Simplifying System Design Through Hybrid Dependability Measures

By Christian M. Fuchs, Martin Langer and Carsten Trinitis

How to Achieve System Dependability

- Classics
  - Proven legacy technology
  - Specialized space-grade components
- Holy Grails
  - More robust manufacturing techniques
  - New Technologies
- Redundancy
  - Supervised fail-over (Dual Modular)
  - Voting (Triple Modular)
- Software Measures

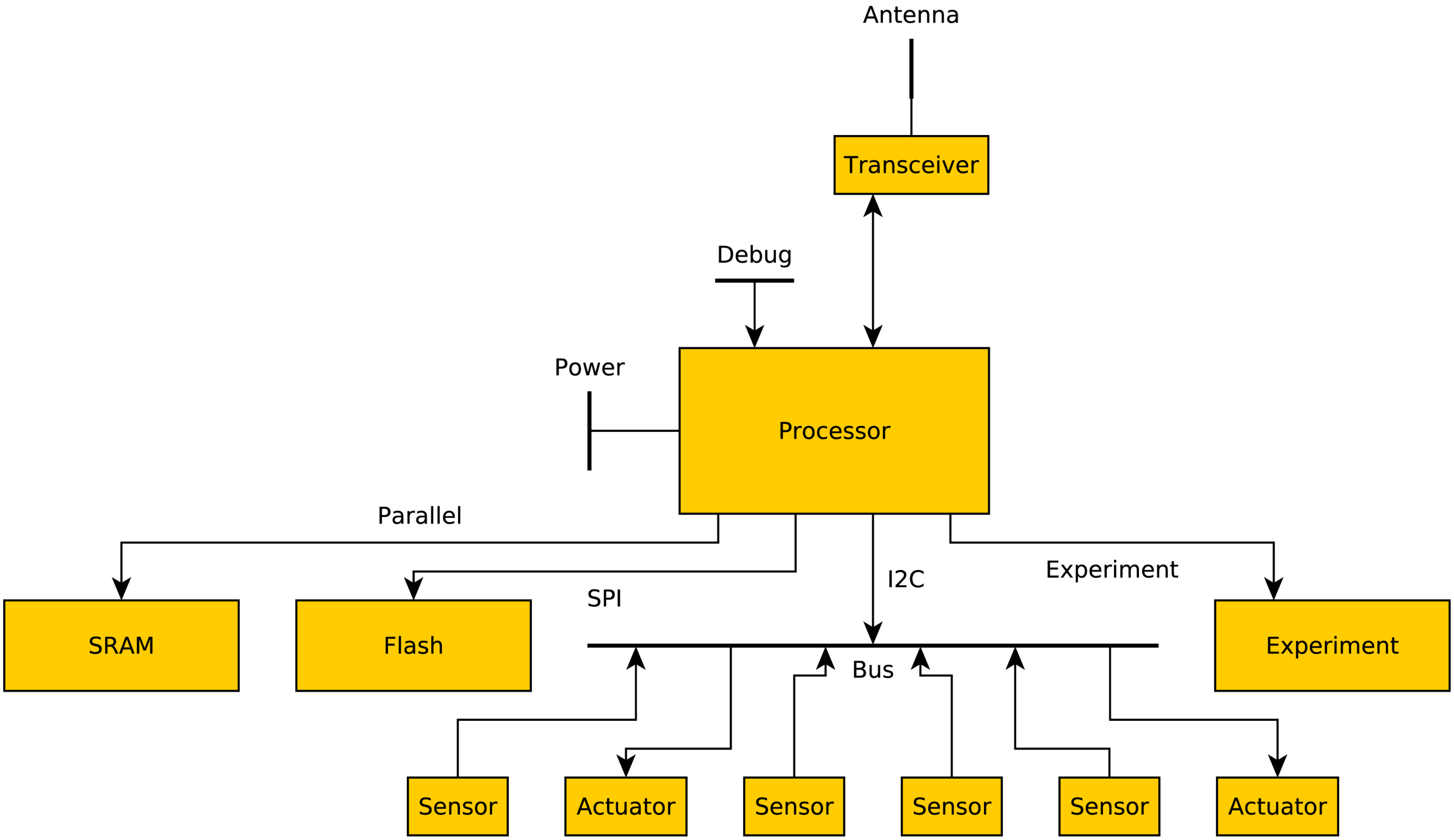
How to Achieve System Dependability

- Classics
  - Proven legacy technology
  - Specialized space-grade components
- Holy Grails
  - More robust manufacturing techniques
  - New Technologies
- Redundancy
  - Supervised fail-over (Dual Modular)
  - Voting (Triple Modular)
- Software Measures

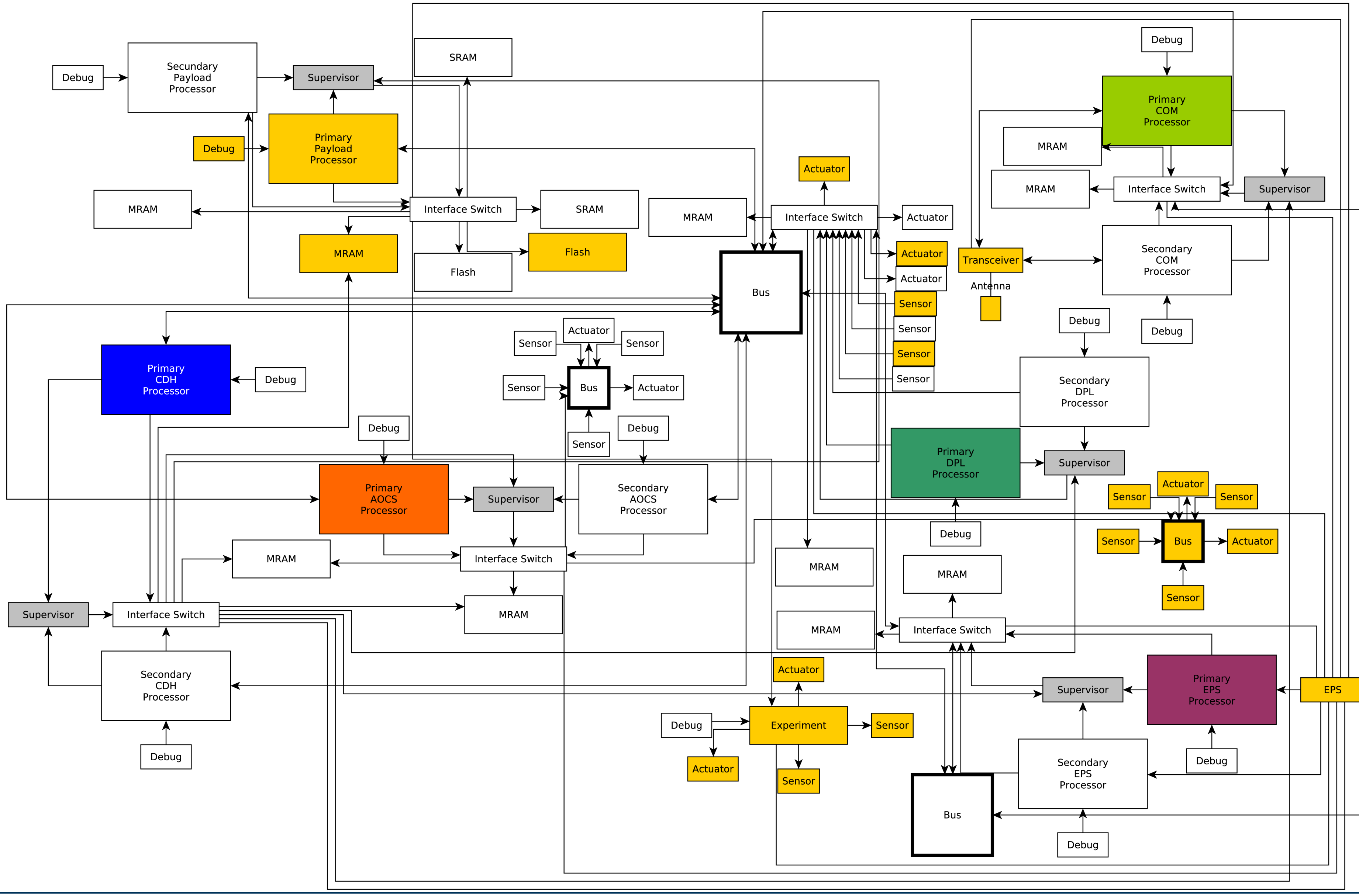
How to Achieve System Dependability

- Classics
  - Proven legacy technology
  - Specialized space-grade components
- Holy Grails
  - More robust manufacturing techniques
  - New Technologies
- Redundancy
  - Supervised fail-over (Dual Modular)
  - Voting (Triple Modular)
- Software Measures

Why not in just more REDUNDANCY?



Complexity!


Debug
Secundary
Payload
Processor
Supervisor
SRAM
Debug
Primary
Payload
Processor
MRAM
Interface Switch
SRAM
MRAM
Flash
Flash
MRAM
Interface Switch
Actuator
Actuator
Actuator
Actuator
Bus
Sensor
Sensor
Sensor
Sensor
Transceiver
Antenna
Debug
Primary
COM
Processor
MRAM
MRAM
Interface Switch
Supervisor
Secondary
COM
Processor
Debug
Primary
CDH
Processor
Debug
Sensor
Actuator
Sensor
Sensor
Bus
Actuator
Sensor
Debug
Debug
Debug
Secondary
DPL
Processor
Primary
AOCS
Processor
Supervisor
Secondary
AOCS
Processor
Primary
DPL
Processor
Supervisor
Actuator
Sensor
Sensor
Sensor
Bus
Actuator
Sensor
Debug
MRAM
Interface Switch
MRAM
MRAM
Supervisor
Interface Switch
MRAM
MRAM
Interface Switch
Secondary
CDH
Processor
Debug
Actuator
Debug
Experiment
Sensor
Actuator
Sensor
Supervisor
Primary
EPS
Processor
EPS
Secondary
EPS
Processor
Debug
Bus
Debug

How to Achieve System Dependability

- Classics
  - Proven legacy technology
  - Specialized space-grade components
- Holy Grails
  - More robust manufacturing techniques
  - New Technologies
- Redundancy
  - Supervised fail-over (Dual Modular)
  - Voting (Triple Modular)
- **Software Measures**

Dependable Data Storage on Modern Memory

- Foundation for advanced consistency concepts

- System Storage
  - RTOS Firmware, Kernel + Userland, ApplicationCode
  - Small footprint but consistency critical

- Payload Data Storage
  - Data storage for scientific data
  - Only lossy technology available (MLC-flash)
  - Large & challenging throughput requirements

- Volatile Memory

Protecting Operating System Data, an Example

Application	logger
API	POSIX
Virtual File System	Virtual File System
File System	File System Infrastructure
Block Abstraction	blockdev
Device Abstraction	/dev/eeprom0
Device Driver	SPI driver
Memory Technology	EEPROM

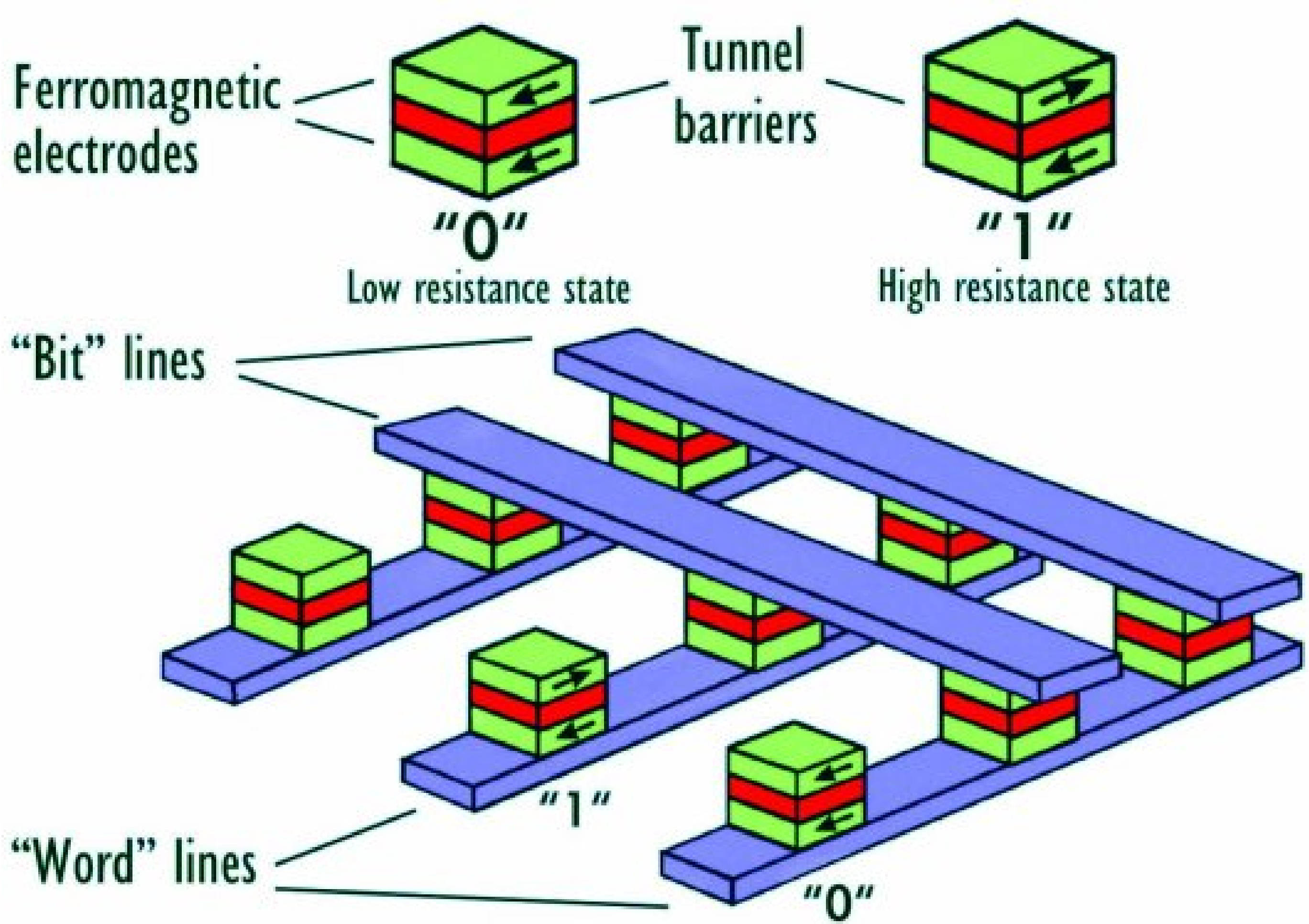
Protecting Operating System Data, an Example

Application	logger
API	POSIX
Virtual File System	Virtual File System
File System	File System Infrastructure
Block Abstraction	blockdev
Device Abstraction	/dev/eeprom0
Device Driver	SPI driver
Memory Technology	EEPROM

Protecting Operating System Data, an Example

Application	logger
API	POSIX
Virtual File System	Virtual File System
File System	File System Infrastructure
Block Abstraction	blockdev
Device Abstraction	/dev/eeprom0
Device Driver	SPI driver
Memory Technology	EEPROM

Magnetoresistive RAM

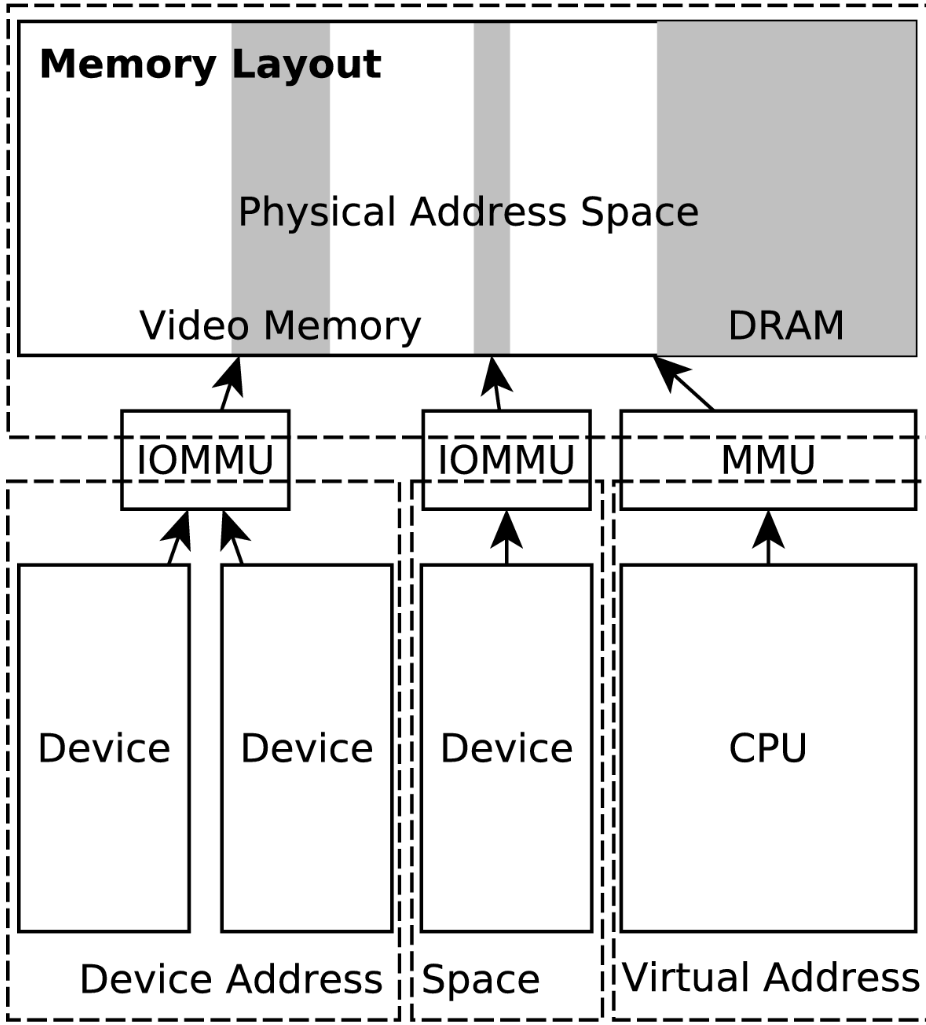


Source: Fert A, George J-M, Jaffres H, Mattana R and Seneor P, Europhysics News 34(6) 227, 2003

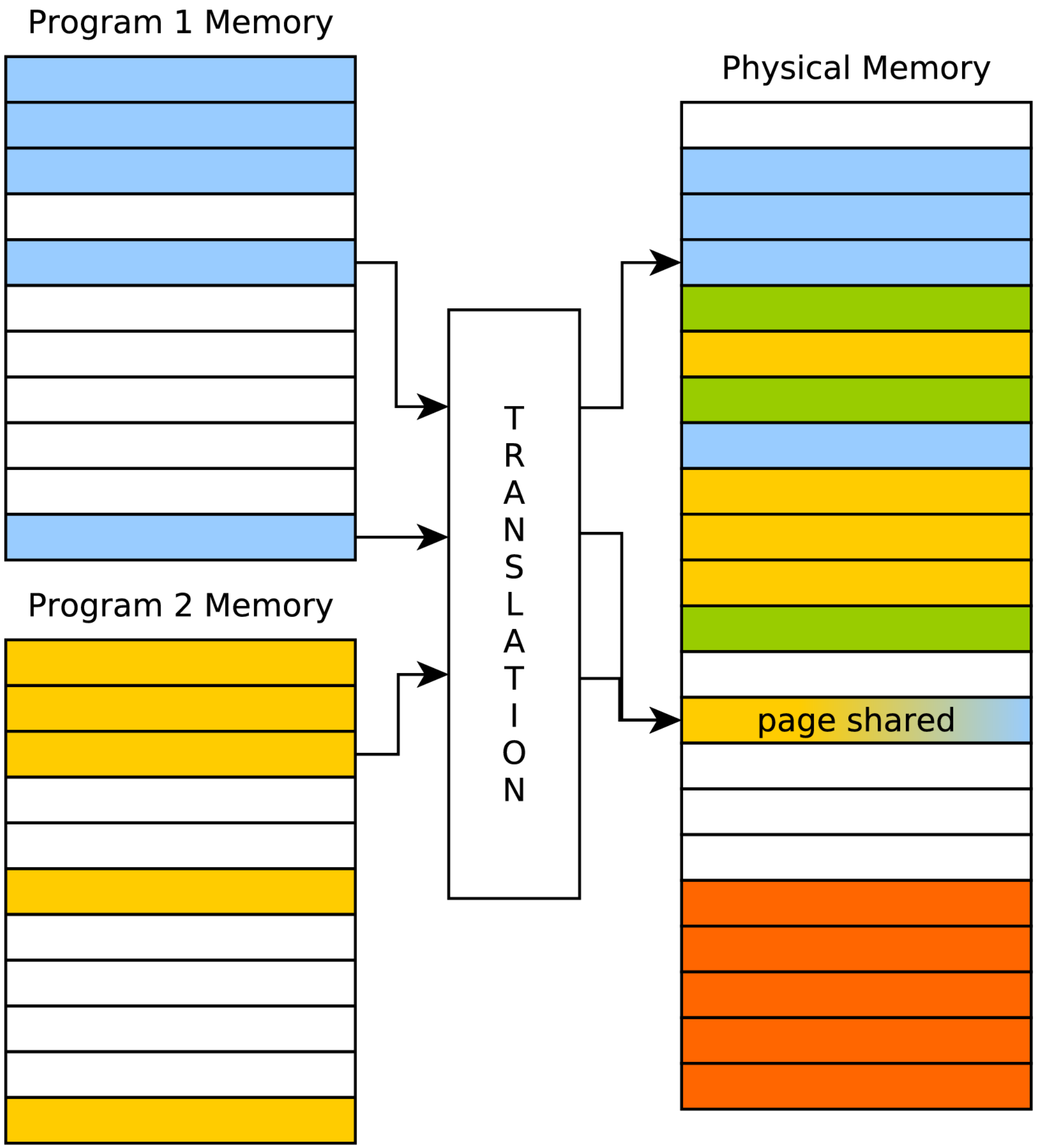
High-Rel Data Storage on MRAM

Application	Application
API	API
Virtual File System	Virtual File System
File System	File System Infrastructure
Block Abstraction	
Device Abstraction	
Device Driver	Memory Mapping
Memory Technology	MRAM

