Exploration of Fault Effects on Formal RISC-V Microarchitecture Models

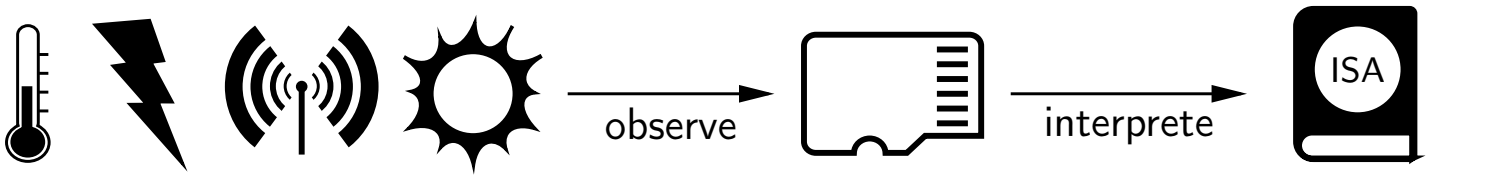
Simon Tollec[1], Mihail Asavoae[1], Damien Couroussé[2], Karine Heydemann[3] and Mathieu Jan[1]

[1]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
[2]Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France
[3]Sorbonne Univ., CNRS, LIP6, F-75005, Paris, France

FDTC, September 16, 2022

Fault Effects Characterization

Experimental characterization



Most common observed effects

- Instruction skips [Riviere, 2015] [Menu, 2020]
- Instruction replacement [Moro, 2013] [Trouchkine, 2020]
- Data corruption

Microarchitectural Fault Effects

Recent work

- Some effects cannot be *explained* at the ISA level
  - *Magic edges* [Proy, 2019]
- Some effects cannot be *captured* at the ISA level
  - *Forwarding* [Laurent, 2018]

ISA-level fault effect modeling

- Encompass broad set of effects
- Convenient for software analysis
- Abstract the implementation details

➔ Not sufficient to fully understand potential of fault injection

Contribution

Automated formal modeling of HW and SW

➔ For exploring microarchitectural fault effects on SW security

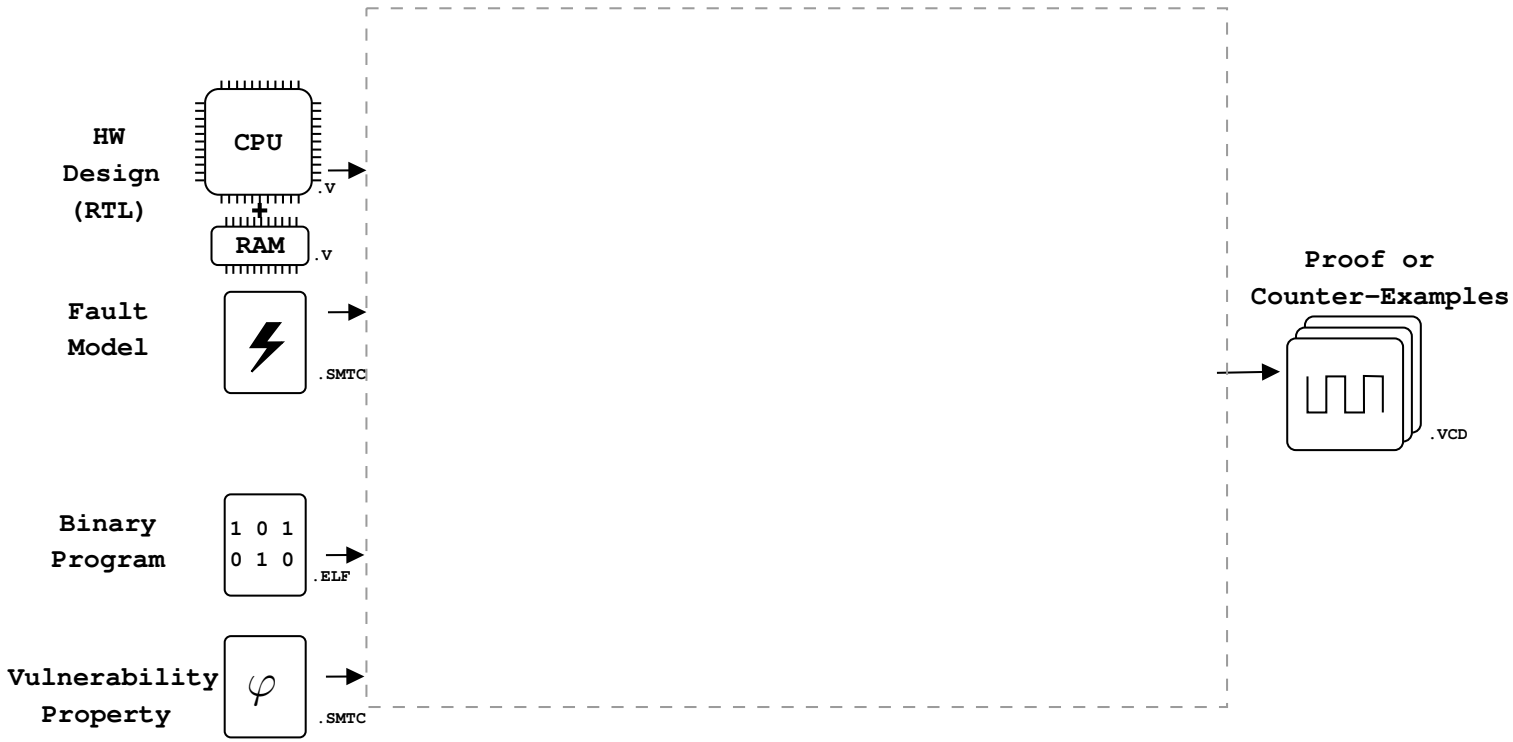
➔ For analyzing the robustness of HW or SW countermeasures

