Multi-core platforms are a reality... but where is the software support?

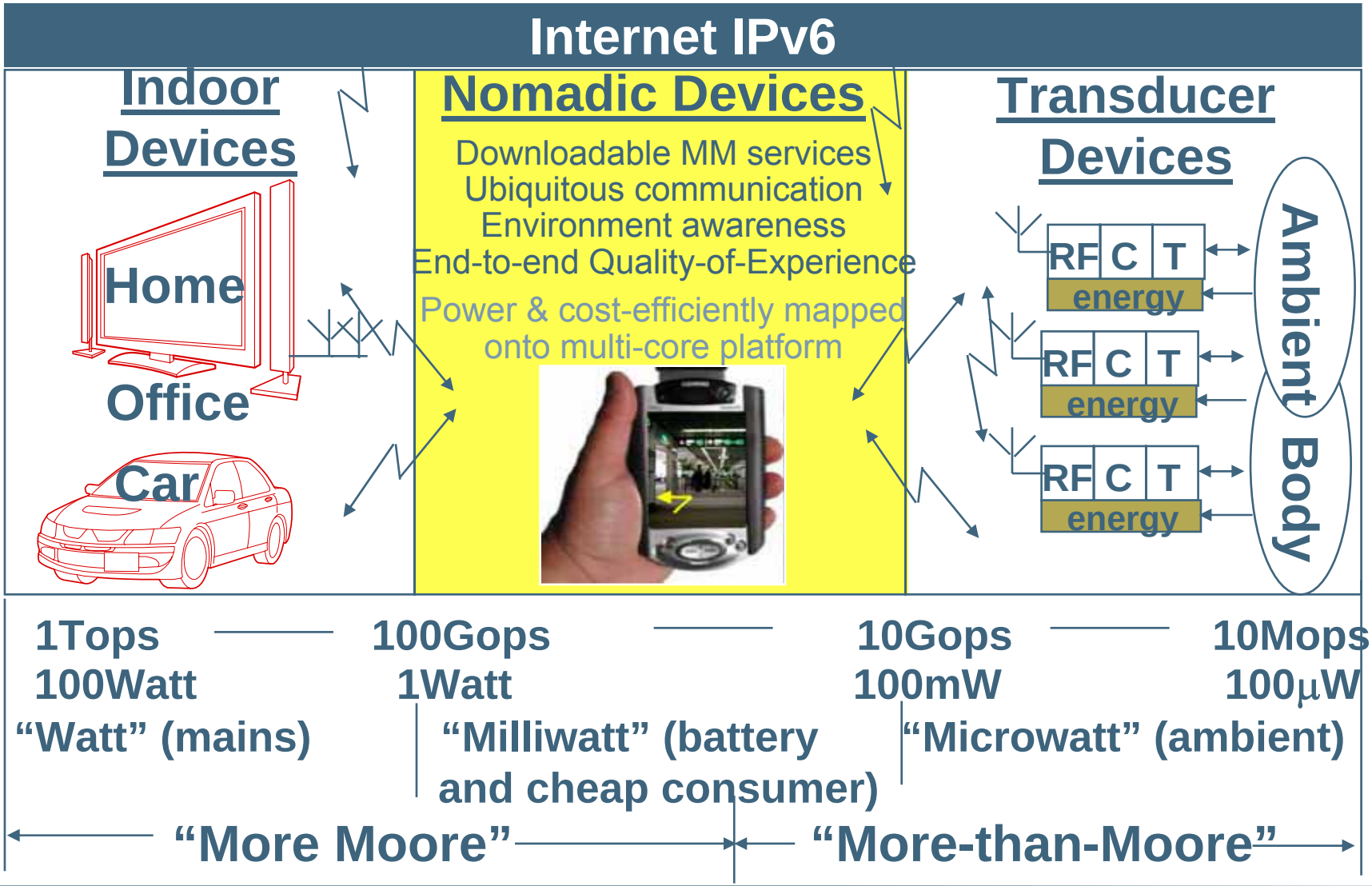
Rudy Lauwereins
Vice-President IMEC
Prof. Katholieke Universiteit Leuven
Director Institute for BroadBand Technologies IBBT

imec

Content

- Introduction: situating the application domain
- Architectural evolution
- Design flow
- Case study 1: software defined radio
- Case study 2: multi-format multimedia codec
- Conclusion

Vision: Multi-core platforms are essential for cheap consumer products and battery operated devices



Multi-core platforms: the devil is in the software heaven… and in the hell of physics

ystem Design

ESL

Platform Design

Uncertainty Metrics!

IP Creation

DFM

CMOS Scaling

Software centric system specs
7th heaven of software

Architectural gap

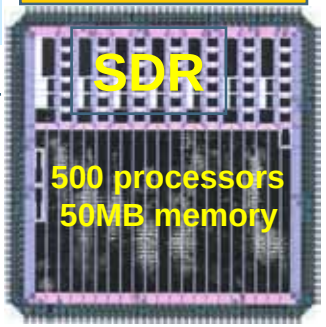
500M tr Platform architecture

xx nm Si IP Blocks

Physical gap

Nano-scale hell of physics

Middleware API's RTOS



1 Watt

CHALLENGE:
Create methods, tools, platform for mapping multiple SW applications on *one* flexible yet energy efficient and scaling tolerant platform

With daunting properties:

- power efficient (>100 GOPS/W)
- cheap (10€)
- flexible (SW centric platform)
- memory intensive (>70% area, power)

Platforms evolve towards better supporting higher degrees of parallelism

Today's multi-core platforms

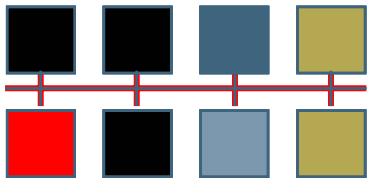
Non-scalable, synchronous bus based interconnect architecture

Limited number of cores (2-8)

Single core design flow supports mapping of single application on single processor

One application runs on single core

Core processors have limited functional (VLIW-8) and data (SIMD-4) parallelism



Next generation multi-core platforms

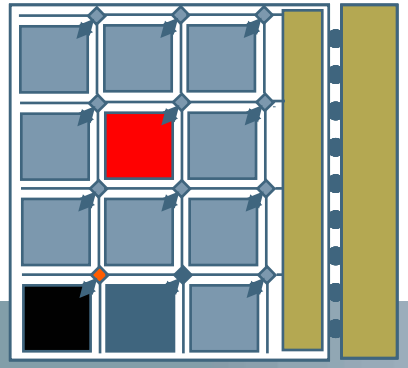
Scalable GALS-NoC based interconnect architecture

Large number of cores (16-100)

Multi-core design flow supports mapping of multiple applications on heterogeneous multi-core platform

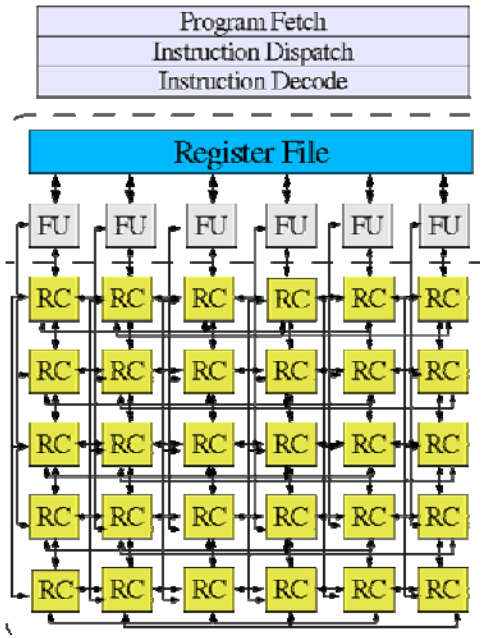
One application runs on N cores, depending on run-time load condition

Core processors have high functional (2D VLIW-64) and data (SIMD-1024) parallelism



Trend towards processing components with increased support for functional parallelism

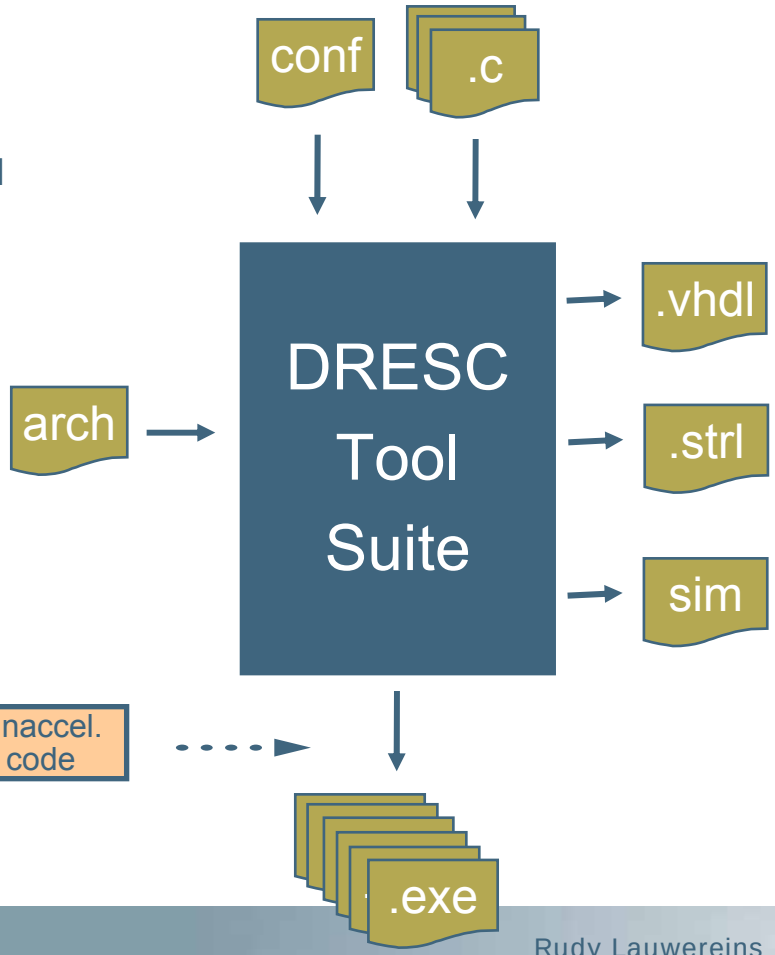
Tightly coupled 1 dimensional and 2 dimensional VLIW processor template



