days
•Basic architecture modifications	2 days
•Grouping and coding of the instructions	1 day
•Writing the LISA model	
-basic syntax and coding	4 days
-behavior section	6 days
•Validation	4 days
•HDL Generation	2 days
Total	23 days

Agenda

- Facts & Conclusions
- Heterogeneous MPSoC
  - » Energy Efficiency vs.Flexibility
  - » How to explore the Design Space?

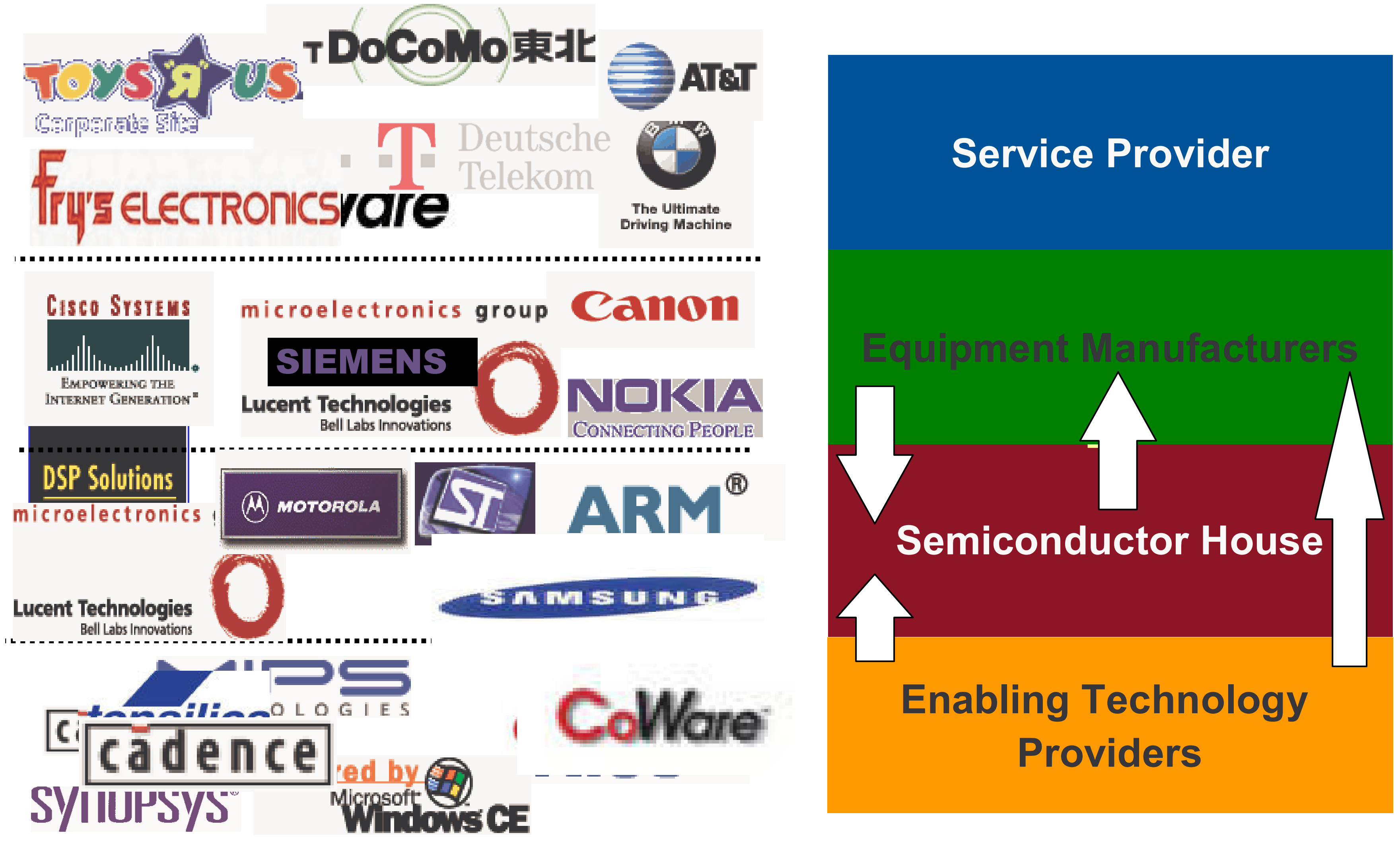
Agenda

- ASIP Design
- **Economics of SoC Development**
- **Conclusions**

The future : MPSoC

Opportunity: For Whom ?

Food Chain in the Wireless Market



MPSoC Characteristics

- Growth potential :
  - » Functionality increases qualitatively with time
    - ➔ Newcomer chases a moving target
- System property:
  - » The whole is more than the sum of the parts
    - ➔Newcomer needs to build up expertise in various fields and ….needs to learn how to manage the interaction between them

The Human Element

Building and managing an inte
engineering te

1. Algorithm Desi gn Competence
2. Computer cts
3. Syst
4.

No psycho babble: It is the most critical element

ISS

Agenda

- Facts & Conclusions
- Heterogeneous MPSoC
  - » Energy Efficiency vs.Flexibility
  - » How to explore the Design Space?

Agenda

- ASIP Design
- Economics of SoC Development
- Conclusions

Summary

- We need to understand the process of engineering a complex SoC as an “Apollo” project

Inc ontent- but How?

Thank You