Virtualization on TrustZone-enabled Microcontrollers? Voilà

S. Pinto, H. Araujo, D. Oliveira, J. Martins, A. Tavares

Presented by: Álvaro Albero Gran

Problem domain and motivation

- An increase of IoT devices, applications and functionality
- Demand for strong isolation
- Available in high-end embedded devices, not in low-cost systems
- Traditional implementation was to physically separate the systems, but this does not scale in size and cost

Contributions

- A virtualization solution for low cost embedded systems
- Use of TrustZone technology on a modern Arm microcontroller
- Lightweight hypervisor
  - Strong isolation
  - Low memory footprint
  - High efficiency
  - Strict timing predictability

Arm TrustZone

- Hardware security-oriented technology
- Secure and non-secure worlds
- Already used in high-end devices (Cortex-A processors)
- TrustZone-M, version for MCUs (Cortex-M MCUs)
- There are a set of challenges

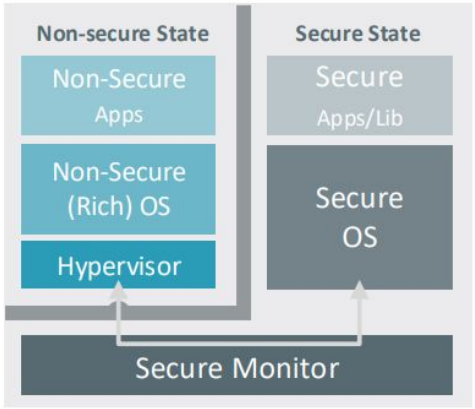
More Arm TrustZone

TrustZone

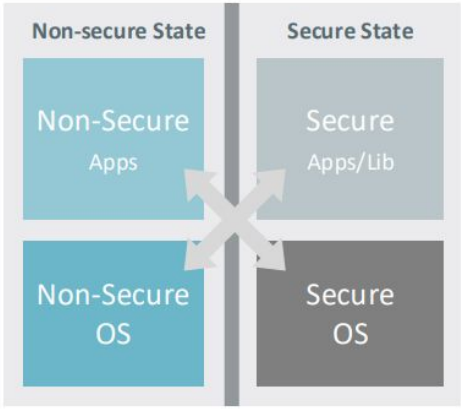
- Non-Secure bit
- Secure Monitor
- TZASC and TZPC

TrustZone-M

- Similar high level, but differences
- Partitioned physical address space
- Secure Attribution Unit (SAU)
- Nested Vectored Interrupt Controller (NVIC)



(a) TrustZone for Cortex-A



(b) TrustZone for Cortex-M

TrustZone Assisted Virtualization

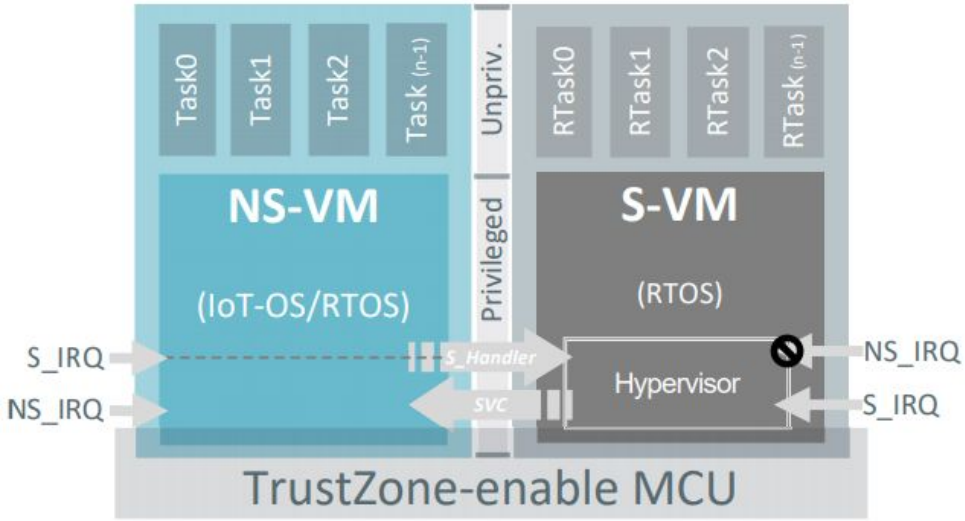
- Hardware-assisted virtualization
- Achieves time and spatial isolation
- Dual-guest and multi-guest configurations

