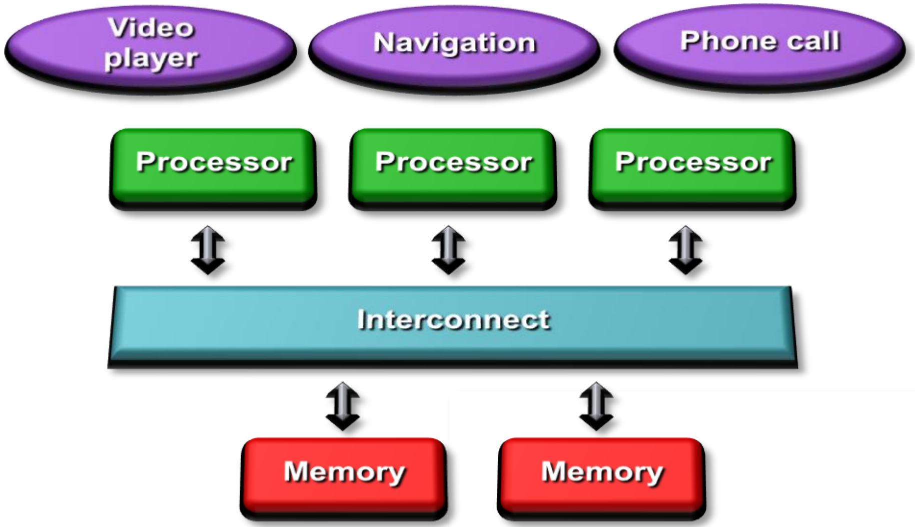
Predictable And Composable System -on- Chip Memory Controllers

Benny Åkesson

Trends in embedded systems

⟶ Embedded systems get **increasingly complex**

- **More hardware components** in a system
  - Many different processors and memories
- **More applications** that run **simultaneously**
  - Video player, Internet browser, phone application, navigation, etc.
- Some applications have **real-time requirements**
  - Must finish some processing before a deadline



Benny Åkesson
k.b.akesson@tue.nl

Verification problem

→ Resources are shared to reduce cost

- Results in **interference** between applications
- Makes timing behavior of sharing applications **inter-dependent**

→ Verifying real-time requirements is problematic

- **All combinations** of running applications must be verified
- Typically done by slow simulation → **poor coverage**

→ **However, this problem is not without possible solutions…**

Predictability

→ **Predictability** has been proposed to address the problem

- A system is predictable if max. execution time for applications are known
- Enables **formal verification** of real-time requirements
- Mathematical proof instead of slow simulations
- Proof **covers all combinations** of running applications

→ Predictability **requires models** of applications and hardware

- Do not always exist
- Predictability hence only **partially** solves the problem

Composability

→ **Complementary** verification approach

- A system is composable if applications do not interfere with each other

→ **Isolation** between applications

- No need to verify all possible combinations of applications
- Each application is verified by simulation in isolation
- Works with **any application**

Benny Åkesson
k.b.akesson@tue.nl

Problem statement

→ Focus on SDRAM memory controllers

- **Essential** components in modern systems
- Used to store large amounts of data
- Bandwidth is a **scarce resource** that must be used efficiently