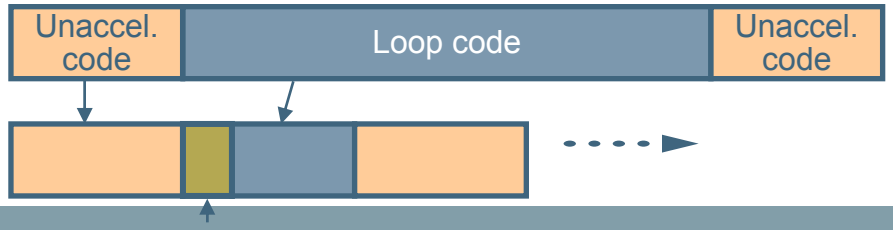
1D VLIW executes control flow code, exploits ILP

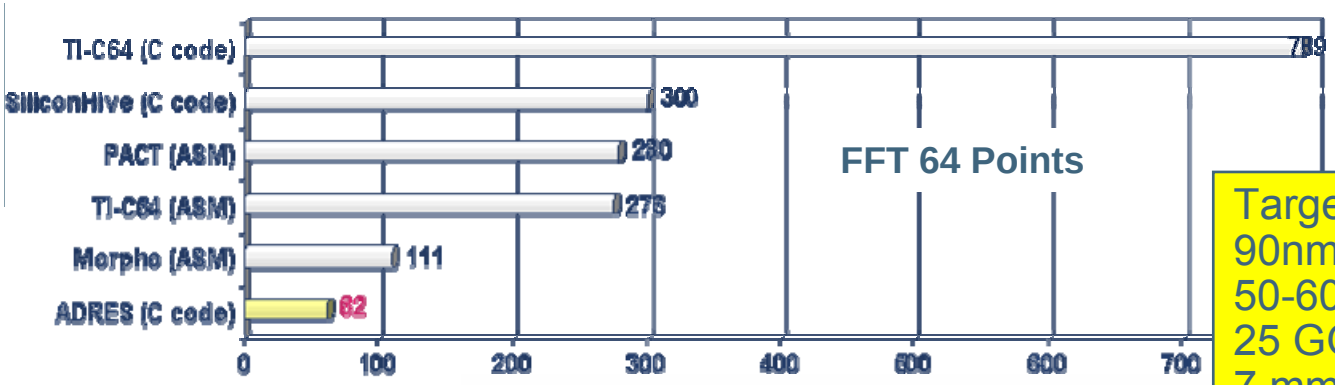
2D VLIW runs data-flow accelerated code (kernels), exploits LLP

Supported by retargetable tool suite: compiler, HW generator, cycle accurate simulator, instruction accurate simulator

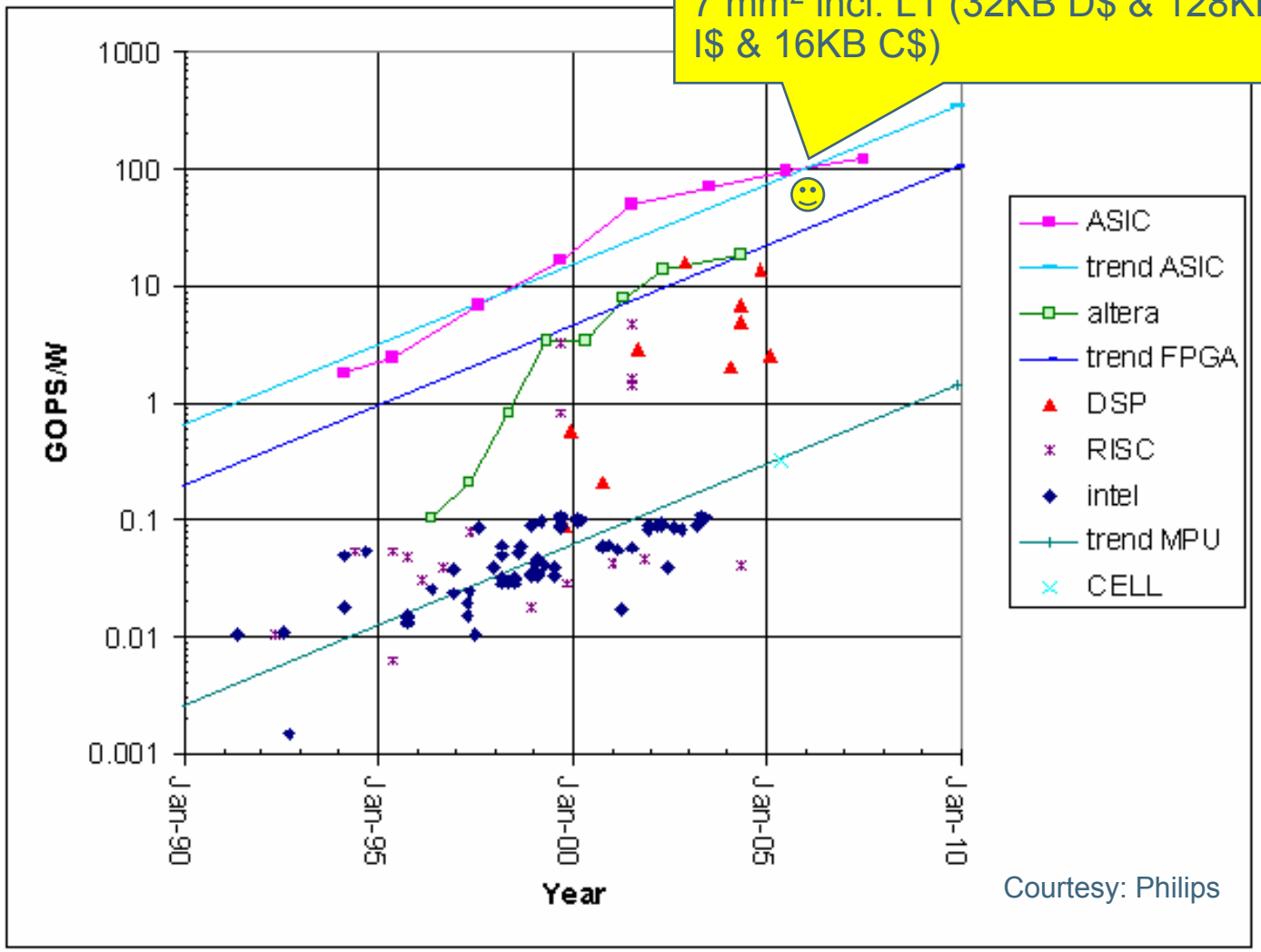


Speed-up and power-gain



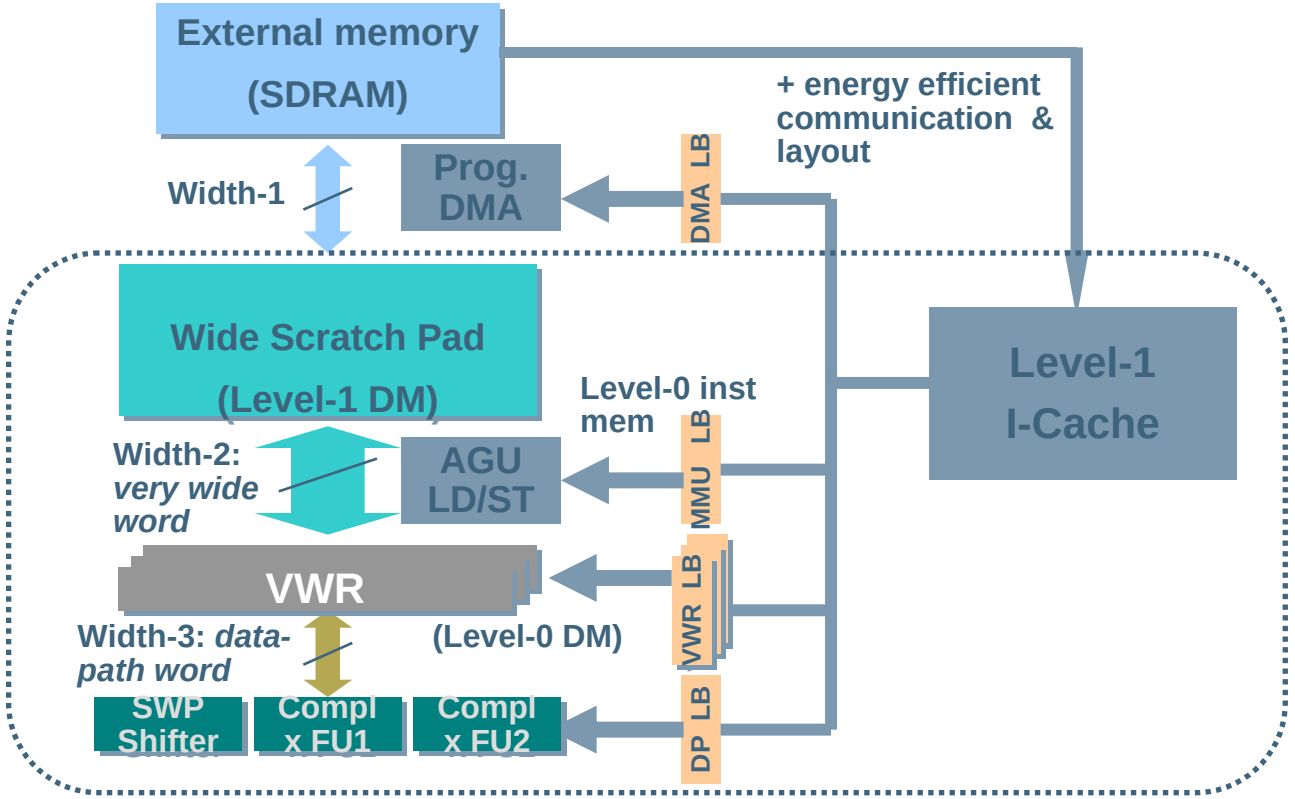


Target:
90nm, 32 bit, 8x8 array
50-60 GOPS/W
25 GOPS peak @ 400 MHz
7 mm^2 incl. L1 (32KB D$ & 128KB I$ & 16KB C$)



