

PreMaDoNA

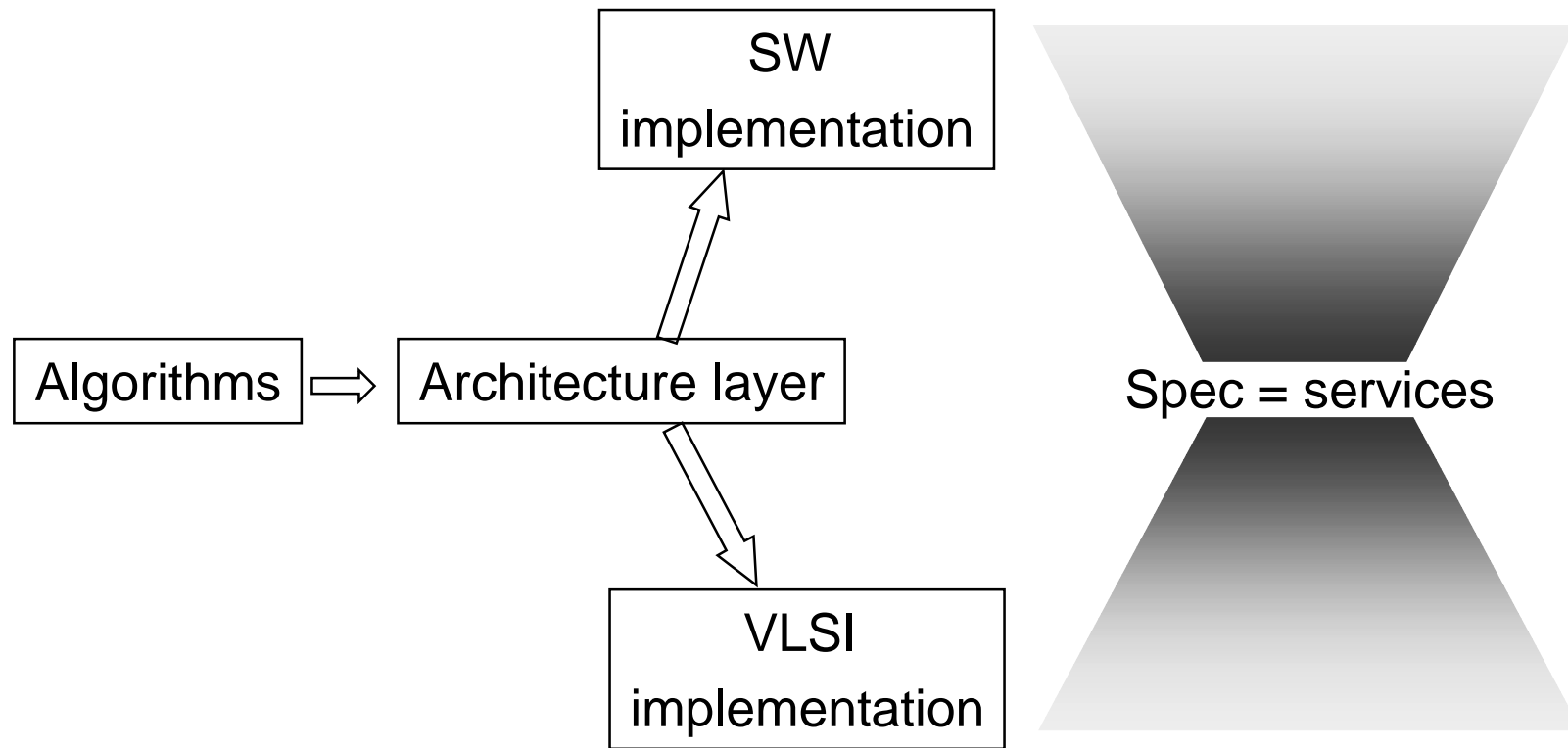
Predictable Matching of Demands on Networked Architectures

Track 1

Network Architecture Layer

J. van Meerbergen, K. Goossens, M. Bekooij

2 Architecture layer



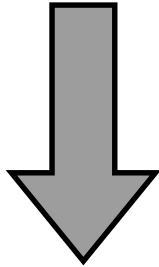
- specification = offered services e.g. GT connection
- impact of non-functional constraints
- similar role as instruction set architecture

