



**JULY 9-13, 2023**

**MOSCONE WEST CENTER  
SAN FRANCISCO, CA, USA**







**DOLPHIN** **SIEMENS**  
DESIGN

# Crossing the RISC-V customization barrier with formal

Pascal Gouédo, Dolphin Design  
Salaheddin Hetalani, Siemens EDA  
Nicolae Tusinschi, Siemens EDA  
Joe Hupcey III, Siemens EDA



# Challenges of processor verification

## Complex architecture

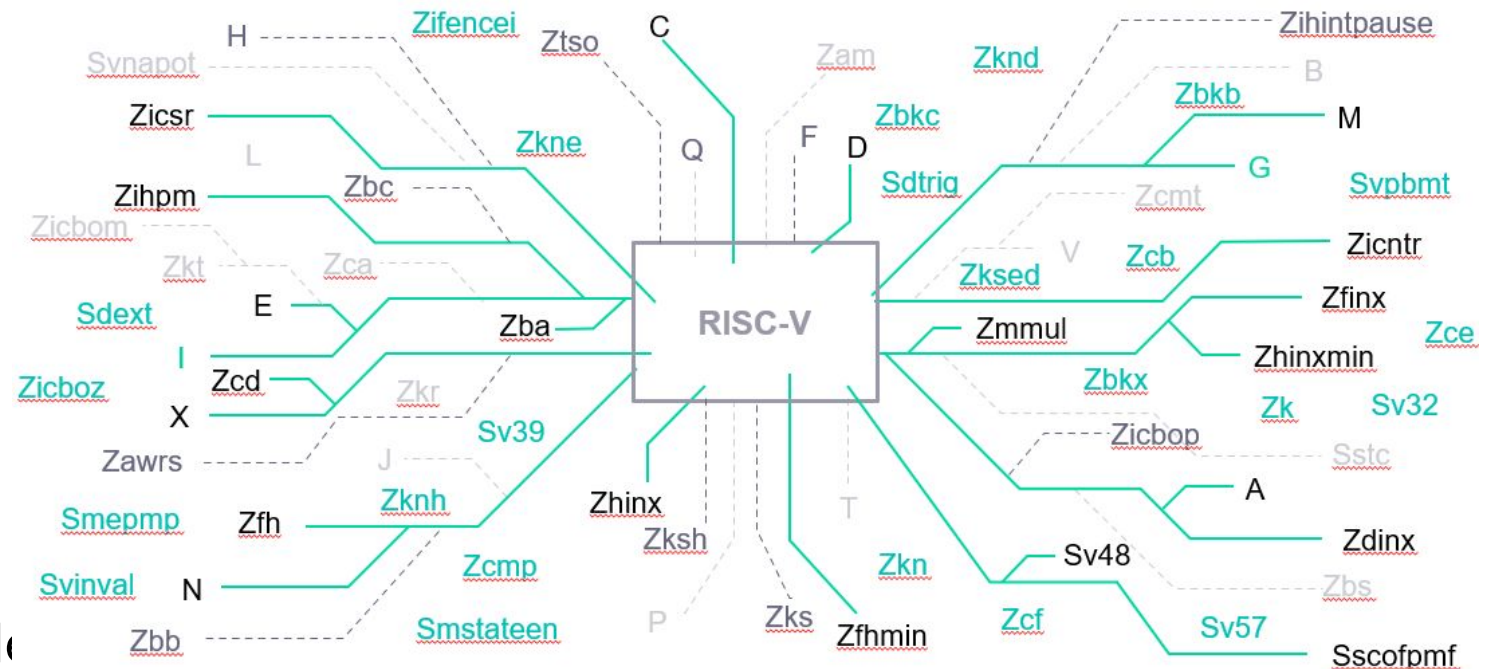
- (Custom) Instructions
- Exceptions/Interrupts

