TEE Boot Procedure with Crypto-accelerators in RISC-V Processors

Authors: Ckristian Duran, Trong-Thuc Hoang, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham

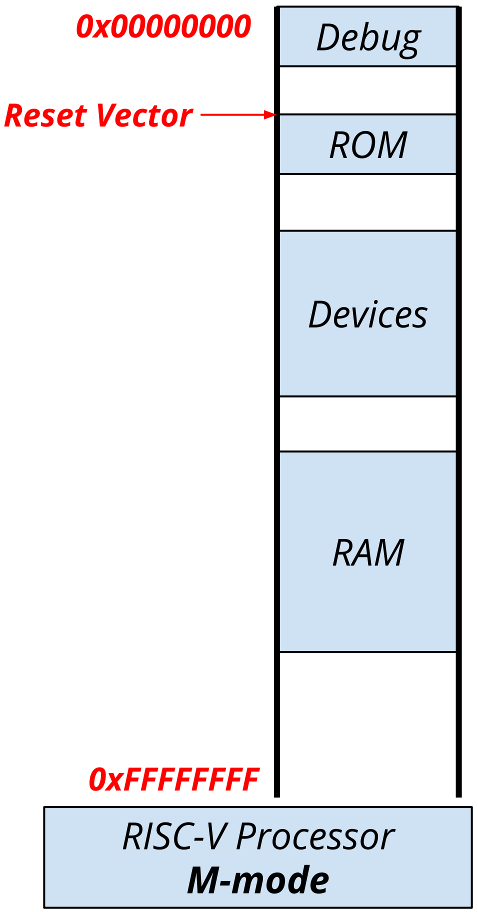
Outline

- Motivation
- Hardware Structure for Trusted Execution Environments
- Boot Procedure with Crypto-accelerators
- Implementation Results
- Conclusions

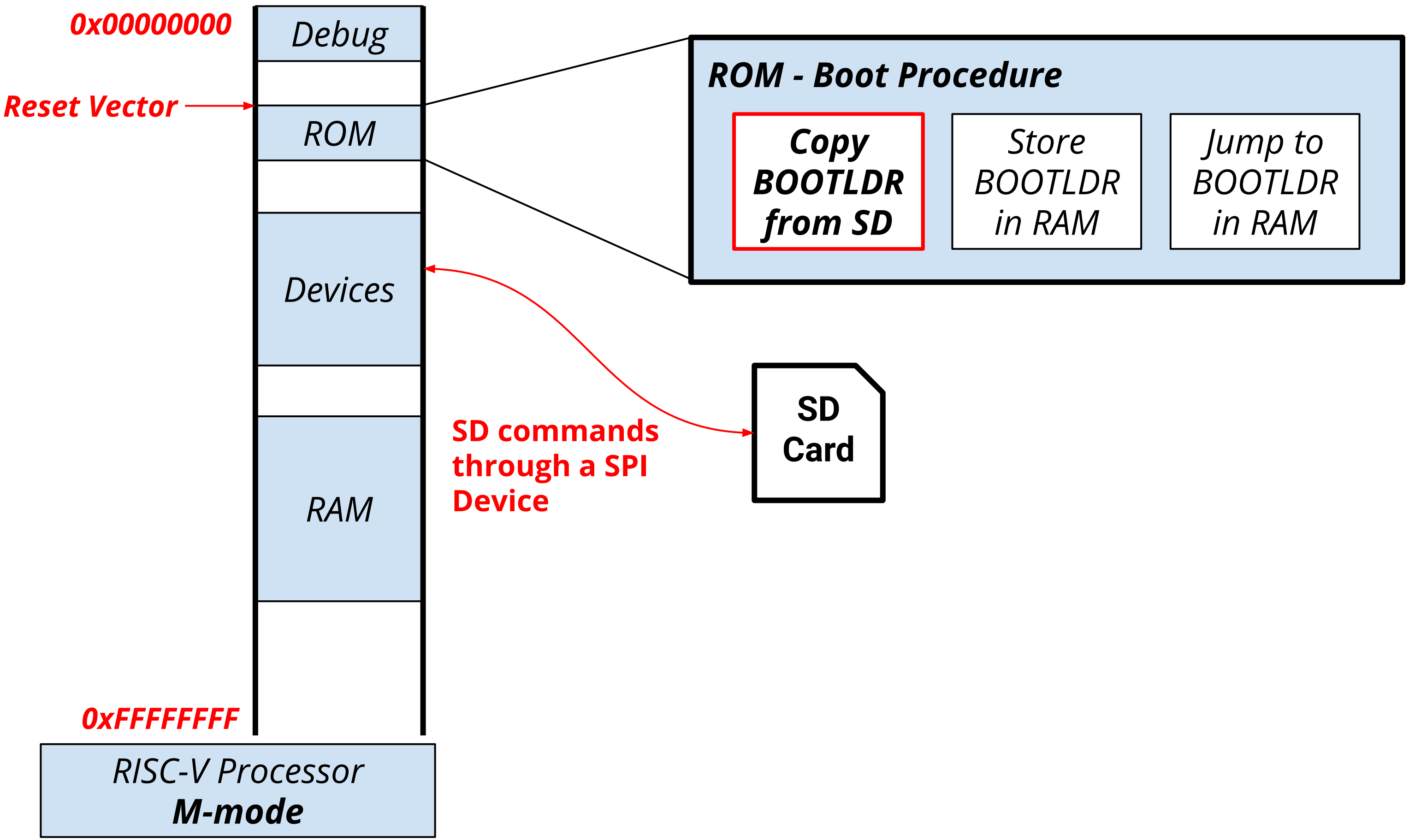
Outline

- Motivation
- Hardware Structure for Trusted Execution Environments
- Boot Procedure with Crypto-accelerators
- Implementation Results
- Conclusions

