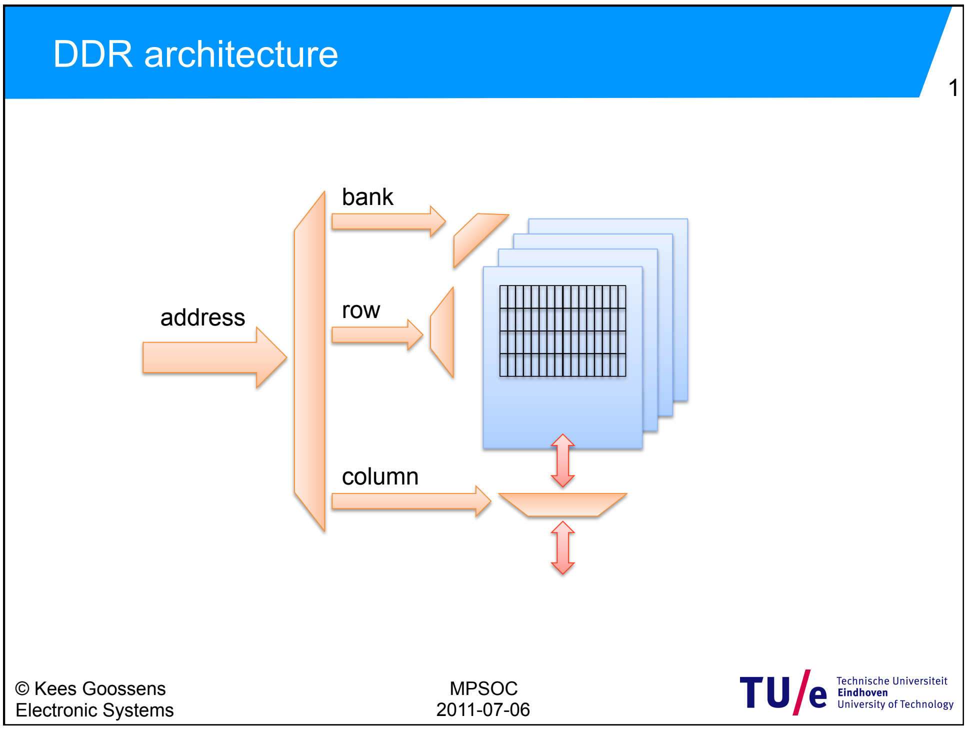
Efficient real-time SDRAM performance

Kees Goossens
with Benny Akesson, Sven Goossens, Karthik Chandrasekar, Manil Dev Gomony, Tim Kouters, and others

Kees Goossens <k.g.w.goossens@tue.nl>
Electronic Systems Group
Electrical Engineering Faculty

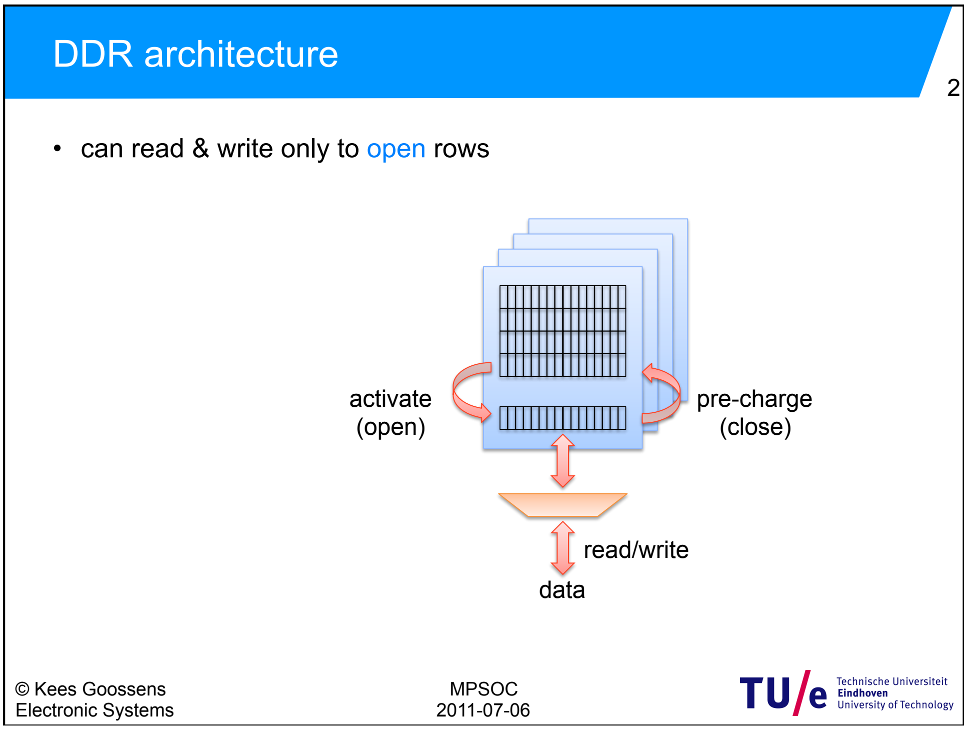
Technische Universiteit
Eindhoven
University of Technology