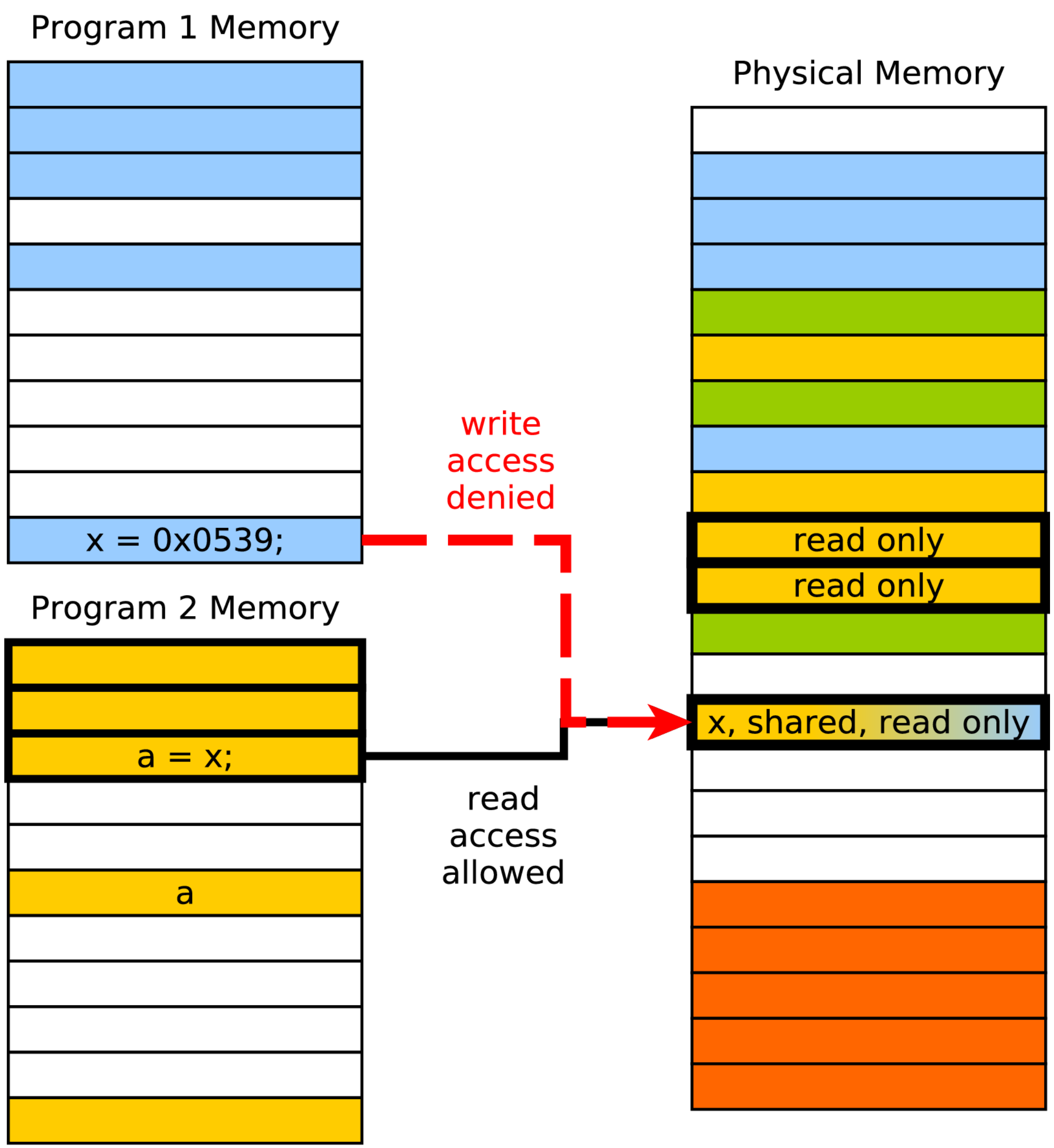
Virtual & Physical Memory



Virtual Memory Mappings

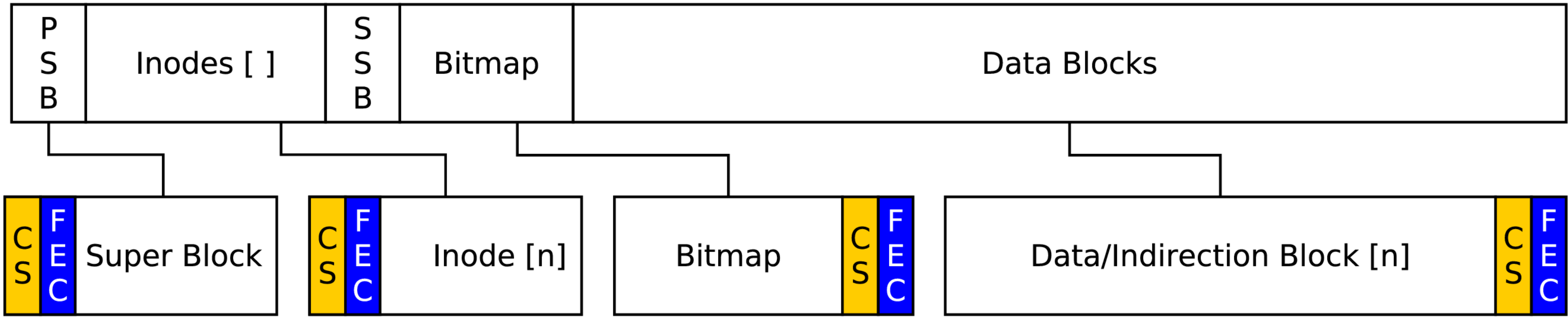


Memory Protection



FTRFS: A Fault-Tolerant Radiation-Robust File System for Space Use

- **Dynamic, variable EDAC strength**
- Erasure coding using Reed Solomon
- CRC32 to enhance performance
- Memory protection



Dependable Data Storage on Modern Memory

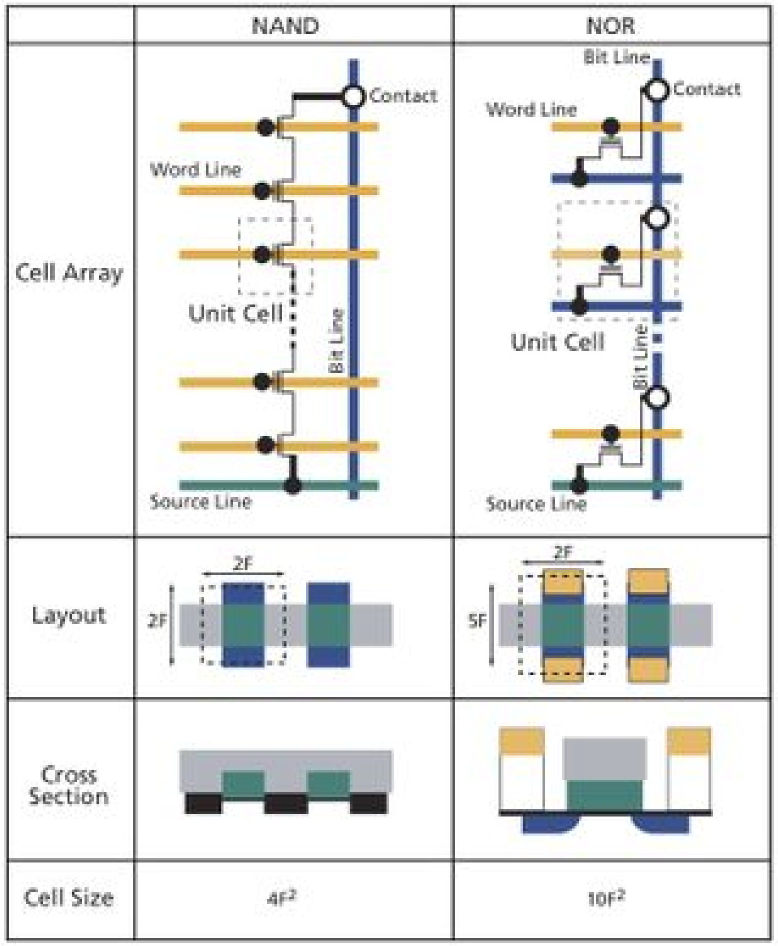
- Foundation for advanced consistency concepts

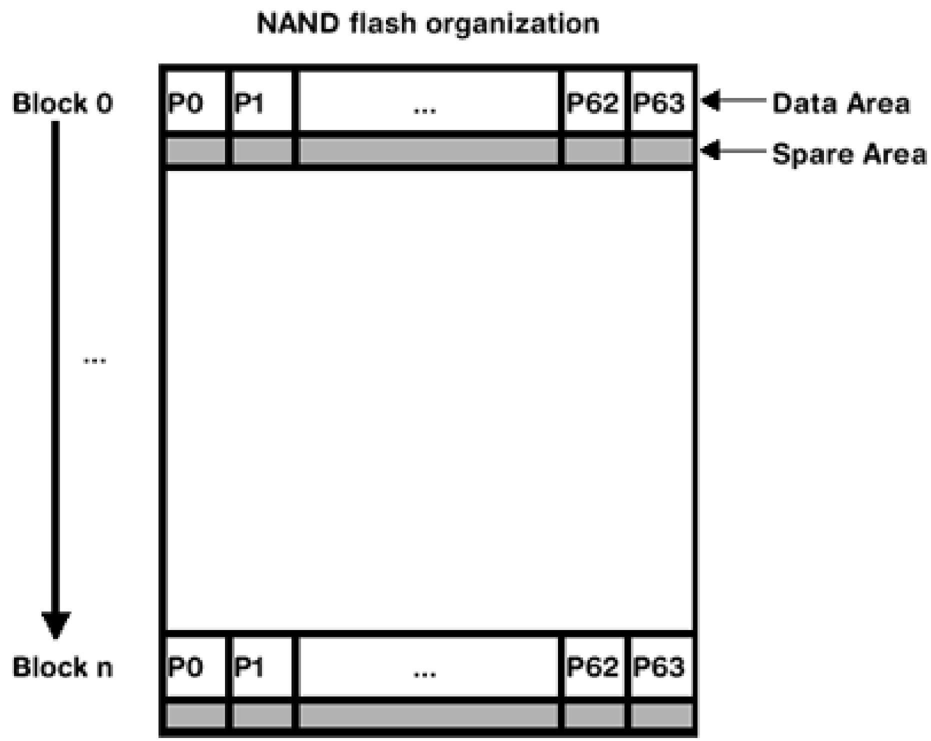
- System Storage
  - RTOS Firmware, Kernel + Userland, ApplicationCode
  - Small footprint but consistency critical

- Payload Data Storage
  - Data storage for scientific data
  - Only lossy technology available (MLC-flash)
  - Large & challenging throughput requirements

- Volatile Memory

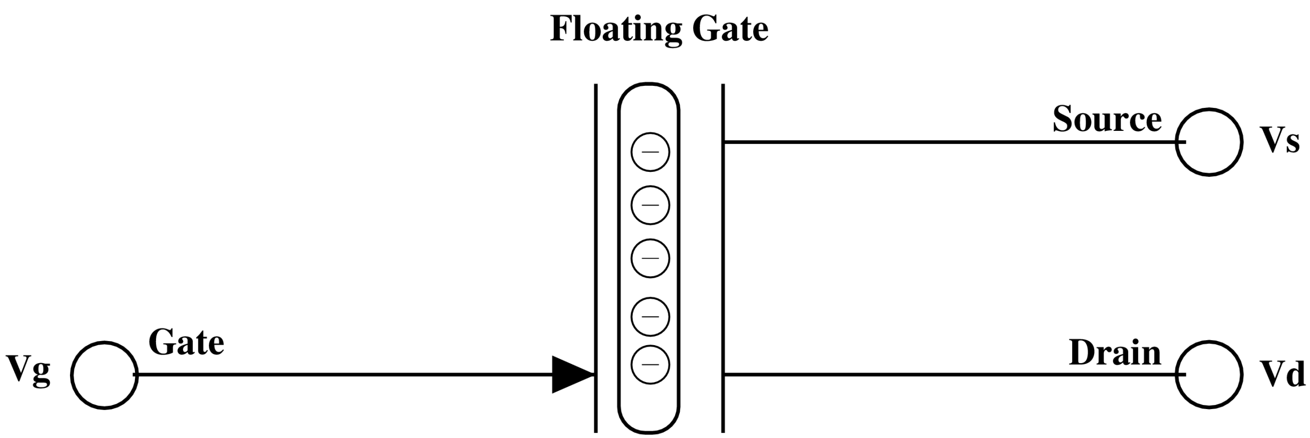
NAND & NOR Flash Memory

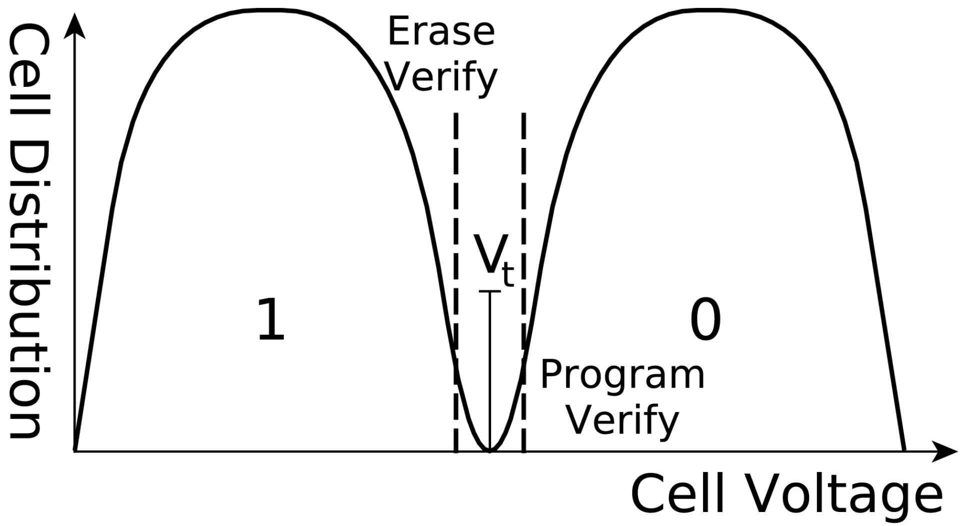




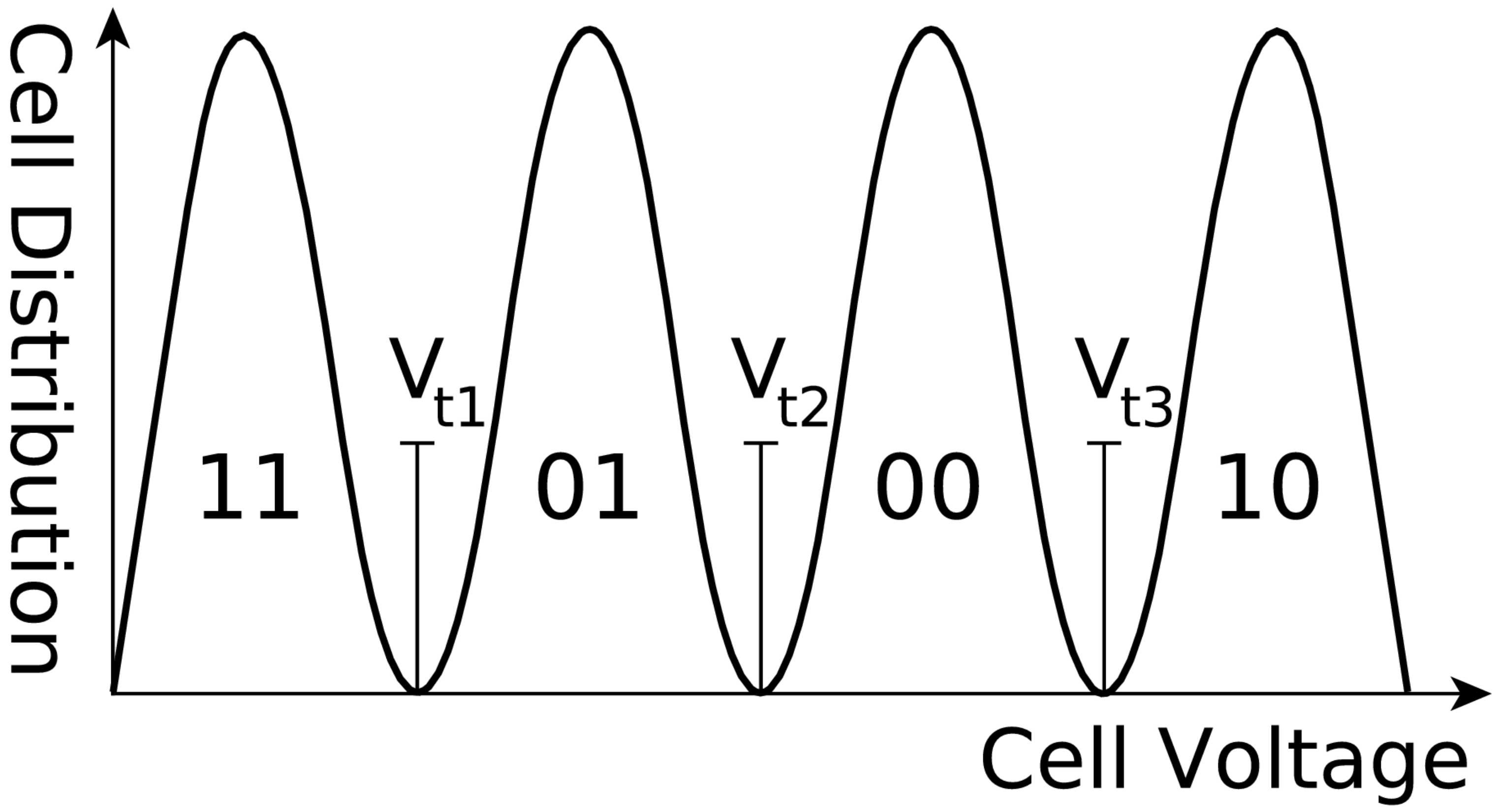
Source: Wind River Systems (L), SEGGER Microcontroller (R)

