CORE-V™

OPENHW™

OpenHW Group

软件任务组工作内容及进展介绍

吴伟

wuwei2016@iscas.ac.cn

www.openhwgroup.org

ASIA WG OPENHW™ PROVEN PROCESSOR IP

自我介绍

- 吴伟 | Wei Wu
  - wuwei2016@iscas.ac.cn | lazyparser@gmail.com
  - github: lazyparser
- OpenHW亚洲工作组成员（负责软件及营销）
- PLCT Lab 创始人、总监
- RISC-V 国际基金会中国区联络人、RISC-V大使

组织架构：工作组和任务组的划分
Working Groups & Task Groups

- 理事会（Board）任命工作组的筹备主席、负责最终批准工作组提交的议案。
  - 技术委员会和营销委员会都是常设委员会
- 技术工作组 / Technical Working Group
  - 处理器核任务组 / Cores Task Group
  - 验证任务组 / Verification Task Group
  - 软件任务组 / SW Task Group <- 本次演讲介绍
  - 硬件任务组 / HW Task Group
- 营销工作组 / Marketing Working Group
  - University Outreach Task Group

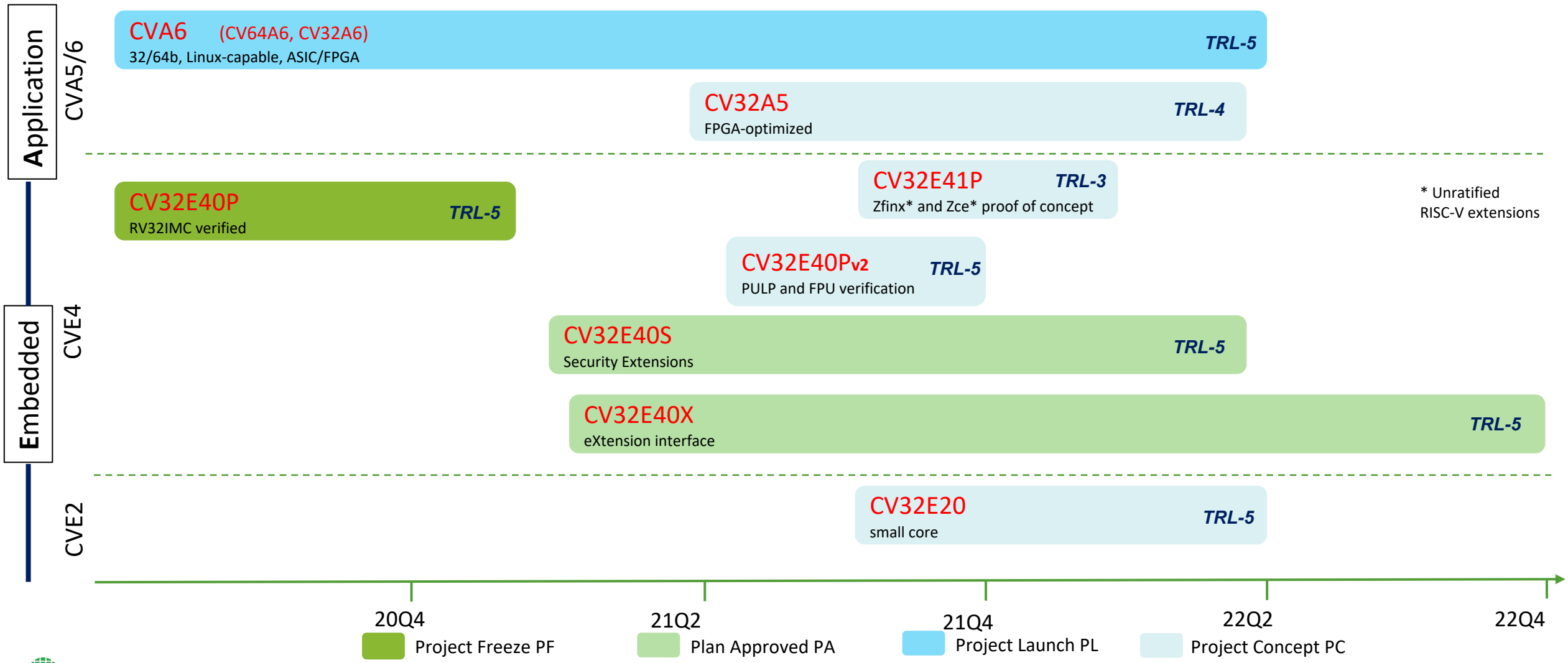
软件任务组 / SW Task Group

- Chair: Jeremy Bennett, Embecosm
- Vice-Chair: 尚云海, 阿里巴巴平头哥

- 职责：为 OpenHW 开发的CPU内核和IP提供软件工具链、操作系统以及固件的定义、开发、和维护支持。
- 活跃开发维护的项目包含： GCC / LLVM、IDEs、 FreeRTOS、 HAL、 CORE-V MCU SDK 等.
- https://github.com/openhwgroup/core-v-sw
  - https://github.com/openhwgroup/core-v-sw/blob/master/charter.md

https://github.com/openhwgroup/core-v-cores/blob/master/README.md

CORE-V™ OpenHW的开源核大家族



ASIA WG OPENHW™ PROVEN PROCESSOR IP

content source & credit: https://github.com/openhwgroup/core-v-cores/blob/master/README.md

