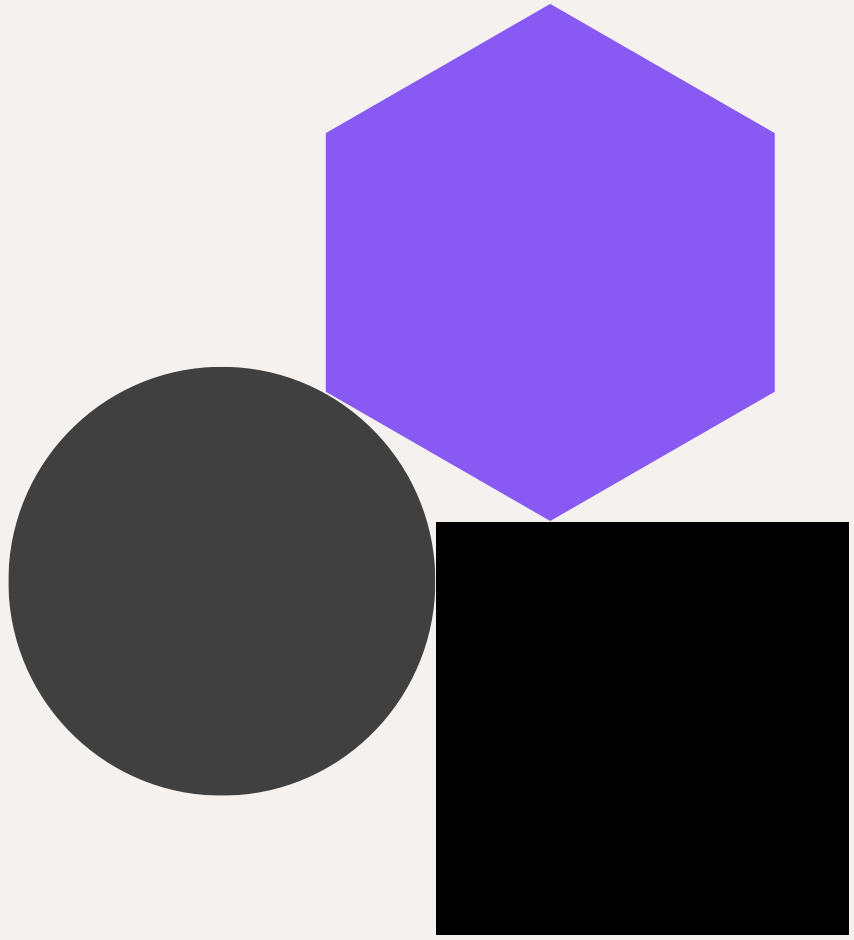
CHERI-RISC-V Changes 0.9.3 to 0.9.10

Codasip



Many changes, but most just involve renaming

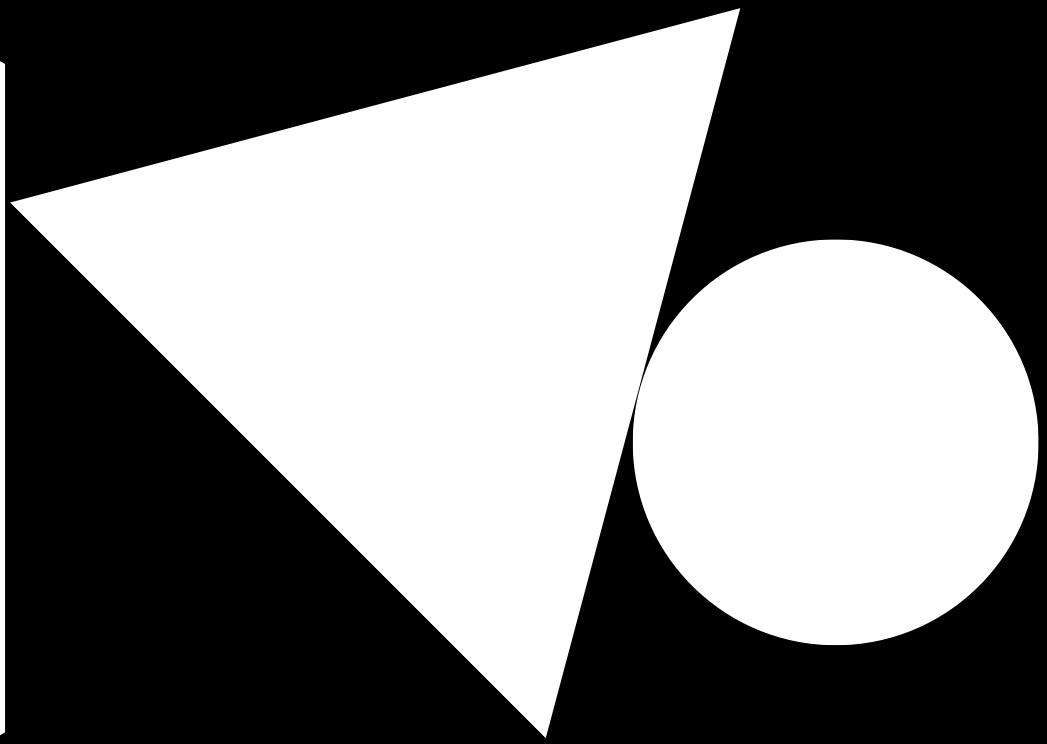
Tariq Kurd

7th October 2026

For Public Presentation

Codasip

→ 0.9.3 to 0.9.8.2 summary



Combining the overall changes of these revisions

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2

Codasip

- CHERI is now a new base architecture
  - RV64LYA, RV32LYA and CHERIoT variants
- Delete xtdc CSRs (mtdc, stdc)