Where innovation starts



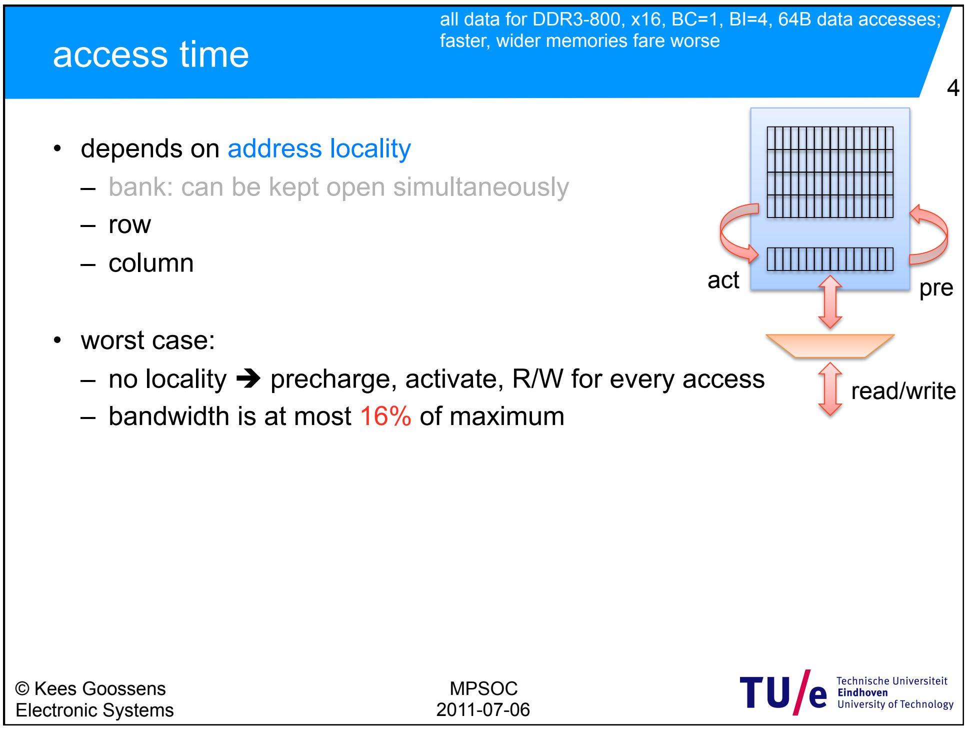
DDR architecture

1



DDR architecture

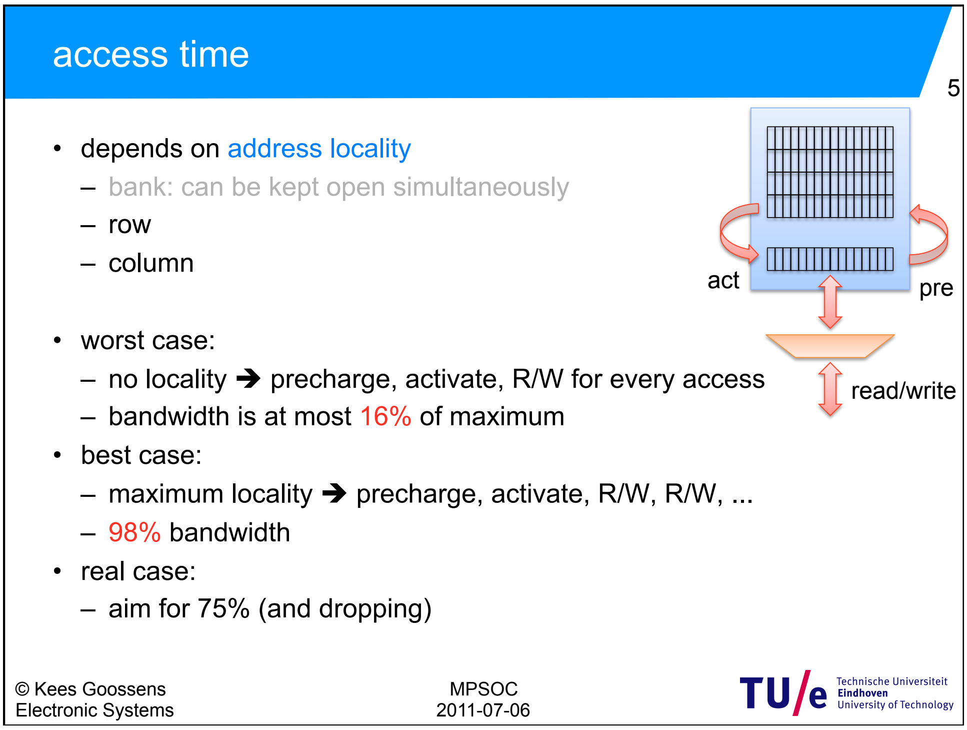
- can read & write only to open rows



access time

all data for DDR3-800, x16, BC=1, BI=4, 64B data accesses;
faster, wider memories fare worse

- depends on address locality
  - bank: can be kept open simultaneously
  - row
  - column
- worst case:
  - no locality ➔ precharge, activate, R/W for every access
  - bandwidth is at most 16% of maximum



access time

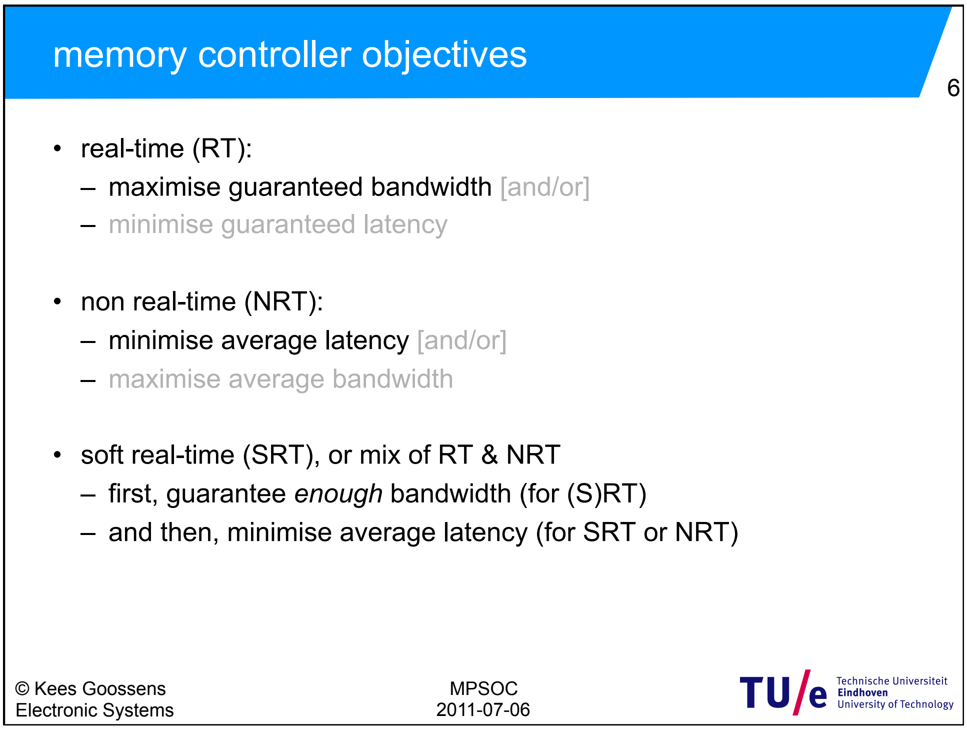
5

- depends on address locality
  - bank: can be kept open simultaneously
  - row
  - column
- worst case:
  - no locality ➔ precharge, activate, R/W for every access
  - bandwidth is at most 16% of maximum
- best case:
  - maximum locality ➔ precharge, activate, R/W, R/W, ...
  - 98% bandwidth
- real case:
  - aim for 75% (and dropping)

act
pre
read/write

© Kees Goossens
Electronic Systems

MPSOC
2011-07-06



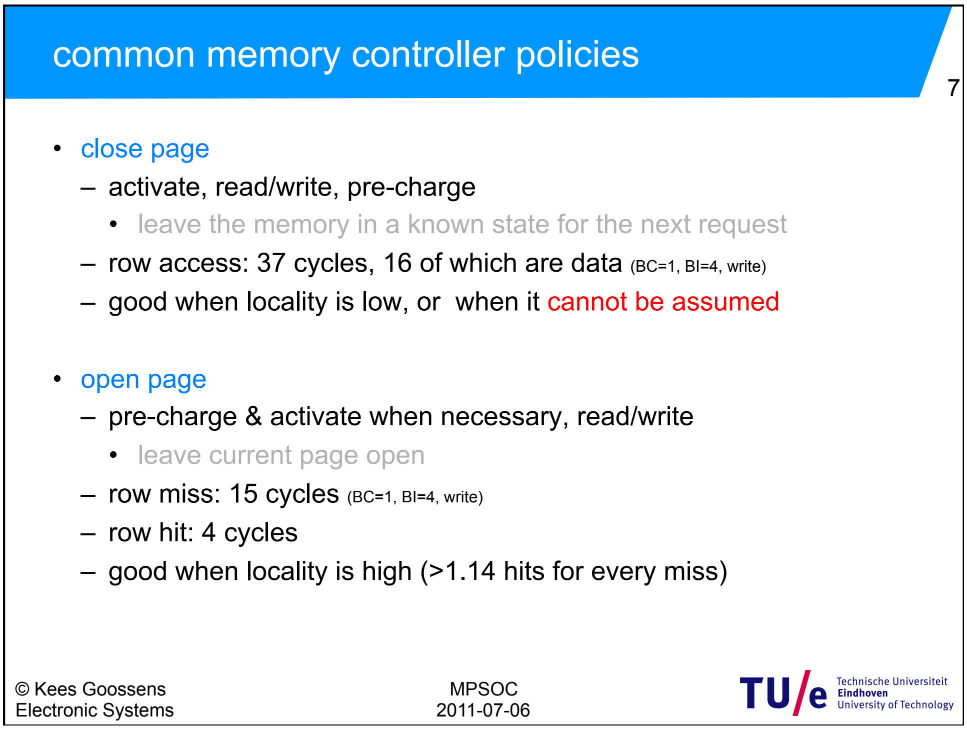
memory controller objectives

6

- real-time (RT):
  - maximise guaranteed bandwidth [and/or]
  - minimise guaranteed latency
- non real-time (NRT):
  - minimise average latency [and/or]
  - maximise average bandwidth
- soft real-time (SRT), or mix of RT & NRT
  - first, guarantee *enough* bandwidth (for (S)RT)
  - and then, minimise average latency (for SRT or NRT)

