東南大學
SOUTHEAST UNIVERSITY

OPERATING SYSTEM CONCEPTS

Chapter A1. TrustZone

A/Prof. Kai Dong

Contents

Intro. to TrustZone

- Arm TrustZone consists of hardware security extensions introduced into Arm application processors (Cortex-A) in 2004. [1][2]
- More recently, TrustZone has been adapted to cover the new generation of Arm microcontrollers (Cortex-M). [3][4]

[1] T. Alves and D. Felton. 2004. TrustZone: Integrated hardware and software security. Tech. In-Depth 3, 4 (2004), 18–24.

[2] Arm Ltd. 2009. ARM Security Technology: Building a Secure System using TrustZone Technology.

[3] Arm Ltd. 2017. TrustZone technology for ARMv8-M Architecture. Version 2.0.

[4] J. Taylor. 2016. Security for the next generation of safe real-time systems. In Proceedings of Embedded World Conference.

Intro. to TrustZone

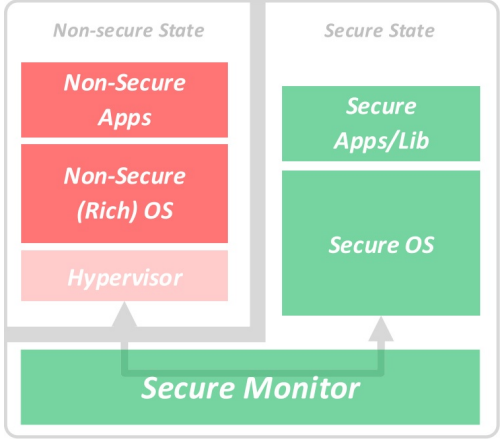
- TrustZone:
  - Follows a System-on-Chip (SoC) and CPU system-wide approach to security.
  - Is centered around the concept of protection domains named secure world and normal world.
    - The software executed by the processor runs either in the secure or non-secure states.
    - On Cortex-A processors, the privileged software referred by the name of secure monitor implements mechanisms for secure context switching between worlds;
    - on Cortex-M processors, there is no secure monitor software and the bridge by both worlds is handled by a set of mechanisms implemented into the core logic.
  - Both worlds are completely hardware isolated and granted uneven privileges, with non-secure software prevented from directly accessing secure world resources.

Contents

TrustZone for Arm Cortex-A Processors

Normal World and Secure World

- The most important architectural change at the processor level:
  - Two protection domains designated by the name of worlds: the secure world and the normal world.

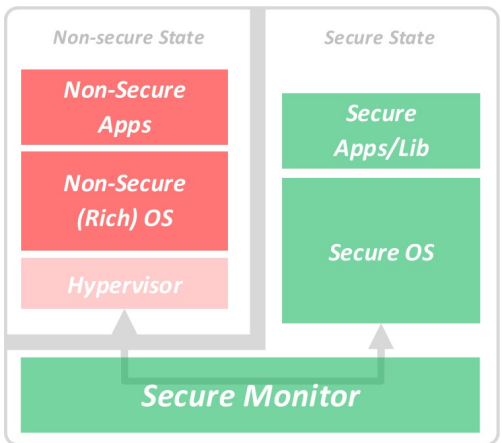


- At a given point in time, the processor operates exclusively in one of these worlds.
- The world where the processor currently executes is determined by the value of a new 33rd processor bit, also known as the Non-Secure (NS) bit.
- The value of this bit can be read from the Secure Configuration Register (SCR).
- It is propagated throughout the system down to the cache, memory and peripheral buses.

TrustZone for Arm Cortex-A Processors

Monitor Mode

- TrustZone introduces an extra processor mode (monitor mode) that is responsible for preserving the processor state whenever world transitions occur.