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 - ISA changes

Codasip

- YTOPR added
- YBLD is in the base
- YBNDSWI immediate encoding changed several times
- Change GCPERM (YPEMR) so that unallocated permissions [23:19] read as 1 not 0
  - For forwards compatibility
- Spec change from ACPERM to YPERMC
  - The permission rules have been refactored to allow rules to be added by the capability encoding
  - Rule added for RV64 that M requires X
- Spec change from CBLD to YBLD
  - YBLD output is always unsealed

© 2026 Codasip. All rights reserved.

→ 0.95 to 0.9.8.2 - ISA renaming

Codasip

- All registers have the original RISC-V name
  - rs1 not cs1, rd not cd etc
  - pc not pcc
  - sp not csp
- Removing the renaming is to avoid problems converting new extensions to be CHERI compatible, and to remove asm porting issues
- Capability width registers are now YLEN-bit (+ capability tag)
  - CLEN -> YLEN to match RVY

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 - ISA renaming

Codasip

- *Most* existing instructions are no longer renamed
  - E.g. AUIPC was renamed to AUIPCC, it is now always AUIPC
    - This affects AUIPC, JAL, JALR, C.ADDI16SPN, C.ADDI4SPN, C.JR, C.JAL, C.JALR
- Where instructions are renamed they always have a **Y** in them
  - The exception for the rule above is C.YMV which is renamed from C.MV to show that it's a YLEN copy not an XLEN copy

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 - ISA renaming – load/store/atomic

Codasip

- All mnemonics renamed and all encodings changed
- C type becomes Y type – so all load/store use Y
- LY, SY, C.LY, C.LYSP, C.SY, C.SYSP
- LR.Y, SC.Y, AMOSWAP.Y
- HLV.Y, HSV.Y

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 - New encodings for YSH[1234]ADD

Codasip

- All have new encodings, none are CHERI mode dependent
- There’s no Y equivalent for ADD.UW
- There is no integer SH4ADD

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 - Extensions

Codasip

- Zish4add – removed – encodings now in Zba
  - Adding Zba also gives YSH[1234]ADD[.UW]
- Zylevels1 separated from the base – subsequently renamed to Zylg in 0.9.10
- Zys added for YSENTRY – subsequently merged again in 0.9.9
- Zcherihybrid renamed to Zyhybrid
- Zcheripte replaces by Svyrg

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 Levels

Codasip

- Zylevels1 extension added and aligned with CHERIoT
  - Was previously in the base ISA
  - LG-perm (replacement for EL-perm)
  - SL-perm (as before – but only 1-bit in the spec)
    - SL-perm requires C and W permission (previously was just C)
  - GL-flag (replacement for CL but inverted)
  - missing LG on the auth cap makes loaded capabilities local even if the auth cap has the GL (global) flag set

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 Integrity checks

Codasip

- Integrity checks were different for different instructions, split into 3 categories
- They are now consistent – all or none are checked.
  - The only exception is YBLD (CBuildCap) which *must* check operand being validated.

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 GCPERM / YPERMR

Codasip

Quadrant 1: Executable capabilities										
bit[0] - M-bit (0-*Capability Pointer Mode*, 1-*Integer Pointer Mode*)										
Bits[4:3]	R	W	C	LM	EL	SL	X	ASR	Mode[1]	
0-1	✔	✔	✔	✔	✔	∞	✔	✔	Mode[1]	Execute + ASR (see Infinite)
2-3	✔		✔	✔	✔	∞[2]	✔		Mode[1]	Execute + Data & Cap RO
4-5	✔	✔	✔	✔	✔	∞	✔		Mode[1]	Execute + Data & Cap RW
6-7	✔	✔				0[2]	✔		Mode[1]	Execute + Data RW

Quadrant 1: Executable capabilities										
bit[0] - M-bit (0-*(CHERI) Capability Mode*, 1-*(Non-CHERI) Address Mode*)										
Field[2:0]	R	W	C	LM	LG	SL	X	ASR	Mode[1]	Notes
0-1	✔	✔	✔	✔	✔	✔	✔	✔	Mode[1]	Execute + Data & Cap RW + ASR
2-3	✔		✔	✔	✔		✔		Mode[1]	Execute + Data & Cap RO
4-5	✔	✔	✔	✔	✔	✔	✔		Mode[1]	Execute + Data & Cap RW
6-7	✔	✔					✔		Mode[1]	Execute + Data RW

- GCPERM / YPERMR a bit different for RV32YLYA – this case returns zero
- Also note that SL is only 1-bit now