Challenges switching to TrustZone-M

- TrustZone-M excludes Non-Secure bit
- TrustZone-M excludes SMC
- TrustZone-M excludes TZASC and TZPC
- NVIC does not provide FIQ

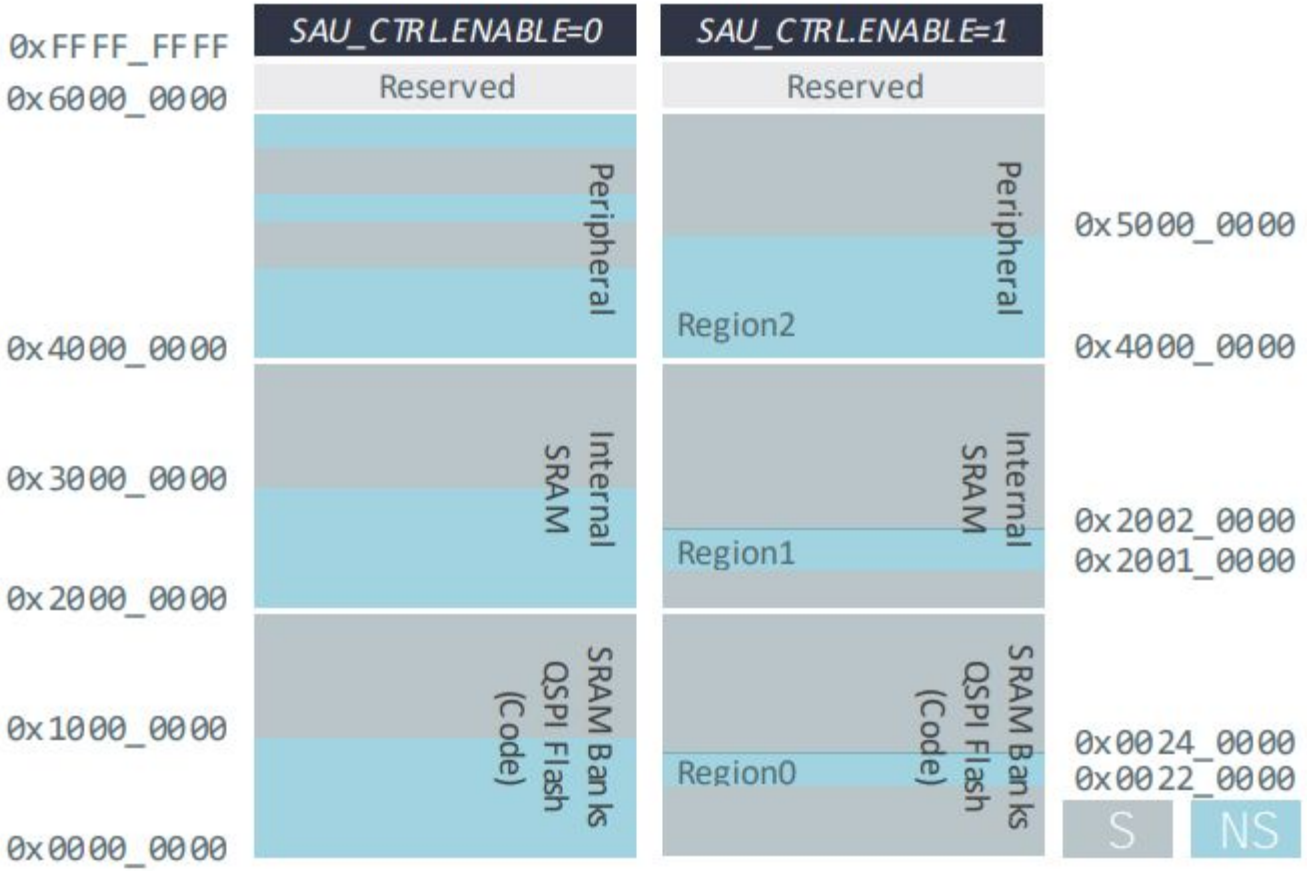
TrustZone-M Hypervisor

- Dual-OS configuration
- Hypervisor decoupled from secure OS
- The figure shows the high level architecture for single-core
- The implementation details are
  - CPU virtualization
  - Memory and device partitioning
  - Interrupt and time management
- AMP Configuration



