technische universität dortmund

fakultät für informatik
informatik 12

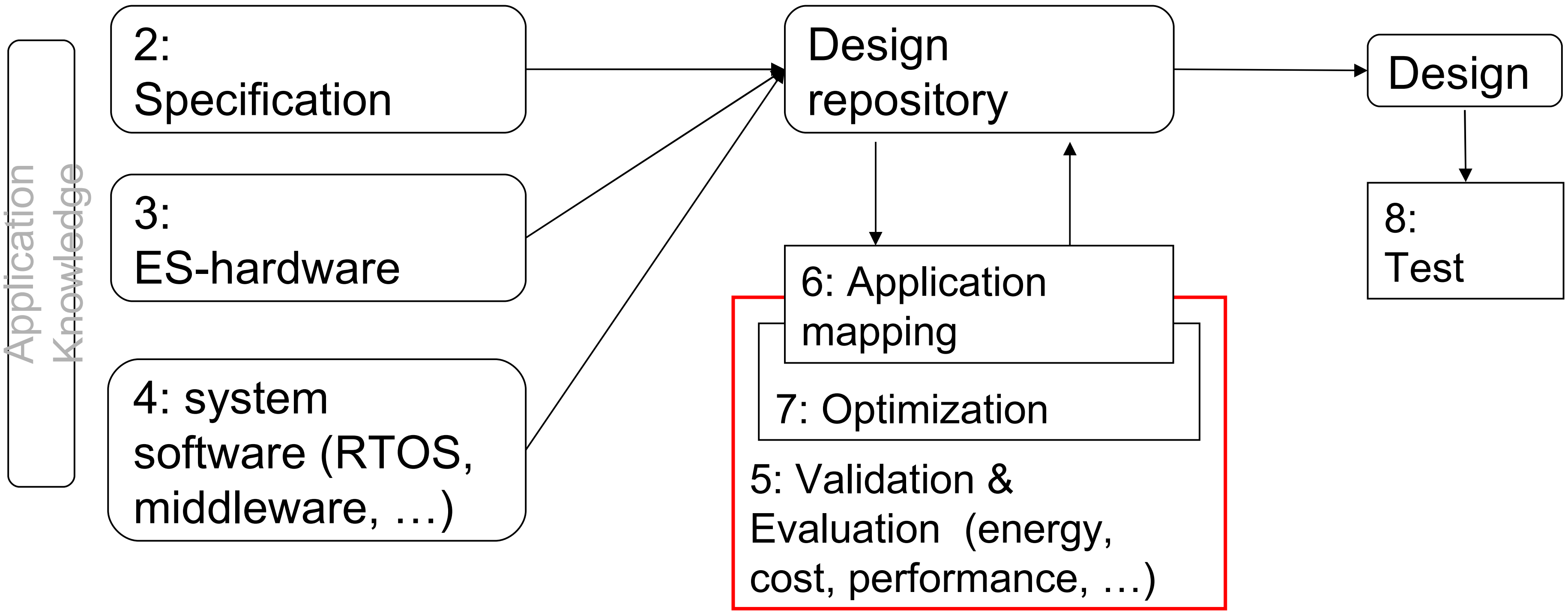
Evaluation and Validation

Peter Marwedel
TU Dortmund, Informatik 12
Germany

2009/12/09

Graphics: © Alexandra Nolte, Gesine Marwedel, 2003

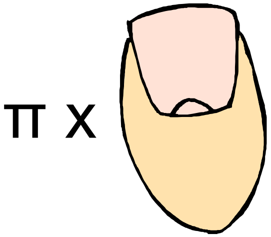
Structure of this course



Numbers denote sequence of chapters

Performance evaluation

- **Estimated cost and performance values:**
  Difficult to generate sufficiently precise estimates;
  Balance between run-time and precision



- **Accurate cost and performance values:**
  Can be done with normal tools (such as compilers).
  As precise as the input data is.

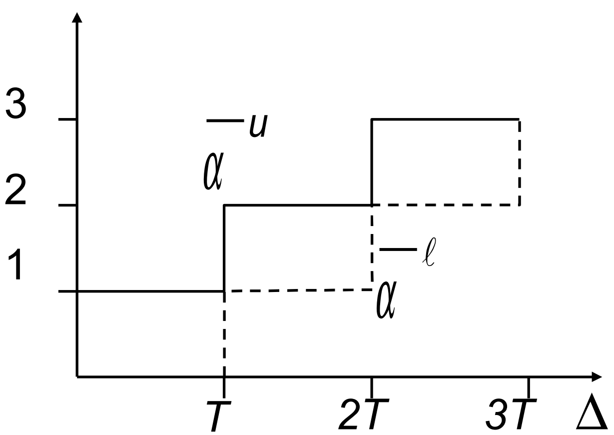
We need to compute average and worst case execution times

Thiele's real-time calculus (RTC)/ Modular performance analysis (MPA) - Arrival curves -

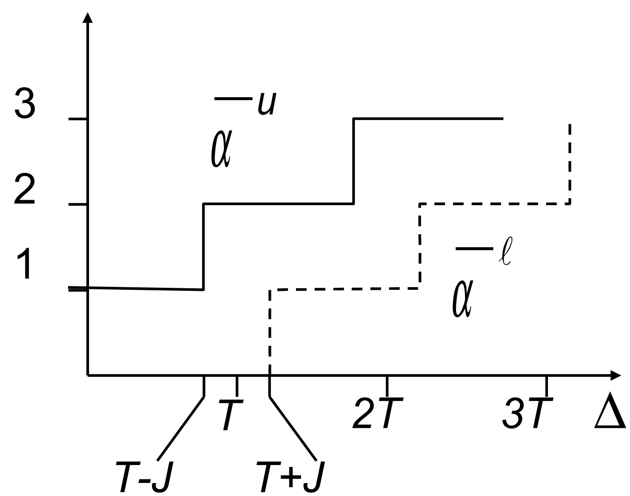
Arrival curves describe the maximum and minimum number of events arriving in some time interval Δ

Examples

periodic event stream



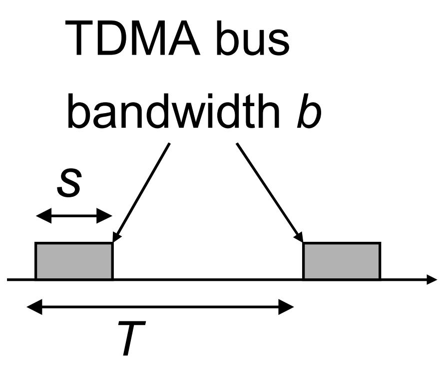
periodic event stream with jitter

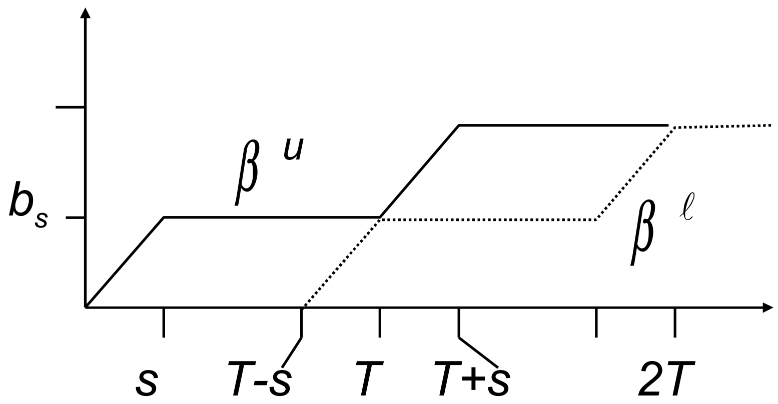


Thiele's real-time calculus (RTC)/ Modular performance analysis (MPA) - Service curves -

Service curves β^u resp. β^ℓ describe the maximum and minimum service capacity available in some time interval Δ

Example:

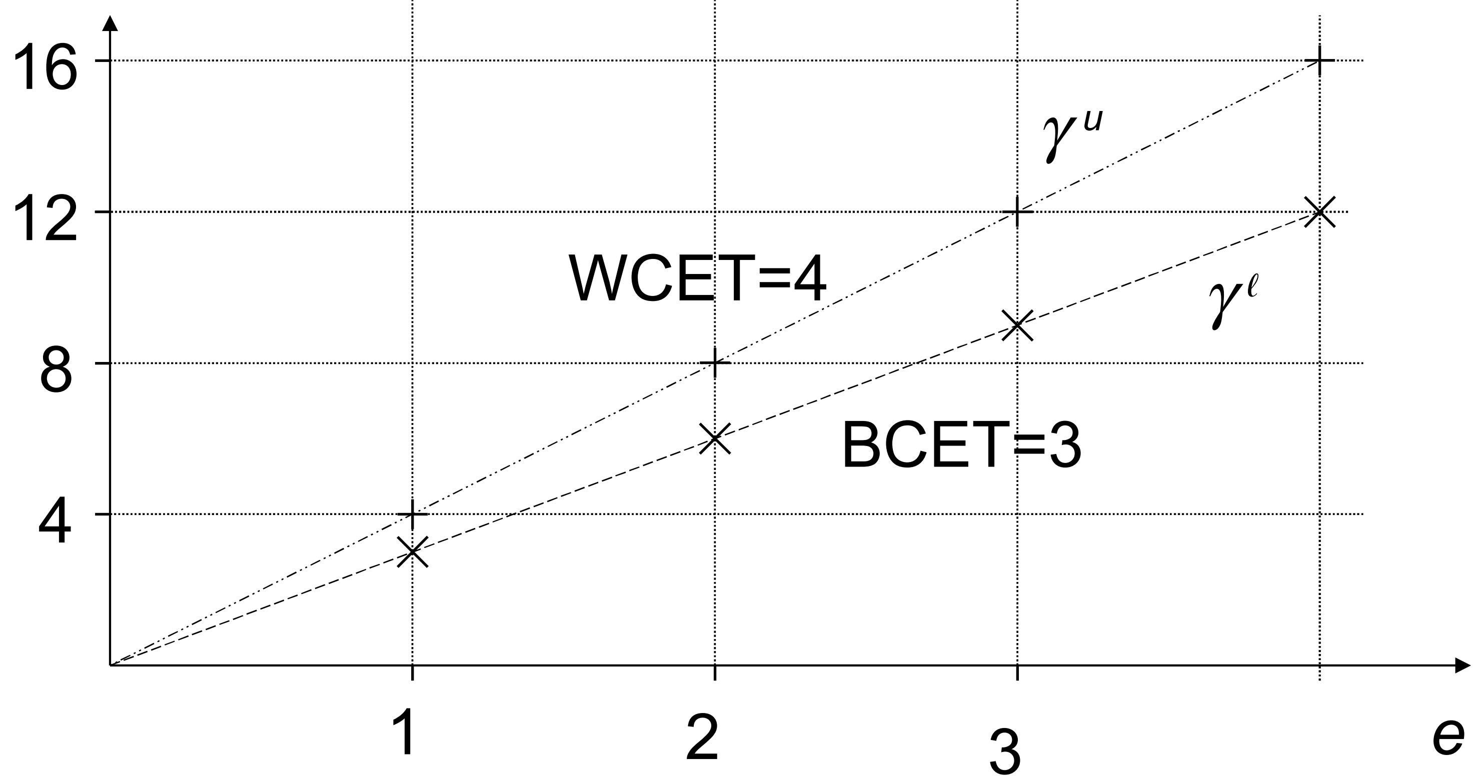




Thiele's real-time calculus (RTC)/ Modular performance analysis (MPA) - Workload characterization -

γ^u resp. γ^ℓ describe the maximum and minimum service capacity required as a function of the number *e* of events

Example



(Defined only for an integer number of events)

