

Dynamic Execution and Integrity Engine



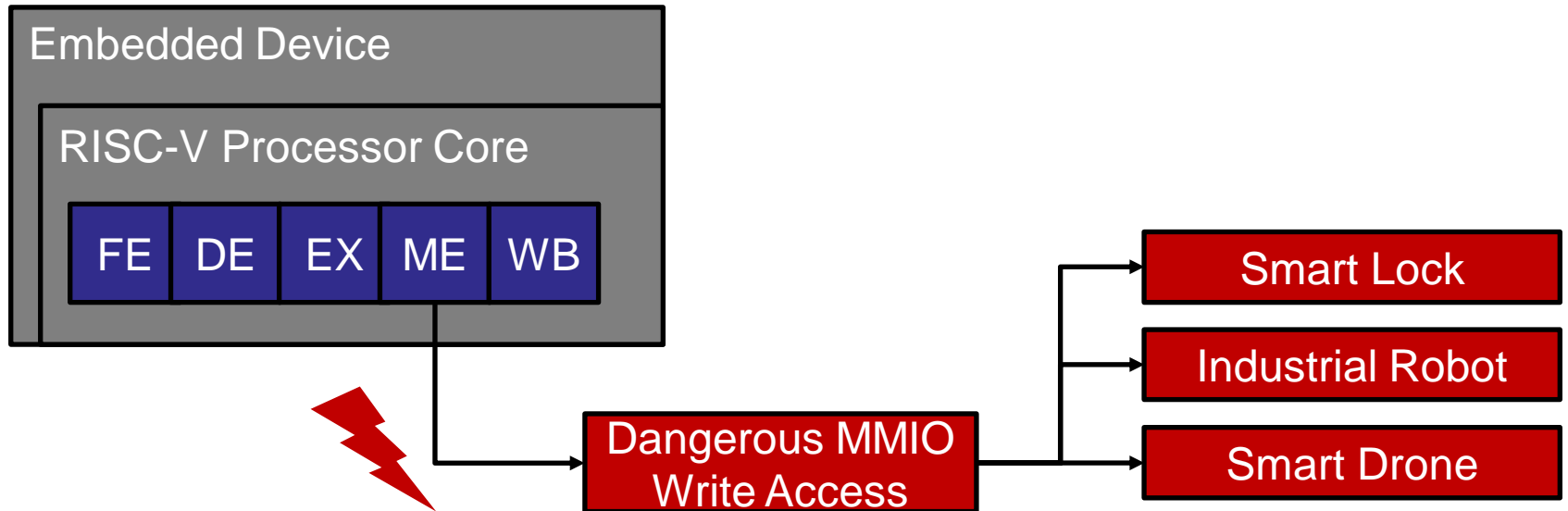
An IoT-Class Hardware Monitor for Real-Time Fine-Grained Control-Flow Integrity



Security Considerations

Scenario

Real-Time IoT device attached to security-critical memory-mapped devices



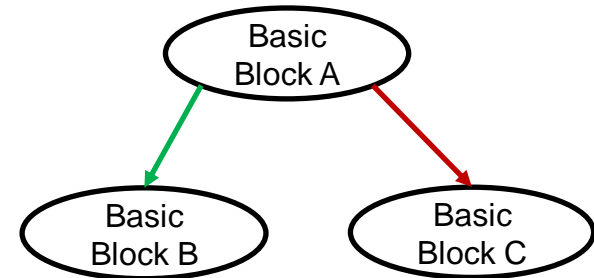
Security Considerations

Attacker Model

Code-Reuse Attacks: Attacker can alter control-flow (CF) instructions

- Tampering with return addresses
 - *Return-oriented Programming Attacks*
 - Return into LibC Attacks
- Tampering also with function-local Control Flow Instructions
 - *Jump-oriented Programming Attacks*

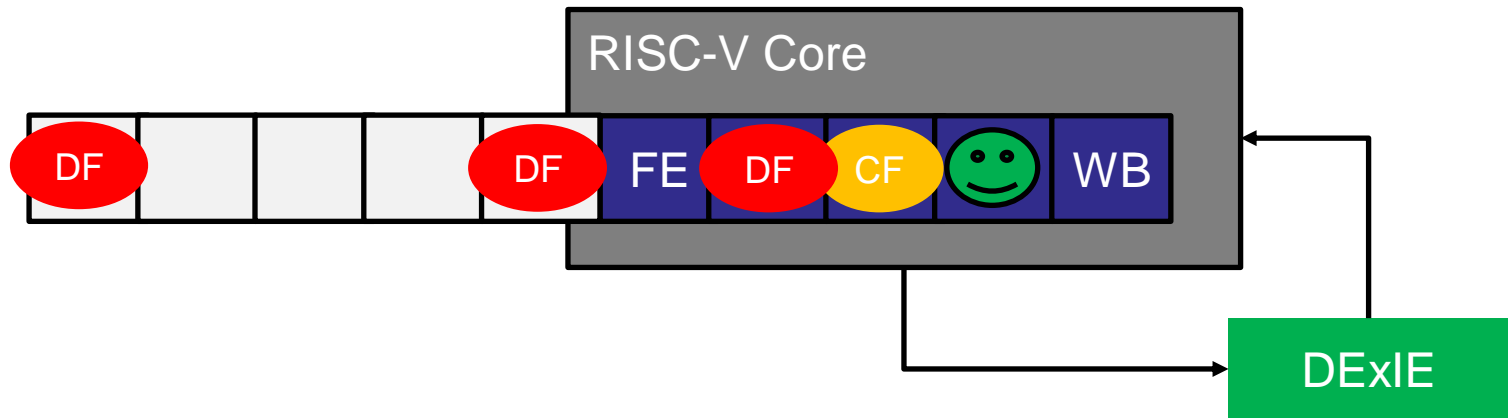
Stack		Shadow Stack
0x88	≠	0x80
0x184	=	0x184



