DESIGN, AUTOMATION & TEST IN EUROPE
9 – 13 March, 2015 · Grenoble · France
The European Event for Electronic System Design & Test

A Generic, Scalable and Globally Arbitrated Memory Tree for Shared DRAM Access in Real-Time Systems

Benny Akesson

Czech Technical University in Prague, Czech Republic

Manil Dev Gomony
Kees Goossens

Eindhoven University of Technology, The Netherlands

Jamie Garside
Neil Audsley

University of York, United Kingdom

european social fund in the czech republic
EUROPEAN UNION
MINISTRY OF EDUCATION, YOUTH AND SPORTS
OP Education for Competitiveness
INVESTMENTS IN EDUCATION DEVELOPMENT

Technische Universiteit Eindhoven University of Technology

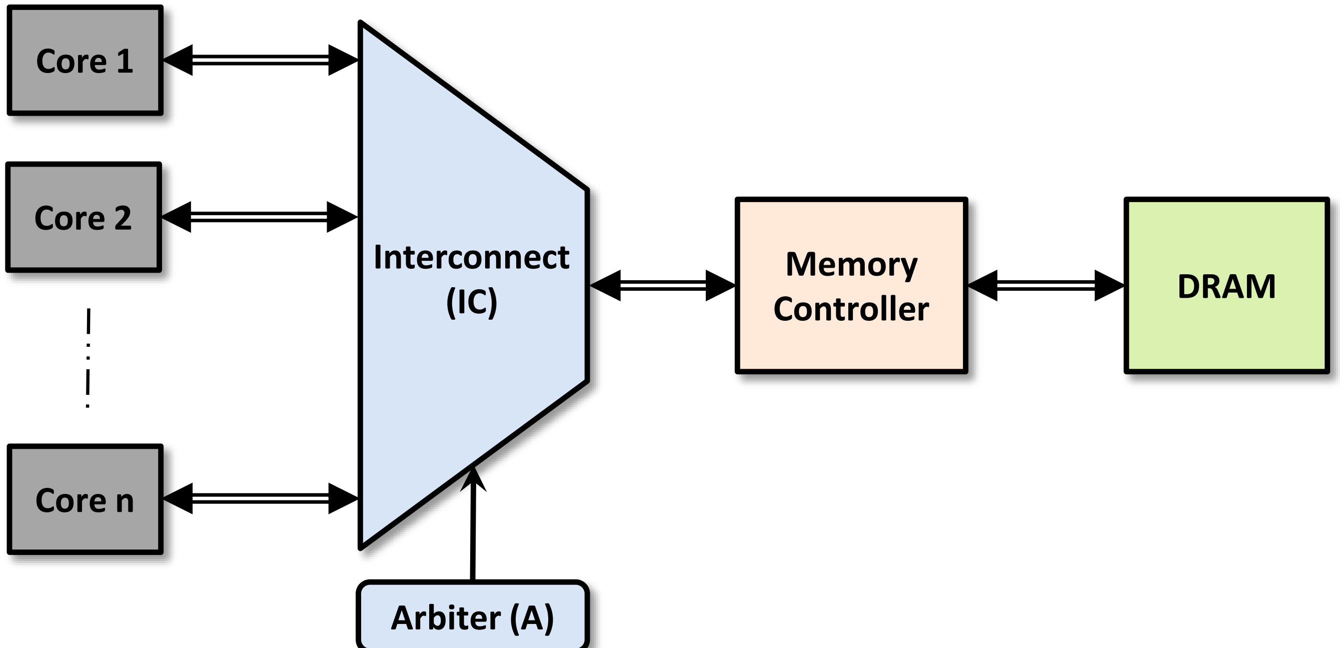
University of York

Outline

- **Introduction**
- **Problem Statement**
- **Proposed Solution**
- **Experimental Results**
- **Conclusion**

