CORES TG – June 11 2020

Arjan Bink arjan.bink@silabs.com

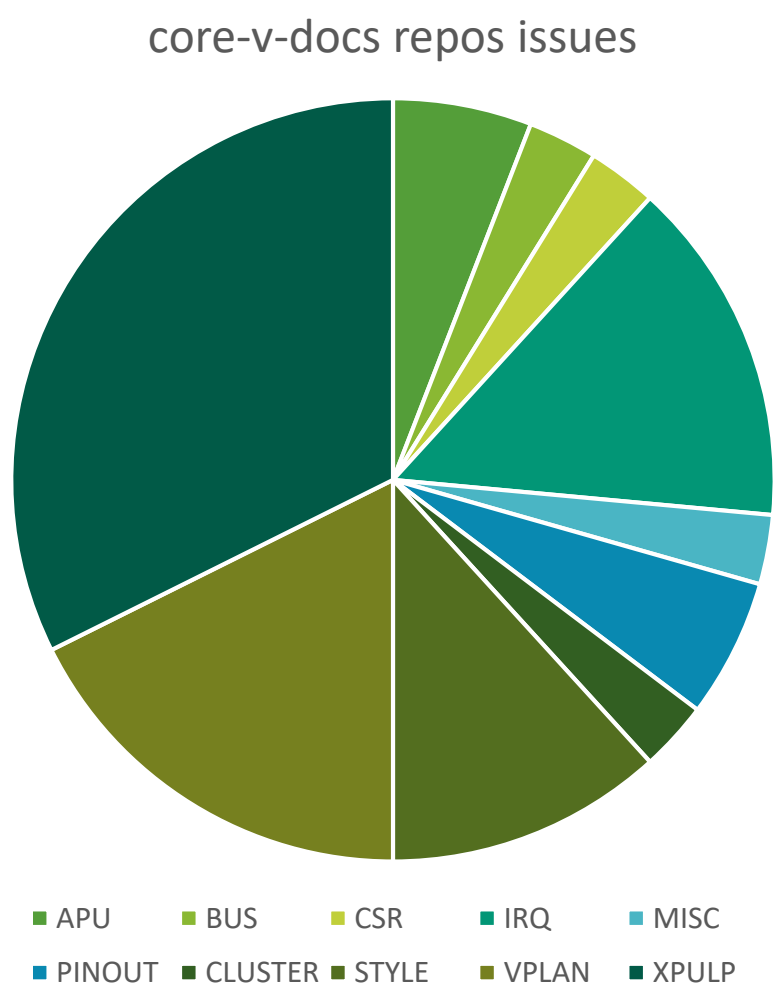
Jérôme Quevremont jerome.quevremont@thalesgroup.com

