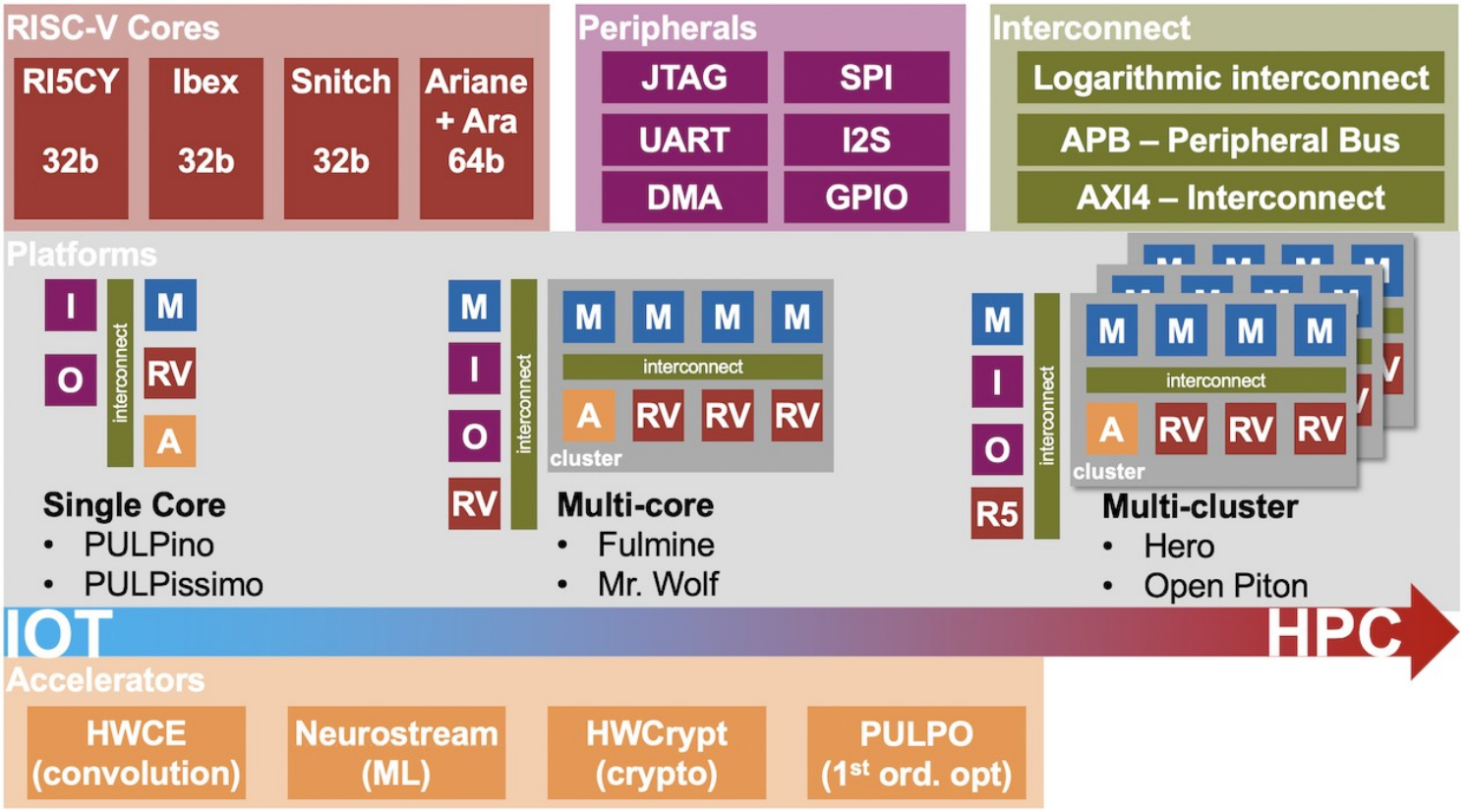
Bringing up the OpenHW Group RISC-V tool chains