Security Considerations

Guarantees

- DExIE will react to any CF instruction violating the currently active control flow constraints within a *guaranteed time interval*
- DExIE blocks *any* MMIO write following an illegal CF instruction




Security Considerations

Assumptions

Code-injection is not covered

- Can be mitigated by other measures: MMU, MPU, DEP, ROM, ...

```
int ATTACKER_main(){  
    float evil = 1.0;  
    if(evil){  
        destroy_world();  
    }  
    return 0;  
}
```



DExIE focuses on defense against code reuse attacks

Software / Hardware Partitioning

Software-Only vs. Hardware-Assisted vs. Hardware-Only

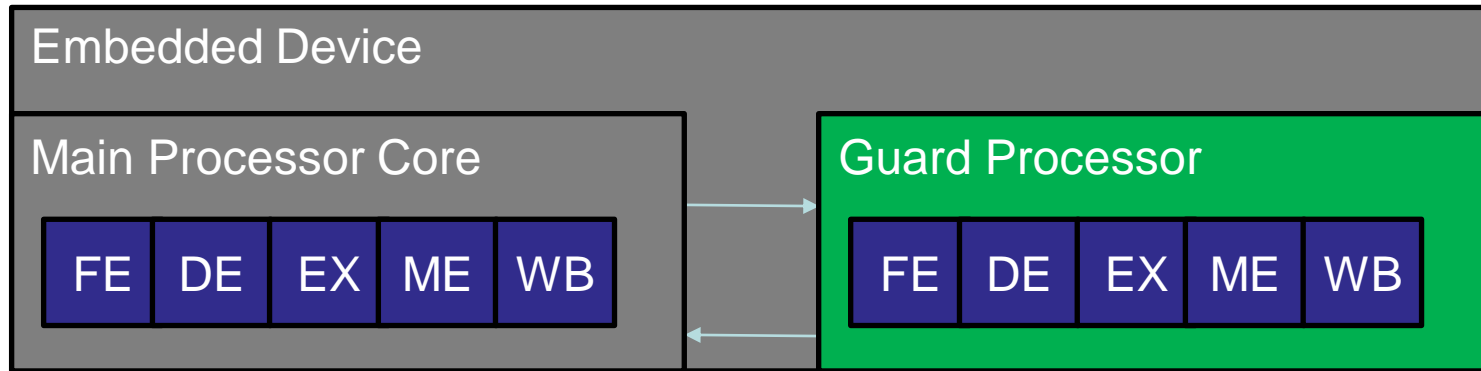
Instrument binary with extra instructions supervise CF instructions

Relatively easy, but high overhead (e.g., 2x slowdown)

```
int main() {  
    // Save return address duplicate or its hash to safe place  
    int b=0;  
    if(b) {  
        get(R);  
    }  
    // Compare return address duplicate  
    return 0;  
}
```