Single-Level Cell Flash

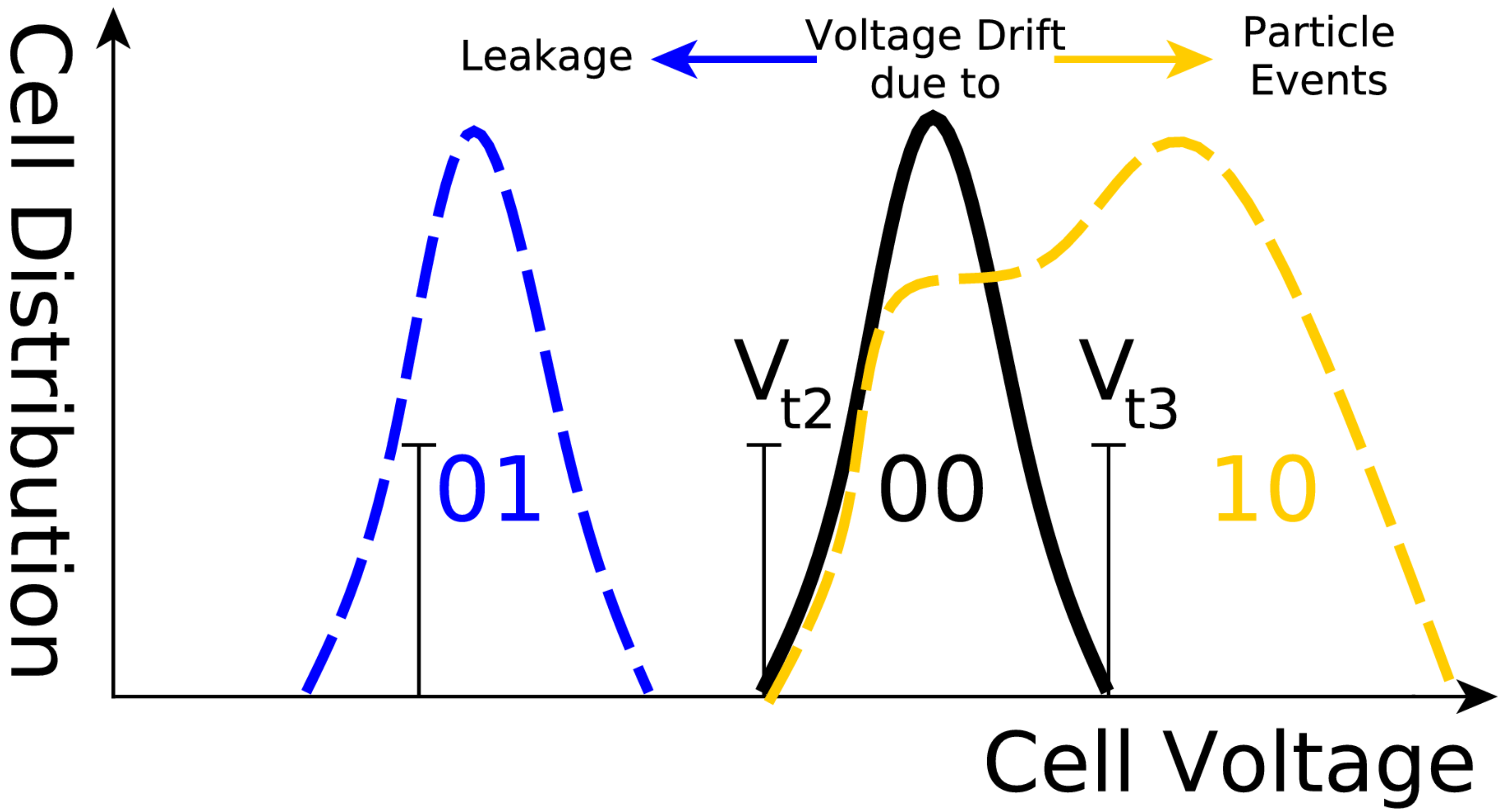




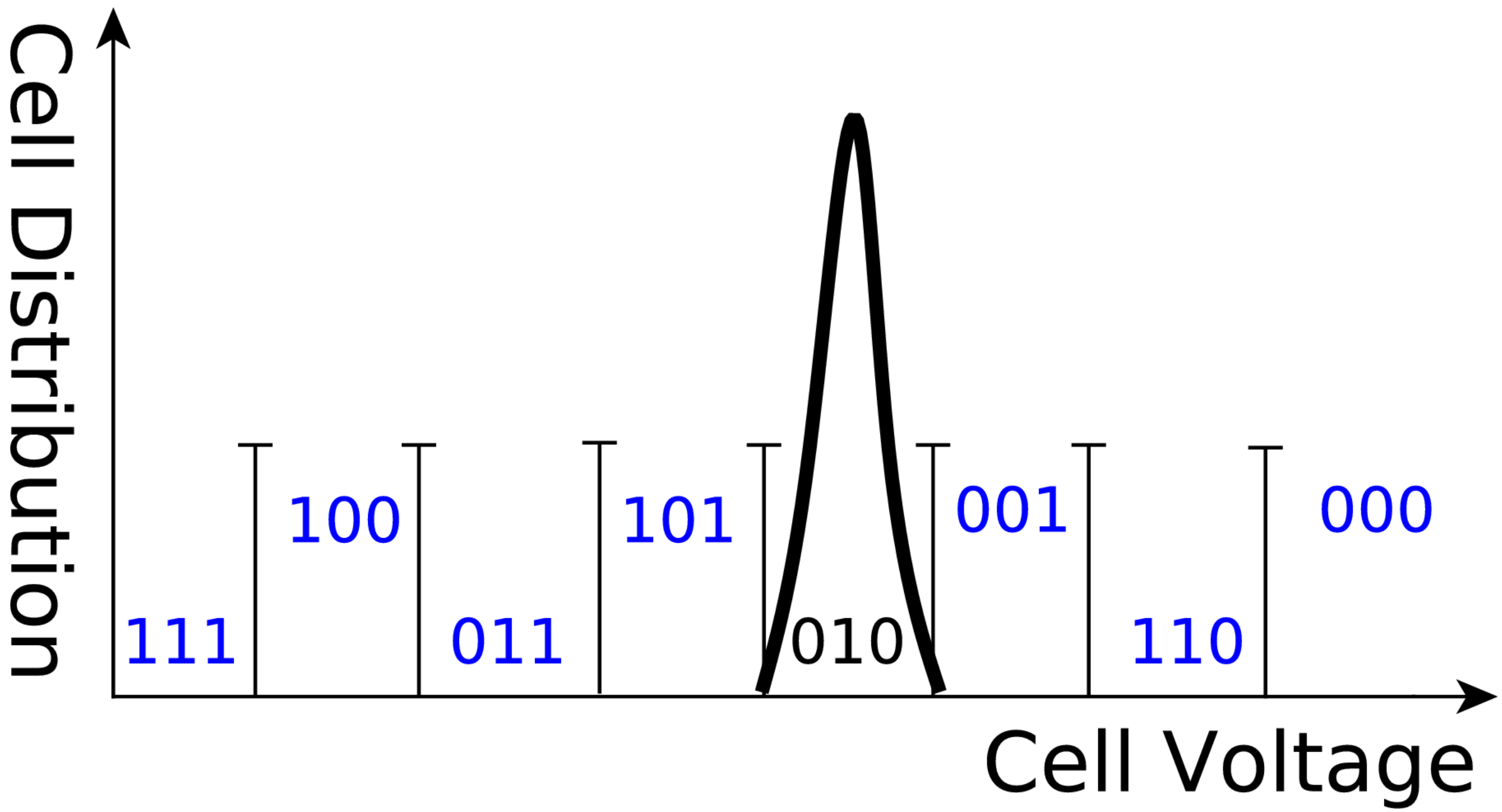
Multi-Level Cell Flash



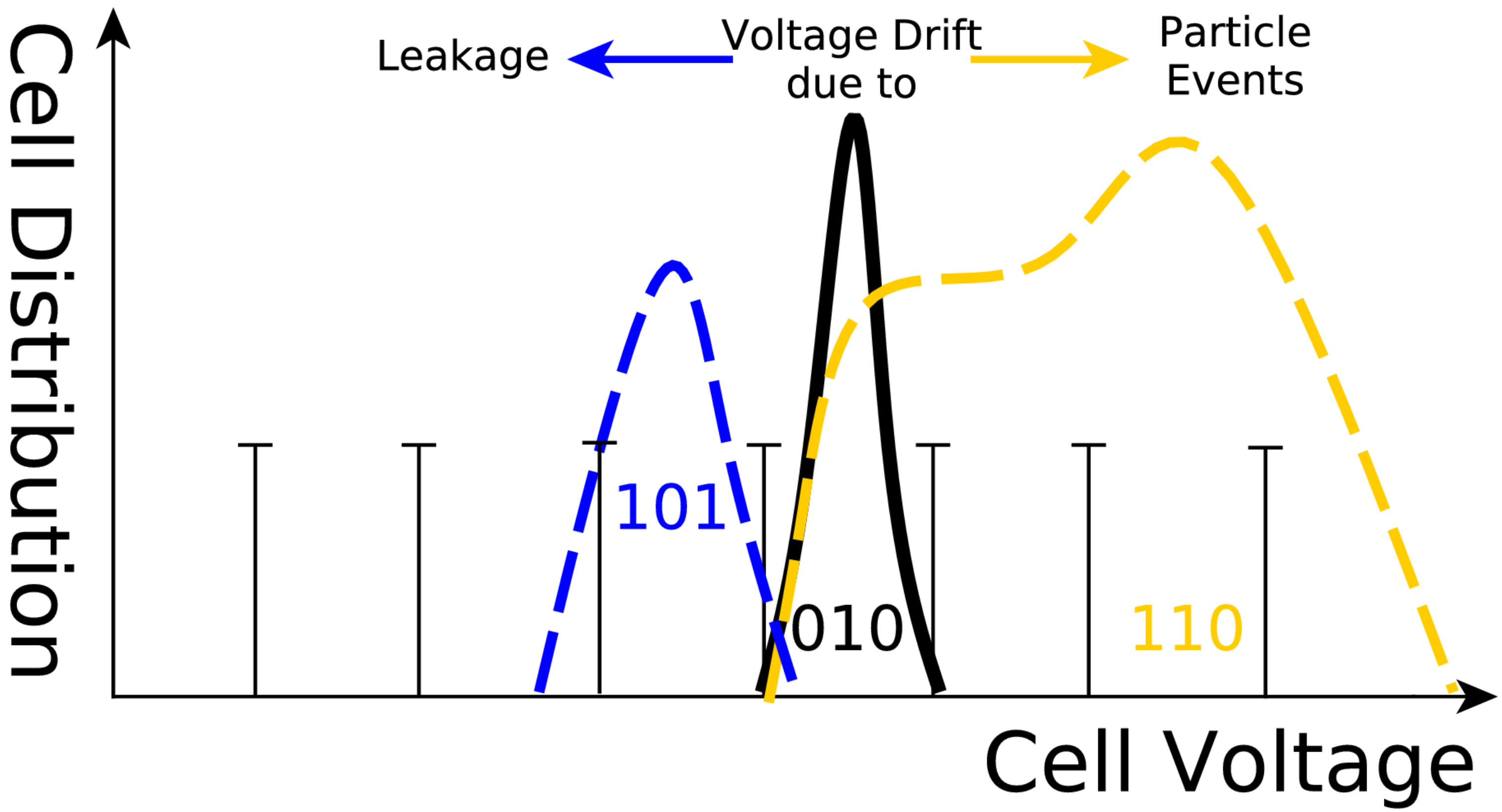
Data Corruption in Multi-Level Cell Flash



Higher Order Multi-Level Cell Flash

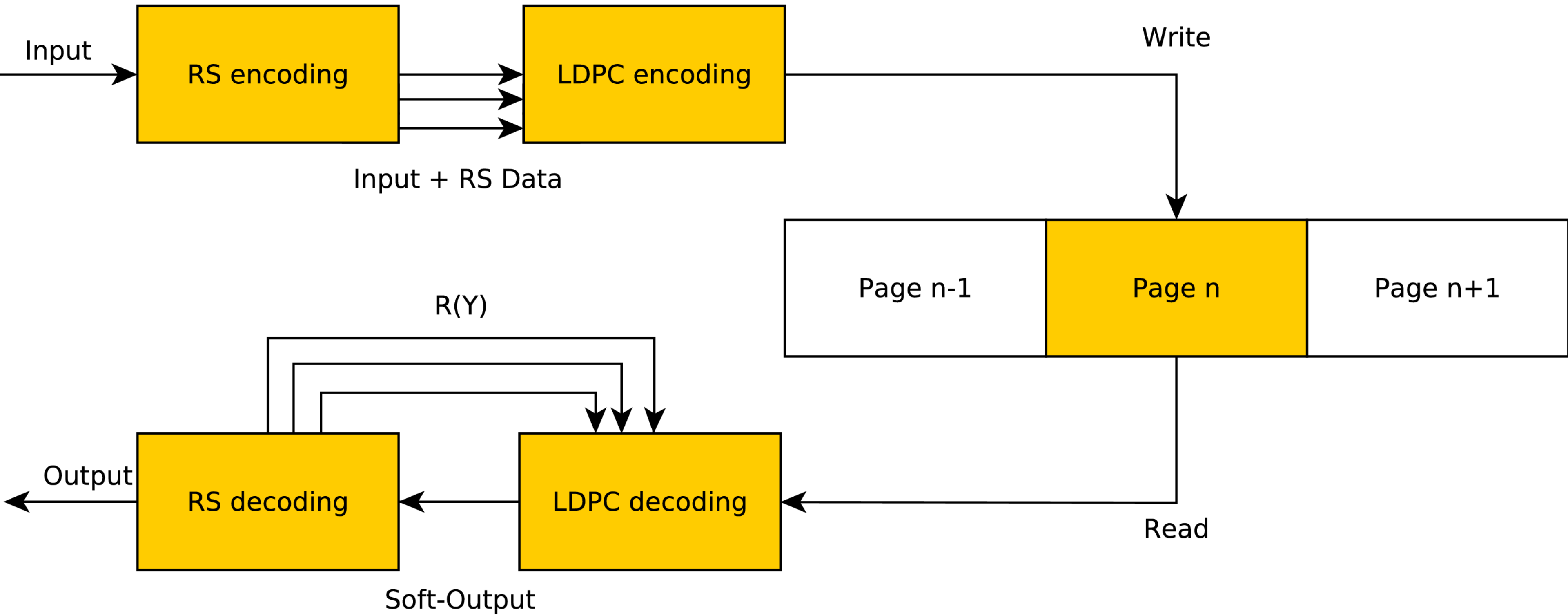


Data Corruption in Multi-Level Cell Flash

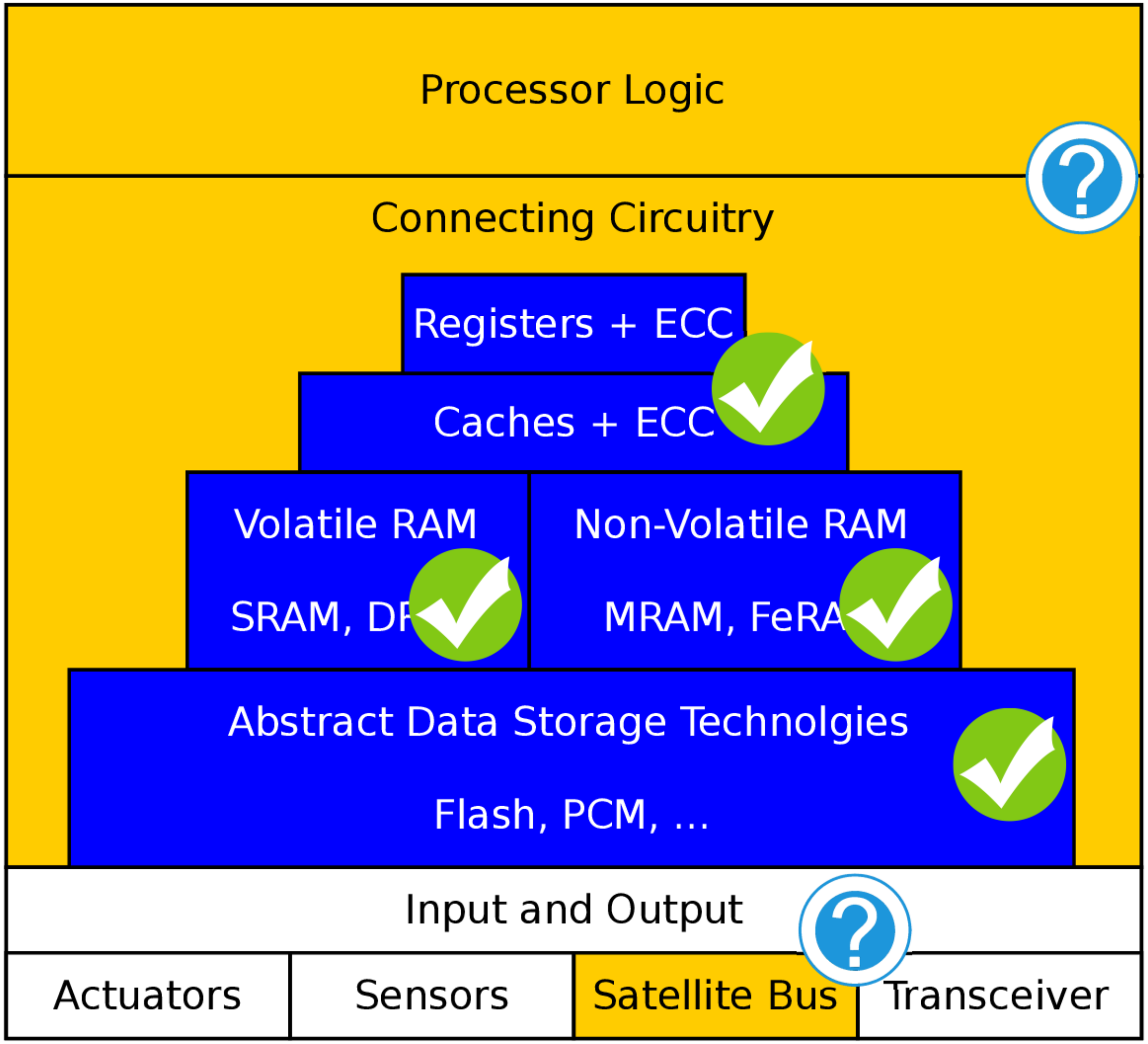


Adapting Erasure Coding for modern MLC-Flash

- MTD-Mirror: a dependable software layer for NAND-flash
- Two Erasure Coding Layers
  - Reed-Solomon to counter multi-bit errors and state-machine issues
  - LDPC to counter single-bit errors



What's Next?