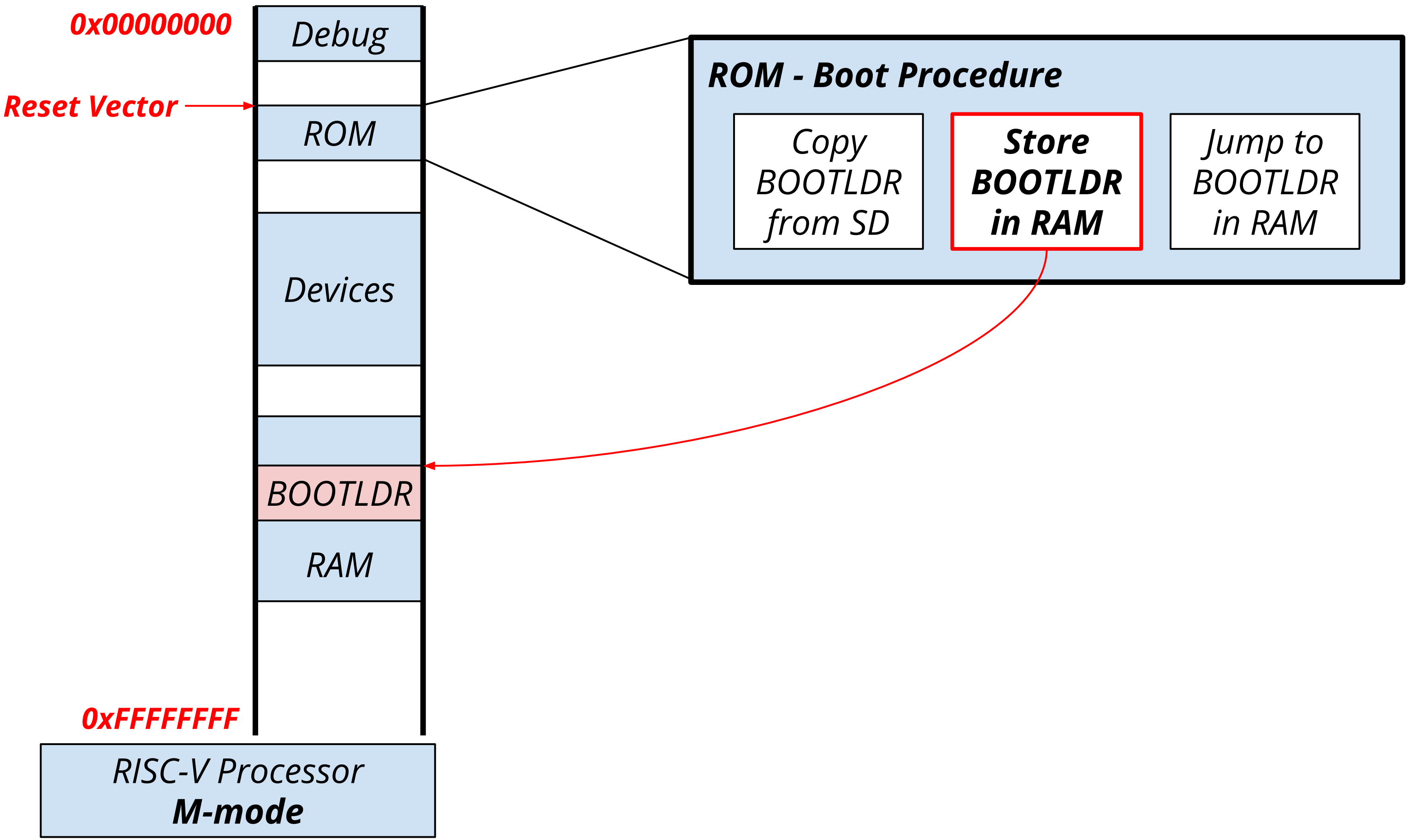
RISC-V Processor Privilege Modes



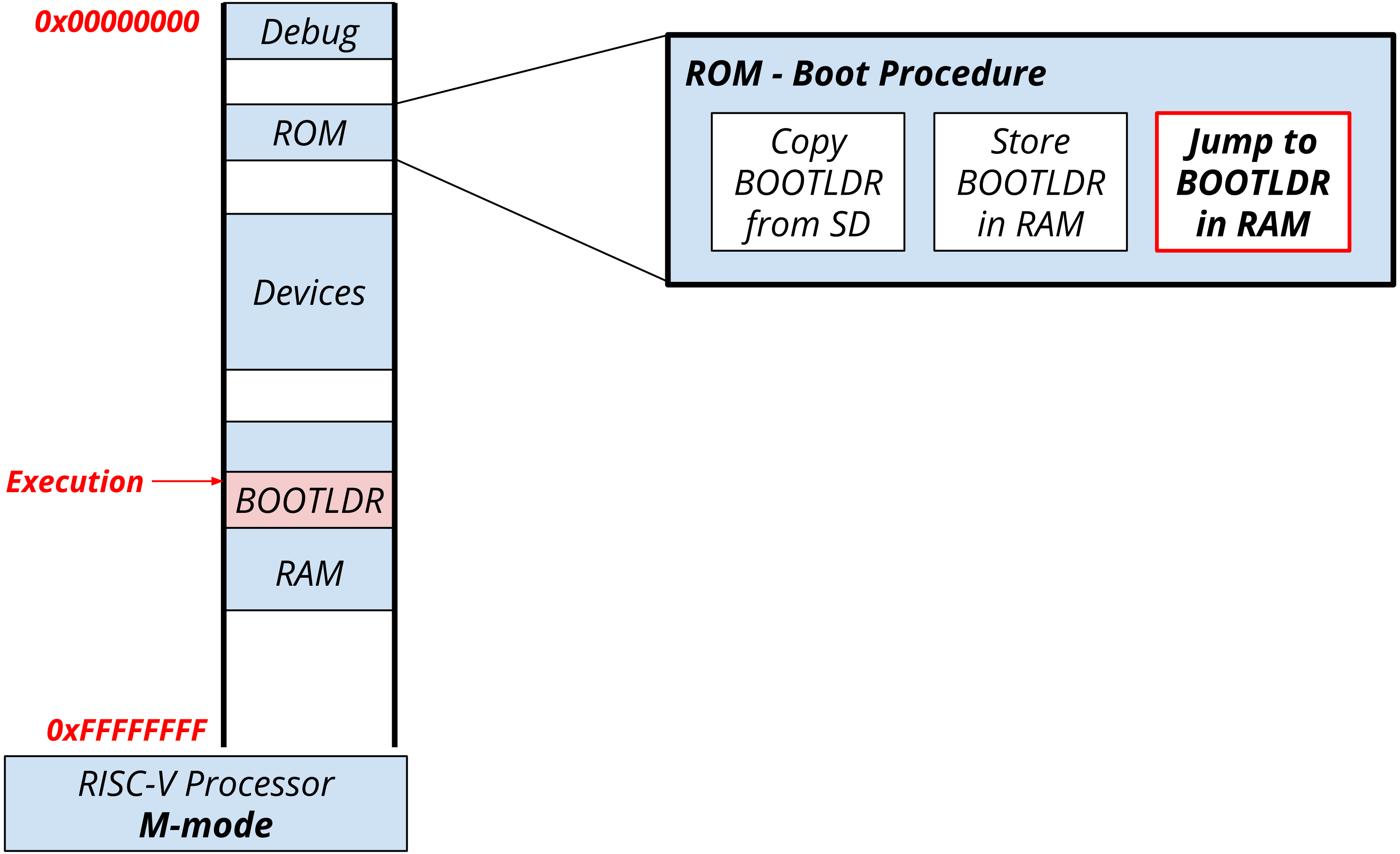
RISC-V Processor Privilege Modes



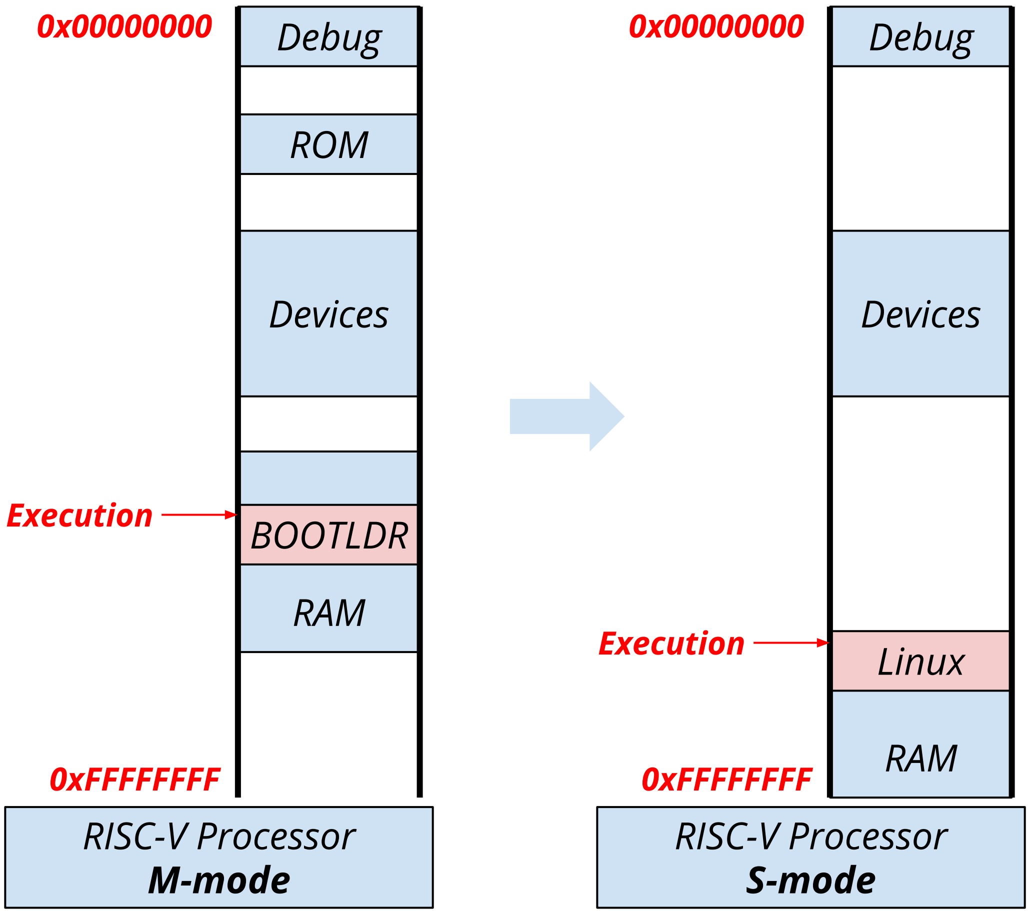
RISC-V Processor Privilege Modes



RISC-V Processor Privilege Modes

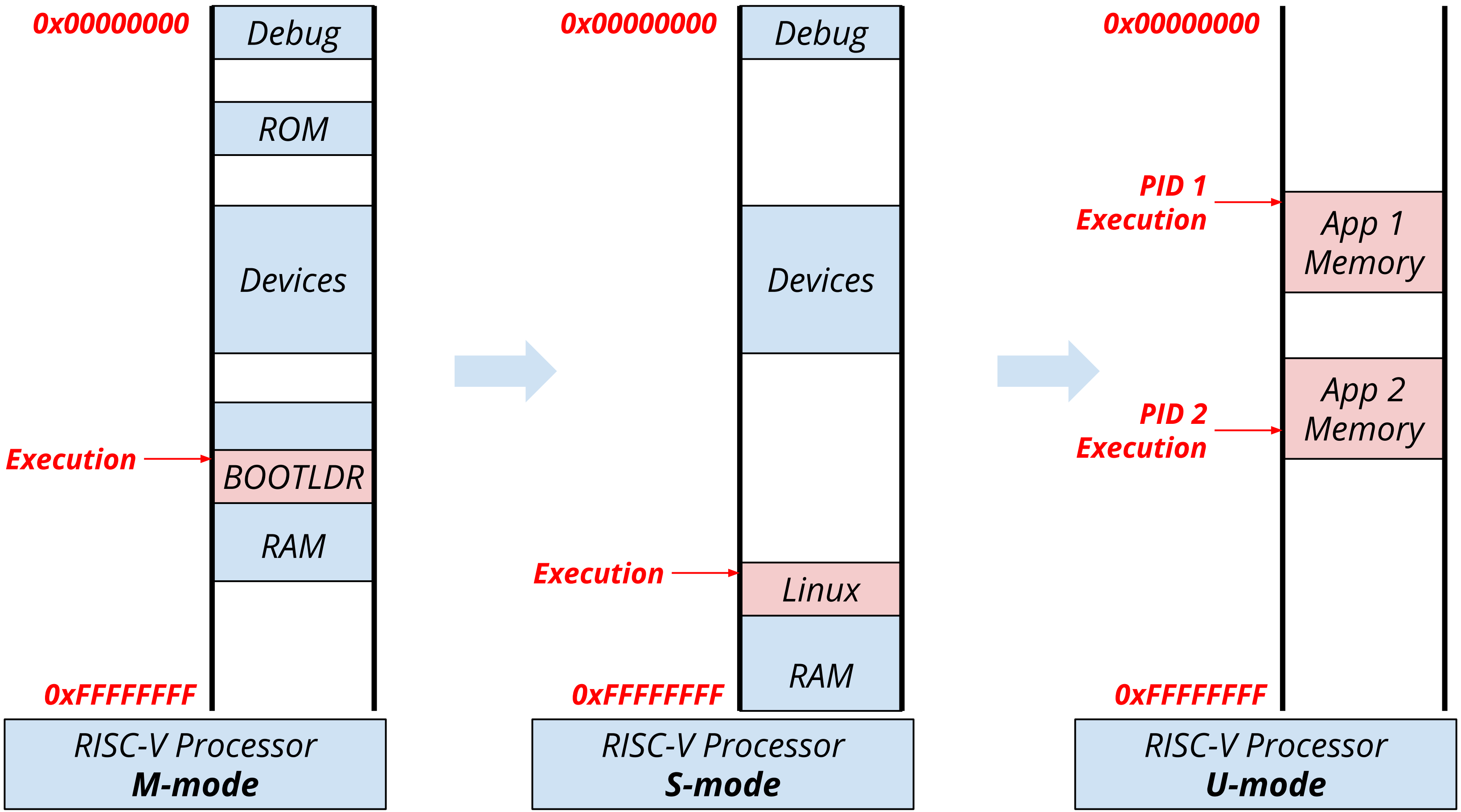


RISC-V Processor Privilege Modes



The bootloader extracts Linux and executes it in **Supervisor-Mode**