3 Domain analysis: summary

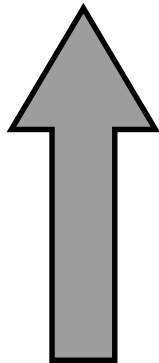
- Level 1: task
 - Compute intensive kernels
- Level 2: application/job
 - an activity (set of tasks) started/stopped by the user
- Level 3: system
 - Multiple jobs active simultaneously
 - Too many combinations to analyze at design time
- Constraints: Hard and soft RT, resources
 - must be seen as an input to the design and not as a check afterwards
- Objective function: maximize quality (or minimize power)

4 Increased unpredictability

Systems/
Applications



Architecture



VLSI Technology

- Late spec changes (What to design ?)
- Dynamic algorithms (e.g. object oriented...)
- Many Use-cases (800)

Deep submicron effects

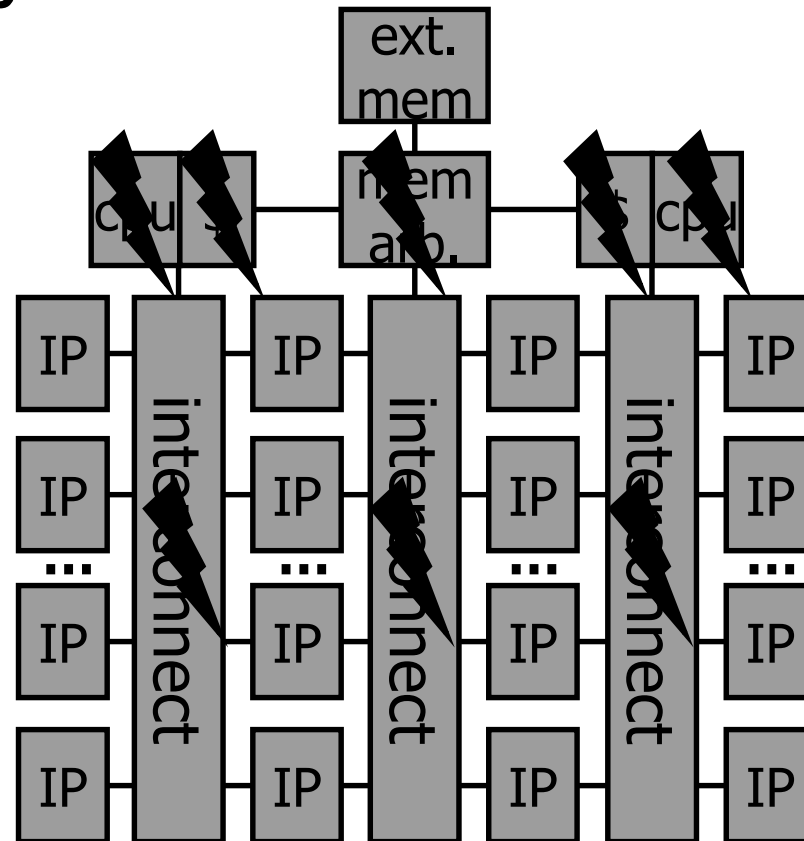
- wiring delay dominant over gate delay (timing closure)
- clock skew
- VDD and VSS voltage drop
- signal integrity, robust design

5 Increased unpredictability: Architecture

local schedulers

- (RT)OS
 - task switching
 - interrupts
- cache strategy
 - cache pollution
- interconnect
 - busses, bridges
 - networks
- memory controllers
 - external memory

e.g. RR, TDMA, FCFS,
LRU, EDF, FIFO,
priority, ...