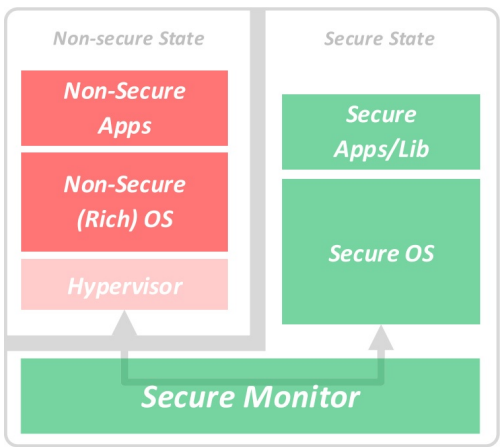


- A new privileged instruction—Secure Monitor Call (SMC)—allows for the software stacks residing in both worlds to be bridged by the monitor software.
- Other than through this instruction, it is possible to enter monitor mode via proper configuration of exceptions, interrupts (IRQ), and fast interrupts (FIQ) handled in the secure world.

TrustZone for Arm Cortex-A Processors

Registers

- To reinforce hardware isolation between worlds:



- The processor has banked versions of the special registers, as well as some system registers.
- In the normal world, the security-critical system registers and processor core bits are either totally hidden or conditioned by a set of access permissions supervised by the secure world software.