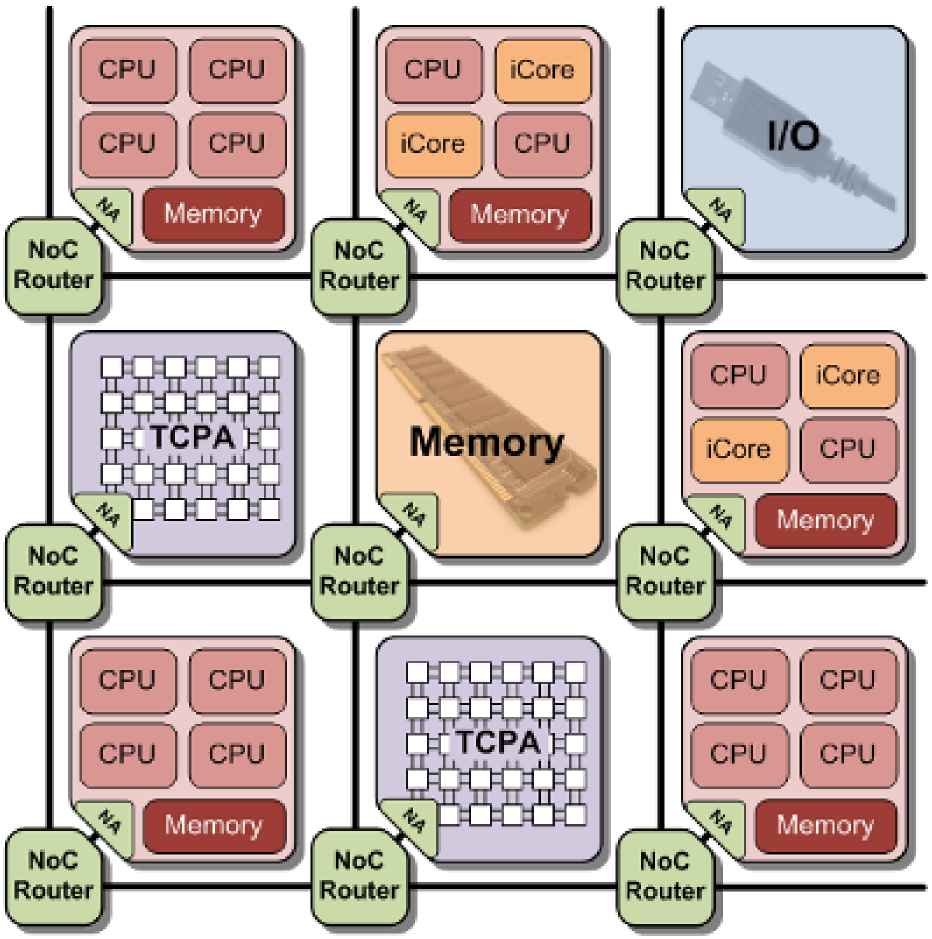

Processor Logic
Connecting Circuitry
Registers + ECC
Caches + ECC
Volatile RAM
SRAM, DR
Non-Volatile RAM
MRAM, FeRA
Abstract Data Storage Technolgies
Flash, PCM, ...
Input and Output
Actuators
Sensors
Satellite Bus
Transceiver

Compute Consistency: The State of Affairs

- Legacy Technology & Coarse Structures
- DMR/Supervised Processors
- TMR Approaches with COTS Processors
- TMR-ed Processor Designs (LEONx-FT)

- Ongoing Research:
  - NoCs, ManyCores & Grid Approaches

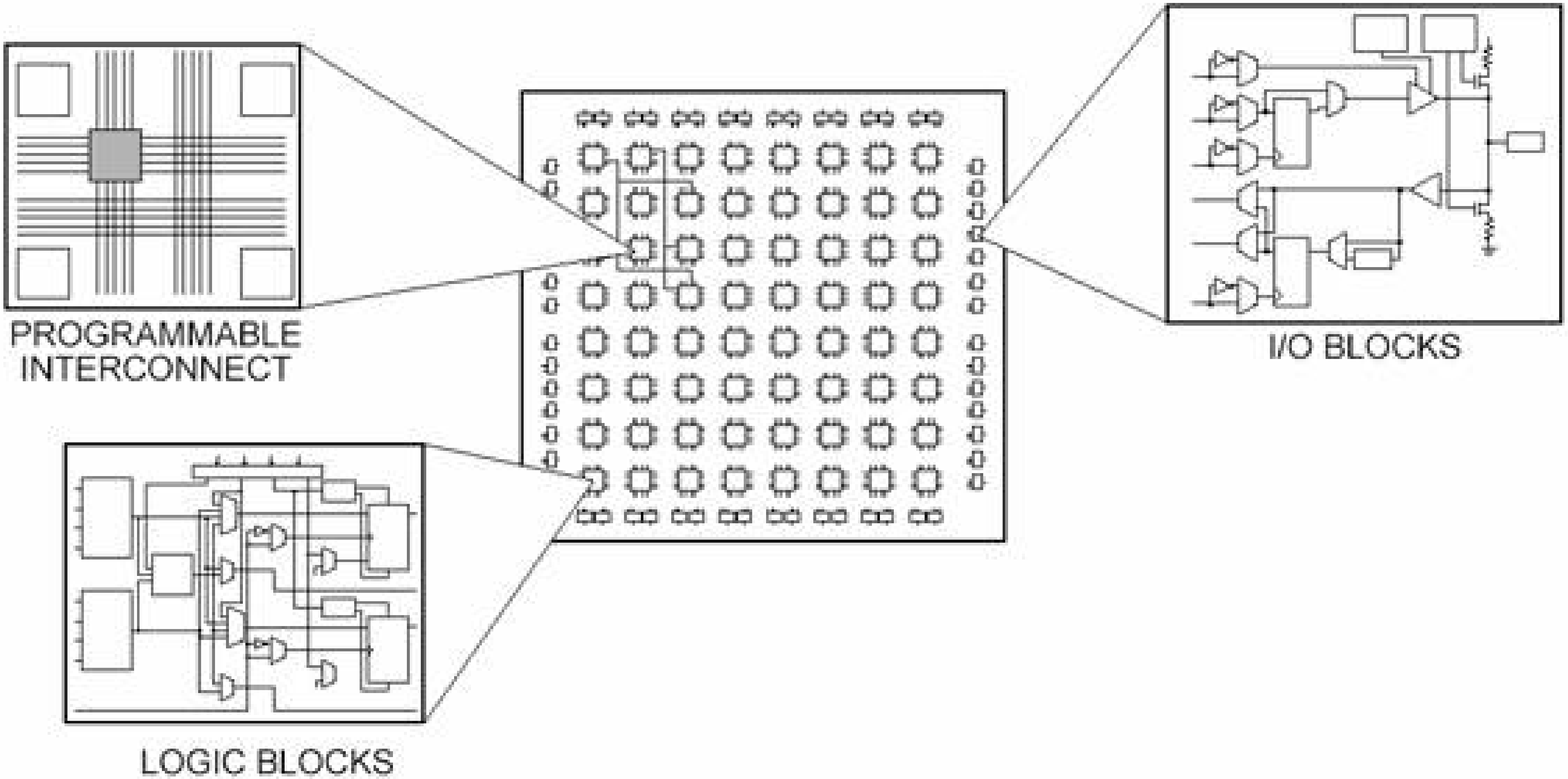
NoCs & Many Cores



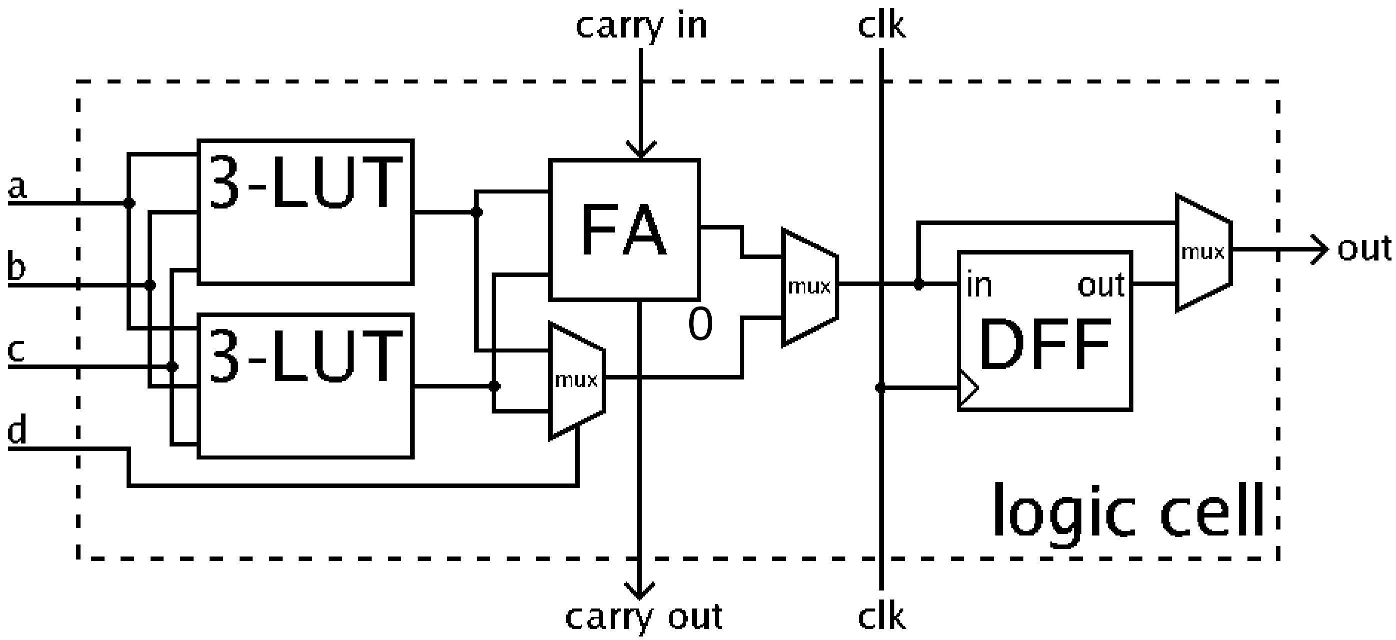
Source: Teich J. et al: "Invasive NoCs – Autonomous, Self-Optimising Communication Infrastructures for MPSoCs,", University of Erlangen, 2012

FPGAs and Dependability

- Excellent fault-recovery protential



- Excellent Fault-recovery protential:
  - Partial reconfiguration
  - Reconfiguration using different routing

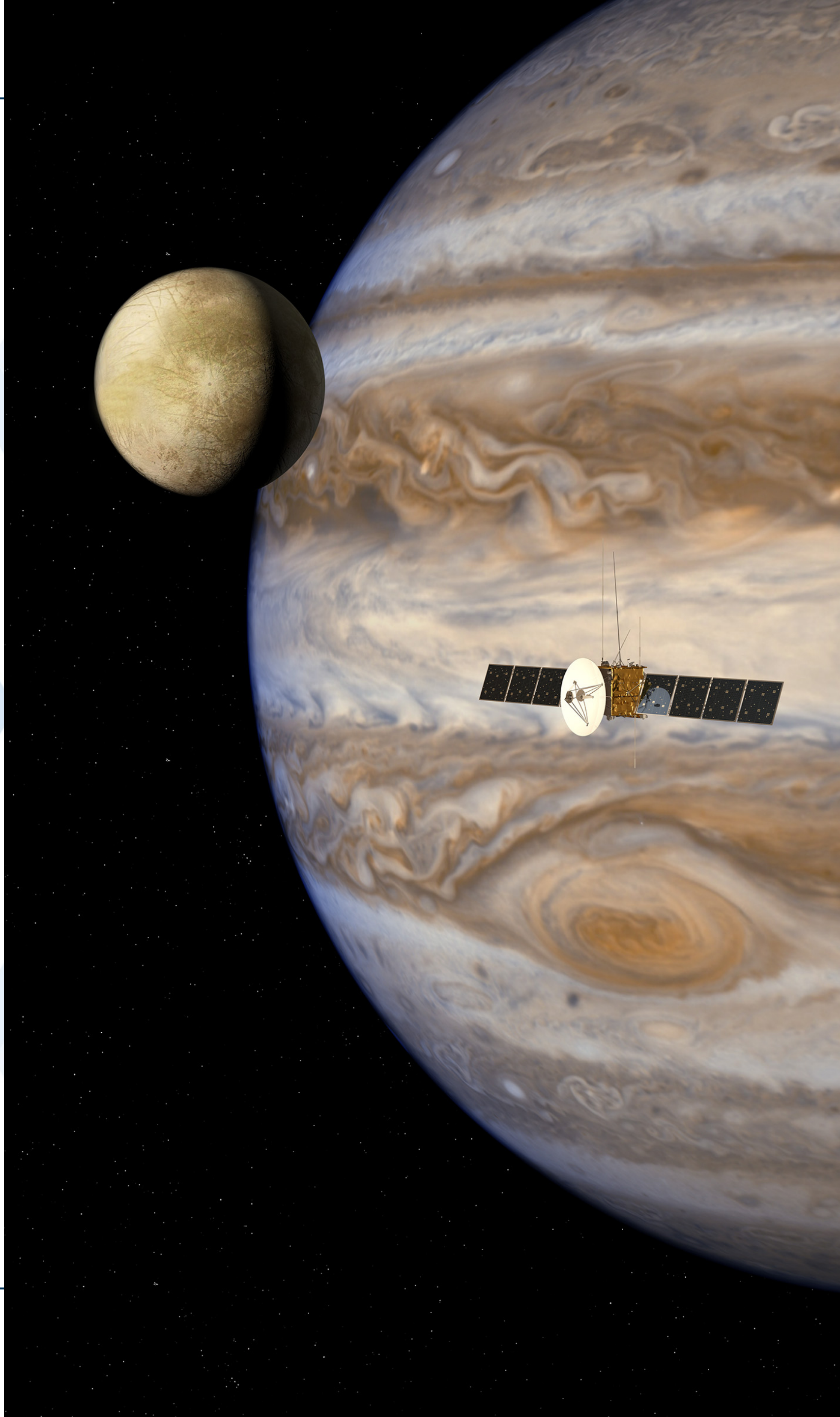


FPGAs and Dependability

- Core Issues:
  - Reliable Storage of Designs (“solved”)
  - Compute Performance
  - **Fault Detection**

Thanks for Listening!

contact:
christian.fuchs@tum.de

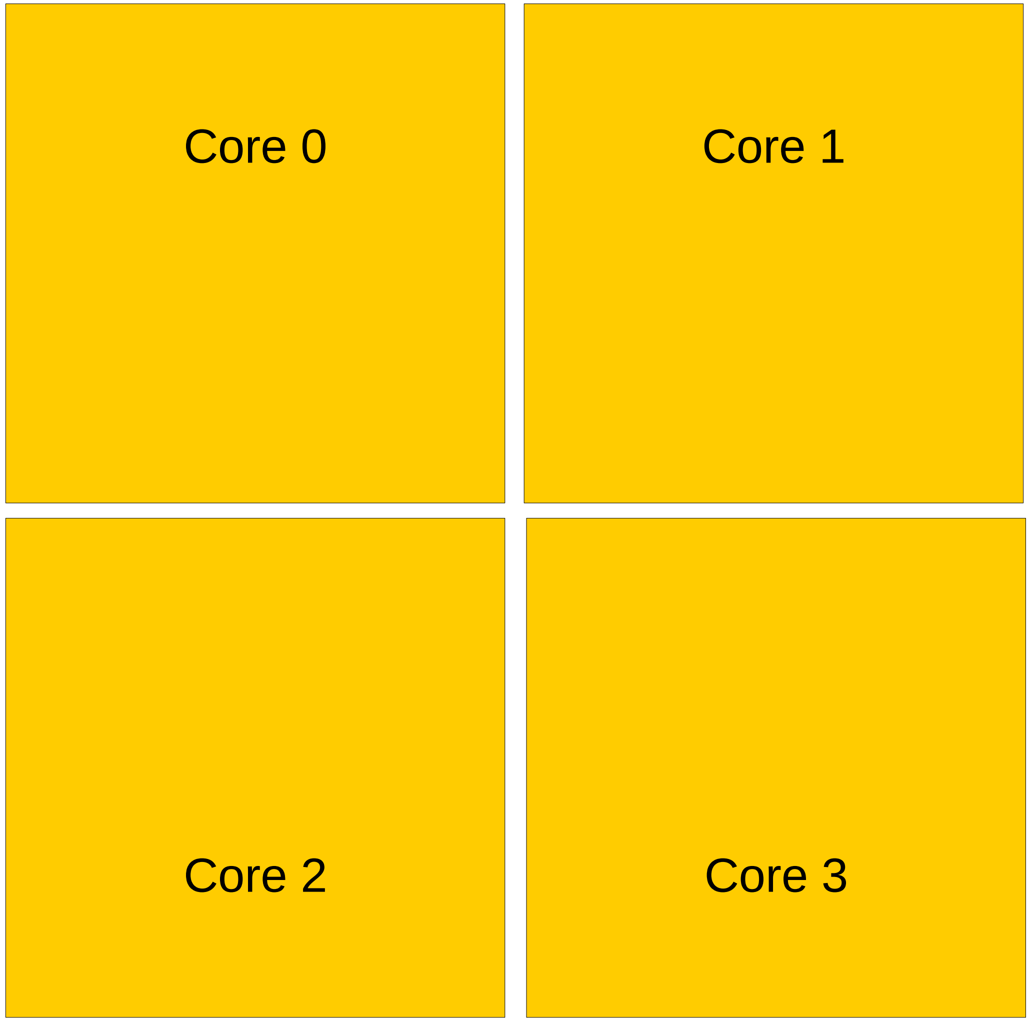


Software-Driven Approaches

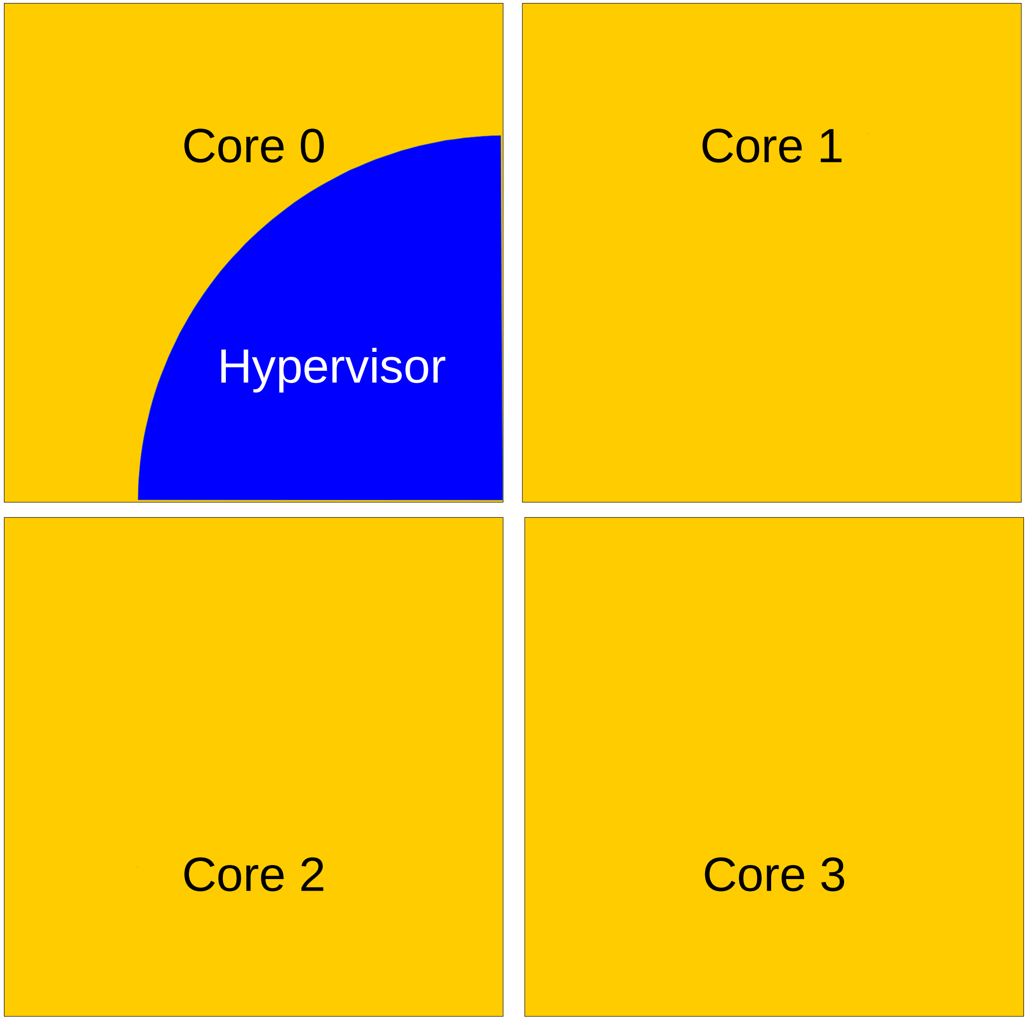
- Lockstepping
  - Multiple Independent Parallel Program Flows
  - Static Compute Overhead

- Vectorized-Voting
  - Only requires SIMD extensions
  - Enables On-demand Voting
  - Non-SIMD operations not protected