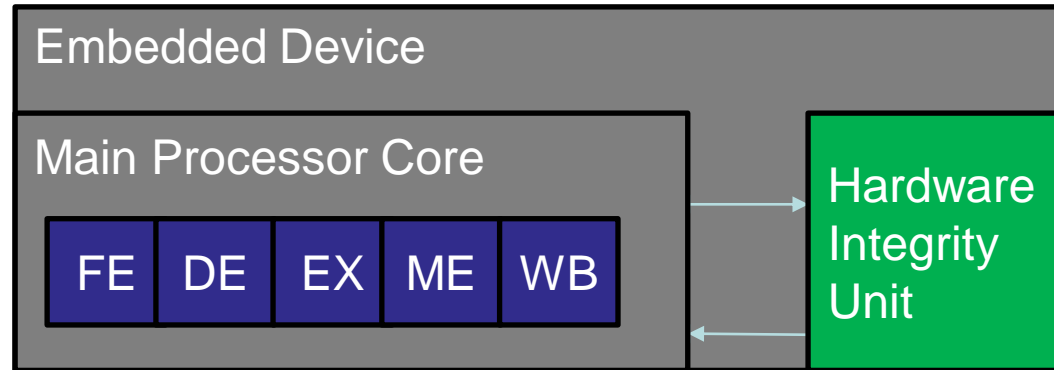
Software / Hardware Partitioning

Software-Only vs. Hardware-Assisted vs. Hardware-Only

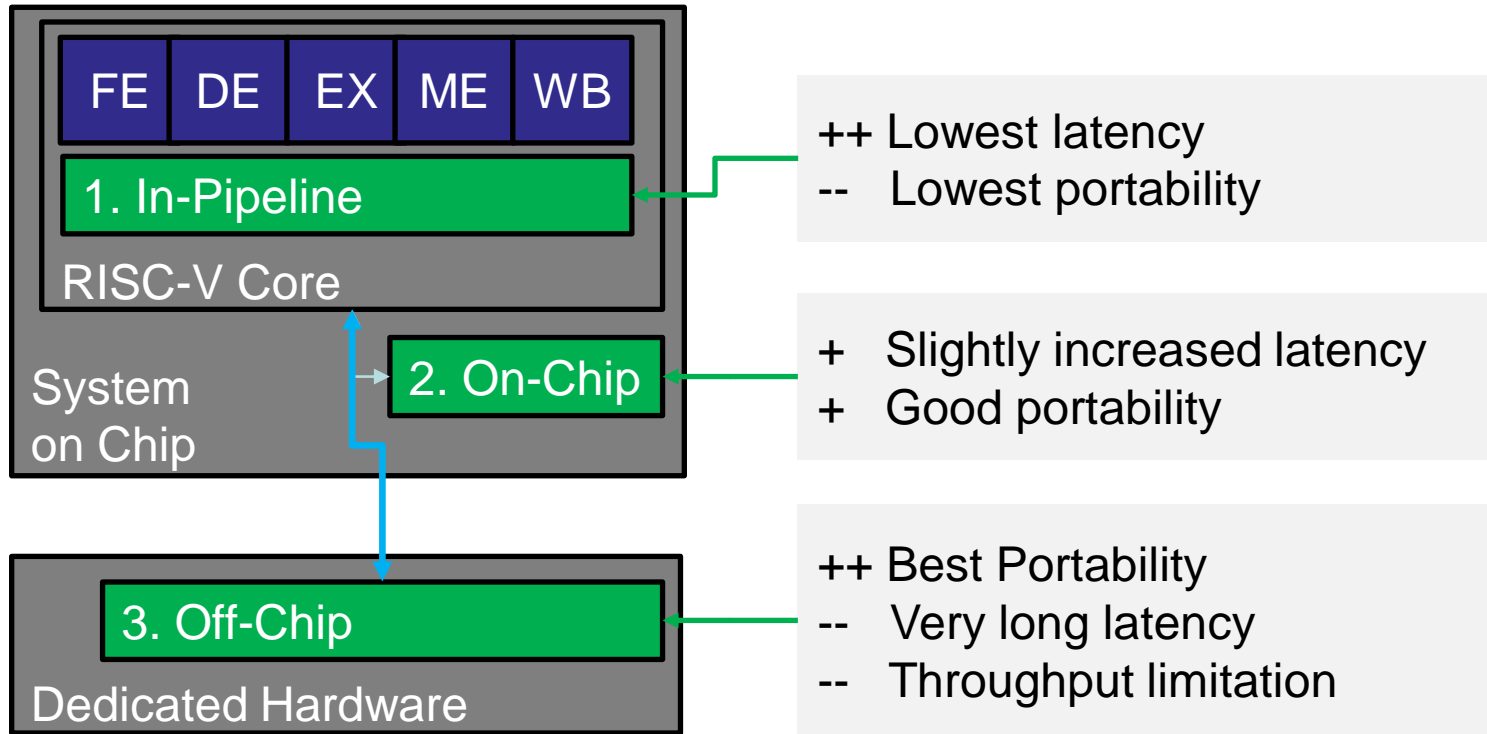


Software / Hardware Partitioning

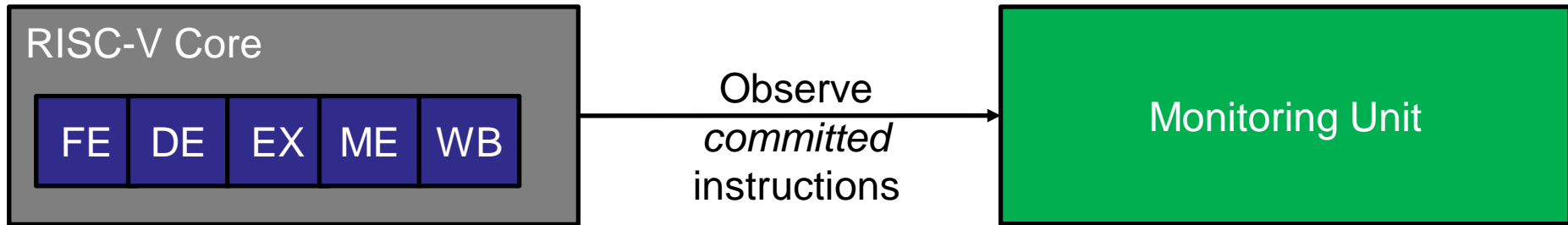
Software-Only vs. Hardware-Assisted vs. Hardware-Only