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 Exceptions

Codasip

- Simpler exception types

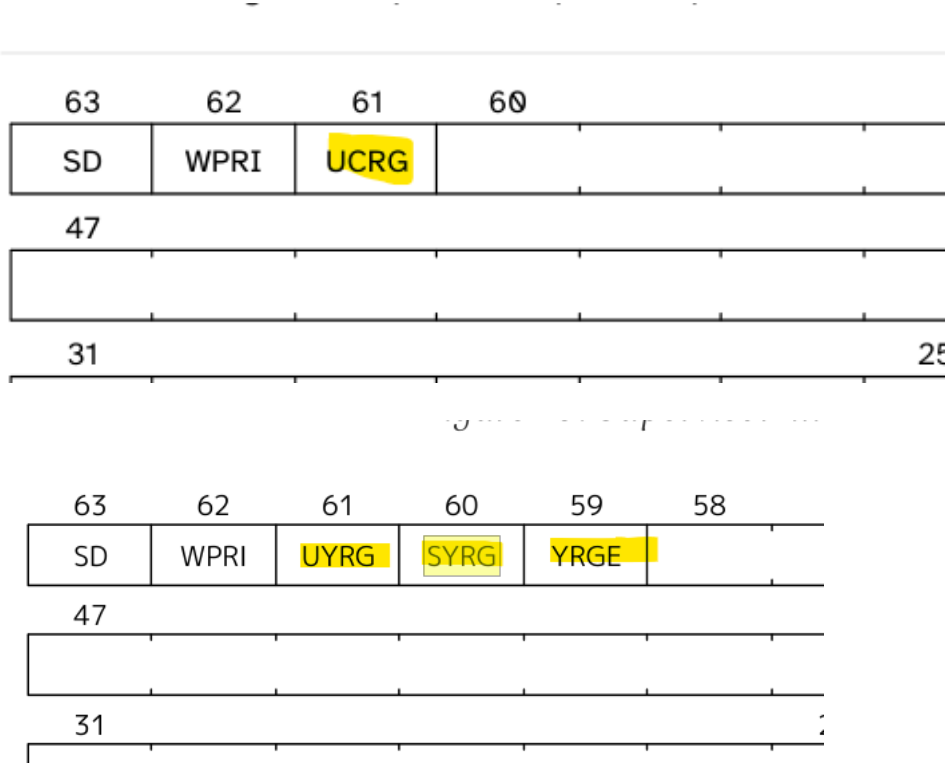
32	CHERI Instruction Access Fault
33	CHERI Load Access Fault
34	CHERI Store/AMO Access Fault
35	CHERI Load Capability Fault
36	CHERI Store/AMO Page Fault

- No more sealed, permissions etc. fault types
- No more use of Xtval2
- 32 is the old PCC fault
- 33 is any fault from a load, 34 is any fault from a store/amo
- 35 is a lower priority fault taken on loads after translation
- 36 is a new PTE fault taken during translation on stores

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 Xstatus

- UCRG renamed to UYRG
- SCRG added (supervisor version of UYRG
- YRGE added – to enable Svyrg



© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 Debug

Codasip

- dinfc replace by drootc and drootcsel
  - Because CHERIoT has multiple roots (i.e. multiple infinite caps – one for code and one for data)
  - We will have multiple roots for application cores when we add sealing/unsealing permission and forwards/backwards sentries.

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 CHERI enables

Codasip

- CRE -> Y
- Mseccfg.CRE -> MISA.Y
- Xenvcfg.CRE -> Xenvcfg.Y

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 Branch target exceptions

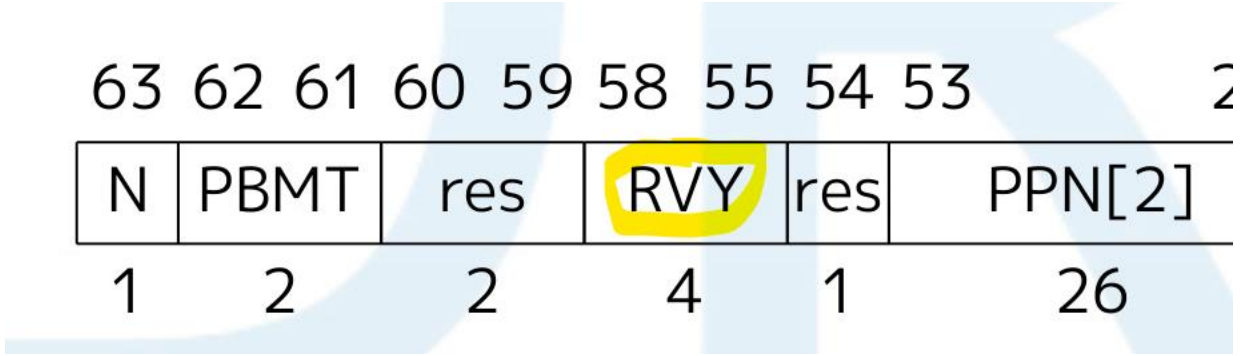
Codasip

- No exceptions on branches, and there's only one type of PC fault
  - No ALU exceptions are possible any more
- So if JALR jumps to a bad capability then it gets installed in the PC and then the exception is taken at the target
  - This is identified as a possible problem for trapping on jumps out of a compartment
  - We may add an extension in future to re-introduce this – but for now it has gone
- If a conditional branch jumps out of bounds then the exception is also taken at the target

© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 Svyrg extension

- Allocate 4 PTE bits which (almost) match Morello, allowing the existing revocation software stack to run
  - Replaces Zcheripte
- Reserve a 4-bit field in the PTE for RVY extensions



© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 PTE update

Codasip

- Rename Xstatus fields and allocate another bit:

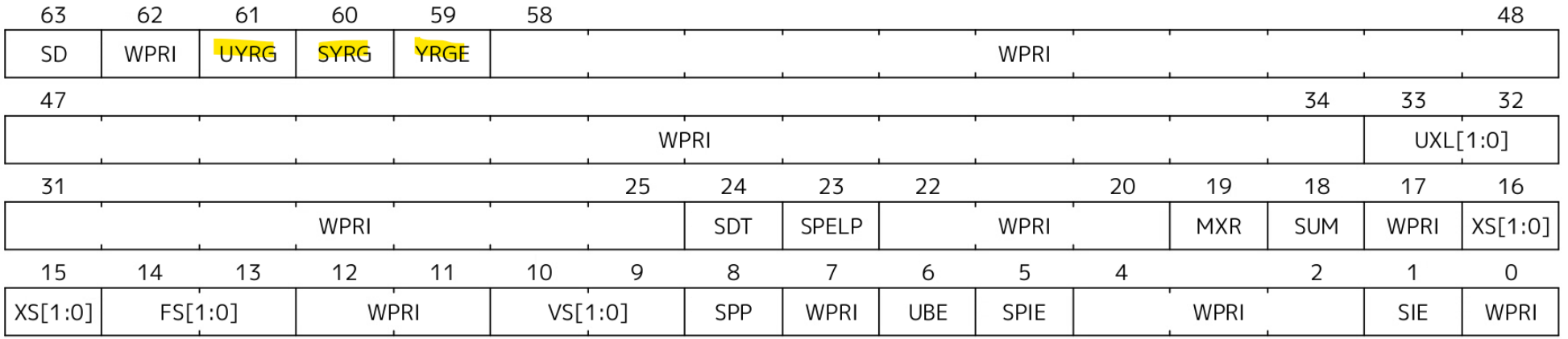


Figure 50. Supervisor-mode status (`sstatus`) register when SXLEN=64.

- UYRG – userspace capability read generation
- SYRG – supervisor capability read generation

© 2026 Codasip. All rights reserved.
For Public Presentation

→ 0.9.3 to 0.9.8.2 encoding space

Codasip

- CUSTOM-1 -> RVY-C : all reserved
- CUSTOM-2 -> RVY-B : all reserved
- CUSTOM-3 -> RVY-A : all CHERI encodings in here
- Leave CUSTOM-0 as custom
- Possibly make CUSTOM-1 custom again later

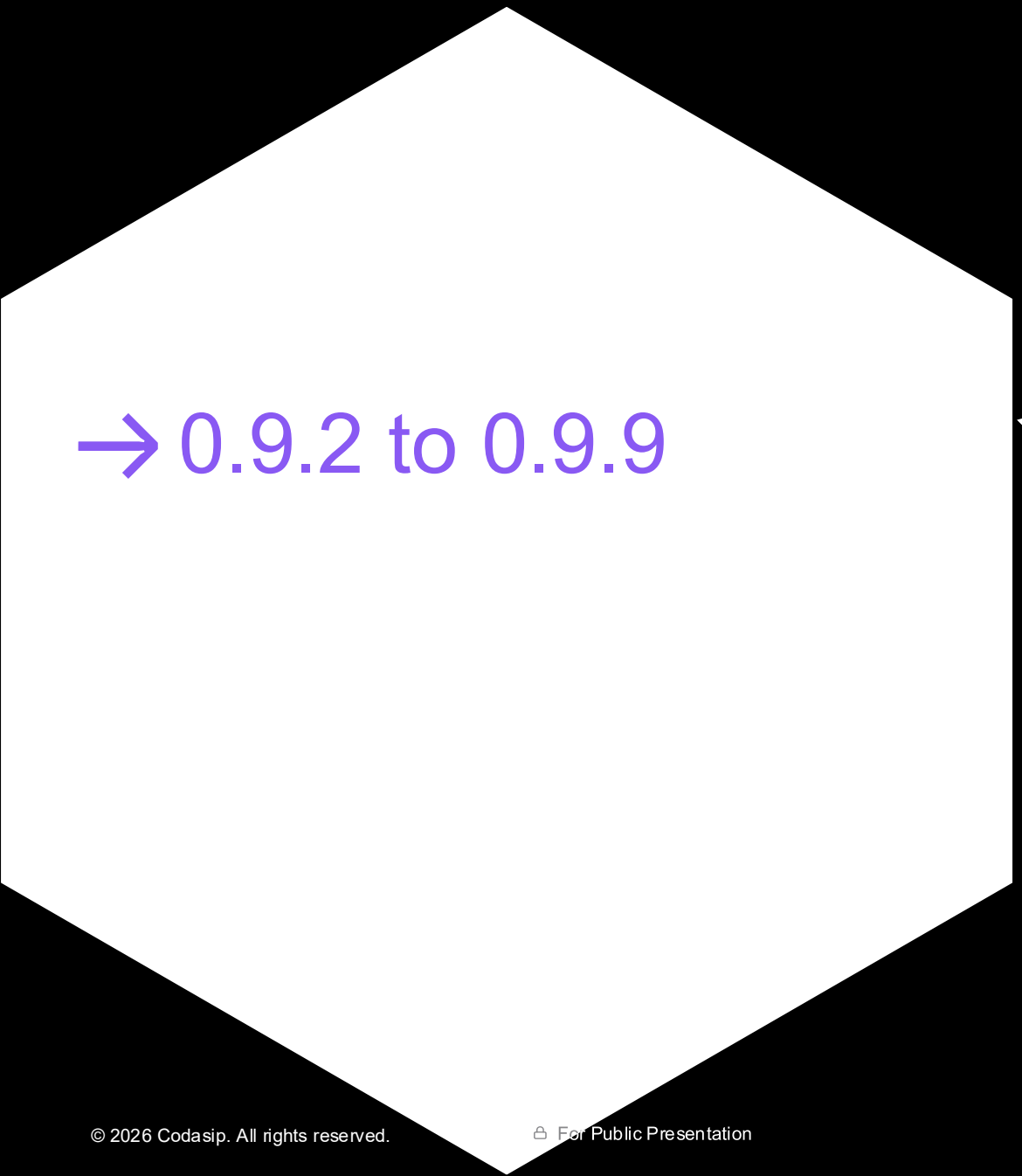
© 2026 Codasip. All rights reserved.