Why using formal methods, e.g., model checking?

- Give counterexamples or a proof
- Verification process guided toward counter-examples
- Handle complex properties in various logics

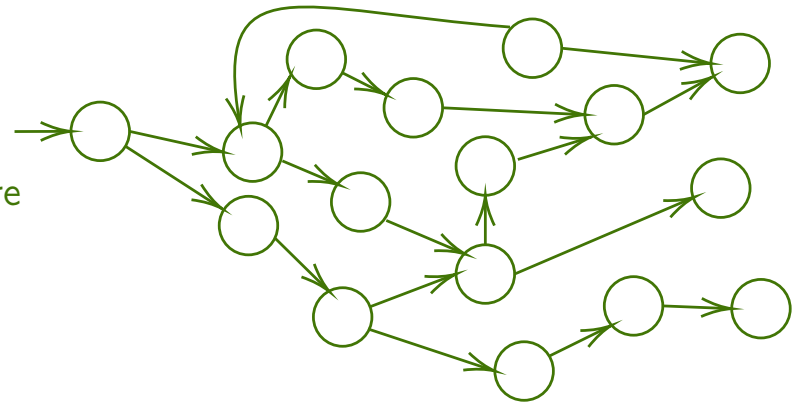
Workflow: Modeling Steps

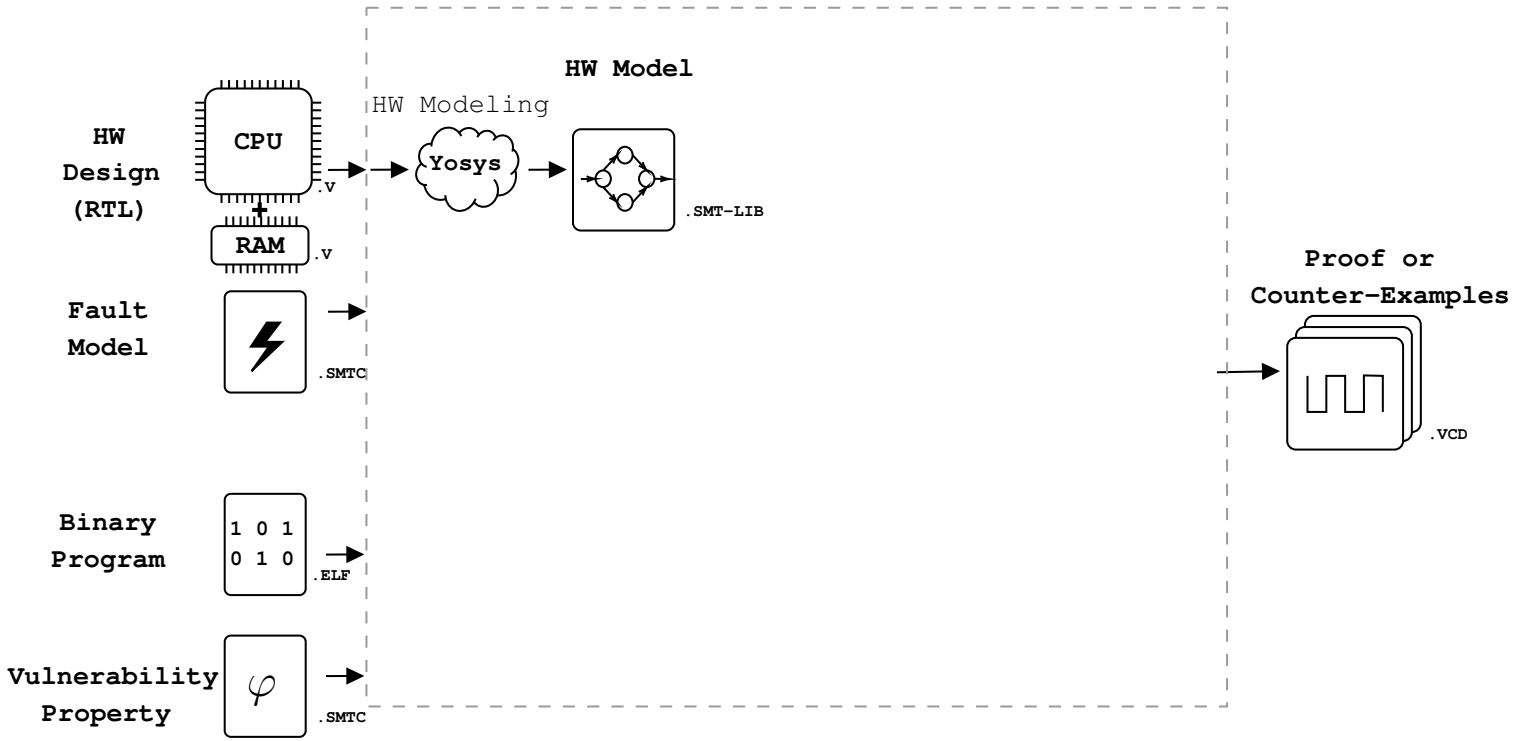
Inputs / Outputs



Workflow: Modeling Steps

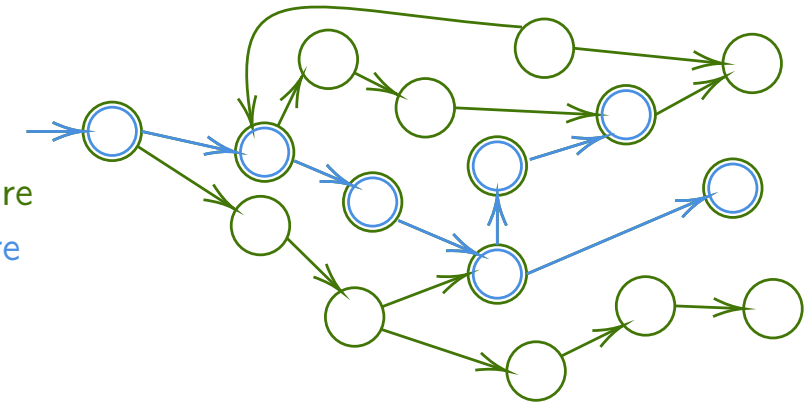
Hardware Modeling



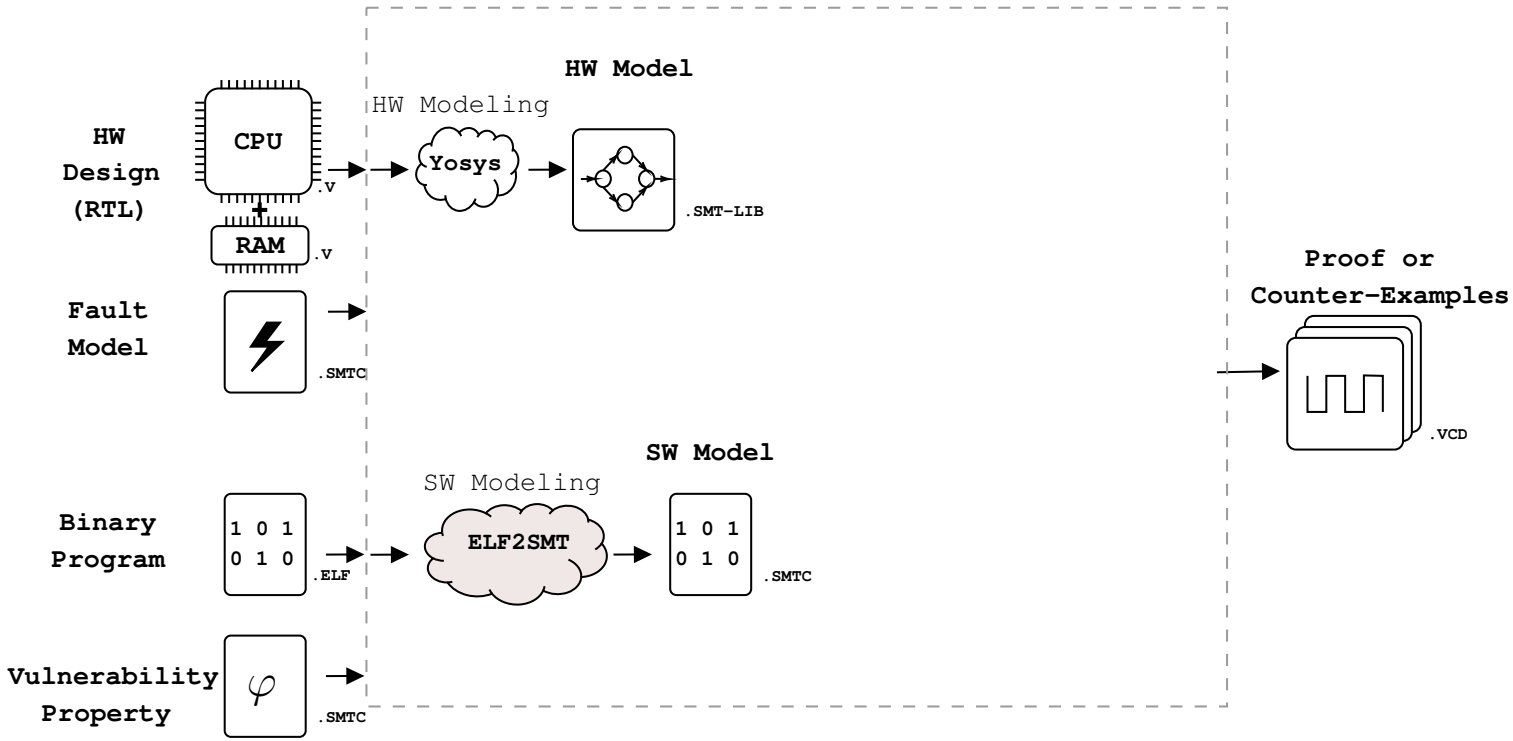


Workflow: Modeling Steps

