Complementing DV by proving formal ISA compliance with Questa Processor

Leveraging formal methods for improving processor verification | DVClub Cambridge

Sven Beyer, Siemens EDA

SIEMENS

Table of contents

Introduction

SIEMENS

Challenges of Processor Verification

ISA, architecture and specification verification deliver cores with integrity

Complex architecture

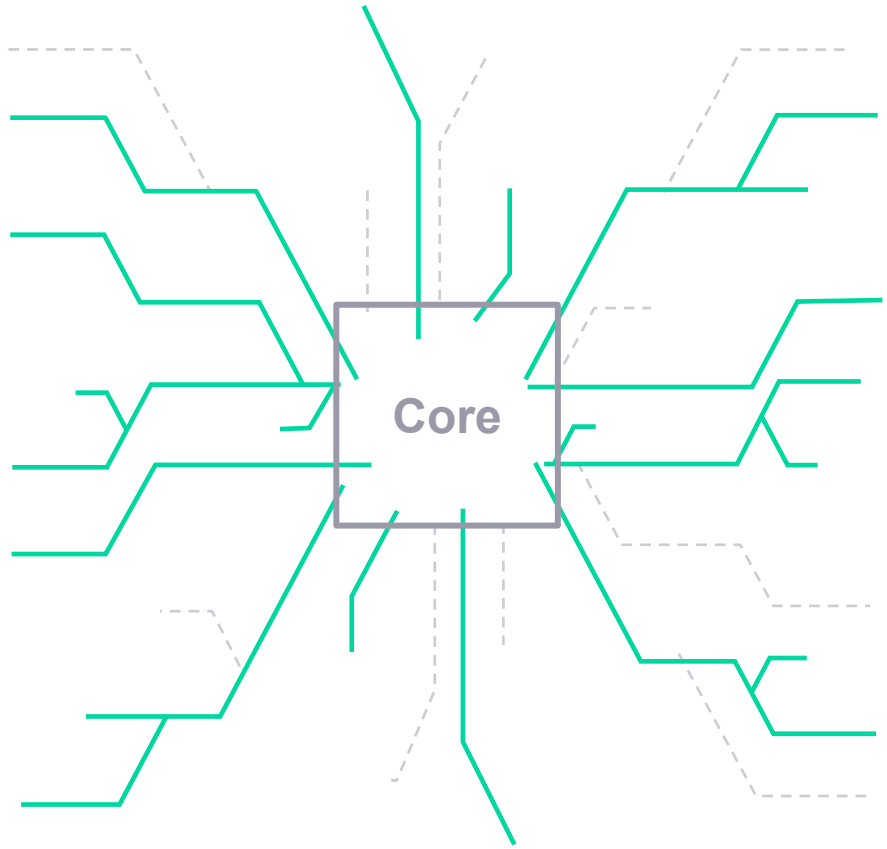
- (Custom) extensions
- Exceptions/ Interrupts

Very complex µArchitecture

- Continuous PPA optimizations
- Pipelined implementation

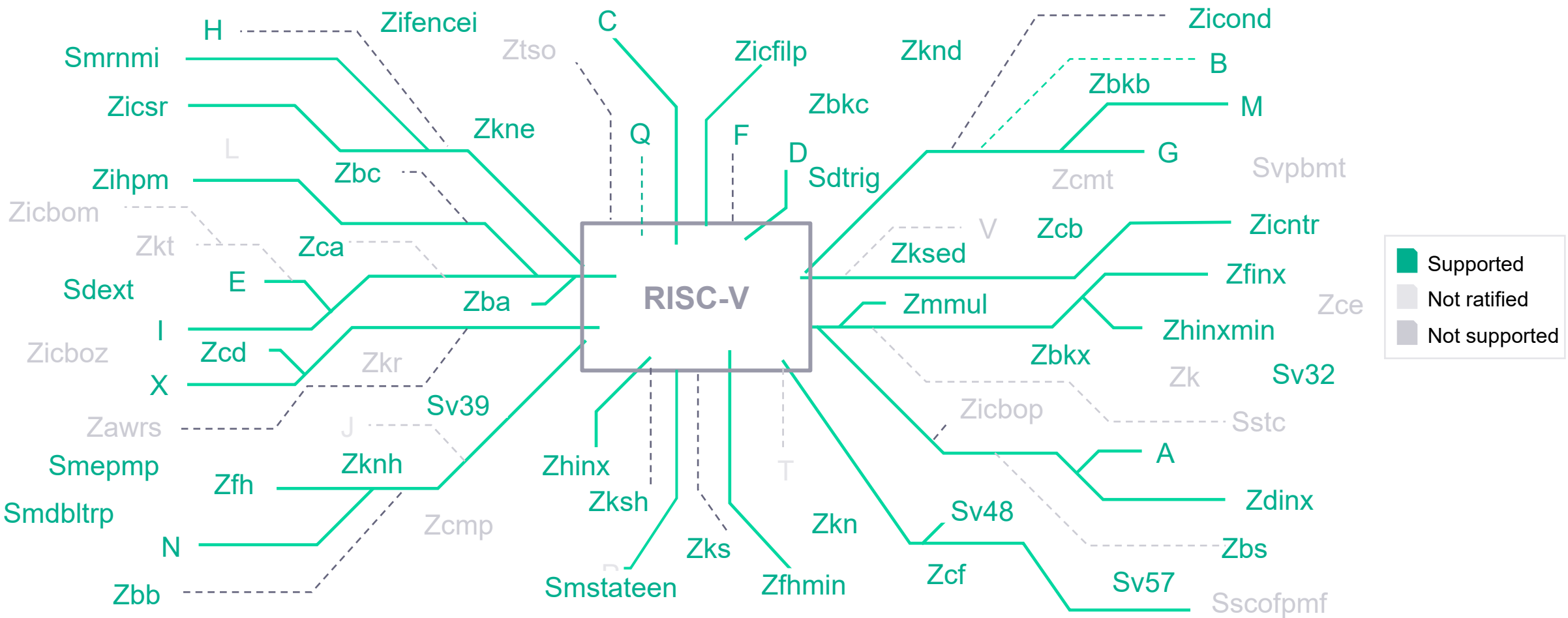
Verification – high effort task

- Writing functional coverage model
- Simulation cannot hit all pipeline corner-cases
- Slow debug process
- Functional and structural coverage closure
- Customization introduces bugs in existing functionality



