A Holistic Hardware/Software Update Mechanism for Embedded Systems

Leandro Batista Ribeiro
lbatistaribeiro@tugraz.at

Tobias Scheipel
tobias.scheipel@tugraz.at

Marcel Baunach
baunach@tugraz.at

Institute of Technical Informatics
Embedded Automotive Systems Group
Graz University of Technology

March 11th, 2021

Outline

- Introduction and Motivation
- Update Mechanisms
  - Dynamic Software Update
  - Partial Logic Reconfiguration
- Holistic Update Mechanism
- Conclusion

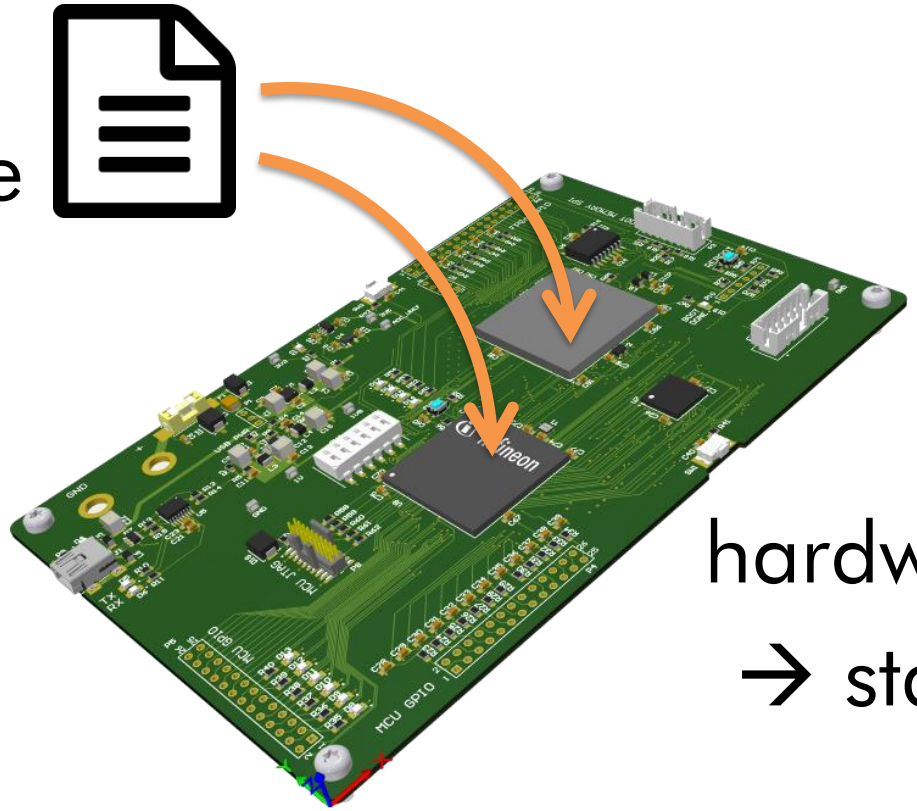
Outline

- Introduction and Motivation
- Update Mechanisms
  - Dynamic Software Update
  - Partial Logic Reconfiguration
- Holistic Update Mechanism
- Conclusion

Introduction and Motivation

software

→ flexible



hardware

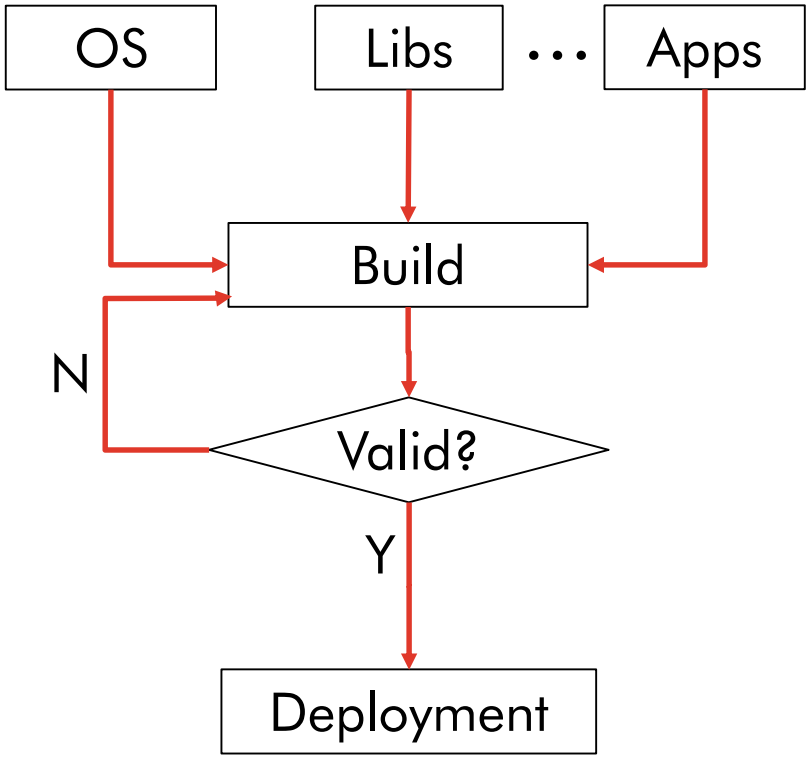
→ static

Introduction and Motivation

Handling Software Issues

- Build new version
- Test / Validate / Verify
- Transmit
  - Wired or Wireless

- Apply
  - Full image replacement
  - Partial update



What if the hardware is buggy / deprecated?

