Software Variability from the Nomadic Devices Perspective

Roel Wuyts
ARES Group
IMEC
KULeuven

SVPP'08, 8/8/8

IMEC

- Micro-Electronics research organization located Leuven, Belgium
  - Mission “To perform R&D, ahead of industrial needs by 3 to 10 years, in microelectronics, nanotechnology, design methods and technologies for ICT systems”
- Numbers
  - Budget: ± 200 M€
  - Staff: ± 1700
  - Cleanroom: ± 10,000 m2

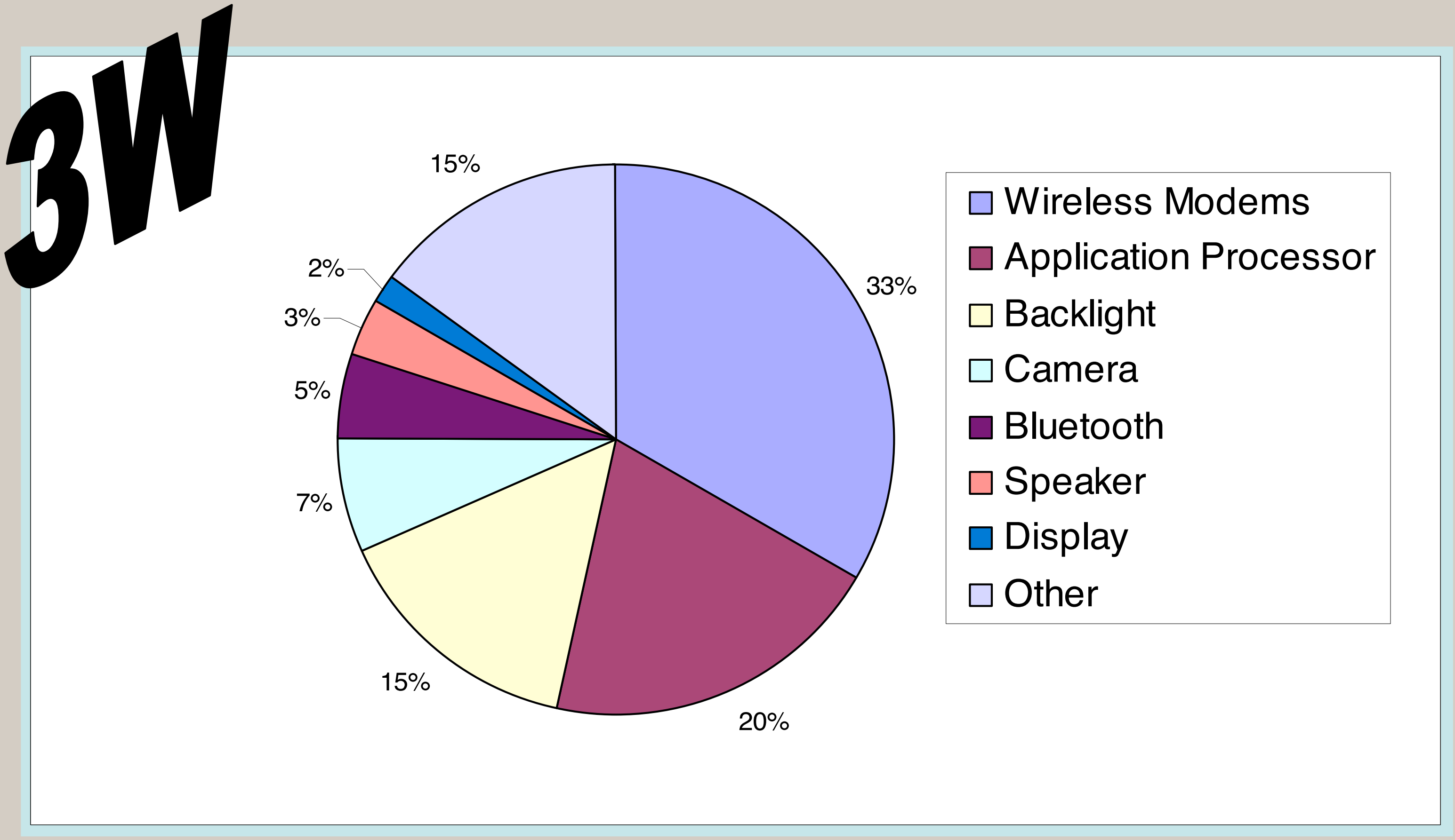
Nomadic Devices

Nomadic Devices

Nomadic Device Characteristics

- Power and energy constraints (battery)
- Design time constraints (time to market)
- Cost
- Real-time constraints
- Flexibility and performance

Handheld Battery-powered Devices



- ➡ Max 1.5W for processing
- ➡ Max 1W for SMP cores

[Yrjö Neuvo. Cellular phones as embedded systems.
In Solid-State Circuits Conference, 2004.
Digest of Technical Papers. ISSCC,
volume 1, pages 32–37, February 2004.]