→ Existing SDRAM controllers are

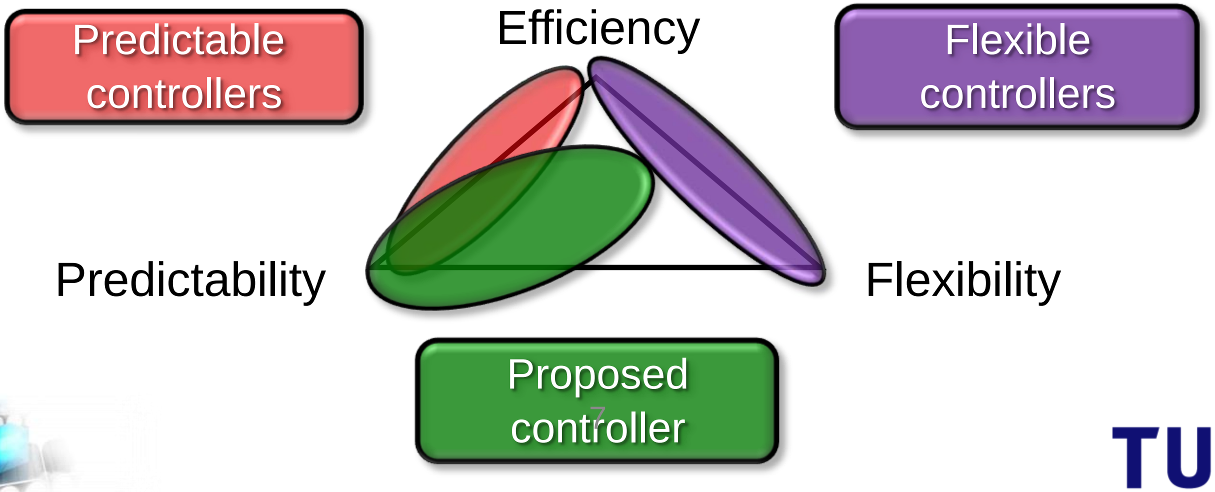
- Either **unpredictable** or **inflexible**
- **Not composable**

The problem is:

To design a predictable and composable SDRAM memory controller, thereby addressing the verification problem

Predictable SDRAM controller back-end

- We propose a predictable SDRAM back-end
  - Combines elements of predictable and flexible memory controllers
  - Does not have to know exact memory traffic at design time
  - **Increases flexibility**, possibly with a reduction in efficiency
- A **single application** can read and write with predictable timings



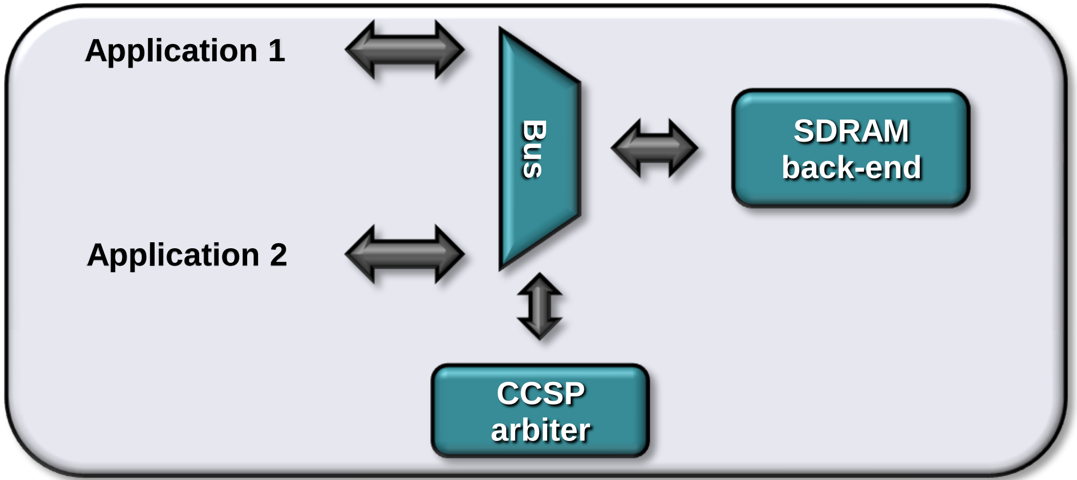
Credit-Controlled Static-Priority arbiter

- More than one real-time application need to use the memory
  - Requires an **arbiter** to schedule requests from the applications
  - Applications have different behavior and requirements
- We propose a Credit-Controlled Static-Priority (CCSP) arbiter
  - It is **predictable**
  - Uses **priorities** to separate urgent applications from the rest
  - Uses **budgets** to stop greedy applications from monopolizing the memory
- The hardware implementation
  - Is **small** and **fast** enough for most modern memories
  - Reserves bandwidth for applications **efficiently**