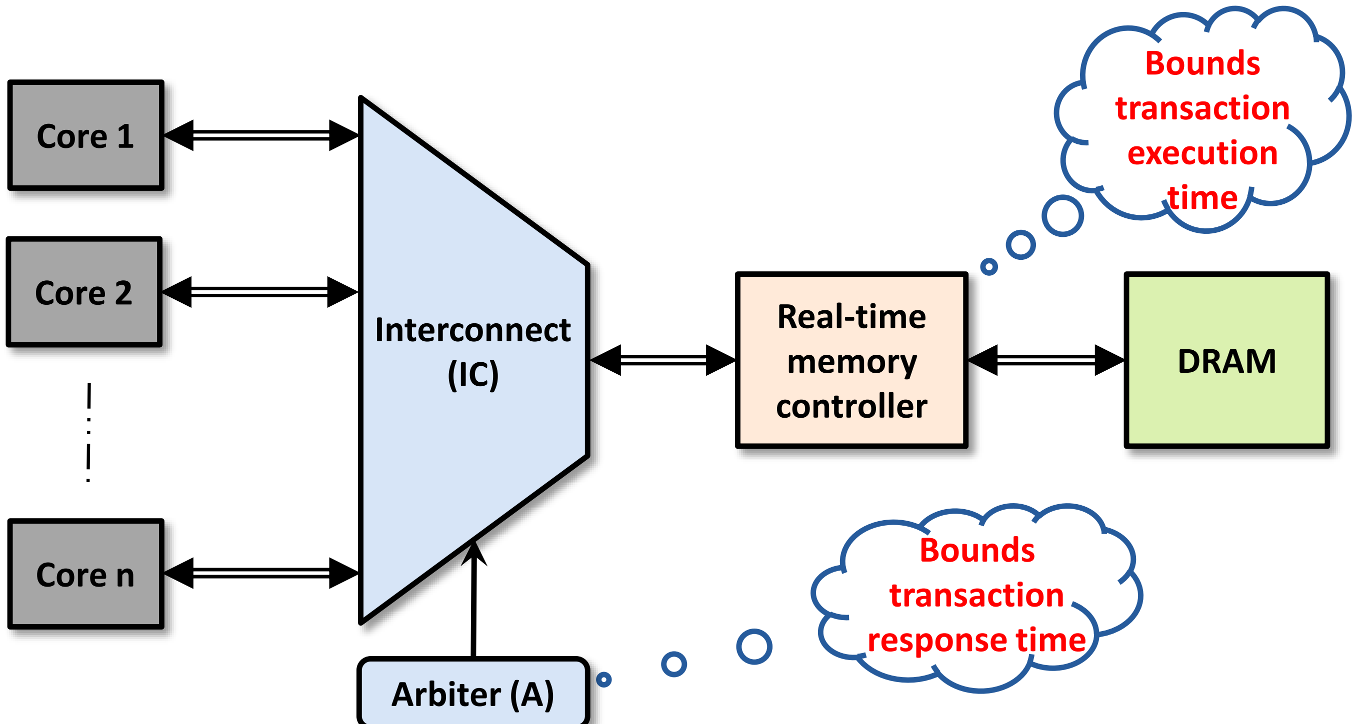
Multi-Core Platforms

- An increasing number of cores share a DRAM memory
- Memory Interconnect with an arbiter grants access to DRAM



Real-Time Systems

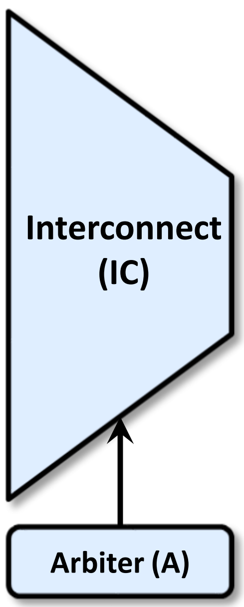
- **Time-predictable hardware:**
  - **Real-time memory controller**
  - **Predictable arbiter for resource sharing**



Memory Interconnect

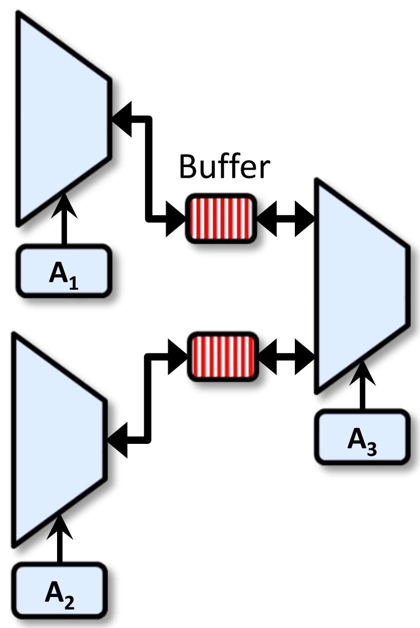
- **Three classes of existing memory interconnects:**