Workload required for incoming stream

Incoming workload

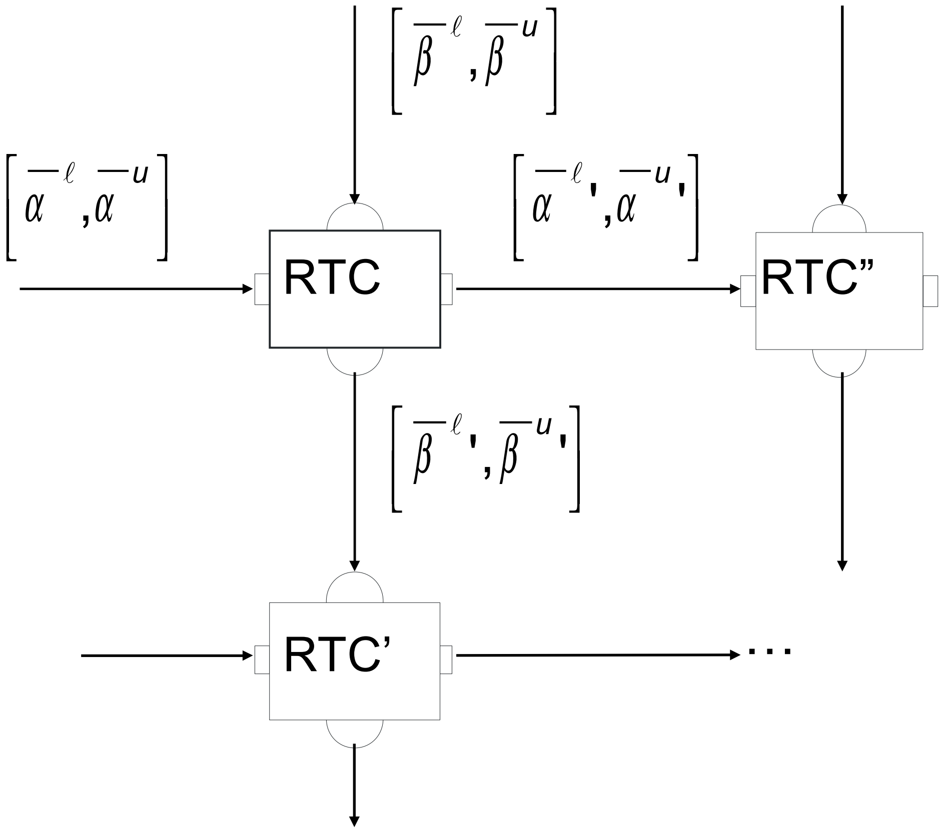
$$\alpha^{u}(\Delta) = \gamma^{u}\left(\overline{\alpha^{u}}(\Delta)\right) \qquad \alpha^{\ell}(\Delta) = \gamma^{\ell}\left(\overline{\alpha^{\ell}}(\Delta)\right)$$

Upper and lower bounds on the number of events

$$\overline{\beta}^{u}(\Delta) = \gamma^{-1}\left(\beta^{u}(\Delta)\right) \qquad \overline{\beta}^{\ell}(\Delta) = \gamma^{-1}\left(\beta^{\ell}(\Delta)\right)$$

Thiele's real-time calculus (RTC)/ Modular performance analysis (MPA) - System of real time components -

Incoming event streams and available capacity are transformed by real-time components:



Theoretical results allow the computation of properties of outgoing streams ☞

Thiele's real-time calculus (RTC)/ Modular performance analysis (MPA) - Transformation of arrival and service curves

Resulting arrival curves:

$$\bar{\alpha}^{u}{}' = \min\left(\left\lfloor \left(\bar{\alpha}^{u} \,\underline{\otimes}\, \bar{\beta}^{u}\right) \overline{\oplus}\, \bar{\beta}^{l} \right\rfloor, \bar{\beta}^{u}\right)$$

$$\bar{\alpha}^{\ell}{}' = \min\left(\left\lfloor \left(\bar{\alpha}^{\ell} \,\overline{\oplus}\, \bar{\beta}^{u}\right) \underline{\otimes}\, \bar{\beta}^{\ell} \right\rfloor, \bar{\beta}^{\ell}\right)$$

Remaining service curves:

$$\bar{\beta}^{u}{}' = \left(\bar{\beta}^{u} - \bar{\alpha}^{\ell}\right) \underline{\oplus}\, 0$$

$$\bar{\beta}^{\ell}{}' = \left(\bar{\beta}^{\ell} - \bar{\alpha}^{u}\right) \overline{\otimes}\, 0$$

Where:

$$\left(f \,\underline{\otimes}\, g\right)(t) = \inf_{0 \le u \le t}\left\{f(t-u) + g(u)\right\} \qquad \left(f \,\overline{\otimes}\, g\right)(t) = \sup_{0 \le u \le t}\left\{f(t-u) + g(u)\right\}$$

$$\left(f \,\underline{\oplus}\, g\right)(t) = \inf_{u \ge 0}\left\{f(t+u) - g(u)\right\} \qquad \left(f \,\overline{\oplus}\, g\right)(t) = \sup_{u \ge 0}\left\{f(t+u) - g(u)\right\}$$

Thiele's real-time calculus Remarks

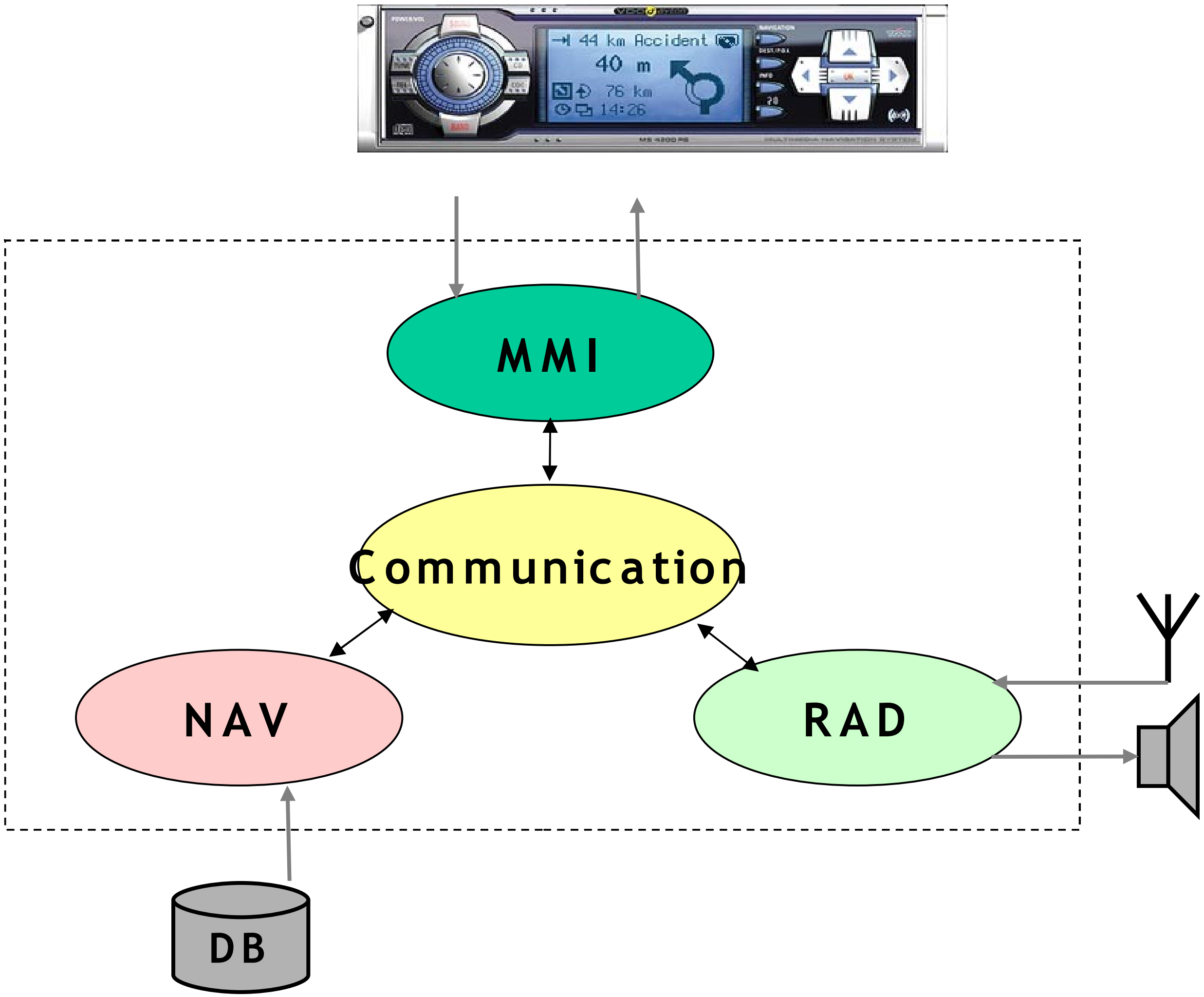
- Details of the proofs can be found in relevant references
- Results also include bounds on buffer sizes and on maximum latency.
- Theory has been extended into various directions, e.g. for computing remaining battery capacities

Application: In-Car Navigation System

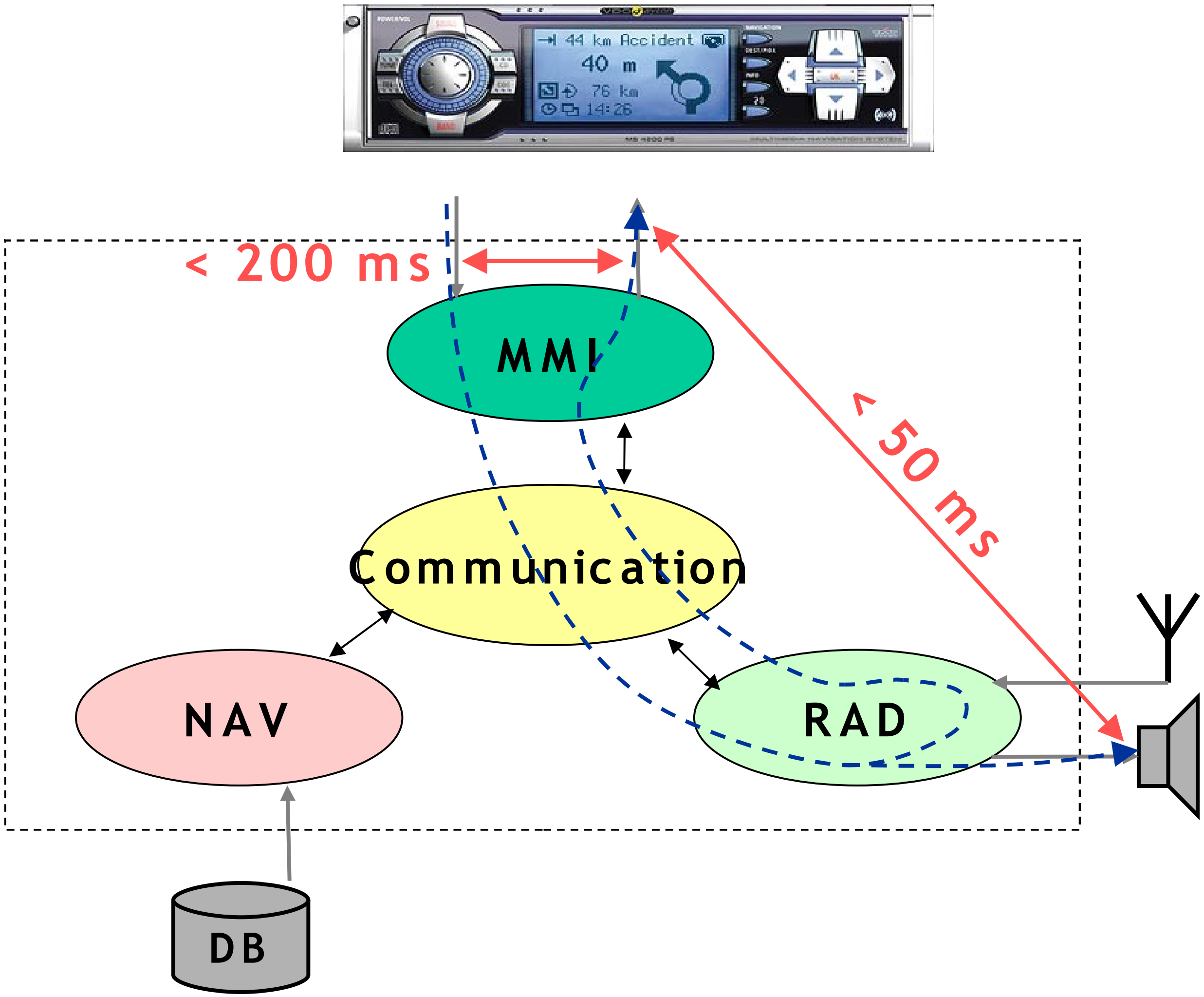
Car radio with navigation system
User interface needs to be responsive
Traffic messages (TMC) must be processed in a timely way
Several applications may execute concurrently