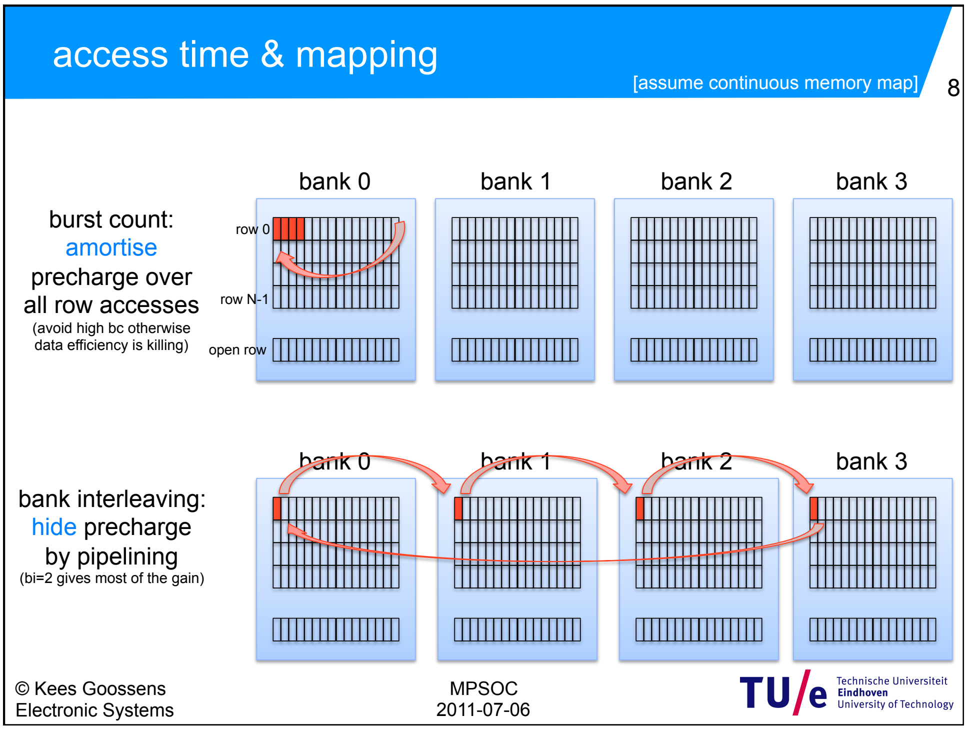
© Kees Goossens
Electronic Systems

MPSOC
2011-07-06



common memory controller policies

- close page
  - activate, read/write, pre-charge
    - leave the memory in a known state for the next request
  - row access: 37 cycles, 16 of which are data (BC=1, BI=4, write)
  - good when locality is low, or when it cannot be assumed
- open page
  - pre-charge & activate when necessary, read/write
    - leave current page open
  - row miss: 15 cycles (BC=1, BI=4, write)
  - row hit: 4 cycles
  - good when locality is high (>1.14 hits for every miss)

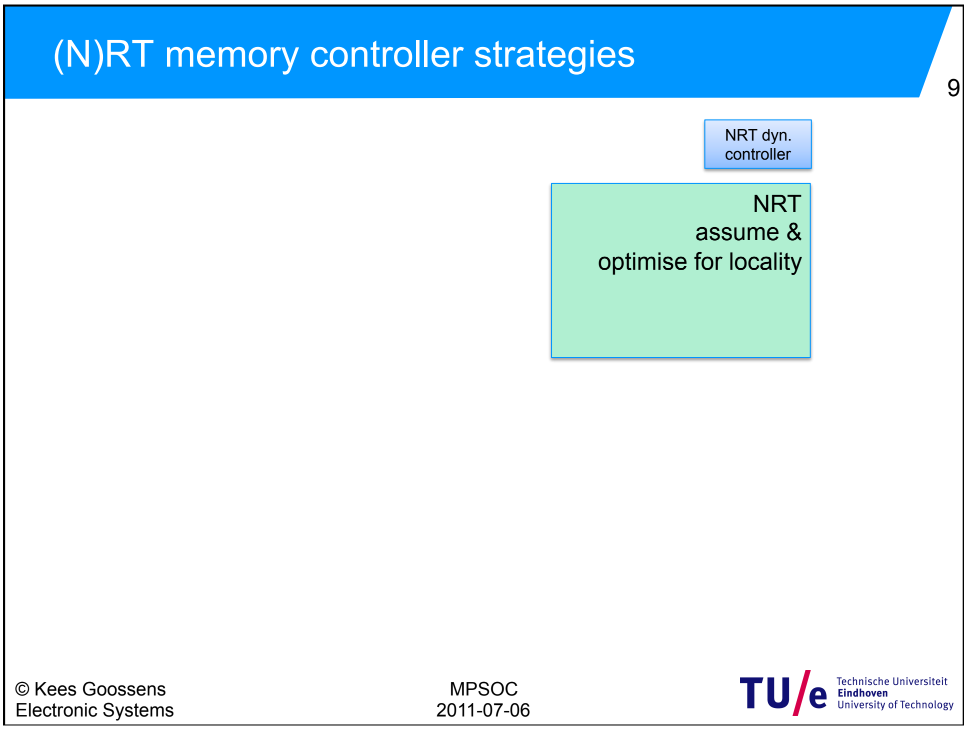


access time & mapping

[assume continuous memory map]

burst count: amortise precharge over all row accesses
(avoid high bc otherwise data efficiency is killing)

bank interleaving: hide precharge by pipelining
(bi=2 gives most of the gain)