Centralized with local arbitration



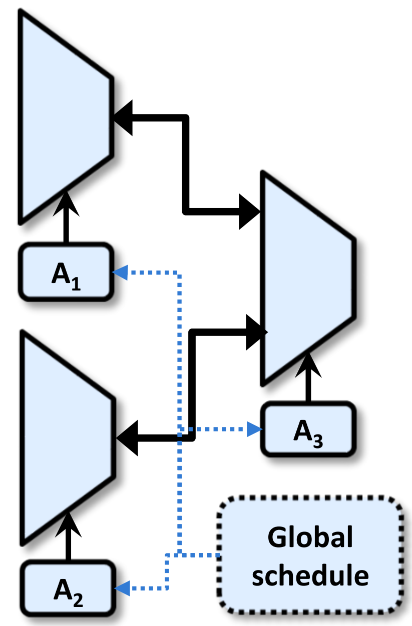
Not scalable

Distributed with local arbitration



Long latency, large area/power usage

Distributed with global arbitration



Only TDM is supported
Need to find global schedule

Contribution

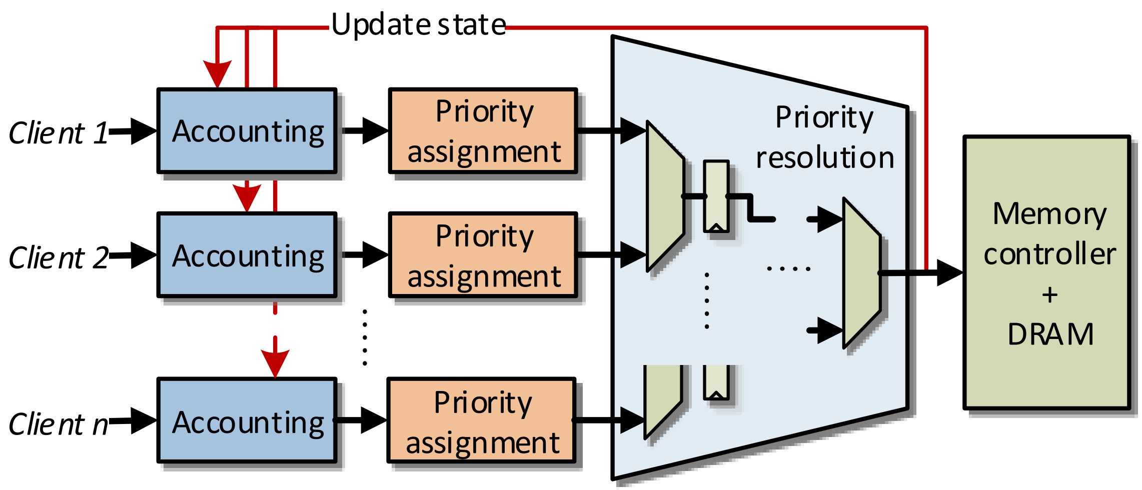
- **Number of memory clients is increasing, more than 64**
- **Client requirements may be diverse**

- **Existing memory interconnects:**
  - ***not scalable* – *cannot be synthesized at higher frequencies***
  - ***decoupled arbitration stages* – *long latencies and larger area/power usage***
  - ***only support TDM* – *cannot support diverse requirements***

- **We propose a *globally arbitrated distributed memory interconnect supporting multiple arbitration policies***
  - **TDM, FBSP, and CCSP in (non)-work-conserving mode**