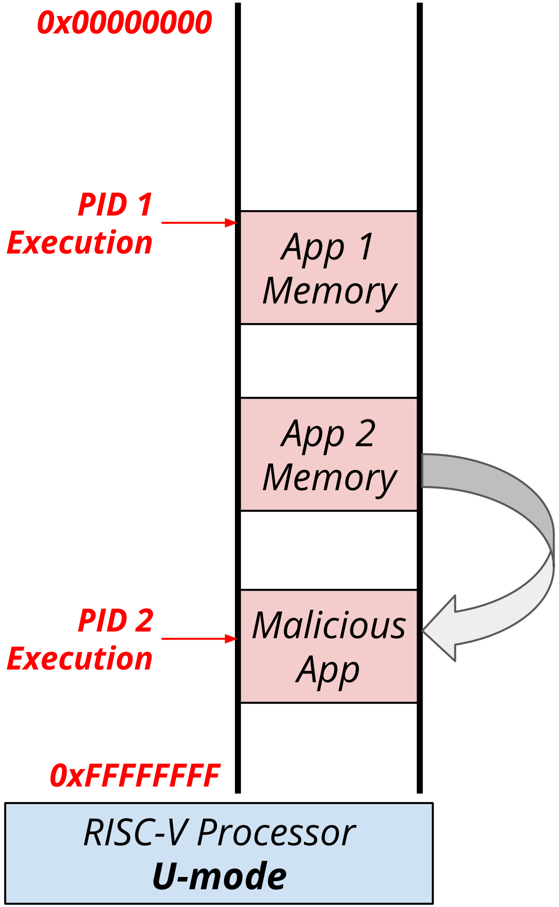
RISC-V Processor Privilege Modes



Non-Protected Applications

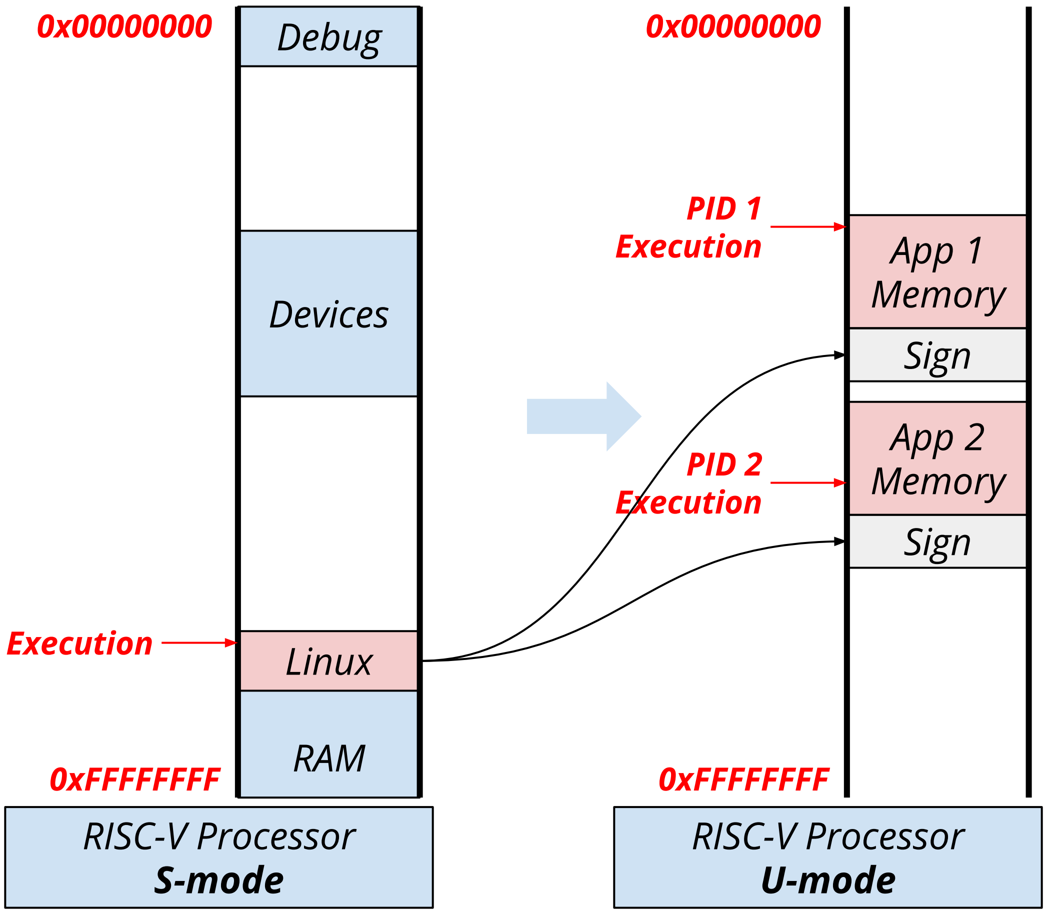
Malicious applications can access and execute code arbitrarily. Some attacks are:

- Cache manipulation
- Privilege mode escalation
- Controlled power glitches



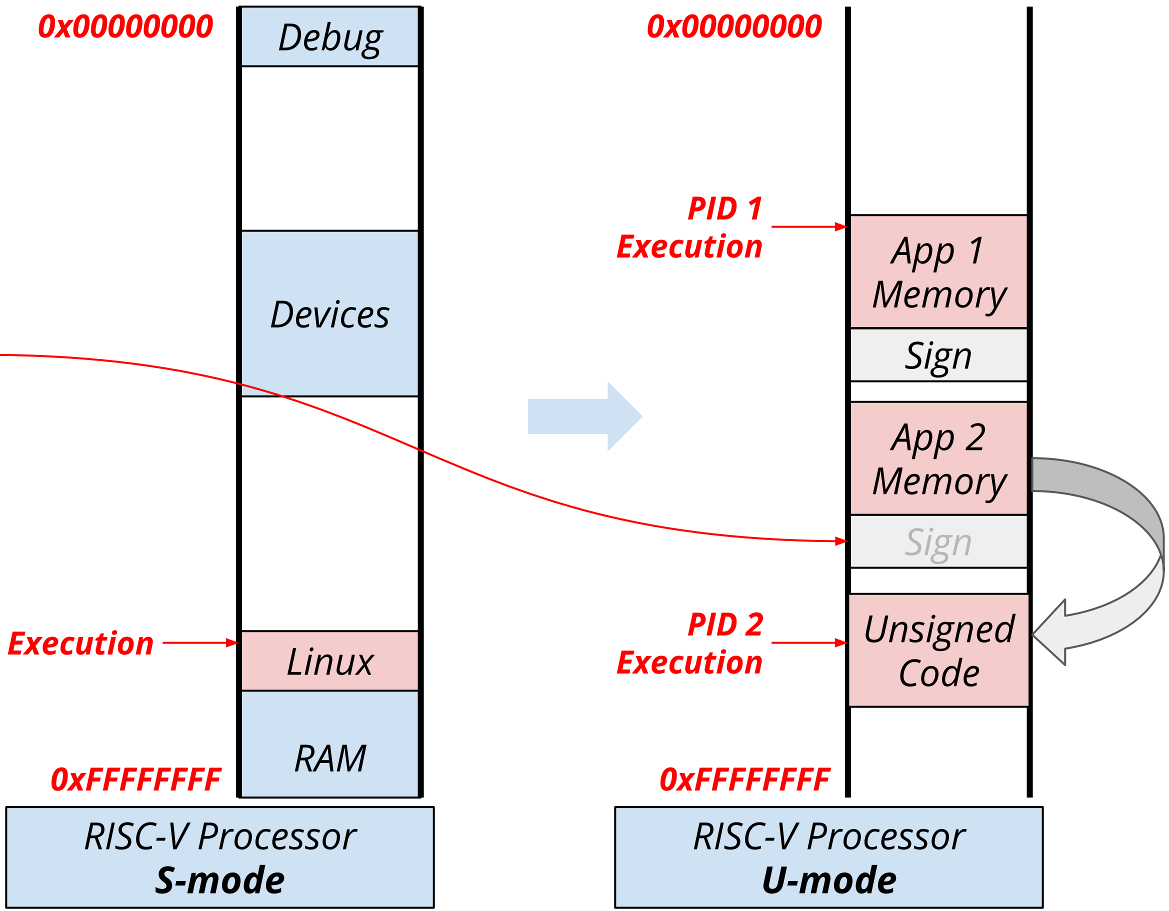
Making a Secure Environment

Linux only executes the application if the signature is authenticated.

