Secure Partitioning and Multi-Core Processors

Thomas PAREAUD, Alain ROSSIGNOL
EADS Astrium

Peter MENDHAM, Stuart FOWELL
SciSys UK Ltd

“Securely Partitioning Spacecraft Computing Resources” studies

- Study lead by SciSys
  - SciSys (Prime contractor)
  - SYSGO (Technology provider)
  - Astrium (Operational Scenario Analysis)
  - University of York (External security experts)
- Study lead by Astrium
  - Astrium (Prime contractor)
  - Universitat Politècnica de València (Technology provider)
  - Teletel (Validation and demonstration activities)
  - Laas/Cnrs + Airbus (External security experts)
- Duration: < 2 years

scisys

ASTRIUM
AN EADS COMPANY

Study Objectives

- Objectives
  - Define operational scenarios w.r.t. Avionics TSP WG
  - Establish requirements for the identified scenarios
  - Select/develop a Separation Kernel
  - Bench/validate the selected Separation Kernel
  - Demonstrate security requirements implementation
  - Gain experience in reaching high EAL/comparison with Category B software validation techniques
- End products/Outputs
  - Security Specification based on SKPP
  - Validation platforms
  - Tested and validated Security Kernels
  - Demonstrator and performance measurements

scisys

ASTRIUM
AN EADS COMPANY

Security Concerns (1/2)

- SW Trends in Space
  - Security requirements (e.g. in commercial applications)
  - SW developed by third party / use of COTS
  - Operation of Space platform shared by various entities
  - Use of low-cost service
  - Downgrading of data quality
- Reference scenarios
  - Multi-use missions
  - Payloads from different stakeholders
  - Integrated Modular Avionics
    - (i.e. communalization of hardware resources among several sub-systems of different criticality / confidence such as payload/equipment/OBSW)

scisys

ASTRIUM
AN EADS COMPANY

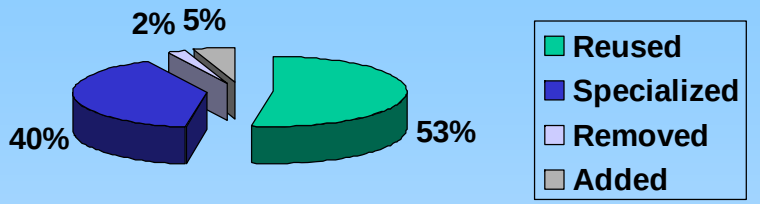
Security Concerns (2/2)