Courtesy: Philips

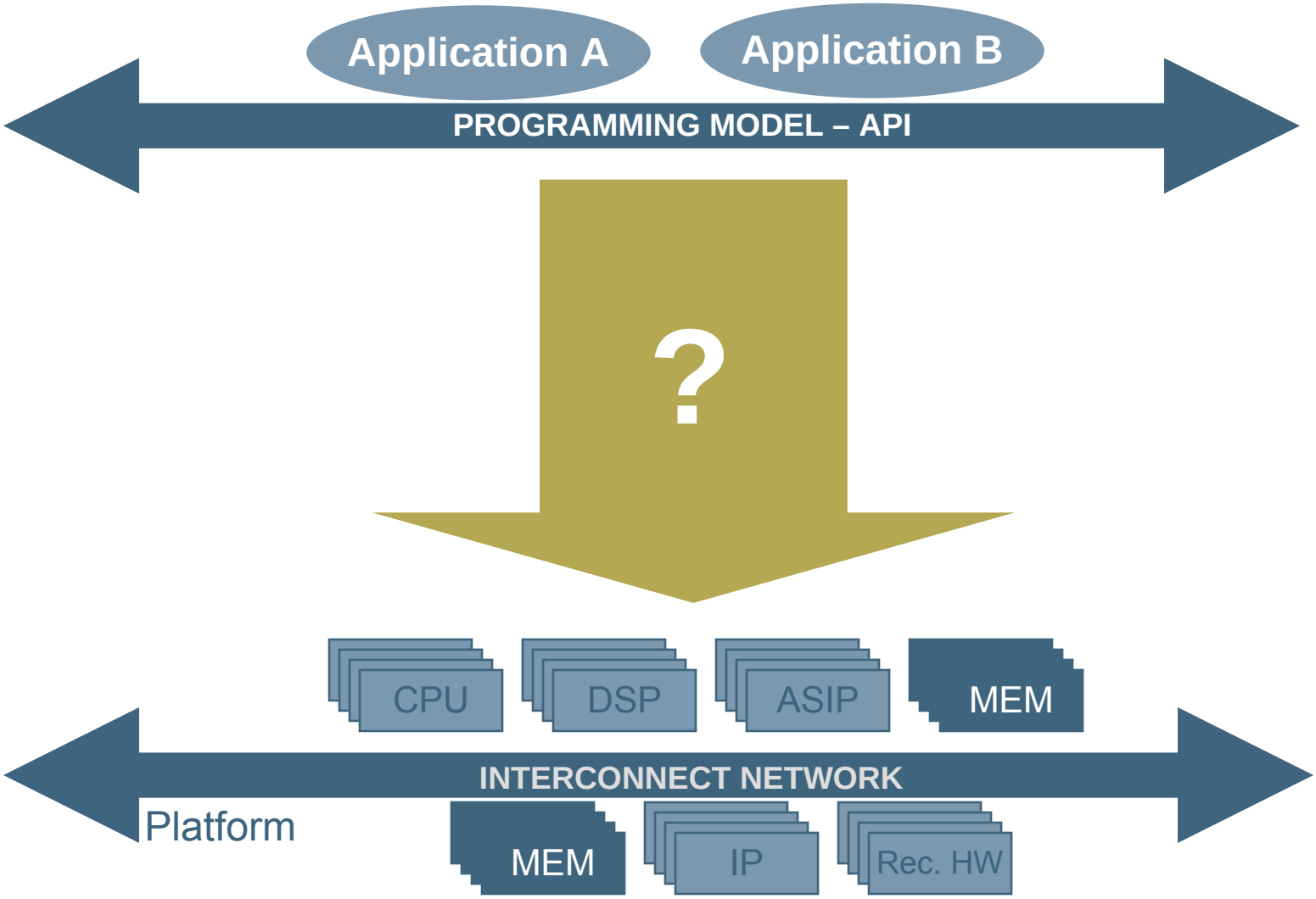
Trend towards processing components with increased support for data parallelism



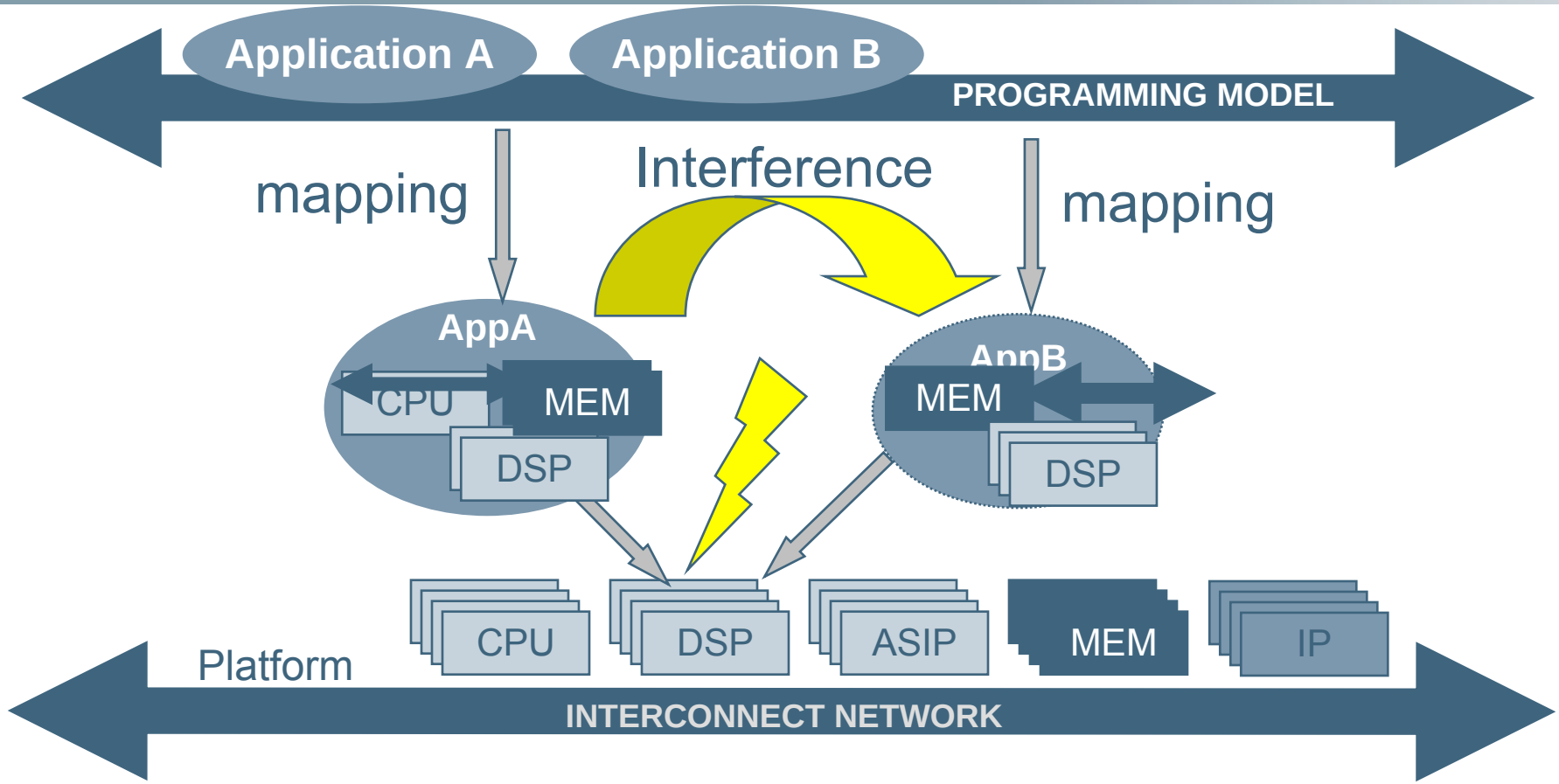
Similar idea: “A 40 GOPS 250 mW massively parallel processor based on matrix architecture”, M. Nakajima et.al., Renesas, ISSCC2006