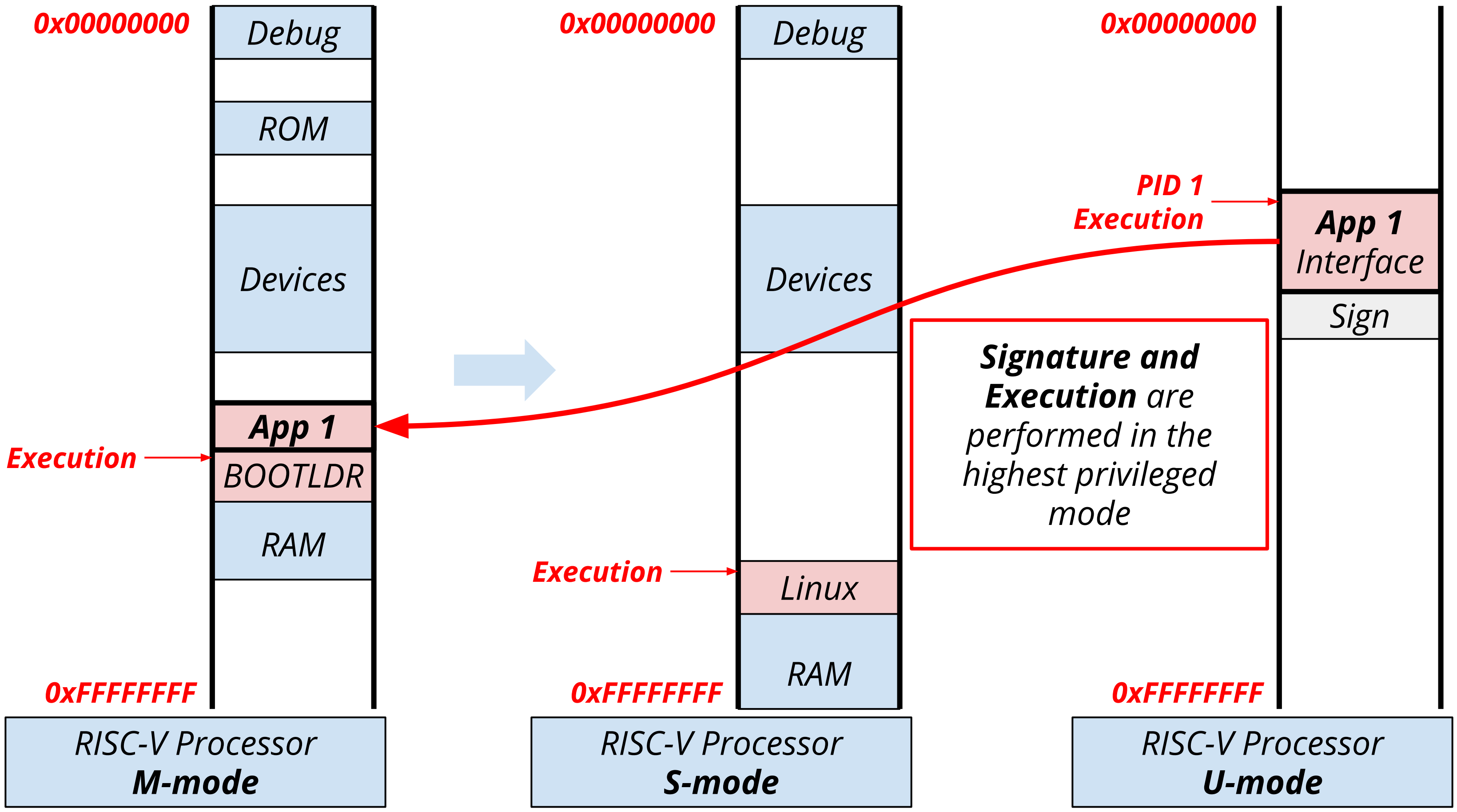


Making a Secure Environment

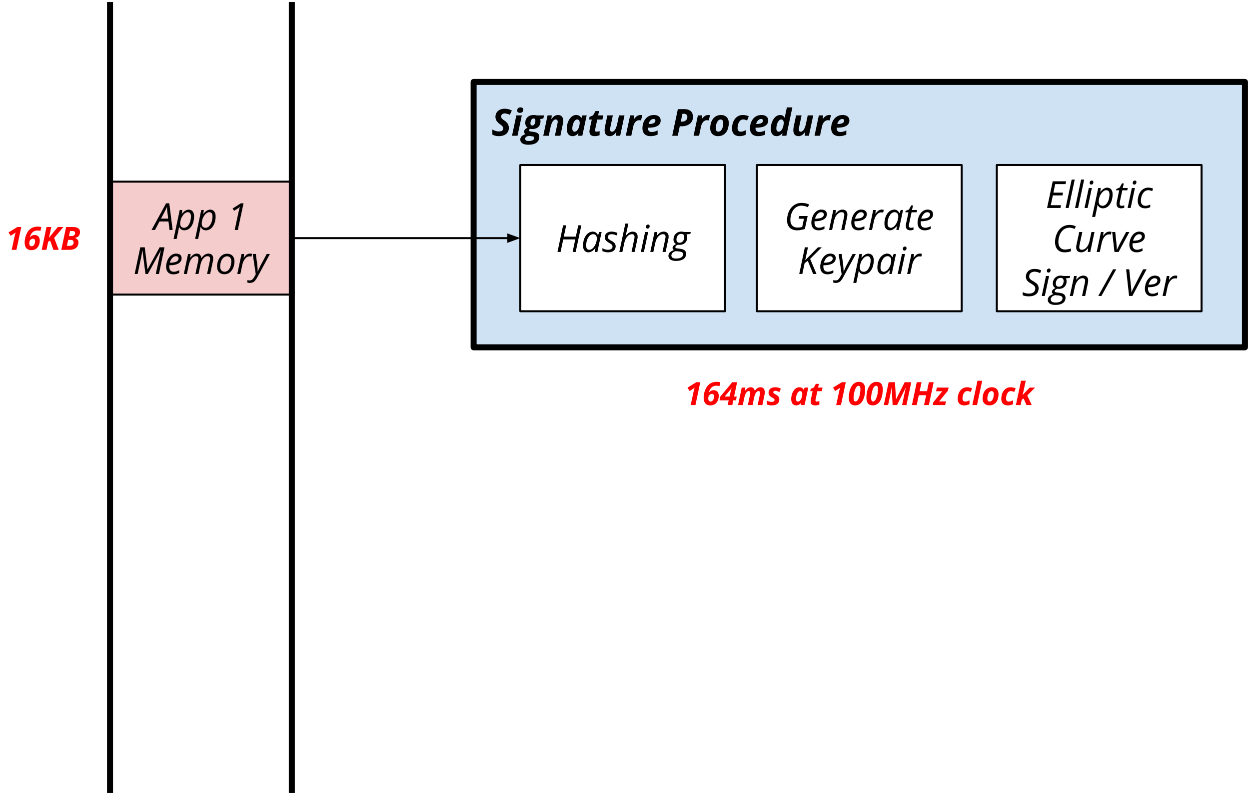
Once the signature verification is performed, the ***attack*** can rewrite the instructions of any application to execute ***unsigned code***.



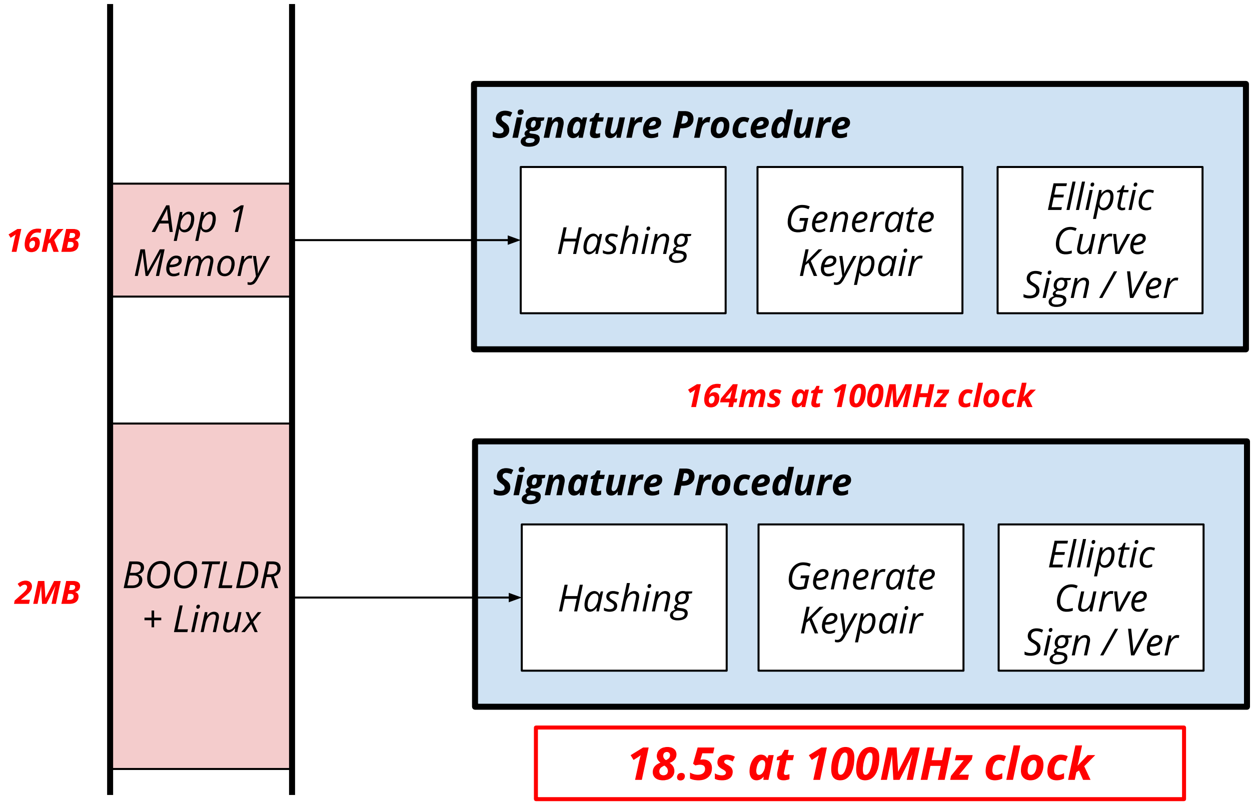
Making the Trusted Execution Environment