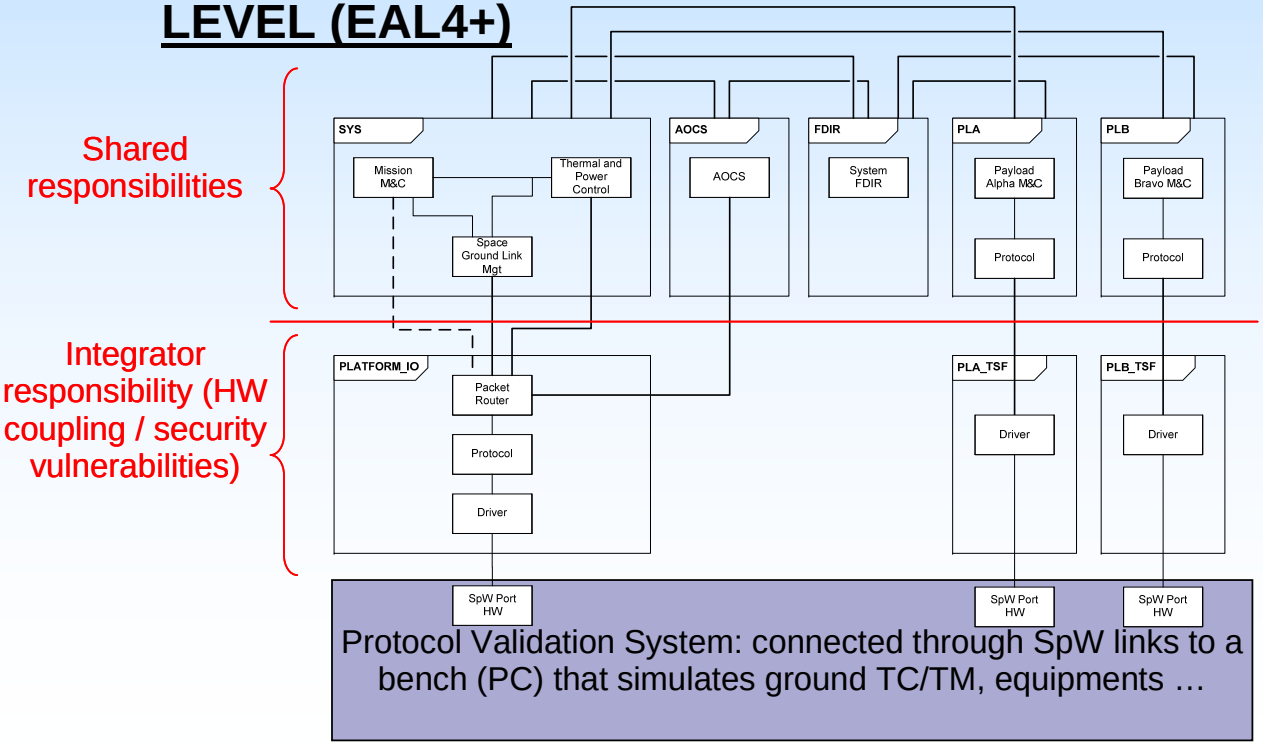
- Security objectives
  - Safe boot (no corruption of boot sequence or detection)
  - Data confidentiality/integrity/authentication
  - Observability
  - Compatibility with operational phases (e.g. FDIR, maintenance)
  - Control of resources (including CPU time, RAM, I/O, devices)
  - Prevent from error propagation, data leaks, covert/side channels
- Security Threats
  - Tampering with software (malware injection)
  - Equipment/Software malfunction
  - Saturation of the information system
  - Unauthorized use of equipment
  - Corruption or interception of data (i.e. encryption keys)
  - Illegal processing + abuse/forcing of rights
  - Error injection + denial of service

Astrium Study Overview

Focus on Security Specification and TOE Validation

- Security specification for Space greatly inherit from SKPP.
- Target Of Evaluation: XtratuM hypervisor (UPV)
- Maximization of automatic testing: 75% fully or mainly automated
- Validation methodology and EAL
  - **Test (65%) + Review of code/design (45%)**
  - **Test and Vulnerability Assessment compatible with <u>MIDDLE ASSURANCE LEVEL (EAL4+)</u>**





ASTRIUM Demonstrator :

• PF Applications Partitions with common communication bus

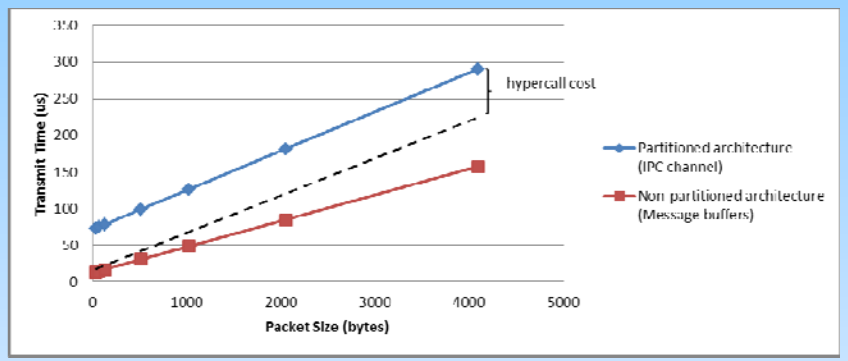
• Payloads Applications Partitions with specific communication bus