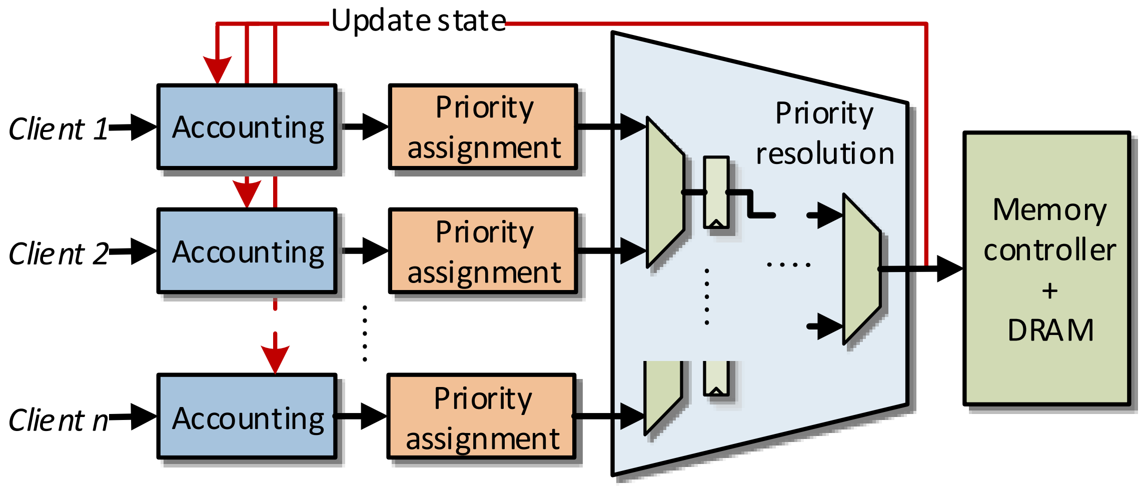
Generic, Scalable Memory Tree (GSMT)

- **Four main components:**
  1. ***Accounting*** **– keeps track of the eligibility status of a client**
  2. ***Priority Assignment*** **– assigns a unique priority to a client**
  3. ***Priority Resolution*** **– grants access to highest priority client**
  4. ***Update State*** **– Informs accounting about scheduled client**