(N)RT memory controller strategies

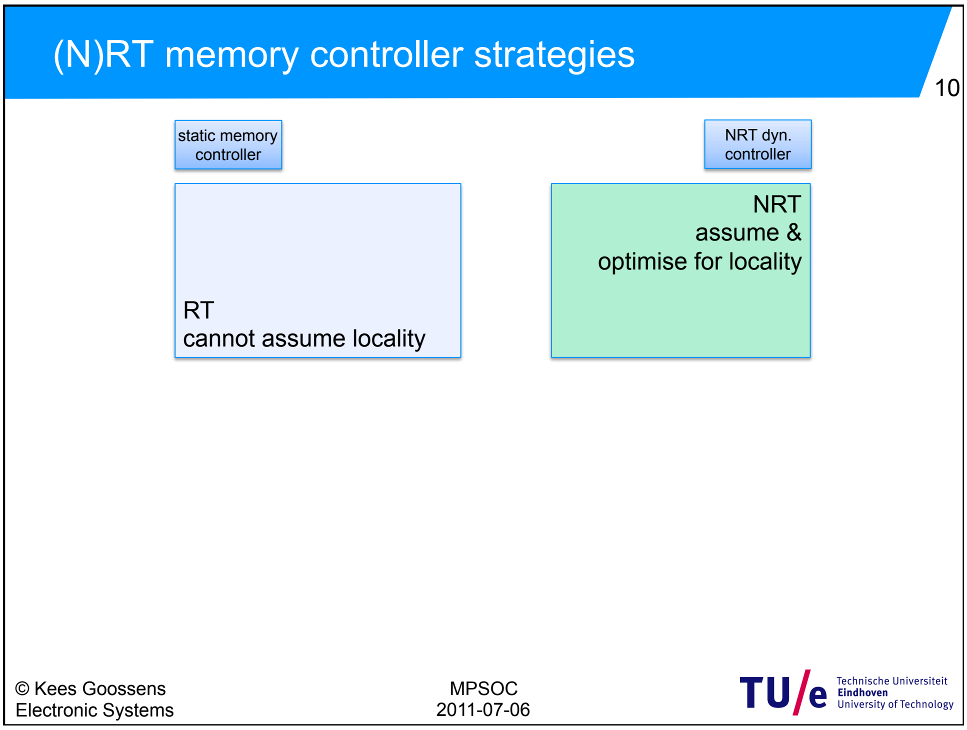
9

NRT dyn. controller

NRT
assume &
optimise for locality

© Kees Goossens
Electronic Systems

MPSOC
2011-07-06



(N)RT memory controller strategies

10

static memory controller

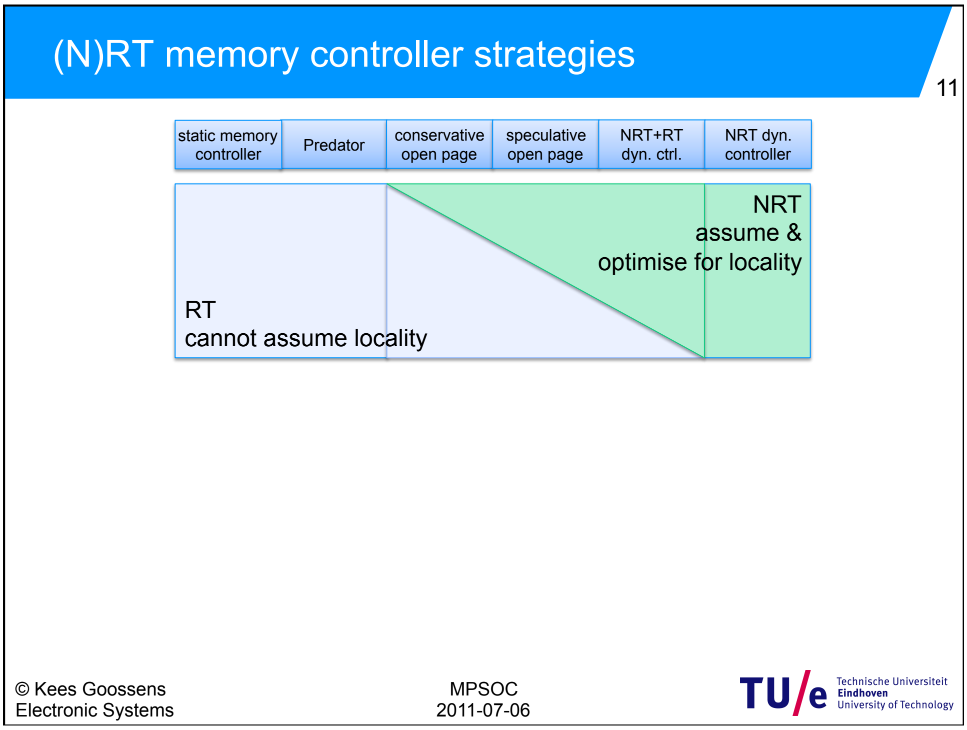
RT
cannot assume locality

NRT dyn. controller

NRT
assume &
optimise for locality

© Kees Goossens
Electronic Systems

MPSOC
2011-07-06

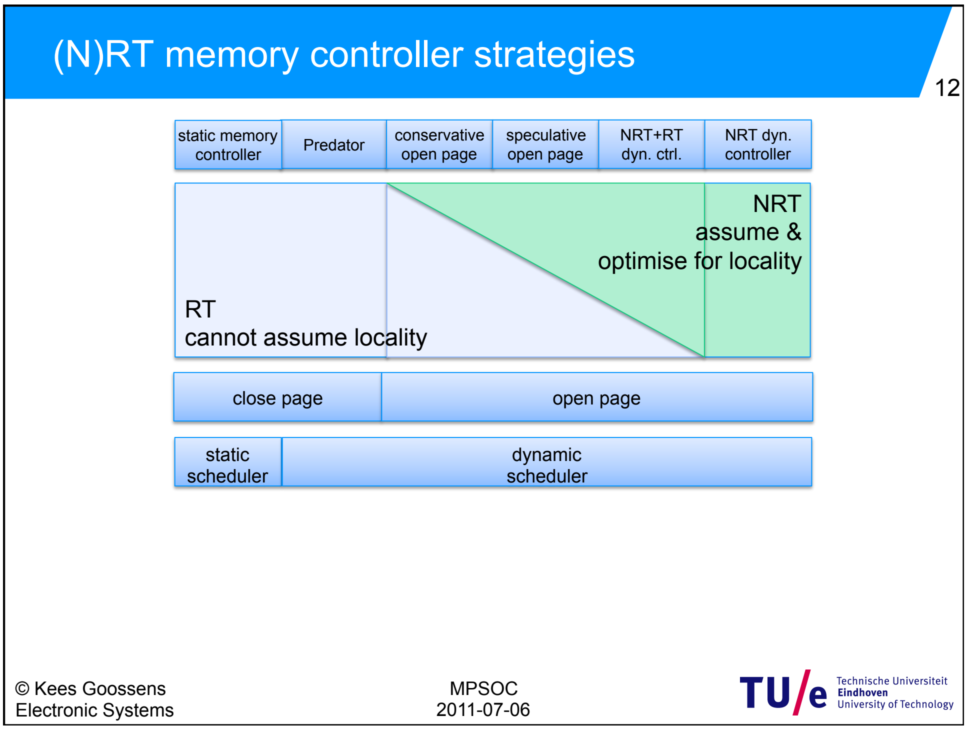


(N)RT memory controller strategies

static memory controller	Predator	conservative open page	speculative open page	NRT+RT dyn. ctrl.	NRT dyn. controller

NRT
assume &
optimise for locality

RT
cannot assume locality



(N)RT memory controller strategies

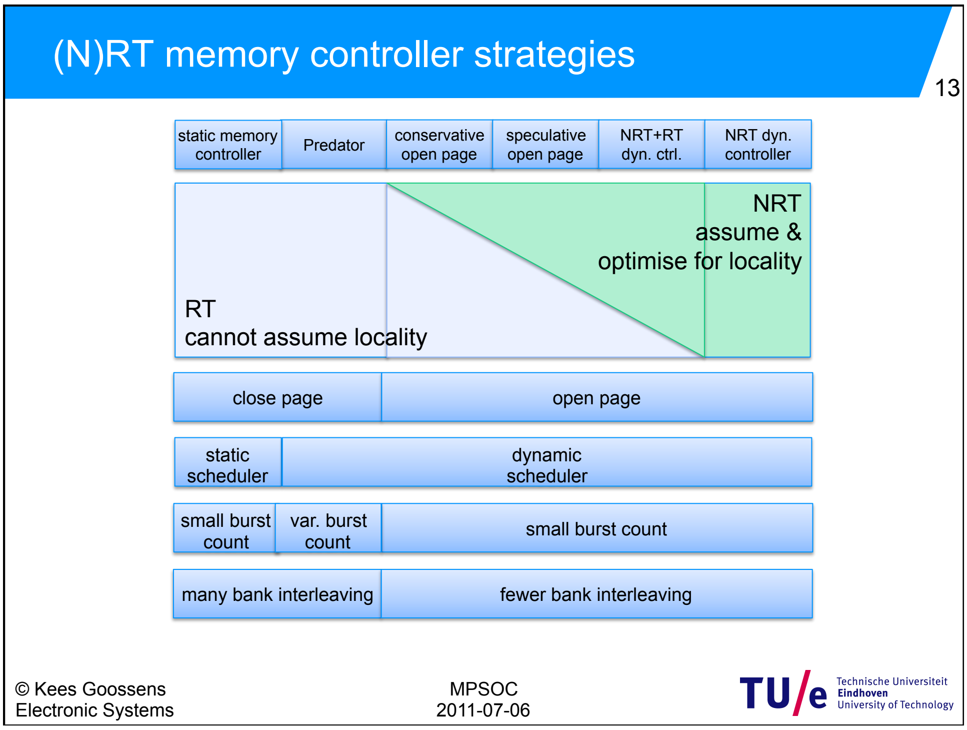
static memory controller	Predator	conservative open page	speculative open page	NRT+RT dyn. ctrl.	NRT dyn. controller

NRT
assume &
optimise for locality

RT
cannot assume locality

close page	open page

static scheduler	dynamic scheduler



(N)RT memory controller strategies

13

static memory controller | Predator | conservative open page | speculative open page | NRT+RT dyn. ctrl. | NRT dyn. controller

NRT assume & optimise for locality

RT cannot assume locality

close page | open page

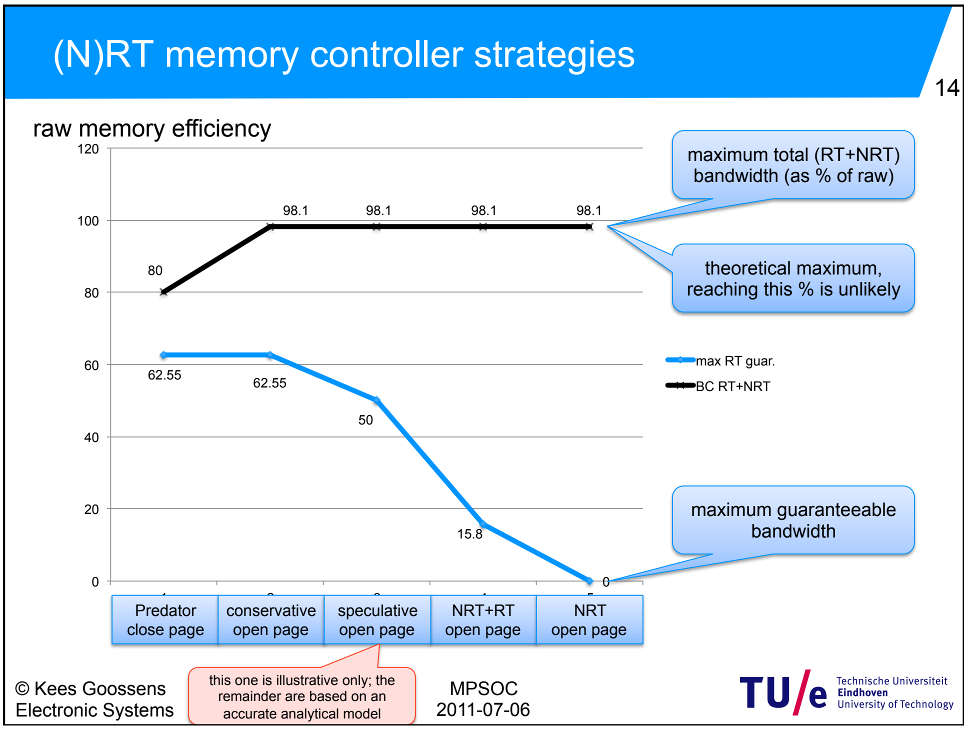
static scheduler | dynamic scheduler

small burst count | var. burst count | small burst count

many bank interleaving | fewer bank interleaving

© Kees Goossens
Electronic Systems

MPSOC
2011-07-06



(N)RT memory controller strategies

14

raw memory efficiency

maximum total (RT+NRT) bandwidth (as % of raw)