RISC-V Lack of Crypto-Hardware



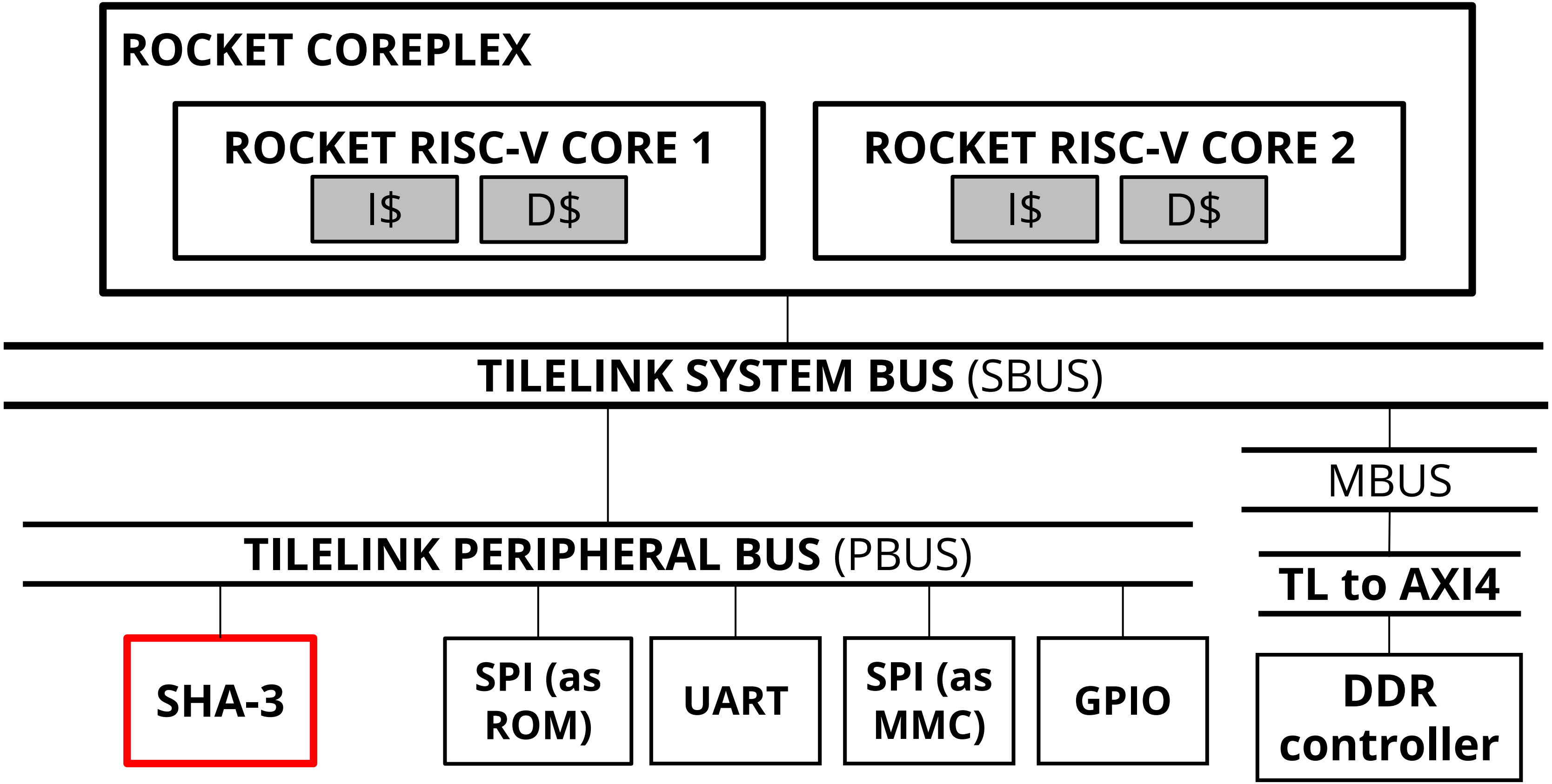
RISC-V Lack of Crypto-Hardware



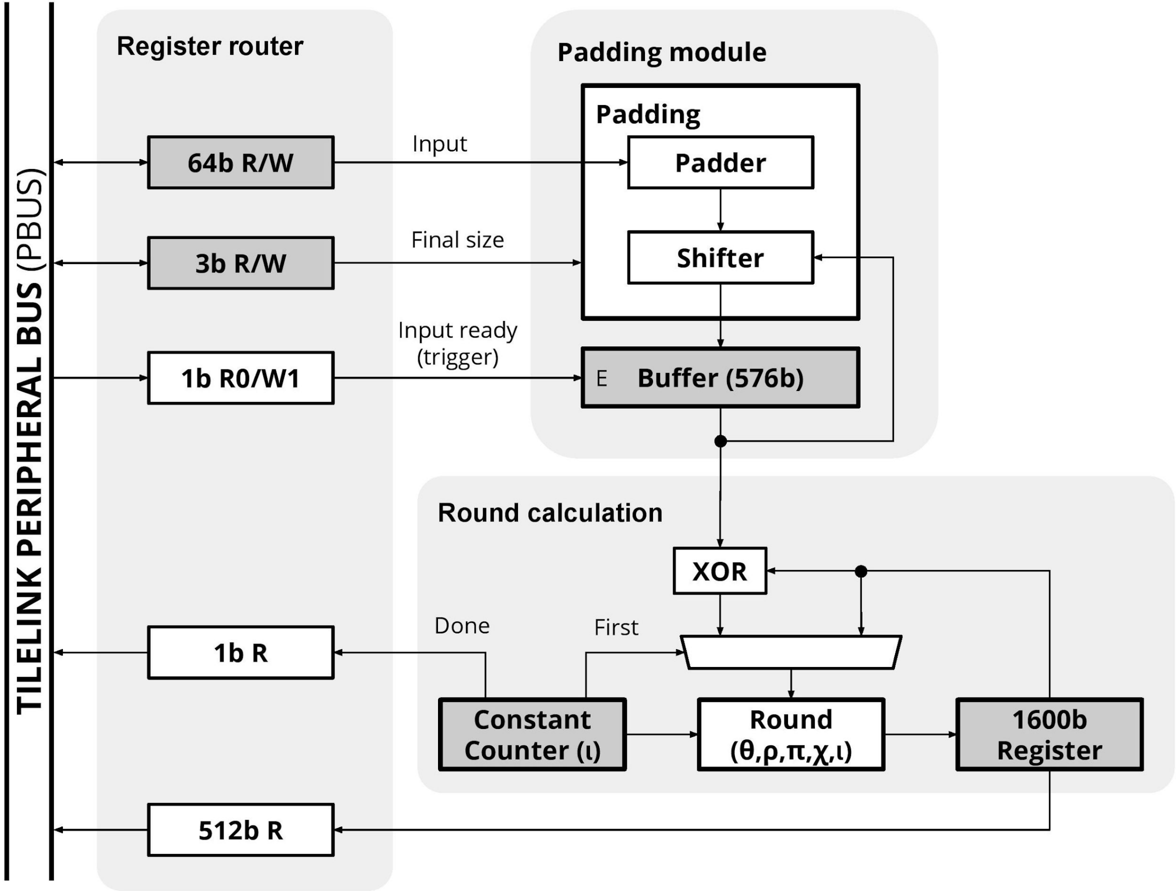
Outline

- Motivation
- Hardware Structure for Trusted Execution Environment
- Boot Procedure with Crypto-accelerators
- Implementation Results
- Conclusions