2048-way SIMD with 2-bit processors

MPSoC Platform Mapping Problem



Application mapping on MPSoC platforms: Predictable, Modular and Scalable

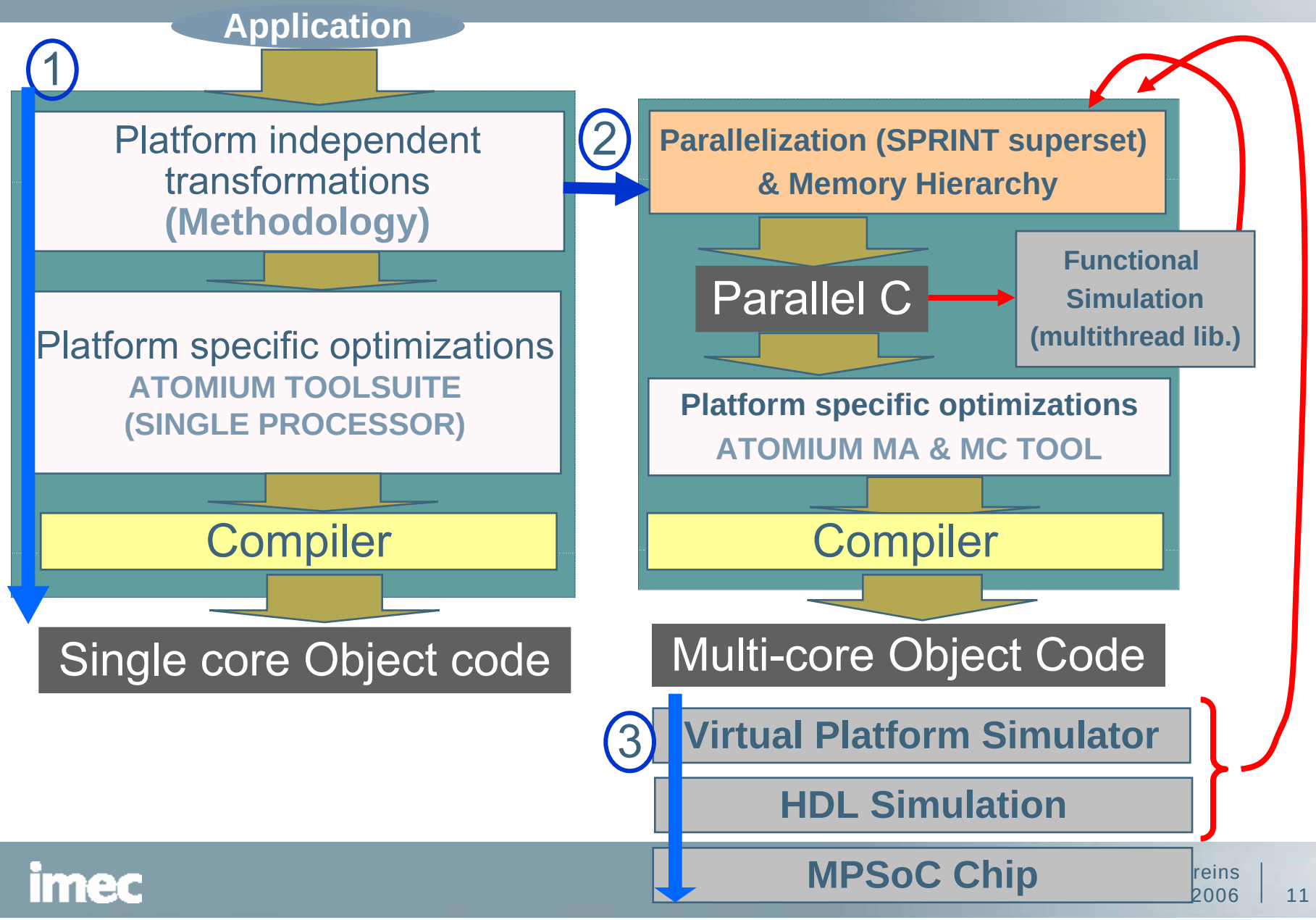


Design-Time Mapping Predictability: how to ensure that multiple independently mapped applications will simultaneously work with the same behavior on a single platform

Design-Time Mapping Scalability: offering different solutions for different platform resources and run-time requirements (e.g. resolution)

Run-Time Decision Making: Exploiting design-time solutions for handling run-time conditions?

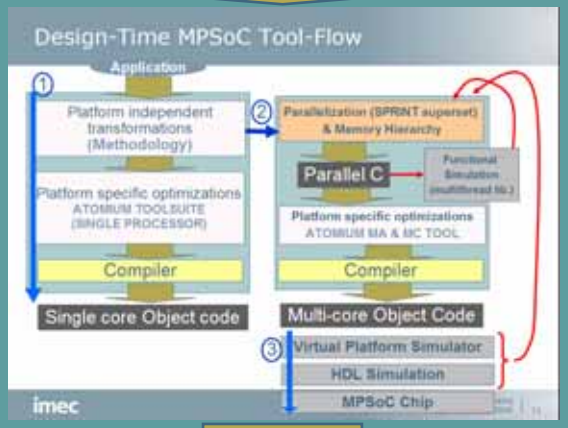
Design-Time MPSoC Tool-Flow



MPSoC: Design-Time, Run-Time Manager & Platform

DESIGN TIME

Set of independent applications



Applications Pareto Surface

Applications Parallel Object Code

RUN TIME (RT)

Pareto Point Selection

Platform Resource Assignment

MPSoC Platform

Virtual Platform Simulator

HDL Simulation

MPSoC Chip

Applications Parallel Object Code

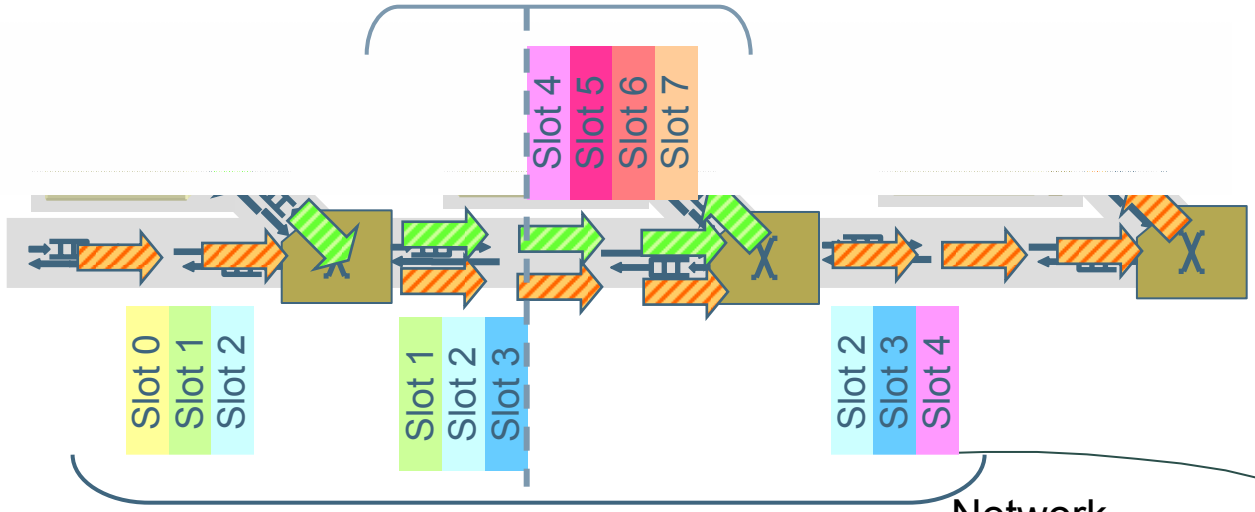
Operating System

RT

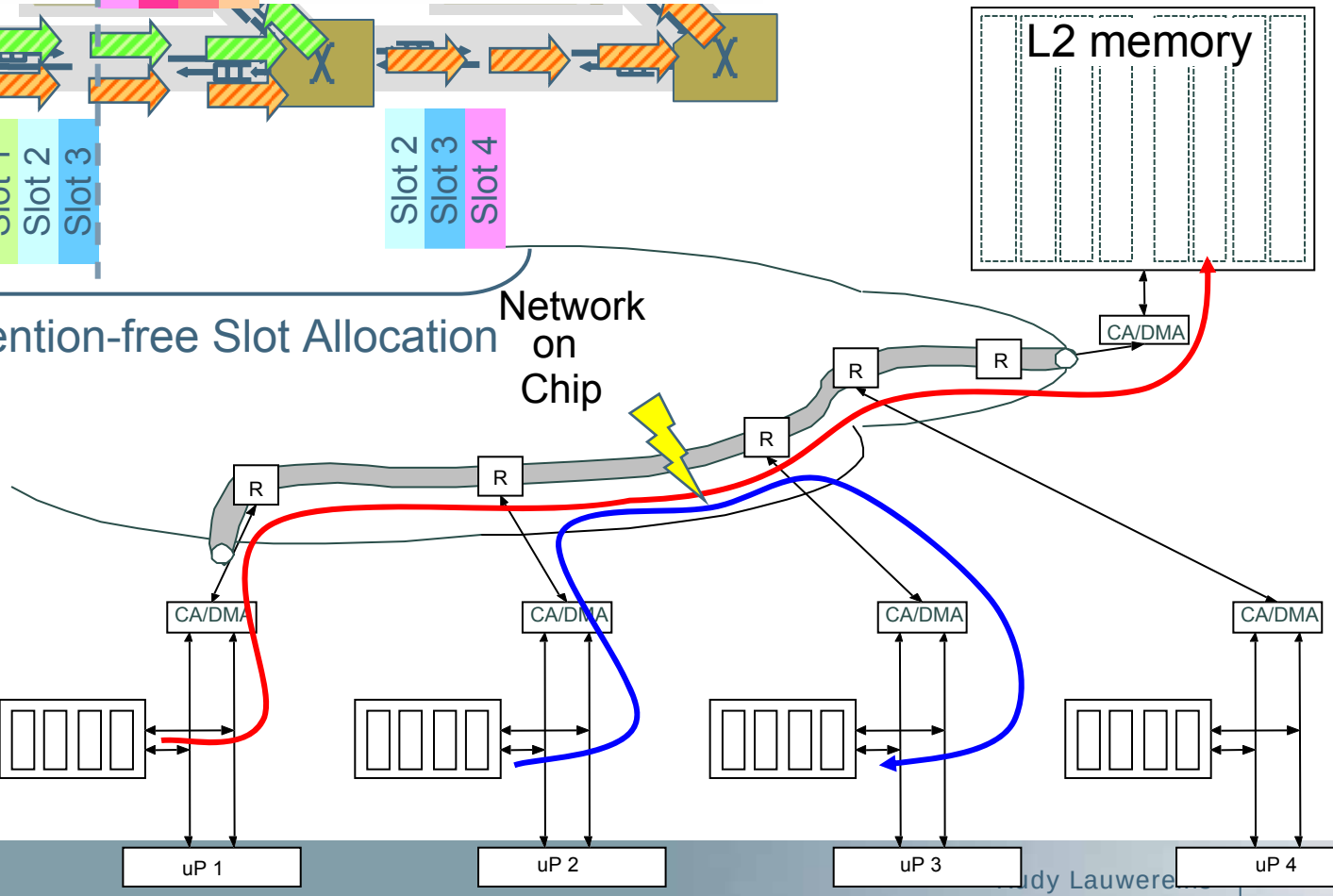
PE | PE | GPP (e.g ARM)