General design & Configuration

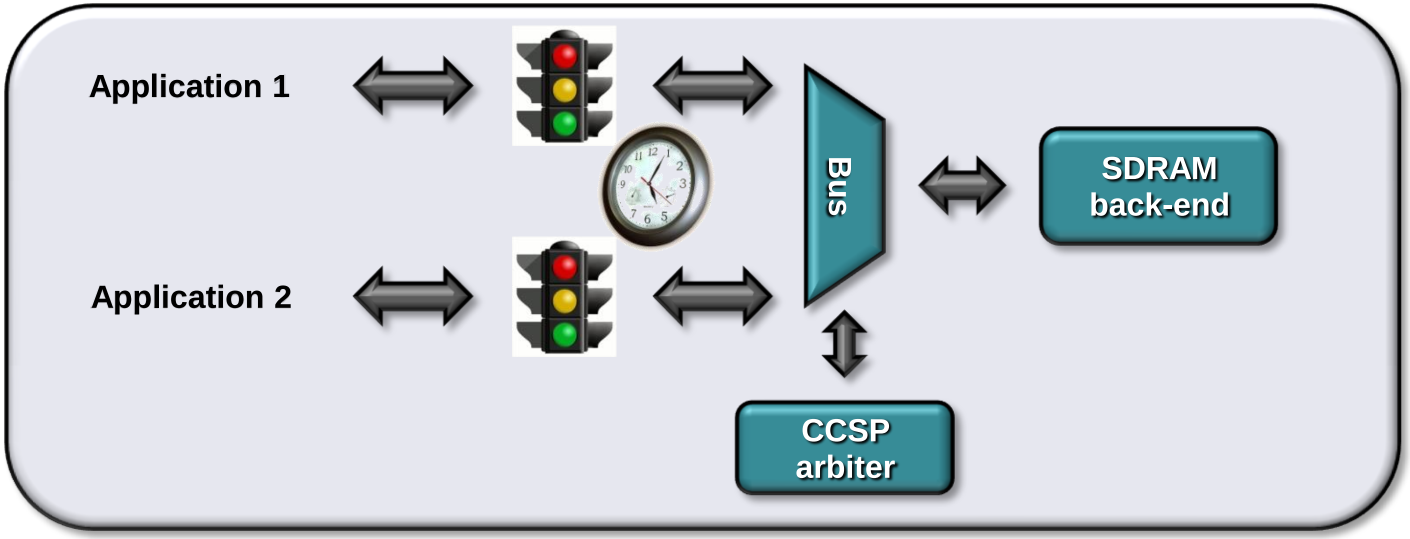
- The proposed predictable memory controller is **general**
  - Arbiter type can be changed to fit with application requirements
  - Supports different memory types by changing the back-end
  - Timings are guaranteed **for all combinations** of memory and arbiter types

- We present an **automatic configuration** tool
  - Computes arbiter and memory configuration that satisfies requirements



Composable memory controller

- Novel approach to composable resource sharing
  - Always delay a requests to **emulate worst-case interference**
  - Makes timing behavior of applications independent
- Extends composability to work with a wider range of systems
  - Supports **any combination** of predictable arbiter and predictable memory



Conclusions

⟶ **Verification problem** in embedded systems

- Increasing number of applications
- Not possible to verify real-time requirements for all combinations

⟶ **Predictable** and **composable** systems are promising solutions

- No good predictable and composable memory controllers

Contributions

- A **predictable SDRAM back-end**
- A **predictable CCSP arbiter** that is suitable for sharing memories
- A **general design** where the arbiter and memory can be changed
- A tool that **automatically** computes arbiter and memory configurations
- An approach that makes any predictable arbiter and memory **composable**

Predictable And Composable System -on- Chip Memory Controllers

Benny **Åkesson**