Introduction and Motivation

Known Hardware issues

- State cases where HW got deprecated / buggy
  - 2011 – SHA deprecation
    - HW accelerators in CPUs became useless
  - 2011 – Intel Sandy Bridge
    - Faulty chipsets [intro1]
  - 2018 – Meltdown/Spectre
    - Security vulnerabilities in Intel CPUs [intro2]

Introduction and Motivation
Handling Hardware Issues

- Microcode update
  - Not usual or never happens in the embedded world
- Software update
  - SW workaround
    - Execution overhead
  - Software simulation
    - Even more execution overhead
- Recall
  - Expensive and inconvenient

If only we could handle HW like we handle SW …

Outline

- Introduction and Motivation
- Update Mechanisms
  - Dynamic Software Update
  - Partial Logic Reconfiguration
- Holistic Update Mechanism
- Conclusion

Update Mechanisms

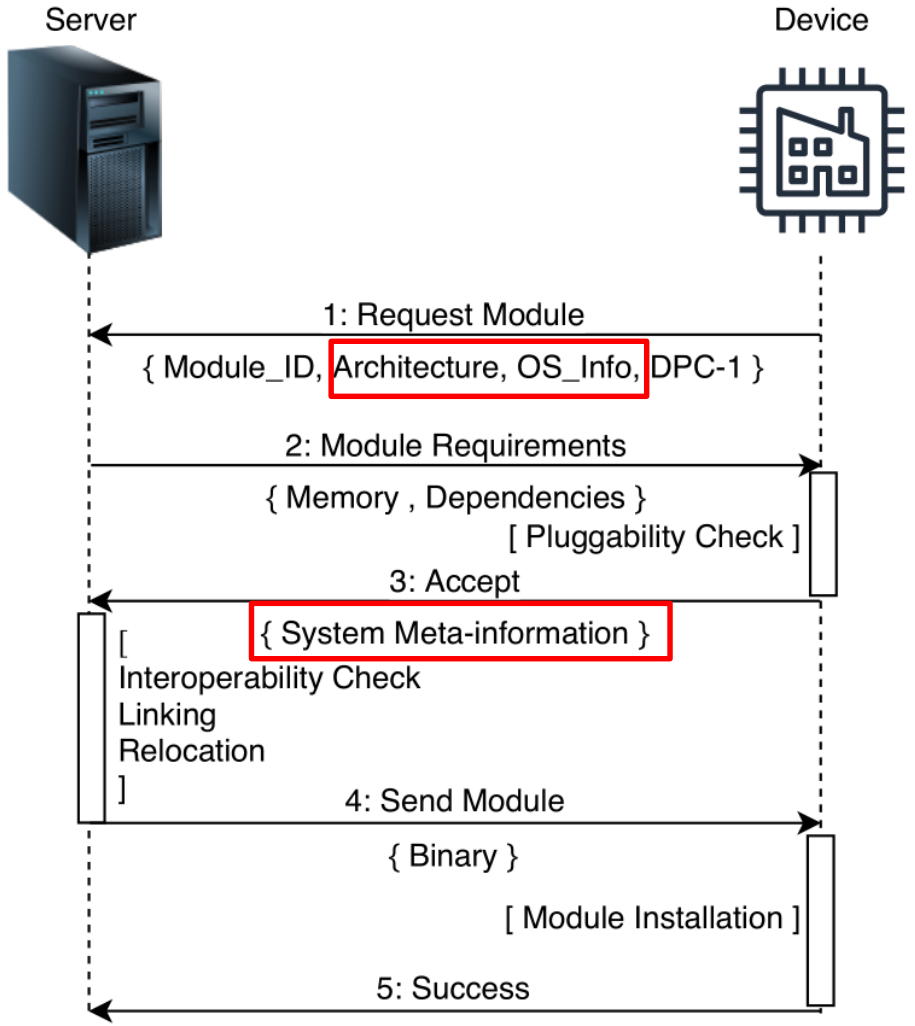
Dynamic Software Update - Overview

- MCSmartOS supports dynamic updates

MCSmartOS

Multi-Core Sustainable modular adept real-time Operating System

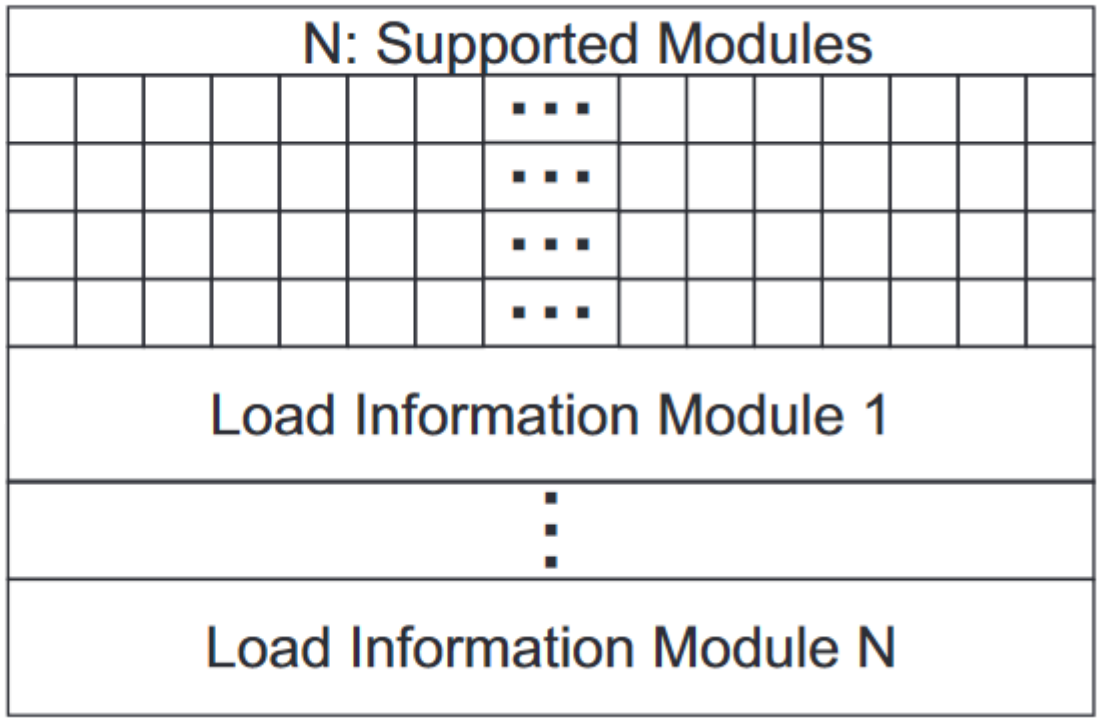
- Update Protocol covers HW/SW diversity [sw1]



Update Mechanisms

Dynamic Software Update – Inherent Management Overhead

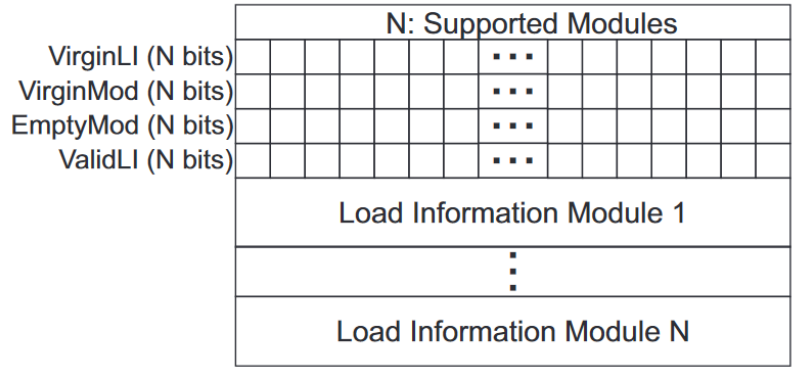
- Modules management
- Fault-tolerance on module updates