MPSoC Platform

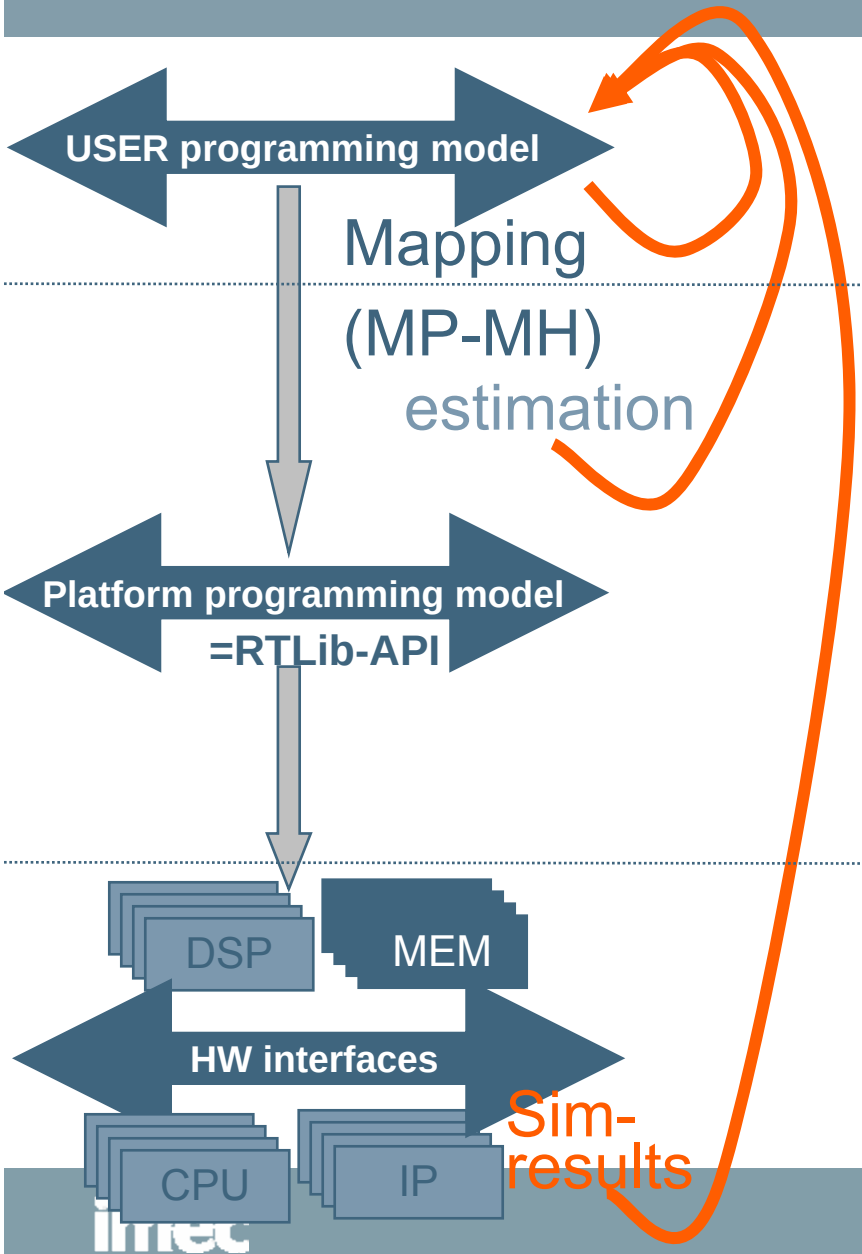
Major task of the design flow: reduce memory access conflicts