SIEMENS

Supported Extensions

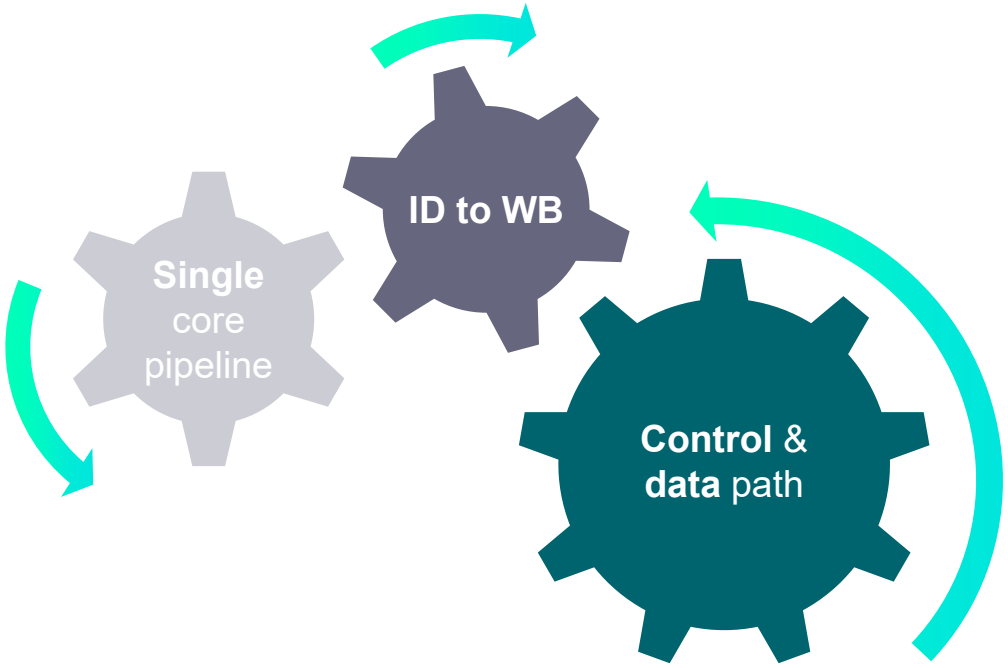


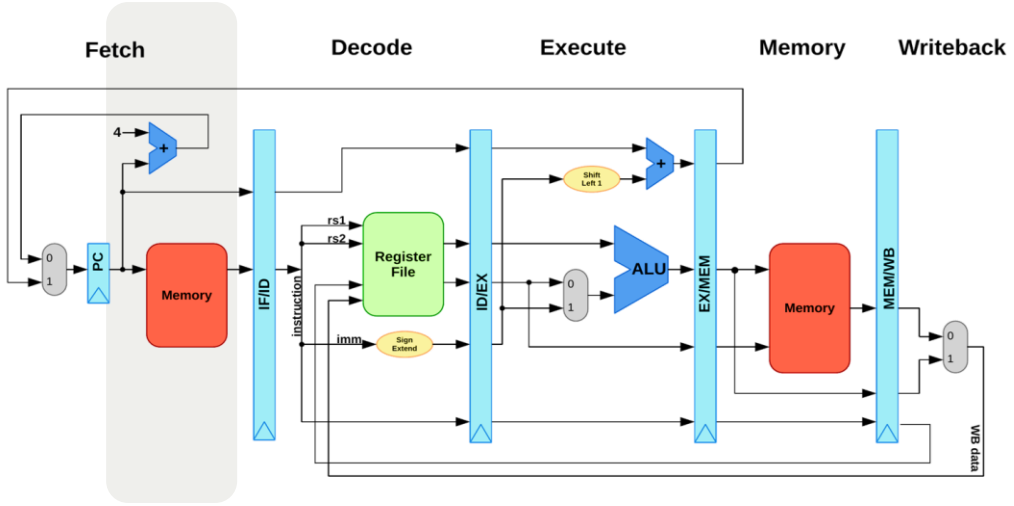
SIEMENS

Questa Processor

Enables highest confidence in the trust and security of processor implementations

- Ideal for in-order execution core implementations with single & multi-issue capabilities
- Detect any inconsistencies between RTL core implementation and the ISA
- Address the needs of core providers and core integrators





SIEMENS

Siemens EDA's Industry Proven Solutions

Codasip

How the Right **Mindset** Increases **Quality** in RISC-V Verification

2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE MUNICH, GERMANY DECEMBER 6 - 7, 2022

ECI EDAPTIVE COMPUTING, INC.

Complete Formal Verification of **RISC-V Processor IPs** for **Trojan-Free Trusted ICs**

GOMAC Tech

BOSCH Invented for life

Complete Formal Verification of a Family of **Automotive DSPs**

2016 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

RENESAS

Formal Verification Applied to the Renesas **MCU Design Platform**

51 DESIGN AUTOMATION CONFERENCE

Infineon technologies

Complete Formal Verification of **TriCore2** and Other Processors

The content of this article was presented at DVCon 2007 and is posted with DVCon's permission.

~20 years

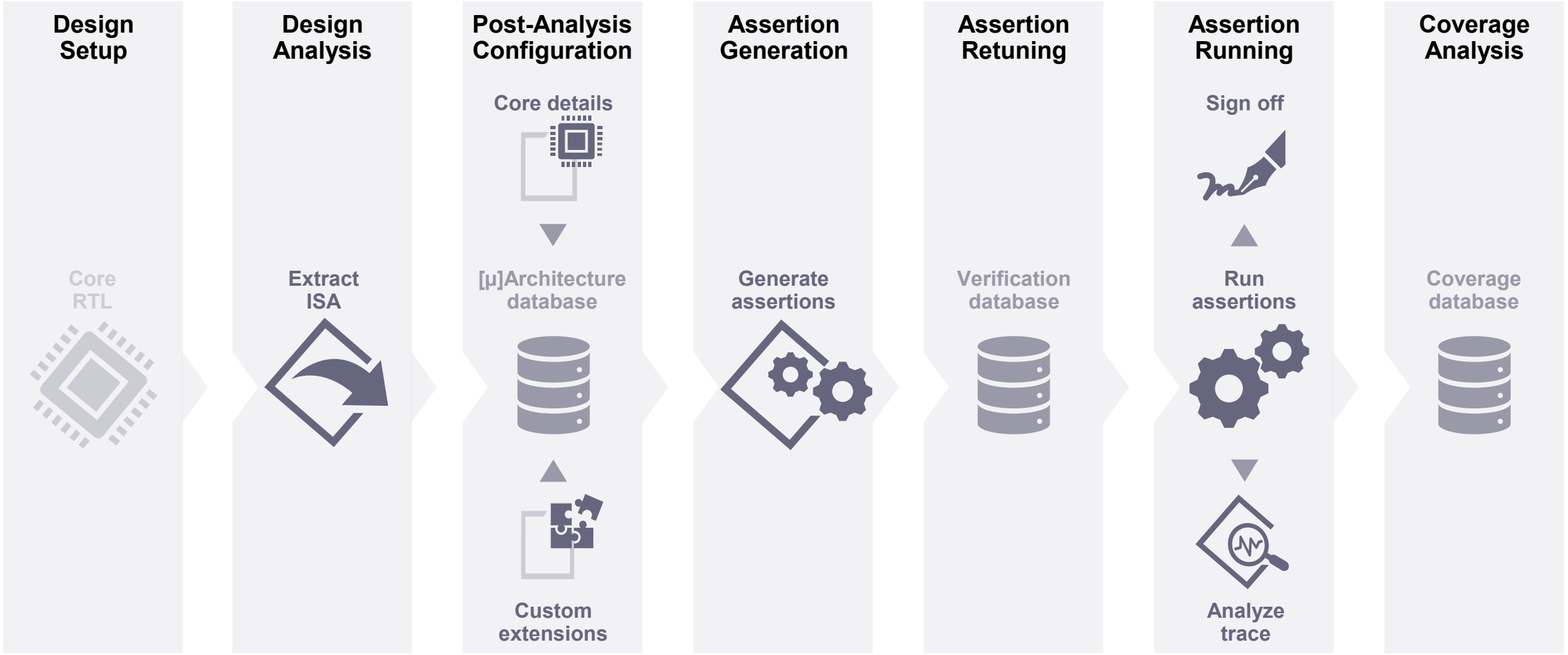
of cutting-edge formal processor verification solutions