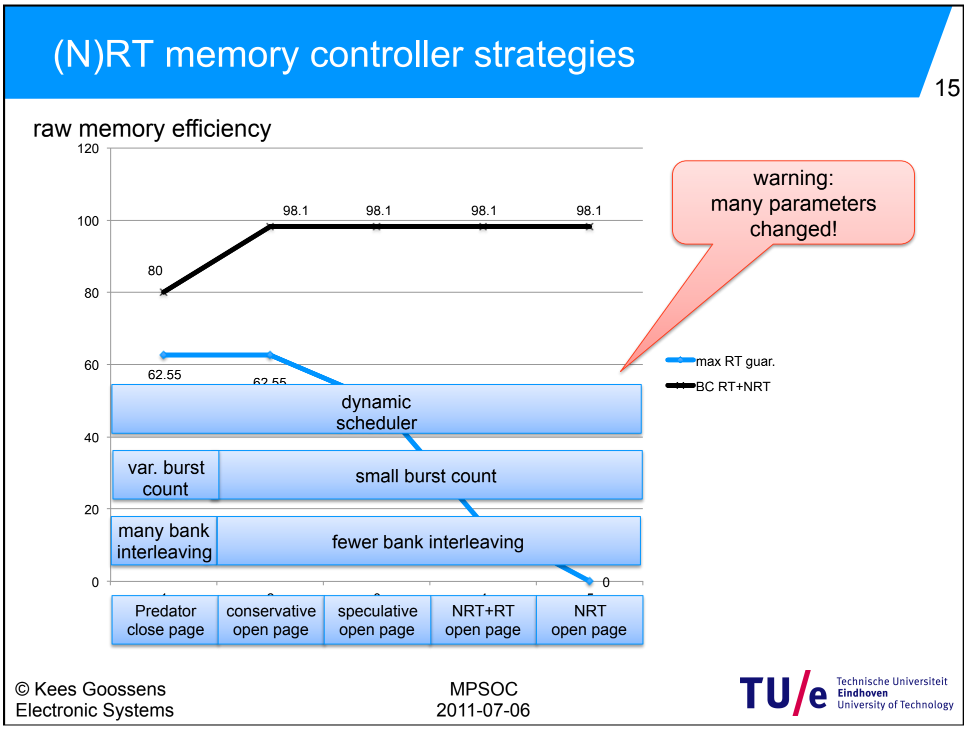
theoretical maximum, reaching this % is unlikely

maximum guaranteeable bandwidth

this one is illustrative only; the remainder are based on an accurate analytical model

© Kees Goossens
Electronic Systems

MPSOC
2011-07-06



(N)RT memory controller strategies

15

raw memory efficiency

warning:
many parameters
changed!

dynamic
scheduler

var. burst
count

small burst count

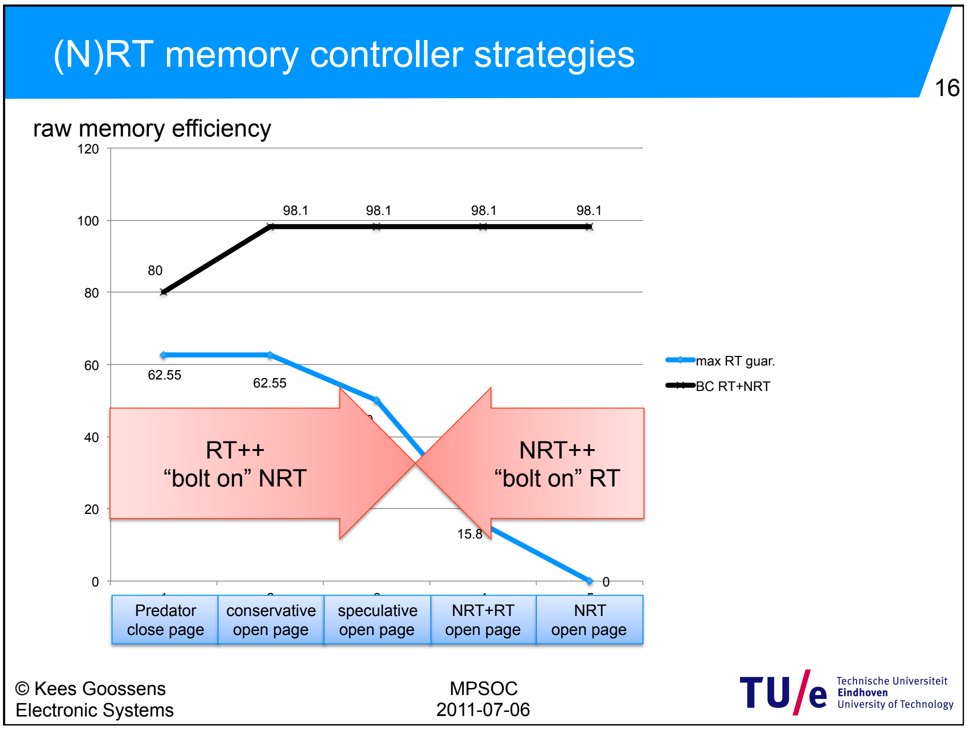
many bank
interleaving

fewer bank interleaving

Predator close page | conservative open page | speculative open page | NRT+RT open page | NRT open page

© Kees Goossens
Electronic Systems

MPSOC
2011-07-06



(N)RT memory controller strategies

16

raw memory efficiency

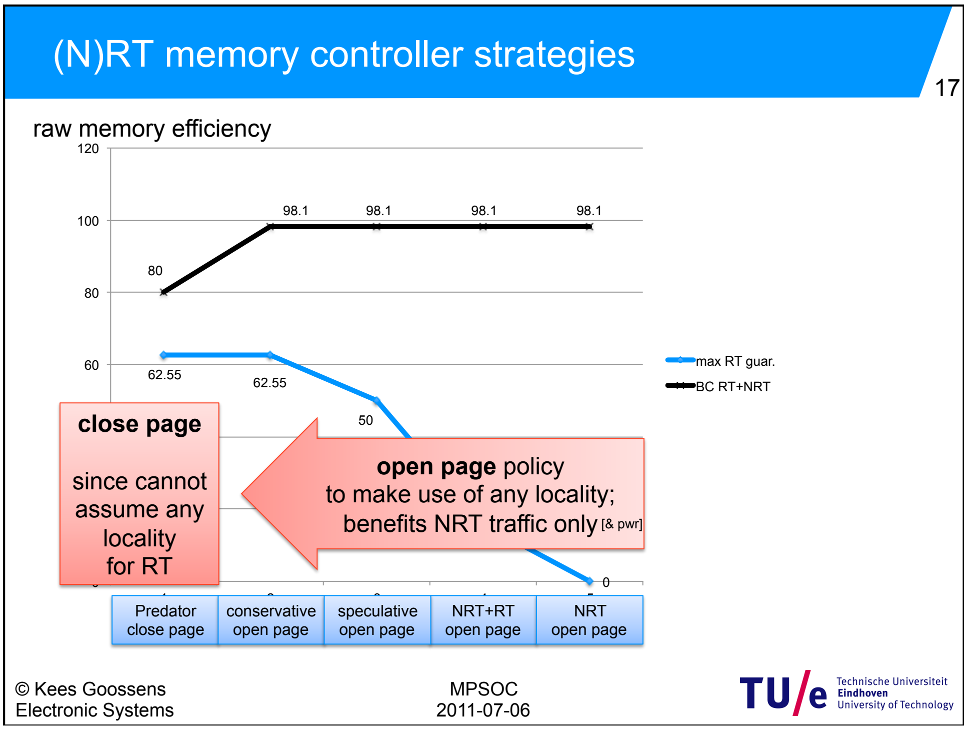
RT++
“bolt on” NRT

NRT++
“bolt on” RT

Predator close page | conservative open page | speculative open page | NRT+RT open page | NRT open page