Pipelined Contention-free Slot Allocation

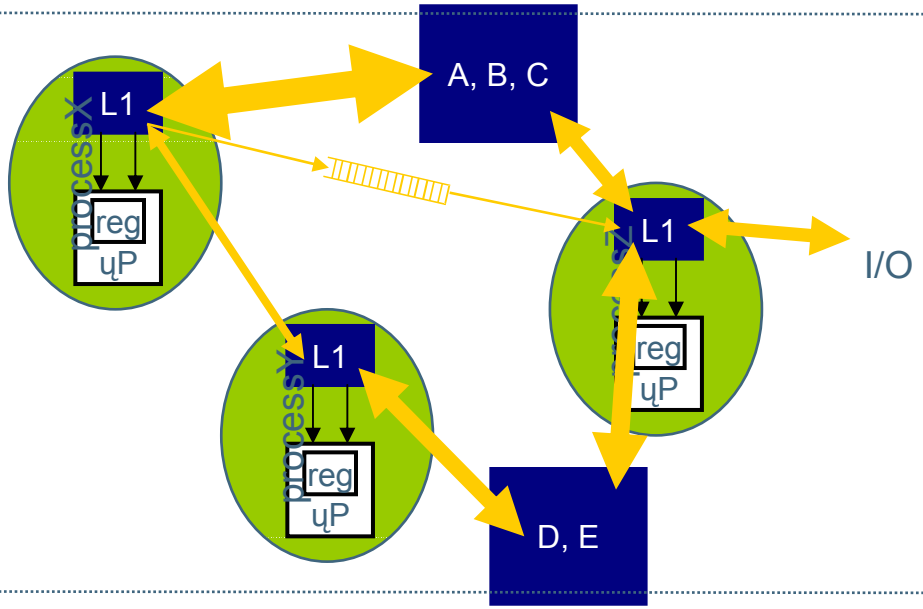


Different programming model levels



AppA +constraints,hints

Sequential code



Designer should stay (is most productive) at highest level

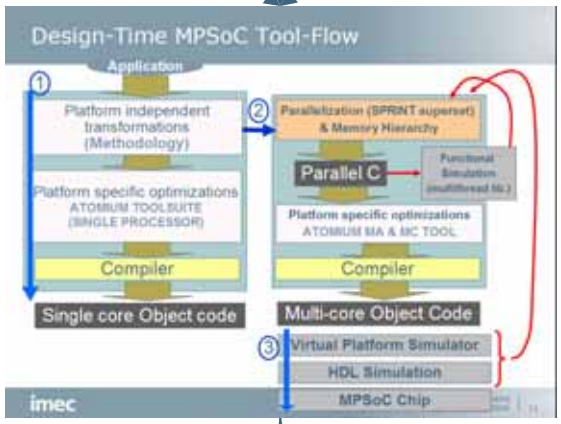
Design-Time/Run-Time MPSoC picture

Potential user & environment influence

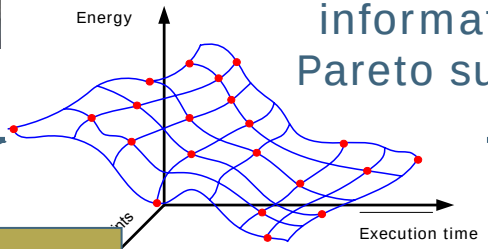
Application

Potential platform parameters

Design time



Exploration information: Pareto surface



Implementation optimized application

Run time

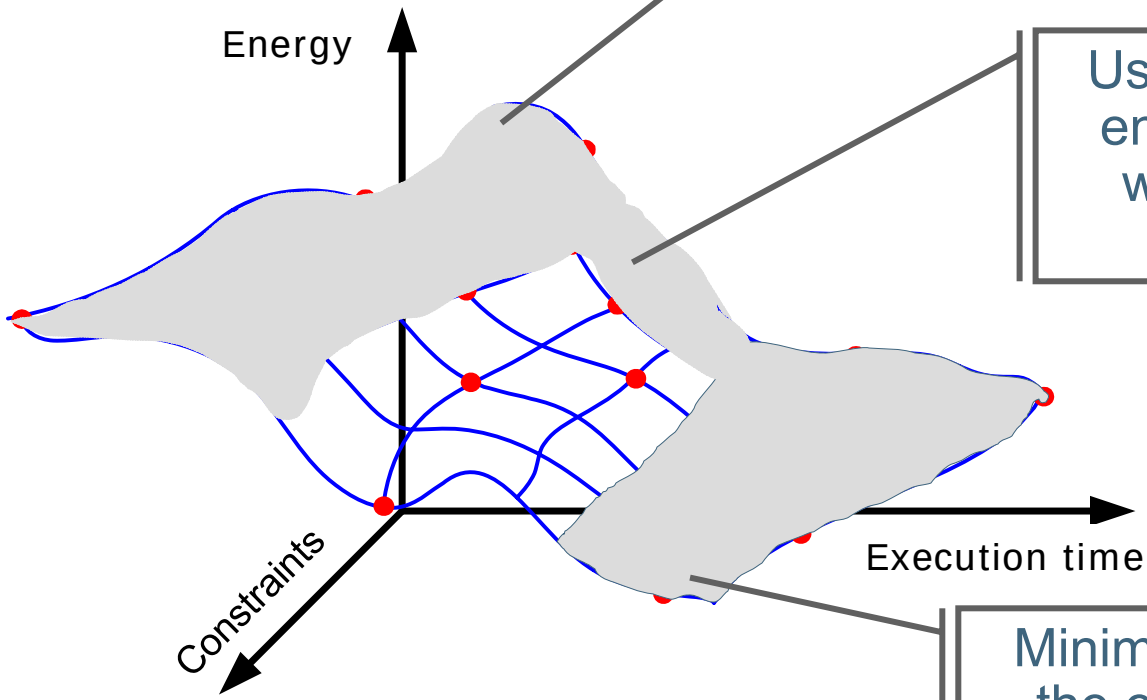
Platform monitoring Information

Platform comp., comm., mem. resource management

Pareto management

User & environment constraints

Application Exploration Information is stored in a Pareto Surface

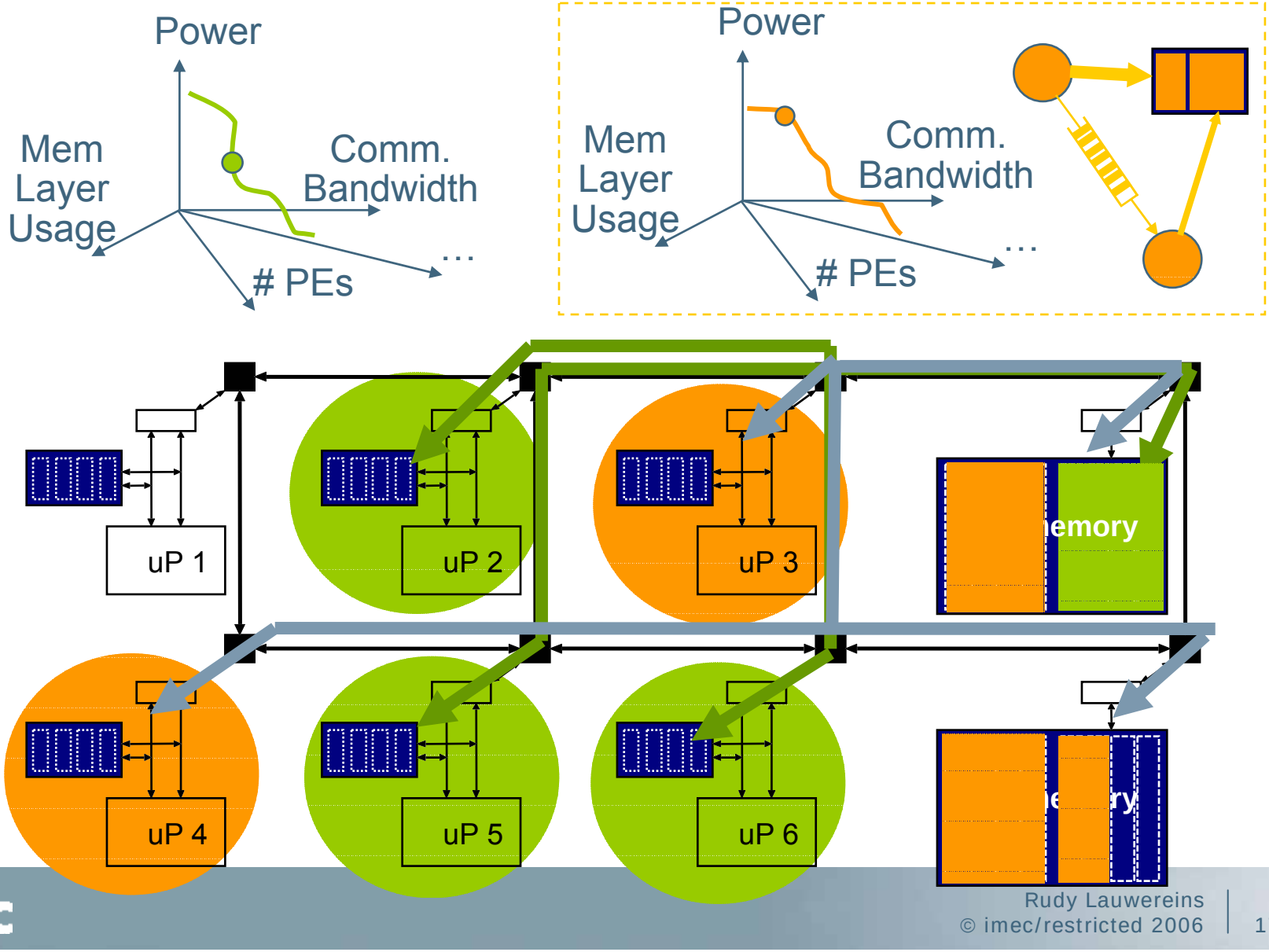


Maximum speed at the cost of more energy

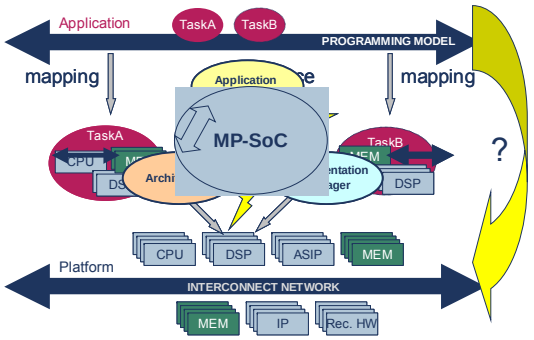
User (e.g. quality) or environmental (e.g. wireless channel) constraints

Minimum energy at the cost of longer execution time

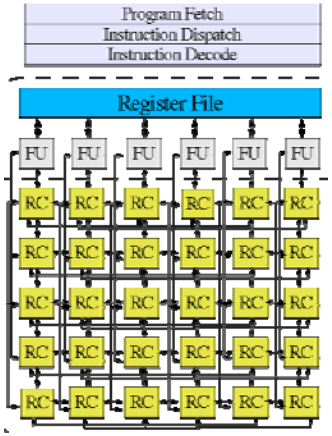
Choosing a pareto point and assigning resources



Case studies showing the use of the mapping tools and the 2D VLIW processor for software defined radio and multi-format multimedia codecs