© Thiele, ETHZ

System Overview

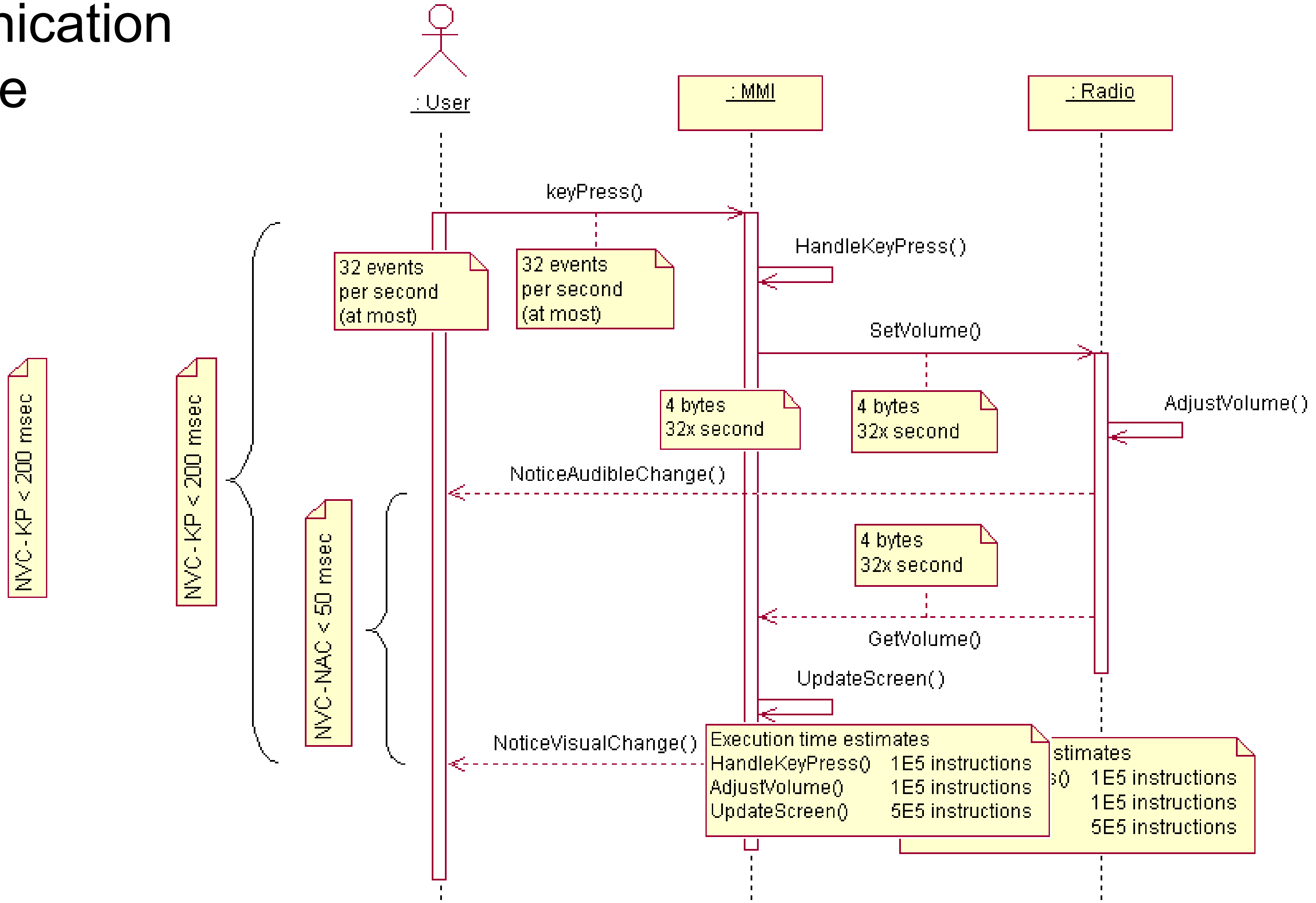


Use case 1: Change Audio Volume

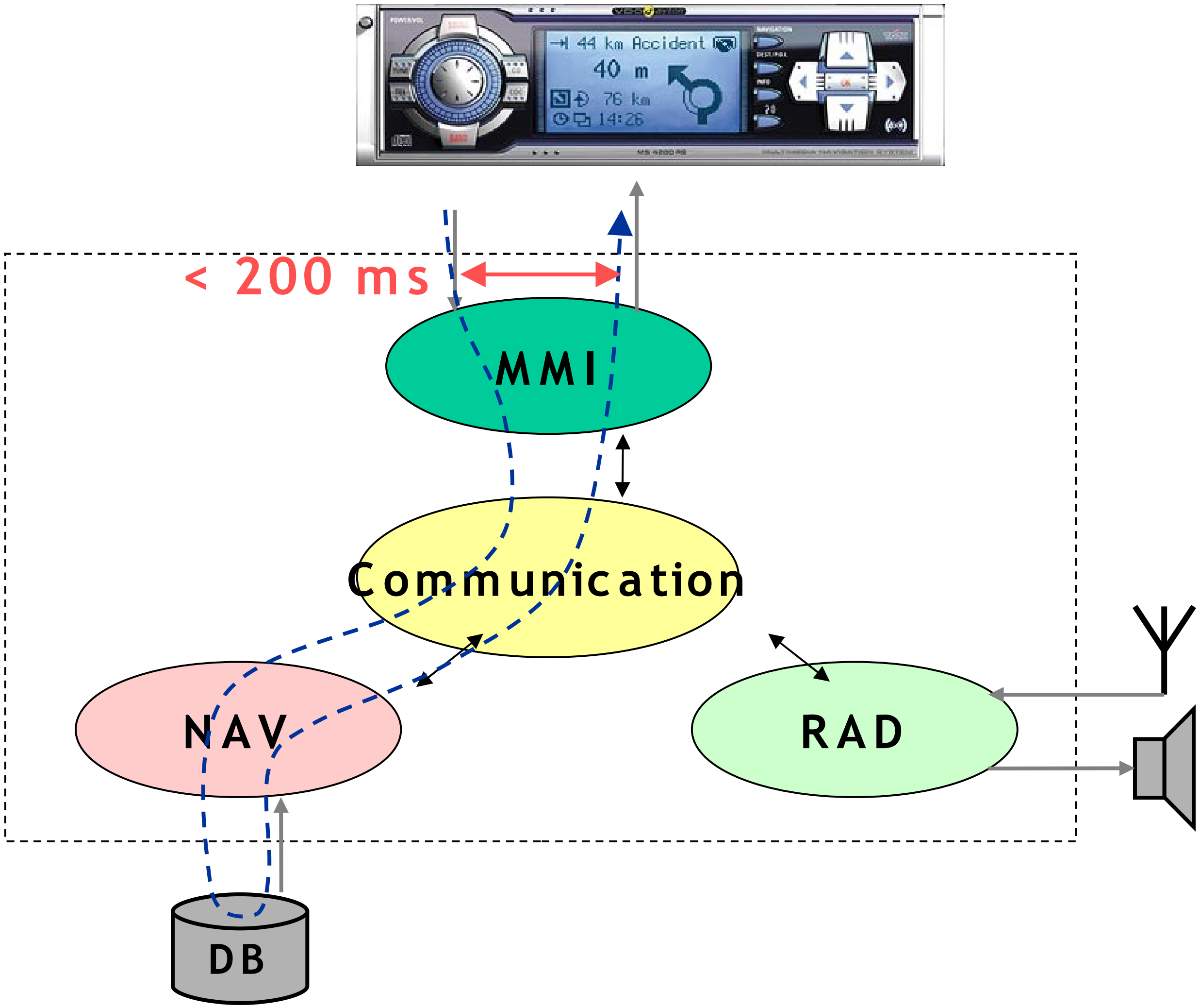


Use case 1: Change Audio Volume

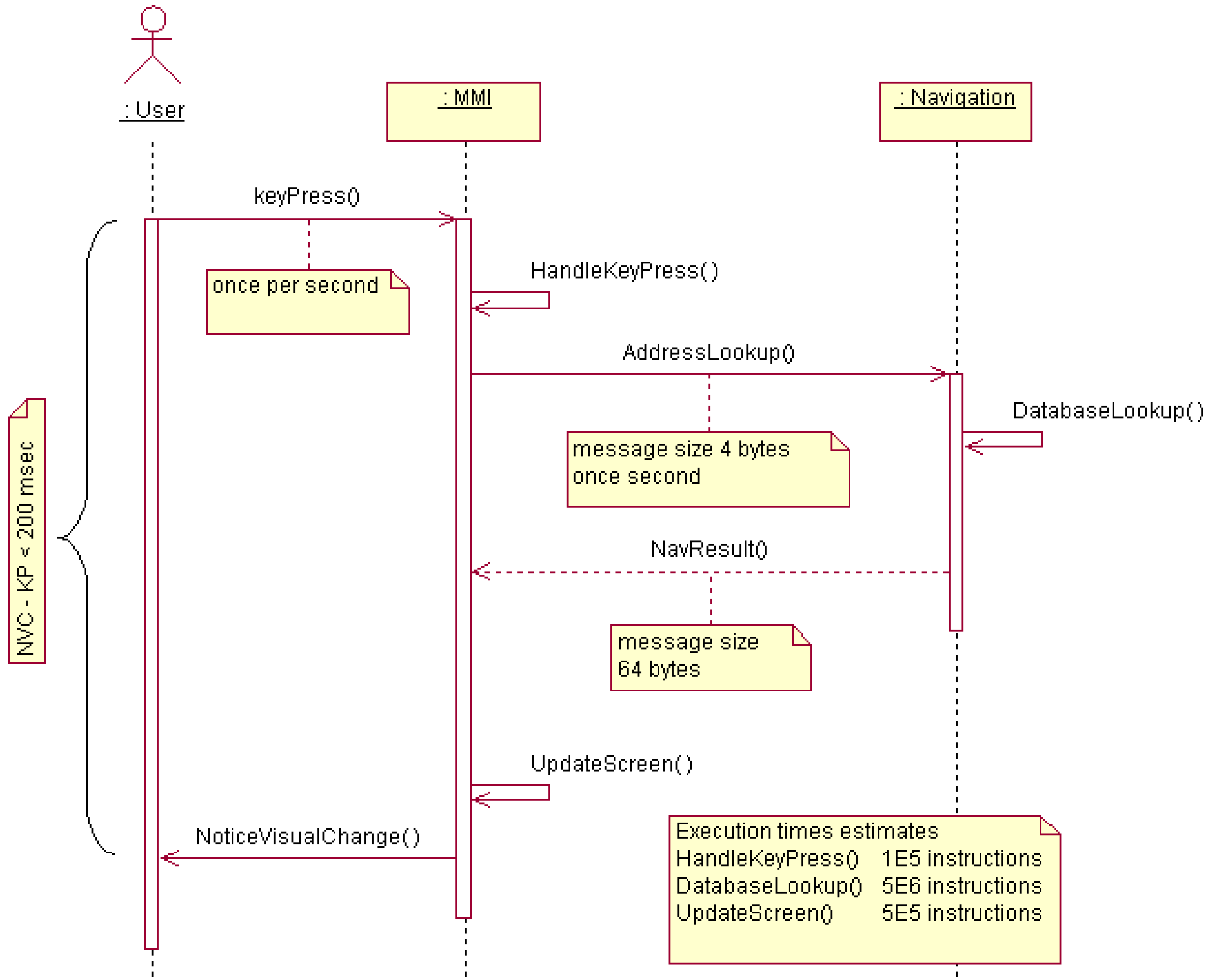
Communication
Resource
Demand



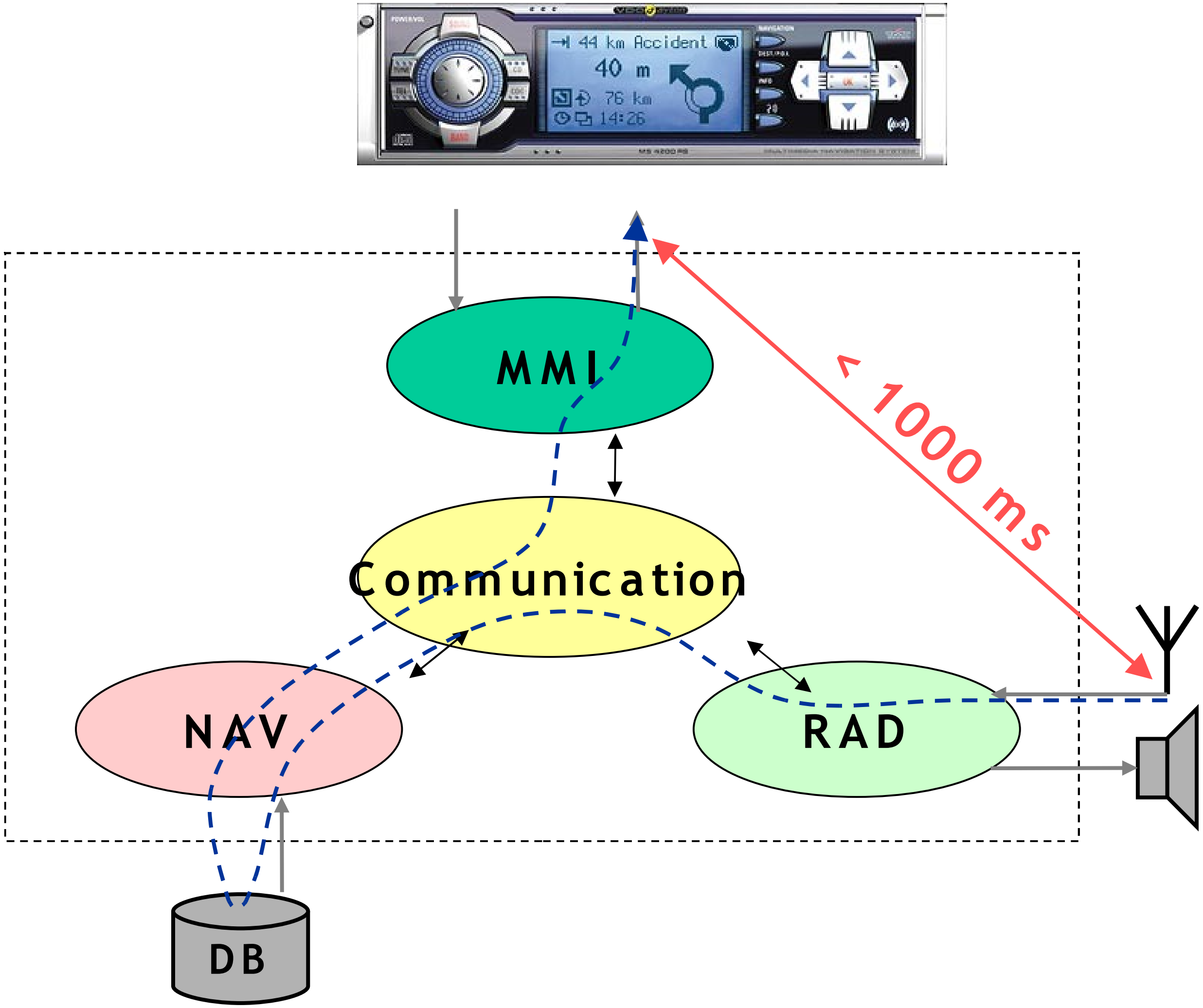
Use case 2: Lookup Destination Address



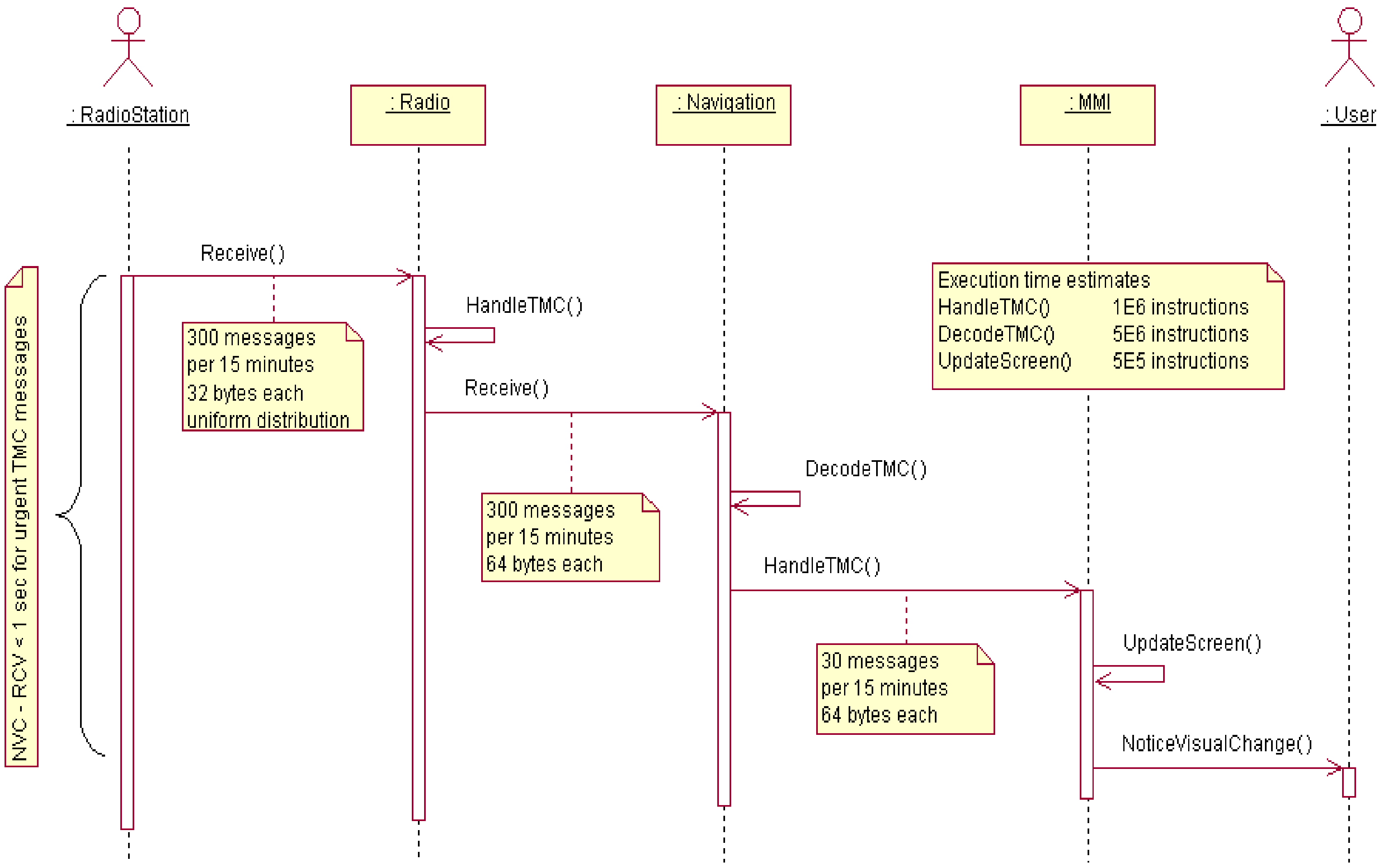