→ 0.9.3 to 0.9.8.2 Hybrid mode

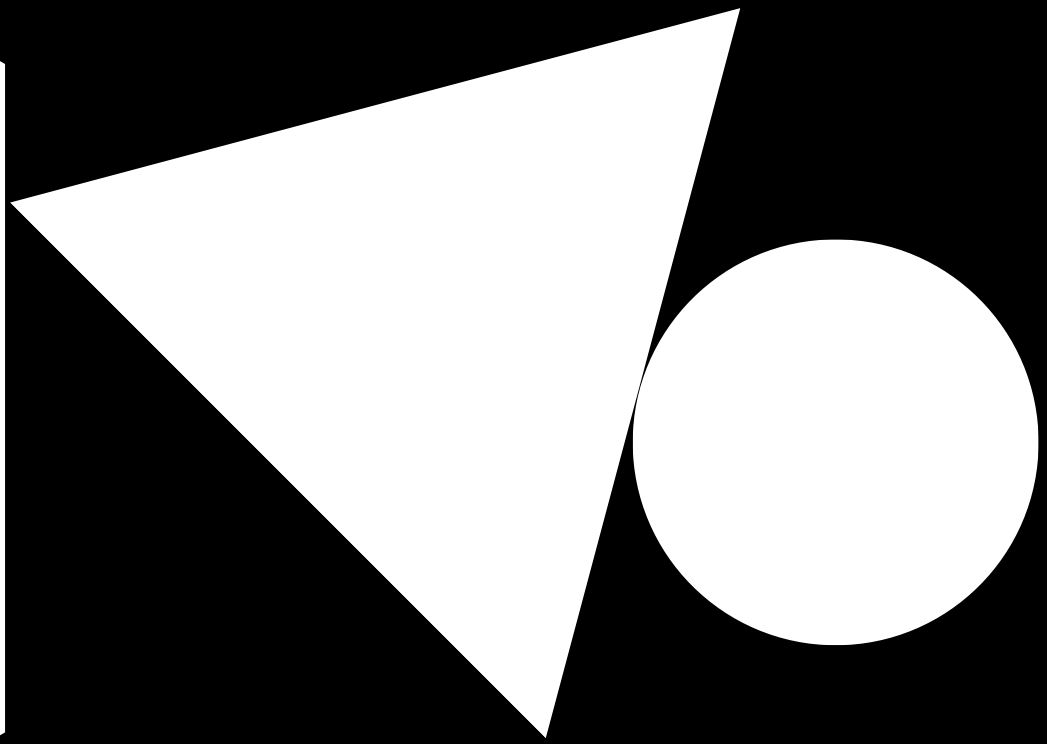
Codasip

- M-bit now P-bit – for pointer mode
- CHERI Execution Mode now the CHERI Pointer Mode, and they are:
  - Capability Pointer Mode
  - Integral Pointer Mode
- Important: when in Integral Pointer Mode, the base arch is RVY
  - Custom-1/2/3 are RVY-C/B/A
  - 16-bit load/store encodings still revert to RVI (e.g. C.LY reverts to C.FSD)
  - So we end up with a bit of a hybrid monster for encodings
- The only way to get custom-1/2/3 back is to fully disable CHERI.

© 2026 Codasip. All rights reserved.



→ 0.9.2 to 0.9.9



© 2026 Codasip. All rights reserved.

For Public Presentation

→ 0.9.8.2 to 0.9.9-ar20260707

Codasip

- ARC Review feedback
  - MISA.Y=0
    - Disabling MISA.Y disables all trace of CHERI
    - Reset is MISA.Y=1, and it can only be disabled if DDC/PCC etc. bounds are infinite
    - We can be more loose about letting it be set to 0, e.g. detect any write to DDC/xTVEC/xEPC or any JALR
    - See 16.3.1
  - Merge Zysentry into the base
    - YSENTRY now has rs2 specified, although it must be x0 – **encoding has changed as a result**
    - The intention is to move to YSEAL with rs2!=x0 to add an authorising cap for sealing/unsealing
    - See 2.8.2.9
  - Rename the capability encoding format name
    - It was RV64LYmw14rc1ps which is a string of parameters concatenated
      - Very unpopular
      - Now it's RV64LYA and the parameters are still present in a table in the chapter
      - See chapter 3

© 2026 Codasip. All rights reserved.