© Kees Goossens
Electronic Systems

MPSOC
2011-07-06



(N)RT memory controller strategies

17

raw memory efficiency

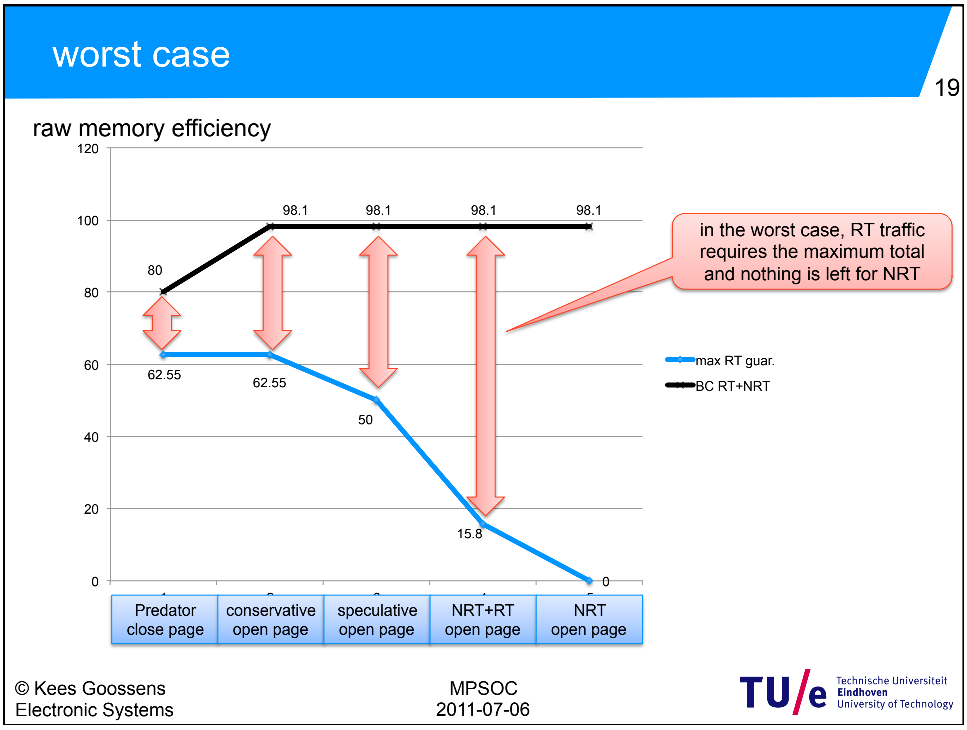
close page

since cannot assume any locality for RT

open page policy
to make use of any locality;
benefits NRT traffic only [& pwr]

© Kees Goossens
Electronic Systems

MPSOC
2011-07-06



worst case

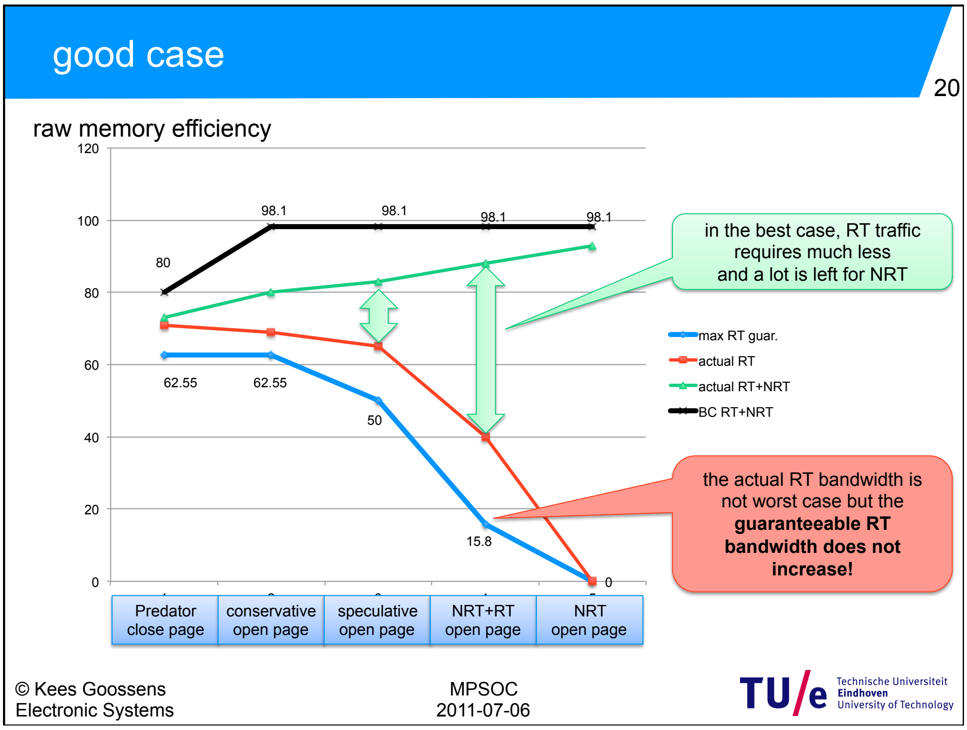
19

raw memory efficiency

in the worst case, RT traffic requires the maximum total and nothing is left for NRT

© Kees Goossens
Electronic Systems

MPSOC
2011-07-06



good case

20

raw memory efficiency

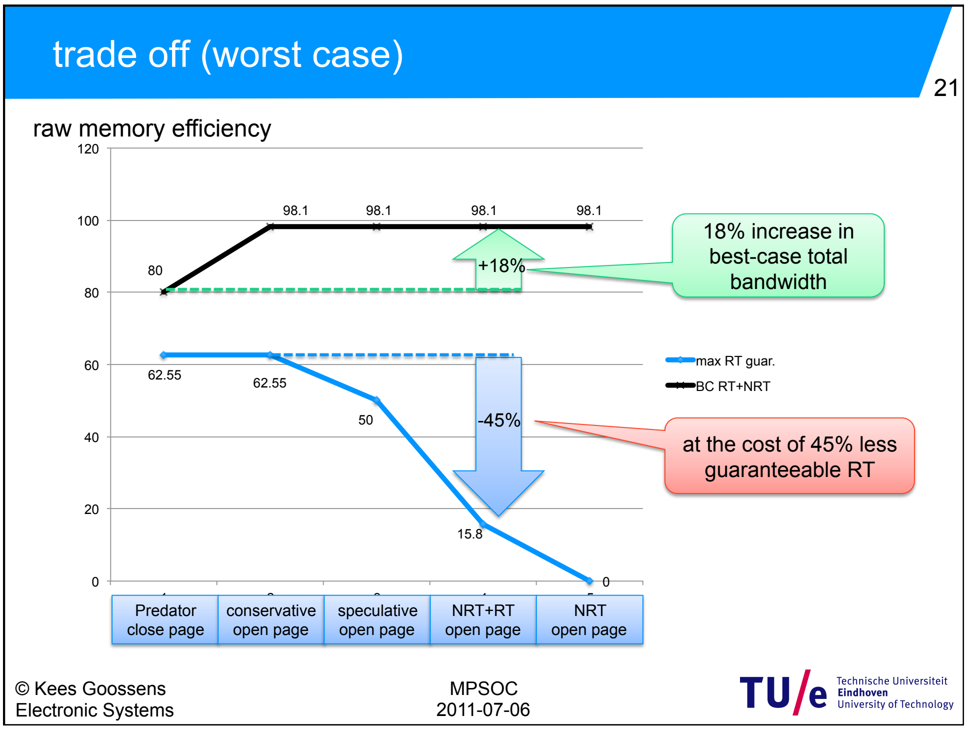
in the best case, RT traffic requires much less and a lot is left for NRT

the actual RT bandwidth is not worst case but the **guaranteeable RT bandwidth does not increase!**