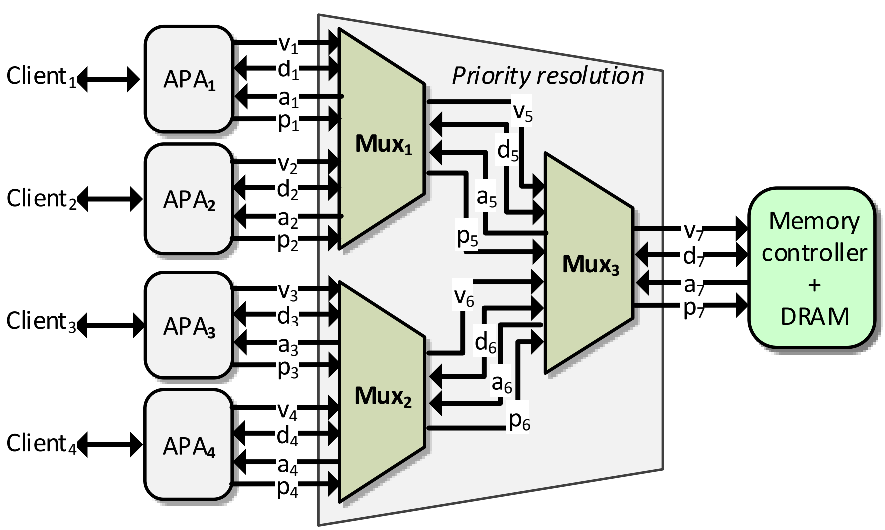
GSMT Properties

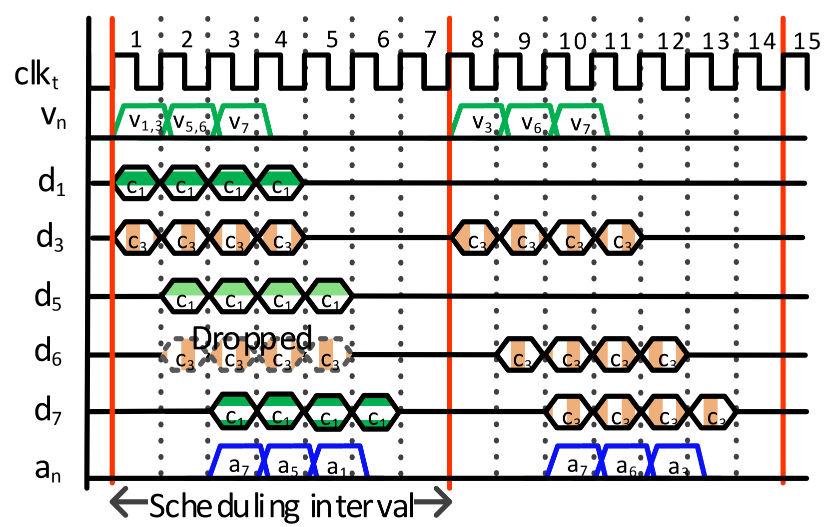
- **Distributed implementation – *scalable***
  - **Dedicated accounting and priority assignment (APA) for each client**
- **Global arbitration – *low latency, area and power***
  - **Global *scheduling interval* for all clients**
- **Generic – *configurable to support diverse requirements***
  - **Supports three different arbitration policies**



GSMT Interface and Operation

- **Accounting and Priority assignment (APA) schedules requests and assigns a unique priority on the priority lines**
- **Request with lowest priority are dropped at the *Mux* stages and are rescheduled during the next scheduling interval**
- **Acknowledgement is sent to scheduled client**





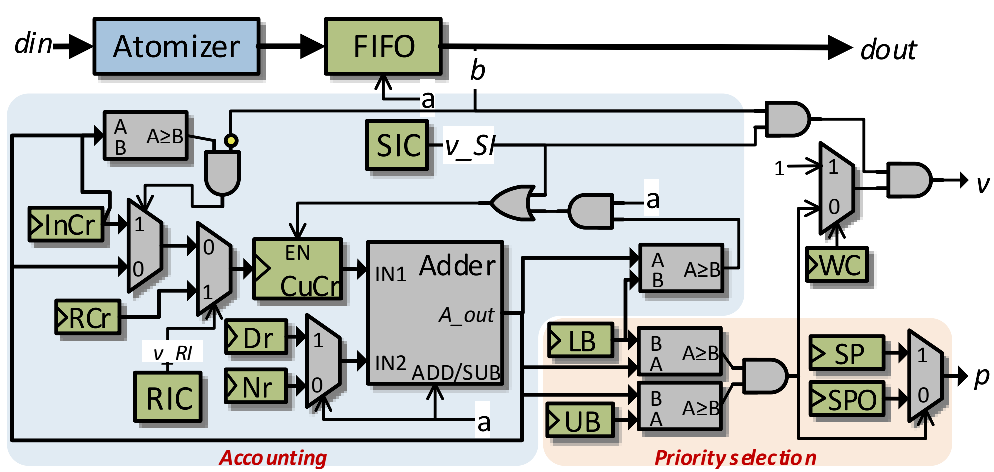
Generic Configurable APA Architecture

Input signals: Acknowledgement (a), Backlogged (b)
Output signal: Priority (p)

procedure ACCOUNTING(a, b)
  if *v_SI* **then**
    if ((!b) & (*A_out* ≥ InCr)) **then**
      CuCr ← InCr
    else if *v_RI* **then**
      CuCr ← RCr
    else
      CuCr ← CuCr + Nr
    end if
  else if ((a) & (*A out* ≥ LB)) **then**
    CuCr ← CuCr - Dr
  end if
end procedure

procedure PRIORITY ASSIGNMENT(*A_out*)
  if LB ≤ *A_out* ≤ UB **then**
    p ← SP
  else
    p ← SPO
  end if
  return p
end procedure

Register	TDM	FBSP	CCSP
InCr	f	f.ρ	σ.dr
CuCr	0	f.ρ	σ.dr
RCr	0	f.ρ	Not used
Nr	1	0	nr
Dr	0	1	dr
SP	Unique for each client	Unique for each client	Unique for each client
SPO	SP + Offset	SP + Offset	SP + Offset
UB	End position in TDM frame	> f.ρ	
LB	Start position in TDM frame	1	nr-dr
SIC	SI	SI	SI
RIC	f.SI	f.SI	Not used



Experimental Setup

- **RTL-level implementation of GSMT and centralized implementations of TDM and CCSP**

- **Cadence Encounter RTL compiler**
  - **40*nm* nominal V_t CMOS standard cell library**
  - **Worst-case process corner**