Use case 2: Lookup Destination Address



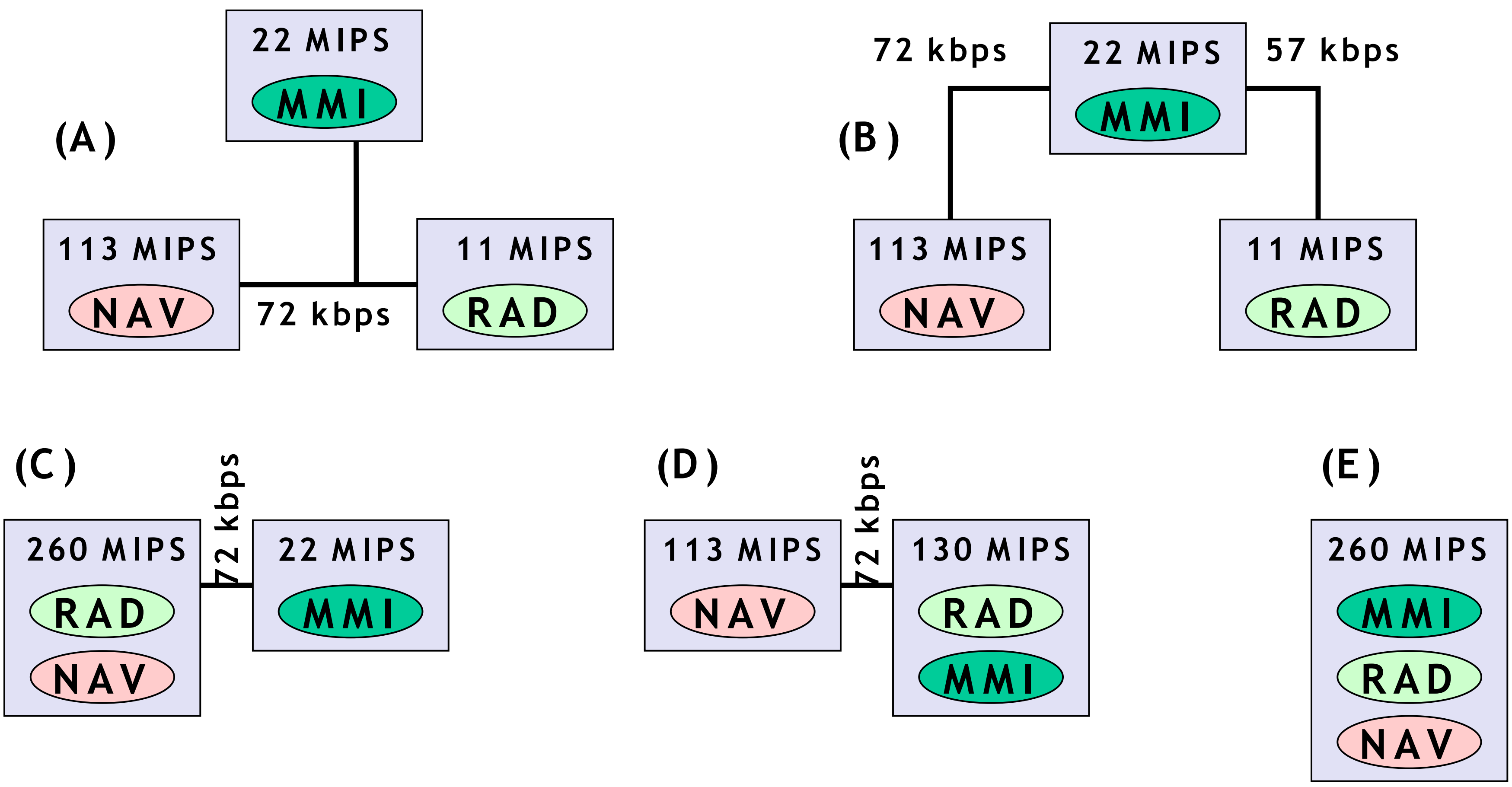
Use case 3: Receive TMC Messages



Use case 3: Receive TMC Messages



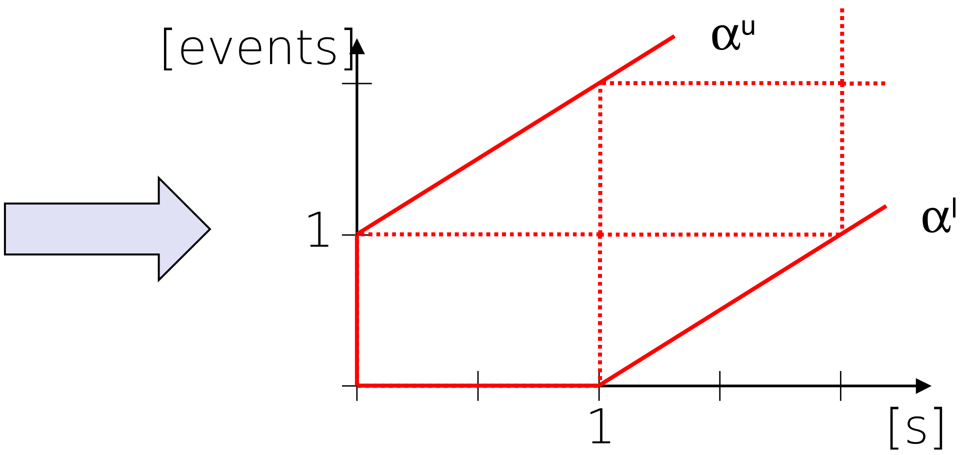
Proposed Architecture Alternatives



Step 1: Environment (Event Steams)

Event Stream Model

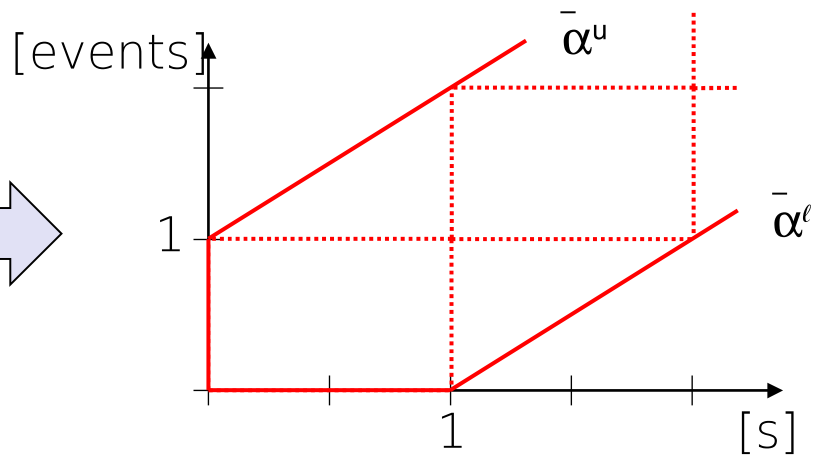
e.g. Address Lookup
(1 event / sec)