[Kimmo Kuusilinna, Nokia, Date'08]

Current Market

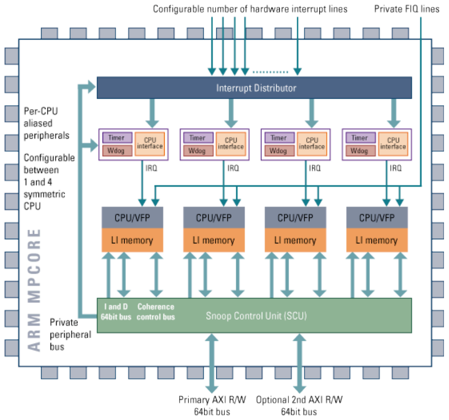
0.2mi
Trafalgar Studios At The Whitehall Theatre
Adobe Reader
1/1
75%
Options
Back
Amazon.com: Online Shop...
Close
Cart | Your Lists | Help |
Today's Deals | Sell Your Stuff
Find Gifts
Web Search
GO
Terrific Textbook Discounts
Save up to 20% on a wide
10:19 PM
Options
March
11:00 AM-Monthly me...
7:00 PM-Yoga, Main hall
Options
Open
Exit
Fido
8:53 pm 2/9/06
NOKIA
Now Playing
Shine
Matthew Jordan
SAMSUNG mobile
©LetsGoDigital

Concrete Numbers for the Belgian market

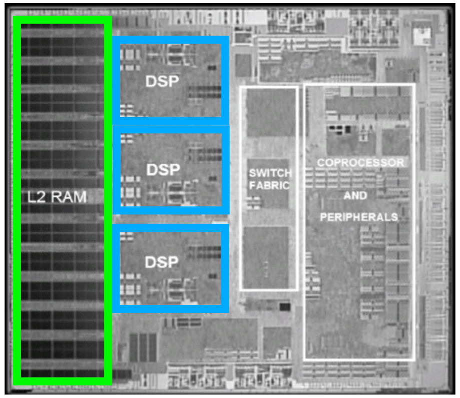
- Motorola: 21 models, 66 models in support
- Nokia: 94 models, 144 models in support
- Samsung: 34 models, 174 models in support
- LG: 18 models

Multicore is here to stay

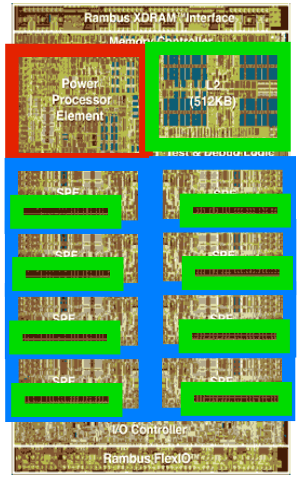
ARM



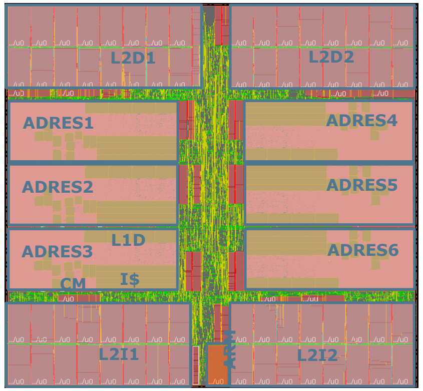
Texas Instruments

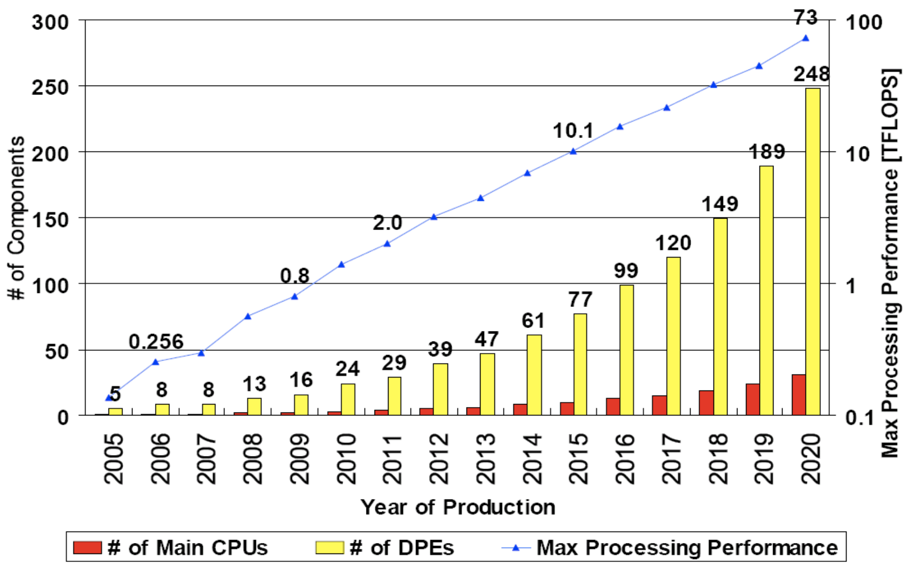


IBM Cell



IMEC



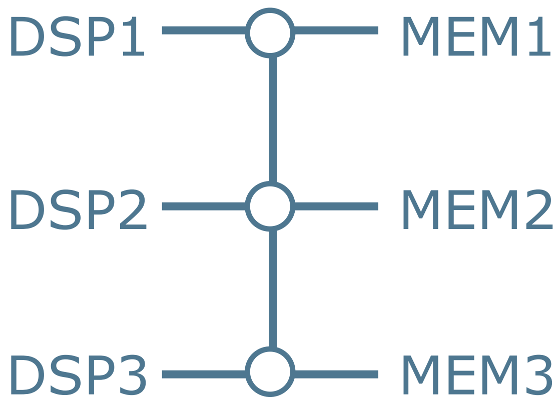


The key question:

