Bringing up the OpenHW Group RISC-V tool chains
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

RISC-V Cores
- RISC5CY: 32b
- Ibex: 32b
- Snitch: 32b
- Ariane + Ara: 64b

Peripherals
- JTAG
- SPI
- UART
- I2S
- DMA
- GPIO

Interconnect
- Logarithmic interconnect
- APB – Peripheral Bus
- AXI4 – Interconnect

Platforms
- Single Core
  - PULPino
  - PULPissimo
- Multi-core
  - Fulmine
  - Mr. Wolf
- Multi-cluster
  - Hero
  - Open Piton

IOT
- Accelerators
  - HWCE (convolution)
  - Neurostream (ML)
  - HWCrypt (crypto)
  - PULPO (1st ord. opt.)

HPC
Open Hardware Group Members
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

**CORE-V Cores Roadmap**

- **CVA6** (CV64A6, CV32A6)
  - 32/64b, Linux-capable, ASIC/FPGA
  - TRL-5

- **CV32A5**
  - FPGA-optimized
  - TRL-4

- **CV32E40P**
  - RV32IMC verified
  - TRL-5

- **CV32E41P**
  - 2linx* and Zee* proof of concept
  - TRL-3

- **CV32E40Pv2**
  - PULP and FPU verification
  - TRL-5

- **CV32E40S**
  - Security Extensions
  - TRL-5

- **CV32E40X**
  - eXtension interface
  - TRL-5

- **CV32E20**
  - small core
  - TRL-5

Timeline:
- 20Q4
- 21Q1
- 21Q2
- 21Q4
- 22Q1
- 22Q2
- 22Q4

* Unratified RISC-V extensions

© OpenHW Group
April 2022
Software TG Projects

- Discussed today
  - LLVM project
  - GNU tools
  - QEMU
  - Verilator model

- Other
  - SDK
  - Hardware abstraction layer
  - FreeRTOS
  - Linux
OpenHW Process Gates

Project initiation

1. Project concept development
2. PC Gate → Project organization
3. PL Gate → Project plan and requirements spec development
4. PA Gate → Plan Approved

Project Work

PF Gate → Project Freeze
OpenHW Process Gates (SW)

Top-level project: PC and PL gates cover a core family (e.g. CV32)

- Project concept development
- PC Gate
- Project organization
- PL Gate

Project Concept

Project Launch

Sub-project: PA and PF cover a specific target in the family (e.g. CV32E40Pv2)

- Project plan and requirements spec development
- PA Gate
- Project Work
- PF Gate

Project Freeze
What is a Compiler Tool Chain?

Source code → GCC → GAS → Object code

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

Compiler libraries (libgcc, libstdc++v3)

libc/libm

CGEN

binutils

objdump → disassemble → target sim

GDB
What is a Compiler Tool Chain?

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

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

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

TableGen → objdump → binutils

compiler libraries
CompilerRT libc++
libc/libm

LLDB

OpenMP/OpenACC

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

TableGen

objdump

disassemble

target sim

LLVM project: 6.5 MLOC
Newlib: 0.9 MLOC
Glibc: 1.3 MLOC
CORE-V ISA Extensions

- Post-incrementing load/store (25): \(-\text{march}=\text{rv32i}_*\_xmem\)
- Hardware loops (6): \(-\text{march}=\text{rv32i}_*\_xhwlp\)
- General ALU operations (31): \(-\text{march}=\text{rv32i}_*\_xalu\)
- Immediate branching operations (2): \(-\text{march}=\text{rv32i}_*\_xbi\)
- Multiply-accumulate (22): \(-\text{march}=\text{rv32i}_*\_xmac\)
- Event Load (1): \(-\text{march}=\text{rv32i}_*\_xelw\)
- PULP Bit manipulation (16): \(-\text{march}=\text{rv32i}_*\_xbitmanip\)
- PULP SIMD (220): \(-\text{march}=\text{rv32i}_*\_xsimd\)
- Zc* 0.7.5 (36): \(-\text{march}=\text{rv32i}_*\_zc*\)
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-built-in-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, riscv32-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

embecosm.com
www.openhwgroup.org