TrustZone for Arm Cortex-A Processors

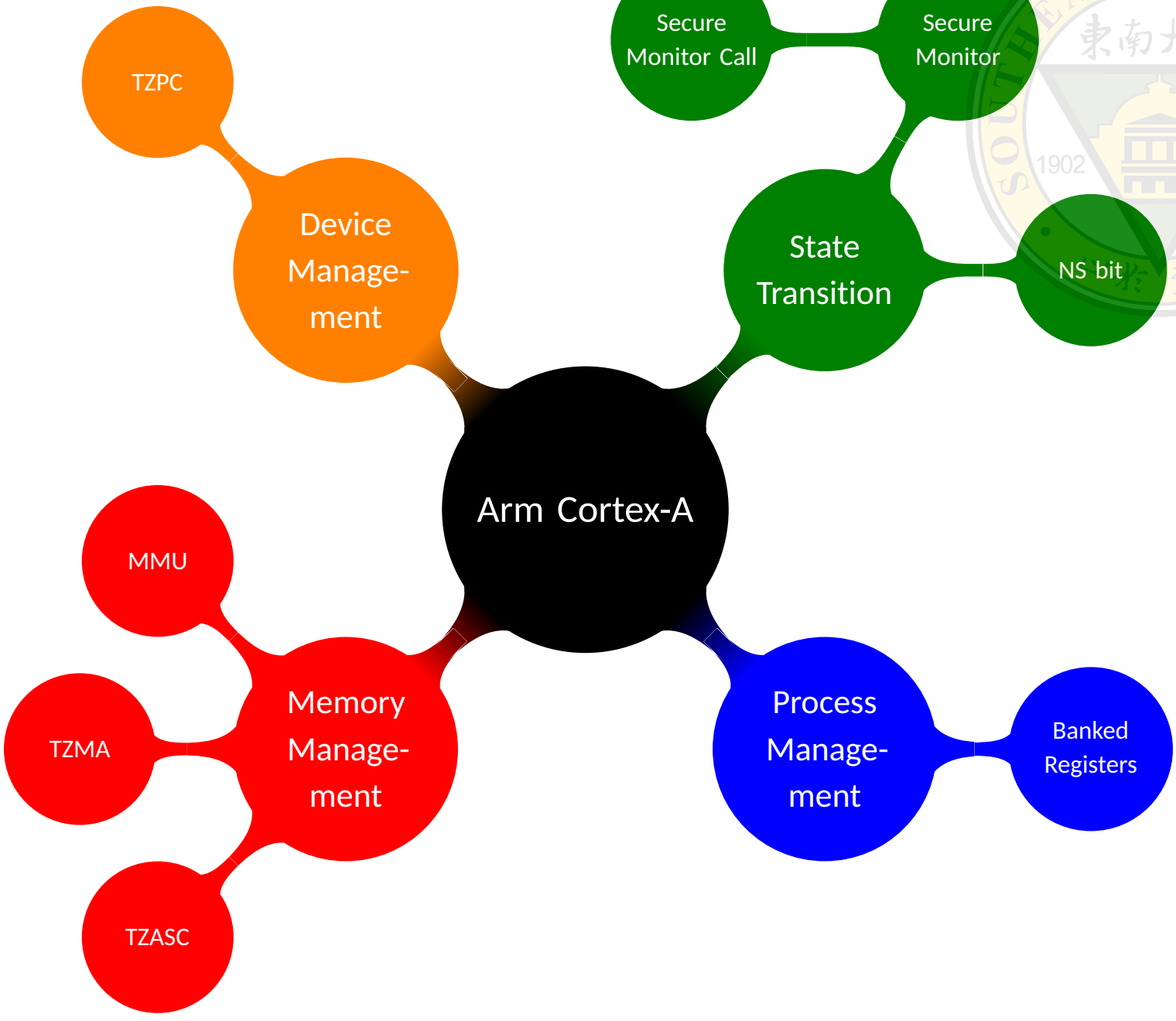
Memory

- Two optional components defined by the TrustZone specification.
  - TrustZone Address Space Controller (TZASC)
    - The TZASC can be used to configure specific memory regions as secure or non-secure, such that applications running in the secure world can access memory regions associated with the normal world, but not the otherwise.
    - Partitioning the DRAM into different memory regions and its respective association with a specific world is performed by the TZASC under the control of a programming interface restricted to the software running with secure world privileges.
  - TrustZone Memory Adapter (TZMA)
    - A similar memory partitioning functionality is implemented by the TZMA, but targeting off-chip ROM or SRAM.
- The TrustZone-aware Memory Management Unit (MMU) allows for each world to have its own virtual-to-physical memory address translation tables.

TrustZone for Arm Cortex-A Processors

Devices

- An optional component.
  - TrustZone Protection Controller (TZPC)
    - » The fact that the TZPC is an implementation specific component leads to diversity in the number and type of TrustZone-aware devices that can be found across hardware platforms.
- The TrustZone architecture extends the Generic Interrupt Controller (GIC) with support for prioritized secure and non-secure sources.

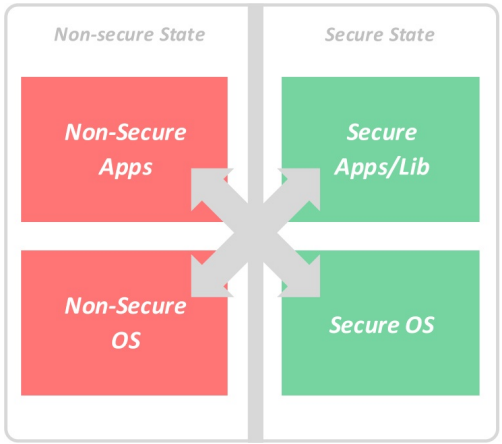

Arm Cortex-A
Device Manage-ment
TZPC
State Transition
Secure Monitor
Secure Monitor Call
NS bit
Memory Manage-ment
MMU
TZMA
TZASC
Process Manage-ment
Banked Registers

Contents

TrustZone for Arm Cortex-M Microcontrollers

{Thread, Handler}×{Secure, Non-secure}

- Cortex-M is optimized for faster context switch and low-power applications.
- Mainstream requirements: low power consumption, real-time processing, deterministic behavior, and low interrupt latency.



- The division between worlds is memory map-based and the transitions take place automatically in exception handling code.
- When running code from the secure memory, the processor state is secure, and, when running code from non-secure memory, the processor state is non-secure.
- There are both a Thread and Handler mode in secure and non-secure states.
- There is no monitor mode, or any secure monitor software, thus considerably reduces the world switch latency, which translates to more efficient transitions.

TrustZone for Arm Cortex-M Processors

State transitions

- Three new instructions bridging software between both worlds.
  - The secure gateway (SG) instruction is used for switching from the non-secure to the secure state at the first instruction of a secure entry point;
  - The branch with exchange to non-secure state (BXNS) instruction is used by secure software to branch or return to the non-secure program;
  - The branch with link and exchange to non-secure state (BLXNS) instruction is used by secure software to call non-secure functions.
- State transitions can also happen due to exceptions and interrupts

TrustZone for Arm Cortex-M Processors

Registers

- Most of the register file is shared between secure and non-secure states, excepting for stack pointers.
- To separate secure and non-secure stacks, four physical stack pointers are supported.
  - Both security states implements the main stack and the process stack.
  - The starting address of the vector table is determined by a memory-mapped register called the Vector Table Offset Register (VTOR) in the System Control Block (SCB).
  - The VTOR register is banked, which means that one instance exists in each world.
- Some of the special registers are also banked: the Priority Mask, Control, as well as the Fault Mask Register and the Base Priority registers, just to name a few.