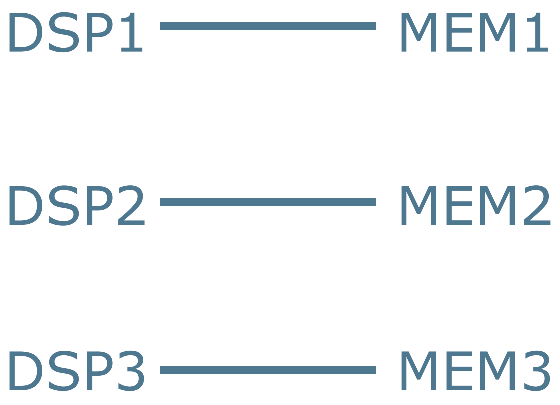
"How to efficiently program them?"

Programming Nomadic Devices

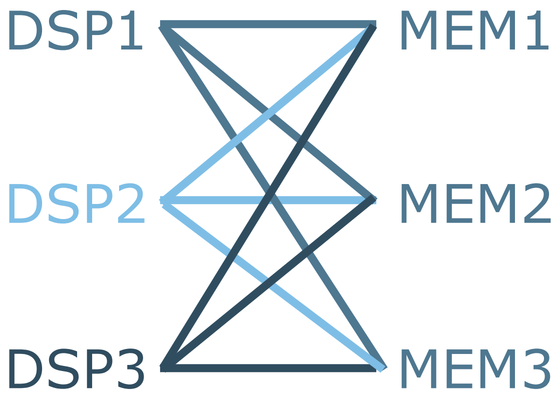
Design variability: example



DSP1 MEM1
DSP2 MEM2
DSP3 MEM3



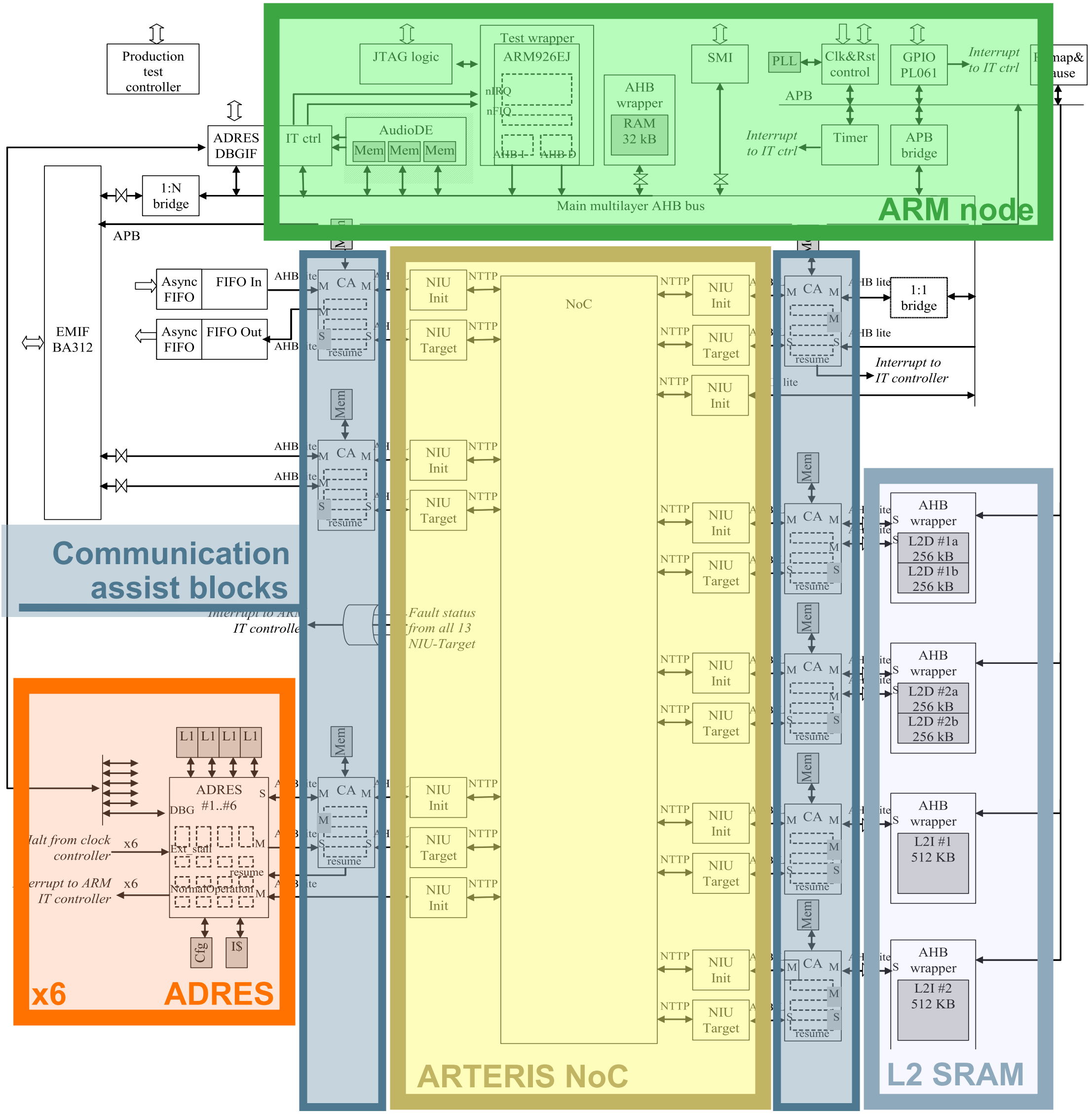
DSP1 MEM1
DSP2 MEM2
DSP3 MEM3



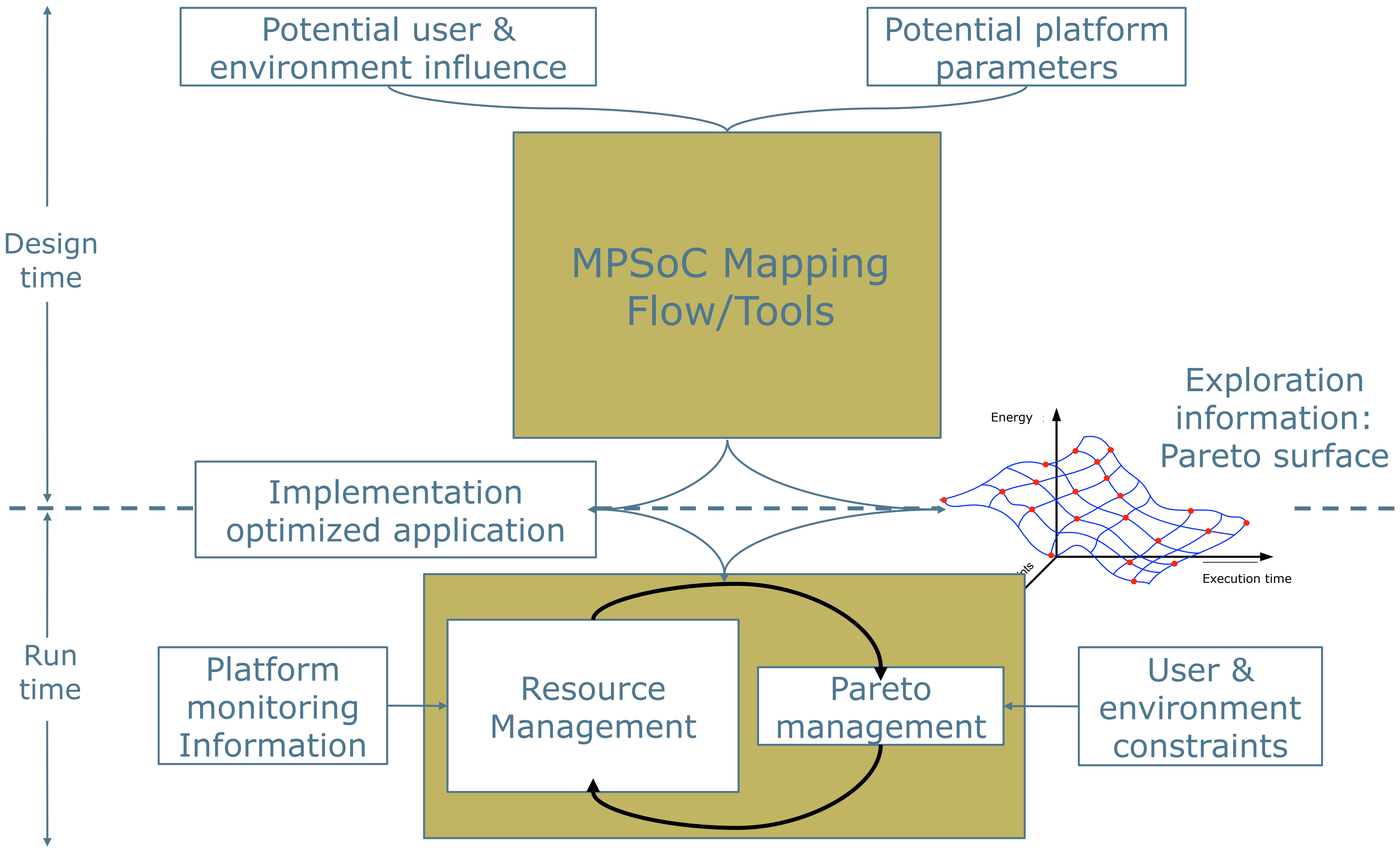
DSP1 MEM1
DSP2 MEM2
DSP3 MEM3

Design Variability: MPSoC Example

- 6 ADRES processors
  - 4x4 array, 3-issue VLIW
  - 32-bit datapath
  - 16 video CODEC specific instructions
  - 8 FUs with multipliers
  - Performance: 300MHz
- 13 Communication assist
  - Performance: 75/150MHz
- ARTERIS NoC
  - Separate instr. and data NoC
  - Bandwidth: 5Gbps@150MHz
- ARM926
  - System control
  - Performance: 75MHz
- L2 memory
  - L2I: 2 banks of 512kB
  - L2D: 4 banks of 256kB
- Voltage islands
  - ADRES processors
  - L2I and L2D banks
- Multiple clock domains