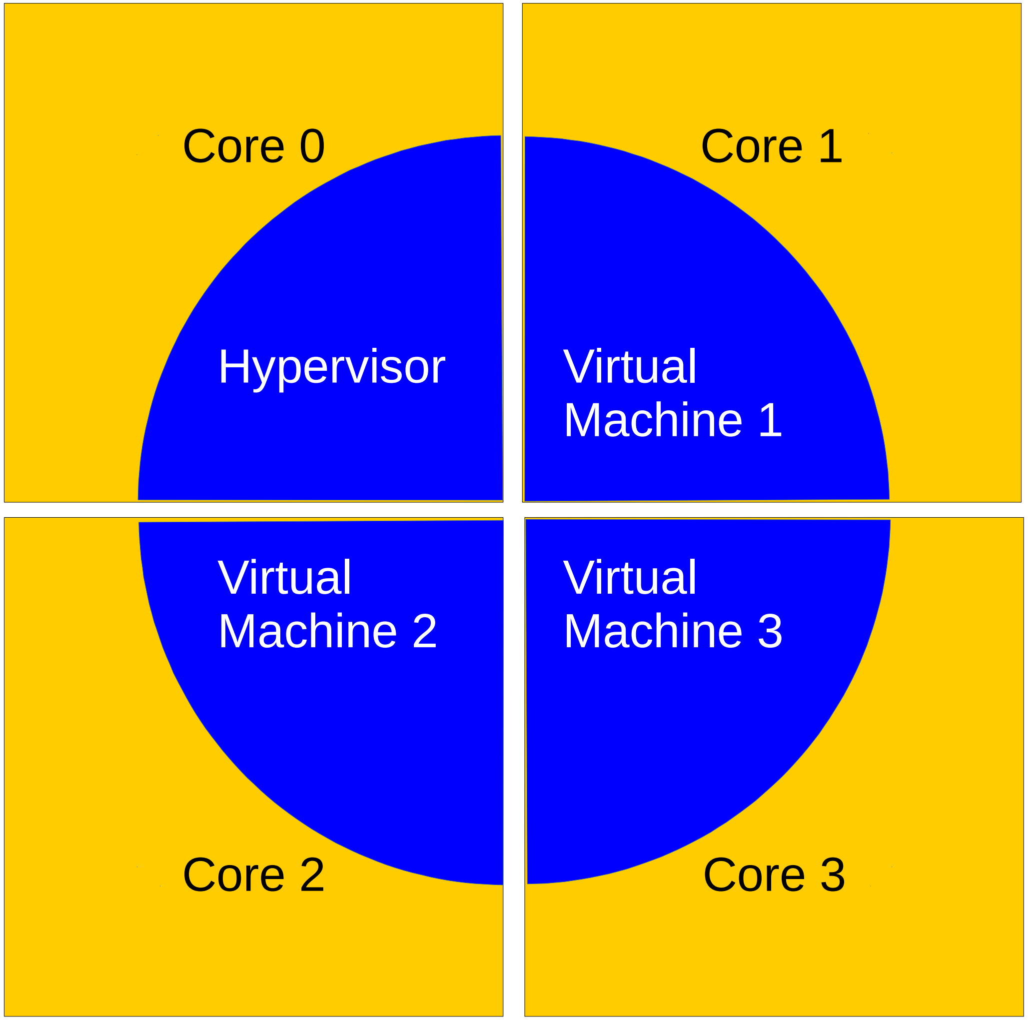
Virtualization Enabled Voting



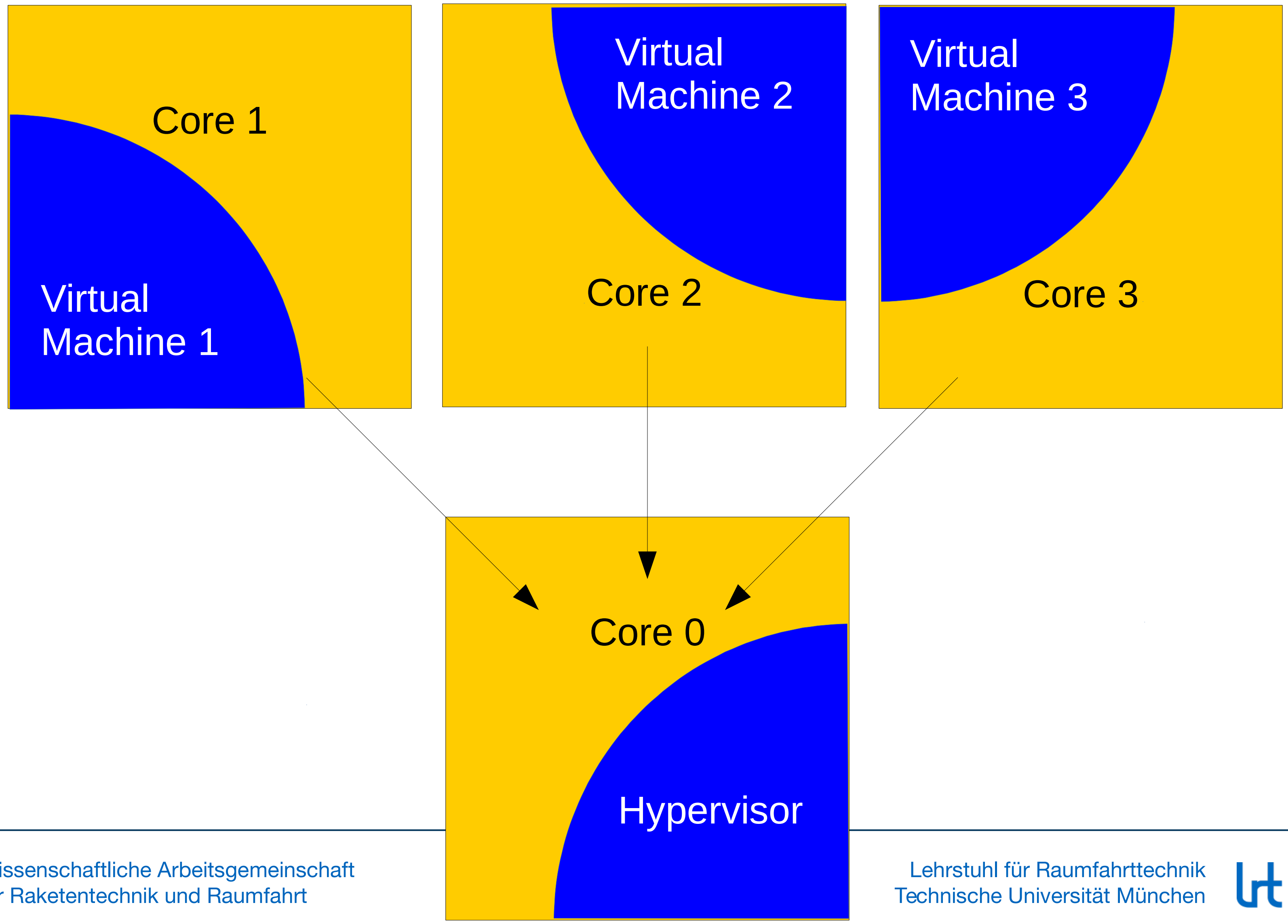
Virtualization Enabled Lockstepping/Voting



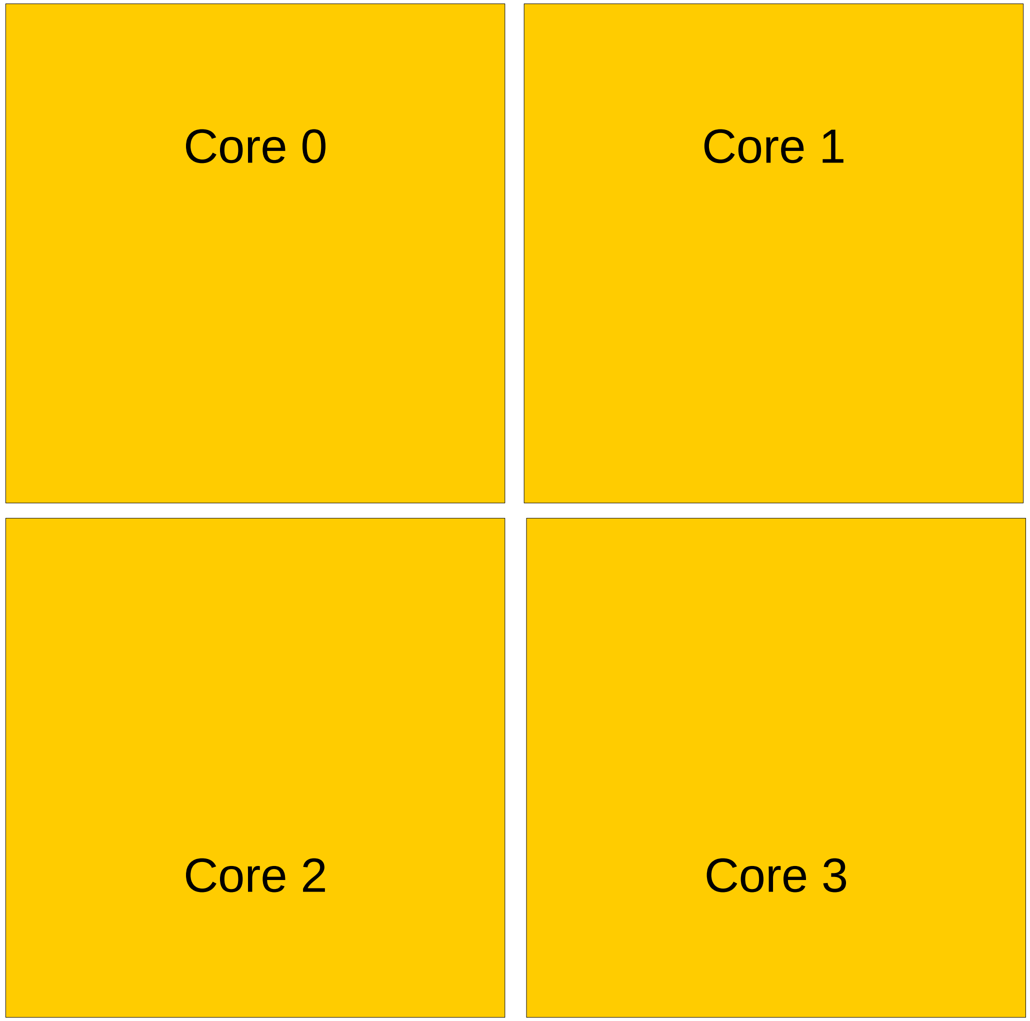
Virtualization Enabled Lockstepping/Voting



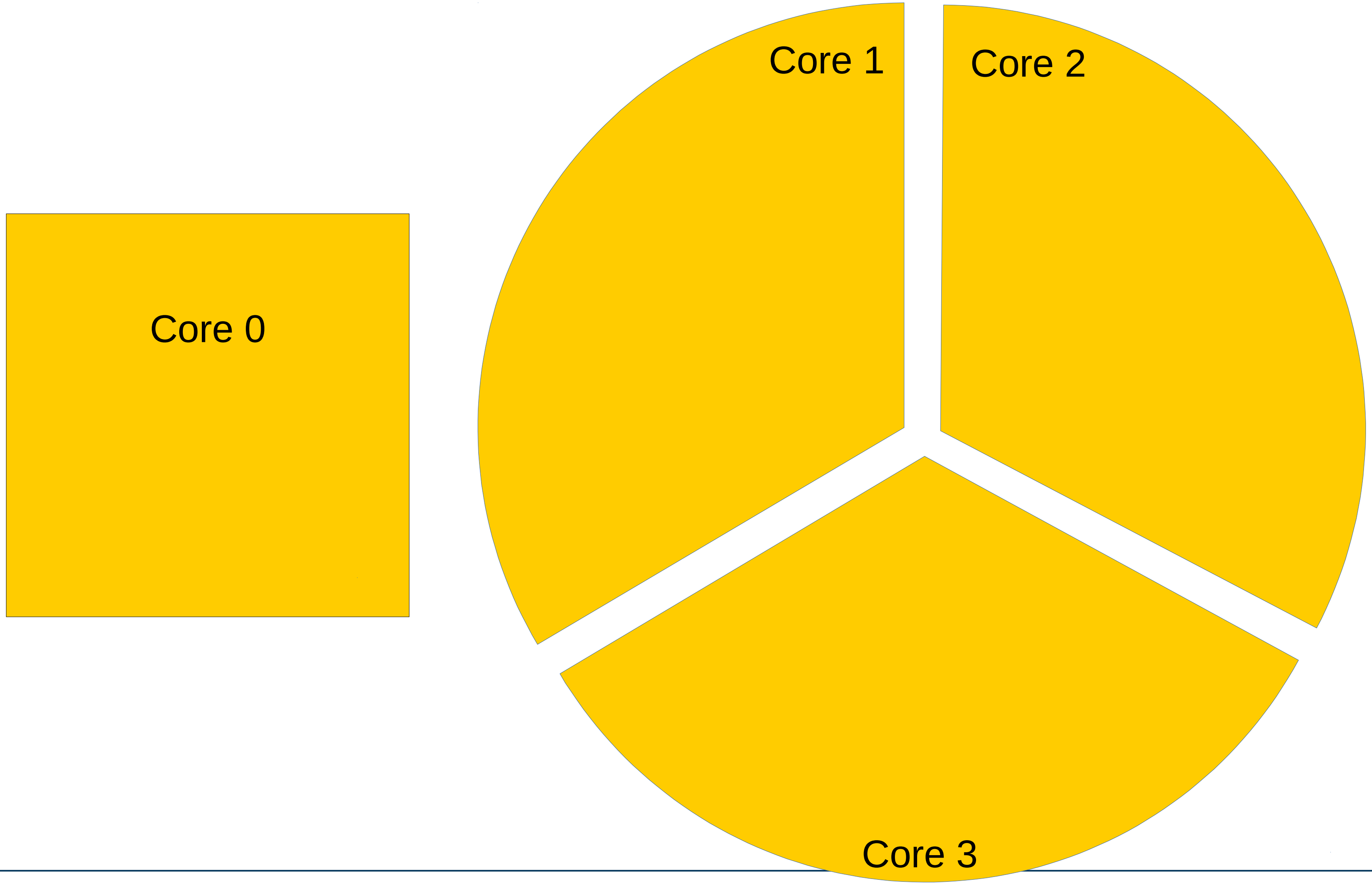
Virtualization Enabled Lockstepping/Voting