Functional Verification

- **Synthetic traffic was generated from 16 clients**
- **Scheduling decisions of the GSMT were compared to reference implementations of TDM, FBSP, and CCSP arbiters**
- **All *scheduling decisions were identical* suggesting *correct implementation***
- **As a result, *existing timing analysis of arbiters apply***

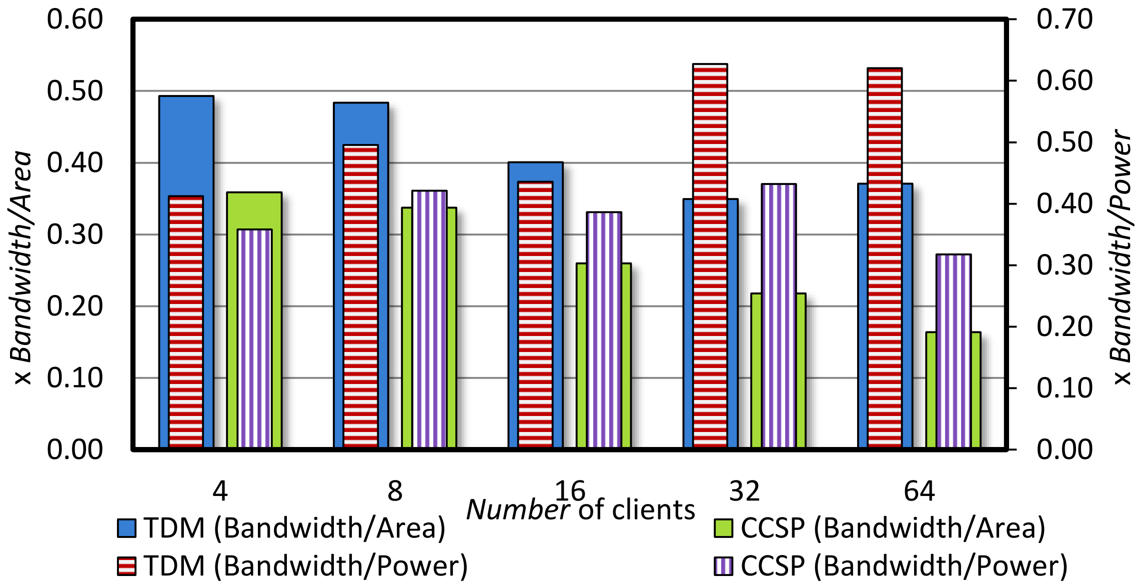
Experimental Results

# Clients	Area (mm^2)			Power (mW)			f_{max} (MHz)		
	TDM	CCSP	GSMT	TDM	CCSP	GSMT	TDM	CCSP	GSMT
4	0.016	0.020	0.017	5.194	5.351	4.55	588	526	1250
8	0.029	0.036	0.035	7.883	8.073	9.77	500	435	1250
16	0.061	0.077	0.070	16.126	14.935	20.20	435	357	1250
32	0.107	0.172	0.141	17.455	25.361	41.07	333	333	1250
64	0.203	0.417	0.282	35.603	63.179	82.81	333	303	1250

- **Area, power for all designs increase with number of clients due to additional logic and wiring**
- **f_{max} of CCSP and TDM scales down with increasing number of clients**
- **The critical path of GSMT is in APA and is independent for each client, and hence scales well with the number of clients**
- **GSMT consumes more power compared to centralized implementations**

Performance Comparison

- **We define two cost-efficiency metrics: *bandwidth/area* and *bandwidth/power***
- ***GSMT has over 51% and 37% gain in terms of area and power***



- ***GSMT is suitable when high bandwidth is needed and client requirements are diverse***

Conclusions

- **The number of memory clients in multi-cores is increasing**
- **Existing interconnects are either not scalable for a large number of clients or do not support diverse requirements**

- **We presented a generic, scalable and globally arbitrated memory interconnect (GSMT)**
- **Compared to centralized implementations**
  - **Runs at four times higher frequency**
  - **Provides over 51% and 37% gain in bw/area and bw/power**
  - **Supports three different arbitration mechanisms**

Questions?

Thank you for your attention!

kessoben@fel.cvut.cz