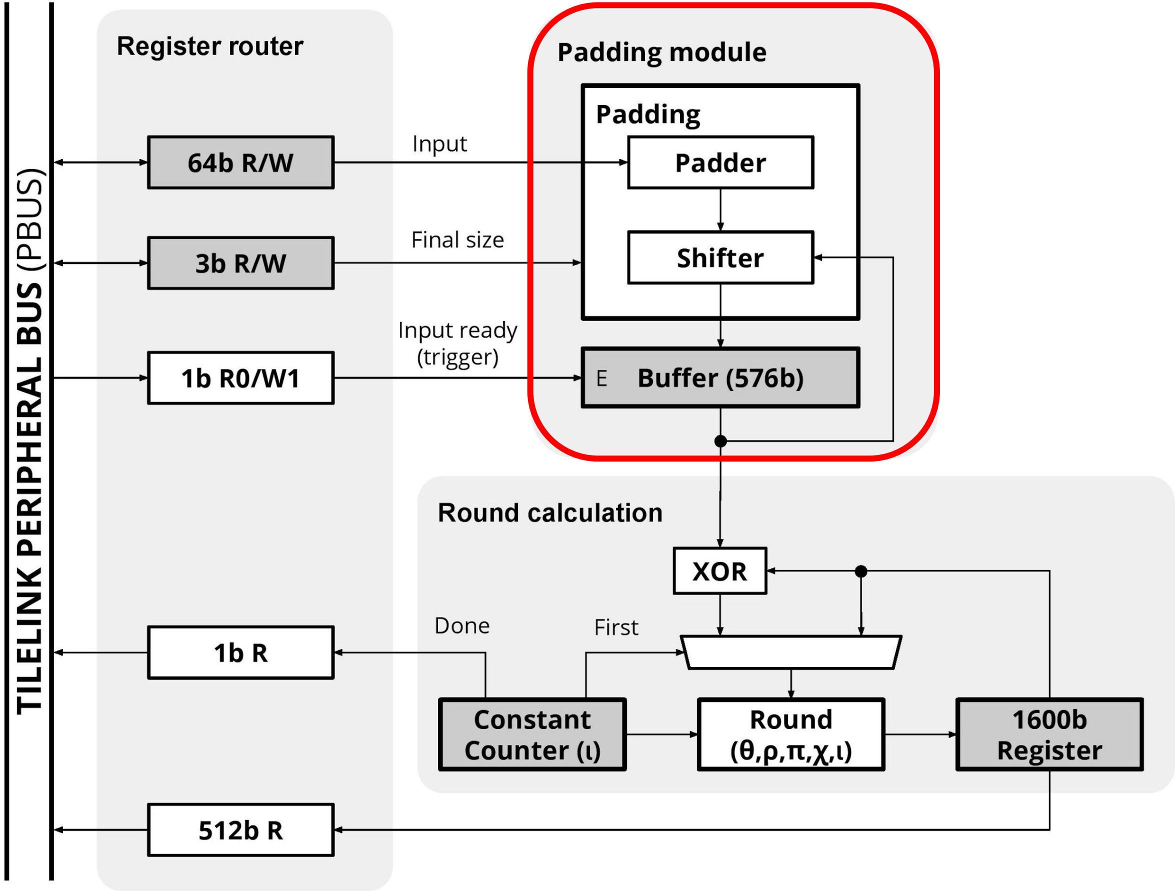
SoC Architecture



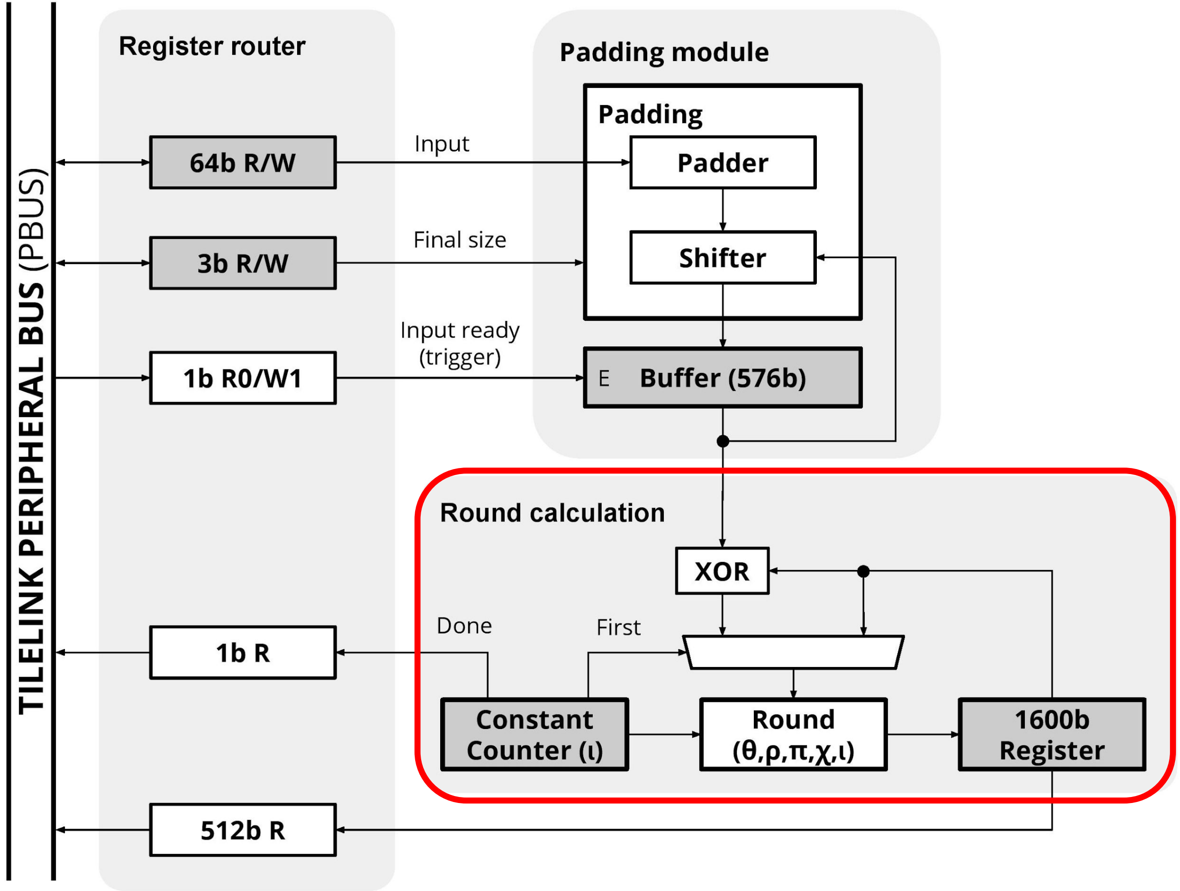
SHA-3 Device Architecture



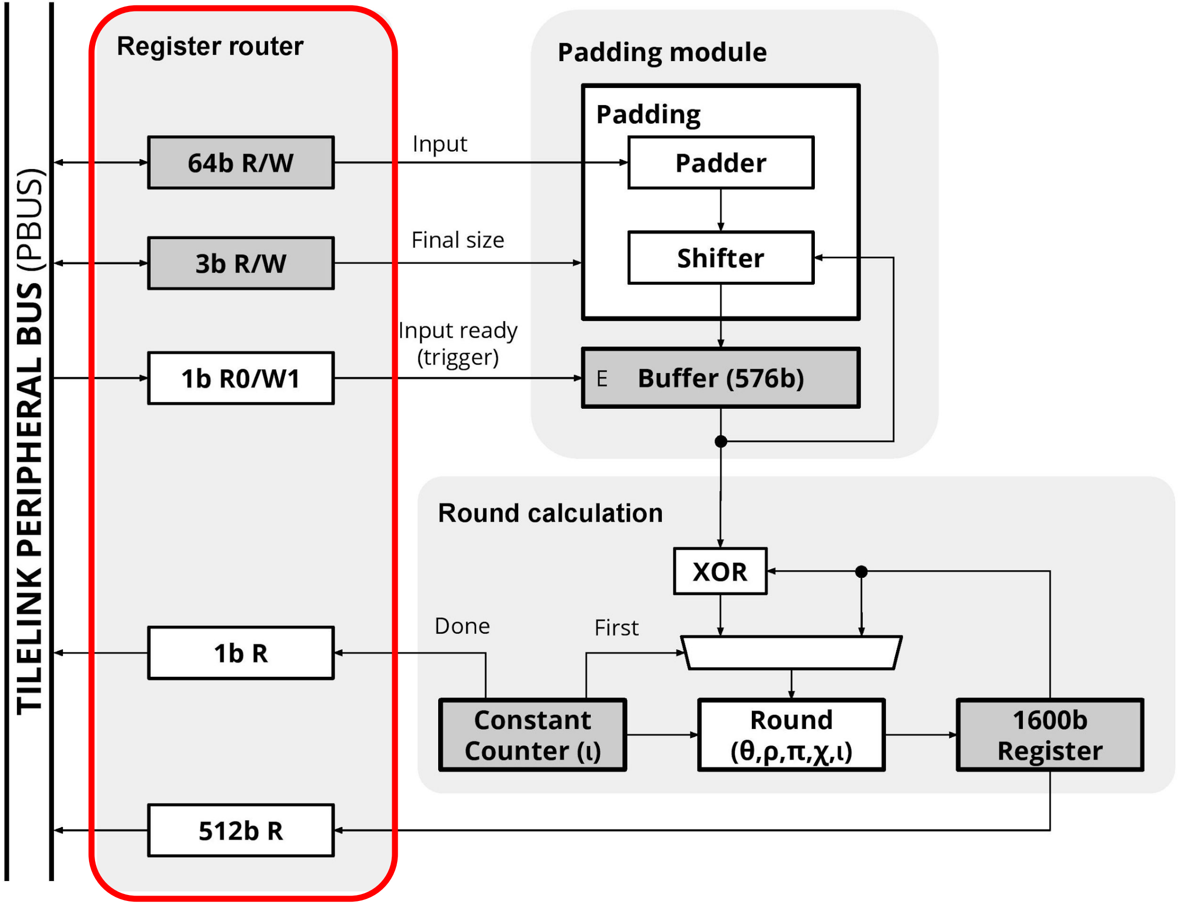
SHA-3 Device Architecture



SHA-3 Device Architecture