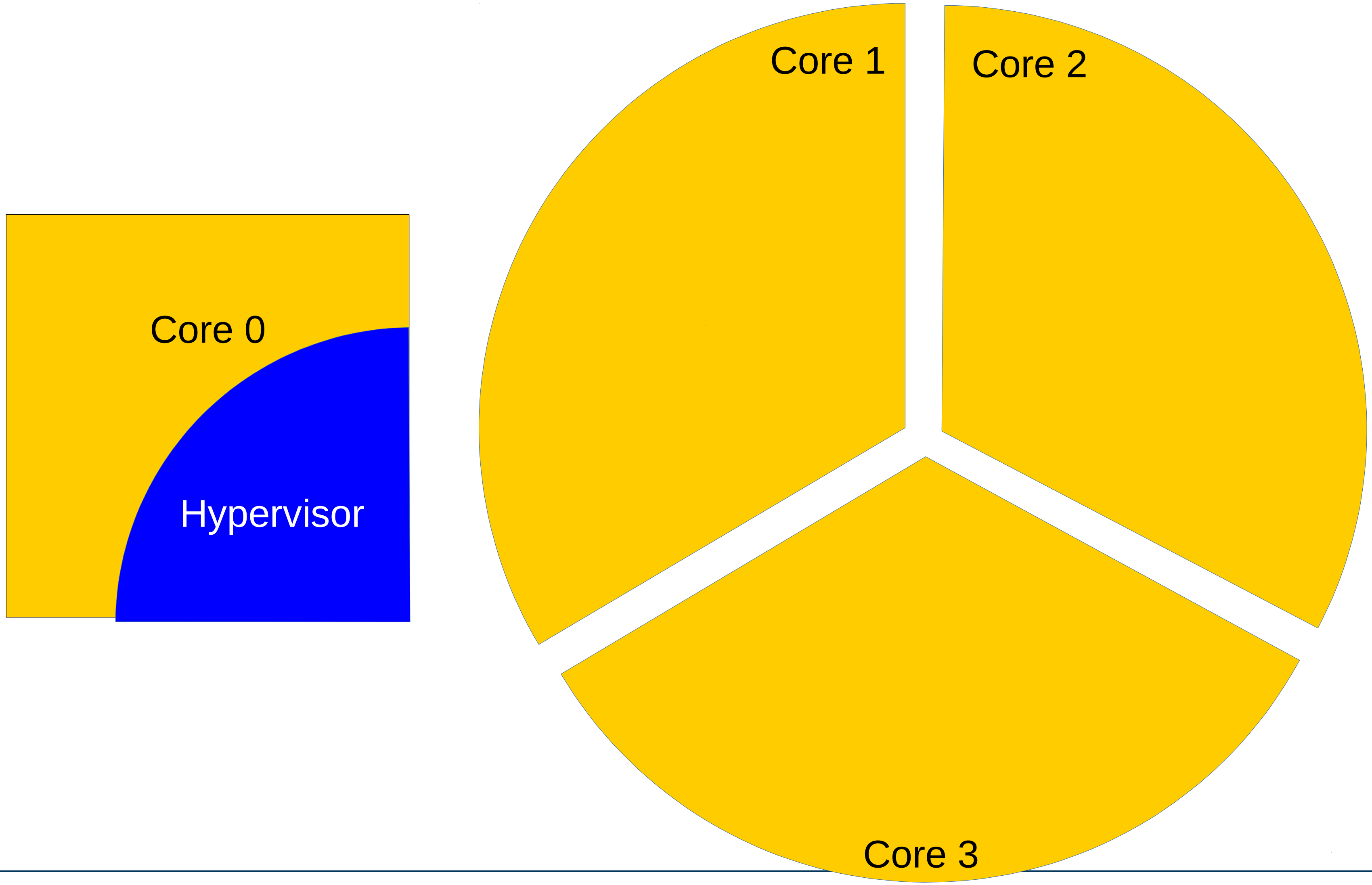
Virtualization Enabled Voting



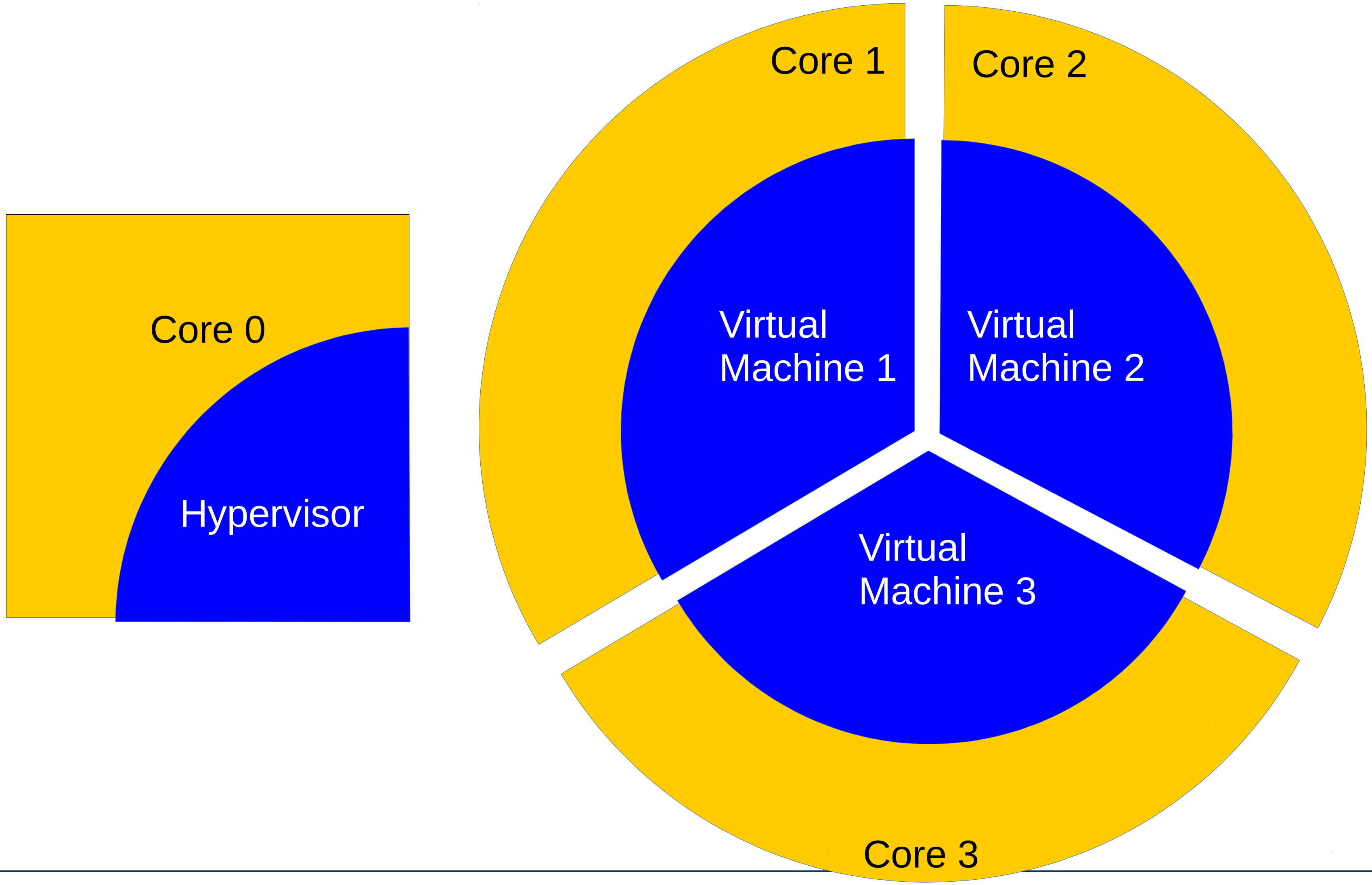
Virtualization Enabled Voting



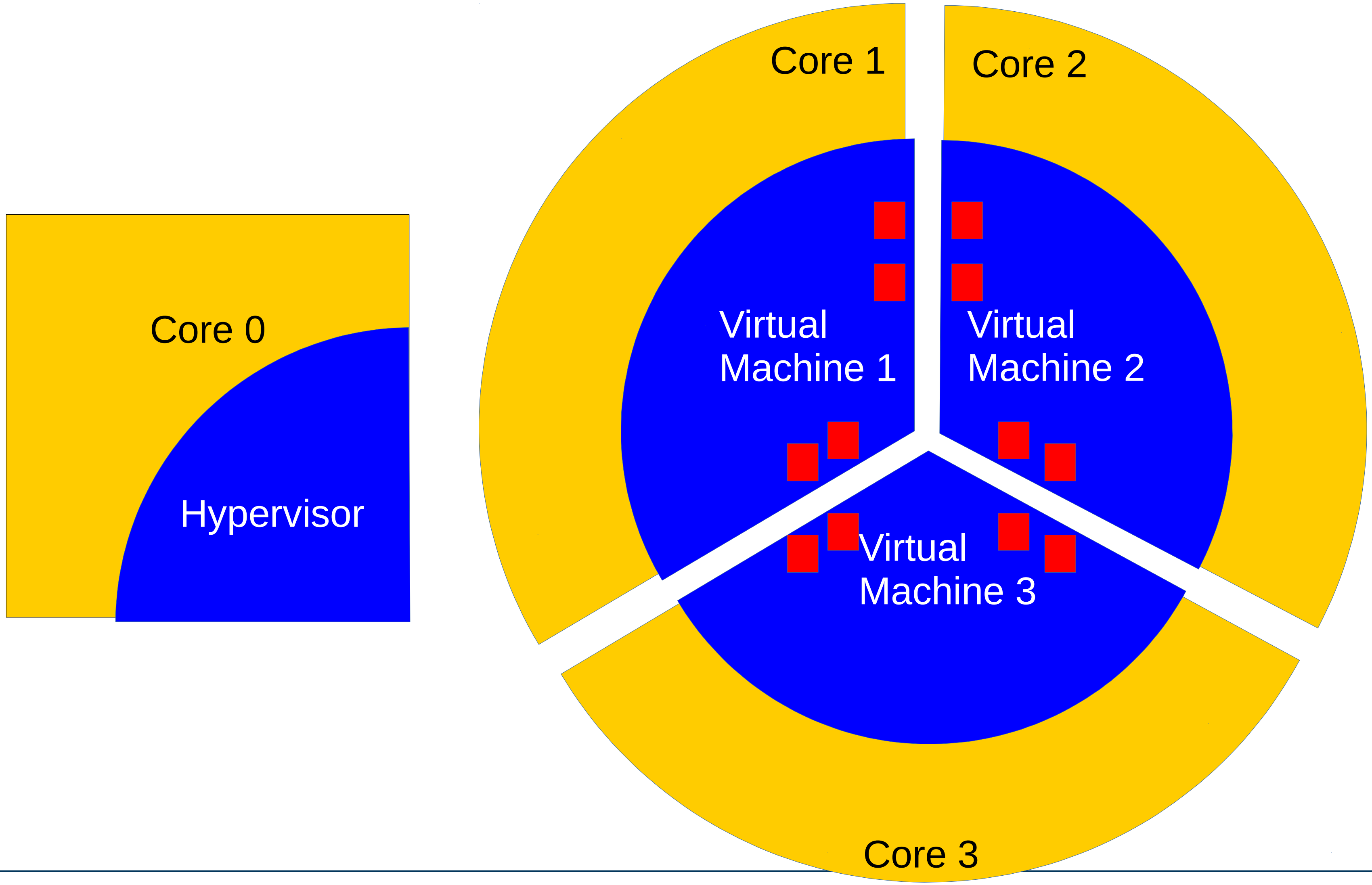
VM Cross Validation



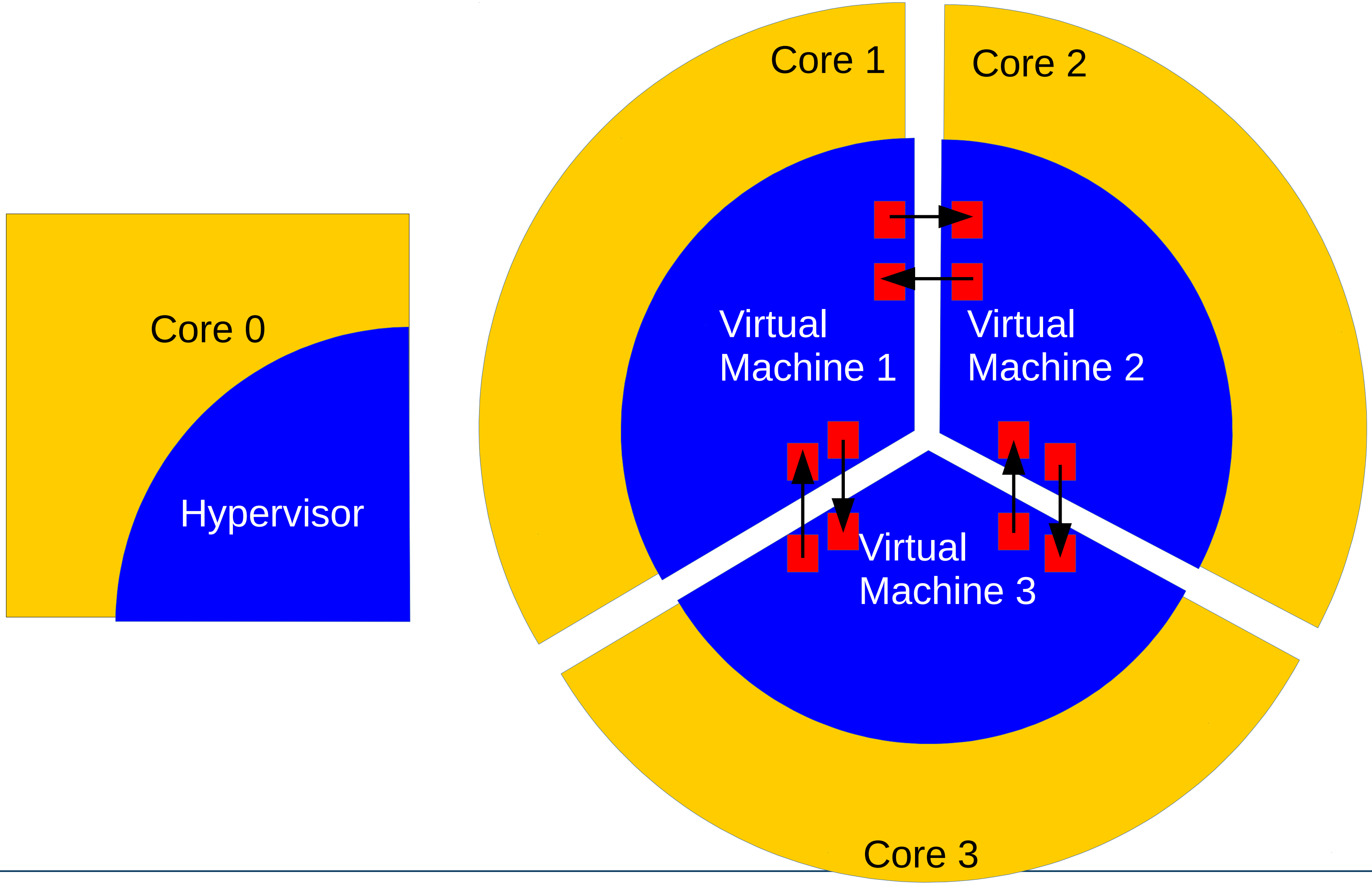
VM Cross Validation



VM Cross Validation



VM Cross Validation



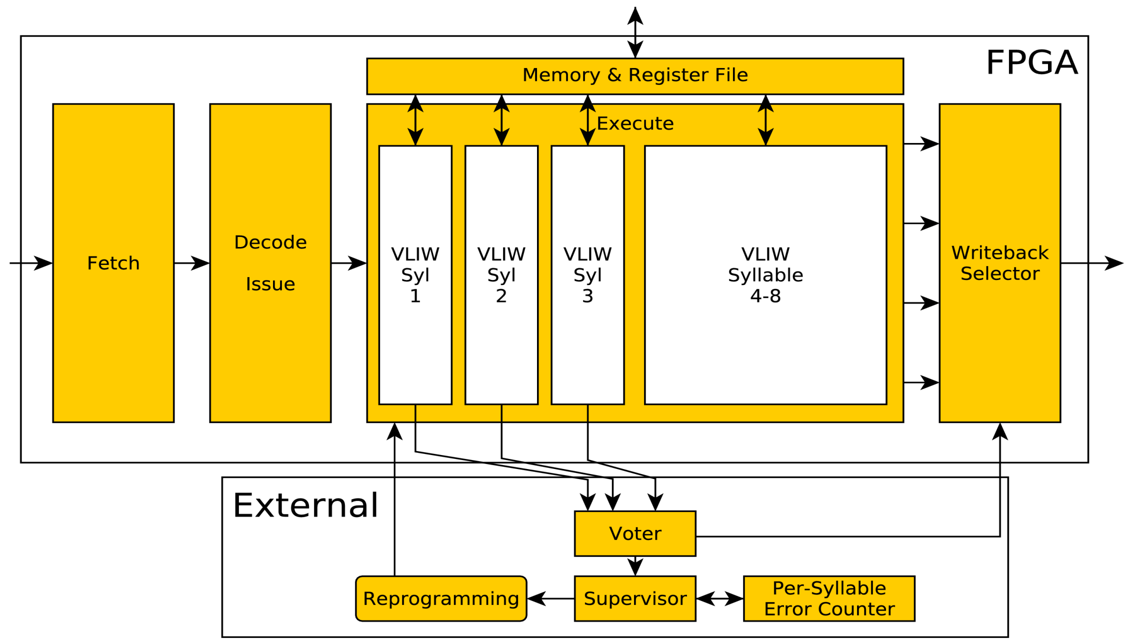
VLIW Based Voting

- 3+ Parallel Program Flows On-Demand
- Voting on Results OR Control Flow
- Dynamic Reconfiguration of Voting Setup

- Variants:
  - Static VLIW Based
  - Dynamic VLIW Based

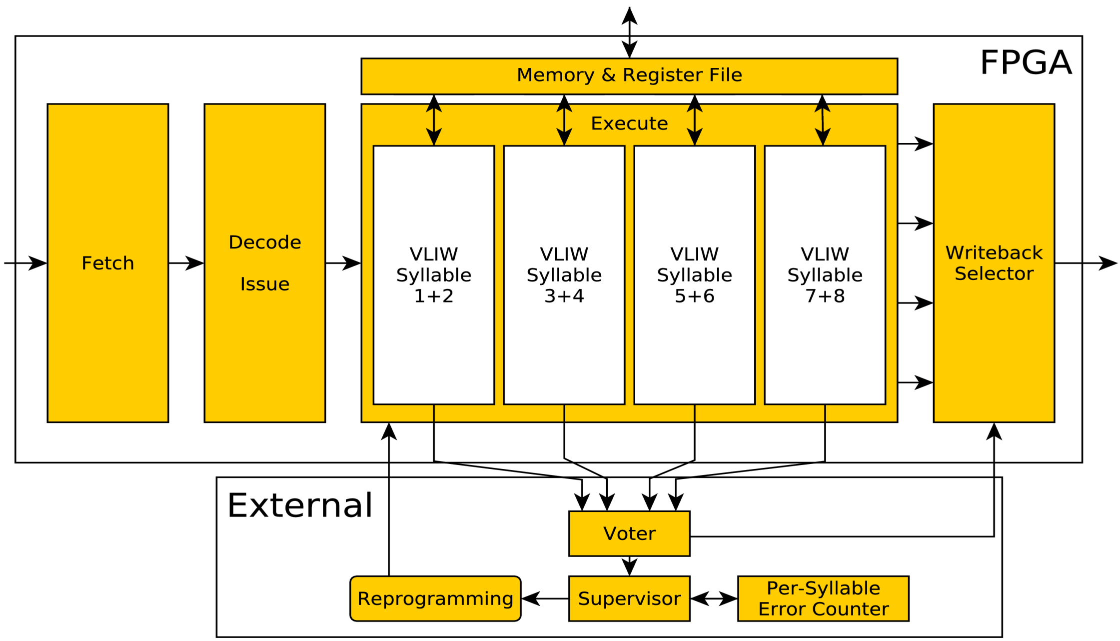
Towards Compute Dependability

- Low Complexity
- Low Power
- Low Cost
- Testability
- Self Validation
- Voting Accessibility
- Trade Performance vs Consistency using dynamic VLIW Issue Queue
- Autonomous Reconfiguration



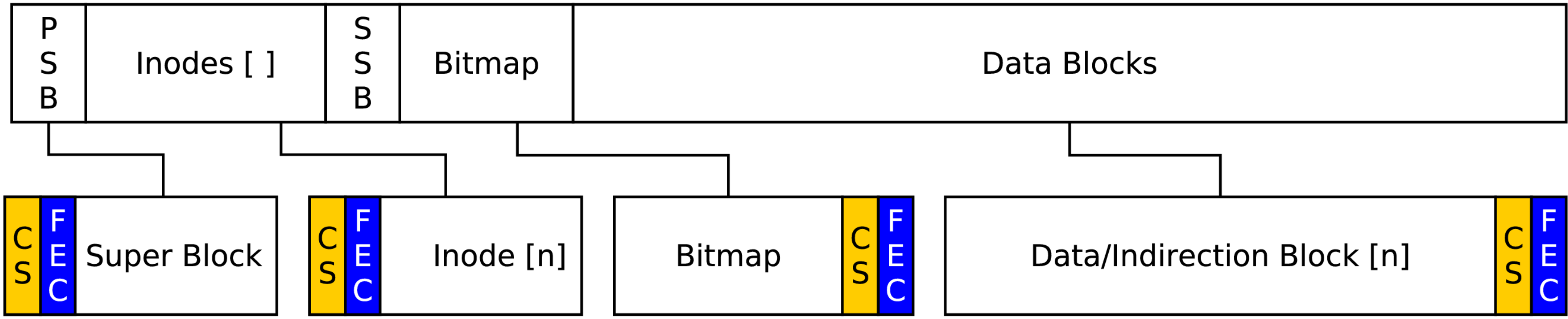
Towards Compute Dependability

- Low Complexity
- Low Power
- Low Cost
- Testability
- Self Validation
- Voting Accessibility
- Trade Performance vs Consistency using dynamic VLIW Issue Queue
- Autonomous Reconfiguration



FTRFS: A Fault-Tolerant Radiation-Robust File System for Space Use

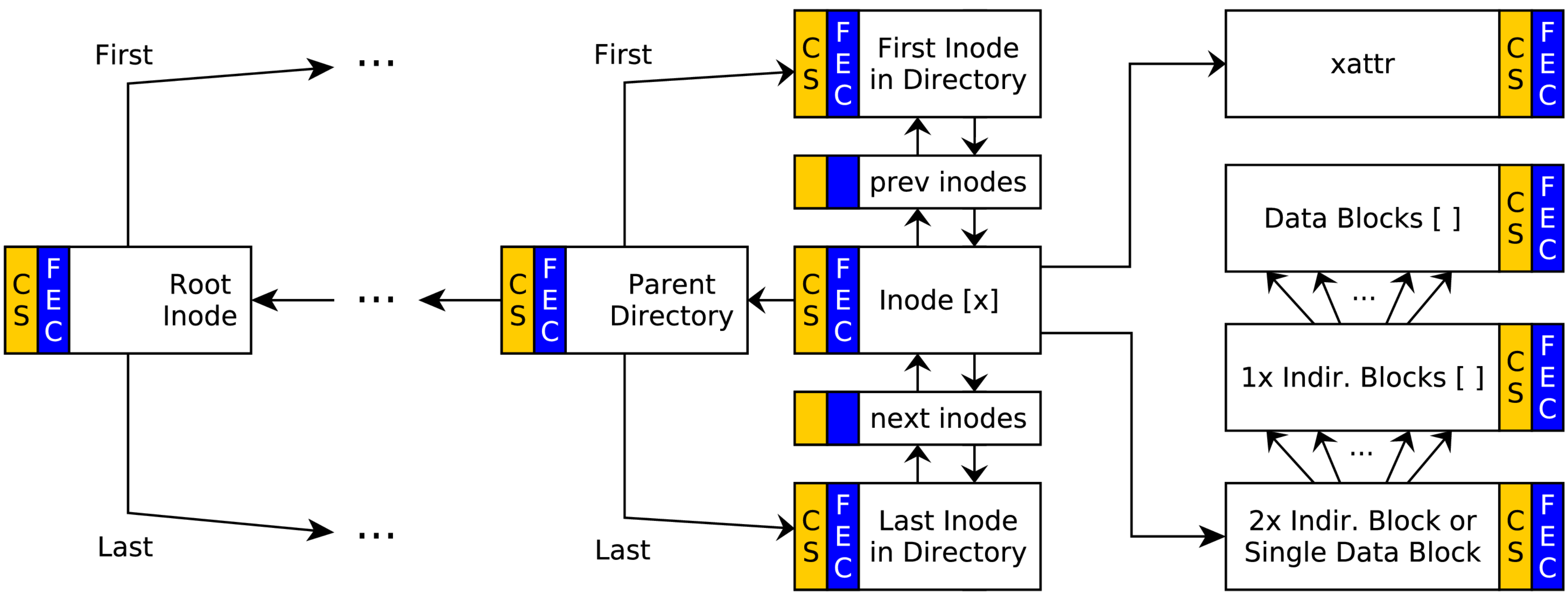
- **Dynamic erasure coding strength**
- EDAC using Reed Solomon
- CRC32 to enhance performance
- Memory protection