→ 0.9.8.2 to 0.9.9-ar20260707

Codasip

- The Y bit was in bit 28 of Xenvcfg, moved it to bit 9
- The metadata arrangement for RV64LYA was suboptimal - the SDP, AP and GL fields had no reserved space next to them. All may grow in future. This is a simple rearrangement to allow these fields to grow into reserved spaces.
  - See Chapter 3

© 2026 Codasip. All rights reserved.

→ 0.9.8.2 to 0.9.9-ar20260707

Codasip

- Debug spec
  - Abstract commands for capability types weren’t defined
    - We’ve defined aarsize/aamsize=7 to support capability types – an optional feature
    - Aarsize/aamsize already supported 128-bits i.e. YLEN access
    - See 1.2.2 of the debug spec

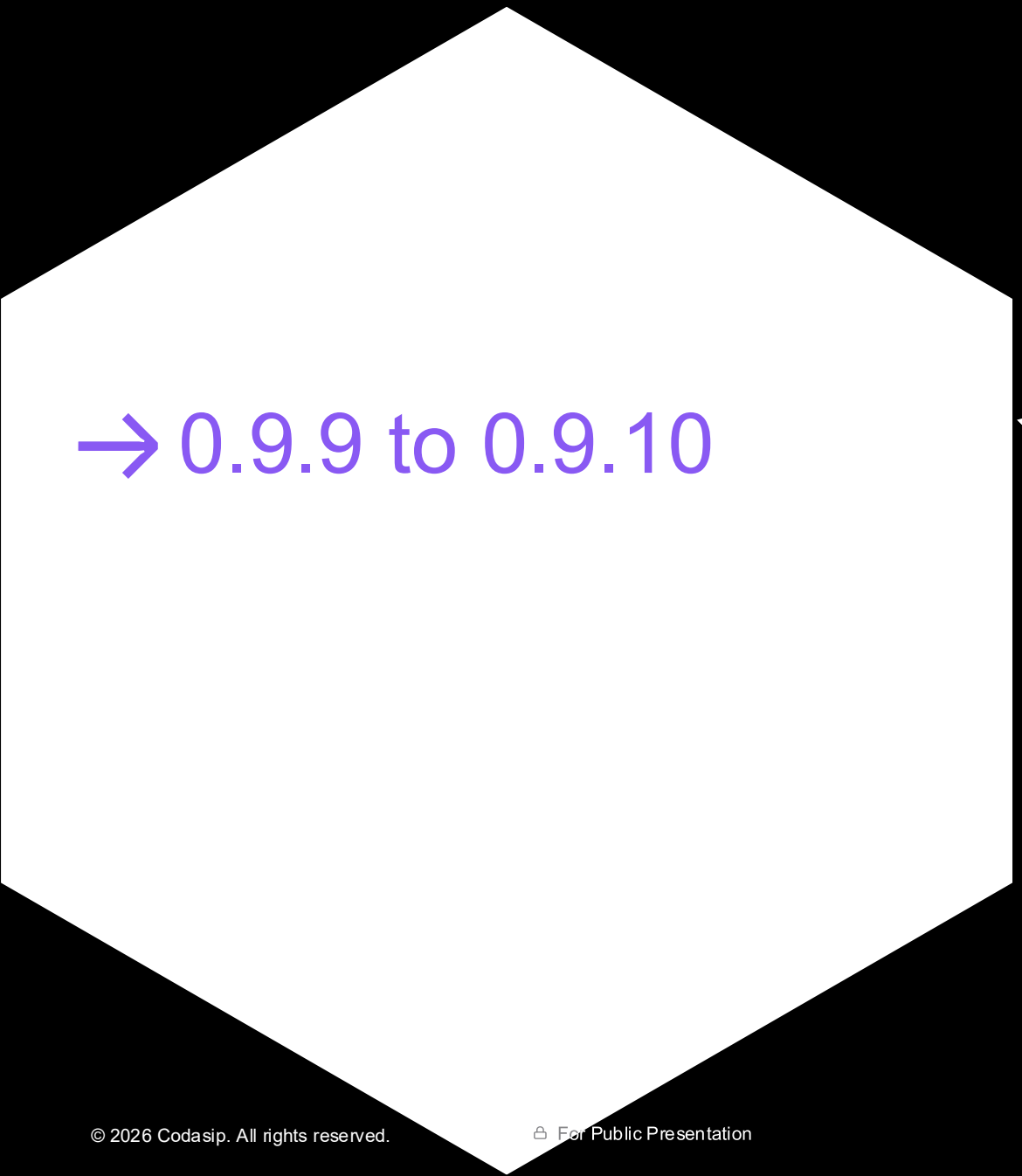
© 2026 Codasip. All rights reserved.

→ 0.9.9-ar20260707 to 0.9.9-ar20260810

Codasip

- If sstatus.YRGE changes state then an SFENCE.VMA is required to make the new state visible
- Fixed a spec bug in integer mode so that LY, SY etc. permit rs1=x0.

© 2026 Codasip. All rights reserved.



→ 0.9.9 to 0.9.10

© 2026 Codasip. All rights reserved.