SIEMENS

Flow

SIEMENS

App Flow



SIEMENS

App Assertions

Assertion Running

Sign off

Run assertions

Analyze trace

Name	Proof Status	Witness Status	Validity	App
!	! <any status>	! <any status>	! <any validity>	
Properties				
RV_chk.ops.BUBBLE_a	open	open	up_to_date	Processor
RV_chk.ops.RESET_a	open	open	up_to_date	Processor
RV_chk.RV32C.ARITH_a	open	open	up_to_date	Processor
RV_chk.RV32C.BRANCH_a	open	open	up_to_date	Processor
RV_chk.RV32C.JUMP_a	open	open	up_to_date	Processor
RV_chk.RV32C.MEM_a	open	open	up_to_date	Processor
RV_chk.RV32C.MEM_MultiAccess_a	open	open	up_to_date	Processor
RV_chk.RV32I.ARITH_a	open	open	up_to_date	Processor
RV_chk.RV32I.BRANCH_a	open	open	up_to_date	Processor
RV_chk.RV32I.FENCE_a	open	open	up_to_date	Processor
RV_chk.RV32I.JUMP_a	open	open	up_to_date	Processor
RV_chk.RV32I.MEM_a	open	open	up_to_date	Processor
RV_chk.RV32I.MEM_MultiAccess_a	open	open	up_to_date	Processor
RV_chk.RV32I.RETURN_a	open	open	up_to_date	Processor
RV_chk.RV32I.WFI_a	open	open	up_to_date	Processor
RV_chk.RV32X.WFE_a	open	open	up_to_date	Processor
RV_chk.RV32Zcb.ARITH_a	open	open	up_to_date	Processor
RV_chk.RV32Zcb.MEM_a	open	open	up_to_date	Processor
RV_chk.RV32Zcb.MEM_MultiAccess_a	open	open	up_to_date	Processor
RV_chk.RV32Zicsr.CSRx_a	open	open	up_to_date	Processor
RV_chk.RV32Zifencei.FENCE_I_a	open	open	up_to_date	Processor
RV_chk.xcpt.EBREAK_a	open	open	up_to_date	Processor
RV_chk.xcpt.ECALL_a	open	open	up_to_date	Processor
RV_chk.xcpt.INTR_Handle_a	open	open	up_to_date	Processor
RV_chk.xcpt.XCPT_IF_ID_a	open	open	up_to_date	Processor
RV_chk.xcpt.XCPT_WB_a	open	open	up_to_date	Processor

RV32IMC_Zicsr_Zifencei

24

Assertions

# of properties, classified based on implemented extensions, is kept to the minimum

SIEMENS

Initial Value Abstraction

Performance Enhancement

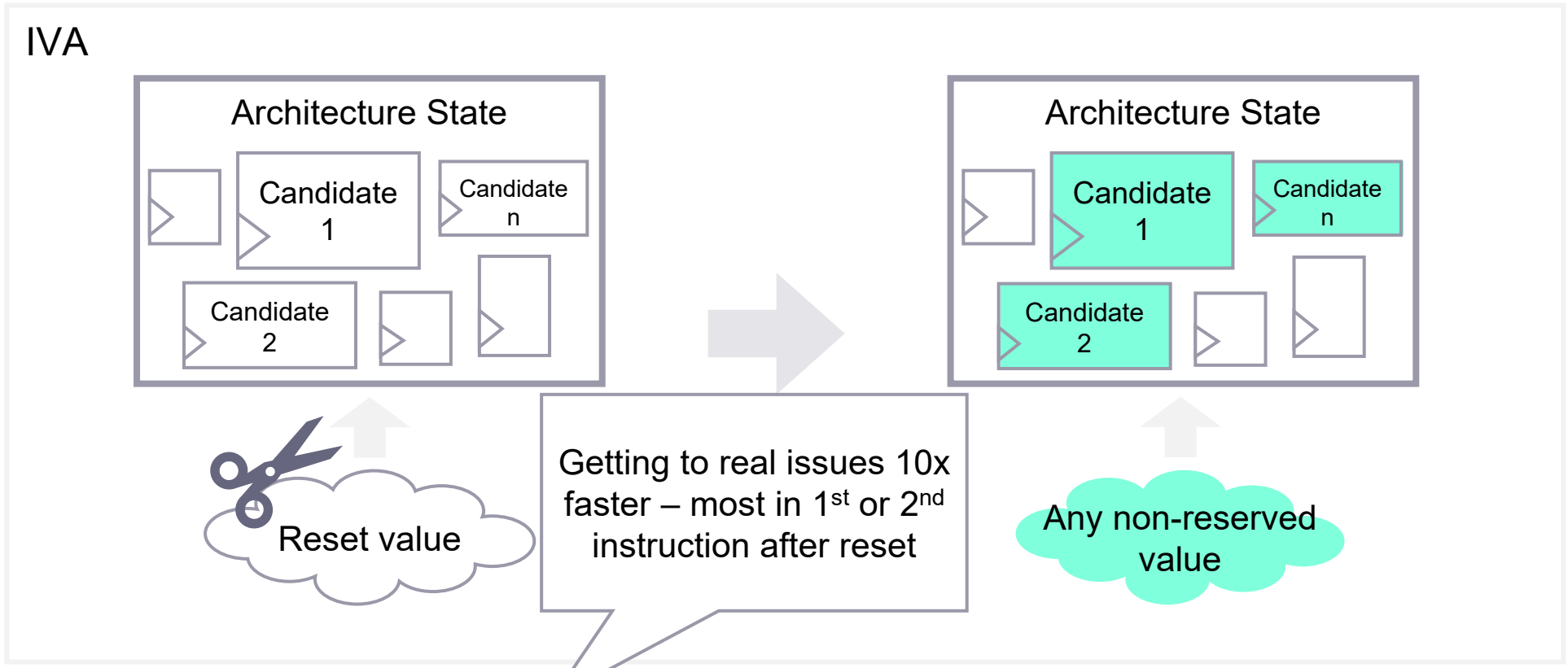
Abstract initial value

IVA

Read assertions

Other techniques

Abstract initial value of architecture state registers by allowing arbitrary reset values



Automated initial value abstraction speeds up hitting corner cases

SIEMENS

Trace Analysis

Assertion Running

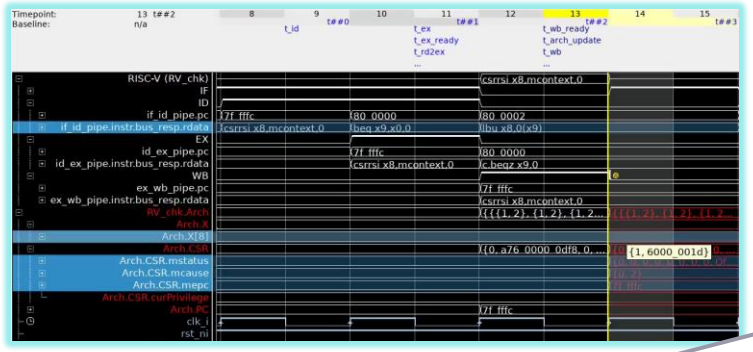
Analyze trace

Run assertions

Sign off

Perform detailed trace analysis of run assertions

Trace Analysis



```
mv> processor_integrity::analyze_trace RV_chk.RV32Zicsr.CSRx_a
-I- Counterexample for property 'RV_chk.RV32Zicsr.CSRx_a' generated.
-I- TraceAnalysis - Current architecture frozen at 't_arch+0'
-I- TraceAnalysis - ISA execution called at 't_model+0'
-I- TraceAnalysis - Execution bound at 't_model+0'
-I- TraceAnalysis - Architecture update checked at 't_arch_update+0'
-I- TraceAnalysis - Analysis of counterexample 'RV_chk.RV32Zicsr.CSRx_a' updating register file at time 13 (t##2, t_wb_ready):
active CSR IVAs at time 3 (t##-3)
     mstatus.MIE: 0x1
         mie.MEI: 0x1
    dcsr.ebreakm: 0x1
  dcsr.stopcount: 0x0
current instruction:
        csrrsi x8,mcontext,0            (PC 0x007ffffc) decoding at time 9 (t##0), updating registers at 13 (t##2, t_wb_ready) (0x7a806473)
        executed in Machine mode
interrupts during instruction execution at time 13 (t##2, t_wb_ready):
          CUSTOM: 0xfff7
             MSI: 0x1
Prioritized ISA interrupt: MEI
decoding fields (freeze variable exe.dec):
            instr: CSRRSI
    ISA decoding: csr uimm[4:0] 110 rd 1110011
           RD[1]: 8
             imm: 0
             csr: 0x7a8
           width: 0x4
expected CSR access (freeze variable exe.csr):
             cmd: cc_r
             csr: csr_mcontext
         illegal: 0x0
           wdata: 0x00000000
            addr: 0x7a8
       _inhibit: 0x0000000d
ister file results (freeze variable result):
       ISA X[ 8]: 0x00000000 (0)
       DUT X[ 8]: 0x6000001d (1610612765) (unexpected)
ISA execution for csrrsi x8,mcontext,0 with current register values:
        X(8) = CSR_access(if 0x00000000!=0 then cc_rs else cc_r,0x7a8,0x00000000)
DUT CSR updates (freeze variable cur_Arch and signal RV_chk.ARch at time 15 (t##3)):
    mstatus.MPIE: 0x0 -> 0x1
     mstatus.MIE: 0x1 -> 0x0
      mip.CUSTOM: 0xfff7 -> 0x0
         mip.MSI: 0x1 -> 0x0
          mcause: xcpt_Fetch_Addr_Align -> xcpt_Illegal_Instr
            mepc: 0x0 -> 0x7ffffc
-I- TraceAnalysis - pipe_instr fails in property debugger!
-I- TraceAnalysis - pipe_rf_check fails in property debugger!
-I- TraceAnalysis - rfs_equal fails in property debugger!
-I- TraceAnalysis - check_regs_equal==_.X[8].data (DUT value 0x6000001d at t+3, expected value 0x0) fails in property debugger!
```