Tru

-
-

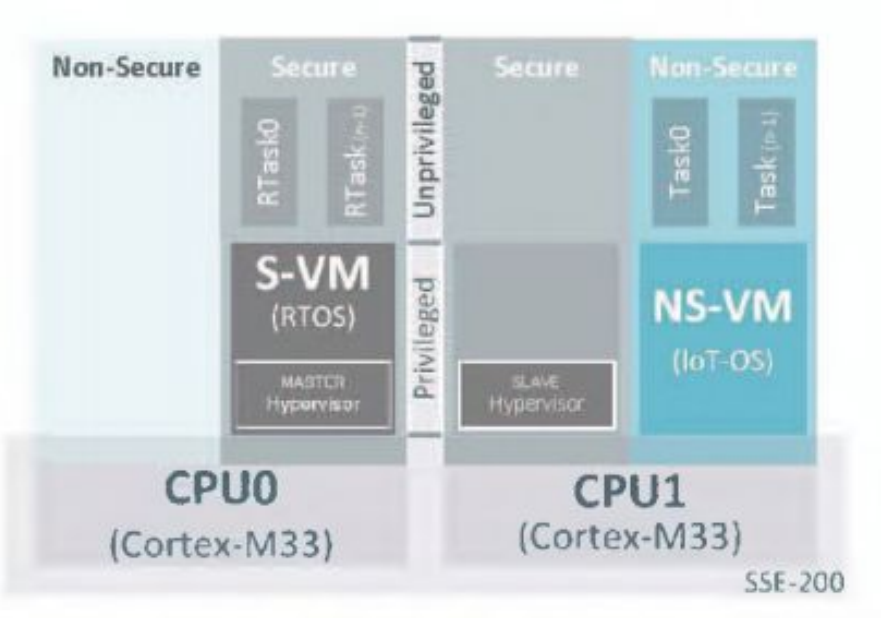

0xFFFF_FFFF
0x6000_0000
0x4000_0000
0x3000_0000
0x2000_0000
0x1000_0000
0x0000_0000
SAU_CTRL.ENABLE=0
Reserved
Peripheral
Internal SRAM
SRAM Banks QSPI Flash (Code)
SAU_CTRL.ENABLE=1
Reserved
Peripheral
Region2
Internal SRAM
Region1
SRAM Banks QSPI Flash (Code)
Region0
0x5000_0000
0x4000_0000
0x2002_0000
0x2001_0000
0x0024_0000
0x0022_0000
S
NS

TrustZone-M Hypervisor

- Interrupt Management
  - Configure IRQs as secure or non-secure
  - Each VM is able to handle their own interrupts
- Time Management
  - Temporal isolation is required for virtualization
  - Leverate TrustZone-M timing facilities
  - SysTick
  - S-VM SysTick has privileges
  - AMP overcomes time limitations

TrustZone-M Hypervisor

- AMP Configuration



- Scalability
  - They do not provide support for multiple guests
  - But existing resources make it possible
  - They plan to implement this architecture

Predictable shared resources management

- Embedded virtualization has benefits but is full of challenges
- AMP hypervisors remove part of this contention
  - Some System wide resources are still subject to contention
- Approaches to solve this are not available on MCUs
- But many of these resources are not used in low-end platforms
- With these ideas in mind, the authors believe they can achieve high determinism on AMP virtualization through an informed and thoughtful layout of VM memory