为什么需要维护一套特定的软件栈？

CORE-V Application Class, 5/6-Stage Cores

- CVA6 就是原先的 PULP Ariane core, 6段流水、单发射、非乱序，支持 RV32GC / RV64GC，实现了 M/S/U 模式，支持 Unix/Linux。可以灵活配置。
- CVA5 面向FPGA设计，支持RV32IMA。SystemVerilog实现，源自 Simon Fraser 大学的 Taiga 项目。

CORE-V Embedded Class, 4-Stage Cores：CVE4 是基于 PULP RI5CY 的衍生系列，32位4流水. 可灵活配置。

- CV32E40P：RV32IMFC[Xpulp], FPU 选配（F扩展），面向DSP扩展了指令：包括 hardware loops, SIMD extensions, bit manipulation and post-increment instructions。
- CV32E40X： RV32[I,E][M|Zmmul][A]Zca_Zcb_Zcmb_Zcmp_Zcmt[Zba_Zbb_Zbs|Zba_Zbb_Zbc_Zbs]ZicntrZihpmZicsrZifencei[X].
- CV32E40S RV32[I|E][M|Zmmul]Zca_Zcb_Zcmb_Zcmp_Zcmt[Zba_Zbb_Zbs|Zba_Zbb_Zbc_Zbs]ZicsrZifenceiXsecure …Machine mode and User mode, an enhanced PMP, as well as various anti-tampering features.
- CV32E41P RV32IM[F,Zfinx]C[Zce] and Xpulp custom