Dealing with variability: IMEC's Approach



Design time

Run time

Potential user & environment influence

Potential platform parameters

MPSoC Mapping Flow/Tools

Implementation optimized application

Exploration information: Pareto surface

Platform monitoring Information

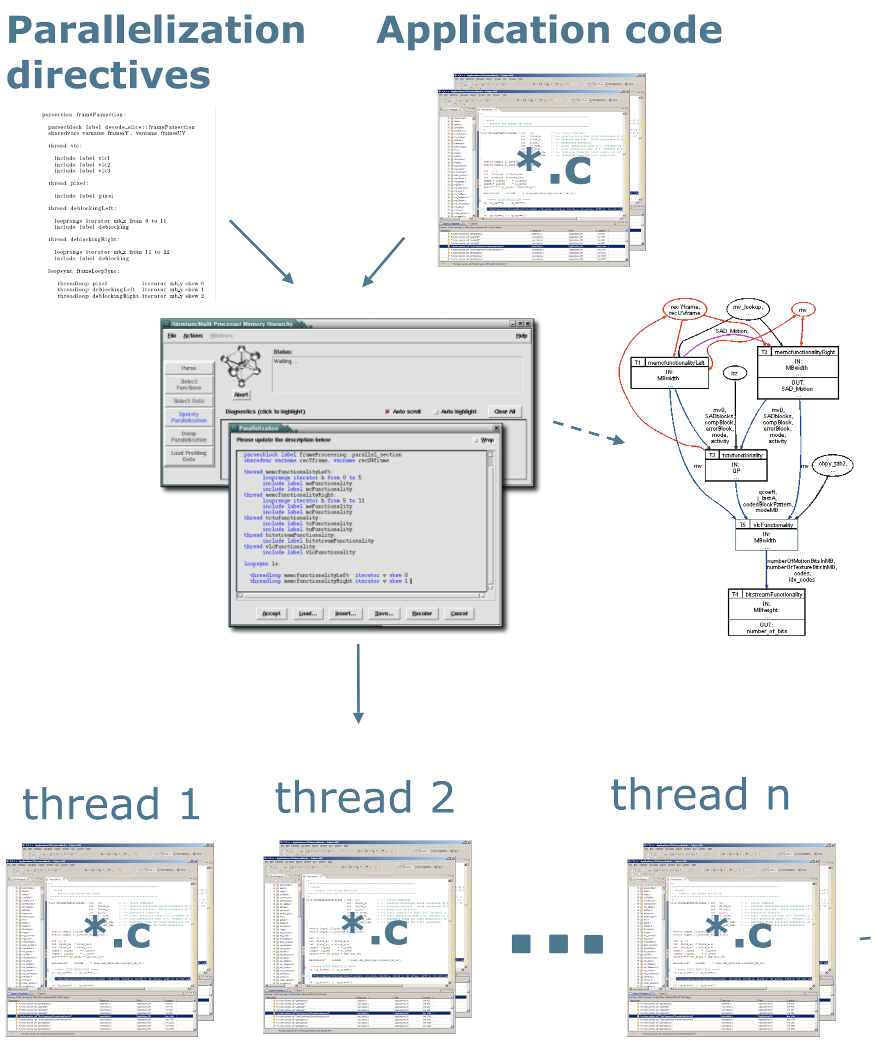
Resource Management

Pareto management

User & environment constraints

Solution for MPSoC: IMEC MPA Tool

- ✓ Parallelizes sequential Clean-C source code
  - ✓ *Correct-by-construction multi-threaded code*
  - ✓ Higher level than OpenMP
  - ✓ Directives in separate file
- ✓ Supported types of parallelism
  - ✓ Functional split
  - ✓ (Coarse) Data-level split
  - ✓ Combinations
- ✓ Dumps parallel code
- ✓ Sets up communication
  - ✓ Communication by means of FIFO's
  - ✓ DMA transfers
  - ✓ FIFO sizes determined by tool (initial version)



Parallelization directives

Application code

```
parsection frameParsection:

  parsecblock label decode_slice::frameParsection
  sharedvars varname framesY, varname framesUV

  thread vlc:

    include label vlc1
    include label vlc2
    include label vlc3

  thread pixel:

    include label pixel

  thread deblockingLeft:

    looprange iterator mb_x from 0 to 11
    include label deblocking

  thread deblockingRight:

    looprange iterator mb_x from 11 to 22
    include label deblocking

  loopsync frameLoopSync:

     threadloop pixel           iterator mb_y skew 0
     threadloop deblockingLeft  iterator mb_y skew 1
     threadloop deblockingRight iterator mb_y skew 2
```

Atomium/Multi Processor Memory Hierarchy

Status:

Waiting ...

Diagnostics (click to highlight)

Parallelization

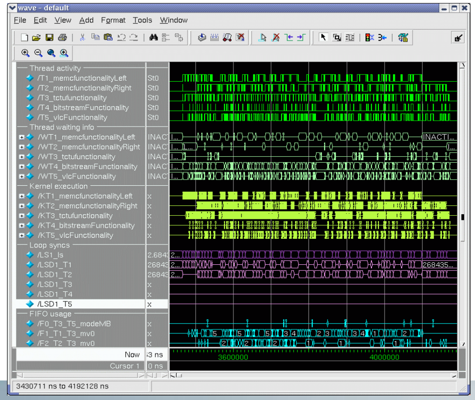
Please update the description below

```
parsecblock label frameProcessing::parallel_section
sharedvar varname recYframe, varname recUVframe

thread memcfunctionalityLeft:
       looprange iterator h from 0 to 5
       include label meFunctionality
       include label mcFunctionality
thread memcfunctionalityRight:
       looprange iterator h from 5 to 11
       include label meFunctionality
       include label mcFunctionality
thread tctufunctionality:
       include label tcFunctionality
       include label tuFunctionality
thread bitstreamFunctionality:
       include label bitstreamFunctionality
thread vlcFunctionality:
       include label vlcFunctionality

loopsync ls:

  threadloop memcfunctionalityLeft  iterator v skew 0
  threadloop memcfunctionalityRight iterator v skew 1
```