## Very complex $\mu$ Architecture

- Continuous PPA optimizations
- Pipelined implementation

- **Verification – high effort task**

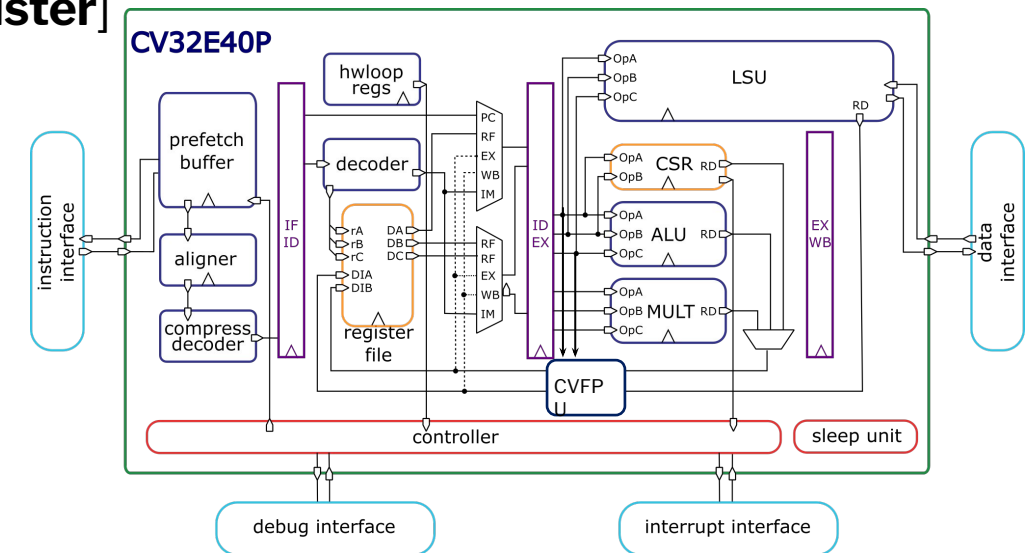
- Slow debug process
- Writing functional coverage mode
- Simulation cannot hit all pipeline corner-cases
- Functional and structural coverage closure
- Customization introduces bug in existing functionality



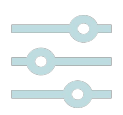
# Highly customizable CV32E40Pv2 core

- 4-stage single-issue in-order pipeline
- OBI protocol memory interfaces
- Standard external debug & interrupt support
- Floating-point extensions
- X custom instruction set extensions
- Highly configurable

**RV32IM[F]C\_Zicsr\_Zifencei[\_Zfinx][\_Xpulp][\_Xcluster]**



<https://github.com/openhwgroup/cv32e4>



**7** Standard extensions  
**+2** Custom extension



**114** Standard instructions  
**+320** Custom instructions



**21** Standard CSRs  
**+8** Custom CSRs



# X-extension verification effort is down to adding its specification

- Example instruction: Sum of dot product on 2 vectors of four unsigned 8-b data

cv.sdotup.b rd, rs1, rs2  
 $rd = rd +$

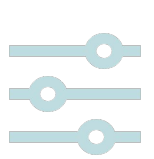
$$\sum_{k=0}^3 rs1[8 * (k + 1) - 1 : 8 * k] * rs2[8 * (k + 1) - 1 : 8 * k]$$

31	25	24	20	19	15	14	12	11	7	6	0
funct7				rs2		rs1		funct3	rd		opcode
1001100				src2		src1		001	dest		11 110 11 custom-3

- User required input: provided using app’s JSON format for regression runs

Name	Decoding	Execution	Restrictions
CV.SDOTUP.B	1001100 rs2 rs1 001 rd/rs3 1111011	$X(rd) = X(rs3) +$ $X(rs1)[7..0] * X(rs2)[7..0] +$ $X(rs1)[15..8] * X(rs2)[15..8] +$ $X(rs1)[23..16] * X(rs2)[23..16] +$ $X(rs1)[31..24] * X(rs2)[31..24]$	

# Application results



**5**

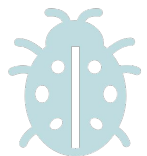
Configuration

s

**~2 H**



Runtime of 70%  
of assertions per  
CFG



**31**

Bugs



Review bug specification details

here:

<https://github.com/openhwgroup/cv32e40p/issue>

s



**~400**

Assertions per  
CFG



**100%**

Unbounded proofs

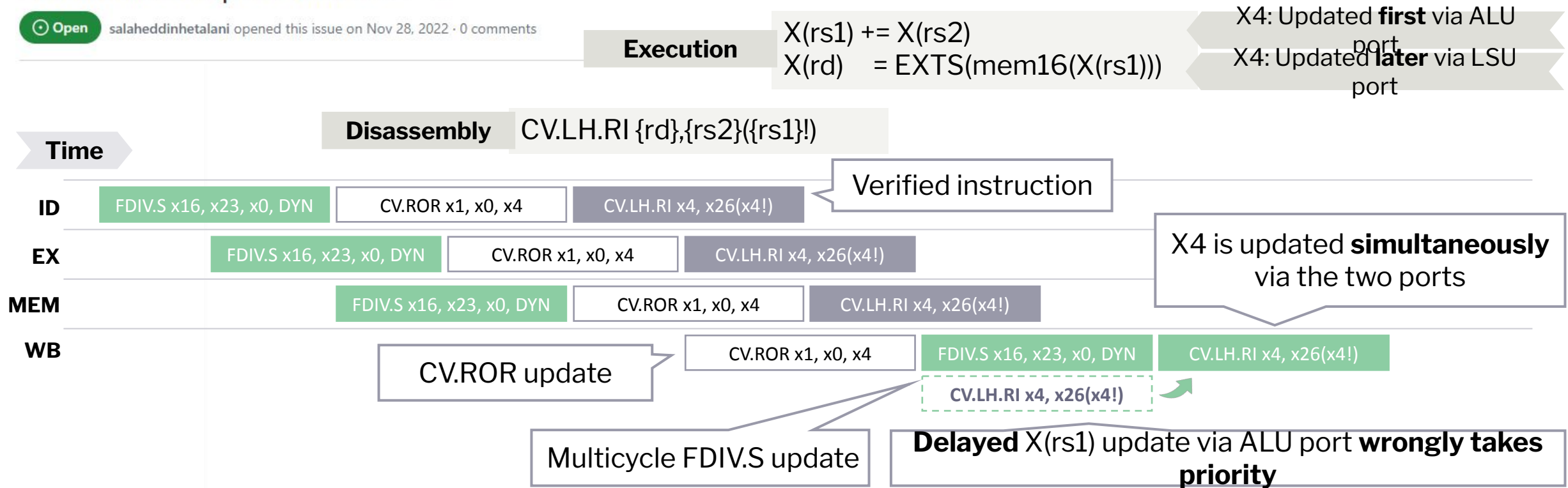
32-bit CV32E40Pv2



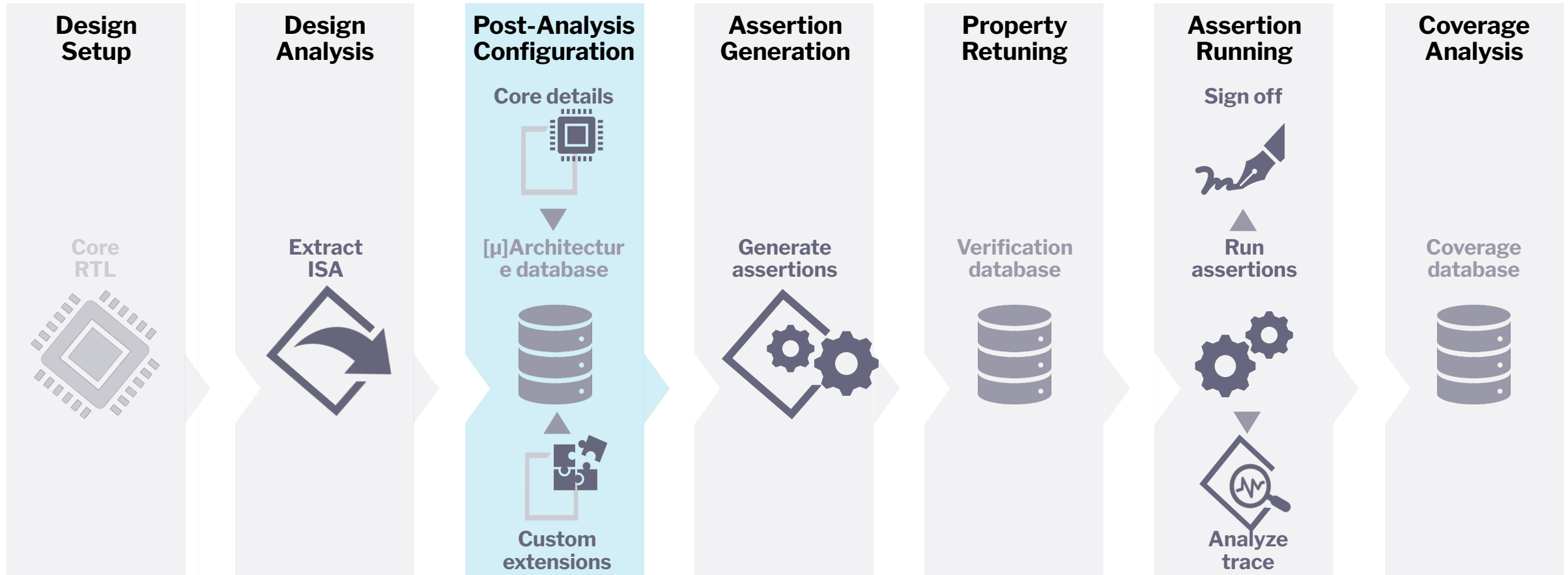
# Bug case study

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

 Open salaheddinhetalani opened this issue on Nov 28, 2022 · 0 comments



# Where to add X-extensions in the processor verification app flow





# Design Analysis

## Extract ISA



```
graph LR; A[Automated design analysis] --- B[Design ISA information]; A --- C[Design μ-Architecture information]
```