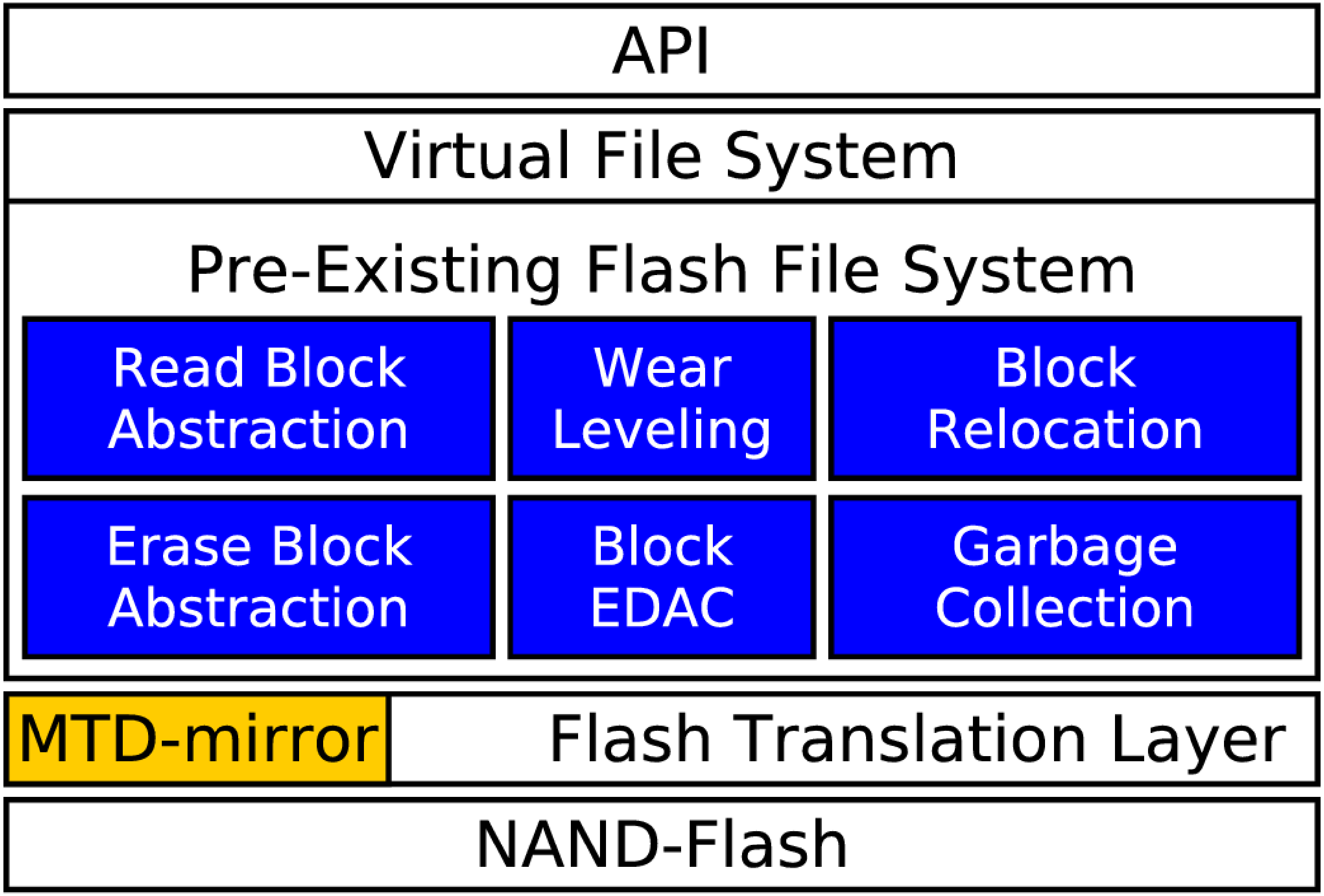
FTRFS: A Fault-Tolerant Radiation-Robust File System for Space Use

- Inspired by EXT2 with direct addressing
- **Fully featured POSIX-FS**

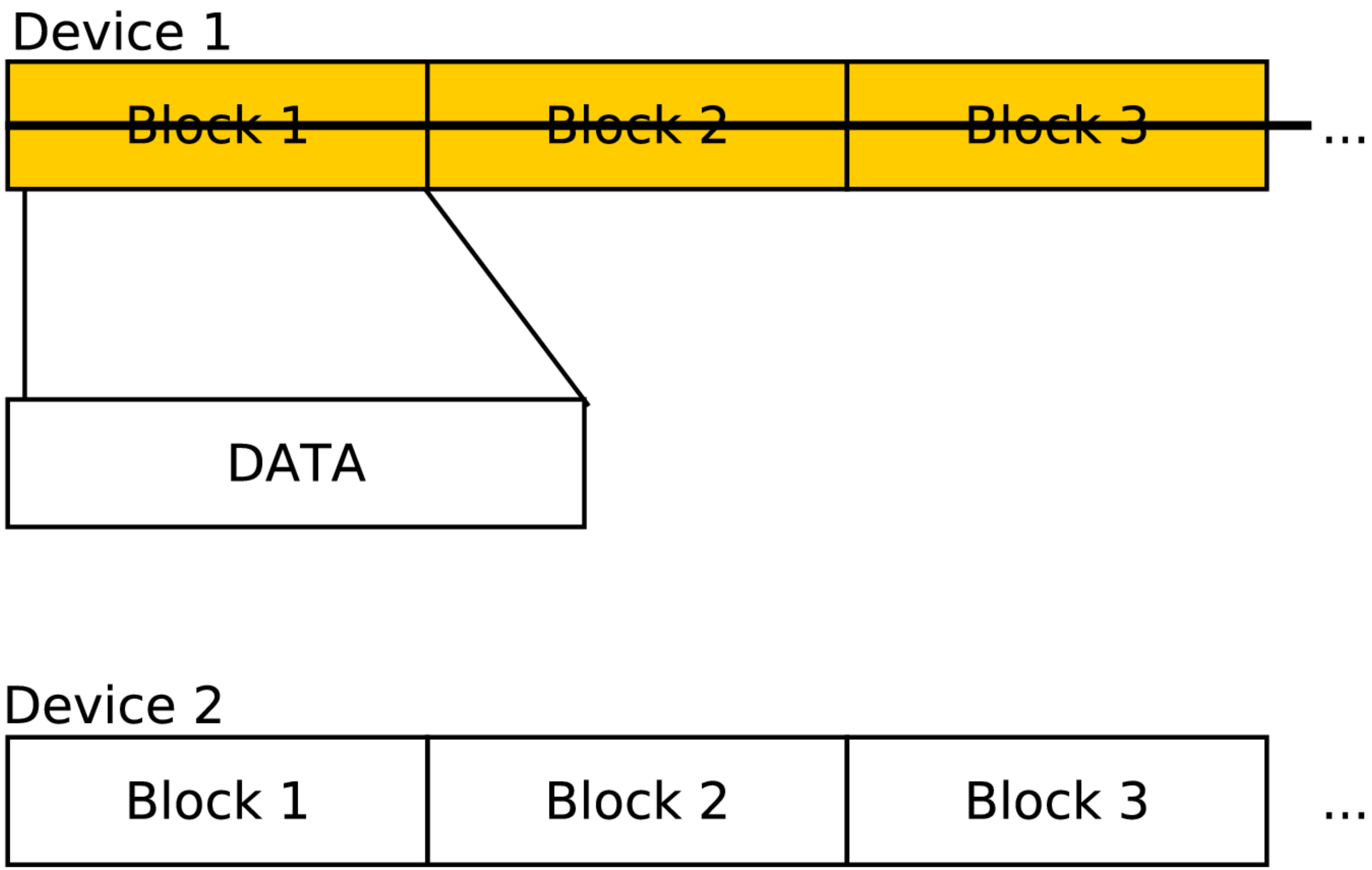
- Dynamic read/write protection (PRAMFS)
- Run-time static super block
- EDAC using RS(255, 128, t ≤ 63)
- CRC32 to enhance performance



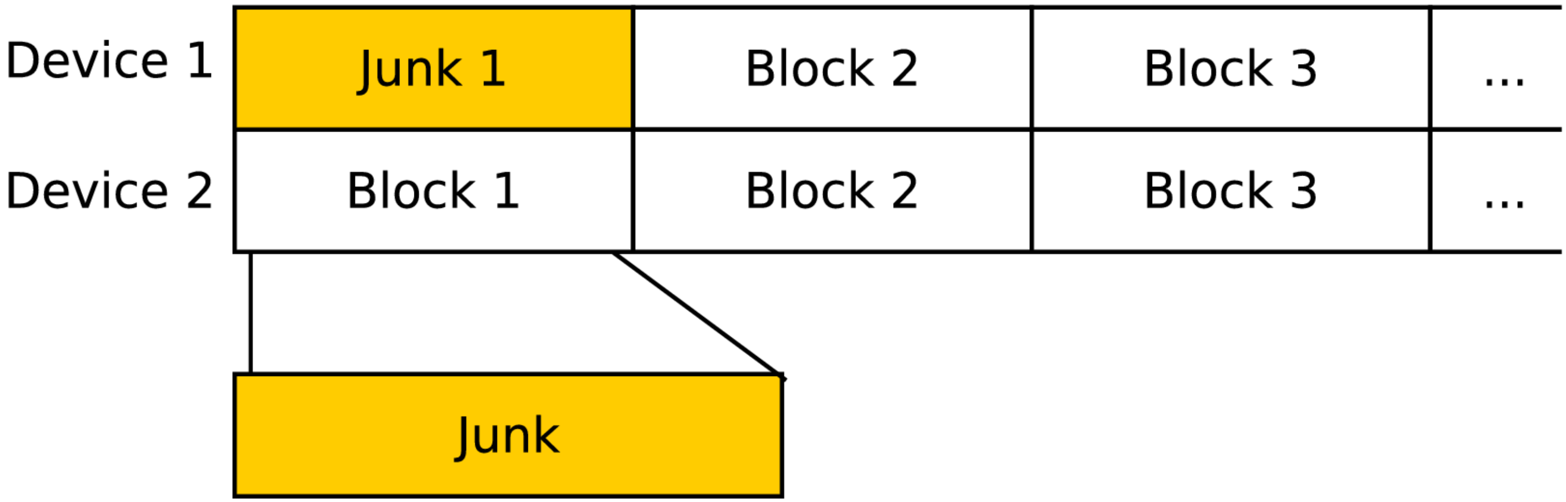
The MTD-Mirror Middleware



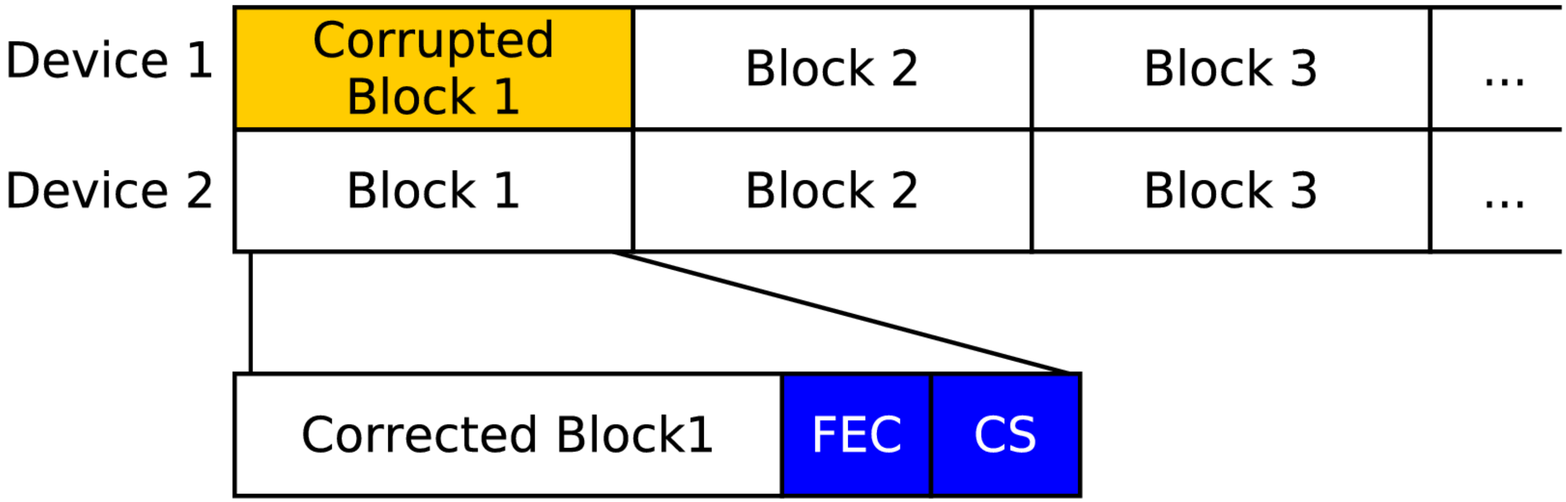
Why not RAID on NAND-Flash?



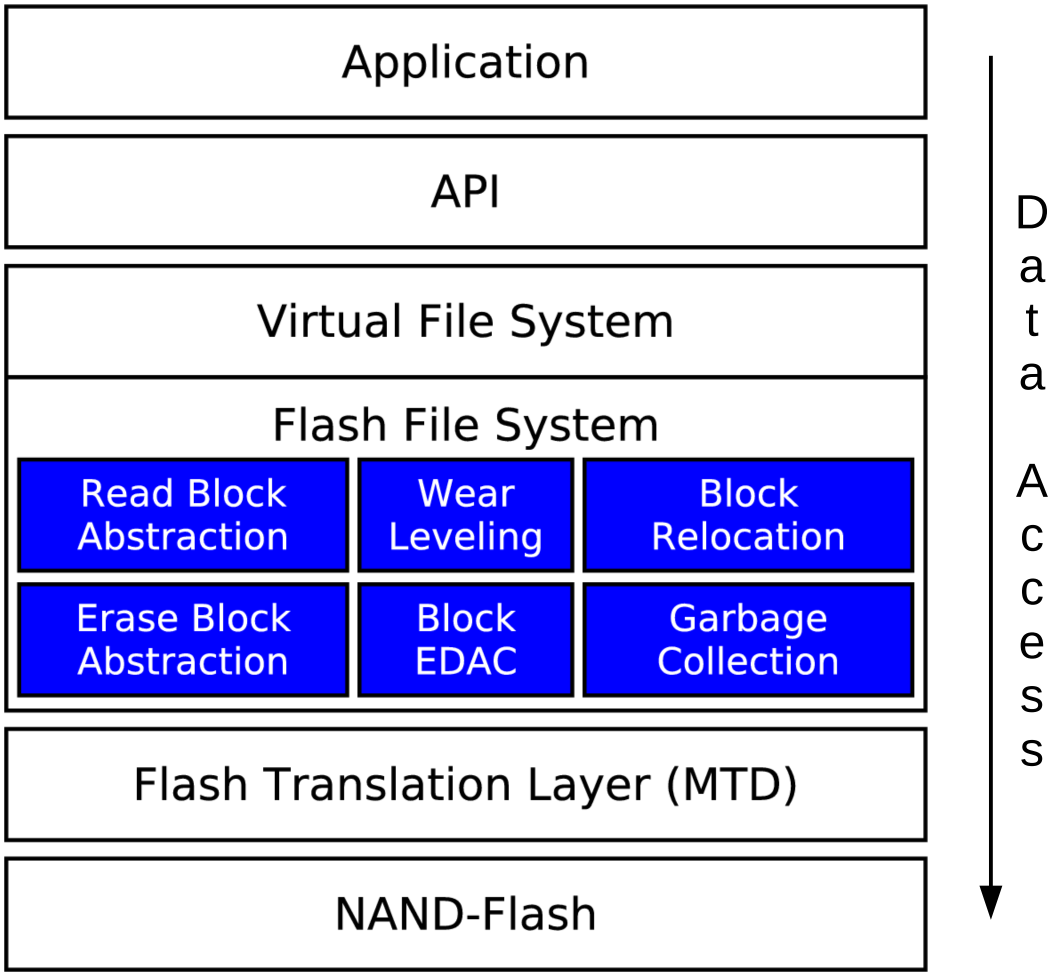
RAID and Data Integrity



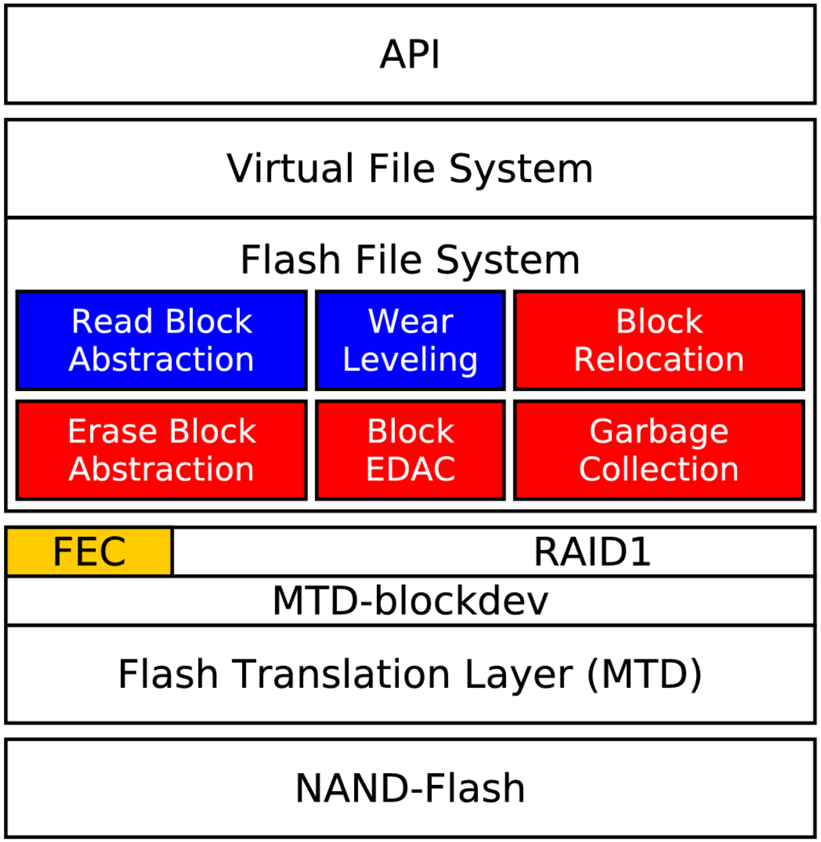
RAID1 with added Forward-Error-Correction



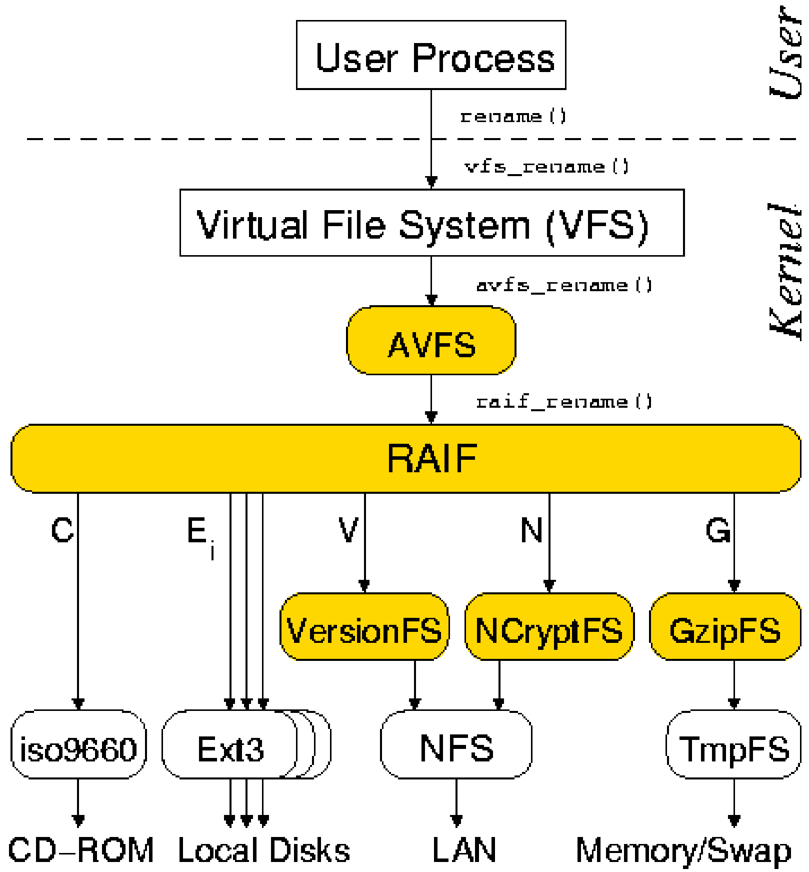
The Big Picture of Flash Access



Error Correcting RAID as an Overlay?