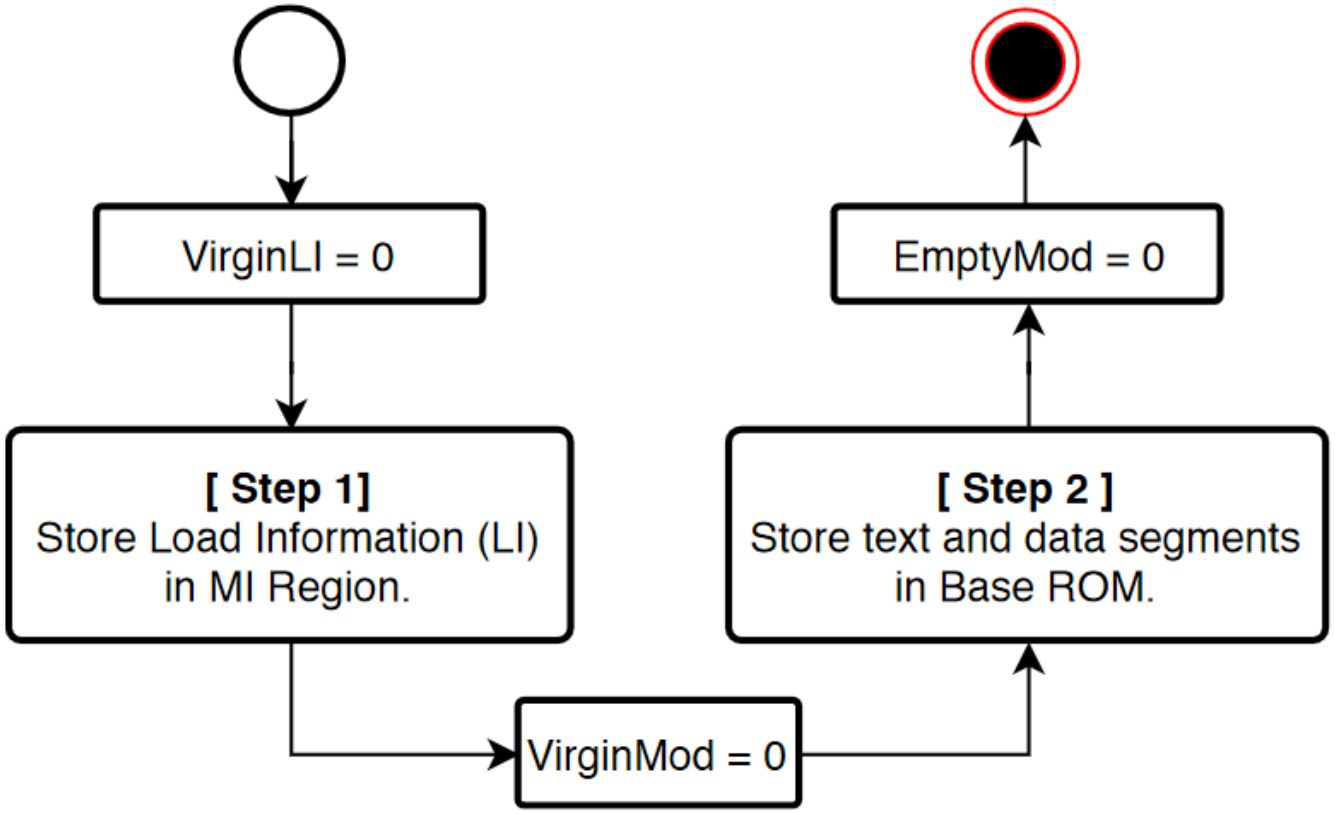


Update Mechanisms

Dynamic Software Update – Inherent Management Overhead

- Fault-tolerance on module updates

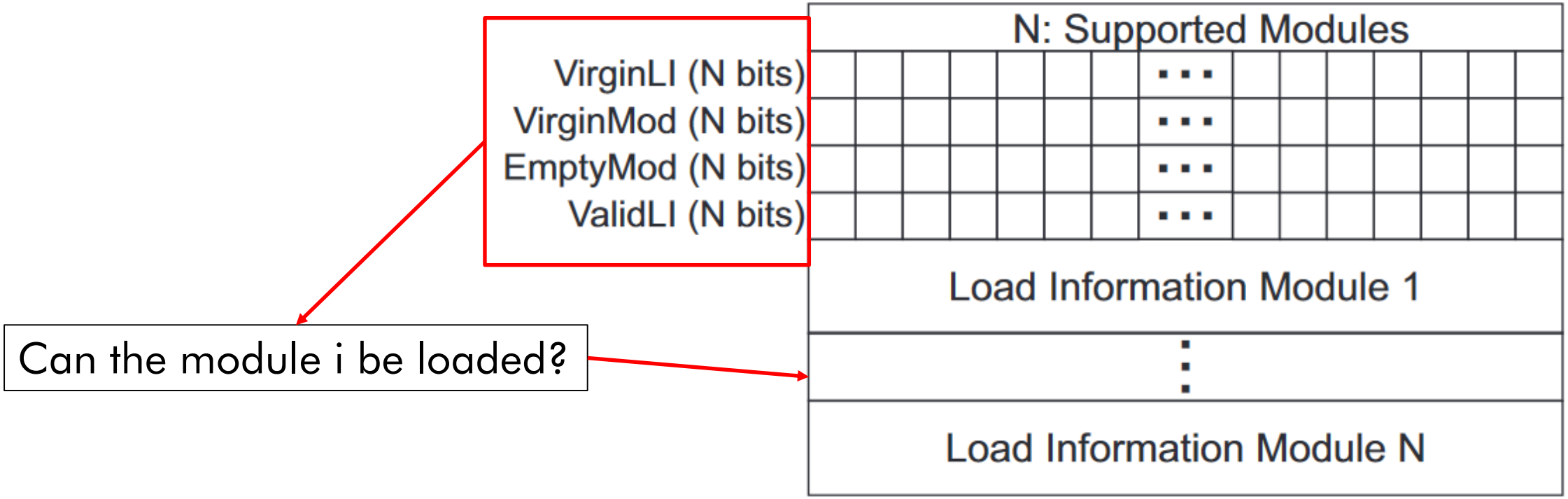




Update Mechanisms

Dynamic Software Update – Inherent Management Overhead

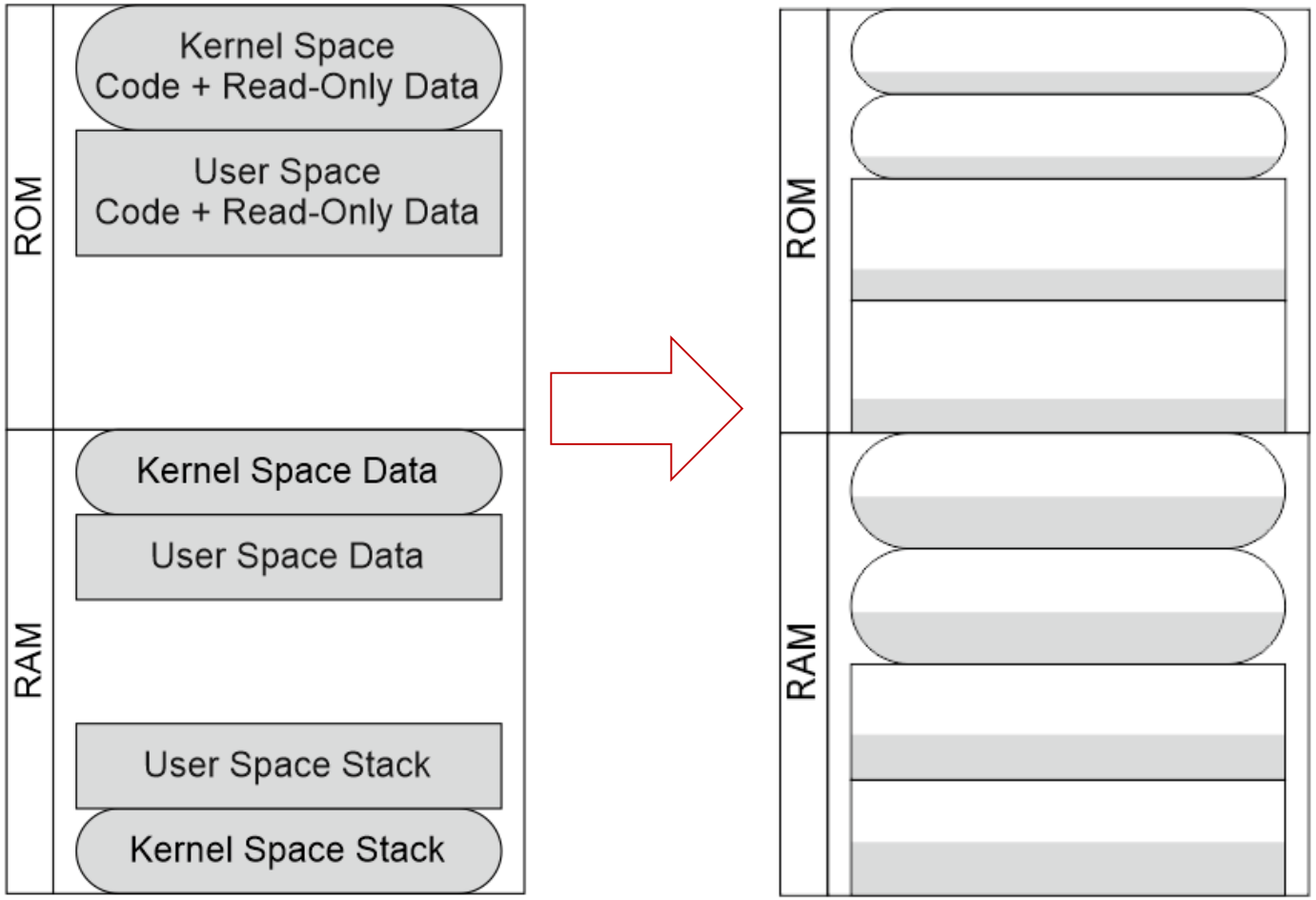