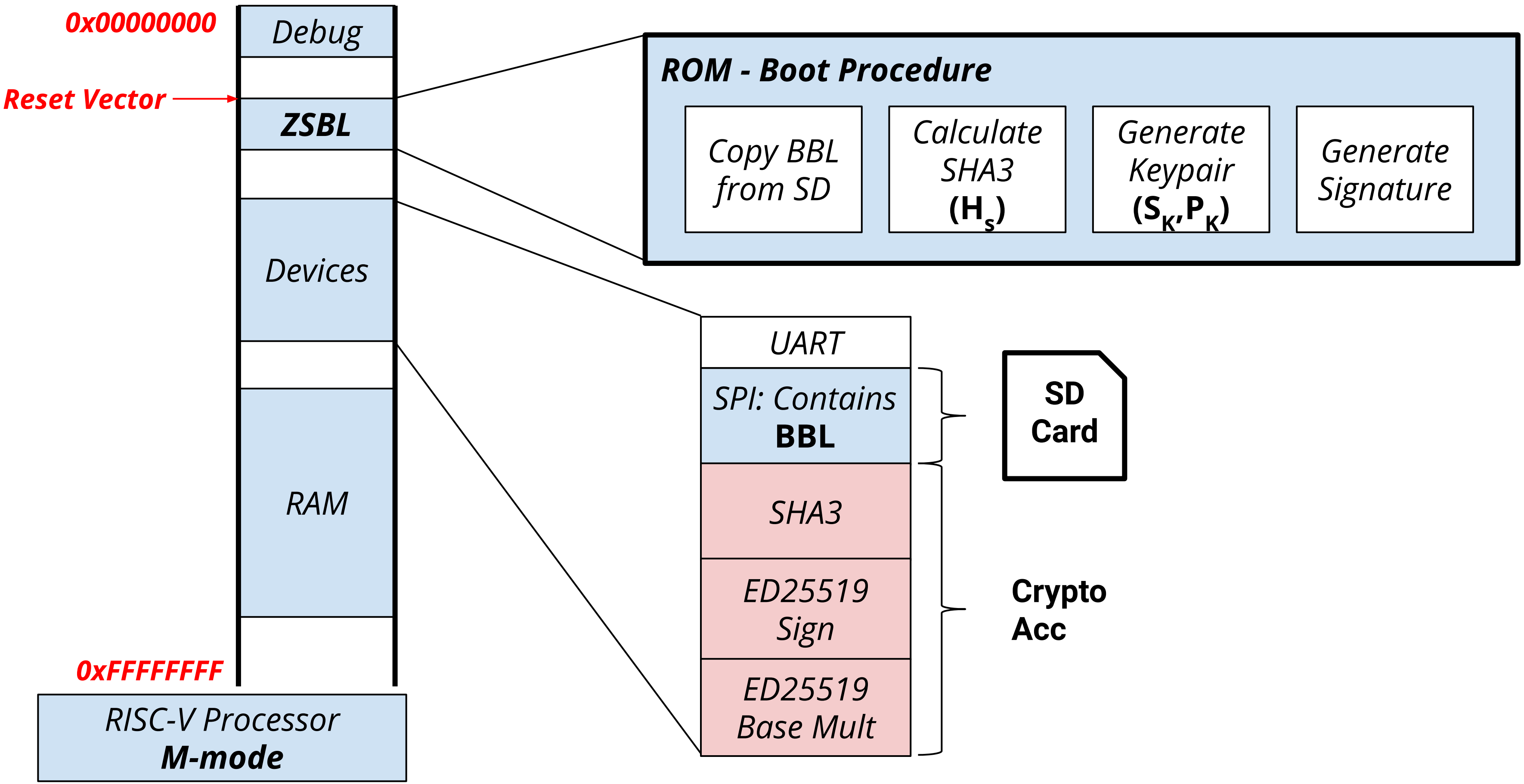
SHA-3 Device Architecture



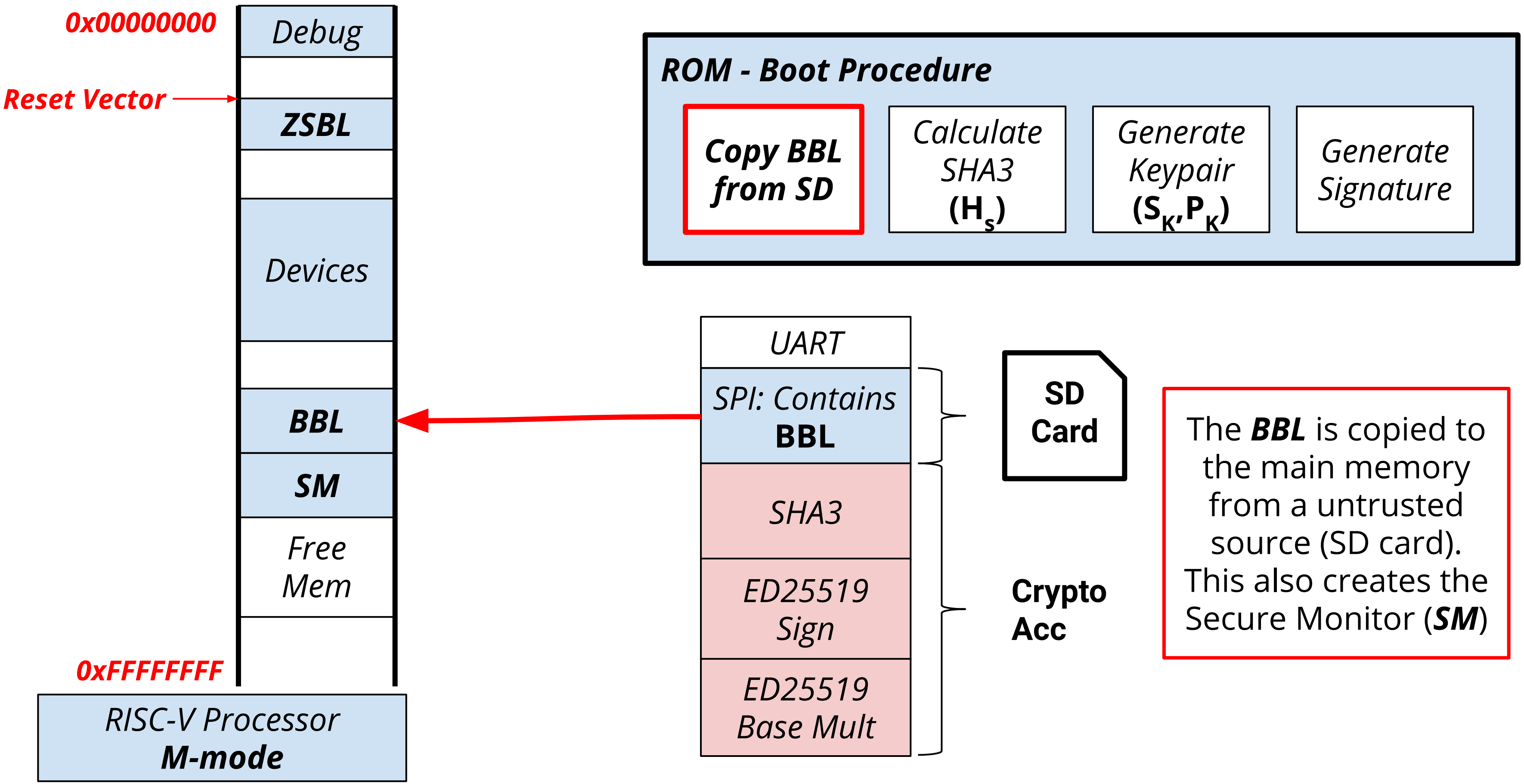
Outline

- Motivation
- Hardware Structure for Trusted Execution Environments
- Boot Procedure with Crypto-accelerators
- Implementation Results
- Conclusions