Step 2: Architectural Elements

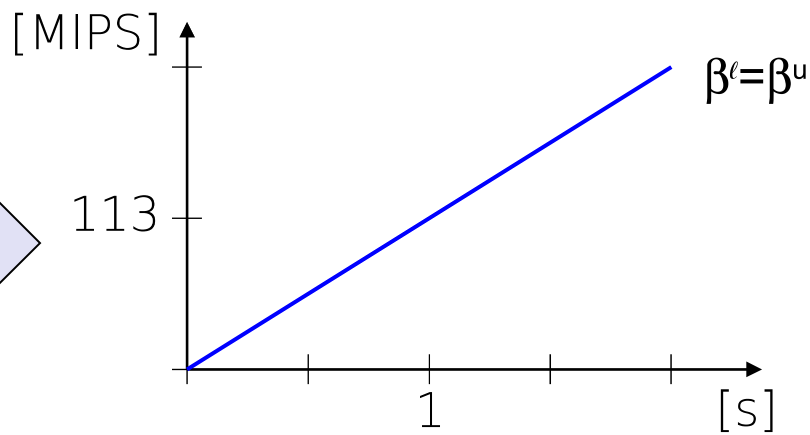
Event Stream Model

e.g. Address Lookup
(1 event / sec)



Resource Model

e.g. unloaded RISC CPU
(113 MIPS)

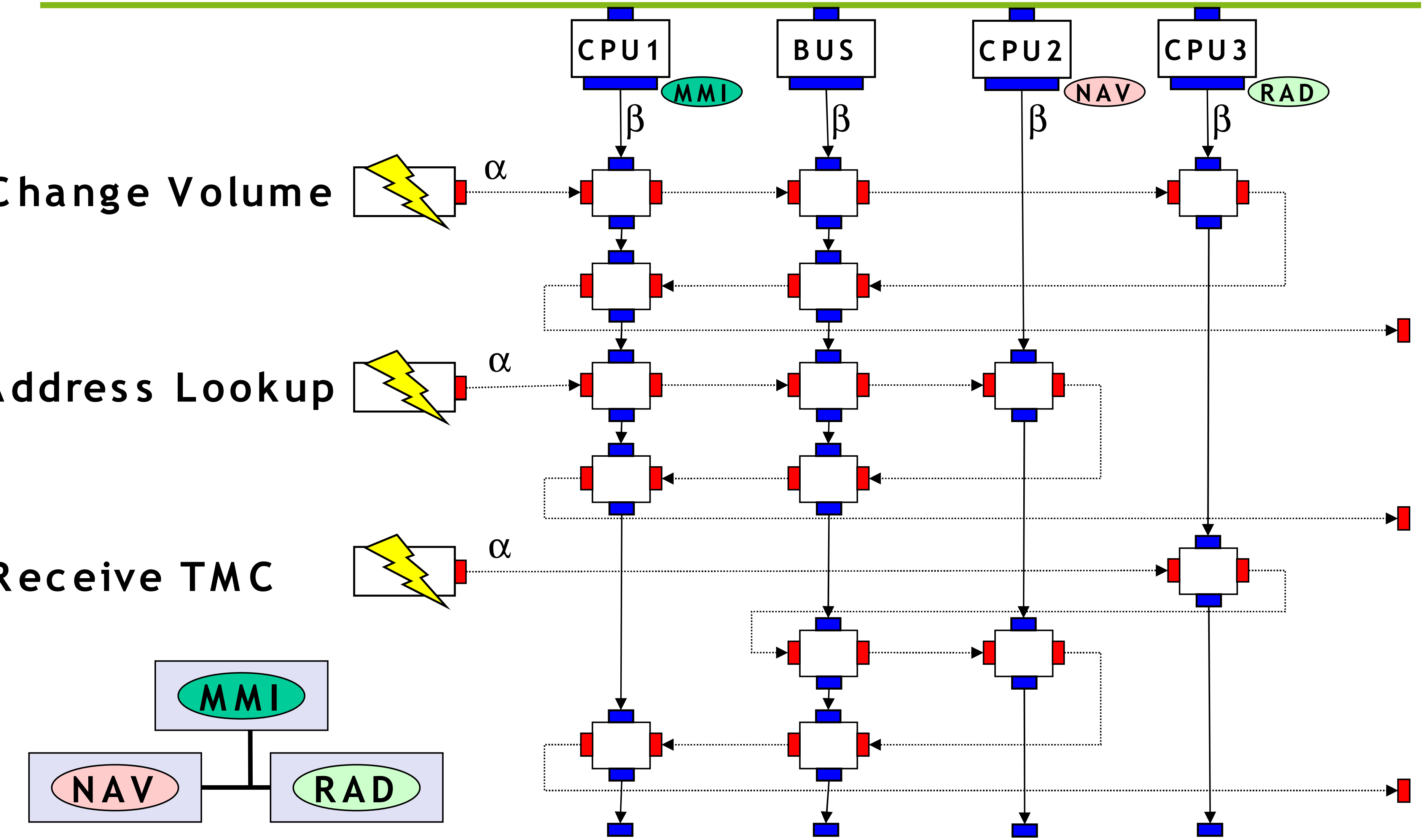


Step 3: Mapping / Scheduling

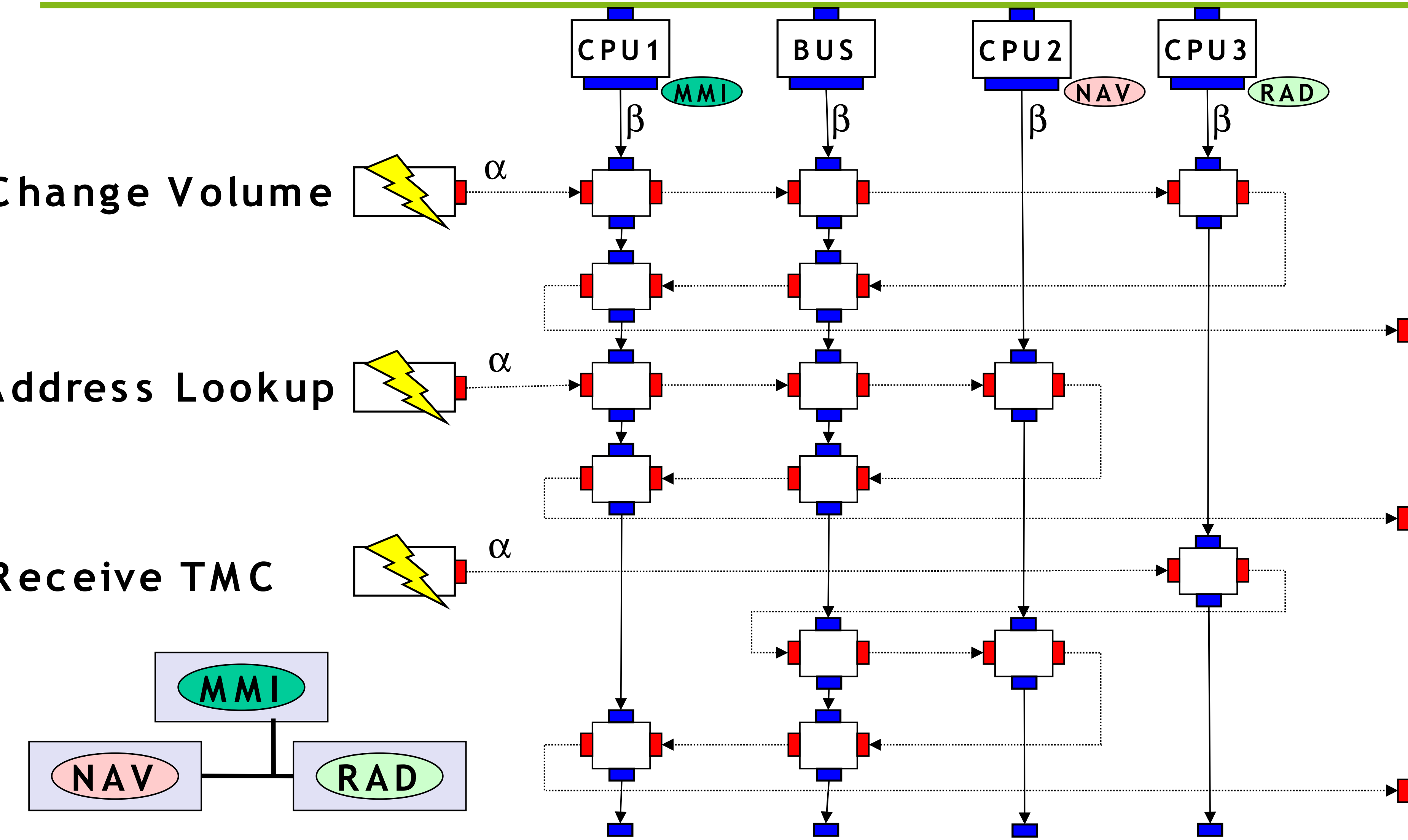
Rate Monotonic Scheduling
(Pre-emptive fixed priority scheduling):

- Priority 1: Change Volume (p=1/32 s)
- Priority 2: Address Lookup (p=1 s)
- Priority 3: Receive TMC (p=6 s)

Step 4: Performance Model



Step 5: Analysis

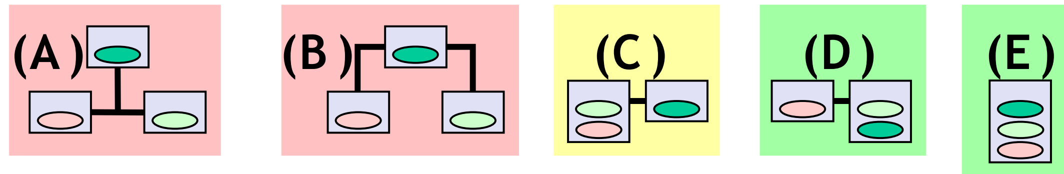


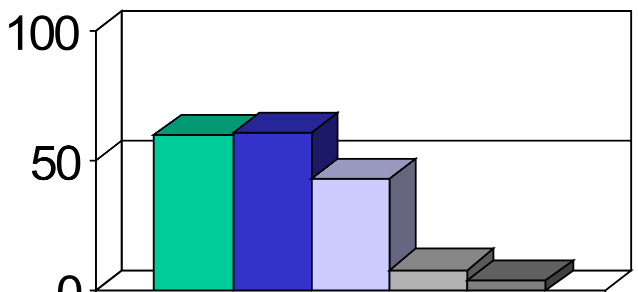
Analysis – Design Question 1

How do the proposed system architectures compare in respect to end-to-end delays?

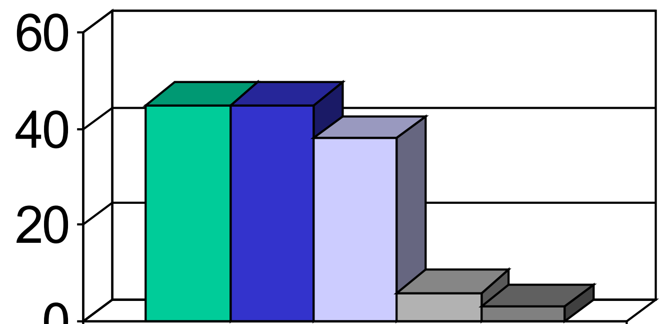
Analysis – Design Question 1

End-to-end delays:

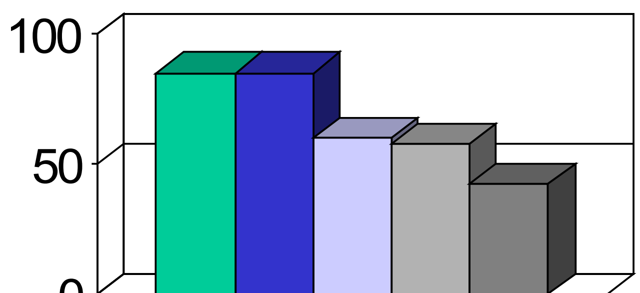




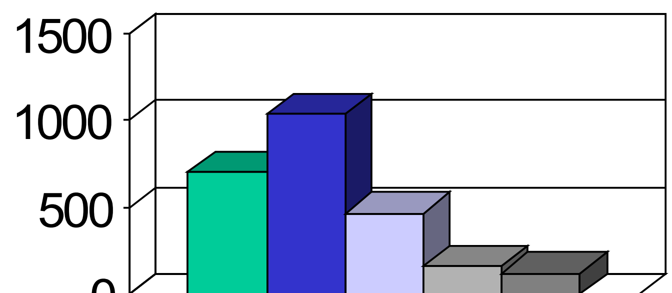
Vol Key 2 Audio [ms]



Vol Vis. 2 Audio [ms]



Address Lookup [ms]

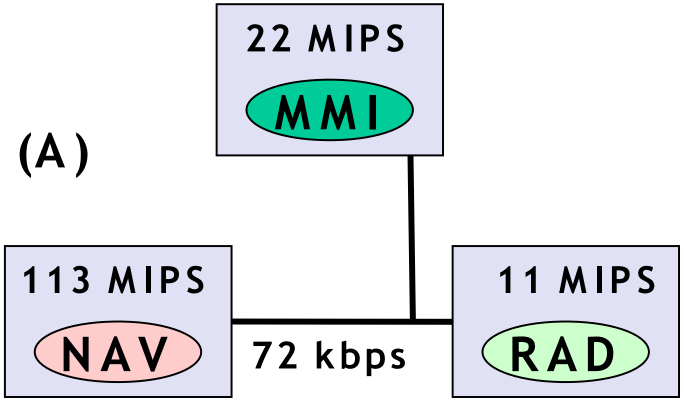


TMC Decode [ms]

Analysis – Design Question 2

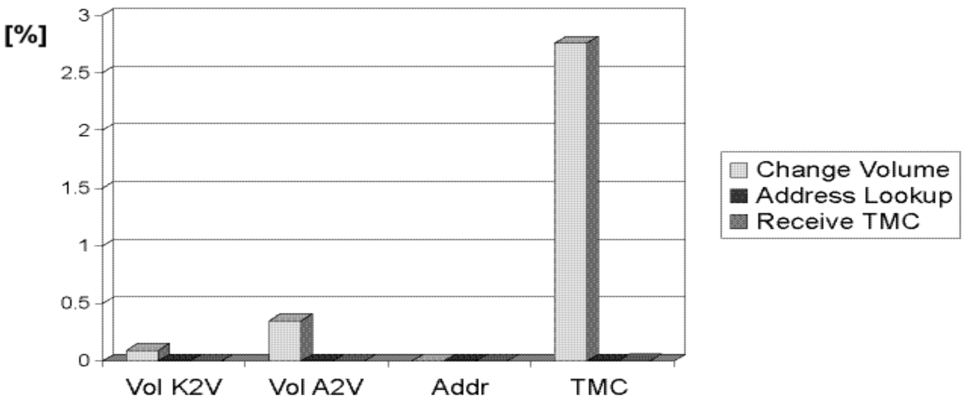
How robust is architecture A?