Jeremy Bennett

About Open Hardware Group

- Not-for-profit member driven RISC-V collaboration
  - global: industry, academic and individuals
- Goal is high quality, open source hardware development
  - collaborative and open development model
- Cores are developed as the ***CORE-V*** family
  - smallest RV32 to largest RV64 designs
  - standard RISC-V with custom ISA extensions

Open Hardware Group Ancestry: PULP



Open Hardware Group Members

OPEN-HW

Members

Partners

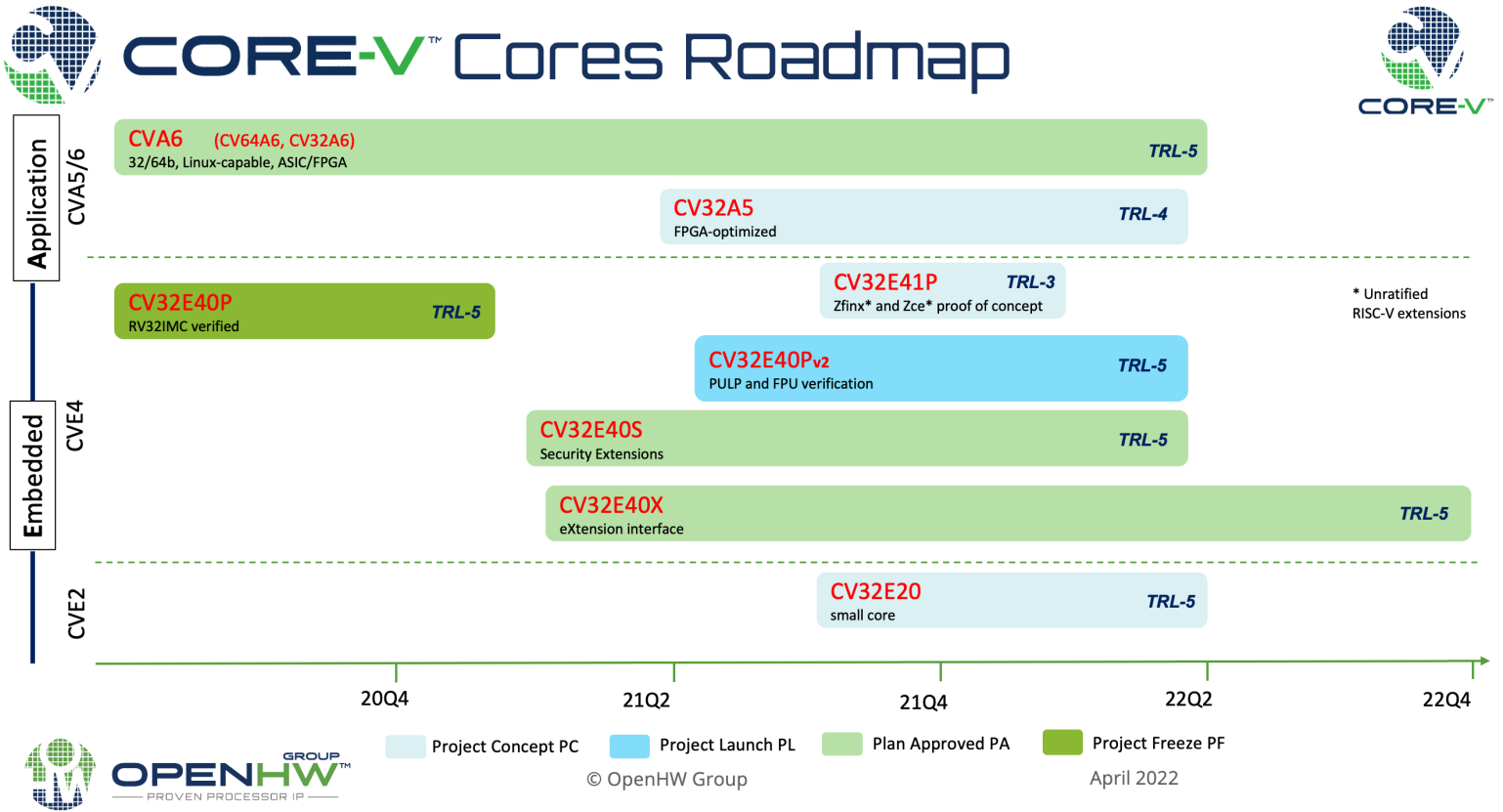
CORE-V

EMBECOSM®

OpenHW Engineering Organization

- **Technical Working Group** (TWG): Jérôme Quevremont, David Lynch
  - top level oversight
  - **Cores TG**: Arjan Bink, Jérôme Quevremont
    - oversees development of cores
  - **Verification TG**: Simon Davidmann, Jean-Roch Coulon
    - oversees verification of cores
  - **Hardware TG**: Hugh Pollitt-Smith, Tim Saxe
    - responsible for reference SoC implementations
  - **Software TG**: Jeremy Bennett, Yunhai Shang
    - responsible for all software projects