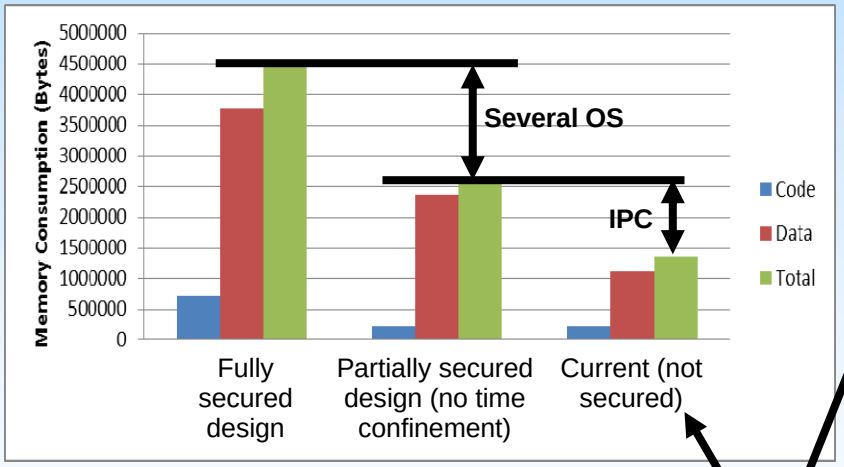
scisys

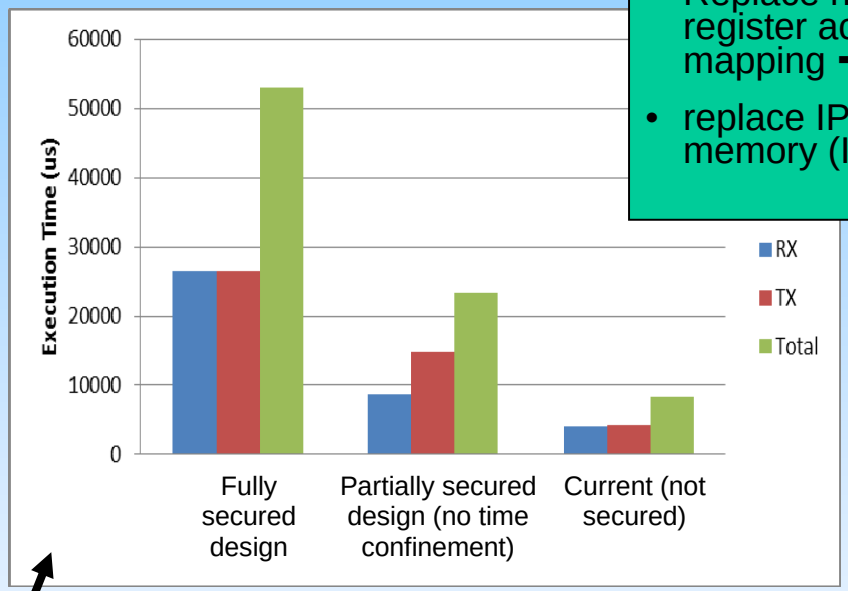
ASTRIUM
AN EADS COMPANY

Astrium Demonstrator

Focus on Performances Results







Foreseen improvements:

- Replace hypercall I/O register access with direct mapping ➔ 53µs/call
- replace IPC with shared memory (I/O) ➔ 110µs/KB

I/O Throughput	**Full Partitioned architecture**	**Space – Partitioned Architecture (with copy)**	**Non- partitioned architecture (zero copy)**
Max Throughput in each direction	19.82 Mbps	44.85 Mbps	127.28 Mbps

Three contributions to performances:

- The Secured Partitioned Architecture (design choices)
- The Partitioning Kernel (XtratuM)
- Use of Partitioning mechanisms (e.g. I/O register hypercall VS direct mapping)

scisys

ASTRIUM
AN EADS COMPANY

SciSys Study Overview

- Dual-use EO mission with two payloads
  - One producing confidential data
- Partitioned system using 6 partitions, supported:
  - PUS-based data handling system
  - Software maintenance
  - FDIR
  - I/O
- Using SYSGO PikeOS on LEON3 with MMU
  - RTEMS and POSIX guest OSs used
- Simulated spacecraft using modified ESA ATB and TSIM
- Exercised PikeOS in a realistic space context
- Provided indicative performance results