→ 0.9.9-ar20260810 to 0.9.10-ar20262909 - unpriv

Codasip

- More ISA renaming unfortunately:
  - Drop PACKY name, just keep YHIW, no encoding change
  - Remove SRLIY encoding and re-encode YHIR as a dedicated instruction – **re-encoded** as not a shift
  - More renaming:
    - YSUNSEAL -> YUNSEALS (S for Superset) no encoding change
    - YSENTRY -> YSEALE (E for Entry) no encoding change
- Zylevels1 -> Zylg (local/global) as having a digit as a suffix isn't legal
- 16-bit HINTs which clear metadata are now **reserved**
  - C.SLLI, C.SRAI, C.SRLI where shamt=0 but rd!=x0 – see next slide
  - C.ADDI with imm=0
- BEQ/BNE no longer have **reserved** states where rs1>=rs2 – removal of a special case
- YPERMC zeroes the output permissions on integrity check failure
  - This makes it consistent with YPERMR (found during SAIL development)
- LY/SY in integer mode can have rs1=x0 without being reserved (bug fix)
- CSRs now in 2 categories – XLEN and YLEN (not widened XLEN->YLEN)
  - Widened CSR concept pushed into Zyhybrid only as cap mode is unaffected
- **CSR addresses have all changed**

© 2026 Codasip. All rights reserved.

→ 0.9.9-ar20260810 to 0.9.10-ar20260929 – new CSR addresses

- utidc: 0x07f
- stidc: 0x17f
- vstidc: 0x27f
- mtidc: 0x37f
- ddc: 0x03f
- drootcsel: 0x7bd
- dddc: 0x7bf
- drootc: 0x7be

© 2026 Codasip. All rights reserved.

→ 0.9.9-ar20260810 to 0.9.10-ar20260929 - unpriv

Codasip

Table 81. RVC HINT instructions.

Instruction	Constraints	Code Points	Purpose
C.NOP	*imm*≠0	63	*Designated for future standard use*
C.ADDI	*rd*≠ `x0` , *imm*=0[1]	31	
C.LI	*rd*= `x0`	64	
C.LUI	*rd*= `x0` , *imm*≠0	63	
C.MV	*rd*= `x0` , *rs2*≠ `x0`	31	
C.ADD	*rd*= `x0` , *rs2*≠ `x0` , *rs2*≠ `x2-x5`	27	
C.ADD	*rd*= `x0` , *rs2*= `x2-x5`	4	(rs2=x2) C.NTL.P1 (rs2=x3) C.NTL.PALL (rs2=x4) C.NTL.S1 (rs2=x5) C.NTL.ALL
C.SLLI	*rd*= `x0` or *shamt*=0[1]	63 (RV32), 95 (RV64)	*Designated for custom use*
C.SRLI	*shamt*=0[1]	8	
C.SRAI	*shamt*=0[1]	8	

[1] If the base architecture is RVY, encodings with *rd*≠ `x0` are *reserved* instead of HINTs, because executing an integer arithmetic instruction that writes *rd*≠ `x0` sets the destination register's metadata and capability tag to zero (leaving only the 32 (RV32) or 64 (RV64) C.SLLI code points with *rd*= `x0` as HINTs).

© 2026 Codasip. All rights reserved.

→ 0.9.9-ar20260810 to 0.9.10-ar20260929 YHIR -> new encoding

Codasip

2.8.4.4. YHIR

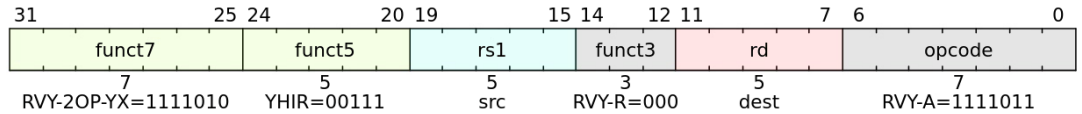
Synopsis

Read capability metadata

Mnemonic

`YHIR rd, rs1`

Encoding



The instruction encoding is not final and is highly likely to change prior to v1.0

Description

Copy the metadata (bits [YLEN-1:XLEN]) of capability `rs1` into `rd`.

The value of `rs1.tag` does not affect the result.

Operation

```
let capVal = C(cs1);
X(rd) = capToMetadataBits(capVal).bits;
RETIRE_SUCCESS
```

© 2026 Codasip. All rights reserved.