Davide Schiavone davide@openhwgroup.org

Outline

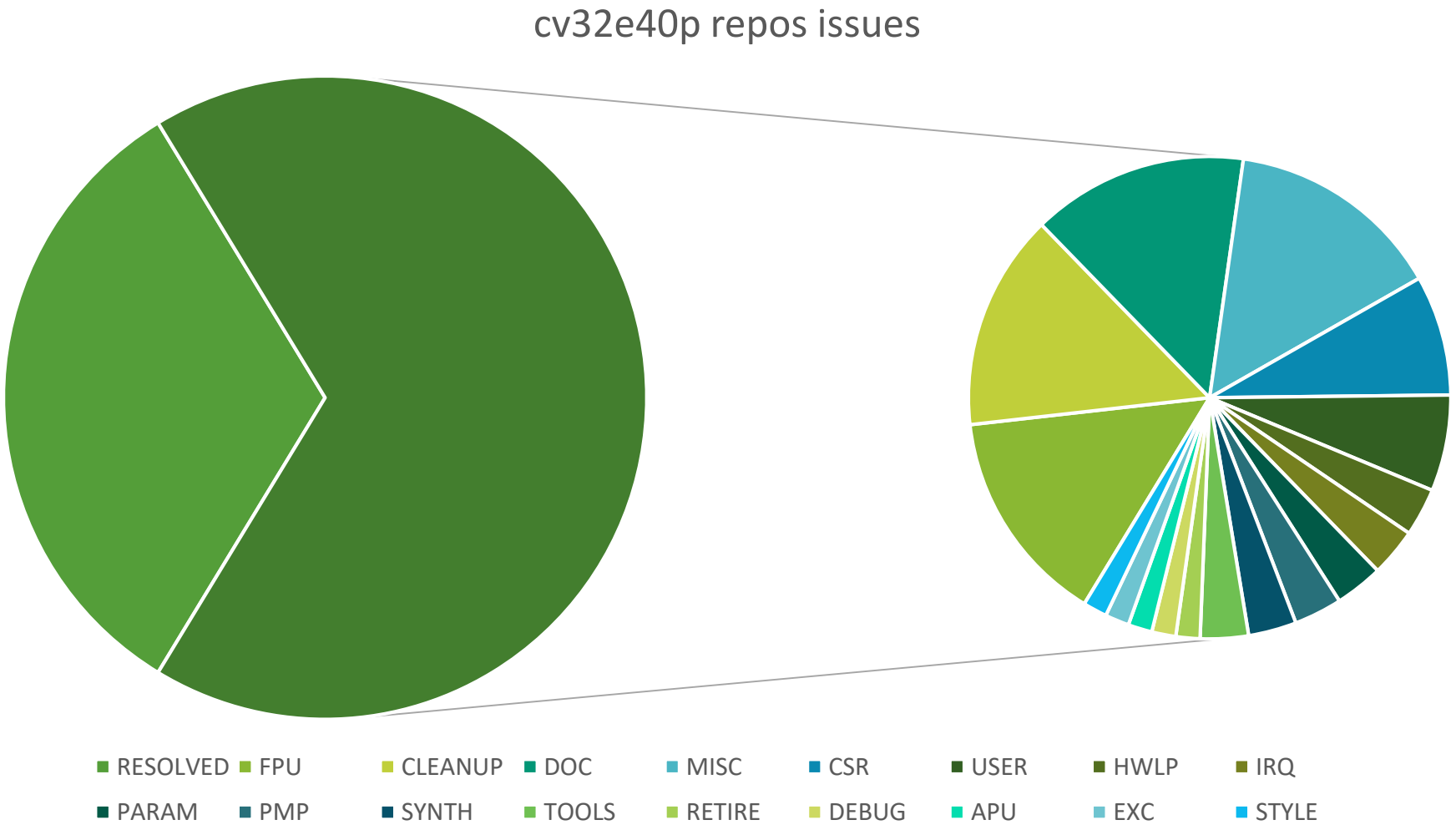
- CV32E40P sprint
  - core-v-docs repos - open issues
  - cv32e40p repos - open issues
- github way of working
  - Issue classification - proposal
  - Github issue work flow - proposal
- IRQ interface - discussion
- Thales presentation
  - CV64A/CV32A
- Upcoming