scisys

ASTRIUM
AN EADS COMPANY

SciSys Demonstrator and Results

- Demonstrator showed that use of partitioning technology with onboard software is feasible and has many advantages
- Highlights many issues to consider from architectural design stage onwards
  - I/O: No interrupts, no DMA, partition schedule coupling with I/O schedule
  - FDIR: Can be highly trusted, (semi-) centralised approach used in this case
  - Maintenance: Highly trusted
- Performance seems acceptable in testing
- Major performance limitations hardware-related

scisys

ASTRIUM
AN EADS COMPANY

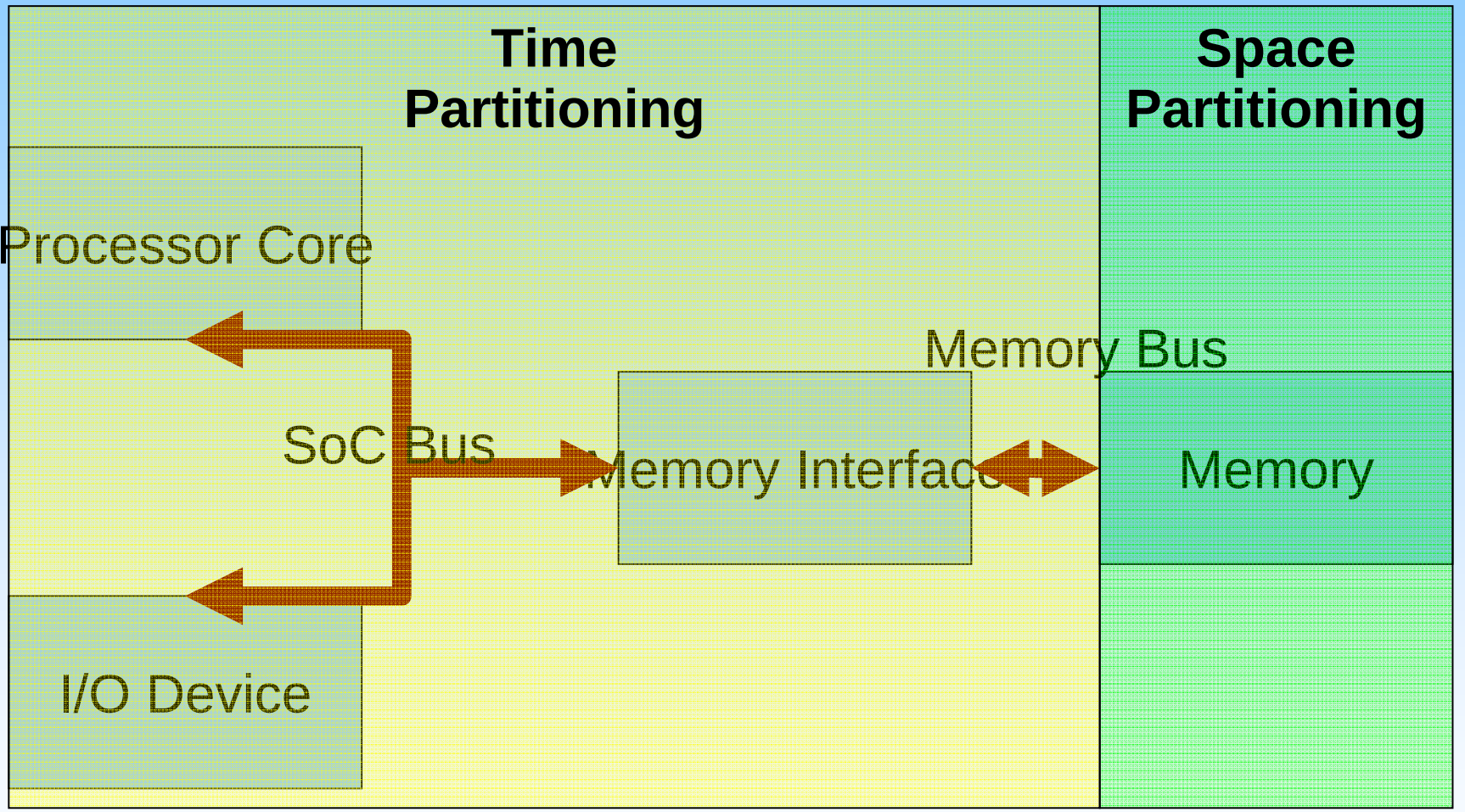
TSP and Multi-Core: Strategies and Benefits