TrustZone for Arm Cortex-M Processors

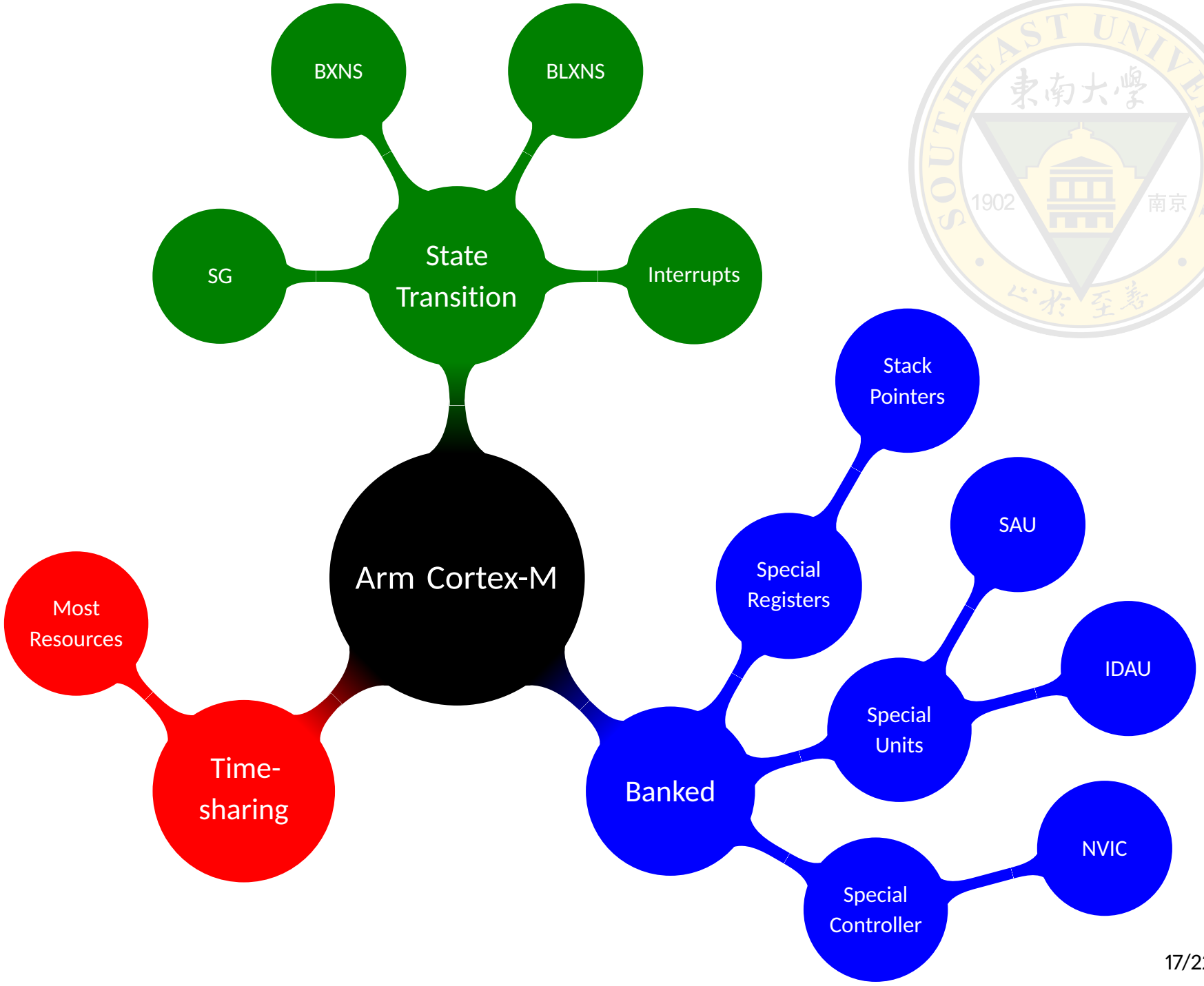
Memory

- The memory space is also partitioned into secure and non-secure sections.
  - Non-secure addresses are used for memory and peripherals accessible by all software that is running on the device.
  - The secure memory space is further divided into two types: secure and non-secure callable (NSC).
    - » Secure addresses are used for memory and peripherals accessible only by secure software.
    - » NSC addresses are used to hold SG instructions that allow software to transition between non-secure and secure states.
  - The security state attributed to each address is determined by the internal Secure Attribution Unit (SAU) or by an external Implementation Defined Attribution Unit (IDAU).
    - » SAU is always present but the number of regions is implementation-specific, and can be programmed in the secure state.
    - » IDAU is optional and processor-specific.
  - Optionally two distinct TrustZone-aware Memory Protection Units (MPU) related to memory access permissions for privileged and unprivileged software.

TrustZone for Arm Cortex-M Processors

Interrupt

- The Nested Vectored Interrupt Controller (NVIC) was also extended for security.
  - Each interrupt can be configured as secure or non-secure through the Interrupt Target Non-secure register (NVIC_ITNS).
  - NVIC_ITNS is only programmable in the secure world.
  - There are no restrictions regarding whether a non-secure or secure interrupt can take place when the processing is running non-secure or secure code.
    - » If the arriving exception or interrupt has the same state as the current processor state, then the exception sequence is similar to the previous M-series processors.
    - » When a non-secure interrupt takes place and is handled by the processor during the execution of secure code: the processor automatically pushes all secure information onto the secure stack and erases the contents from the register banks — this mechanism avoids any leakage of information.