Software Modeling

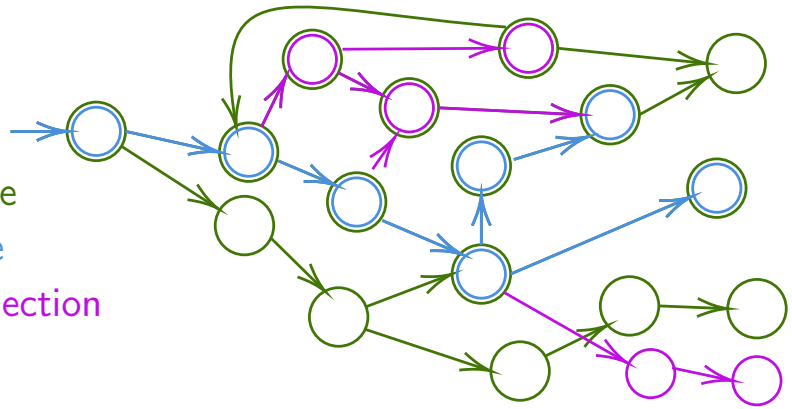


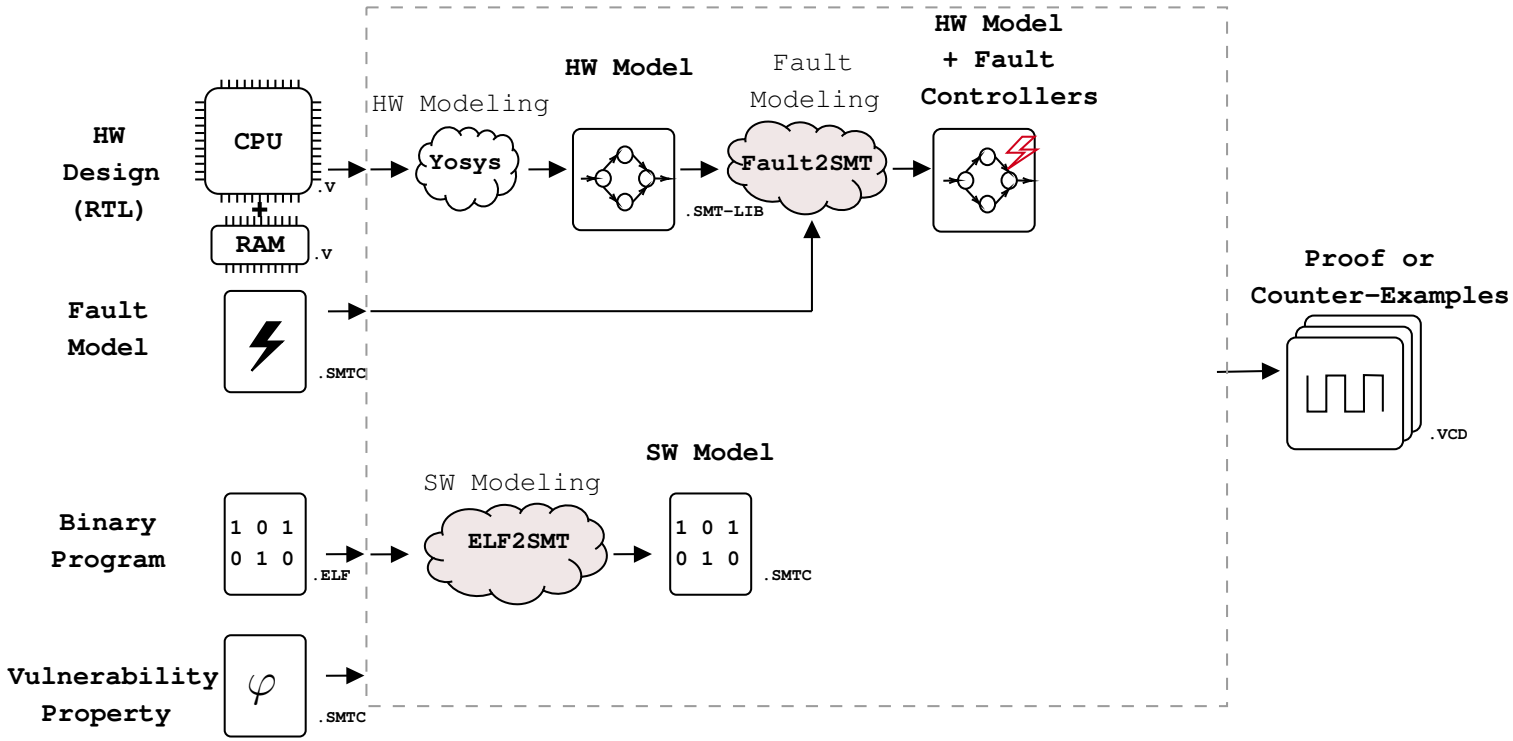
— Hardware
— Software



Workflow: Modeling Steps

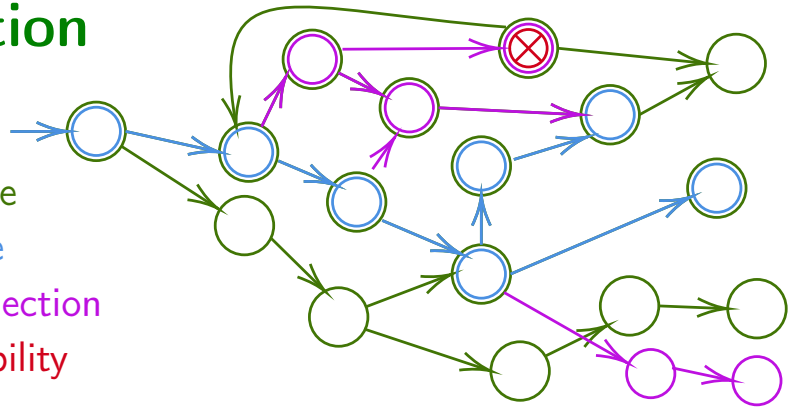
Fault Modeling



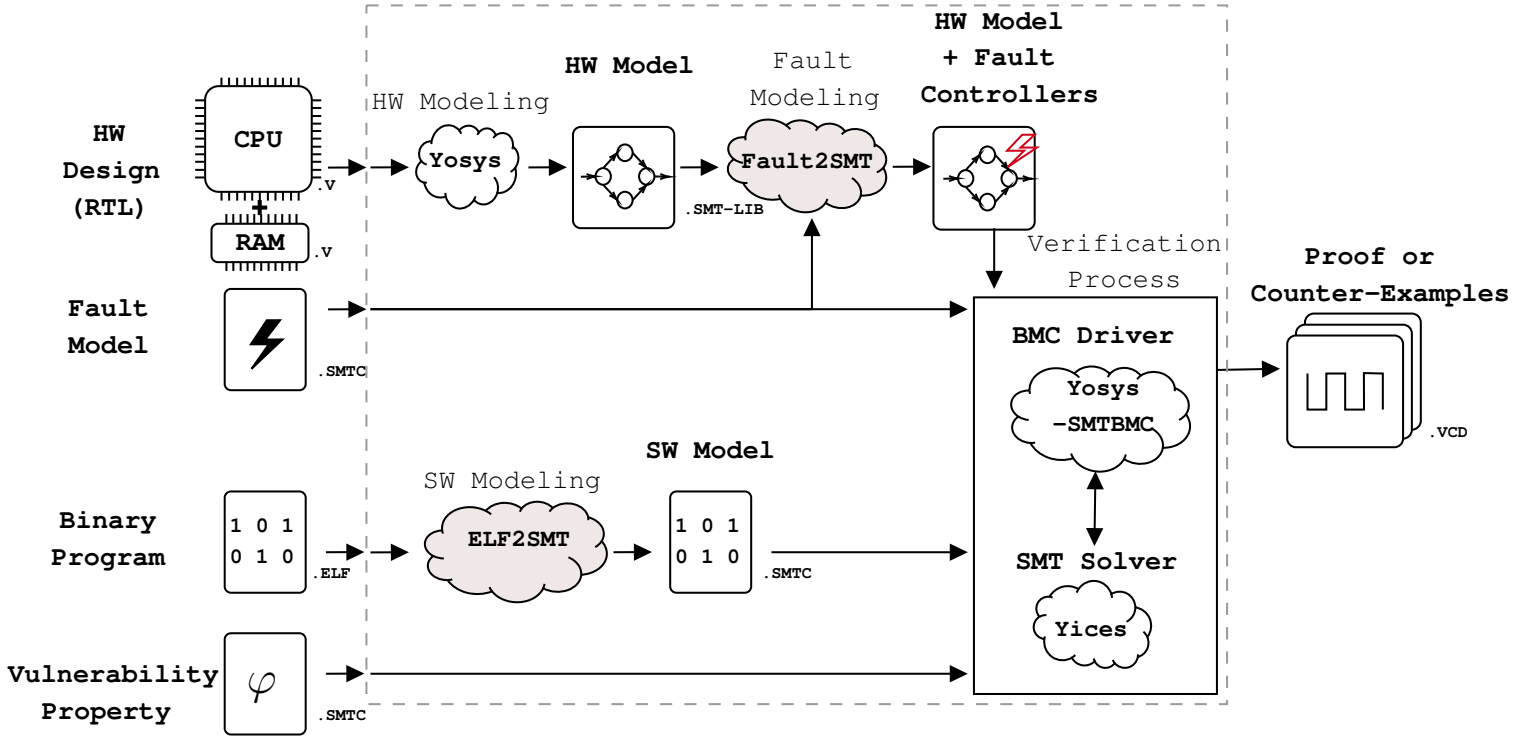


Workflow: Modeling Steps

Property Specification

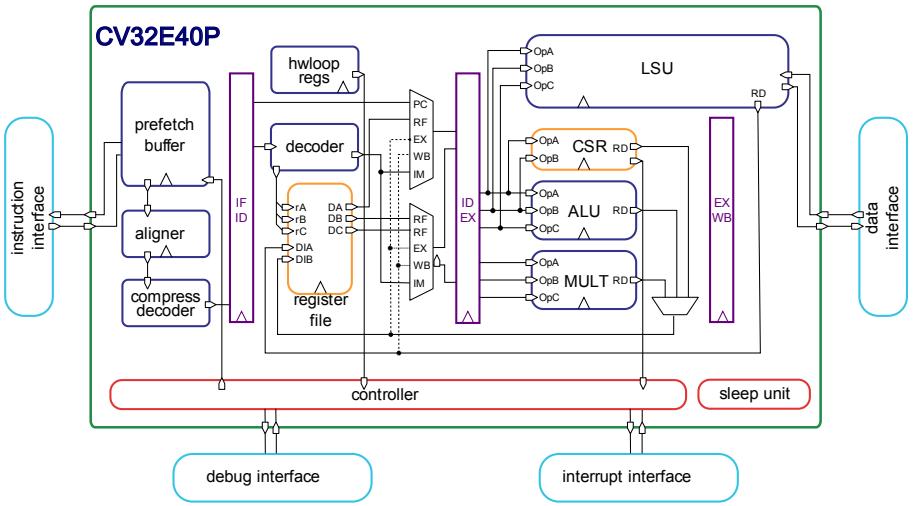


— Hardware
— Software
— Fault Injection
— Vulnerability



Hardware Part

CV32E40P



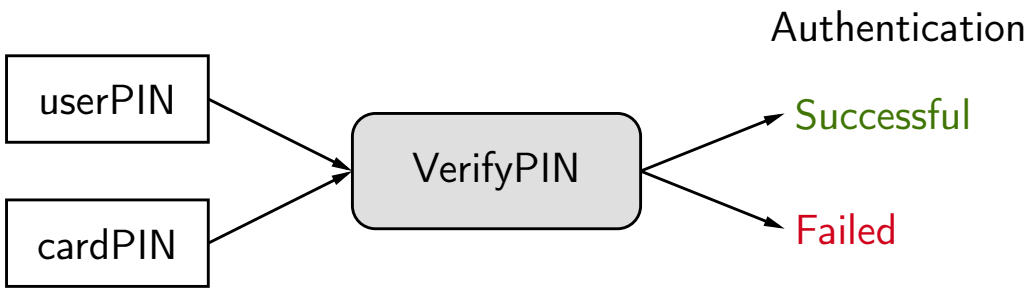