Architectural Location for Integrity Unit

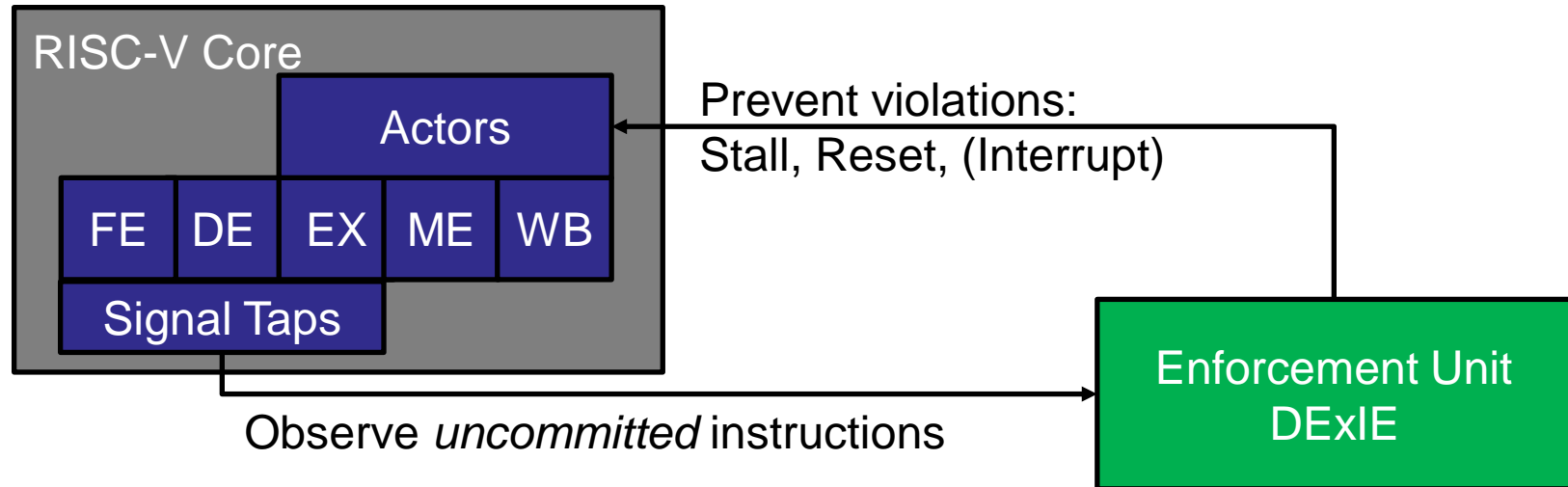


Monitoring vs. Enforcement



Monitoring: One-way flow of observations, unit passively *monitors* violations

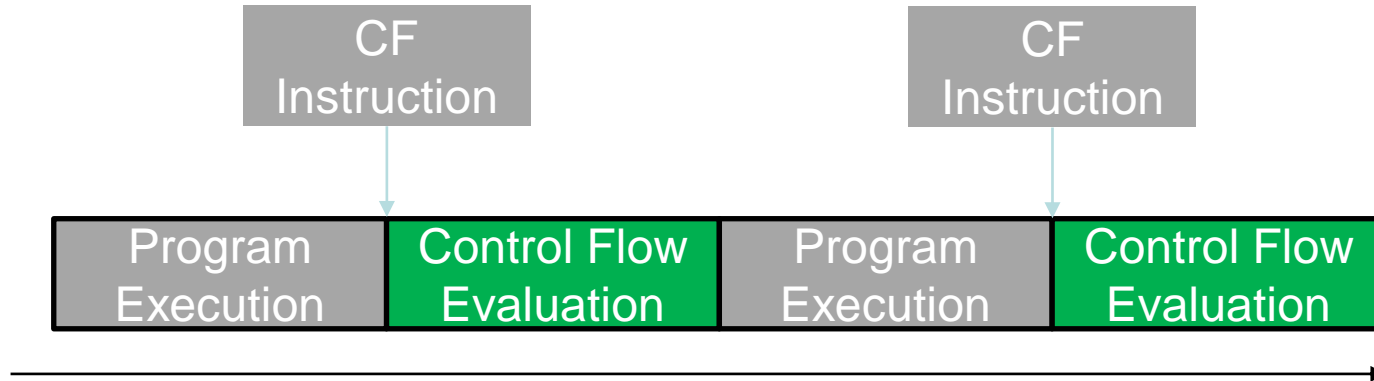
Monitoring vs. Enforcement



Enforcement: Closed loop operation,
unit can actively *prevent* violations from taking effect