For Public Presentation

→ 0.9.9-ar20260810 to 0.9.10-ar20260929 - priv

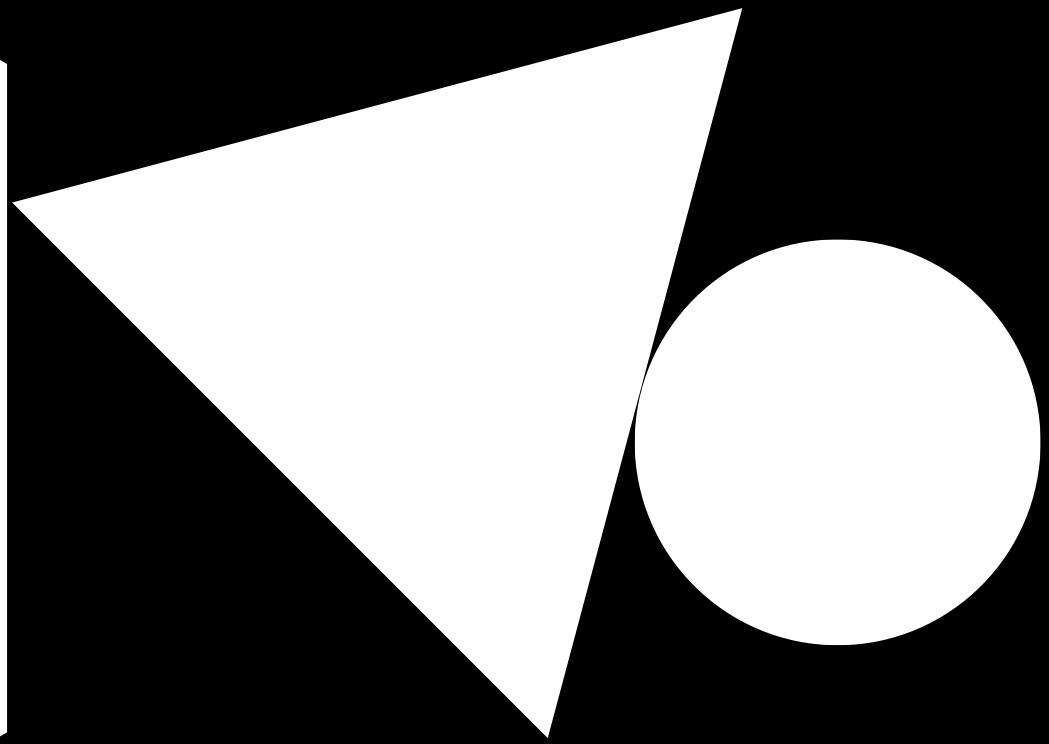
Codasip

- Priv extensions get names
- SFENCE.VMA is required after update sstatus.YRGE
- WARL capability CSRs are clarified as tag-clearing if an unprogrammable value is written to them.
  - This allows the frontend not to store the reserved bits of bad PCC targets, the reserved bits of xepc can be read-only-zero and so do *not* get reported on a fault.

- All changes can be found here:
  - https://github.com/riscv/riscv-cheri/pulls?q=is%3Apr+state%3Aclosed+label%3Achange-post-0.9.9

© 2026 Codasip. All rights reserved.

→ Summary



0.9.3 to the latest

© 2026 Codasip. All rights reserved.

→ ISA renaming – 0.9.3 to 0.9.10

Codasip

0.9.10	0.9.3	0.9.10	0.9.3
YTAGR	GCTAG	YEQ	SCEQ
YPERMR	GCPERM	YSEALE	SENTRY
YMV	CMV	YSS	SCSS
YBNDSW	SCBNDS	YBASER	GCBASE
YBNDSRW	SCBNDSR	YLENR	GCLEN
YMODEW	SCMODE	YTYPER	GCTYPE
YMODER	GCMODE	YMODESWY	MODESW.CAP
YADDRW	SCADDR	YMODESWI	MODESW.INT
YAMASK	CRAM		

© 2026 Codasip. All rights reserved.

For Public Presentation

→ ISA renaming – 0.9.3 to 0.9.10

Codasip

0.9.10	0.9.3	0.9.10	0.9.3
YADD	CADD	LR.Y	LR.C
YADDI	CADDI	AMOSWAP.Y	AMOSWAP.C
YSH[1234]ADD	SH[1234]ADD	SC.Y	SC.C
YSH[1234]ADD.UW	SH[1234]ADD.UW	LY	LC
YHIR	GCHI	SY	SC
YHIW	SCHI	HLV.Y	HLV.C
C.YMV	C.MV	HSV.Y	HSV.C

© 2026 Codasip. All rights reserved.

→ ISA changes – 0.9.3 to 0.9.10

Codasip

0.9.10	0.9.3	Change
YBNDSWI	SCBNDSI	Immediate format has been re-encoded
YBLD	CBLD	Output is always unsealed
YPERMC	ACPERM	Change from bit AND to bit clear
YTOPR	N/A	New instruction
YUNSEALS	N/A	New instruction

© 2026 Codasip. All rights reserved.

→ Change summary 0.9.3 to 0.9.10

Codasip

1. RV64LYA Base architecture defined, and metadata re-organised
2. All encodings changed (moved into CUSTOM-3) and Hybrid mode redefined to be an RVY architecture not RVI
3. PC, and all registers and most redefined instructions (e.g. AUIPC) now have the normal RISC-V names
4. All mnemonics renamed and re-encoded
5. All CSR addresses changed and mtdc, stdc deleted
6. 16-bit HINTs which clear metadata are now reserved
    1. Reviewing this for c.addi rd, 0 as it's a standard HINT
7. Some instructions redefined
    1. YPERMC (bit clearing)
    2. YBLD (unsealed output)
    3. YBNDSWI (immediate format)
    4. YPERMR for some reserved permissions reading as 1
    5. YPERMC zeros output permissions on integrity failure
8. Zysentry (YSEALE) now in the base

© 2026 Codasip. All rights reserved.

→ Change summary 0.9.3 to 0.9.10

Codasip

1. Zylg (formerly Zylevels1) slightly redefined to match CHERIoT
2. Integrity checks made uniform
3. 4-bit PTE scheme defined matching Morello (Svyrg)
4. Privileged extensions named
5. Exception reporting simplified and exceptions removed from branches and jumps
6. CHERI enabled by MISA.Y=1
    1. Can only be set to 0 if DDC/PCC bounds are infinite.
7. xENVCFG.Y moved into bit 9
8. dinfc replace by drootc and drootcsel, and abstract commands added

© 2026 Codasip. All rights reserved.