Digital and mixed signal Design technology



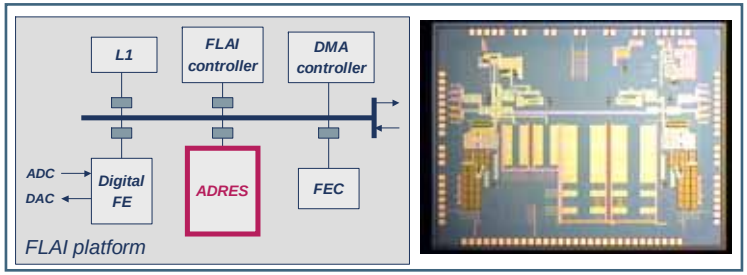
Digital and RF implementation platforms

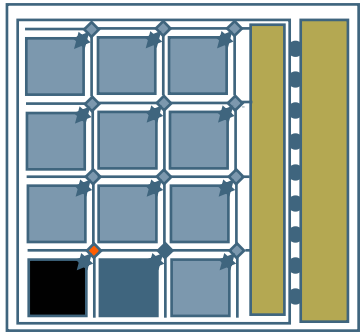
Instantiate in SDR platform

Instantiate in multimedia platform

Multi-mode wireless communication platform

Heterogeneous platform, 4x4 SIMD-4 ADRES

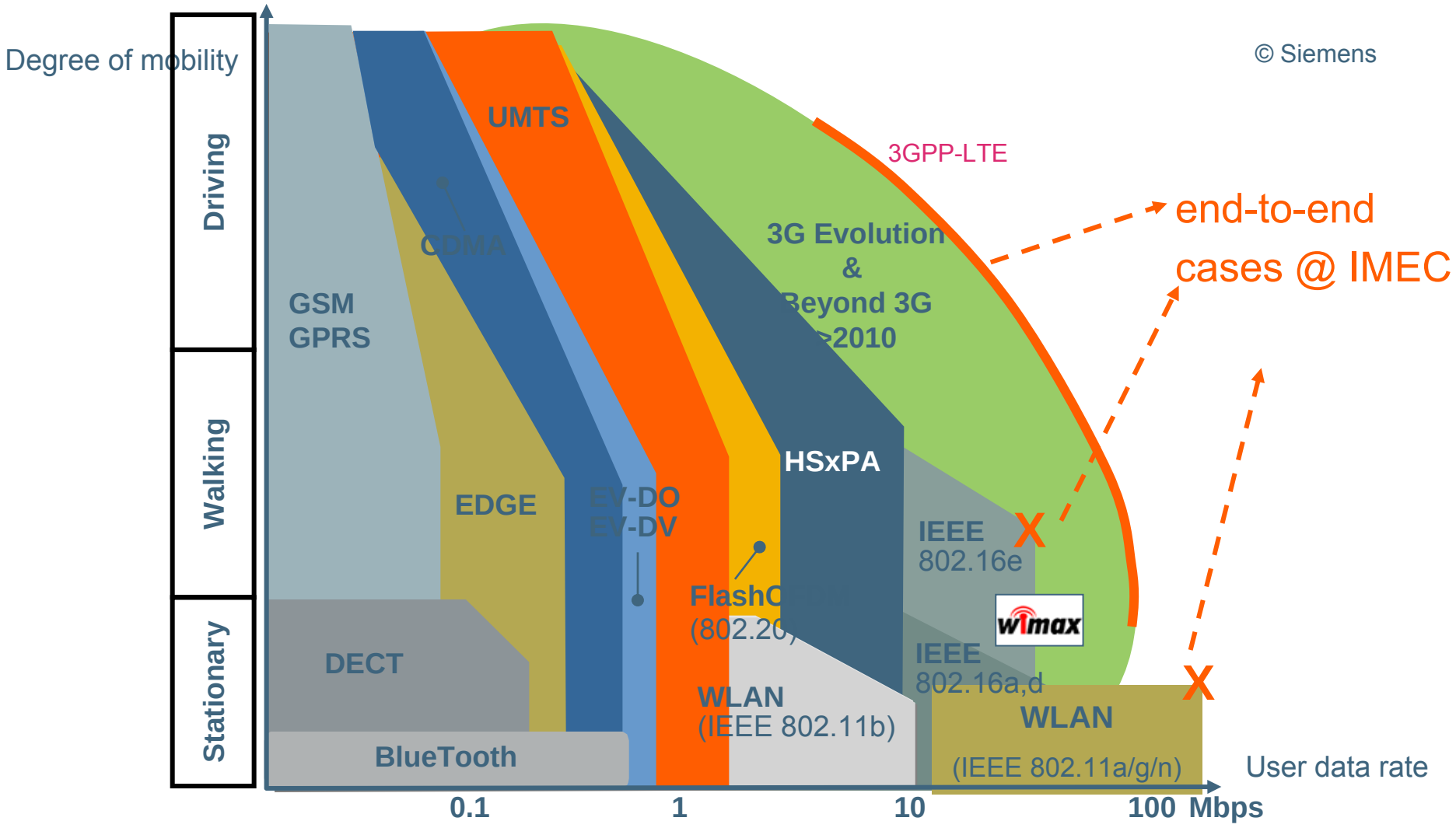




Multi-format multimedia platform

6 ADRES cores, NoC

Wireless standards: a large variety, we show coverage through limit cases



Algorithm (code)–architecture co-design to bridge huge gaps

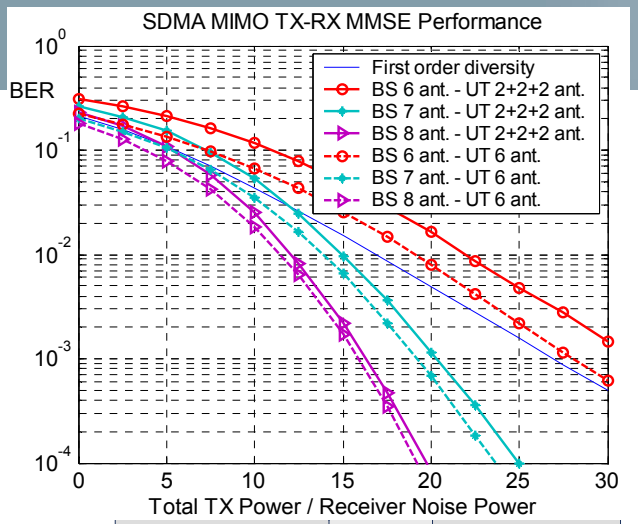
Matlab to automated C
>1000 GOPS

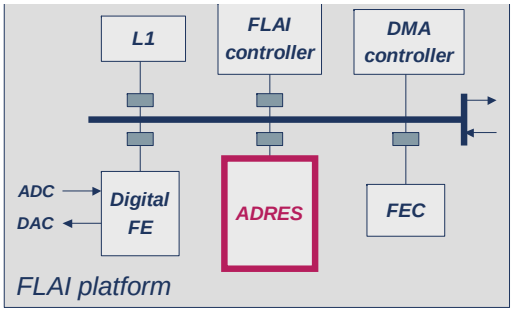
Implementation ready C
>60 GOPS

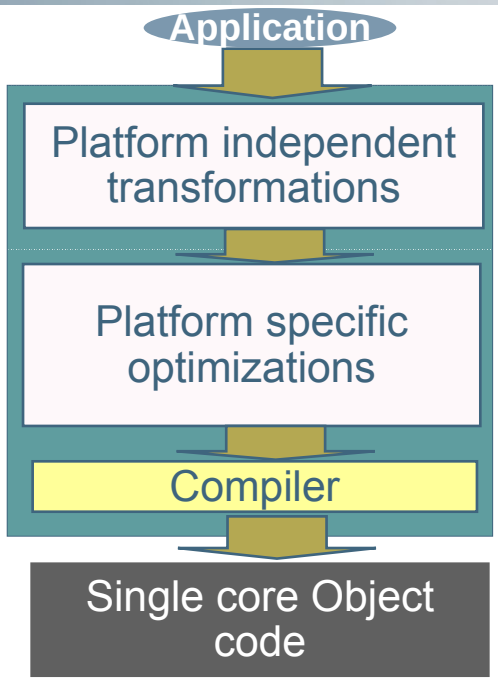
Platform optimized C on standby optimized platform

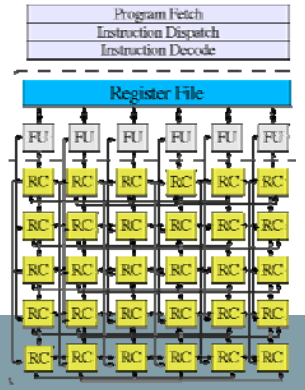
SDR optimized 2D VLIW
<30 GOPS

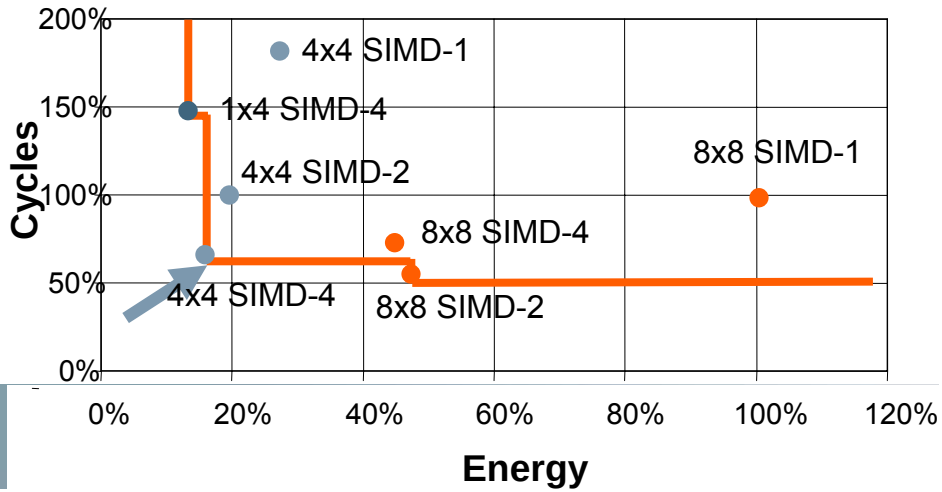
Generic 2D VLIW
<6 GOPS









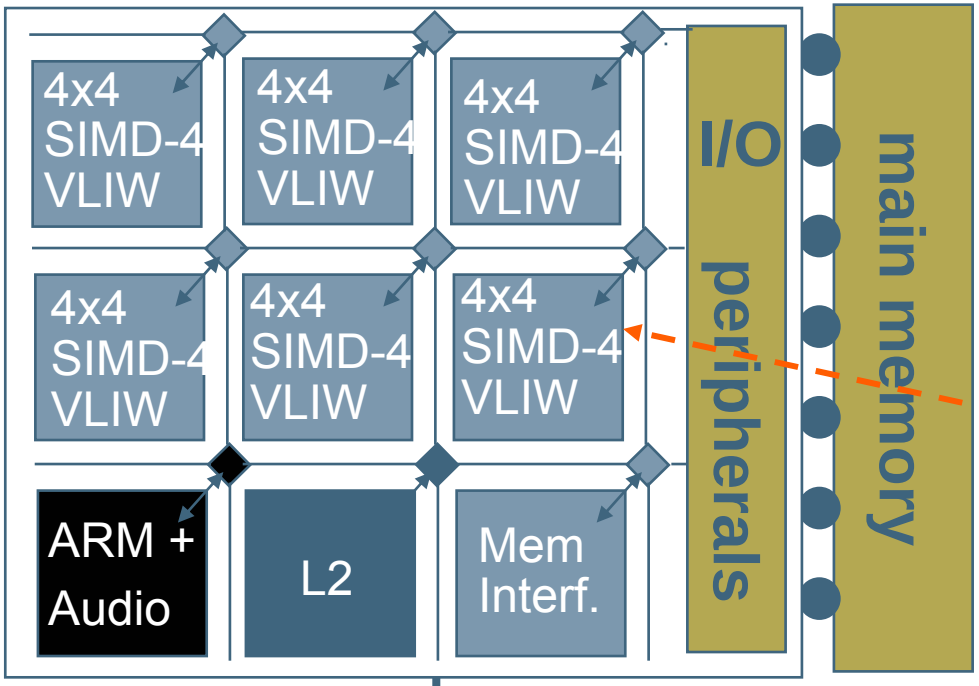


imec

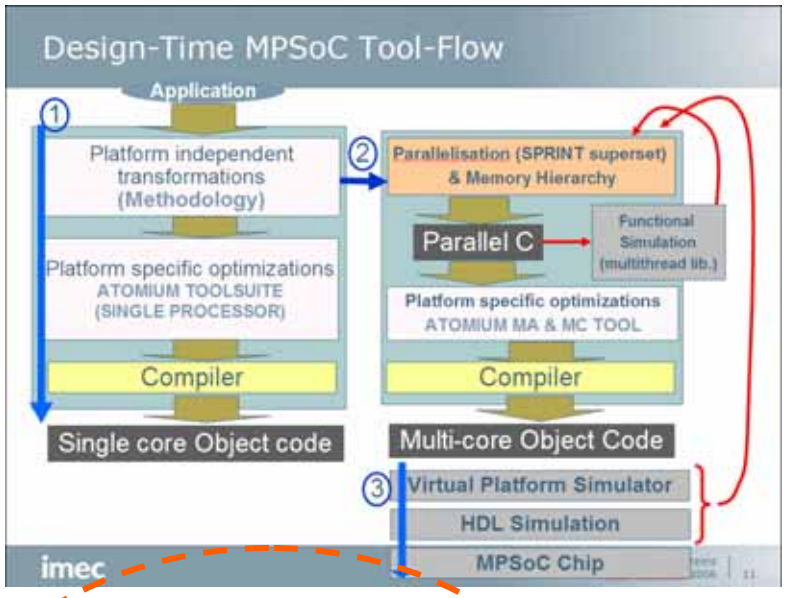
Single flexible platform is optimized for many multimedia coding standards