thread 1

thread 2

thread n

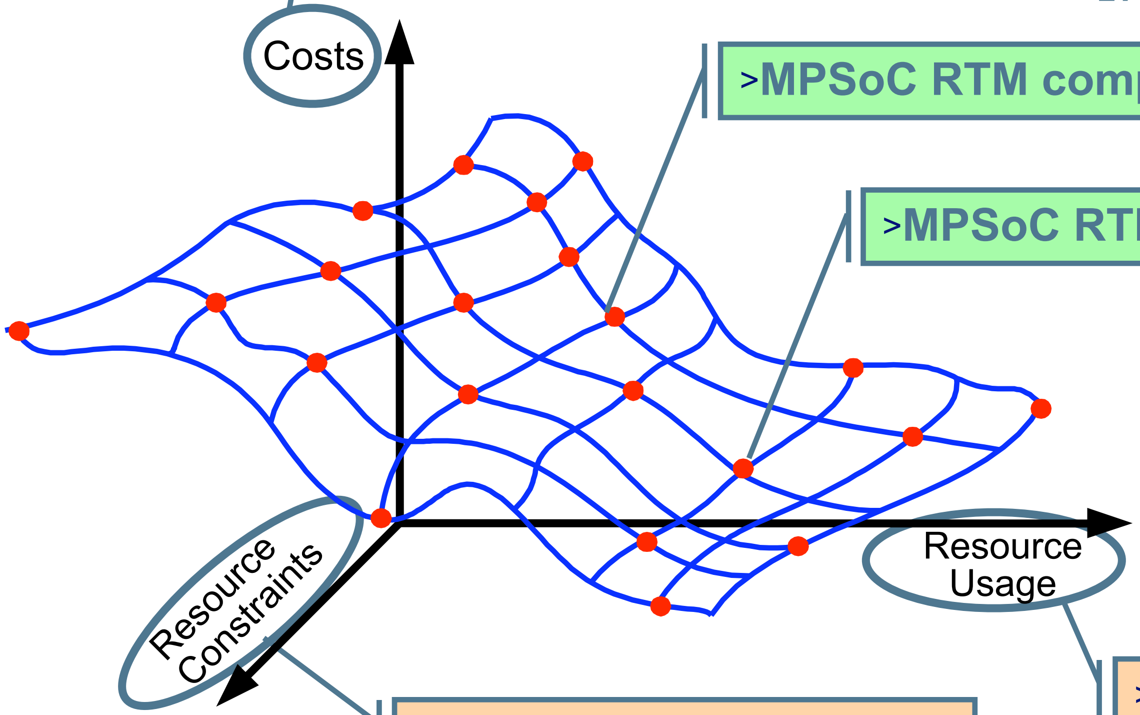


Design Time Exploration

- Energy consumption
- …

Per application

Set of operating points
In a multi-dimension space



>MPSoC RTM component configuration 1

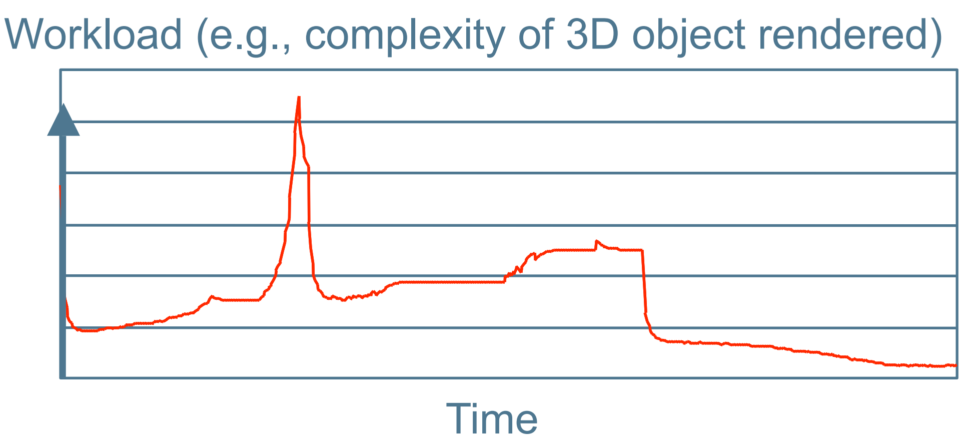
>MPSoC RTM component configuration 2

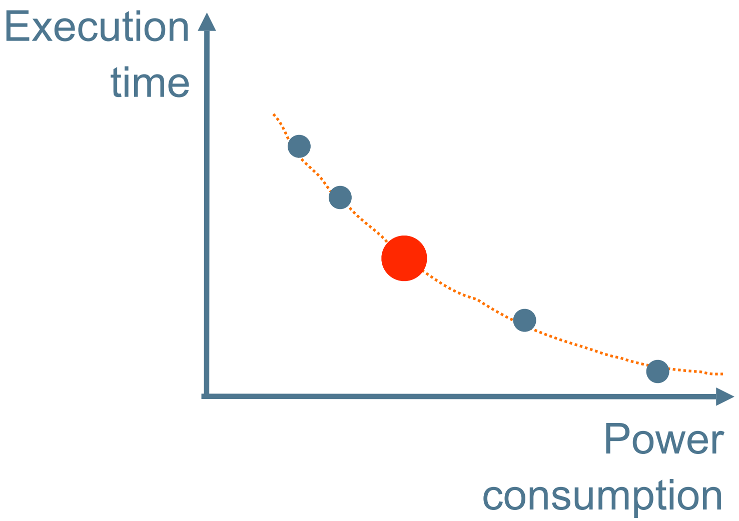
- Speed (Execution time)
- …

- Number of used processors
- Memory usage