Predictable shared resources management

- The Arm Musca-A memory subsystem

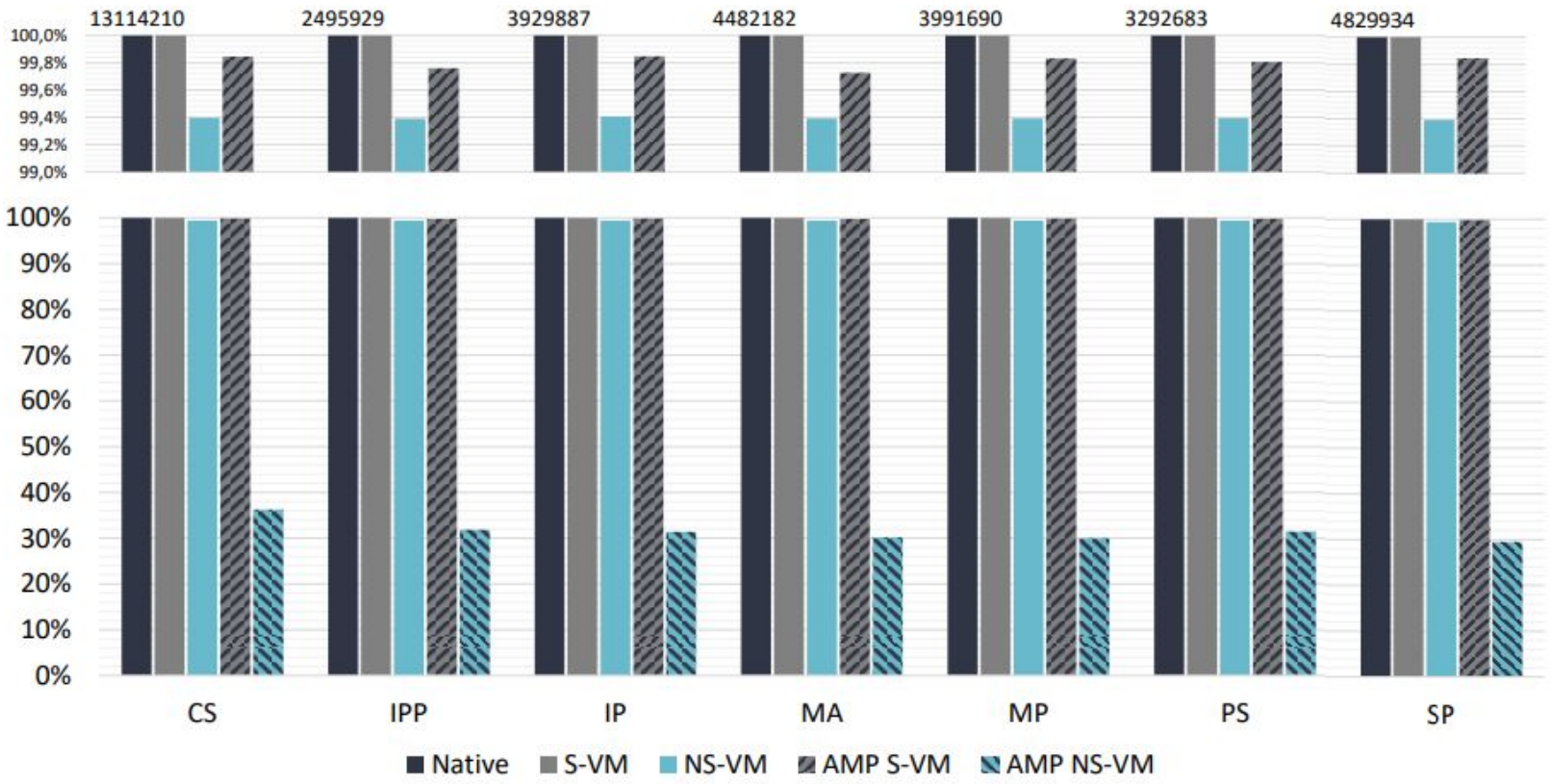
- Contention-Aware memory layout
  - Memory layout to minimize contention of shared resources
  - Start with ideal scenario and iterate over it
  - They envision an automated tool
  - Peripheral assignment is out of scope and may result in contention