- Partitioning similar in principle to a distributed system
  - With additional design constraints
- Partitioning/multi-core is a logical and powerful combination
- Designing for partitioning is a good way to move to a multi-core system
- Partitions could be assigned to cores statically or dynamically (AMP vs SMP)
  - Static assignment preferred
- Permits load balancing
- Partition-core assignment could be changed without a full re-validation

TSP and Hardware

- Back to first principles...
- Two types of system resources
  - Atomic (indivisible) resources e.g. processor
  - Non-atomic (divisible) resources e.g. memory
- Non-atomic resources can be divided up
  - Spatial partitioning
  - Protected by MMU
- Atomic resources must be multiplexed in time
  - Temporal partitioning
  - Utilising timer interrupt
  - ”Protected” by hypervisor software

Single-Core Temporal Partitioning



- Processor core, SoC bus, all interfaces treated as a single resource
- Partitioned by timer interrupt + software

scisys
ASTRIUM
AN EADS COMPANY

Single-Core I/O Handling Issues

- Interrupts
  - Alter system timing and affect the schedule
  - Hypervisor can trap interrupt to ensure system integrity
  - Timing still affected = covert channel
  - Interrupts **not permitted** in a secure system
- DMA
  - Devices with DMA capability can access memory
  - Memory accessed using physical addresses
  - Unprotected access to memory
  - Timing affected = loss of integrity + confidentiality
  - DMA **not permitted** in a secure system
- Major loss of I/O performance

Multi-Core Hardware Issues

- Multi-core partitioning introduces resource contention
- Processor core and bus can no longer be treated as one resource
  - Other cores affect timing on the bus
- Time slicing by processor not sufficient
  - Bus access not regulated or protected
- Same problem as for DMA on single-core systems
  - Unpartitioned atomic resource
- Difficult to validate for
- No guaranteed integrity = **not safe**
- No guaranteed confidentiality = **not secure**

Hardware Support for Partitioning

- Spatial partitioning for DMA devices can be solved by using an IOMMU
  - Simple version avilable on SCOC3
  - Full IOMMU in GRLIB and on LEON4
- For temporal partitioning time-based arbitration must be added to atomic resources
  - SoC bus(es) using bus controller
  - Interrupts using interrupt controller
- Temporal partitioning issues not considered by current hardware

SPARC/LEON-Specific Issues

- Cache handling
  - Cache must be flushed on partition switch
  - Gives predictable environment so no loss of integrity or confidentiality
  - LEON3 cache does not store permissions
  - Supervisor-mode cache contents available to user-mode partition code after hypervisor call
  - Cache must be flushed after hypervisor call
- Register windowing
  - Over/underflow interrupt must be handled by hypervisor
  - Complete window set must be saved/restored on partition switch

Recommendations

- ***Some clear points:***
  - Do provide IOMMUs but these are no use on their own
  - Configurable schedule could be added to bus controller/arbiter
    - Simple schedule based on credit of cycles
    - Schedule slot detected based on MMU context ID
  - Investigate possibility for handling register window over/underflow in partitions
  - Limited hypervisor mode c.f. UltraSPARC
- ***Other points to be deeper analysed:***
  - Configurable schedule could be added to interrupt controller
    - Bitmask indicating permission to raise an interrupt
    - MMU context ID used again
  - User/Supervisor mode permissions should be added to cache
  - Support cache freeze on software trap (not just async trap)