- Standard version [CV32E40P]
- Hardened version [Chamelot, 2022]
  - Control flow integrity
  - Code integrity
  - Execution integrity

Microarchitectural Fault Model

- Single fault injection
- During the whole program
- Everywhere in the circuit
- Symbolic fault effect

Software Part

VerifyPIN




```
Compare {
   for ( i = 0 ; i < 4 ; i ++) {
      if(userPIN[i] != cardPIN[i])
         return false;
   return true;
}

VerifyPIN {
   authentification = false;
   if (tries > 0) {
      if(Compare()) {
         tries = 3 ;
         authentification = true;
      } else {
         tries -- ;
      }
   }
}
```

- Standard version [Dureuil, 2016]
- Versions implementing SW countermeasures
  - Constant iteration number loop
  - Inline function calls
  - Duplication of critical tests

Security Property

- $\texttt{userPIN} \neq \texttt{cardPIN} \implies$ Authentication is not possible

Countermeasures have attack oracles to detect fault injections

Fault Effects Exploration Results

Find known attacks

- Exploiting the forwarding mechanism

Find new fault effects

- **Immediate one-time effect**, e.g., replay the Prefetch Buffer instructions
- **Immediate recurring effect**, e.g., incorrect order of the (replayed) instructions
- **Long-term effect**, e.g., corruption of the next branch target

➔ **Fault effects depend on the microarchitectural details and the execution context**