© Kees Goossens
Electronic Systems

MPSOC
2011-07-06

TU/e Technische Universiteit Eindhoven University of Technology



trade off (worst case)

21

raw memory efficiency

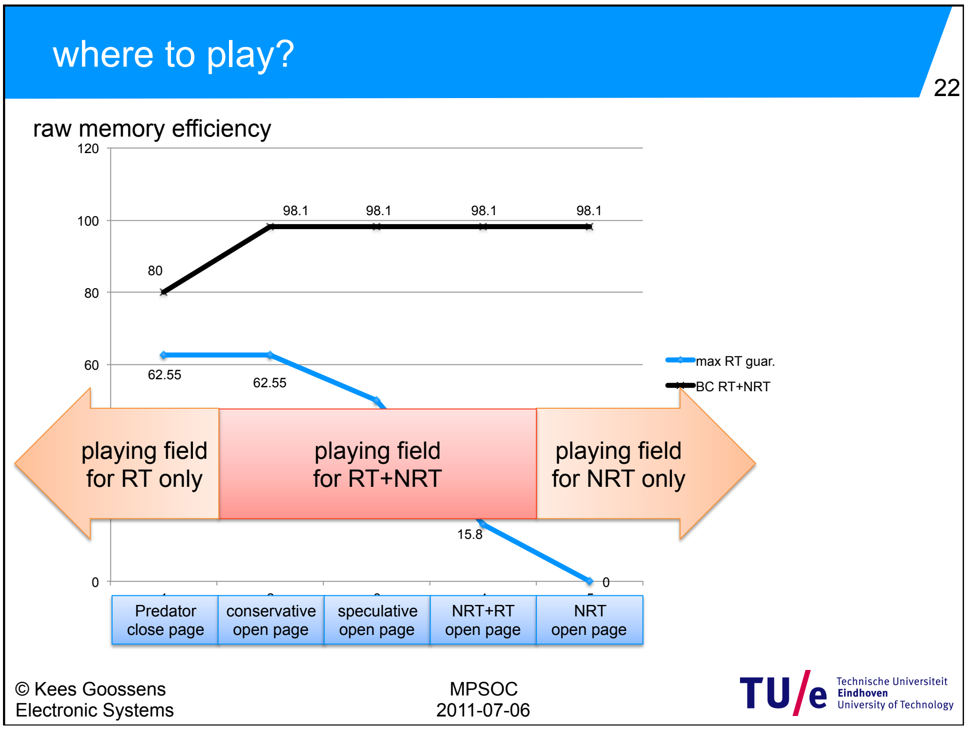
18% increase in best-case total bandwidth

at the cost of 45% less guaranteeable RT

© Kees Goossens
Electronic Systems

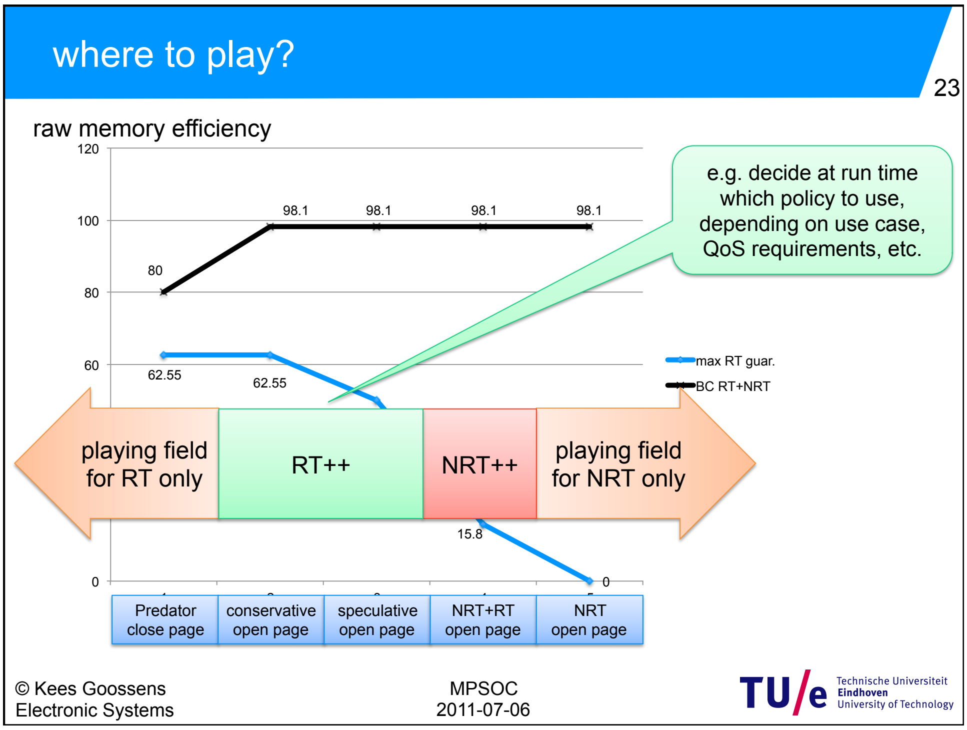
MPSOC
2011-07-06

TU/e Technische Universiteit Eindhoven University of Technology



where to play?

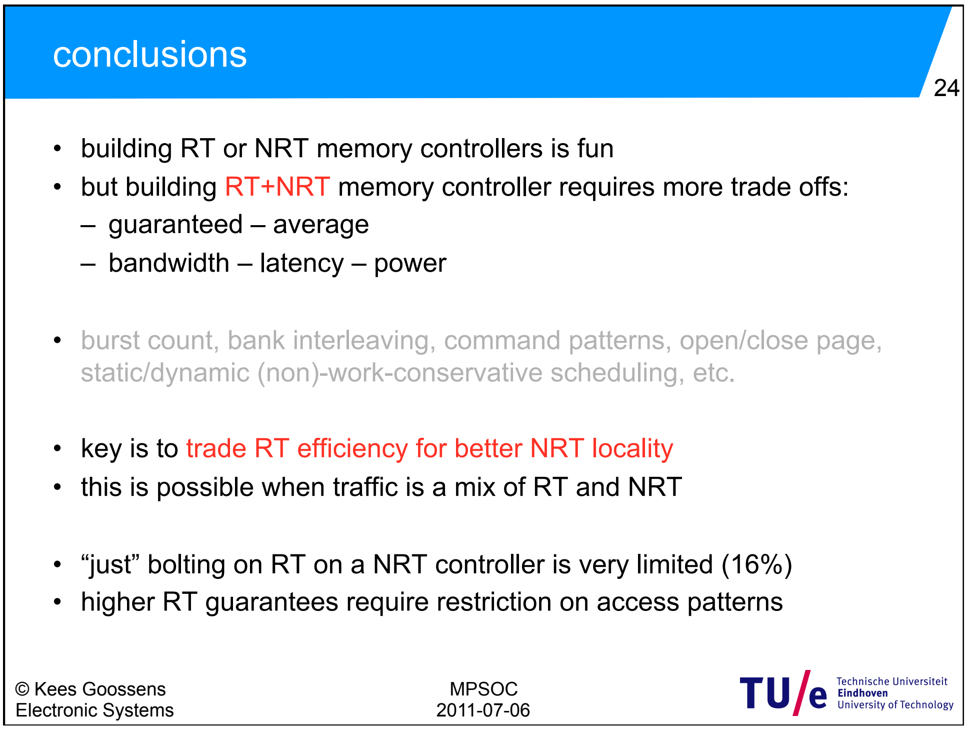
raw memory efficiency



where to play?

raw memory efficiency

e.g. decide at run time which policy to use, depending on use case, QoS requirements, etc.