Understanding current instruction and its effects, including exceptions and links to Sail model and ISA document

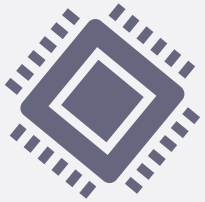
Automated trace analysis saves tremendous debugging time and pinpoints bugs

SIEMENS

App GUI

Design Setup

Core RTL



Questa Processor SIEMENS

Status

Initial

Merge

Sync

Clear

Commands

Extract from design

Generate assertions

ISA

XLEN: 32 **Number of Counters:** 0 **Number of PMPs:** 0

Extensions: A B C D E F I M N S U X

Z: Zifencei Zicsr Zfinx Zdinx Zba Zbb Zbc Zbs ...

S: Smepmp Smstateen

Advanced

Custom Extensions:

Instructions | Registers | Register Files | CSR Map | CSR Attributes

µ-Architecture

DUT Module: **DUT Instance:**

Fetch Interface: **Data Memory Interface:**

Resp. Valid:	**Req. Valid:**	**Req. Valid:**	**Resp. Valid:**
Resp. Ready:	**Req. Ready:**	**Req. Ready:**	**Resp. Ready:**
Resp. Data:	**Req. PC:**	**Req. Address:**	**Read Data:**
Resp. PC:		**Write Data:**	**Req. Cancel:**