SoC Memory Map

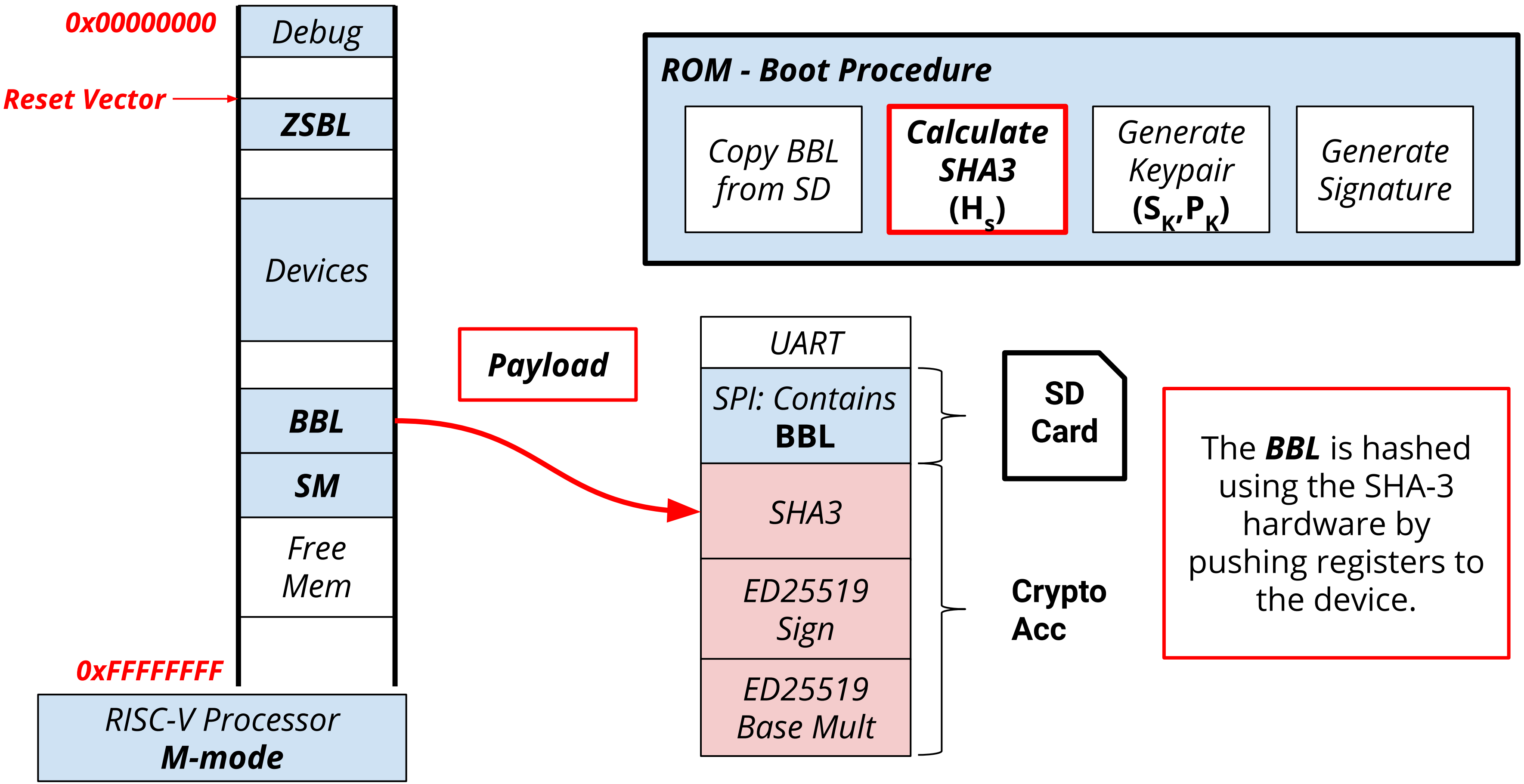


Boot Procedure

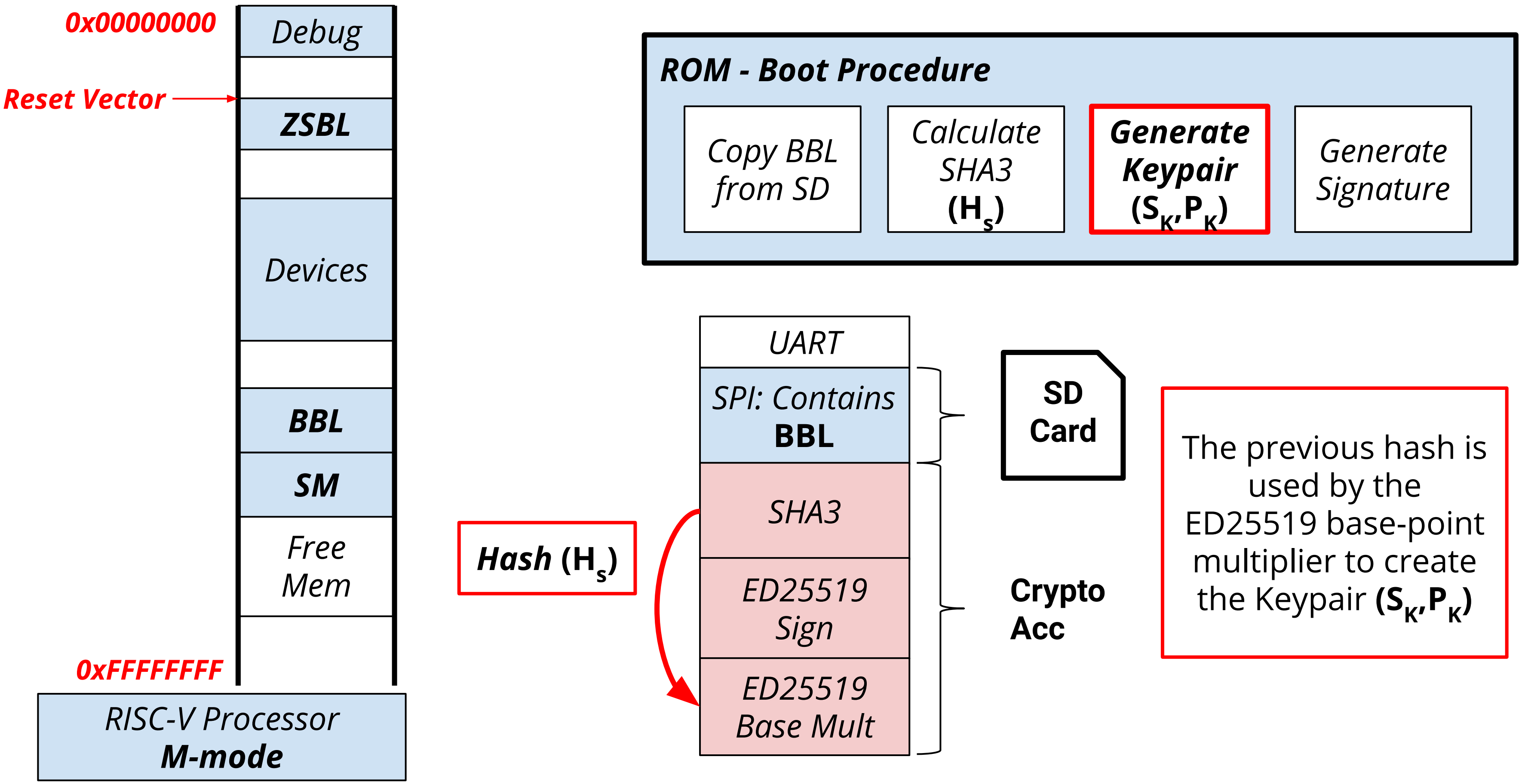


The ***BBL*** is copied to the main memory from a untrusted source (SD card). This also creates the Secure Monitor (***SM***)

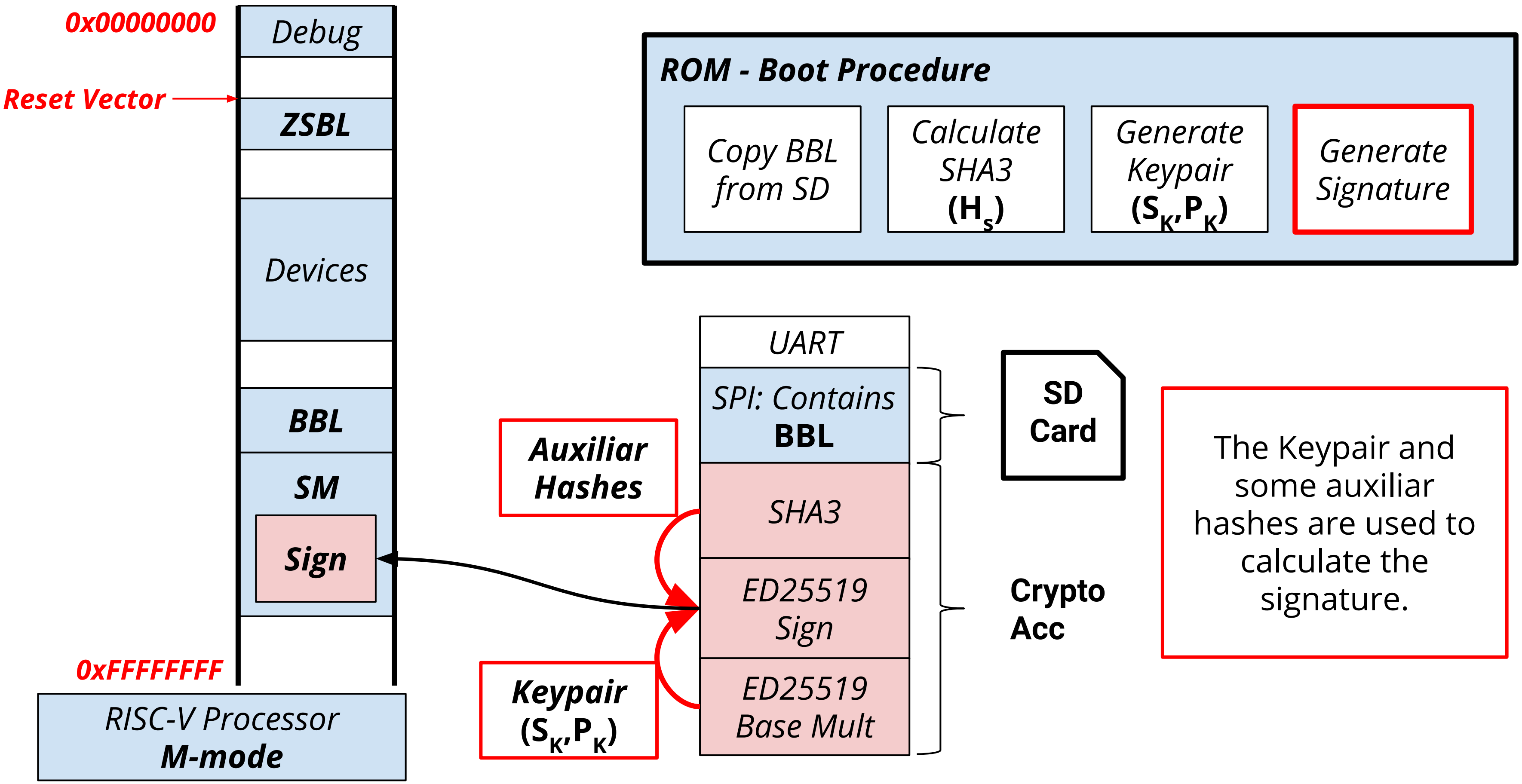
Boot Procedure



Boot Procedure



Boot Procedure



Outline

- Motivation
- Hardware Structure for Trusted Execution Environments
- Boot Procedure with Crypto-accelerators
- Implementation Results
- Conclusions

Implementation Results

Table 1: Synthesis result on Stratix-IV GX Altera FPGA.

	SHA-3	**RocketTile**
ALUTs	8108	24332
FFs	2790	15325
RAM Bits	0	17680
DSP	0	32
Total	10898	57369
Logic Utilization	3.4%	12.4%
RAM Utilization	0%	1%
DSP Utilization	0%	2.4%

Implementation Results

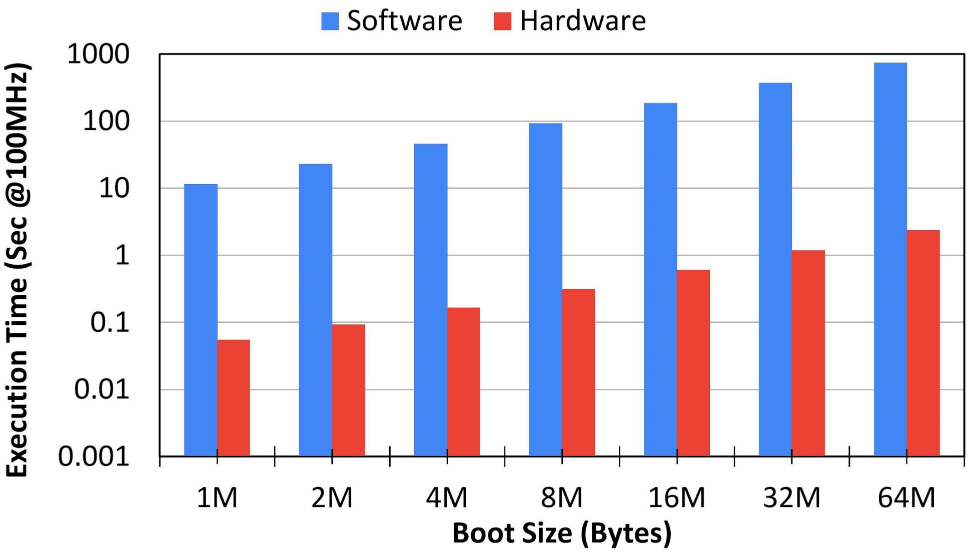


Figure 1: Comparison between software and hardware with different bootloader sizes.

Table 2: Execution results for Ed25519 task.

2MB Bootloader	Software	HW SHA-3 with SW Ed25519
Ed25519 keypair (ms)	109.5	93.4
Ed25519 signature (ms)	231019	82.6

Outline

- Motivation
- Hardware Structure for Trusted Execution Environments
- Boot Procedure with Crypto-accelerators
- Implementation Results
- Conclusions

Conclusions

- We presented a system platform for trusted execution environments (TEEs) featuring the SHA-3 accelerator.
- ISC-V core with RV64IMAFDC ISA using the Rocket chip generator.
- The SHA-3 accelerator hashes data using a 64-bit register as input.
- The software authenticates the bootloader and utilizes the accelerators.
- The execution time drops significantly compared to software.

Questions?