core-v-docs repos - 34 non-closed issues

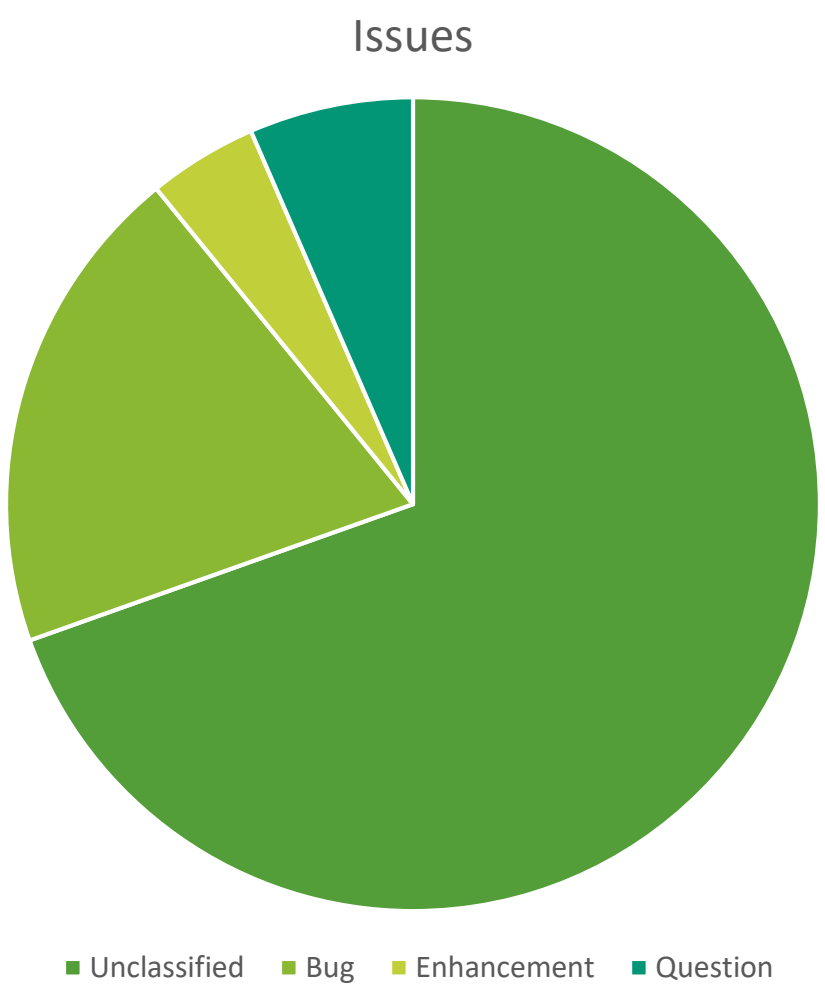


- Most issues relate to User Manual
  - 6 issues are VPLAN
- Sprint focusing on User manual
  - Target completion by June 19
  - >30% already resolved

cv32e40p repos - 92 non-closed issues



92 non-closed issues (~18 bugs)



- Need better classification
  - 70% of issues is not classified
  - Bugs are not always marked as such
  - Issues not necessarily relevant
    - E.g. PMP, USER mode
  - Oldest issue is four years old

Issue classification (proposal)

- Type
  - `bug` (RTL bugs only)
  - `task` (e.g. style, structure)
  - `question`
  - `improvement` (e.g. doc)
  - `new feature` (requests)
- Resolution
  - `open`
  - `resolved` (awaiting closure)
  - `closed`
- Labels
  - FPU
  - PMP
  - IRQ
  - TESTBENCH
  - DOCUMENTATION
  - WAIVED_CV32E40P
  - etc.
- Issue can have multiple labels
- Waived tickets remain open

Github issue work flow (proposal)

Phase	Task	Who	What	
Opening	Open	Everybody	• One topic per issue • Provide steps to reproduce (git hash, command line, test case, waves, etc.) • Mark as bug, task, question, improvement or new feature	Please file doc issues in core-v-docs Will provide issue template
	Classify	Committer	• Check issue type (fix if needed) • Add further labels (e.g. FPU) • Ask further info (if needed) • Move to core-v-docs, core-v-verif if needed	
	Prioritize	Davide/Arjan/Jérôme	• Prioritize depending on severity, etc.	
	Assign	Davide	• Add assignee (Member)	Please volunteer (or get volunteered ☺)
Resolving	Fix	Assignee (Member)	• Provide fix (passing at least 'smoke' regression) • Make/link associated core-v-docs PR • Make/link associated core-v-verif PR • Make cv32e40p pull request (PR)	
	Review	Davide	• Review and/or assign reviewer	Please volunteer for reviews
	Merge	Davide	• Possibly assign merge task to a Committer	
Closing	Close	Issuer or Davide	• Close issue (will close by Davide after 7 days)	

IRQ interface - CV32E40P – Today

Pin	Function	MIE* (MIP*)	Priority
irq_fast_i[47:16]		MIE1[31:0]	56-25
irq_fast_i[15:0]		MIE[31:16]	24-9
- (4)	Reserved: Future standard use	MIE[15:12]	Not yet specified by RISC-V
irq_external_i	MEI (Machine External Interrupt)	MIE[11]	8
- (1)	Reserved: Future standard use	MIE[10]	Not yet specified by RISC-V
- (1)	Reserved: SEI	MIE[9]	5
- (1)	Reserved: UEI	MIE[8]	2
irq_timer_i	MTI (Machine Timer Interrupt)	MIE[7]	6
- (1)	Reserved: Future standard use	MIE[6]	Not yet specified by RISC-V
- (1)	Reserved: STI	MIE[5]	3
- (1)	Reserved: UTI	MIE[4]	0
irq_software_i	MSI (Machine Software Interrupt)	MIE[3]	7
- (1)	Reserved: Future standard use	MIE[2]	Not yet specified by RISC-V
- (1)	Reserved: SSI	MIE[1]	4
- (1)	Reserved: USI	MIE[0]	1

Holes in vector table

Confusing priority scheme

Non-unified pin names

IRQ interface - Wish list

- Ability to be RISC-V ‘CLINT’ compatible
- Ability to use the 'holes in the CLINT vector table’
- Ability to use a priority scheme that is monotonous
  - i.e. not CLINT compatible
- Future proof with CLIC (ideally by not changing the pinout)
- Forward compatible with other CV* cores
  - CV32A, CV64A, CV32E* cores with USER mode