Pipeline | Mappings | Parameters | Invariants

Processor apps

Design ISA

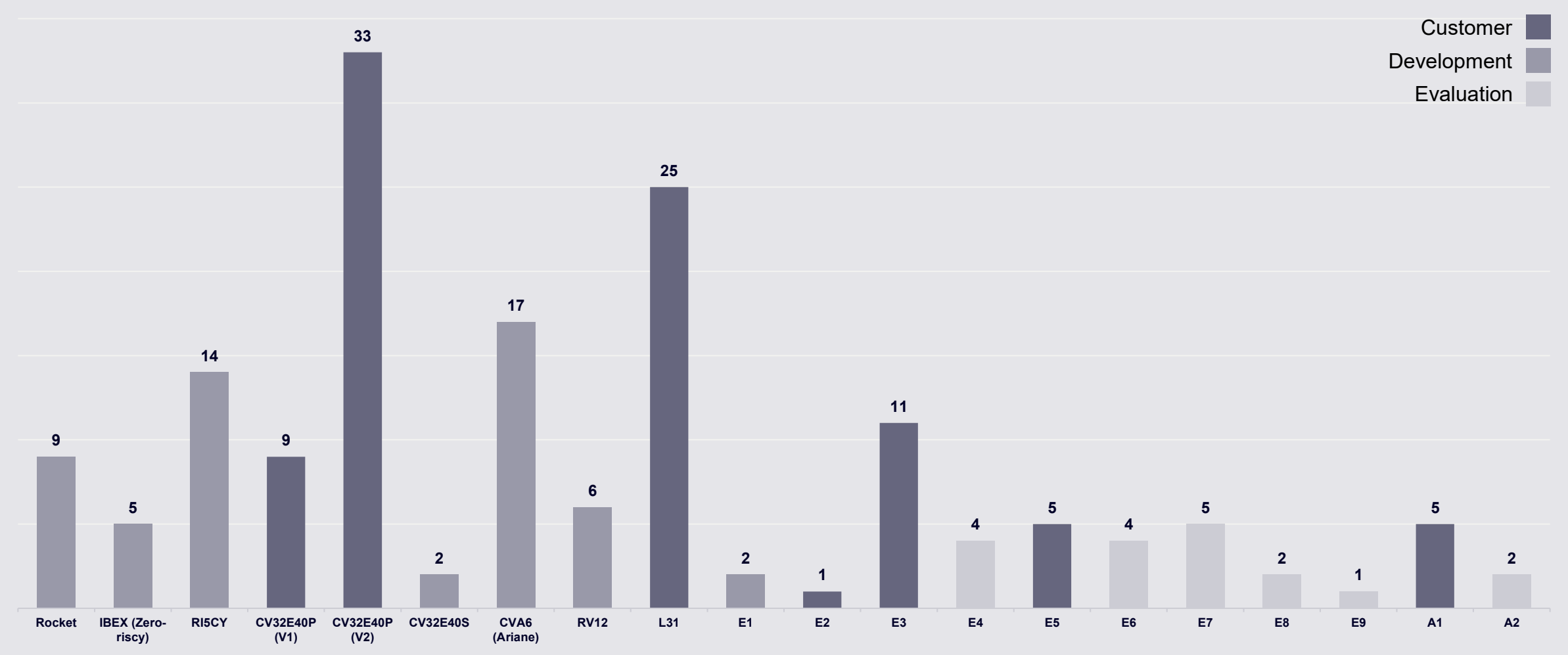
Design µ-Architecture

SIEMENS

Application Experience

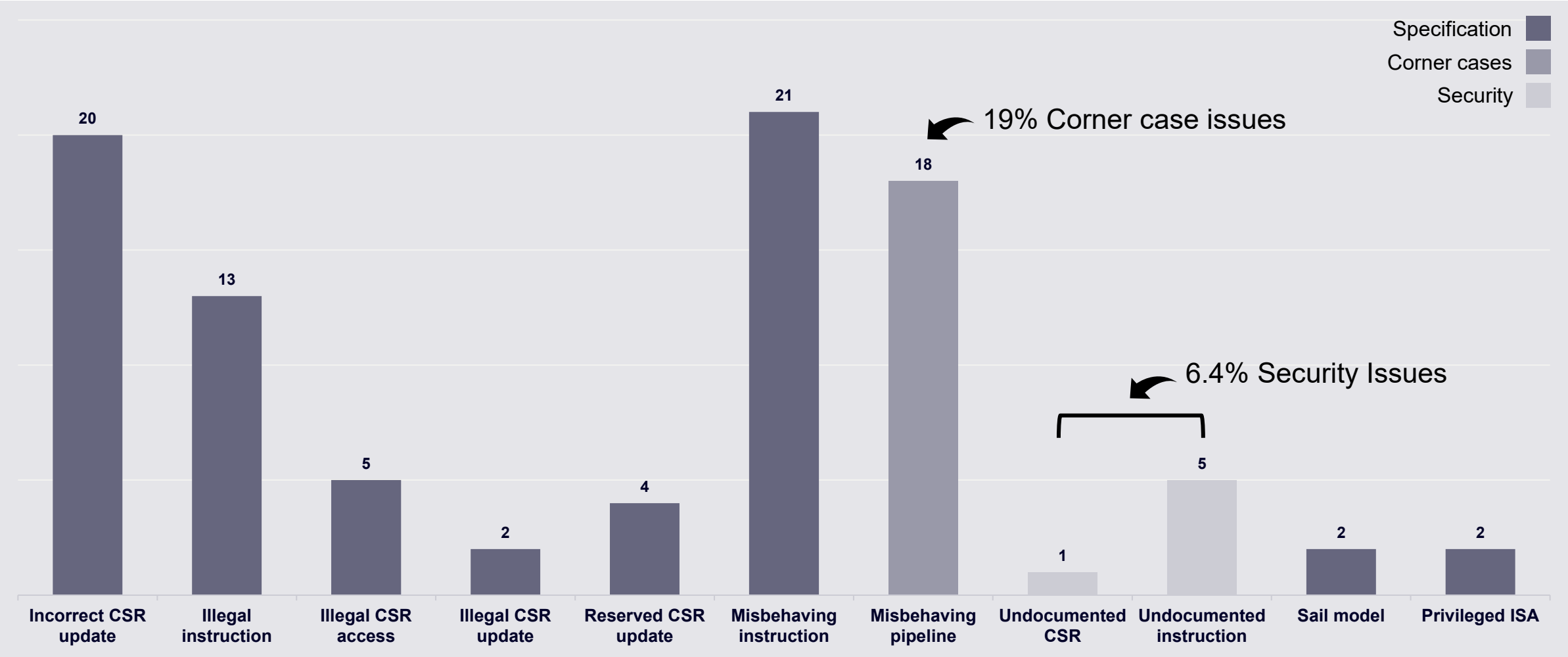
SIEMENS

#Bugs Found



SIEMENS

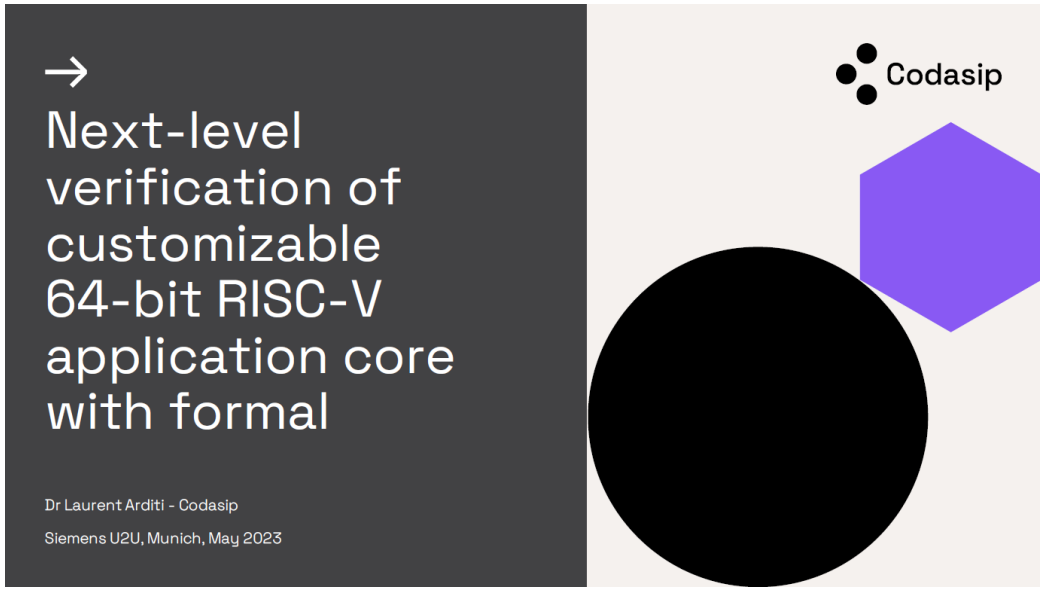
Bug Types | Open-Source Designs

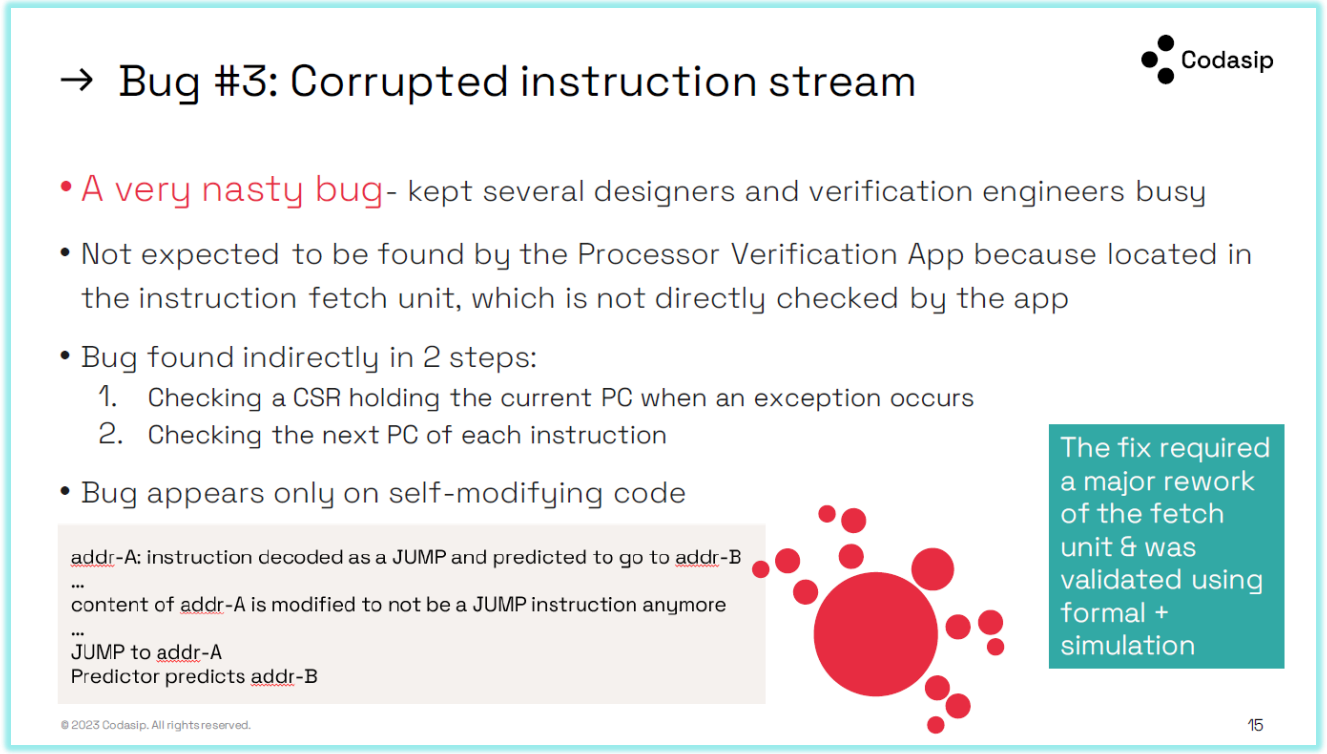


SIEMENS

Processor Verification Success: Codasip

RISC-V community recognizes the need for stronger verification





> "…In all cases it caught bugs even after all other verification methods stopped finding new ones."
>
> *– Dr Laurent Arditi, Director of Verification, Codasip*