- Communication bandwidth
- …

Run-time management

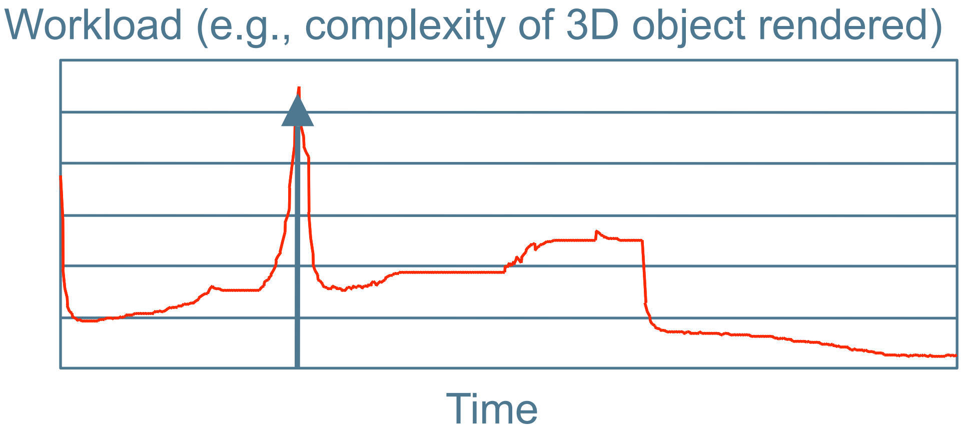
- For each thread frame, run-time scheduler changes management dynamically according to the run-time situations

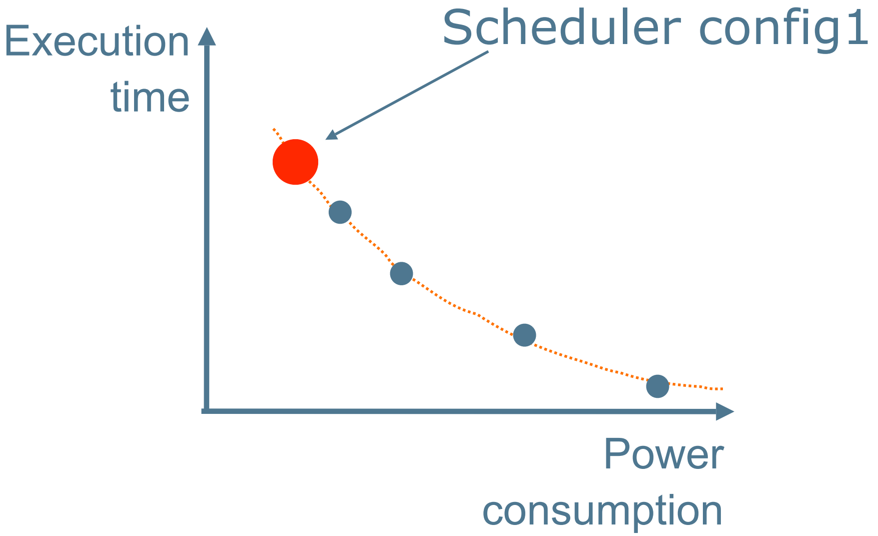




Run-time management

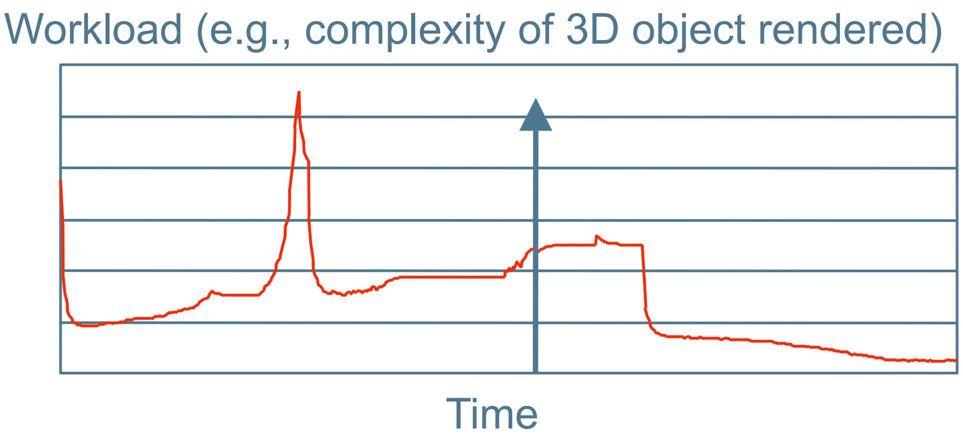
- For each thread frame, run-time scheduler changes management dynamically according to the run-time situations