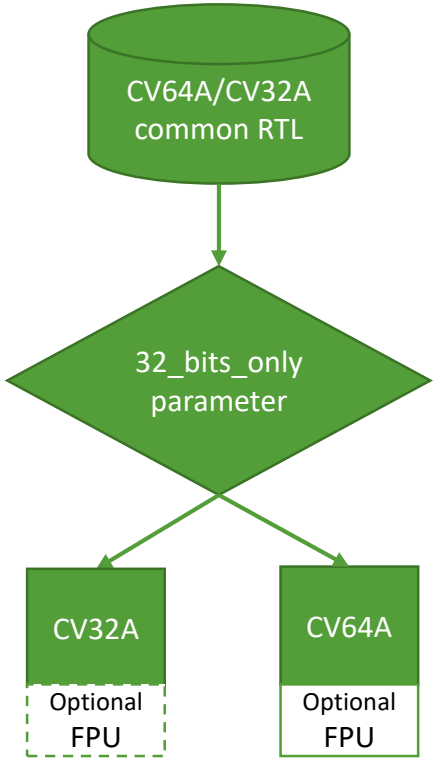
- Please join https://mattermost.openhwgroup.org/all-users/channels/twg--cores--interrupts if you want to contribute

Contributions

- Please let us know if you can contribute to
  - CV32E40P User Manual review
  - core-v-docs repos issue resolution
  - cv32e40p repos issue resolution
- Send email to
  - Arjan Bink arjan.bink@silabs.com
  - Jérôme Quevremont jerome.quevremont@thalesgroup.com
  - Davide Schiavone davide@openhwgroup.org

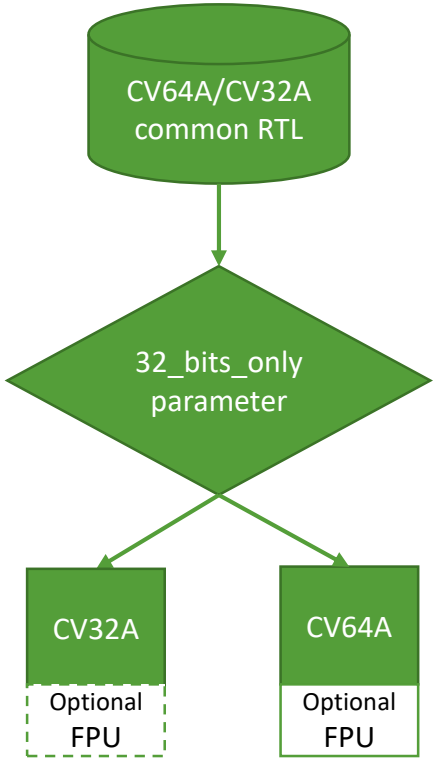
CV64A/CV32A

- CV32A is just a new flavor of ARIANE
  - It's a just a parameter added to CV64A source code to switch between the 64-bit and 32-bit ISA
  - Get 2 cores for (almost) the effort of 1
- CV64A/CV32A kick-off
  - June 4th: Rick clarified that the project can start @OpenHW
  - ARIANE repo to soon migrate to OpenHW
  - Thales ready to pull request "32 bits only" parameter
    - Floating-point support (RV32IF) support not included
  - Verification environment to set up
    - Maximize testbench commonalities with CV32E
    - "Sandbox" CV64A verification to prevent regression for CV32E
- Which OpenHW members are interested to join this effort?



CV64A/CV32A

- Thales interests:
  - INVIA (Thales DIS Design Services) (Jean-Roch, André, Zbigniew)
    - **CV32A** and **CV64A** for **ASIC** targets
    - Focus on security (e.g. avoid vulnerabilities) and verification
  - TRT (Thales Research & Technology) (Jérôme, Sébastien, Kevin)
    - **CV32A** optimized for **FPGA** targets, optional **FPU**
      - Avoid fork and branches as much as possible
    - Security and safety features (e.g. cache flush, cache lock…)
    - CV64A "overkill" for the application domains, but nice to have to be future-proof
  - Other SoC IPs in the future (interconnect, i/f ctrl…)
  - Share the CV64A/CV32A development and verification with OpenHW members



Upcoming

- Mechanics of working on two projects (Ariane, CV32E40P) in parallel inside OpenHW is currently under discussion
  - Discussing core-v-verif sandbox mechanics
  - Discussing common work between Ariane (CV64A/CV32A) and CV32E40P (e.g. compliance suite, CSR checkers, etc.)
  - Discussing on when to adopt Ariane by OpenHW
- More in next Cores TG meeting

Thank you!