Evaluation - Setup

- Arm Musca-A chip
- Both cores at 50 MHz
- Single and multi-core configuration (AMP)
- FreeRTOS as guest OS for both S-VM and NS-VM
- The evaluation focuses on performance, interrupt latency and contention
- Runtime overhead of the hypervisor on VM execution
- Interrupt latency, additional jitter at the VM level
- Contention, how the memory layout can lead the NS-VM to create contention on shared buses and affect time predictability on S-VM

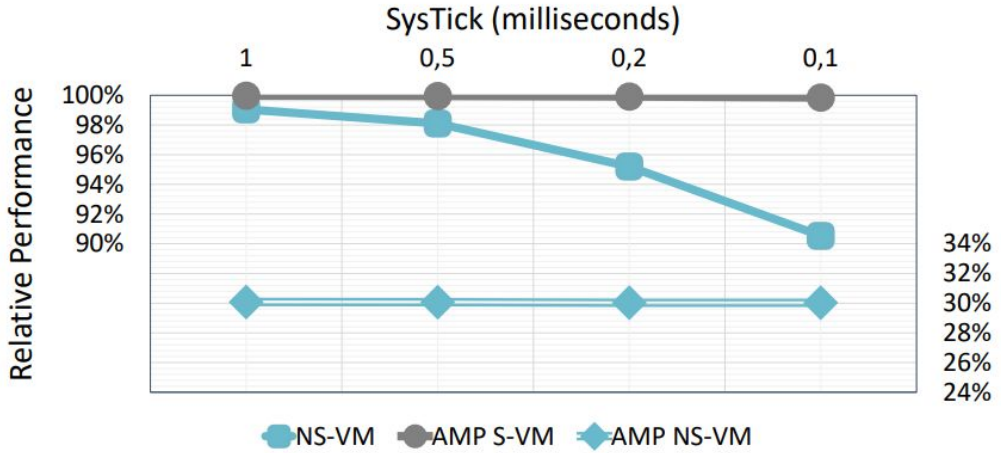
E

-


13114210
2495929
3929887
4482182
3991690
3292683
4829934
100,0%
99,8%
99,6%
99,4%
99,2%
99,0%
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
CS
IPP
IP
MA
MP
PS
SP
Native
S-VM
NS-VM
AMP S-VM
AMP NS-VM

Evaluation - Performance Overhead

- SysTick Overhead

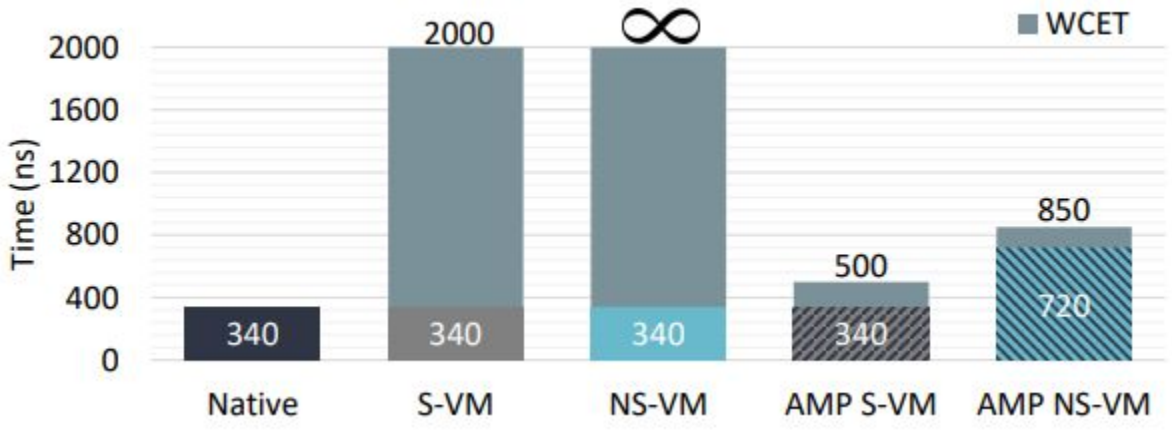


- Starvation
  - Starvation in NS-VM
  - Task in S-VM with workloads of 0-75%
  - Single core - performance decreases linearly, complete starvation at 100%
  - AMP overcomes starvation

Evaluation - Interrupt Latency

The time from the moment an interrupt is triggered till the moment the handler starts to execute, it is critical in real time systems