Where is the bottleneck of this architecture?

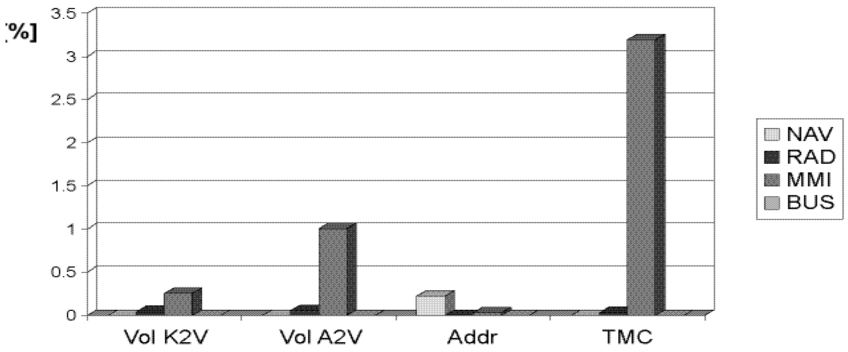


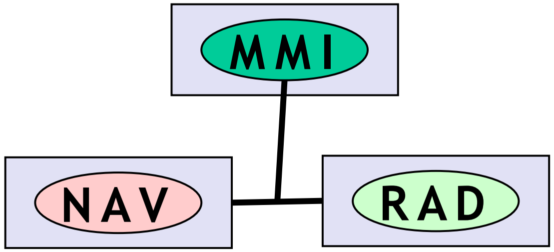
Analysis – Design Question 2

Sensitivity to input rate:



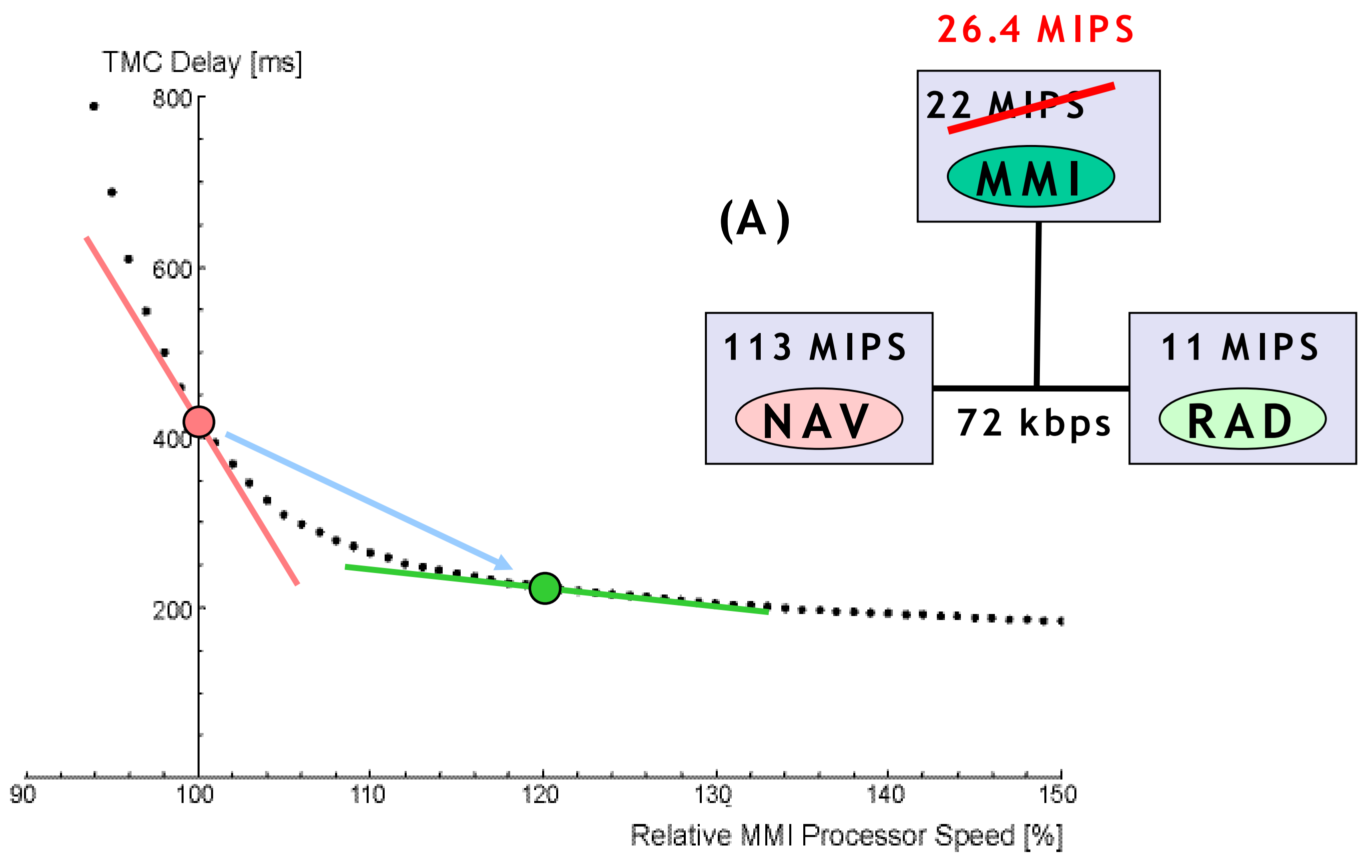
Sensitivity to resource capacity:





Analysis – Design Question 2

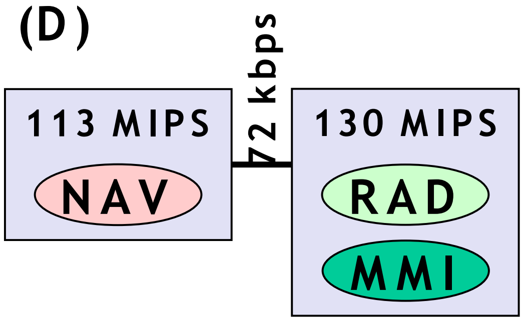
TMC delay vs. MMI processor speed:



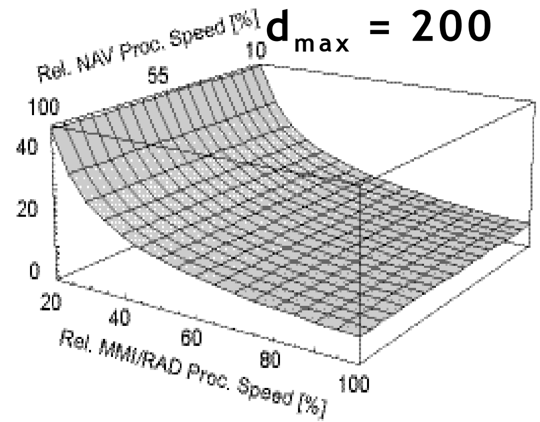
Analysis – Design Question 3

Architecture D is chosen for further investigation.

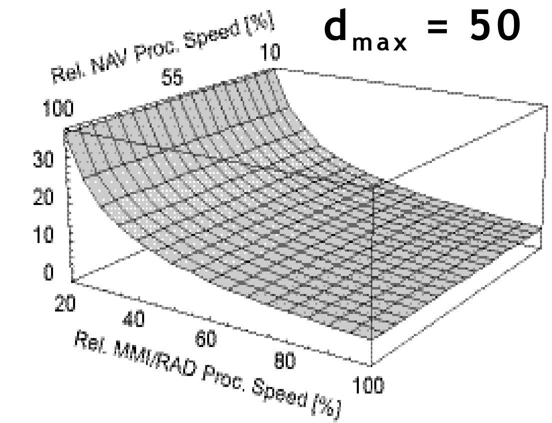
How should the processors be dimensioned?



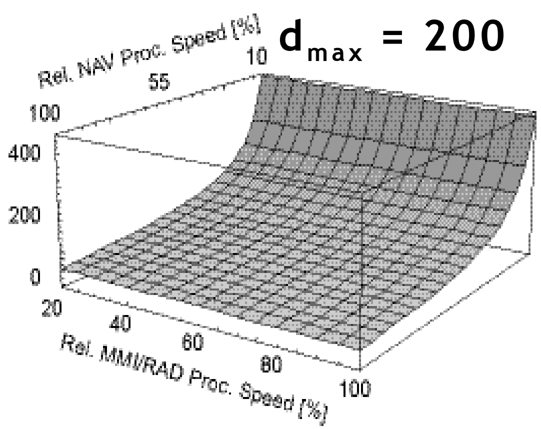
Analysis – Design Question 3



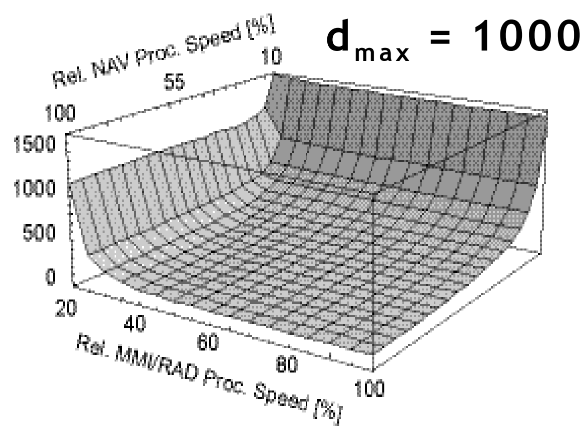
Vol K2V Delay [ms]



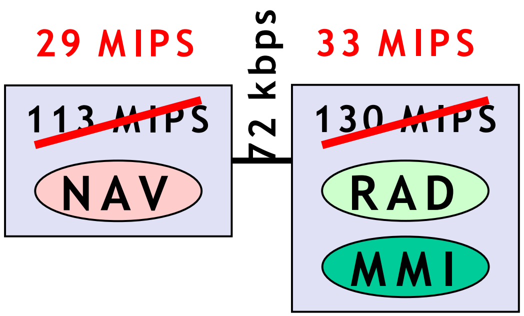
Vol A2V Delay [ms]



Addr Delay [ms]



TMC Delay [ms]



Conclusions

- Easy to construct models (~ half day)
- Evaluation speed is fast and linear to model complexity (~ 1s per evaluation)
- Needs little information to construct early models (Fits early design cycle very well)
- Even though involved mathematics is very complex, the method is easy to use (Language of engineers)

Acknowledgement and References

The presentation contains contributions by

- Samarjit Chakraborty (NUS)
- Simon Künzli, Ernesto Wandeler, Alexander Maxiaguine (ETHZ)
- Andreas Herkersdorf, Patricia Sagmeister (IBM)
- Jonas Greutert (Netmodule)