SIEMENS

Application Example | CV32E40Pv2

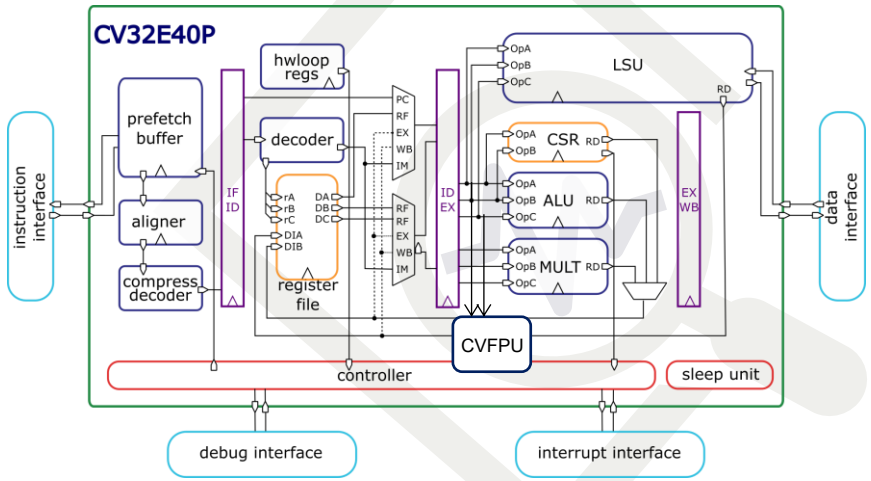
Design specification

- **+ Floating point & X custom instruction set extensions**
  - Post-incrementing load & store
  - ALU & Multiply accumulate
  - Single instruction multiple data (SIMD)
  - Hardware loops (zero-overhead loops)

Selection of issues reported

- #722 Wrong instruction fetch caused by multicycle F instructions
- #723 Misaligned memory instructions set wrong memory access
- #725 No illegal instruction exception raised for non-Zfinx instructions
- #729 FMUL.S sets underflow flag of fflags wrongly
- #731 Custom Xpulp memory instructions set extra memory access
- #742 Simultaneous register file update by custom instructions

RV32IM**F**C_Zicsr_Zifencei**_Zfinx_Xpulp_Xcluster**



32-bit CV32E40Pv2

33 bugs

+2 Standard extensions **+8** Custom CSRs

+2 Custom extensions **+320** Custom instructions

SIEMENS

Bug Case Study | CV32E40Pv2

Custom Xpulp memory instructions update register file wrongly | Simultaneous port writes #742

Closed **salaheddinhetalani** opened this issue on Nov 28, 2022 · 1 comment

Execution

```
X(rs1) += X(rs2)
X(rd)   = EXTS(mem16(X(rs1)))
```

X4: Updated **first** via ALU port

X4: Updated **later** via LSU port

CV32E40Pv2

31 sec

Disassembly CV.LH.RI {rd},{rs2}({rs1}!)

Time

ID	FDIV.S x16, x23, x0, DYN	CV.ROR x1, x0, x4	CV.LH.RI x4, x26(x4!)
EX	FDIV.S x16, x23, x0, DYN	CV.ROR x1, x0, x4	CV.LH.RI x4, x26(x4!)
MEM	FDIV.S x16, x23, x0, DYN	CV.ROR x1, x0, x4	CV.LH.RI x4, x26(x4!)
WB	CV.ROR x1, x0, x4	FDIV.S x16, x23, x0, DYN	CV.LH.RI x4, x26(x4!)
			CV.LH.RI x4, x26(x4!)

Verified instruction

X4 is updated **simultaneously** via the two ports

CV.ROR update

Multicycle FDIV.S update

Delayed X(rs1) update via ALU port **wrongly takes priority**

SIEMENS

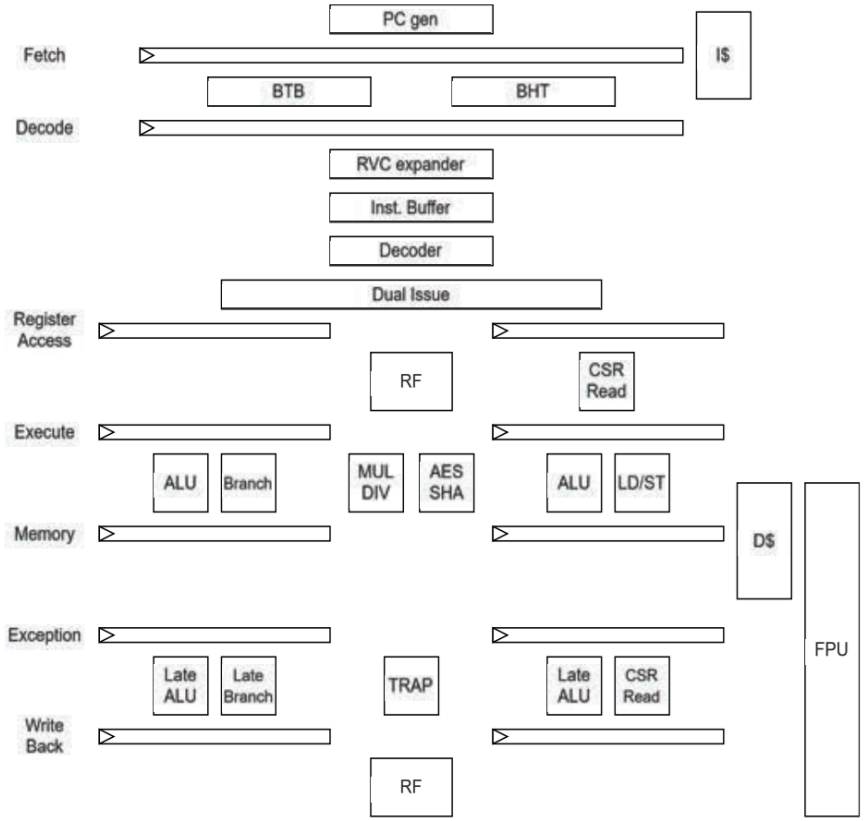
Application Example | Noel-V

Design specification

- 7-stage single/dual-issue in-order pipeline
- Highly configurable (32/64 bit, MC_lite, MC, GP_lite, GP, HP)
- Many privileged extensions
  - H, Sscofpmf, Sstc, Smstateen, Smrnmi, Smdbltrp, Sstimetmp, Smcdeleg, Smepmp, Zicfilp, Zicfiss
- https://www.gaisler.com/products/noel-v

Selection of issues reported on GRLIB 2025.2

- mnret instruction does not clear sstatus.SDT
- mret instruction on elp=1 executes without landing pad exception
- Missing hazard handling between mcounteren and scountinhibit
- Unexpected influence of mstatus.TVM in VS-mode



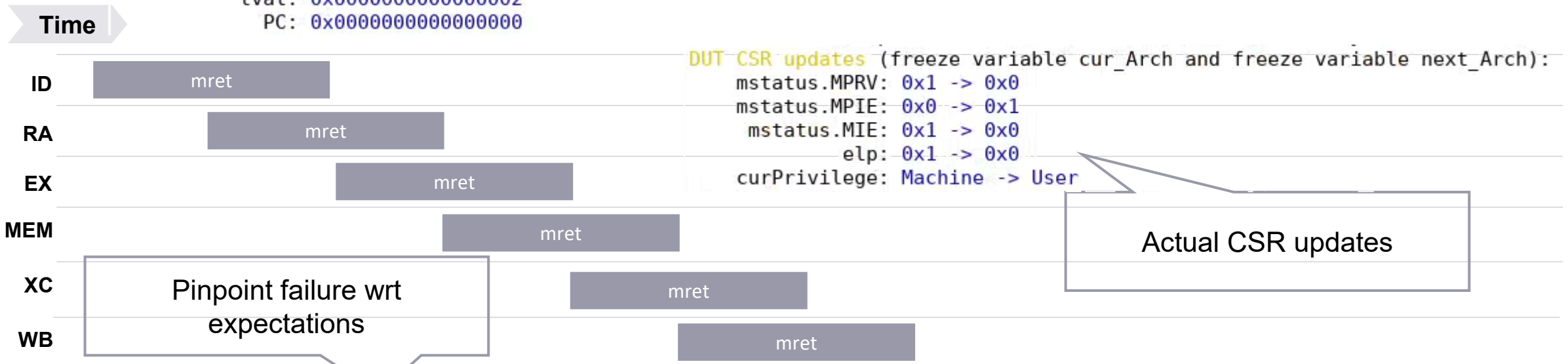
SIEMENS

Bug Case Study | Noel-V Zicfilp corner case (non-critical)

mret instruction on elp=1 executes without landing pad exception

```
expected exception (freeze variable xcpt_exe.xcpt):
  classification: cls_Software_Check_Landing_Pad_Exception
            mode: M
           cause: xcpt_Software_Check
      unexpected: 0x0
  critical_error: 0x0
             epc: 0x00000000c0000000
            tval: 0x0000000000000002
              PC: 0x0000000000000000
```