- Timer triggers every 10ms



Conclusions

- Lightweight virtualization infrastructure for low-end systems
- Using TrustZone-M technology
- Use a low-end Arm multi-core platform
- Reduced memory footprint, high efficiency and determinism
- First virtualization infrastructure for TrustZone enabled MCUs

Critiques

- @nikorev: “TrustZone-M will be a game-changer for low-end virtualization. However, as of this writing, existing TrustZone-assisted hypervisor have no support for Arm8-M.”
- @nikorev and @tuhinadasgupta: single-core performance using virtualization goes down drastically
- @nikorev: single-core solution opens up opportunity for a denial-of-service through contention
- @tuhinadasgupta: worried about compatibility as not all MCU implement TrustZone-M

Appendix

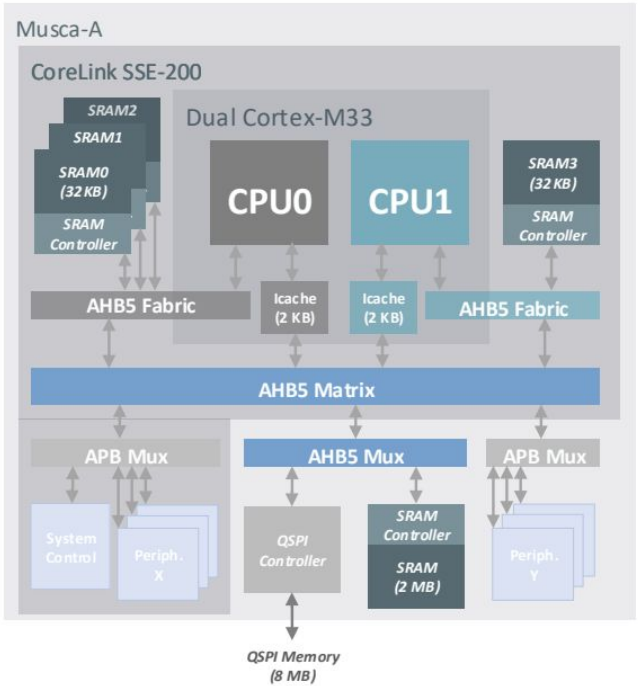
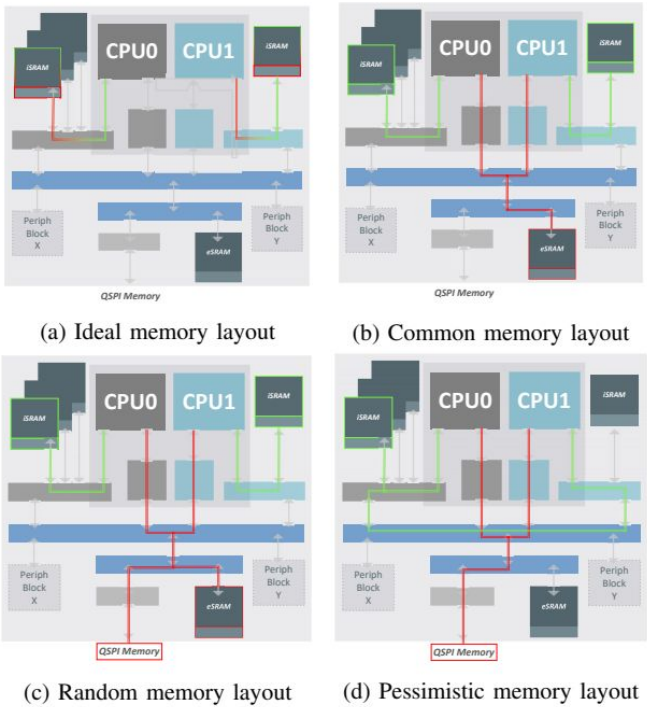


Fig. 9: Musca-A chip memory and interconnect block diagram. Adapted from [27].



(a) Ideal memory layout

(b) Common memory layout

(c) Random memory layout

(d) Pessimistic memory layout

Fig. 10: Path of memory configurations in Musca-A. Green line for data and red line for code.