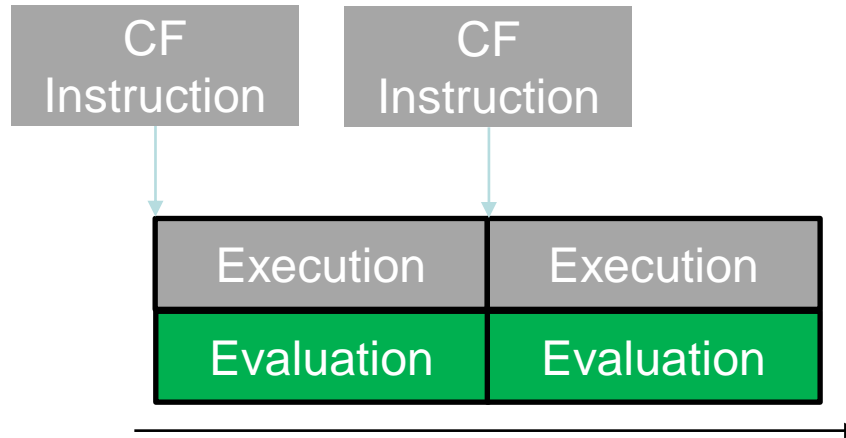
Parallelism of Execution and Checking

Serial vs. Parallel vs. Hybrid



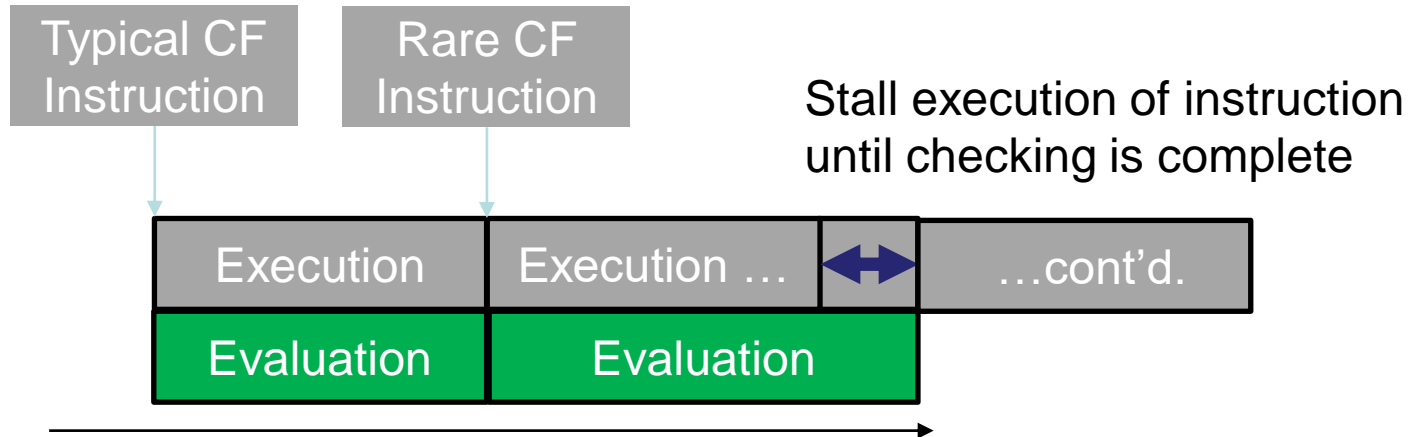
Parallelism of Execution and Checking

Serial vs. Parallel vs. Hybrid

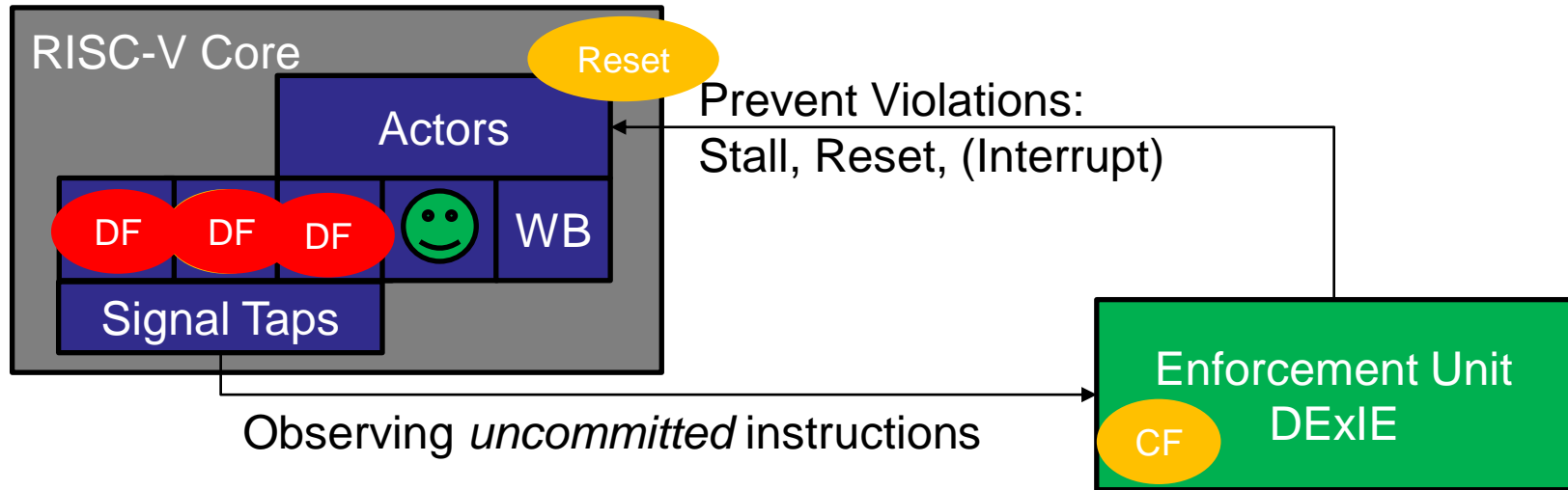


Parallelism of Execution and Checking

Serial vs. Parallel vs. Hybrid



Mitigating the Worst-Case Attack




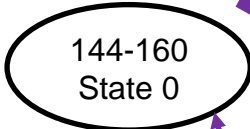
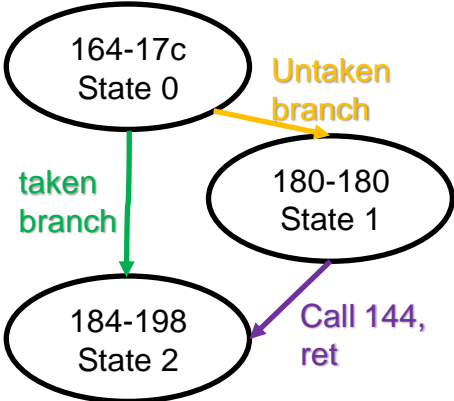
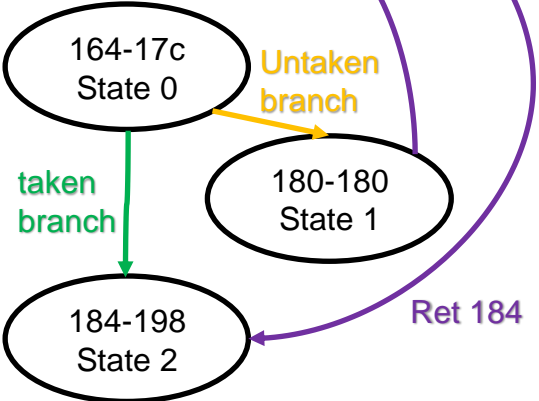
Major Challenges for DExIE