Summary and Conclusions

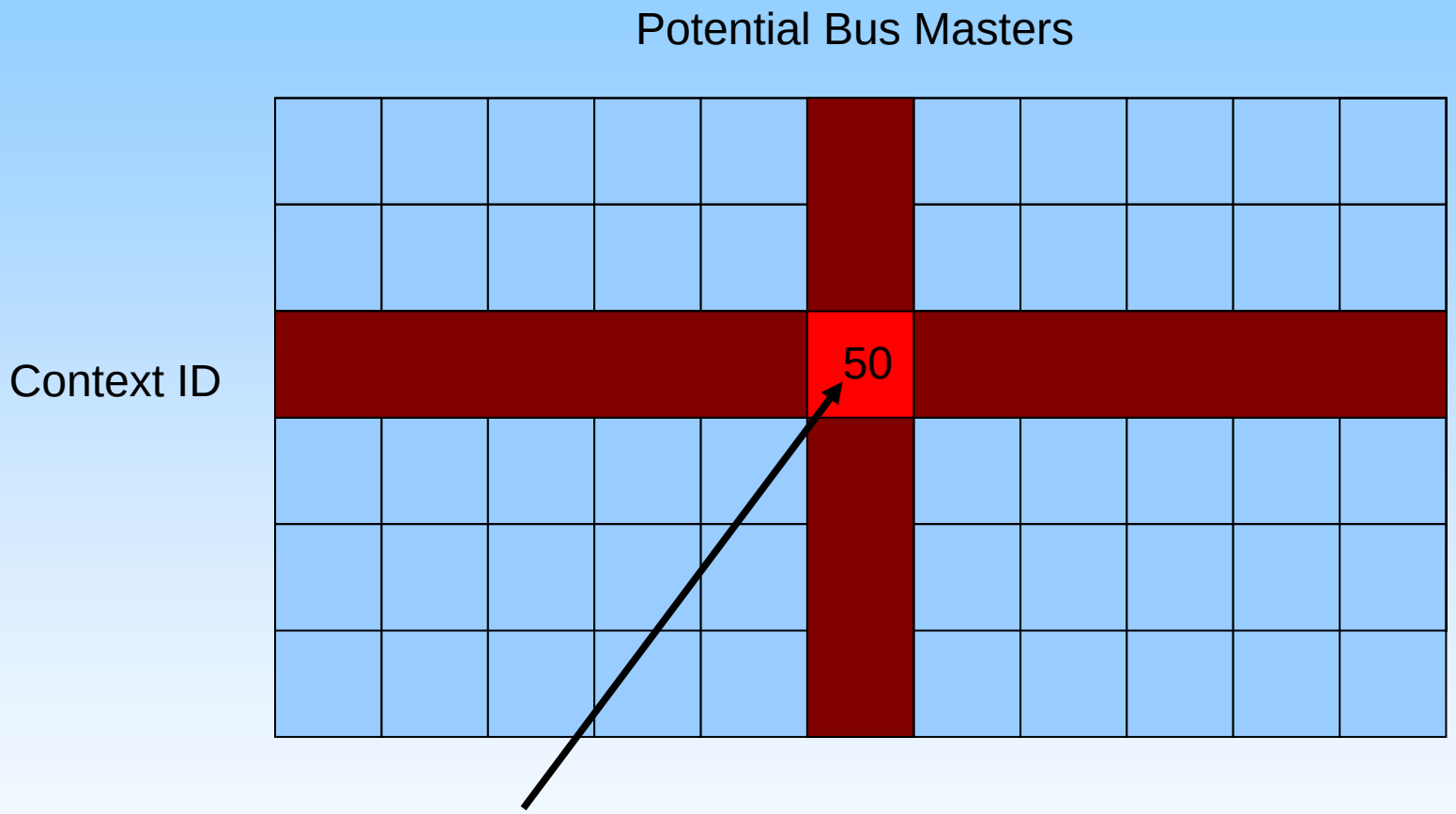
- Two parallel studies conducted
  - Grounded in realistic security needs
- Astrium study focussed on space security needs & validation
- SciSys study focussed on space-industry needs
- While demonstrated on two different platforms, both studies identified shortcomings in hardware support for secure partitioning
- Multi-core could be an excellent match for TSP but highlights hardware issues
- Recommend additional hardware support be considered
- Essential/mid-term: Schedule-based bus/interrupt arbitration
- Longer-term: Hypervisor mode

