CORES TG – November 9 2020

Arjan Bink

Jérôme Quevremont

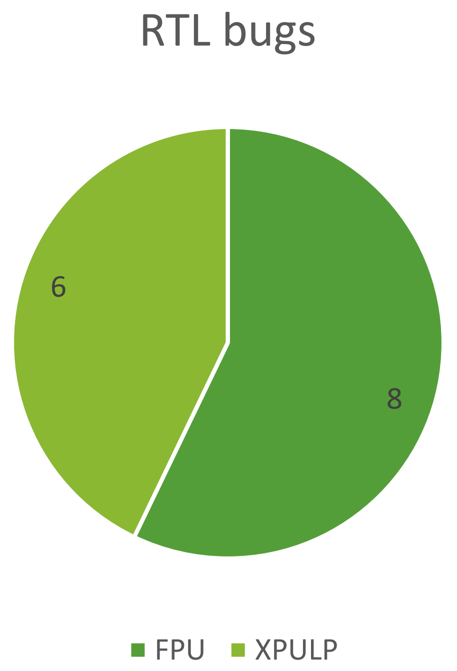
Davide Schiavone

Agenda

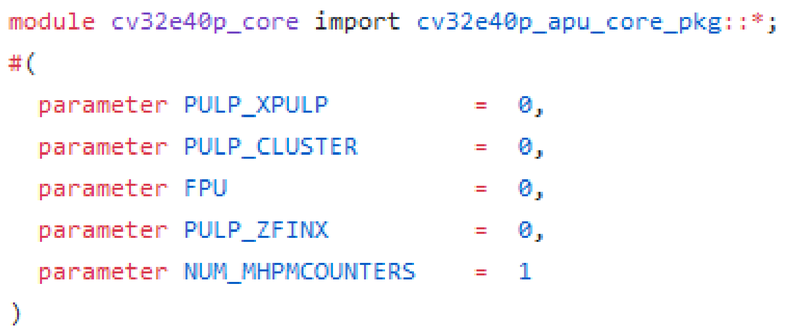
- CV32E40P status
- CORE-V versioning
- CV32E40P continuation (i.e. *mimpid* = 1) after RTL freeze
  - ‘v2’ proposal by John Martin (EM Micro)
  - ‘FPU=1’ proposal by Davide Schiavone (OpenHW)
- CV32E40X proposal by Arjan Bink (Silicon Laboratories)
- CV32A6

CV32E40P issues

https://github.com/openhwgroup/cv32e40p/issues filter	#	Comment
is:issue is:open	45	Open
is:issue is:open -label:status:resolved	44	Non-closed
is:issue is:open -label:status:resolved label:component:rtl label:type:bug -label:waived:cv32e40p	14	RTL bugs



- 0 bugs related to RTL freeze target configuration
  - PULP-only bugs are waived
  - FPU-only bugs are waived



```
module cv32e40p_core import cv32e40p_apu_core_pkg::*;
#(
  parameter PULP_XPULP          =  0,
  parameter PULP_CLUSTER        =  0,
  parameter FPU                 =  0,
  parameter PULP_ZFINX          =  0,
  parameter NUM_MHPMCOUNTERS    =  1
)
```

Allowed open CV32E40P issues at RTL freeze

- RTL freeze is gated by *open github issues* that represent *RTL bugs* for the *to be frozen configuration* which are *not explicitly waived*

Terminology	Github filter	Comment
'open github issues'	is:issue is:open	
'RTL bug'	label:Component:RTL label:Type:Bug	Excluding build flow, etc.
'frozen configuration'	-label:param:pulp_xpulp -label:param:fpu	Excluding PULP and FPU only bugs
'not explicitly waived'	-label:waived:cv32e40p	Excluding PMP, User mode, etc.

- Accepted open issues at RTL freeze will be documented, e.g.
  - Document in github and/or
  - Use of github project board
- Currently no gating issues

Versioning of CORE-V cores

Agreed feature set

- Agreed feature set at the start of a specific core project
- At a specific RTL freeze not all features might have been implemented or verified

- Example: CV32E40P has agreed features as M and C extensions as well as the PULP extension
- Example: At initial RTL freeze of the CV32E40P non-0 settings of PULP_XPULP, PULP_CLUSTER, and FPU parameters are not yet supported/verified, which does not mean that support for initially not supported parameter configurations can/will not get added after RTL freeze

What happens after RTL freeze when …?