Expected exception




```
DUT CSR updates (freeze variable cur_Arch and freeze variable next_Arch):
    mstatus.MPRV: 0x1 -> 0x0
    mstatus.MPIE: 0x0 -> 0x1
     mstatus.MIE: 0x1 -> 0x0
             elp: 0x1 -> 0x0
    curPrivilege: Machine -> User
```

Actual CSR updates

Pinpoint failure wrt expectations

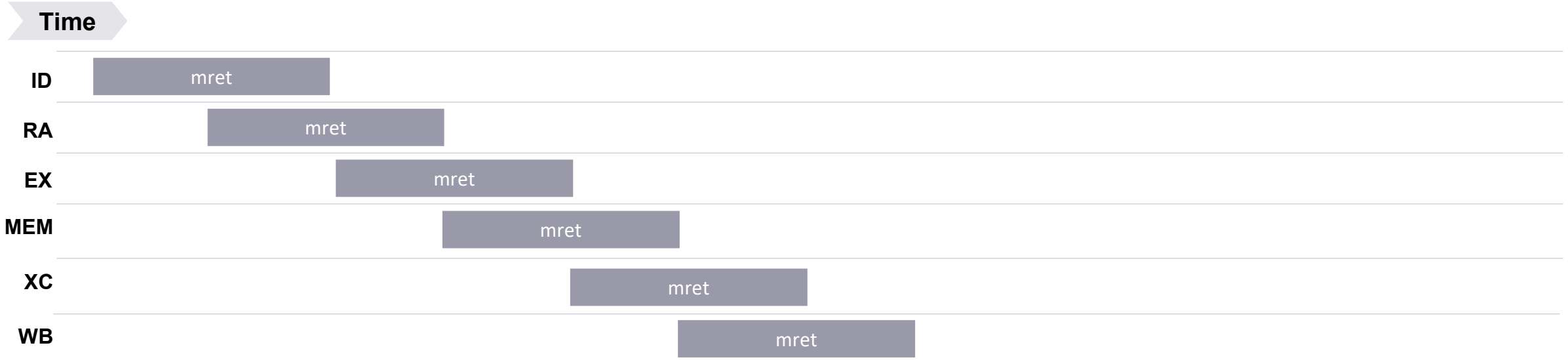
```
-I- TraceAnalysis - Failures in check (property debugger):
-I- TraceAnalysis - `--pipe_instr fails!
-I- TraceAnalysis -     `--==_.CSR.mcause.Cause (DUT value 0x0 at t+6, expected value 0x12) fails!
```


SIEMENS

Bug Case Study | Noel-V Zicfilp corner case (non-critical)

Assessment:

Executing an xRET would cause an illegal instruction exception if done from a lower priv mode, and in the correct mode it would return to the previous (normally lower) privilege mode.
It seems unlikely that making use of this bug can do anything useful for an attacker.



SIEMENS

Bug Case Study | Noel-V Smrnmi+Smdbltrp (non-critical)

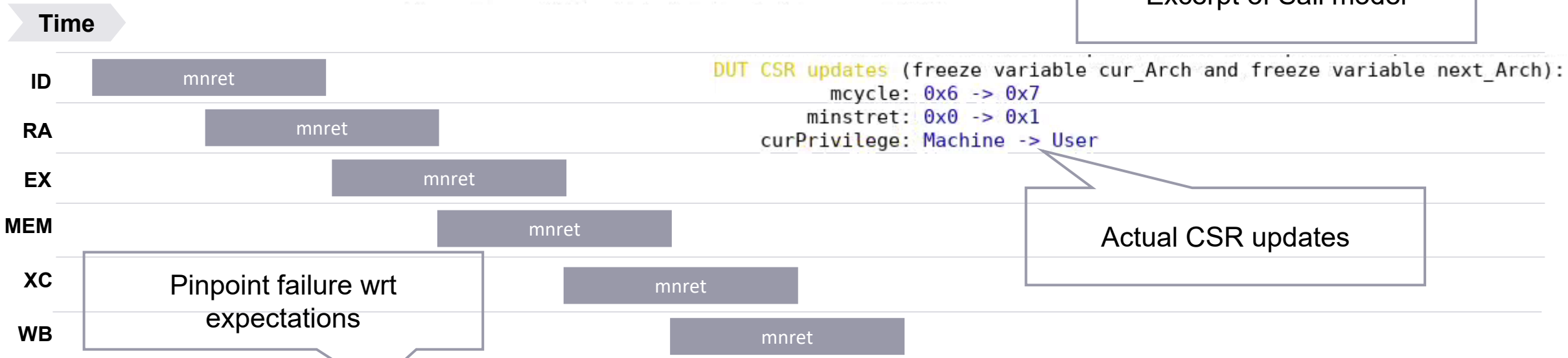
mnret instruction does not clear sstatus.SDT

Two relevant specification sections

```
ISA description for mnret:
        Privileged specification, section 3.1.6.2: The MNRET instruction, provided by the Smrnmi extension, sets the MDT bit to 0 if the new privilege mode is not M. If it is U, VS, or
VU, then sstatus.SDT is also set to 0. Additionally, if it is VU, then vsstatus.SDT is also set to 0.
        Privileged specification, section 8.4: MNRET is an M-mode-only instruction that uses the values in mnepc and mnstatus to return to the program counter, privilege mode, and
virtualization mode of the interrupted context. This instruction also sets mnstatus.NMIE. If MNRET changes the privilege mode to a mode less privileged than M, it also sets
mstatus.MPRV to 0. If the Zicfilp extension is implemented, then if the new privileged mode is y, MNRET sets ELP to the logical AND of yLPE (see Section 22.1.1) and mnstatus.MNPELP.
```

```
if mnstatus.MNPP==User | misa.H & mnstatus.MNPV then {
        mstatus.SDT=0b0;
```

Excerpt of Sail model




```
DUT CSR updates (freeze variable cur_Arch and freeze variable next_Arch):
        mcycle: 0x6 -> 0x7
      minstret: 0x0 -> 0x1
  curPrivilege: Machine -> User
```

Actual CSR updates

Pinpoint failure wrt expectations

```
-I- TraceAnalysis - Failures in check (property debugger):
-I- TraceAnalysis - `--pipe_instr fails!
-I- TraceAnalysis -     `--==_.CSR.mstatus.SDT (DUT value 0x1 at t+6, expected value 0x0) fails!
```

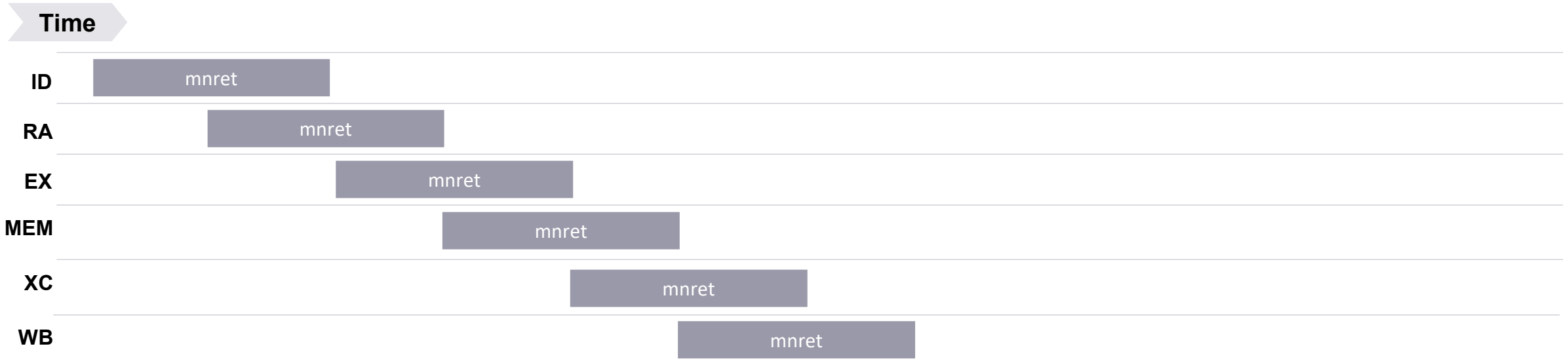

SIEMENS

Bug Case Study | Noel-V Smrnmi+Smdbltrp (non-critical)

Assessment:

SDT not handled by hardware, software could have set SDT manually.