- System Startup – Load Modules

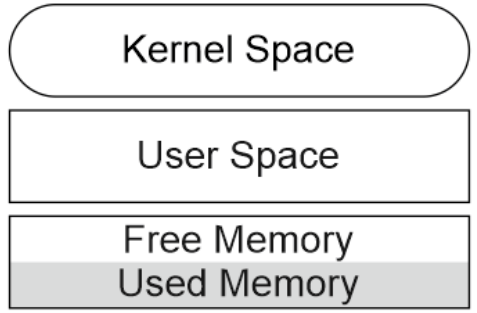


Update Mechanisms

Dynamic Software Update – Inherent Management Overhead

- Memory Management

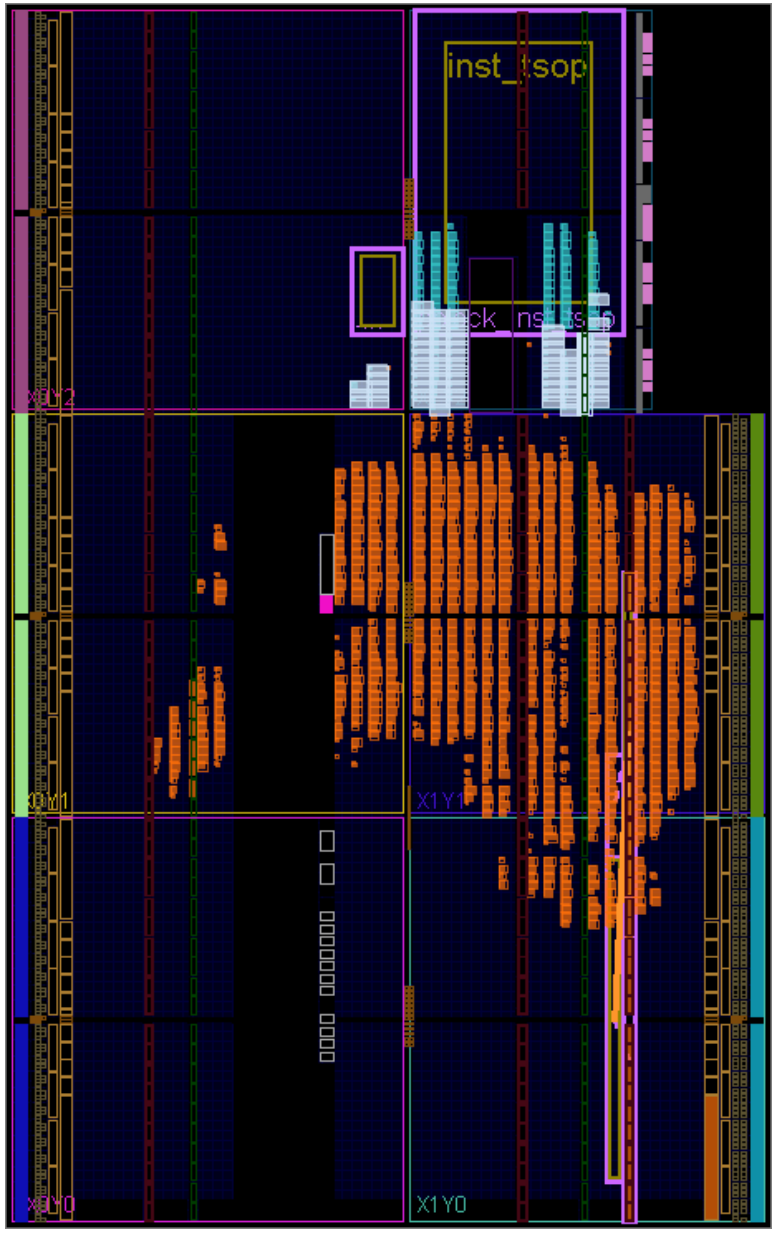




Update Mechanisms
Partial Logic Reconfiguration

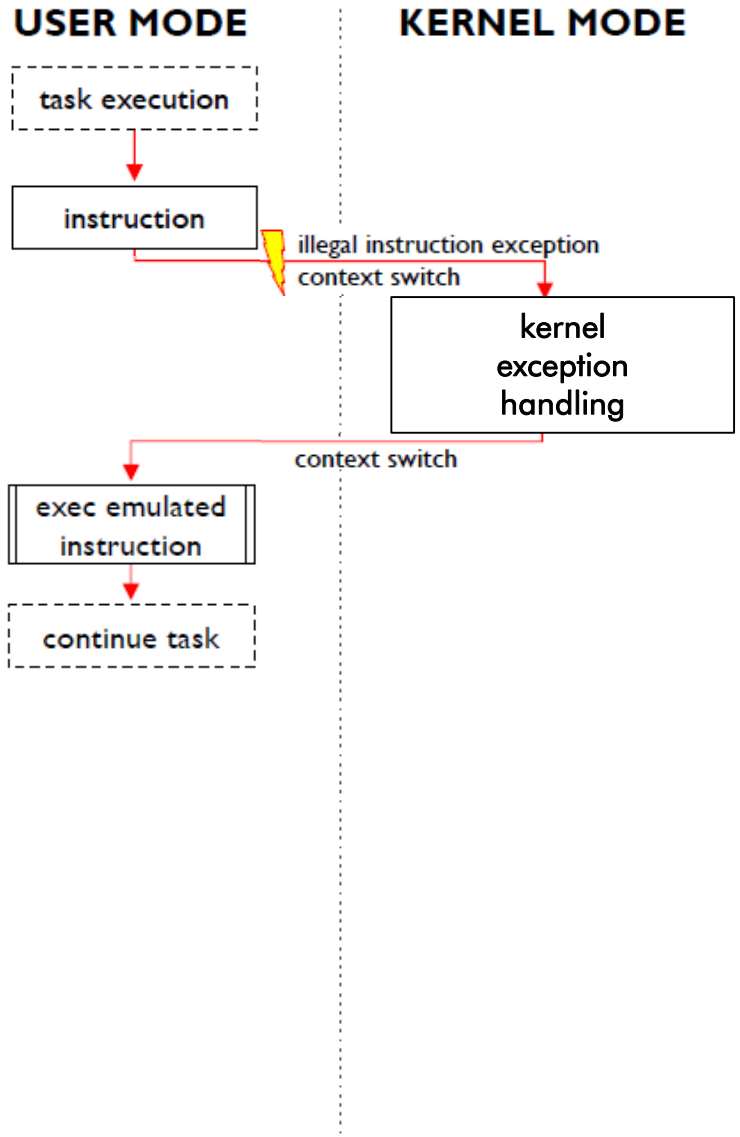
- Given: An embedded System based on an
  - FPGA board [hw1] running a
  - RISC-V MCU [hw2] and an
  - MCSmartOS environment.

- Objective: Support changing ISAs within the OS and the MCU including a
  - consistent API for development,
  - OS support,
  - MCU support, and
  - dynamic reconfiguration at runtime.