- There is a bug fix
- There is a power optimization to the RTL
- There is a change in pinout
- Not-yet-supported (but initially agreed) features are implemented
- Not initially agreed software visible features are added
- The performance of the core is optimized impacting the cycle count of already frozen instructions
- …

Repos name, *marchid* CSR and *mimpid* CSR

- Repos name
  - E.g. cv32e40p, cv32e40
  - Used in System Verilog
    - Top module name is (lower case) <repos name>_core
    - File name of top module is (lower case) <repos name>_core.sv
  - Using upper case version of repos name in comments, documentation, etc.
  - Project name normally mirrors the repos name
- Machine Architecture ID (*marchid* CSR)
  - Governed by RISC-V organization, see https://github.com/riscv/riscv-isa-manual/blob/master/marchid.md
  - CV32E40P has *marchid* 4
  - CV32E40 *marchid* needs to be requested (but will not be 4)
  - *marchid* and project name are 'equivalent'
- Machine Implementation ID (*mimpid* CSR)
  - At initial RTL freeze CV32E40P will have *mimpid* 0
  - Each *mimpid* will correspond to a specific (series of) git tag(s)

Changes after RTL freeze

- Minor updates
  - Logically equivalent RTL
    - Comments
    - Power, performance, area improvements
  - Testbench, documentation updates
  - -> Original repos, *mimpid*, *marchid*
  - -> New git tag
- Bug fixes
  - -> Original repos, *marchid*
  - -> Incremented *mimpid*
  - -> New git tag
  - Not LEC equivalent only due to bug fix and *mimpid* increase
- Completion of parameter options
  - E.g. PULP_XPULP
  - -> Original repos, *marchid*
  - -> Incremented *mimpid*
  - -> New git tag
  - LEC equivalent (excluding *mimpid* update) for frozen parameter set
- Major/minor new features
  - E.g. bus error, User mode, PMP, P ext
  - -> New PPL, PL
    - A. Adapt existing project (e.g. CV32E40P)
      - -> Original repos, *marchid*
      - -> Incremented *mimpid*
      - -> New git tag
    - B. Start new project (e.g. CV32E41P)
      - -> New repos, new *marchid*
      - -> Reset *mimpid* to 0

Guidelines

- For the scenario where there are no bugs found with the default parameter settings, then all future feature updates shall result in the new design being “logical equivalent” with the original released version given the default parameter settings, except that *mimpid* can have a logical non-equivalence as this value can have been incremented.
- For the scenario where there are bugs found with the default parameter settings, then the new design updates shall only be logically non-equivalent with the original released version for these bugs and *mimpid* value update, and shall be logically equivalent for all other existing features of the original released version given the default parameter settings.

Examples – 1/2

- For example, if there are no bugs using the default parameter setting and if you find that an originally specified PULP instruction is not implemented or not implemented correctly, then this PULP instruction can be added or fixed and this instruction would not be accessible with the default parameter settings.
- Moreover, if for some reason the design requires a state machine modification or additional cycle insertions to handle this new instruction, then the new design shall still be logically equivalent with the original released version for the previously frozen parameter settings.