BXNS
BLXNS
SG
State Transition
Interrupts
Stack Pointers
SAU
Special Registers
IDAU
Most Resources
Arm Cortex-M
Special Units
Time-sharing
Banked
NVIC
Special Controller

Contents

TrustZone-enabled Platforms

Platform	SoC	Processor	Multicore	Publicly?
CubieBoard4	Allwinner A80	Cortex-A15/A7	quad-core/quad-core	No
Musca-A1 Board	Arm Musca-A1	Cortex-M33	dual-core	Yes
V2M-Juno r2	Arm Juno (r2)	Cortex-A72/A53	dual-core/quad-core	Yes
SAML11 Xplained Pro	Microchip SAML11	Cortex-M23	single-core	Yes
SAMA5D2-XULT	Microchip SAMA5D2	Cortex-A5	single-core	Yes
MiniZed	Xilinx Zynq-7000	Cortex-A9	single-core	Yes
PYNQ-Z1	Xilinx Zynq-7000	Cortex-A9	dual-core	Yes
ZedBoard	Xilinx Zynq-7000	Cortex-A9	dual-core	Yes
ZYBO	Xilinx Zynq-7000	Cortex-A9	dual-core	Yes
ZC702 Eval. Kit	Xilinx Zynq-7000	Cortex-A9	dual-core	Yes
NuMicro M2351	Nuvoton M2351	Cortex-M23	single-core	Yes
Jetson TK1 DevKit	Nvidia	Cortex-A15	quad-core	No
Jetson TX2 DevKit	Nvidia	Cortex-A57/Denver	quad-core/dual-core	No
IMX53QSB	NXP i.MX53	Cortex-A8	single-core	Yes
iMX6UL-EVK	NXP i.MX6 UL	Cortex-A7	single-core	Yes
RD-IMX6Q-SABRE	NXP i.MX6	Cortex-A9	quad-core	Yes
MCIMX7-SABRE	NXP i.MX7	Cortex-A7/M4	dual-core/single-core	Yes
Raspberry Pi 3	Broadcom BCM2837	Cortex-A53	quad-core	Yes
R-Car Starter Kit	Renesas R-Car H3	Cortex-A57/A53	quad-core/quad-core	No
ZCU102 Eval. Kit	Xilinx UltraScale+	Cortex-A53/R5	quad-core/dual-core	Yes

1. SAML11 Xplained Pro is used by Zhen Ling's Group.
2. Raspberry Pi 3 may be used by Kai Dong's Group.
3. Xilinx Zynq-7000 integrates the software programmability of Arm-based processors with the hardware programmability of a Field-Programmable Gate Array (FPGA).

Contents

The State-of-the-Art

- TrustZone-assisted TEE
- TrustZone-assisted virtualization

Kai Dong

Ph.D., Associate Professor