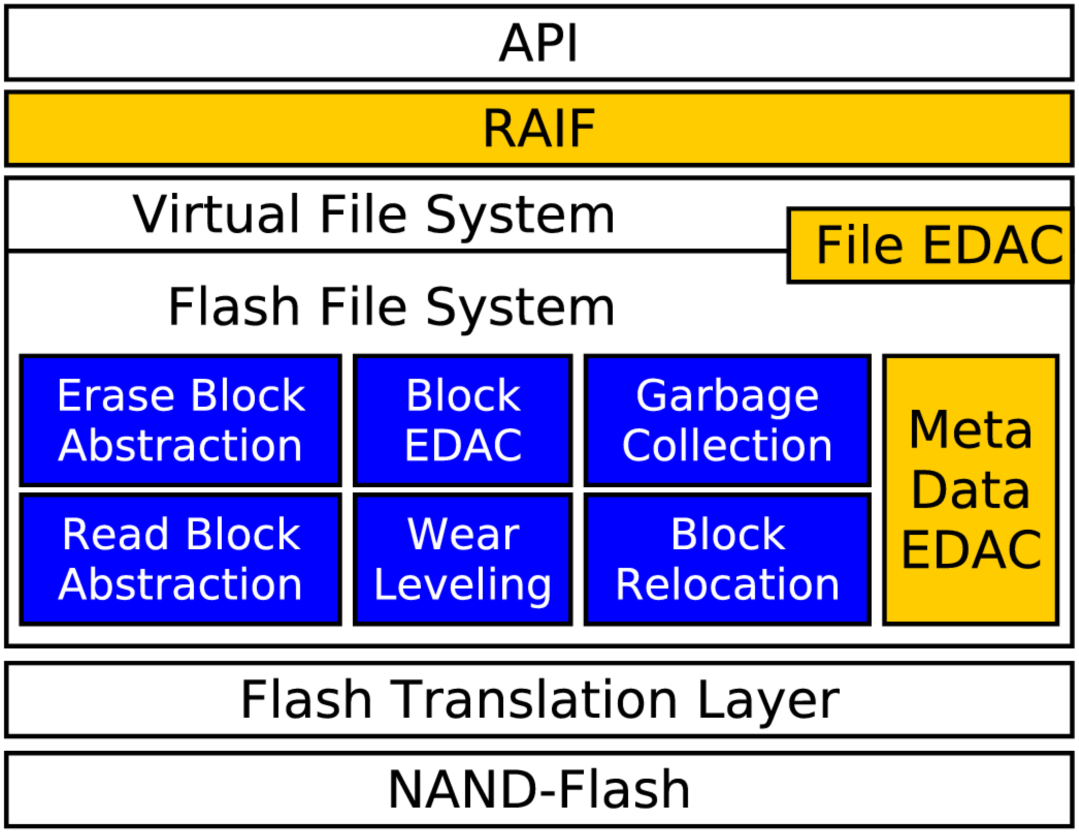


Redundant Array of Independent Filesystems (RAIF)

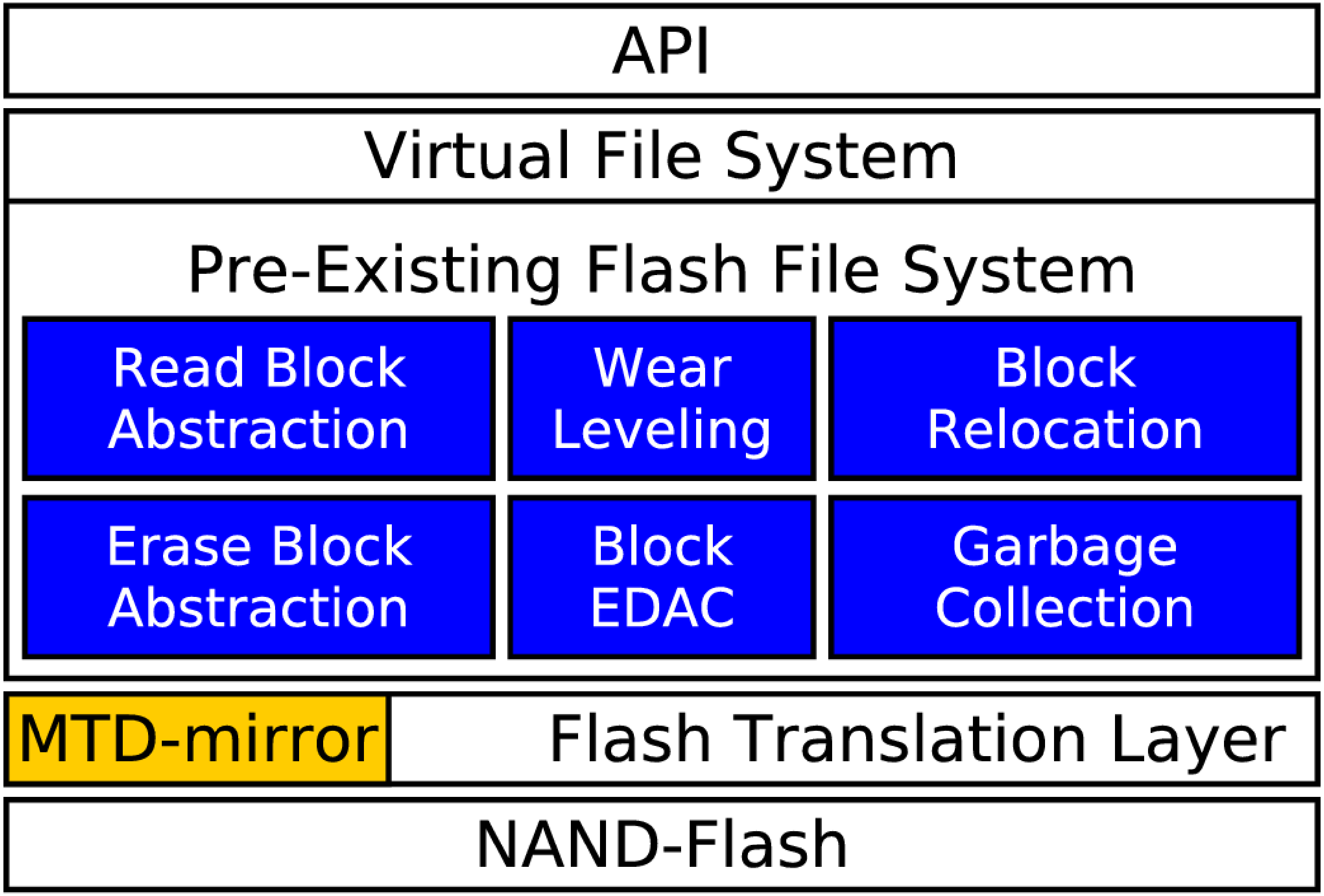


Source: Joukov et al., RAIF, 24th IEEE Conference on Mass Storage Systems and Technologies, 2007

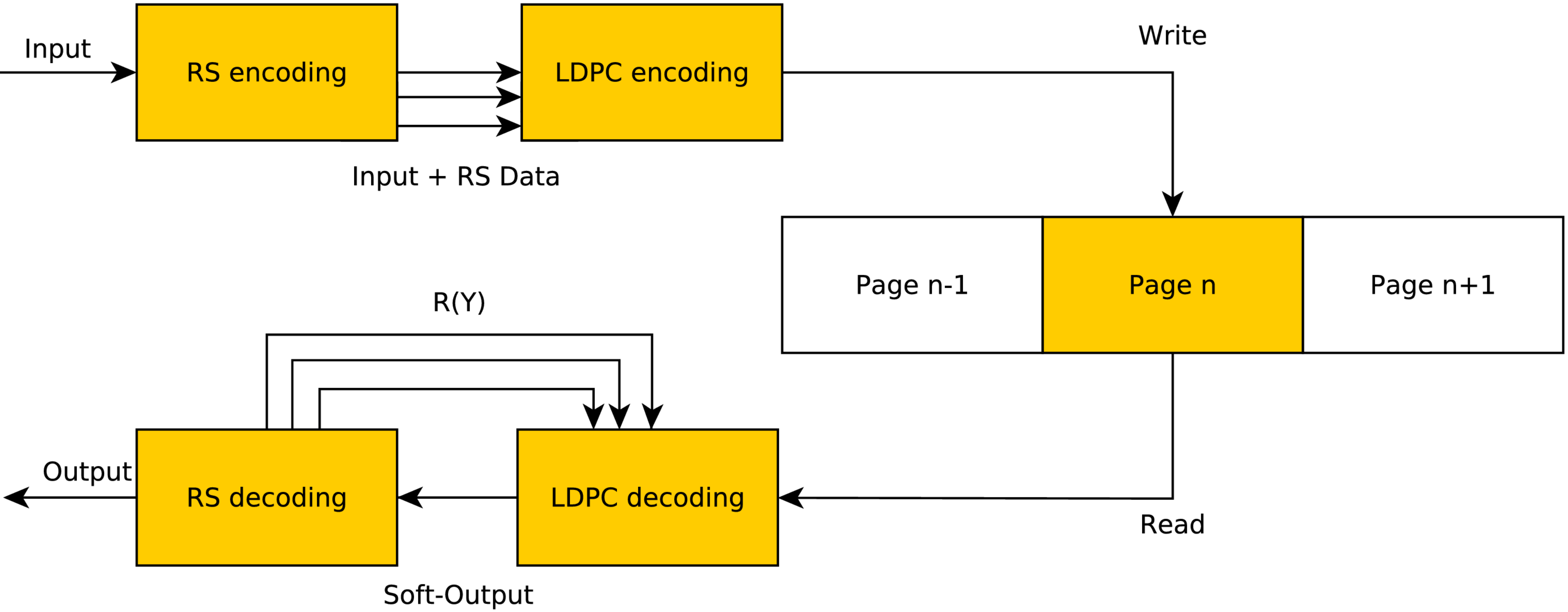
The Limitations of RAIF + FEC



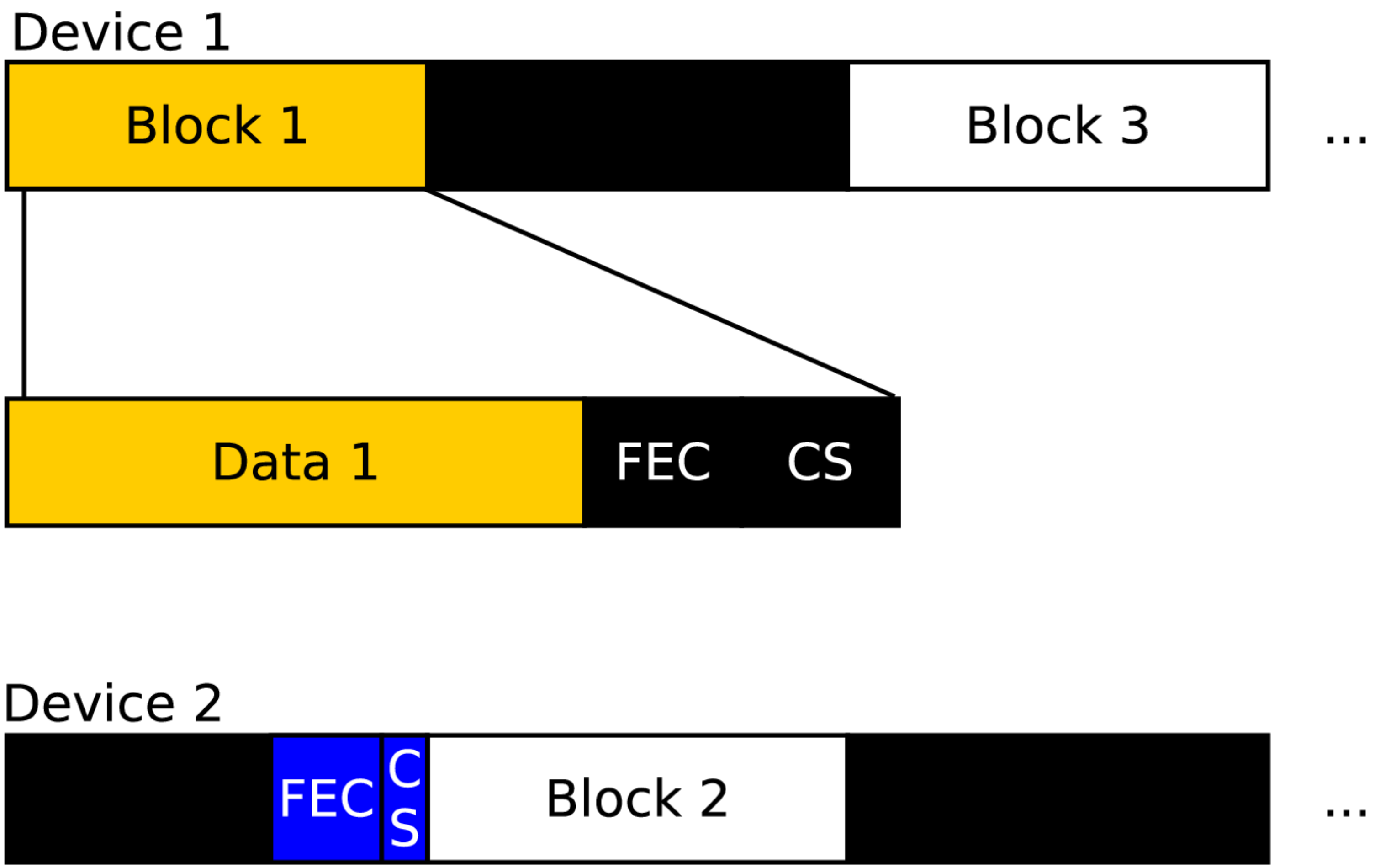
The MTD-Mirror Middleware



The MTD-Mirror Middleware



A Recoverable Worst-Case Scenario



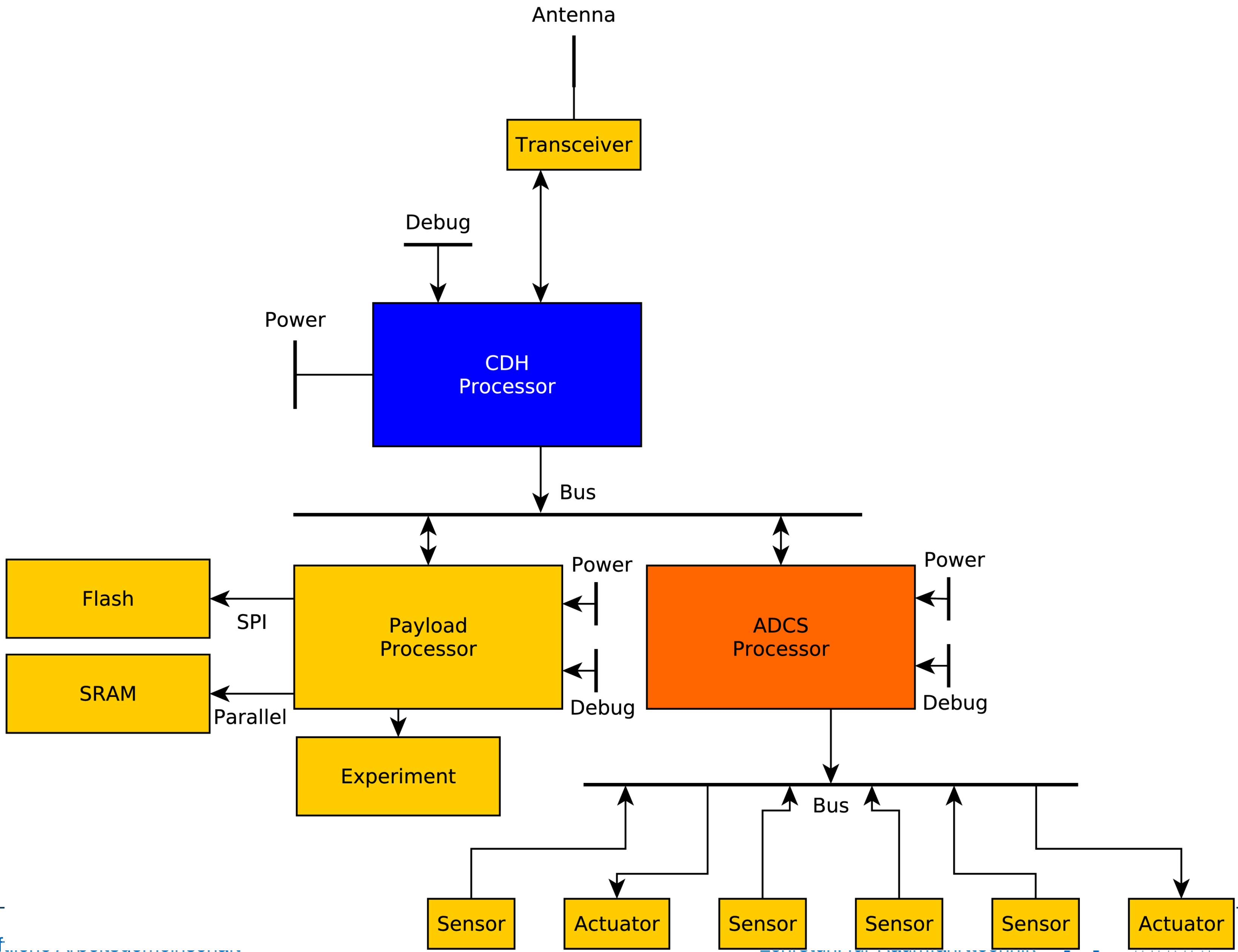
Application Scenario: MOVE-II

- Nanosatellite built at TUM
- 1U Modular CubeSat Bus
- 1U Scientific Payload

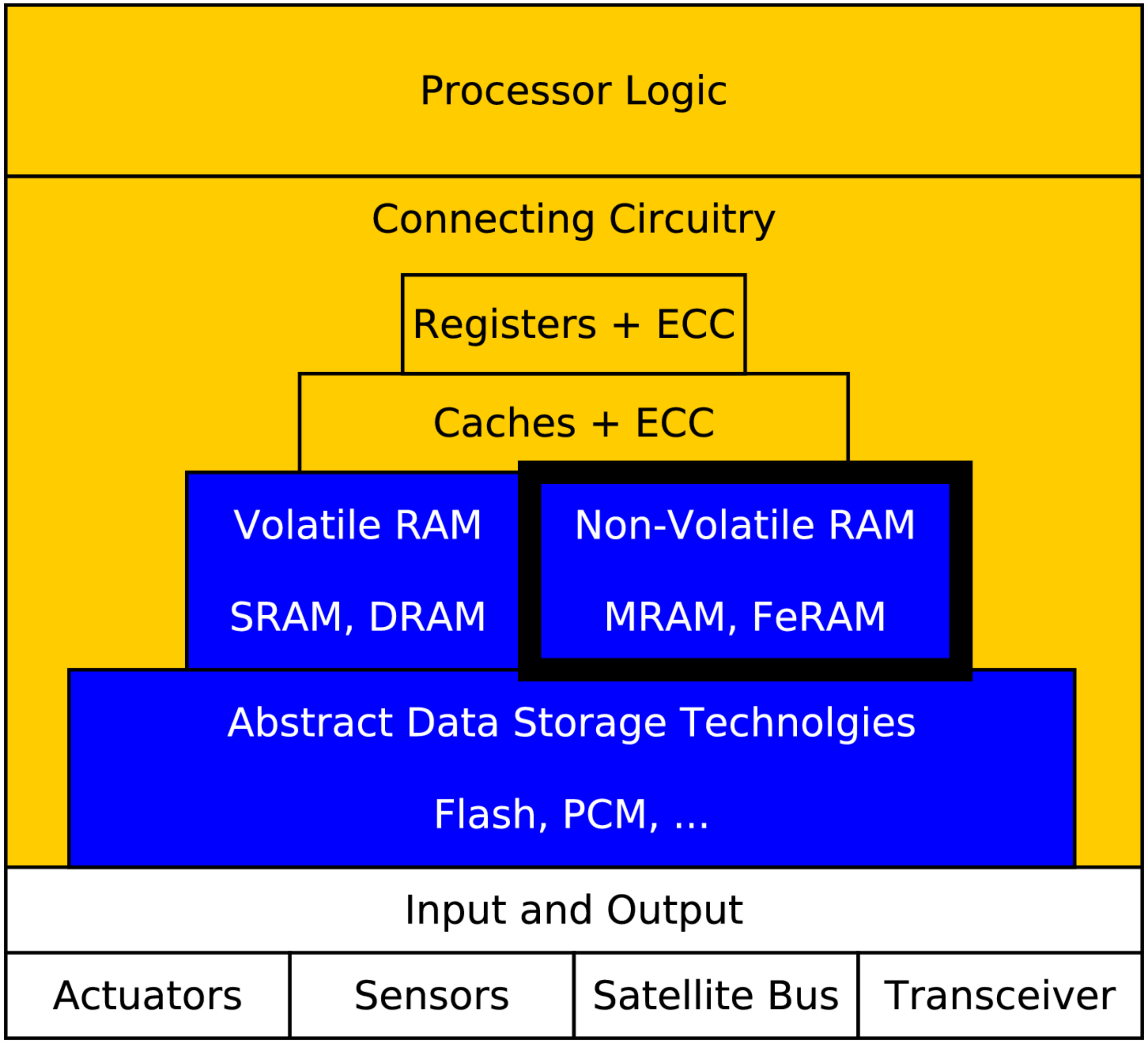
- 12W Average Power Budget
- 2W OBC Power Budget
- UHF/VHF + S-Band Com

- On-Board Computer
  - Application Processor
  - 32 MB ECC-SDRAM
  - 8MB MRAM for the OS
  - NAND-Flash for Payload Data
  - Size Optimized Standard OS

Why not just more REDUNDANCY?



Achieving System Dependability Through Software Measures



Achieving System Dependability Through Software Measures