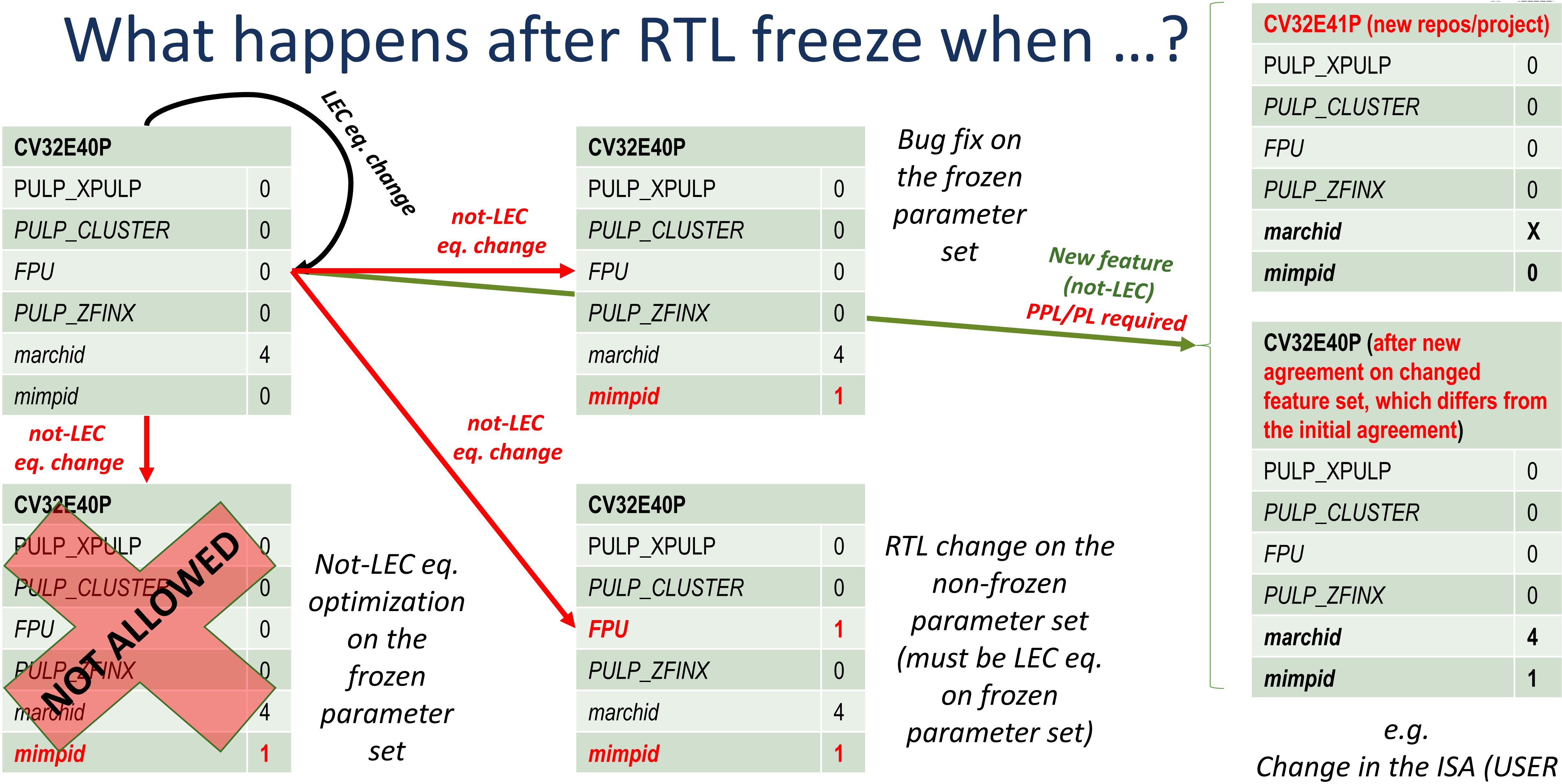
Examples – 2/2

- For example, if there is a request to replace the Xpulp SIMD instructions with the official RISC-V P extension in the CV32E40P.
- In this case this would be considered a new feature (which was originally agreed for the CV32E40) project. We would have to go through the PPL/PL milestones again. The proposal in the PPL/PL could be to change the definition of the CV32E40P (as opposed to starting a new repos), and if accepted, then the change can be done in the original CV32E40P project (same repos, same *marchid*, incremented *mimpid*).
- The CV32E40P will still remain LEC equivalent (except for the incremented *mimpid*) for the already frozen parameter settings.

What happens after RTL freeze when …?

There is a bug fix	Fix and increment *mimpid*
There is a power optimization to the RTL	Allowed if LEC equivalent for the frozen parameter set
There is a change in pinout	Can be discussed if only affecting non-frozen features
Not-yet-supported (but initially agreed) features are implemented	Add feature and increment *mimpid.* Frozen parameter set shall not be affected except for *mimpid*
Not initially agreed software visible features are added	New PPL/PL
The performance of the core is optimized impacting the cycle count of already frozen instructions	Not allowed as it will not be LEC equivalent for the frozen parameter set

What happens after RTL freeze when …?



CV32E40P	
PULP_XPULP	0
PULP_CLUSTER	0
FPU	0
PULP_ZFINX	0
marchid	4
mimpid	0

LEC eq. change

not-LEC eq. change

CV32E40P	
PULP_XPULP	0
PULP_CLUSTER	0
FPU	0
PULP_ZFINX	0
marchid	4
mimpid	**1**

Bug fix on the frozen parameter set

New feature (not-LEC)

PPL/PL required

CV32E41P (new repos/project)	
PULP_XPULP	0
PULP_CLUSTER	0
FPU	0
PULP_ZFINX	0
marchid	**X**
mimpid	**0**

CV32E40P (after new agreement on changed feature set, which differs from the initial agreement)	
PULP_XPULP	0
PULP_CLUSTER	0
FPU	0
PULP_ZFINX	0
marchid	**4**
mimpid	**1**

not-LEC eq. change

CV32E40P	
PULP_XPULP	0
PULP_CLUSTER	0
FPU	0
PULP_ZFINX	0
marchid	4
mimpid	**1**

NOT ALLOWED

Not-LEC eq. optimization on the frozen parameter set

not-LEC eq. change

CV32E40P	
PULP_XPULP	0
PULP_CLUSTER	0
FPU	**1**
PULP_ZFINX	0
marchid	4
mimpid	**1**