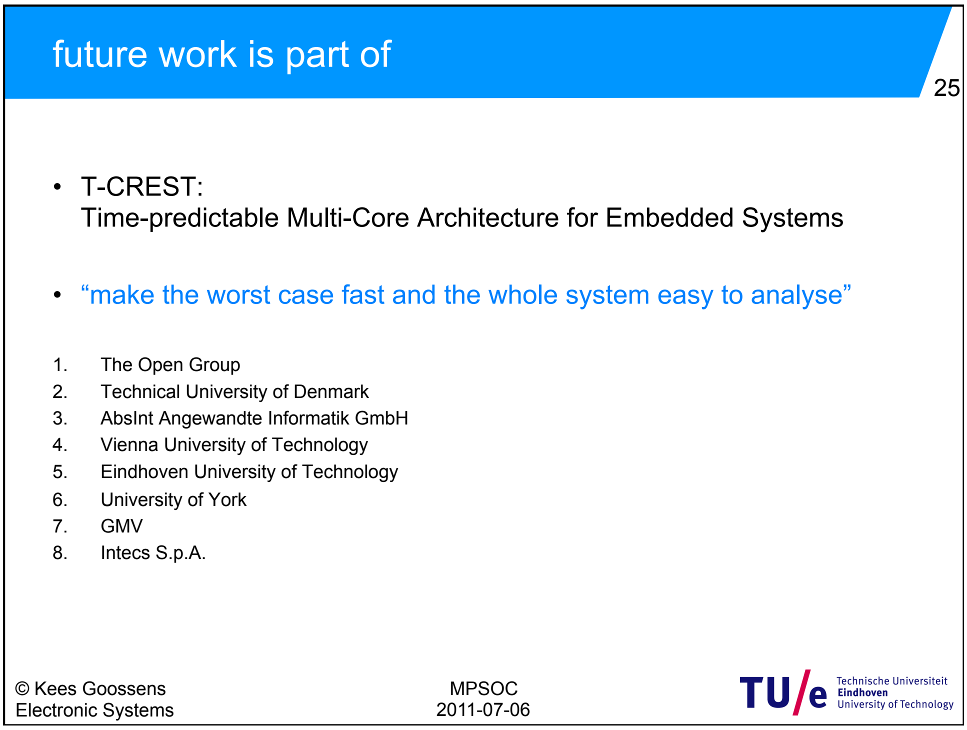
conclusions

- building RT or NRT memory controllers is fun
- but building RT+NRT memory controller requires more trade offs:
  - guaranteed – average
  - bandwidth – latency – power

- burst count, bank interleaving, command patterns, open/close page, static/dynamic (non)-work-conservative scheduling, etc.

- key is to trade RT efficiency for better NRT locality
- this is possible when traffic is a mix of RT and NRT

- “just” bolting on RT on a NRT controller is very limited (16%)
- higher RT guarantees require restriction on access patterns



future work is part of

- T-CREST:
  Time-predictable Multi-Core Architecture for Embedded Systems

- “make the worst case fast and the whole system easy to analyse”

1. The Open Group
2. Technical University of Denmark
3. AbsInt Angewandte Informatik GmbH
4. Vienna University of Technology
5. Eindhoven University of Technology
6. University of York
7. GMV
8. Intecs S.p.A.

Predator: RT DDR memory controllers; out Sep’11

- Memory Controllers for Real-Time Embedded Systems

Memory Controllers for Real-Time Embedded Systems

Predictable and Composable Real-Time Systems
Series: » Embedded Systems, Tentative volume 2

Akesson, Benny, **Goossens**, Kees

1st Edition., 2011, XVIII, 222 p. 95 illus.

Hardcover, ISBN 978-1-4419-8206-3
Due: September 28, 2011

approx. 99,95 €

- Verification of real-time requirements in systems-on-chip becomes more complex as more applications are integrated. Predictable and composable systems can manage the increasing complexity using formal verification and simulation. This book explains the concepts of predictability and composability and shows how to apply them to the design and analysis of a memory controller, which is a key component in any real-time system.
- This book is generally intended for readers interested in Systems-on-Chips with real-time applications. It is especially well-suited for readers looking to use SDRAM memories in systems with hard or firm real-time requirements. There is a strong focus on real-time concepts, such as predictability and composability, as well as a brief discussion about memory controller architectures for high-performance computing.
- Readers will learn step-by-step how to go from an unpredictable SDRAM memory, offering highly variable bandwidth and latency, to a predictable and composable shared memory, providing guaranteed bandwidth and latency to isolated applications. This journey covers concepts for making memories and arbiters behave in a predictable and composable manner, as well as architecture descriptions of hardware blocks that implement the concepts.

http://www.springer.com/engineering/circuits+%26+systems/book/978-1-4419-8206-3

Æthereal: RT interconnect

- On-chip-interconnect with aelite: Composable and Predictable Systems

Andreas Hansson
Kees Goossens
On-Chip Interconnect with aelite
Composable and Predictable Systems
Springer

On-Chip Interconnect with aelite

Composable and Predictable Systems
Series: » Embedded Systems

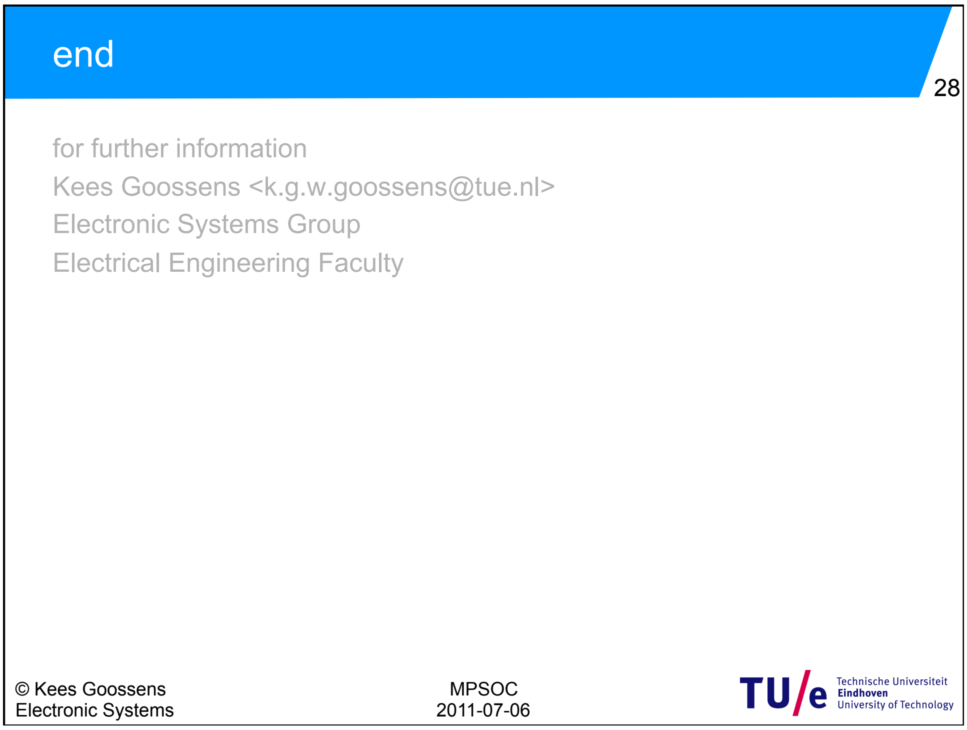
Hansson, Andreas, **Goossens**, Kees

1st Edition., 2010, X, 250 p. 200 illus., 100 in color., Hardcover
ISBN: 978-1-4419-6496-0

Due: November 29, 2010

- Embedded systems are comprised of components integrated on a single circuit, a System on Chip (SoC). One of the critical elements of such an SoC, and the focus of this work, is the on-chip interconnect that enables different components to communicate with each other. The book provides a comprehensive description and implementation methodology for the Philips/NXP Aethereal/aelite Network-on-Chip (NoC). The presentation offers a systems perspective, starting from the system requirements and deriving and describing the resulting hardware architectures, embedded software, and accompanying design flow. Readers get an in depth view of the interconnect requirements, not centered only on performance and scalability, but also the multi-faceted, application-driven requirements, in particular composability and predictability. The book shows how these qualitative requirements are implemented in a state-of-the-art on-chip interconnect, and presents the realistic, quantitative costs. •Uses real-world illustrations extensively, in the form of case studies and examples that communicate the power of the methods presented; •Uses one consistent, running example throughout the book. This example is introduced in the introductory chapter and supports the presentation throughout the work, with additional details given in each chapter; •Content has both breadth (architecture, resource allocation, hardware/software instantiation, formal verification) and depth (block-level architecture description, allocation algorithms, complete run-time APIs, detailed formal models, complete case studies mapped to FPGAs); •Includes numerous case studies, e.g. a JPEG decoder, set-top box and digital radio design.

http://www.springer.com/engineering/circuits+%26+systems/book/978-1-4419-6496-0



end

for further information

Kees Goossens <k.g.w.goossens@tue.nl>
Electronic Systems Group
Electrical Engineering Faculty