- Speed, Real Time: Avoid stalls and long critical path
- Guarantee Security: Be fast enough to block subsequent MMIO access
- Compatibility: Be compatible with multiple RISC-V cores
- Efficient: Compact memory layout supporting fast access

- Constraints memory storage:
 - Random access, no caching
 - Low latency (≤ 1 clock cycle)
 - Tight layout

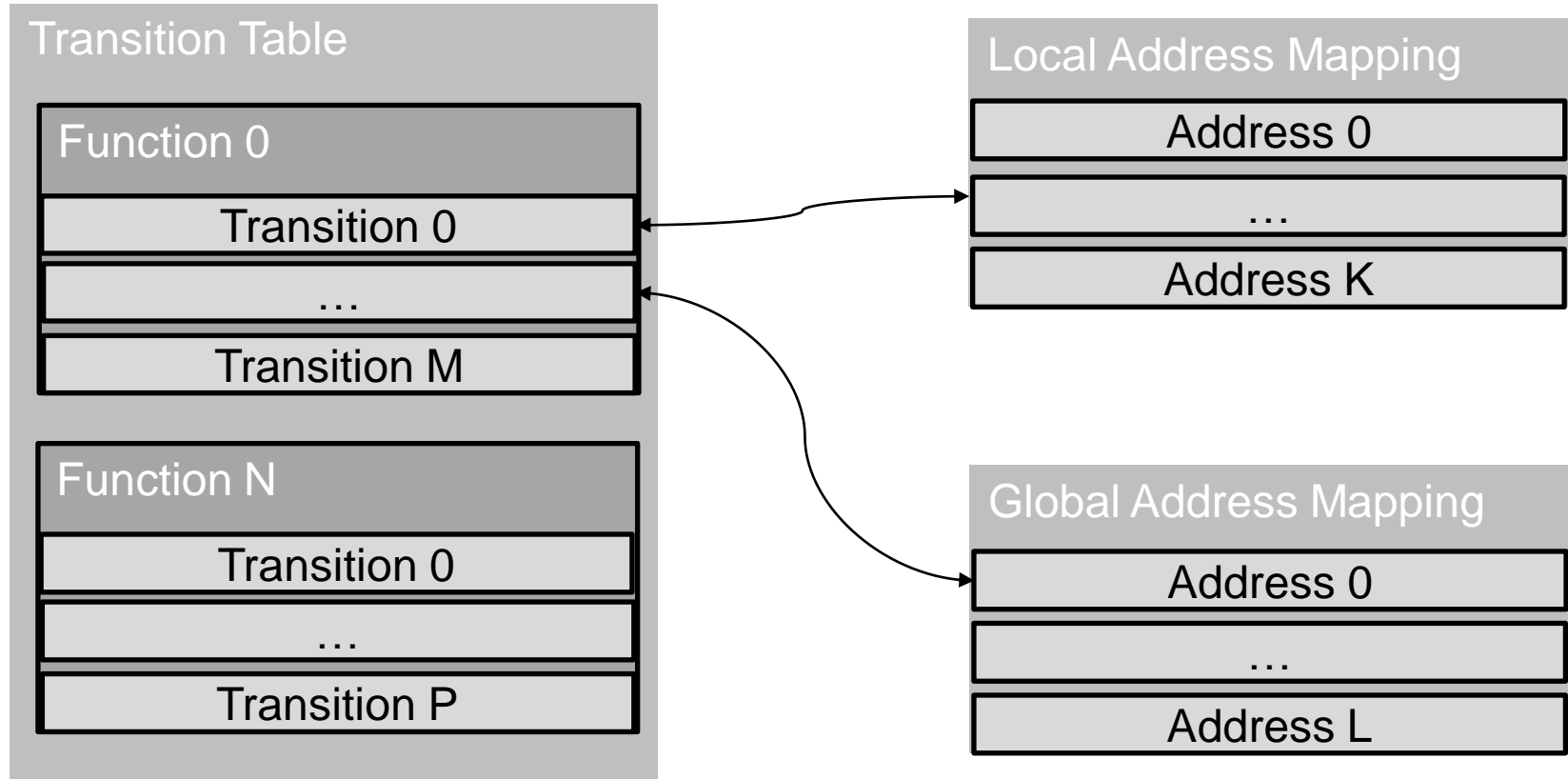
Solution: Encoding CF Constraints as FSMs

Hierarchical Approach

C Code	RISC-V Assembly	Function-local FSMs	Hierarchy of FSMs
<pre>int getR(){ int=42; return i; }</pre>	<pre>144: <getR> 144-15c: non CFI 160: ret</pre>		 <p>FSM-Aware Shadow Stack</p>
<pre>int main(){ int b=0; if(b) { get(R); } return 0; }</pre>	<pre>164: <main> 164-178: non CFI 17c: beqz 184 180: jal<getR> 184-194: non CFI 198: ret</pre>		