RTL change on the non-frozen parameter set (must be LEC eq. on frozen parameter set)

e.g.
Change in the ISA (USER Mode, PMP, RVB, RVP)
Performance optimization

CV32E40P (‘v2’) – Pre-PPL

John Martin (EM Micro)

CV32E40P (‘FPU=1’) – Pre-PPL

Davide Schiavone (OpenHW Group)

CV32E40P – FPU Extensions

- CV32E40P key features
  - RV32IM[F]C[Xpulp]
  - With FPU parameter enabled (modifications will be LEC with FPU=0, but not with FPU=1)
- The FPU instructions are sent to the APU interface (itf)
  - ***At the moment***, RVF are completely decoded in the core, operands are read in the F-RF inside the core.
    The decoded instruction is set to the APU, the core is blocked and waits for the FPU to finish its computation.
- Modifications in the APU behaviour as baseline for CV32E40X (see later)
- The interface needs to be slightly changed, i.e. adding the id is missing in the output and input port (destination register)
- Proposal to merging coming data into single signal apu_rdata and sent data into apu_address as in the OBI

- Proposal:
  - Only part of the decoding inside the core:
    - Recognizing it’s an FPU instruction and read integer operands (casts e.g.)
    - F-RF part of the FPU sub-sytem
    - The FPU sub-system receives:
      - Instruction 32b (now encoded one, i.e. 6b)
      - Up to 3 integers operands (for ZFINX in the future)
      - It decodes the instruction at finer grain level
      - It reads float operands
  - The APU runs independently
    - Not blocking pipeline
    - 1-b scoreboard, i.e. stalls only or data-hazards
    - Stall on structural contentions, i.e. to write back results in the integer register file port

CV32E40X – Pre-PPL

Arjan Bink (Silicon Laboratories)

CV32E40X – Extendable 4-stage CPU with SIMD

- CV32E40X key features
  - RV32I[A][P]MC[X]
  - 4-stage pipeline
  - M-mode only
  - PMP
  - CLINT
  - OBI
- No (internal added) custom instructions
- General purpose eXtension itf
  - Generic (applicable to ALU type instructions, loads/stores)
  - Tightly integrated (e.g. providing read and write access to register file, bypass signals, stall signals, etc.)
  - Low latency (instructions same latency as can be expected when adding the custom instruction directly into the core)
- Simplified pipeline and controller
  - Limit out-of-order completion
    - WB stage for ALU/MUL/DIV
  - Debug FSM simplification
- Area and power optimizations

- Lower worst case interrupt latency
  - Interruptable *div/divu/rem/remu*
  - Improve by ~32 cycles
- Bus error support
- Extended Debug Trigger
  - Multiple breakpoints,
  - Data interface related breakpoints
  - Support for *etrigger*
  - Optional exception instead of halt for triggers
  - Considering Debug spec 0.14
- Other improvements
  - *fence.i* interface
  - *mtval* implementation
  - Faster divide
- Realistic dependencies and scope
  - RVFI interface for formal verification and clean interface for ISS compare
  - Only (near-)ratified spec dependencies
  - Not included
    - User mode, CLIC, F, B

Benefits of X interface

- Does not claim opcodes for non-used functionality
- Does not spend area on non-used functionality
- Enables adding custom extensions without releasing the extensions themselves
- Limits verification and documentation effort to the X interface itself as opposed to the custom instructions
  - Instead burden is put on the user interested in such custom instructions
- Enables a business/support model for tool vendors (e.g. compiler, ISS, assertions, etc.) to provide support for added custom extensions

CV32E40X – Interested in participating?

- Are you interested in contributing to the CV32E40X?
  - Please contact me at arjan.bink@silabs.com

CVA6

CVA6

- PPL on 2020-09-28: before going to PL need:
  - Attract more participants (especially on verification)
  - Draft documentation and Vplan to size verification effort
  - Some tasks will move to LLVM and core-v-verif projects
    - Better to spin off small focused projects
- Thales progress:
  - CV32A6 committed to https://github.com/openhwgroup/cva6
    - MMU and FPU yet to come
  - Thales testbench committed to https://github.com/openhwgroup/core-v-verif/tree/master/cva6
    - Not OpenHW "official" core-v-verif testbench
  - CVA6 FPGA prototype and BSP up and running
  - Launched French student contest to optimize CV32A6 FPGA
    - https://github.com/thalesgroup/cva6-softcore-contest
  - 3 open positions in France and India
  - Intern working on WT caches (basic safe&secure features)

Thank you!