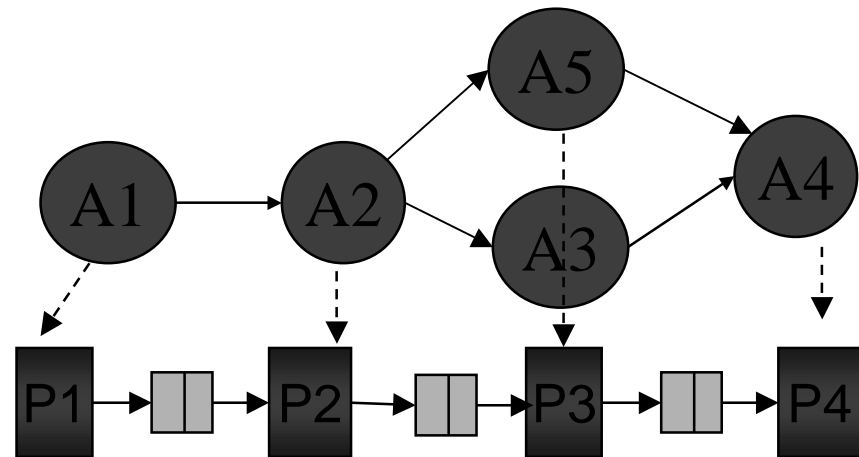
what is the global behaviour (end-to-end),
composed of interacting local solutions ?

6 What do we want ?

- Application/Job level
 - Reasoning about end-to-end timing constraints (for given resources and quality) = predictability
 - Which local arbitration mechanisms are needed ?
 - How to translate this to the global level ?

Example

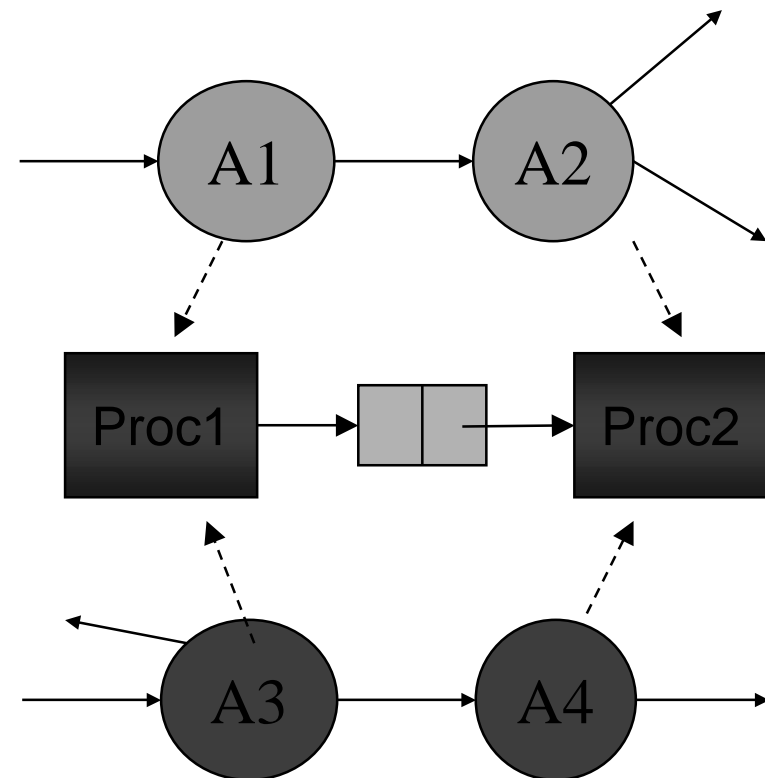
- given
 - Comp. Resources
 - Bandwidth
 - Buffer size
- ? throughput



7 What do we want ?

- System

- If app. 1 and app. 2 fit each individually, what can be said about the combination ?
- Compositionality
- Concept of virtual platform



8 Specification and HL architecture (concepts)

Set of services (APIs) and HL architecture

- Communication
- Computation: instruction set (GP and ASIP)
- Storage:
 - Cache, scratchpath or fifo ?
 - Configurable (instruction, data, coefficient)
 - distributed memories
- Generic support (OS-like)
 - Start/stop jobs = reconfiguration = resource manager
 - Synchronisation, communication
 - timers

Communication services: Aethereal example

- request communication services using connections
 - opening & closing affect resource reservations
- with properties
 - data integrity (uncorrupted data transfer)
 - transaction ordering
 - un/ordered per slave/connection
 - transaction completion
 - flow control
 - data loss or not
 - delivery bounds
 - throughput, latency, jitter

commitment

correctness

completion

bounds

Communication services: Aethereal example

- on a connection several transactions are possible
- transaction consists of messages
 - request message(s)
 - response message(s)