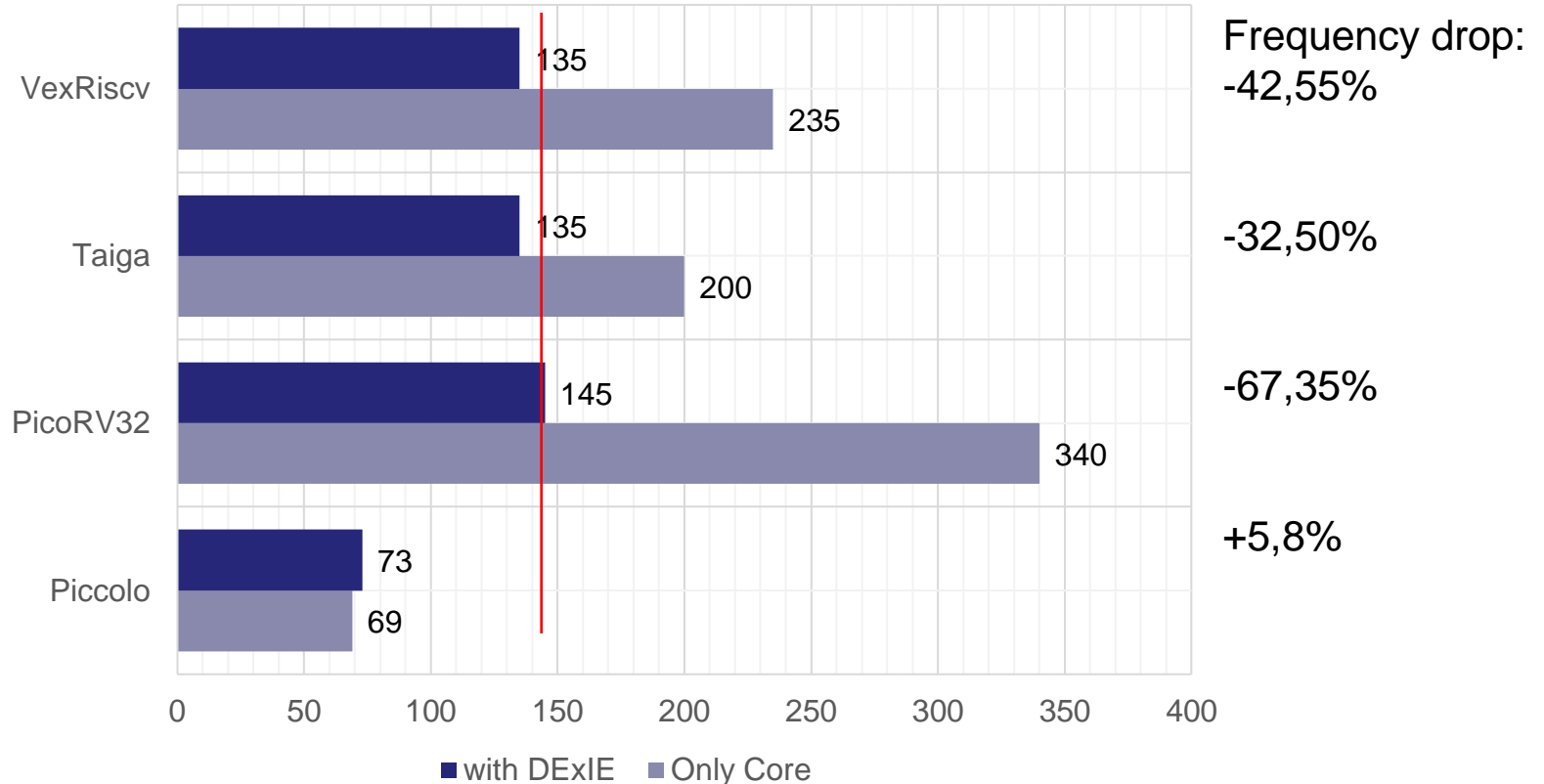
Solution: Encoding CF Constraints as FSMs