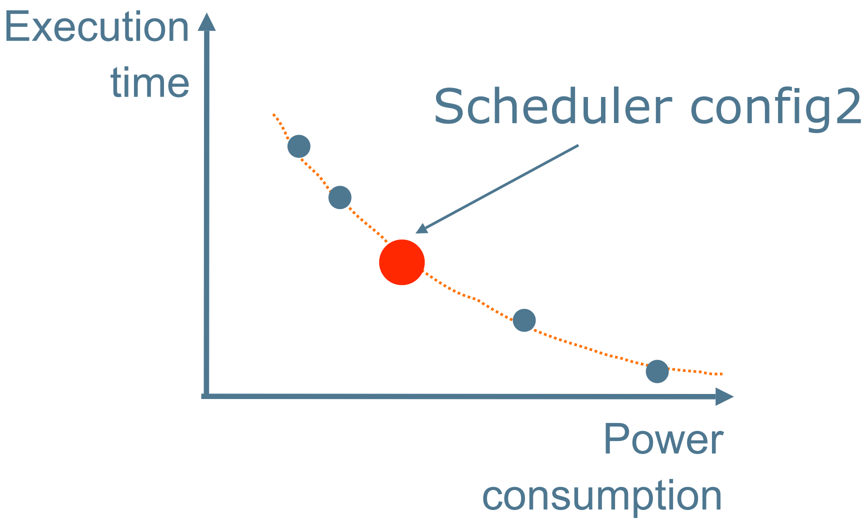




Run-time management

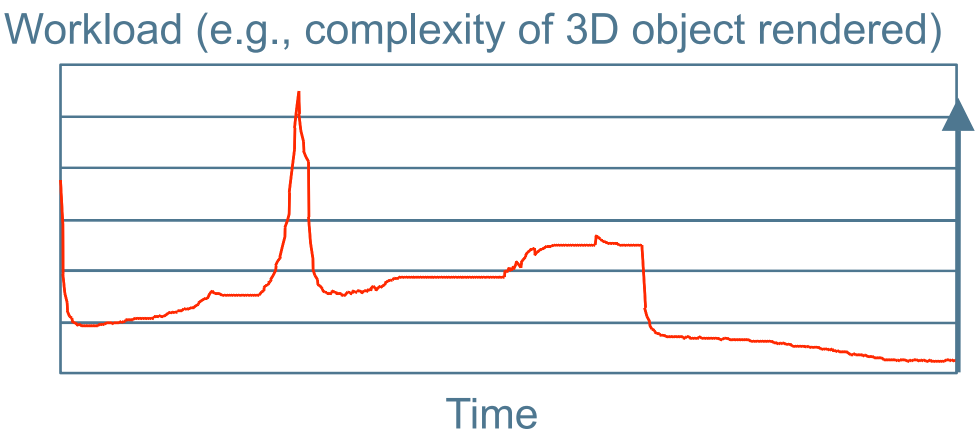
- For each thread frame, run-time scheduler changes management dynamically according to the run-time situations

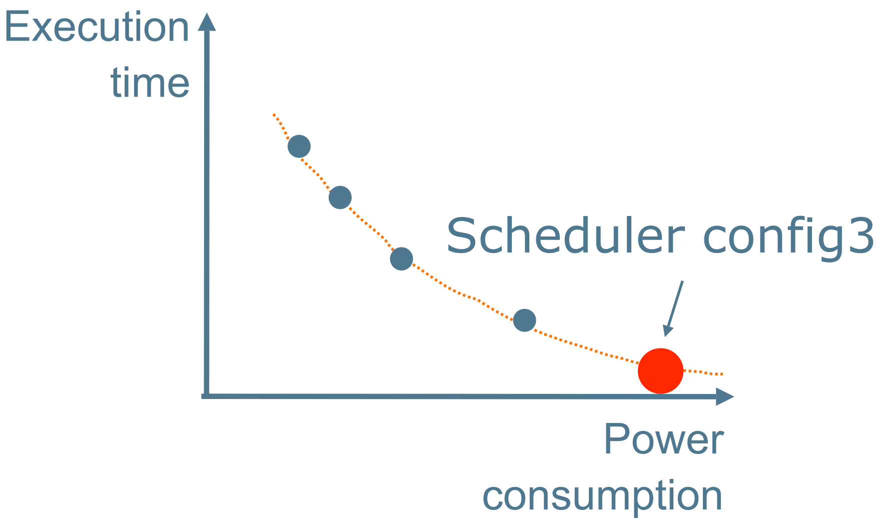




Run-time management

- For each thread frame, run-time scheduler changes management dynamically according to the run-time situations





Recapitulation

- Requirements: energy conservation, real-time constraints, time to market
- Issues tackled:
  - Map source code to hardware
  - Optimized code for various scenarios
  - Switch between scenarios at runtime
- Issues not tackled:
  - Huge design time effort
  - Applications are becoming more and more dynamic
  - Closed world assumption lifted: loading and unloading of new applications

Food for thought: Upcoming Issues

- Not too distant future:
  - Manycore chips
  - 3D chips
  - Chip Variability
  - Chip Unreliability
- Further away:
  - Biological Computing

imec