Automated design analysis

- Design ISA information
- Design  $\mu$ -Architecture information

# Post-analysis configuration

Post-Analysis Configuration

Core details

[μ]Architecture database

Custom extensions

Processor Integrity

SIEMENS

Status

Customized

Merged data for RV32IMFCZicsr\_X from file '/bata/shetalan/cve/test/shell/RISCV/CV32E\_V2/Xpulp.json'

Merge

Extract from design

ISA Custom Extensions - Instructions

	Mnemonic	Decoding	Restrictions	Disassembly	Execution
ISA	<input type="checkbox"/> CV.LB.I	imm[11:0] rs1/rd2 000 rd 0001011		cv.lb.i {rd},{imm}	let addr : xlenbits = X(rs1) + EXTS(imm); X
XLEN: 32	<input type="checkbox"/> CV.LH.I	imm[11:0] rs1/rd2 001 rd 0001011		cv.lh.i {rd},{imm}	let addr : xlenbits = X(rs1) + EXTS(imm); X
Extensions: A C D E F	<input type="checkbox"/> CV.LW.I	imm[11:0] rs1/rd2 010 rd 0001011		cv.lw.i {rd},{imm}	let addr : xlenbits = X(rs1) + EXTS(imm); X
Z: Zifencei Zicsr Zfinx Zi	<input type="checkbox"/> CV.EXTHS	0110000 00000 rs1 011 rd 010101		cv.exths {rd},{r}	X(rd) = EXTS(X(rs1)[15..0])
Custom Extensions:	<input type="checkbox"/> CV.EXTHZ	0110001 00000 rs1 011 rd 010101		cv.exthz {rd},{r}	X(rd) = EXTZ(X(rs1)[15..0])
Instructions	<input type="checkbox"/> CV.DOTSf	1001000 rs2 rs1 001 rd 1111011		cv.dotsp.b {rd},	X(rd) = EXTS(muls(X(rs1)[7..0],X(rs2)[7..0])
μ-Architecture	<input type="checkbox"/> CV.SDOTf	1001100 rs2 rs1 001 rd/rs3 111101		cv.sdotup.b {rd}	X(rd) = X(rs3) + EXTZ(mul(X(rs1)[7..0],X(rs

Add Instruction

Remove Instruction(s)

Cancel

Save

Pipeline

Mappings

Parameters

Invariants



# Automated assertion generation

## Assertion Generation

Generate assertions



Name	Proof Status	Witness Status	Validity	App
! <any status> ! <any status> ! <any validity>				
Properties				
RV_chk.ops.RESET_a	open	open	up_to_date	Processor
RV_chk.ops.BUBBLE_a	open	open	up_to_date	Processor
RV_chk.ops.INTR_Handle_a	open	open	up_to_date	Processor
RV_chk.ops.XCPT_IF_ID_a	open	open	up_to_date	Processor
RV_chk.ops.XCPT_WB_a	open	open	up_to_date	Processor
RV_chk.ops.XCPT_MEM_a	open	open	up_to_date	Processor
RV_chk.RV32I.FENCE_a	open	open	up_to_date	Processor
RV_chk.RV32I.WFI_a	open	open	up_to_date	Processor
RV_chk.RV32I.ECALL_a	open	open	up_to_date	Processor
RV_chk.RV32I.xRET_a	open	open	up_to_date	Processor
RV_chk.RV32I.EBREAK_BreakPoint_a	open	open	up_to_date	Processor
RV_chk.RV32I.EBREAK_HaltReq_a	open	open	up_to_date	Processor
RV_chk.RV32I.EBREAK_ForcedEntry_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.BRANCH_a	open	open	up_to_date	Processor
RV_chk.RV32I.JUMP_a	open	open	up_to_date	Processor
RV_chk.RV32I.ARITH_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.RV32M.DIV_a	open	open	up_to_date	Processor
RV_chk.RV32M.MUL_a	open	open	up_to_date	Processor
RV_chk.RV32C.ARITH_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.RV32C.BRANCH_a	open	open	up_to_date	Processor
RV_chk.RV32C.JUMP_a	open	open	up_to_date	Processor
RV_chk.RV32X.CV_EXTHS_a	open	open	up_to_date	Processor
RV_chk.RV32X.CV_EXTHZ_a	open	open	up_to_date	Processor
SVA Named Properties				
SVA Sequences				

RV32IMC  
\_Zicsr  
\_Zifencei  
**27**  
Assertions



# Summary

- **Successful formal based custom extensions verification**
  - Verification effort is down to adding extension specification
  - 31 bugs were identified by formal that simulation missed
- **Considerable verification speed-up compared to simulation**
  - No writing of testbench
  - Optimized formal engines and assertions
  - Pinpoint bugs - quick fix check
  - Unbounded proofs
- **High degree of automation**
  - No writing of functional coverage model
  - $\mu$ -architecture details extraction
  - Assertion generation
  - Trace analysis & disassembler annotation