scisys

ASTRIUM
AN EADS COMPANY

Backup Slides

Bus Arbiter Schedule Table

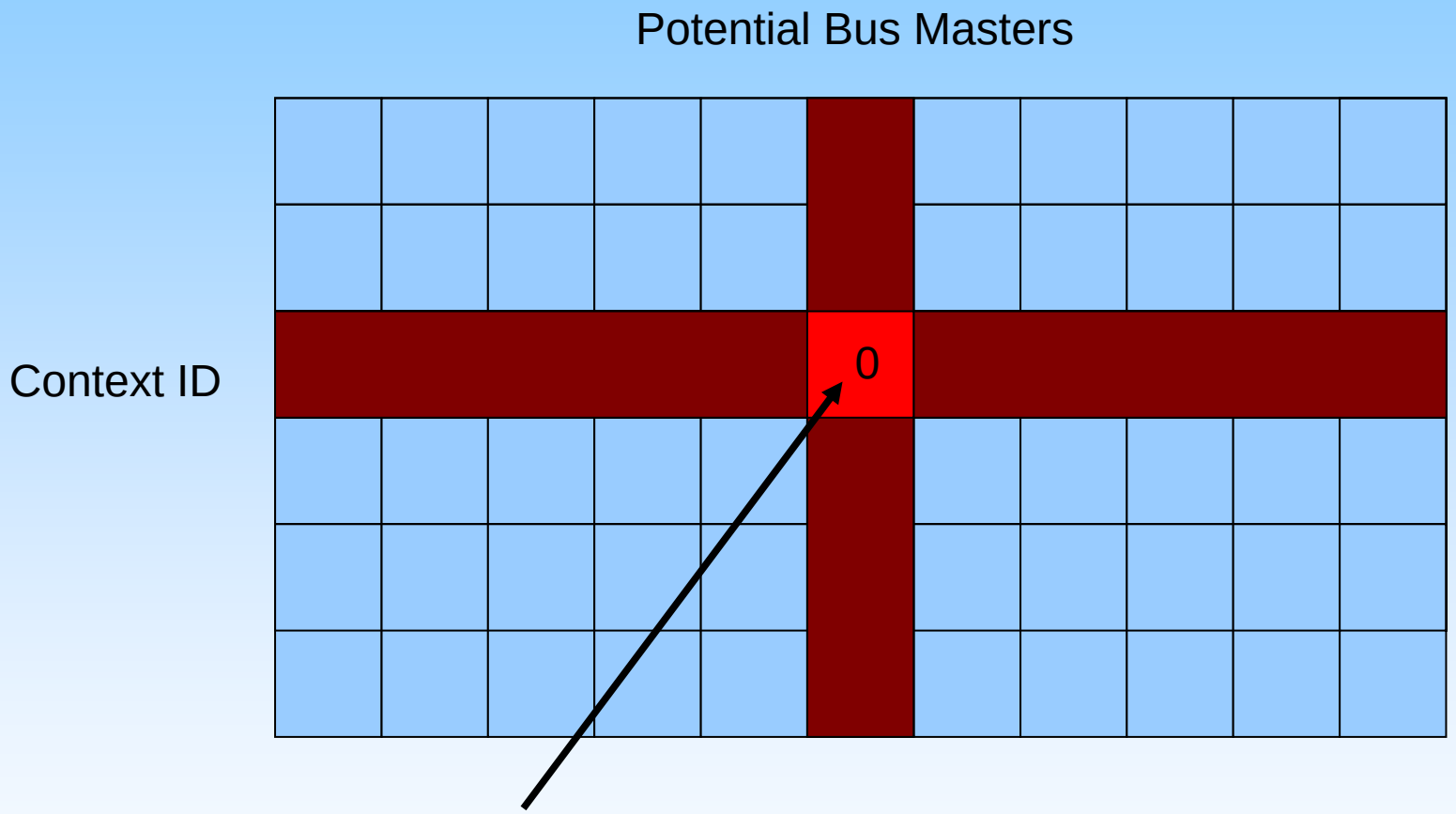


On a multi-core system the arbiter schedule may need to be linked to/controlled by a single “master” or “root” core.

scisys

ASTRIUM
AN EADS COMPANY

Interrupt Controller Schedule



On a multi-core system the arbiter schedule may need to be linked to/controlled by a single “master” or “root” core.

scisys

ASTRIUM
AN EADS COMPANY