Update Mechanisms
Partial Logic Reconfiguration

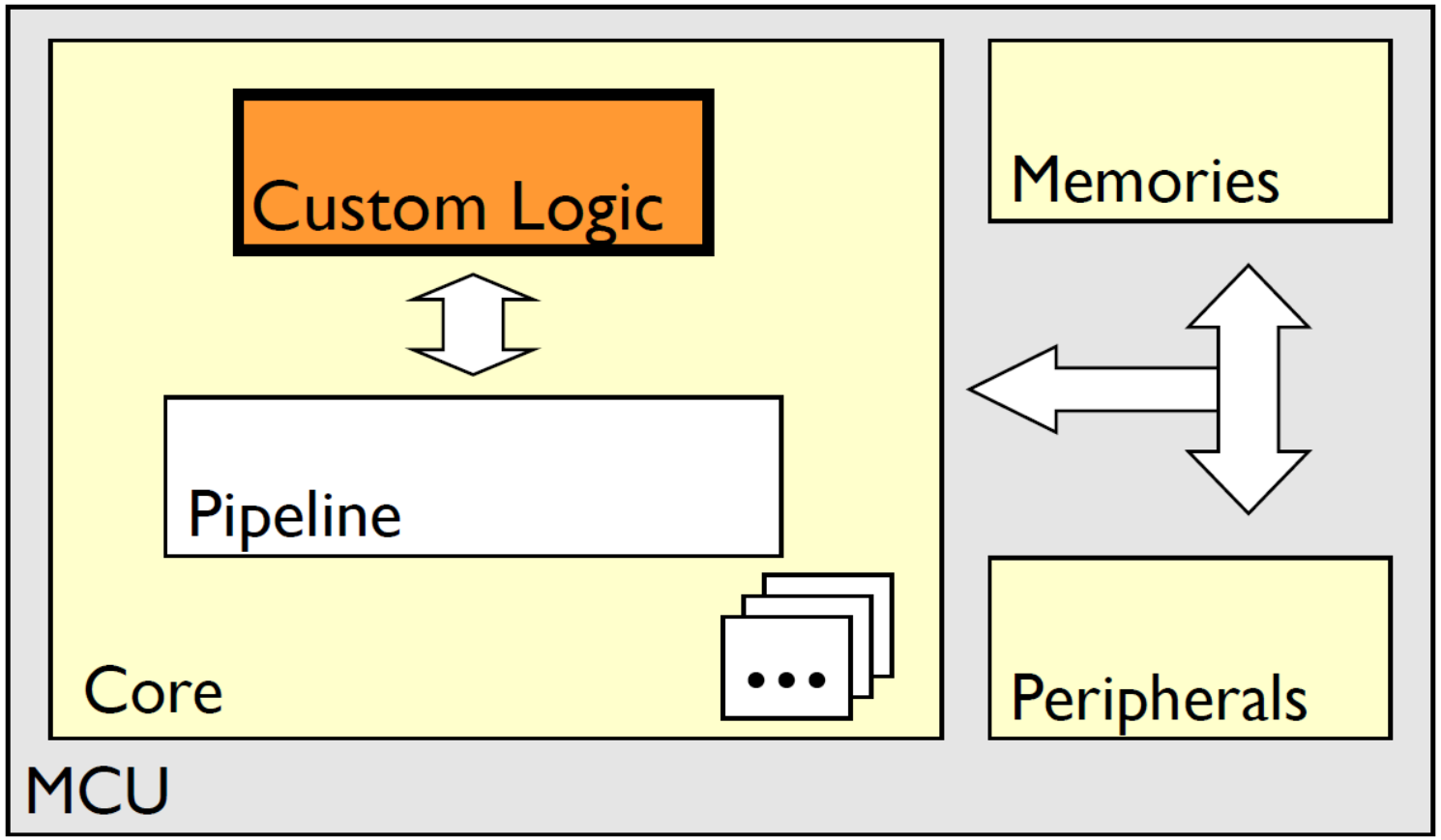
```
100  [...]
101  addi  t4, zero, 12
102  cinsi t3, t4,   10
103  [...]
```



Two Scenarios:

1. CPU knows `cinsi`
2. CPU doesn't know `cinsi`

Update Mechanisms
Partial Logic Reconfiguration



Outline

- Introduction and Motivation
- Update Mechanisms
  - Dynamic Software Update
  - Partial Logic Reconfiguration
- Holistic Update Mechanism
- Conclusion