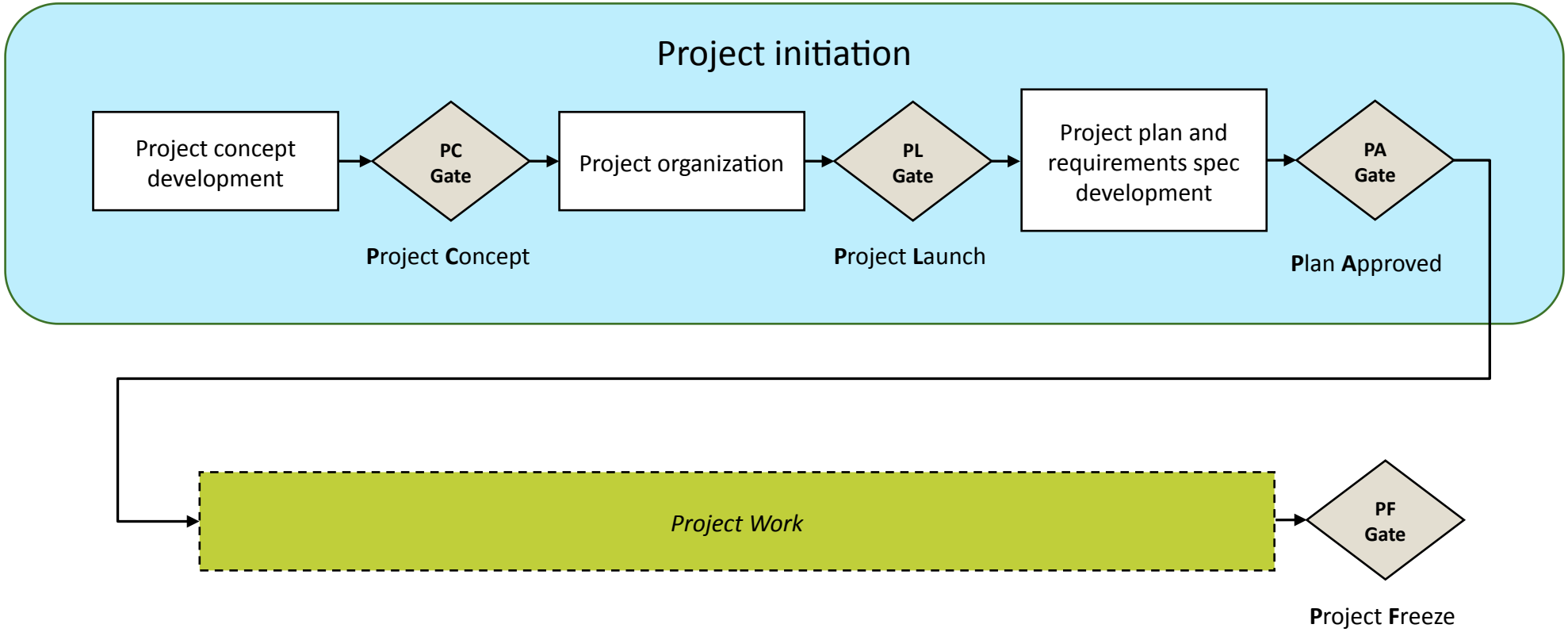
CORE-V Roadmap



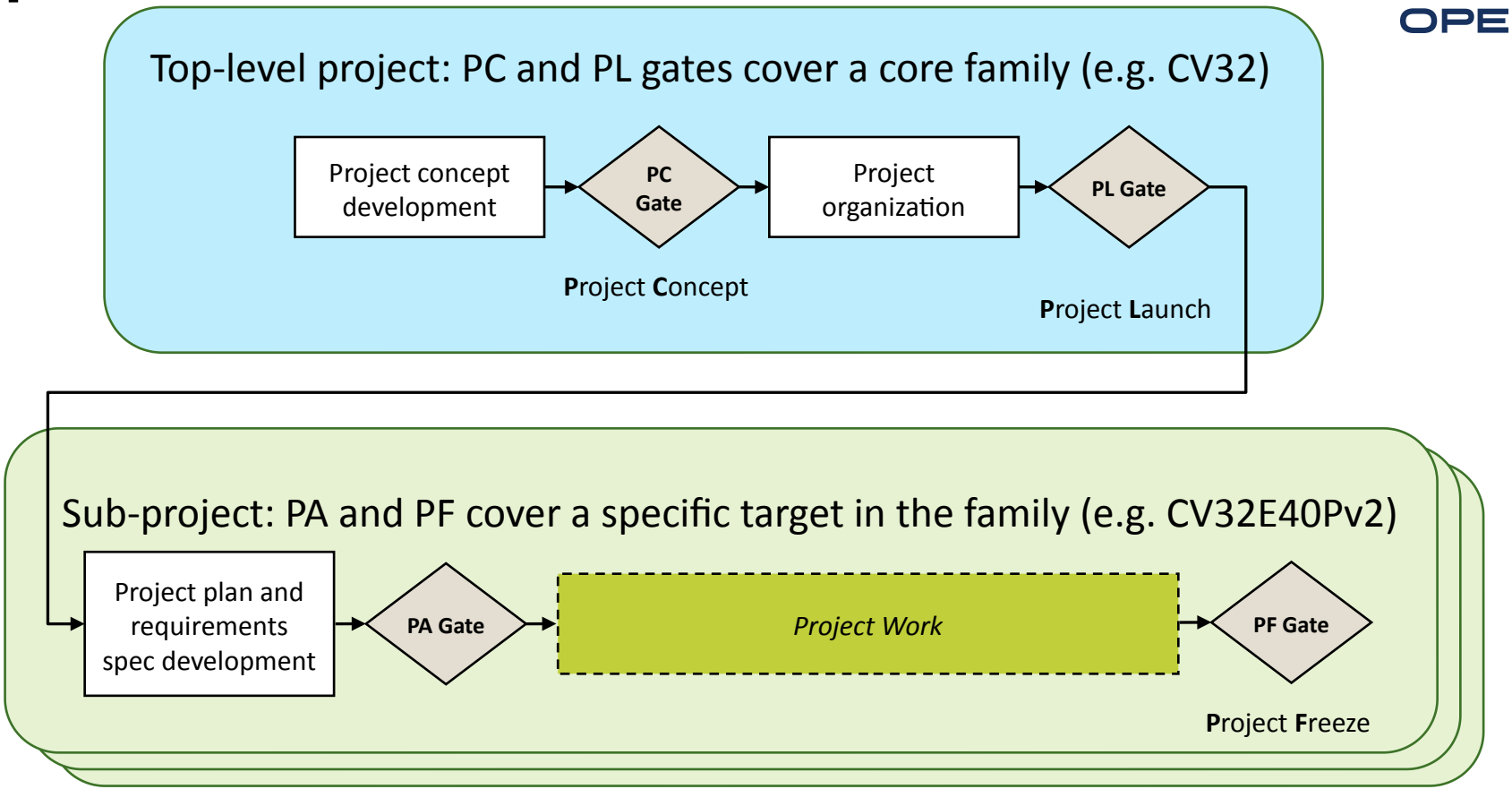
Software TG Projects

- Discussed today
  - LLVM project
  - GNU tools
  - QEMU
  - Verilator model
- Other
  - SDK
  - Hardware abstraction layer
  - FreeRTOS
  - Linux

OpenHW Process Gates



OpenHW Process Gates (SW)



What is a Compiler Tool Chain?