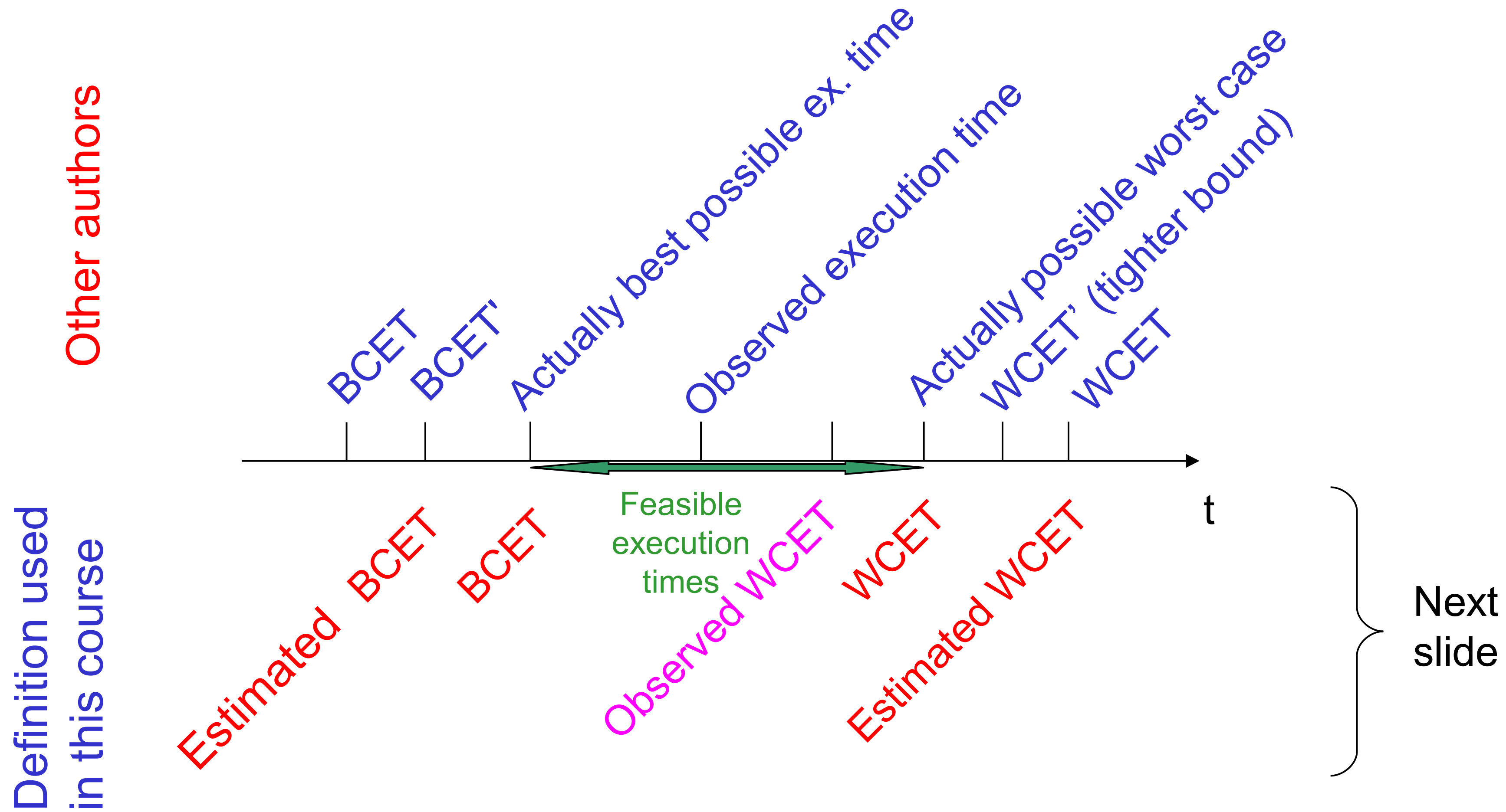
Many publications are available from
http://www.tik.ee.ethz.ch/~thiele

SYMTA

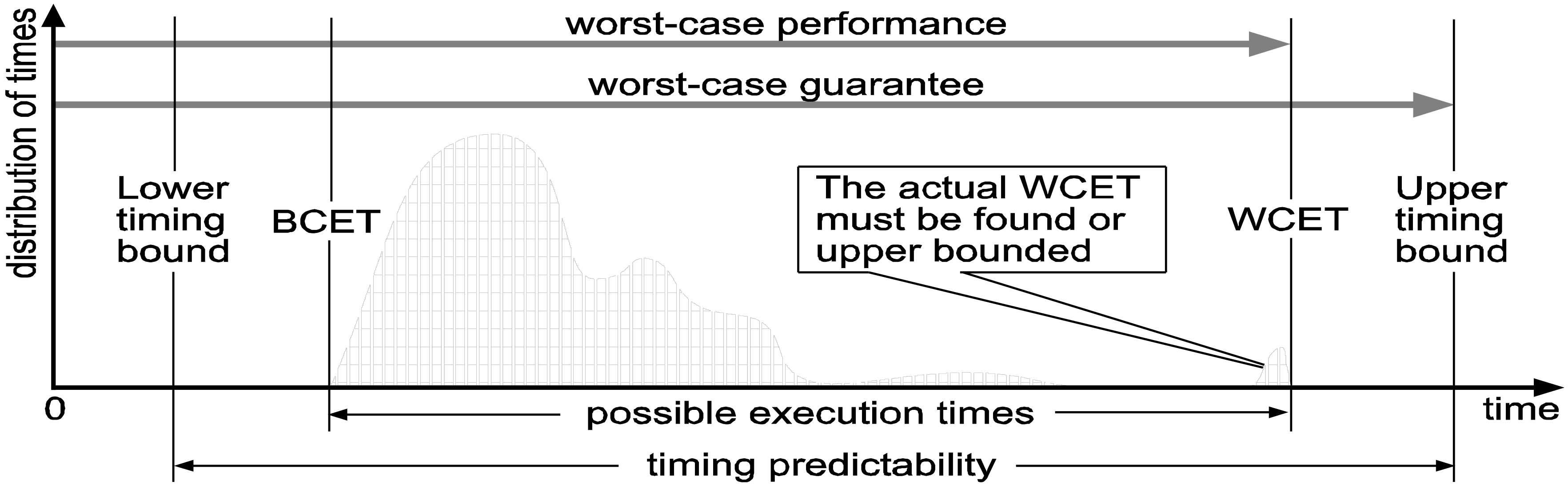
SYMTA (Ernst, TU Braunschweig) works with standard streams (periodic, periodic with jitter streams) and focuses on computing end-to-end guarantees in multiprocessor based systems

See www.symtavision.com

Worst/best case execution times (1)



Worst/best case execution times (2)



$WCET_{EST}$

Requirements on WCET estimates:

- *Safeness:* WCET ≤ $WCET_{EST}$!
- *Tightness:* $WCET_{EST}$ – WCET → minimal

Worst case execution times (3)

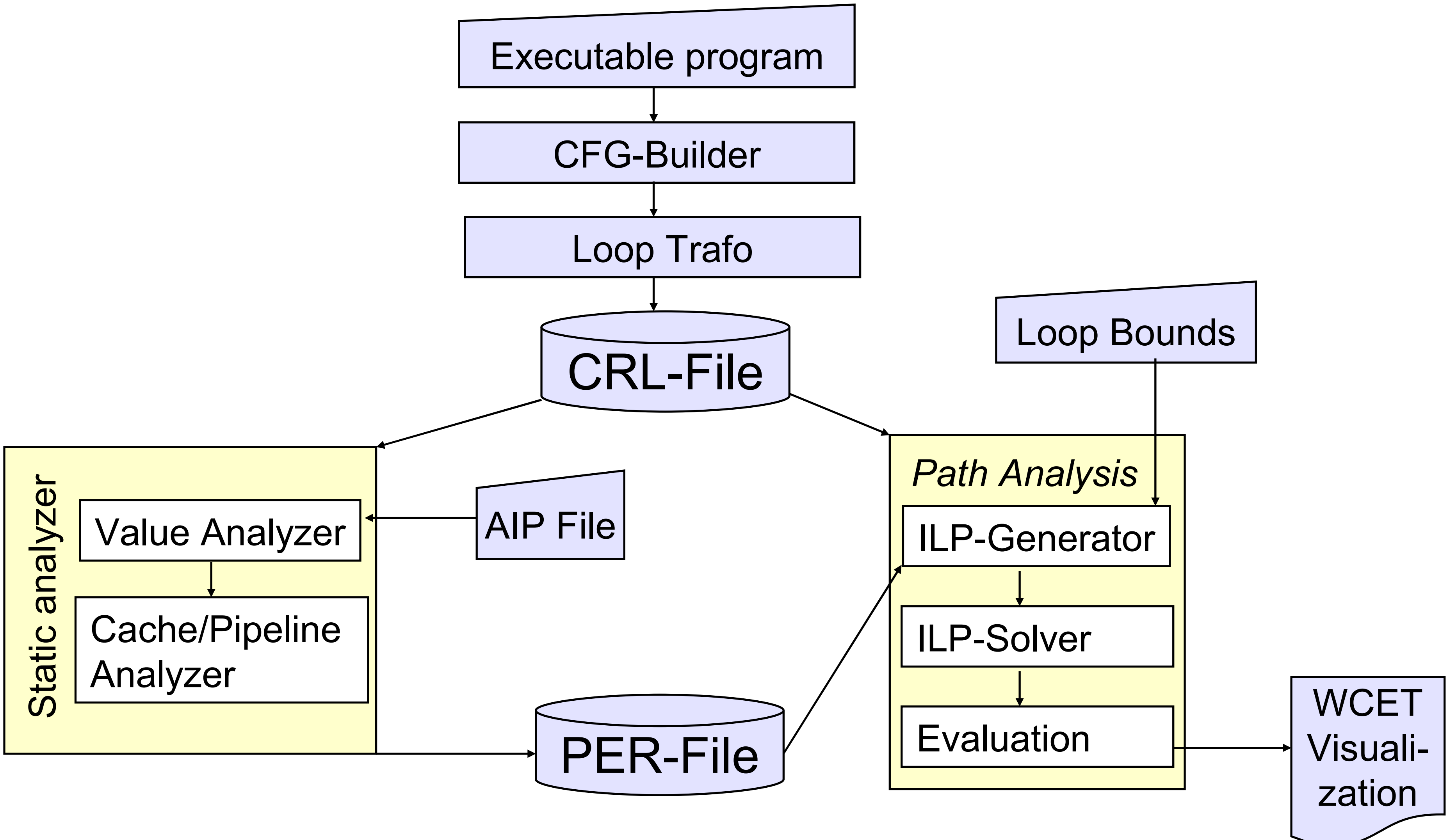
Complexity:

- in the general case: undecidable if a bound exists.
- for restricted programs: simple for “old“ architectures, very complex for new architectures with pipelines, caches, interrupts, virtual memory, etc.

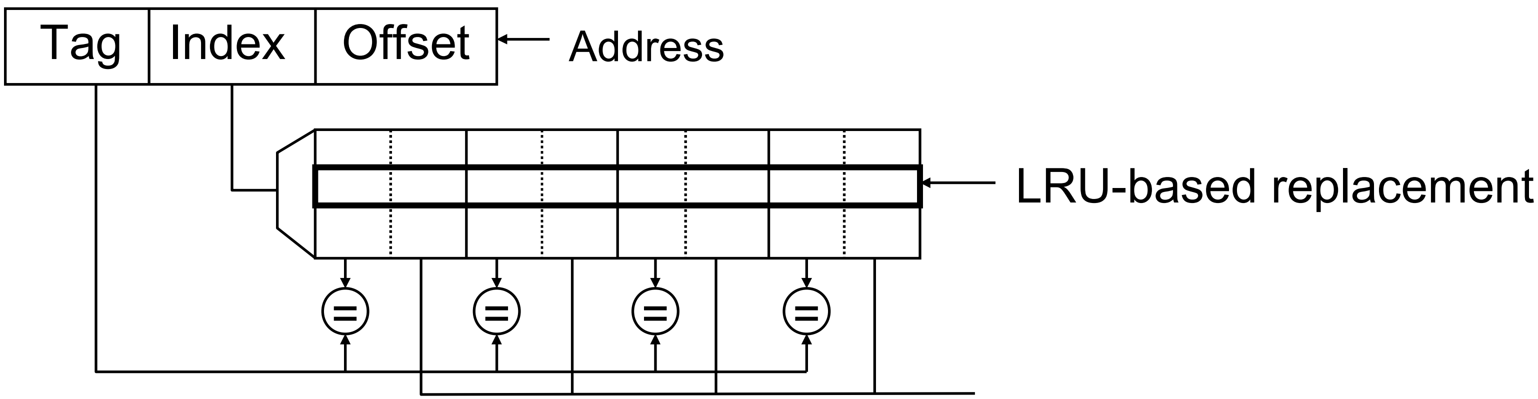
Approaches:

- for hardware: requires detailed timing behavior
- for software: requires availability of machine programs; complex analysis (see, e.g., www.absint.de)