More details and other effects are described in the full paper

Robustness Analysis Results

Baseline CV32E40P + the most hardened VerifyPIN

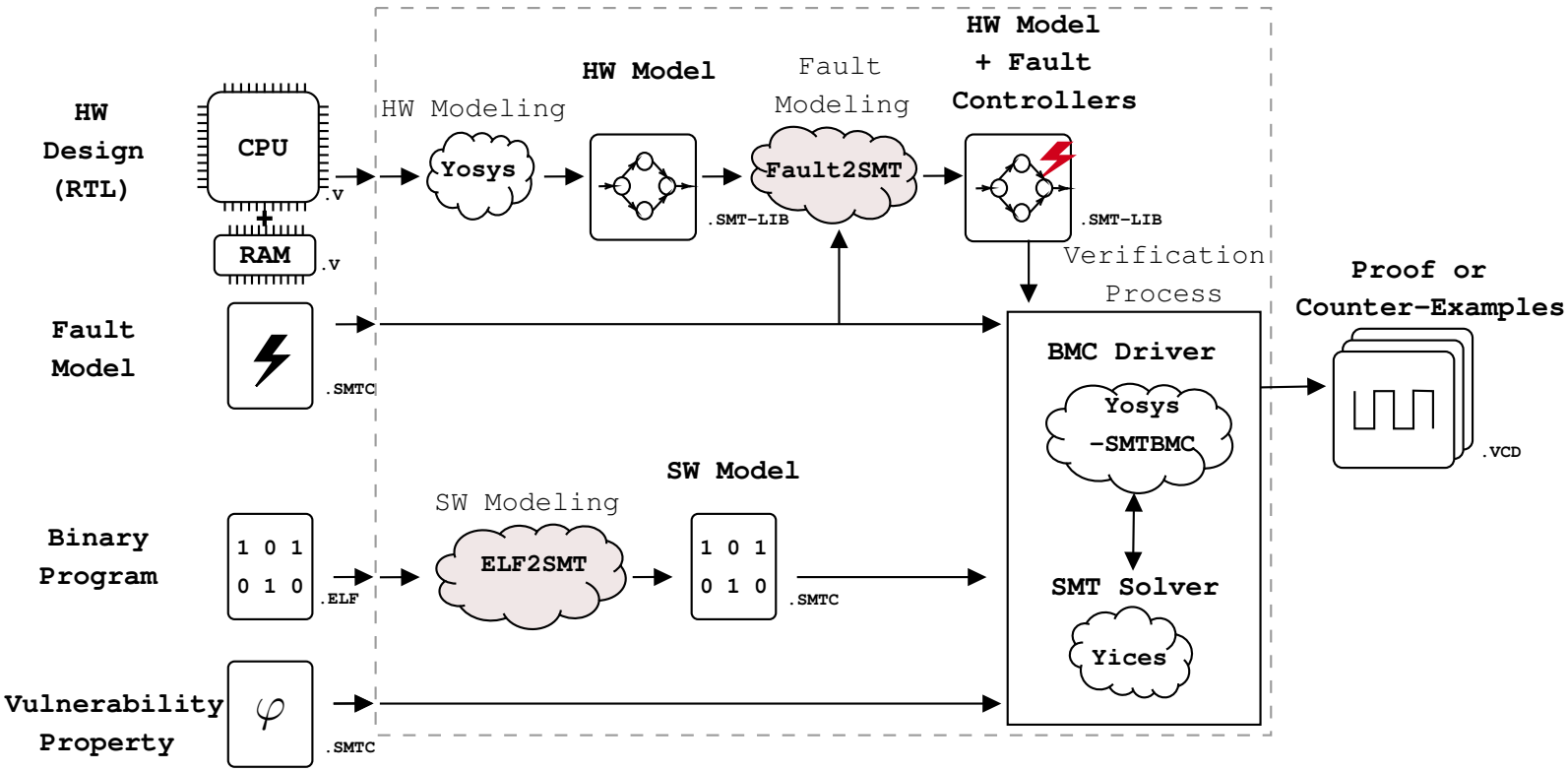
- No fault injection permits bypassing the secure authentication

Hardened CV32E40P + unprotected VerifyPIN

- No fault injection permits bypassing the secure authentication
- The hardware countermeasure is effective

More details and other effects are described in the full paper

Questions ?


HW Design (RTL)
CPU
.V
RAM
.V
HW Modeling
Yosys
HW Model
.SMT-LIB
Fault Modeling
Fault2SMT
HW Model + Fault Controllers
.SMT-LIB
Verification Process
Proof or Counter-Examples
.VCD
Fault Model
.SMTC
BMC Driver
Yosys -SMTBMC
SMT Solver
Yices
Binary Program
1 0 1
0 1 0
.ELF
SW Modeling
ELF2SMT
SW Model
1 0 1
0 1 0
.SMTC
Vulnerability Property
φ
.SMTC

References I

[Riviere, 2015] L. Riviere et al. (2015)
High precision fault injections on the instruction cache of ARMv7-M architectures
IEEE International Symposium on Hardware Oriented Security and Trust (HOST)

[Moro, 2013] N. Moro et al. (2013)
Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller
Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)

[Menu, 2020] A. Menu et al. (2020)
Experimental analysis of the electromagnetic instruction skip fault model
15th Design & Technology of Integrated Systems in Nanoscale Era (DTIS)

[Trouchkine, 2020] T. Trouchkine et al. (2020)
Fault Injection Characterization on Modern CPUs
Information Security Theory and Practice

[Laurent, 2018] J. Laurent et al. (2018)
On the importance of analysing microarchitecture for accurate software fault models
21st Euromicro Conference on Digital System Design (DSD)

References II

[Proy, 2019] J. Proy et al. (2019)
A first ISA-level characterization of EM pulse effects on superscalar microarchitectures: a secure software perspective
Proceedings of the 14th International Conference on Availability, Reliability and Security

[CV32E40P] OpenHW group
CORE-V CV32E40P User Manual
`https://cv32e40p.readthedocs.io/en/latest/intro/`

[Chamelot, 2022] T. Chamelot et al. (2022)
SCI-FI: control signal, code, and control flow integrity against fault injection attacks
Design, Automation & Test in Europe Conference & Exhibition (DATE)

[Dureuil, 2016] L. Dureuil et al. (2016)
FISSC: A fault injection and simulation secure collection
International Conference on Computer Safety, Reliability, and Security