Memory Layout



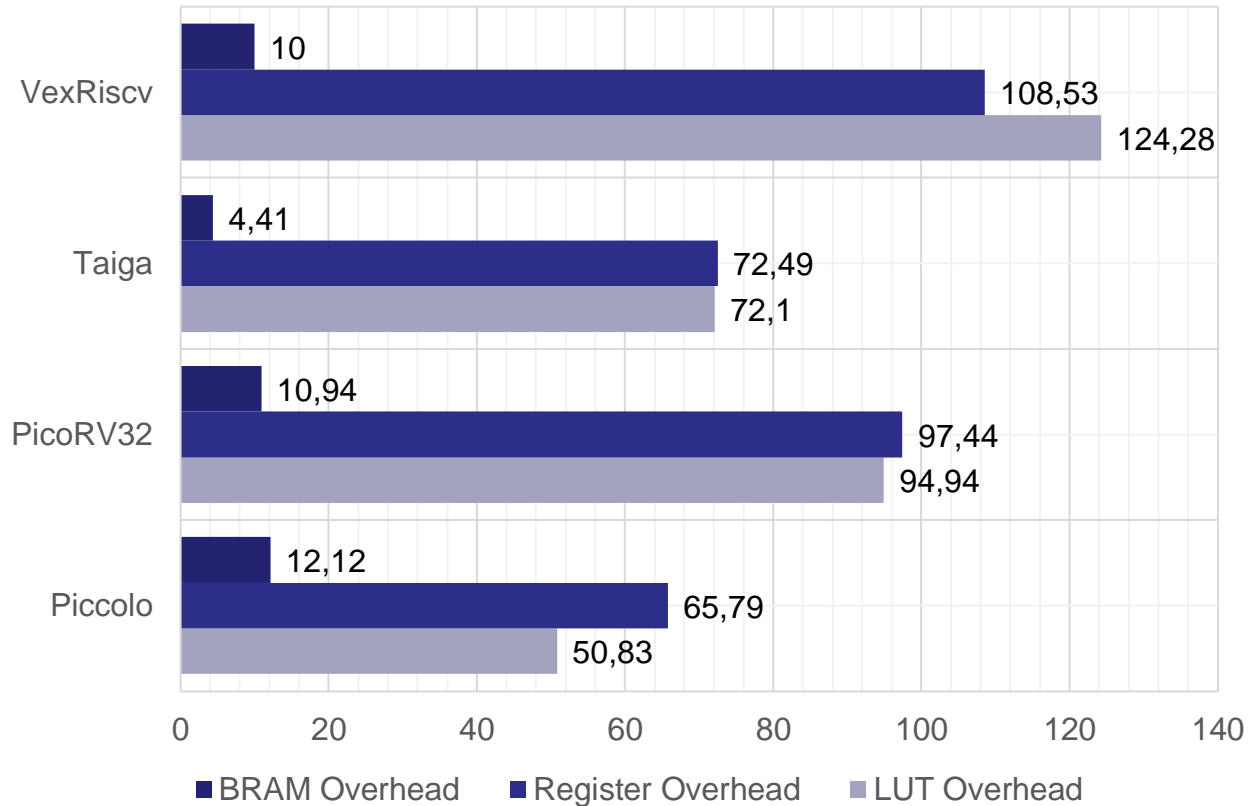
Evaluation: Clock Frequency Cost

Depends on Micro-Architecture of Monitored CPU



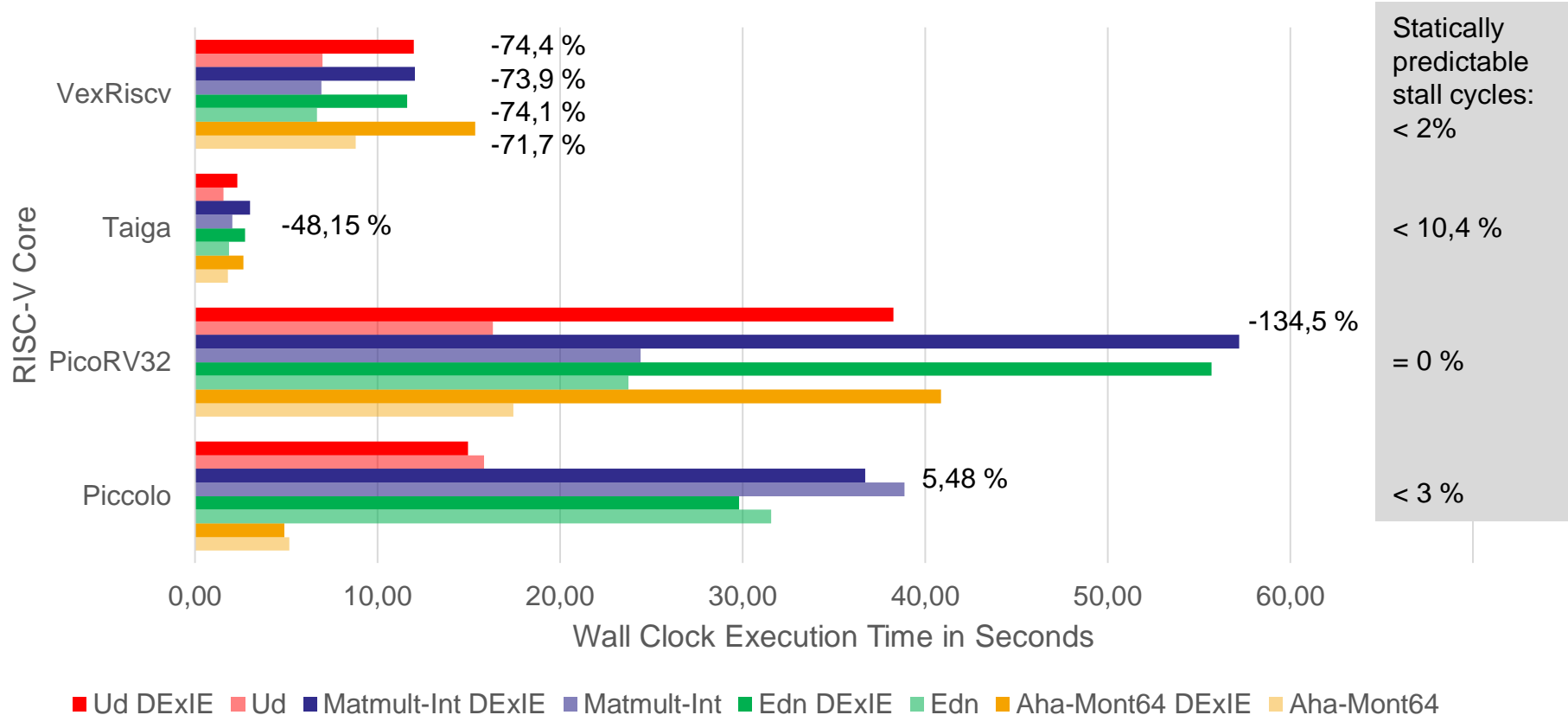
Evaluation: Area Cost

For different FPGA Resources



Evaluation: Execution Time Costs

Wall-Clock Execution Time



Conclusion

DExIE is ...

- ... real-time capable, as all stalls are statically predictable for WCET
- ... generally faster than SW-instrumented code (has less than 2x slowdown)
- ... portable and can easily be attached to different IoT-class processors
- ... smaller than the guard processor approach (which would use 2x area)
- ... flexible, as it can enforce CF at multiple granularities

- Questions?