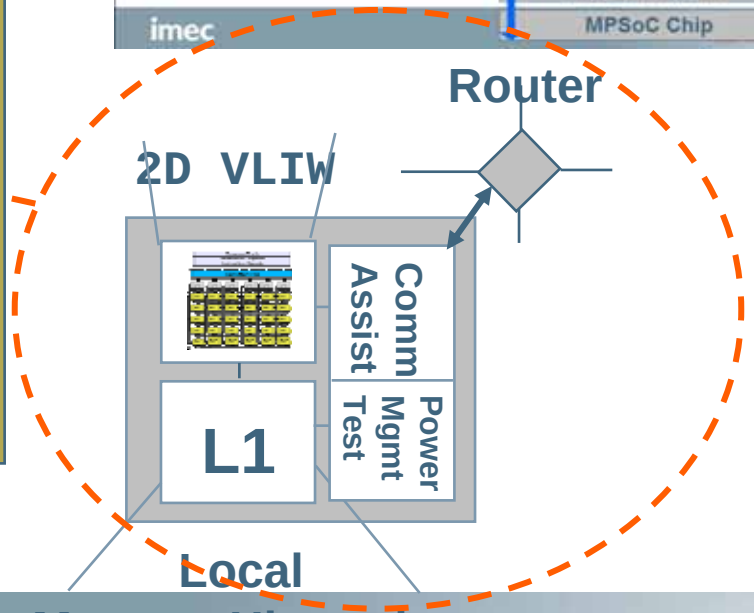
Architecture is fit for following standards:

- MPEG-2, MPEG-4, VC-1
- HDTV/720p level quality AVC (H.264)
- Scalable Video Coding up to VGA quality

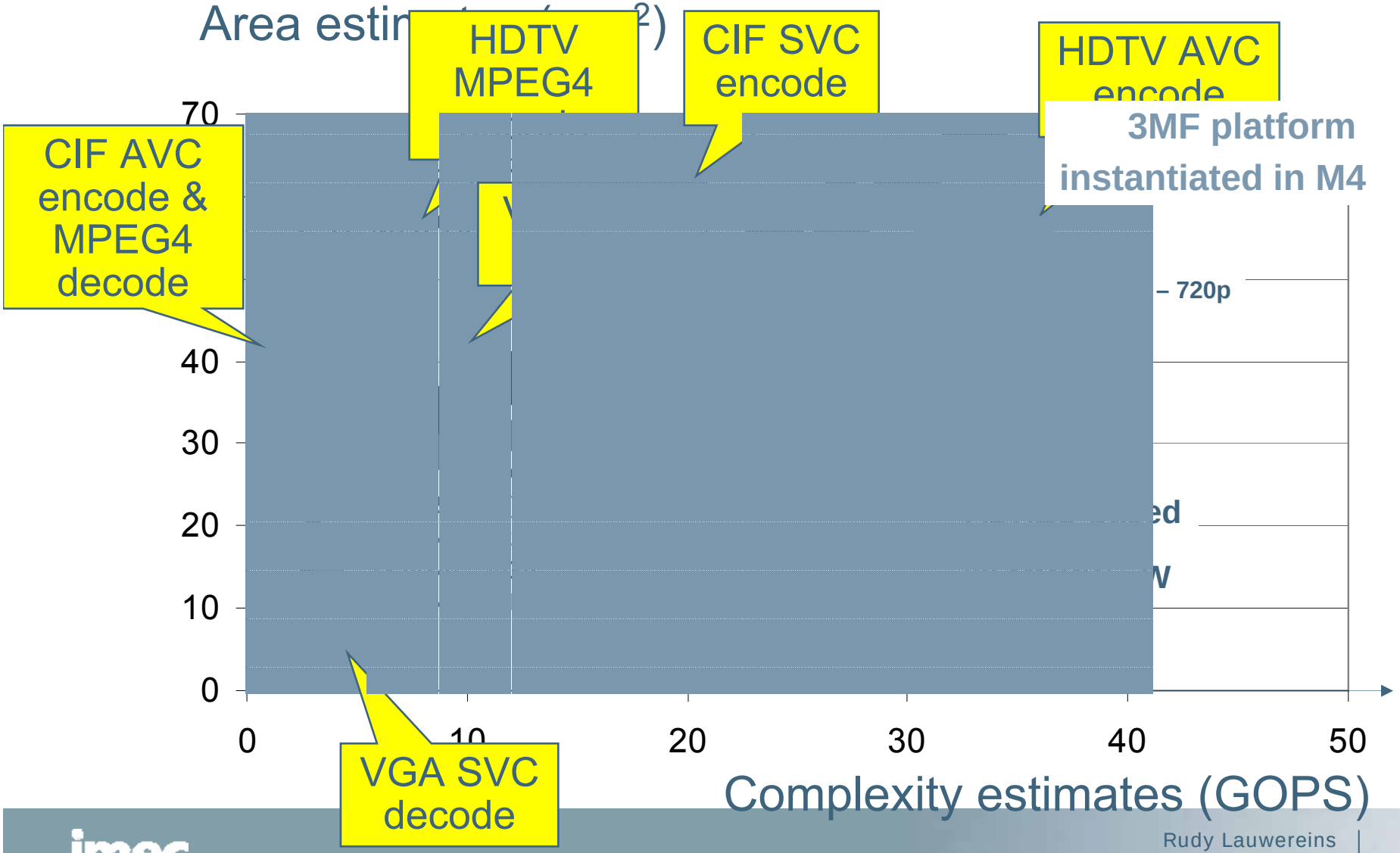


Note: this is not a layout view, but an architectural view

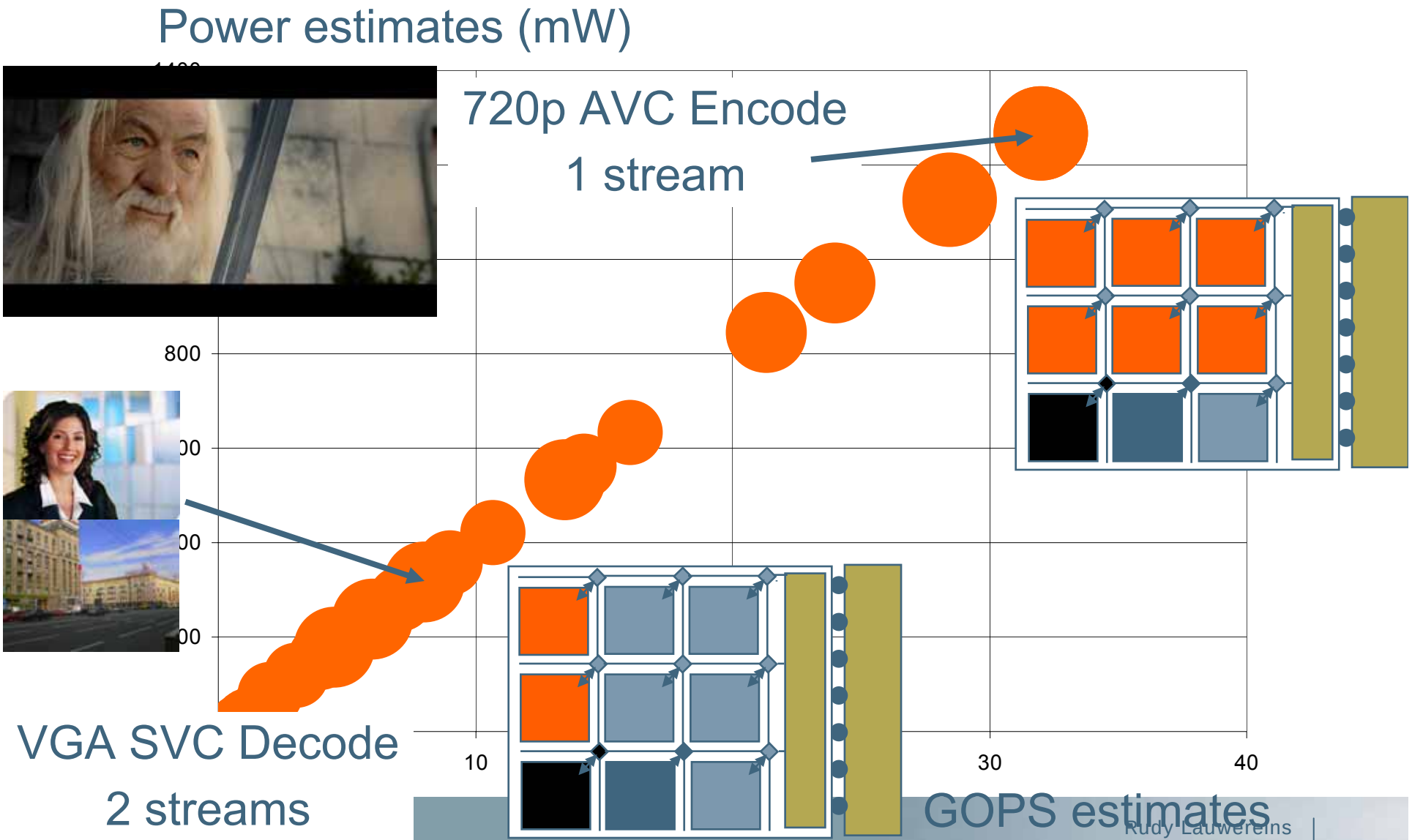




Flexible MM Platform: Design-Time Flexibility
Area determined by highest performance standard needed



Resulting Run-Time Flexibility
Power determined by actual usage scenario



Conclusion

- Platforms evolve towards supporting higher degrees of parallelism
- Beware of the hell of physics
- Be even more aware of the software devil: a multi-core design flow is in urgent need
- Research in all above is becoming mature and proven in multi-mode radios and multi-format multimedia codecs


aspire invent achieve
imec