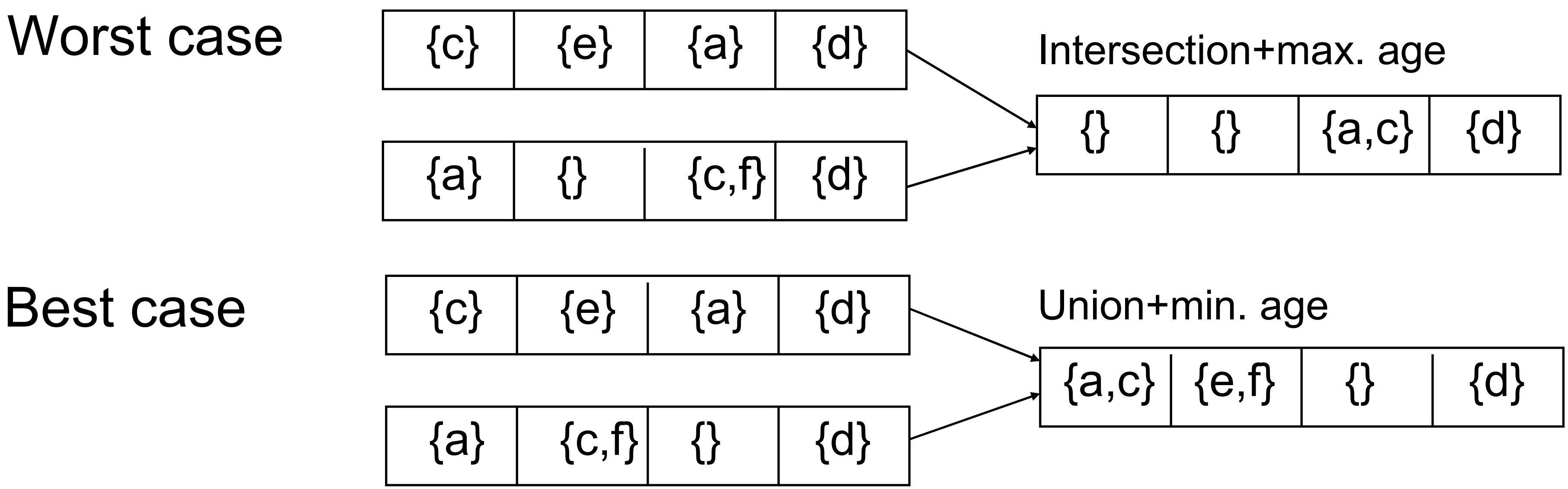
WCET estimation: AiT (AbsInt)



WCET estimation for caches



Behavior at program joins

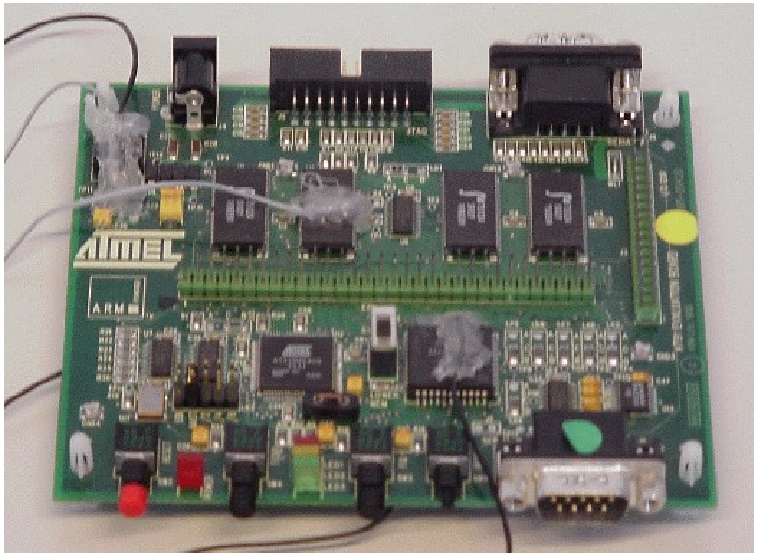


Energy models

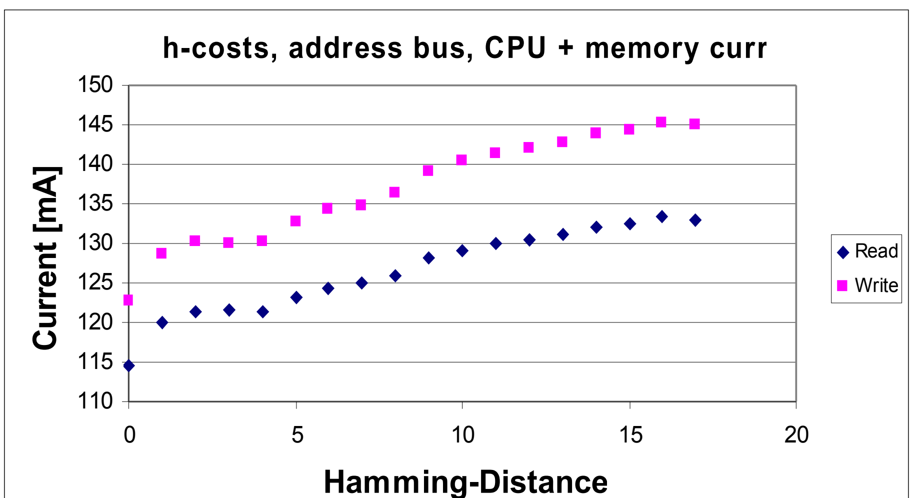
- Tiwari (1994): Energy consumption within processors
- Simunic (1999): Using values from data sheets. Allows modeling of all components, but not very precise.

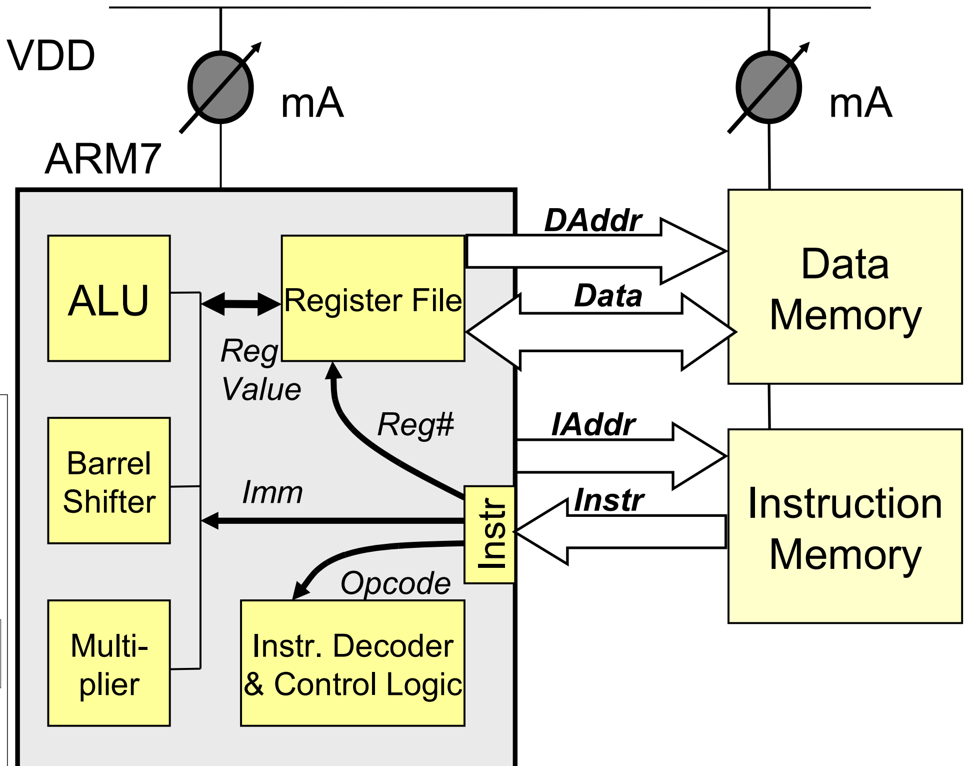
- Russell, Jacome (1998): Measurements for 2 fixed configurations
- Steinke et al., UniDo (2001): mixed model using measurements and prediction
- Jouppi (1996): Energy consumption of caches predicted by CACTI.

Steinke's model

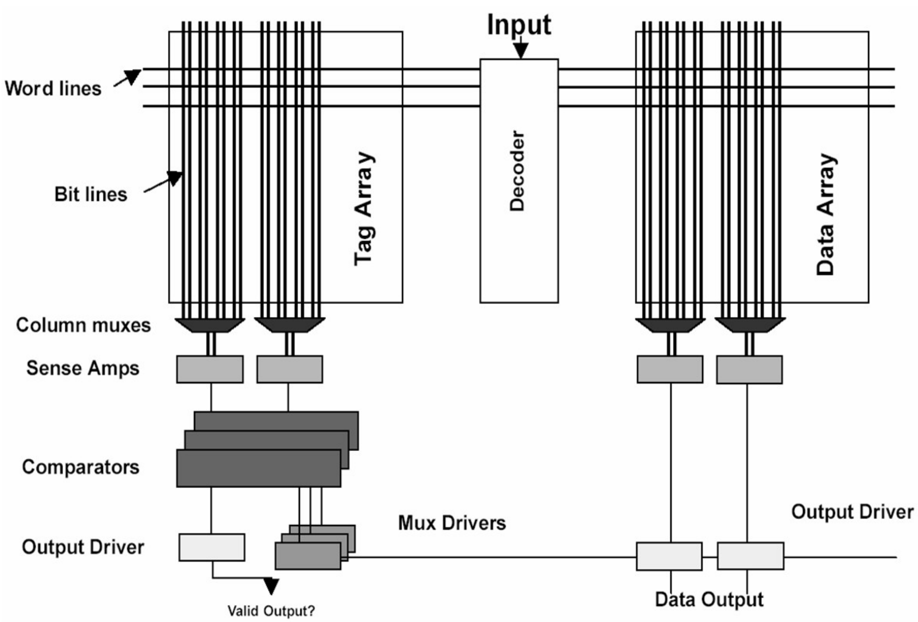


E.g.: ATMEL board with ARM7TDMI and ext. SRAM



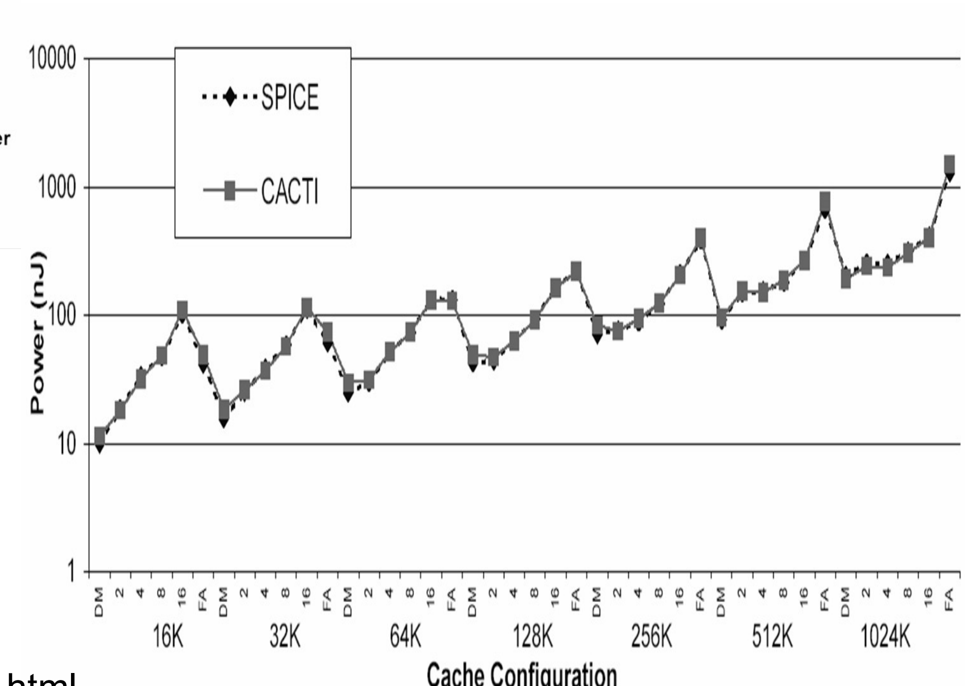


CACTI model



Cache model used

Comparison with SPICE



http://research.compaq.com/wrl/people/jouppi/CACTI.html

Summary

Performance analysis

- Simulation-based techniques
  - Trade-off between speed and accuracy (☞ end of chapter 2)
- Formal performance analysis
  - Thiele's real-time calculus (RTC)/Modular performance analysis (MPA)
    - Using bounds on the number of events in input streams
    - Using bounds on available processing capability
    - Derives bounds on the number of events in output streams
    - Derives bound on remaining processing capability, buffer sizes, …
    - Examples demonstrate design procedure based on RTC
- Evaluation of objective "energy/power consumption"