Ada, C/C++ family
Fortran, Java, Go
Rust, Modula 2
OpenMP/OpenACC

Source code → GCC → GAS → Object code

compiler libraries
libgcc
libstdc++v3

Newlib
Glibc
Musl
libc/libm

binutils
objdump

CGEN
disassemble
target sim
GDB

GCC:	**6.2 MLOC**
Binutils/GDB:	**3.9 MLOC**
Newlib	**0.9 MLOC**
Glibc	**1.3 MLOC**

What is a Compiler Tool Chain?

Ada, C/C++
Fortran, Rust, Swift
OpenMP/OpenACC

Source code → Clang/ LLVM → LLVM int. asm. → Object code

compiler libraries
CompilerRT
libc++

Newlib
Glibc
Musl
libc/libm

binutils
objdump

TableGen → disassemble → LLDB
TableGen → target sim

LLVM project: **6.5 MLOC**
Newlib **0.9 MLOC**
Glibc **1.3 MLOC**

CORE-V ISA Extensions

- Post-incrementing load/store (25): `-march=rv32i*_xmem`
- Hardware loops (6): `-march=rv32i*_xhwlp`
- General ALU operations (31): `-march=rv32i*_xalu`
- Immediate branching operations (2): `-march=rv32i*_xbi`
- Multiply-accumulate (22): `-march=rv32i*_xmac`
- Event Load (1): `-march=rv32i*_xelw`
- PULP Bit manipulation (16): `-march=rv32i*_xbitmanip`
- PULP SIMD (220): `-march=rv32i*_xsimd`
- Zc* 0.7.5 (36): `-march=rv32i*_zc*`

PULP extensions

CORE-V Builtin Functions

- Total around 300 functions defined
- Naming convention
  - **`__builtin_riscv_cv_`***isaext_name*
  - except where map to standard names (e.g. **`__builtin_abs`**)
- Support 32-bit and 64-bit versions with same name
- Not always 1:1 mapping to assembler instructions
- See core-v-sw/specifications/corev-builtin-spec.md on GitHub
  - 57 pages!

Testing

- Need a target with all the ISA extensions supported
  - compile time testing can be done without
- QEMU for CORE-V
  - project led by Weiwei Li at PLCT, Beijing
  - work in progress, due later in 2023
- Verilator model of specific cores
  - needs a debug server interface
  - for CV32E40Pv2 work in progress, due Q1/2023

Testing Policy

- LLVM project uses *lit* and GNU regression test (subset)
- GNU tools project uses GNU regression tests
- Exhaustive positive and negative testing by gas
- Vendor specific GNU ld testing
- Compilation only tests of builtins
  - scan for assembler instructions
- Execution tests of inline assembler and builtins

Key Issues

- Resourcing
  - thanks to **Embecosm**, **PLCT**, **Silicon Labs** and **Dolphin Design**
  - but needs more
- Upstreaming as vendor specific versions
  - OpenHW Group will not maintain forks long term
  - `riscv32-corev-elf-gcc`, `risv32-corev-elf-clang` etc.
  - need PSABI SIG to agree vendor specific relocations
  - ISA extension versioning (especially in gas)

Get Involved

- Repositories (all in GitHub openhwgroup org)
  - corev-llvm-project
  - corev-binutils-gdb
  - corev-gcc
  - embdebug-target-core-v

Get Involved

- Project leads
  - LLVM project: Charlie Keaney (overall) and Chunyu Liao (CV32E40Pv2)
  - GNU tools: Nandni Jamnadas
  - QEMU: Weiwei Li
  - Verilator model and debug server: Jeremy Bennett
- Weekly 30 minute engineering meetings
  - LLVM project: 08:30 UTC every Friday
  - GNU tools: 09:00 UTC every Friday

Thank You

jeremy.bennett@embecosm.com

Jeremy Bennett

embecosm.com

www.openhwgroup.org