Holistic Update Mechanism

The Concept

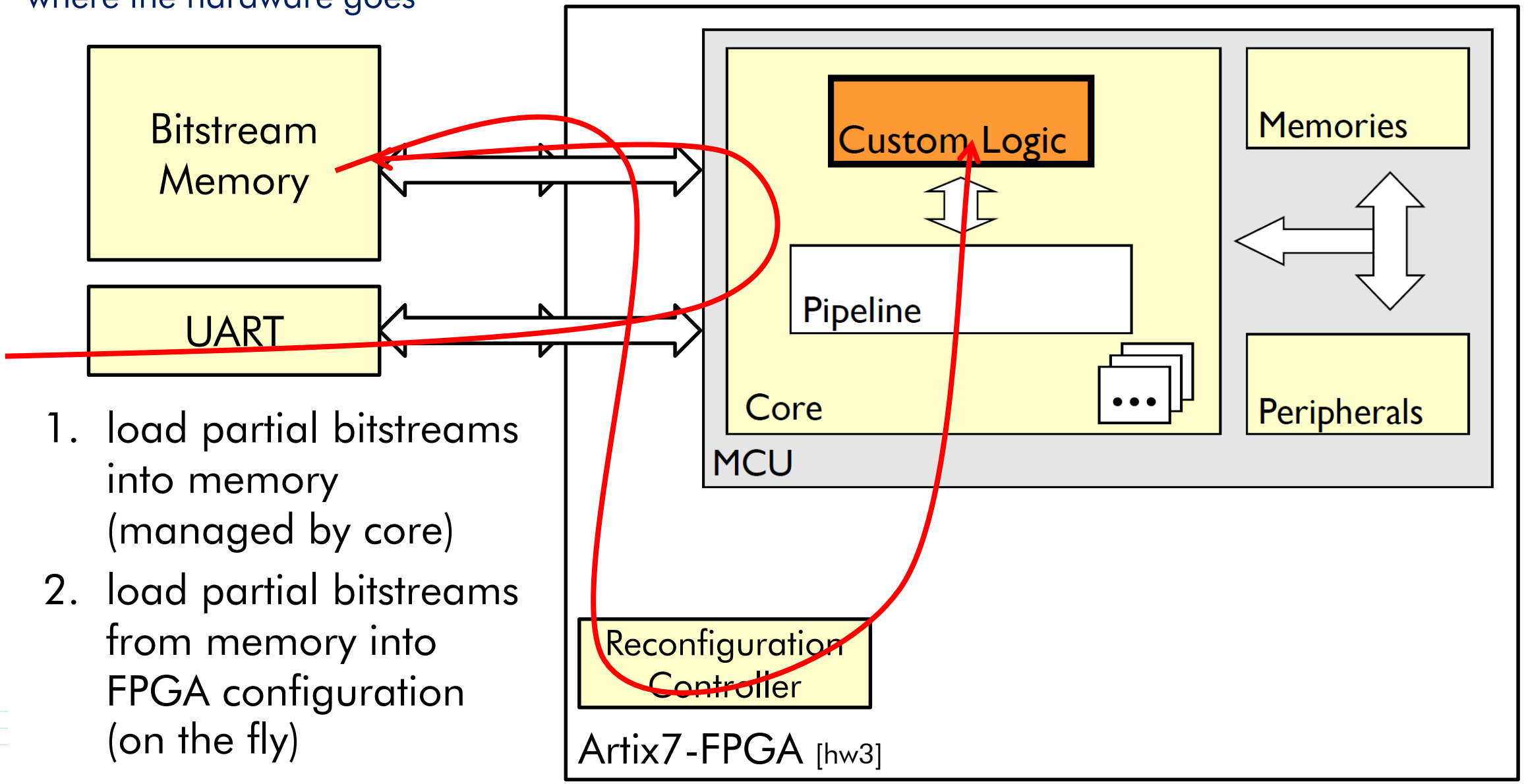
- Rebootless process

- Transmit partial bitstream
- MCSmartOS stores it in the memory
  - Whenever a hardware variant is needed, it is loaded and forwarded to the FPGA
- FPGA performs partial reconfiguration

- MCSmartOS assumes “HW Modules” management

Holistic Update Mechanism

"where the hardware goes"



1. load partial bitstreams into memory (managed by core)
2. load partial bitstreams from memory into FPGA configuration (on the fly)

Holistic Update Mechanism
Possibilities

- "Softer" hardware
  - Easier/cheaper to optimize/fix/upgrade
- Highly customizable hardware
  - ISA extension/reduction
  - Application specific on-chip peripherals
- HW/SW hot swap
  - Profiling is key

Outline

- Introduction and Motivation
- Update Mechanisms
  - Dynamic Software Update
  - Partial Logic Reconfiguration
- Holistic Update Mechanism
- Conclusion

Conclusion

- SW maintainability >>> HW maintainability
- FPGAs → more HW flexibility
  - Potential HW maintainability improvent
- Open HW architectures → more dynamic HW development
- Partial logic reconfiguration → dynamic HW updates
- Challenges / Obstacles
  - Expensive FPGAs (money, power, efficiency)
  - Lack of “holistic” developers

Thank you for your attention

[intro1] https://techreport.com/news/20326/intel-finds-flaw-in-sandy-bridge-chipsets-halts-shipments/
[intro2] https://meltdownattack.com/
[hw1] https://reference.digilentinc.com/reference/programmable-logic/basys-3/start
[hw2] www.risc-v.org
[hw3] https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf
[sw1] Batista Ribeiro, Leandro, Fabian Schlager, and Marcel Baunach. "Towards Automatic SW Integration in Dependable Embedded Systems." *EWSN*. 2020.