No obvious attack vector, software inconvenience



SIEMENS

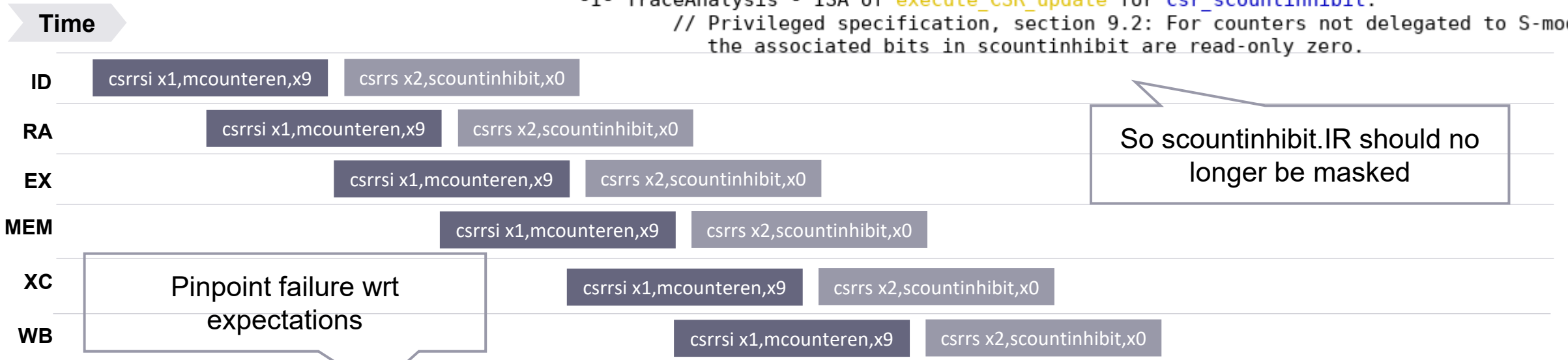
Bug Case Study | Noel-V Smcdeleg corner case (non-critical)

- Missing hazard handling between mcounteren and scountinhibit

```
preceding instructions:
        addi x2,x1,516                  (PC 0x00000000c0000000) at time 19 (t##3, mem, mem_ready)
        csrrs x1,mcounteren,x9          (PC 0x00000000c0000004) at time 21 (t##4, xc, prev_retire, xc_ready)
DUT previous instruction CSR updates (signal u0.c0.c0.RV_chk.Arch at time 21 (t##4, xc, prev_retire, xc_ready)):
          mcycle: 0x7 -> 0x8
  mhpmcounter[3]: 0x5 -> 0x6
   mcounteren.IR: 0x0 -> 0x1
```

Previous instruction enables IR

```
-I- TraceAnalysis - ISA of execute_CSR_update for csr_scountinhibit:
        // Privileged specification, section 9.2: For counters not delegated to S-mode,
           the associated bits in scountinhibit are read-only zero.
```



So scountinhibit.IR should no longer be masked

Pinpoint failure wrt expectations

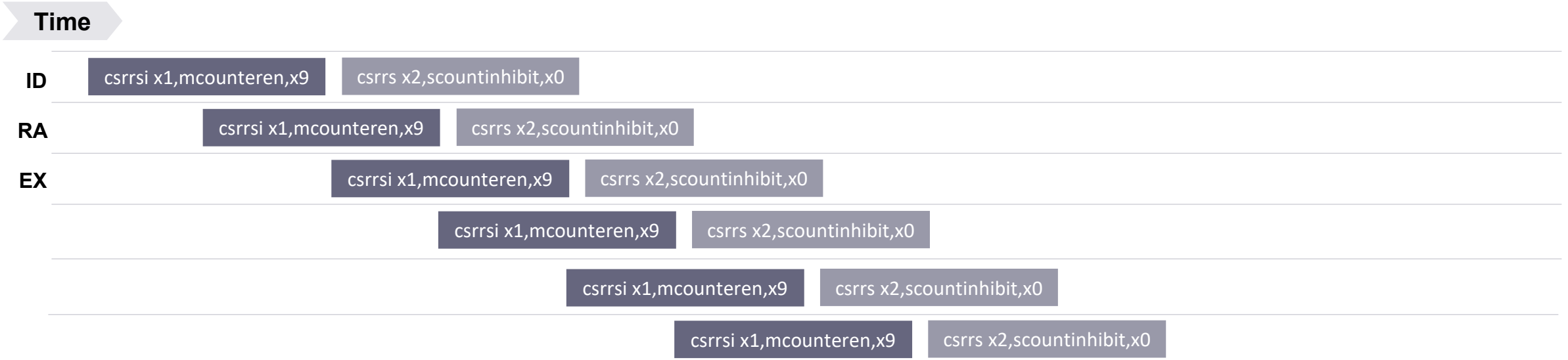
```
-I- TraceAnalysis - Failures in check (property debugger):
-I- TraceAnalysis - `--CSR_read fails!
-I- TraceAnalysis -    `--==csr_scountinhibit:rdata[2] (DUT value 0x0 at t+1, expected value 0x1) fails!
```


SIEMENS

Bug Case Study | Noel-V Smcdeleg corner case (non-critical)

Assessment:

No obvious attack vector, changing priv mode will resolve the RAW/WAW hazard,
M mode already controls the information and no information can leak to a lower priv mode.



SIEMENS

Summary

SIEMENS

Questa Processor

Accelerate, automate and increase quality of processor verification

Verification Speed-up

- No writing of testbench
- Find RTL issues earlier than in UVM flow
- Accelerate coverage closure
- Optimized formal engines
- Pinpoint bugs systematically
- Quick fix check

High degree of automation

- No writing of functional coverage model
- Designed for custom extensions
- µArchitecture extraction
- Assertion generation
- Initial value abstraction
- Disassembler annotation
- Trace analysis

Exhaustive & complete verification

- No undocumented RTL
- 100% functional coverage
- Unbounded proofs
- Finds bugs other technologies can’t
- Essential for state-of-the art processor DV
- ISA & privileged ISA compliance

SIEMENS

Summary

Over 10x improvement

On verification setup and runtime

High degree of automation

Enables automated verification of core RTL

Exhaustive & complete verification

Leaves no bugs and exposes vulnerabilities

Superior & unique

Unrivalled expertise & leverage unique solutions

SIEMENS

Disclaimer

Subject to changes and errors. The information given in this document only contains general descriptions and/or performance features which may not always specifically reflect those described, or which may undergo modification in the course of further development of the products. The requested performance features are binding only when they are expressly agreed upon in the concluded contract.

Contact

Published by Siemens EDA

Sven Beyer
Project Manager, Processor Verification
Nymphenburgerstr. 20a
80335 Munich
Germany

E-mail sven.beyer@siemens.com

SIEMENS