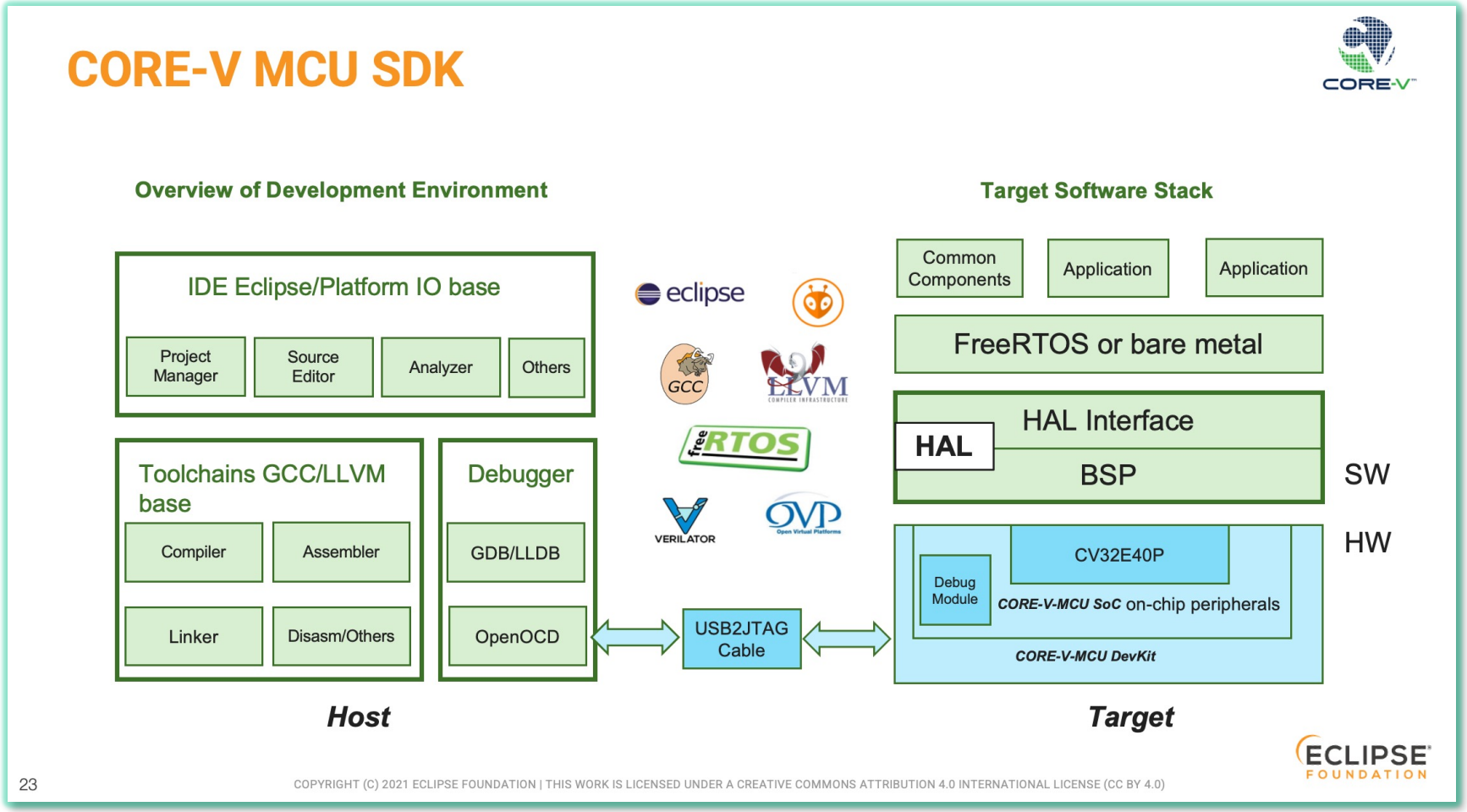
CORE-V Embedded Class, 2-Stage Core

- CVE2 低功耗32位2段流水顺序执行，RV32{E, I}[M]C。源自 lowRISC Ibex core

content source & credit: https://github.com/openhwgroup/core-v-cores/blob/master/README.md

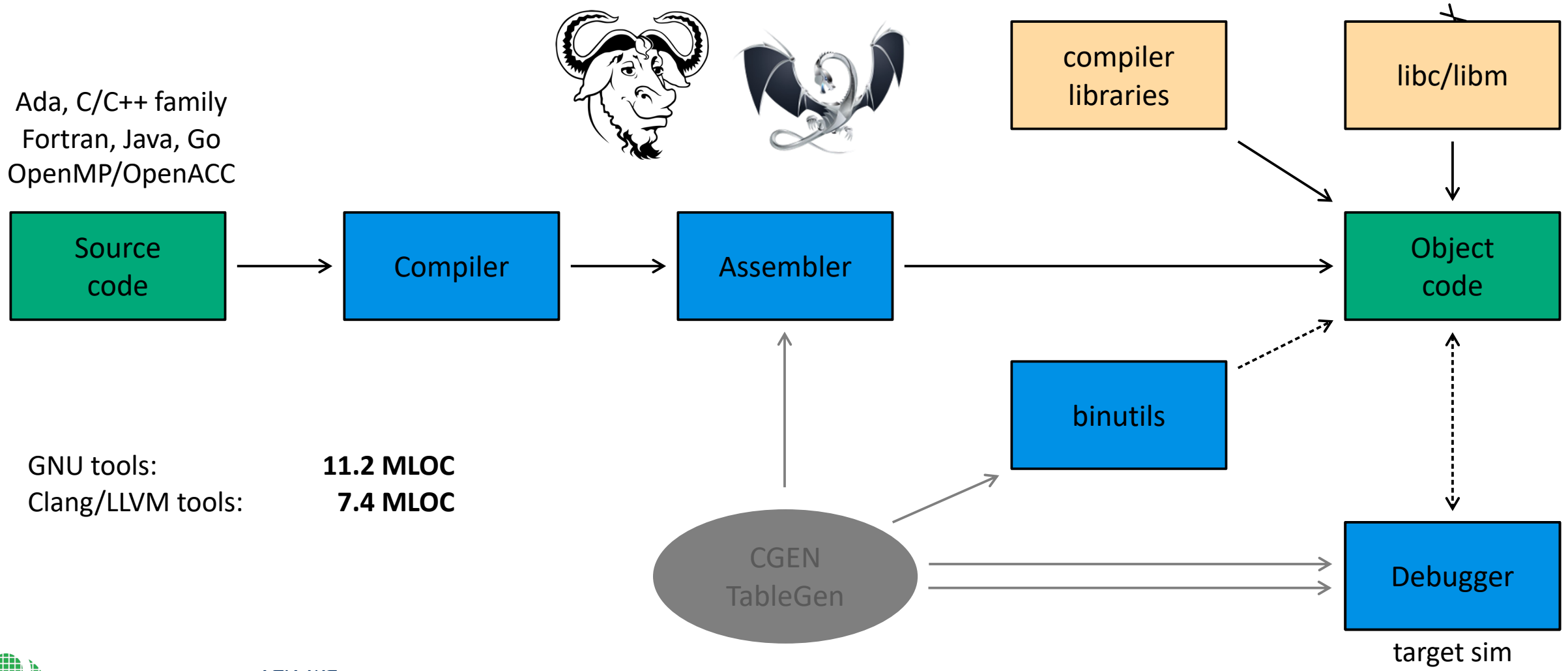
本次技术分享(简略地)介绍以下组成部分

- GCC
- Clang/LLVM
- FreeRTOS
- HAL
- IDE
- SDK



Source & Credit: Frédéric Desbiens et, al. : Towards a Comprehensive Open Source IoT RISC-V Stack

CORE-V工具链的组成简介 （GCC&LLVM）



Source & Credit: OpenHW TV S2 E04: Software Task Group Project Updates, GNU Toolchain part
https://www.openhwgroup.org/resources/openhwtv/s02e04/

CORE-V GNU GCC

支持的 CORE-V 的指令集扩展：

- hardware loops
- multiply-accumulate
- post-increment and register indexed load/store
- direct branches
- ALU extensions

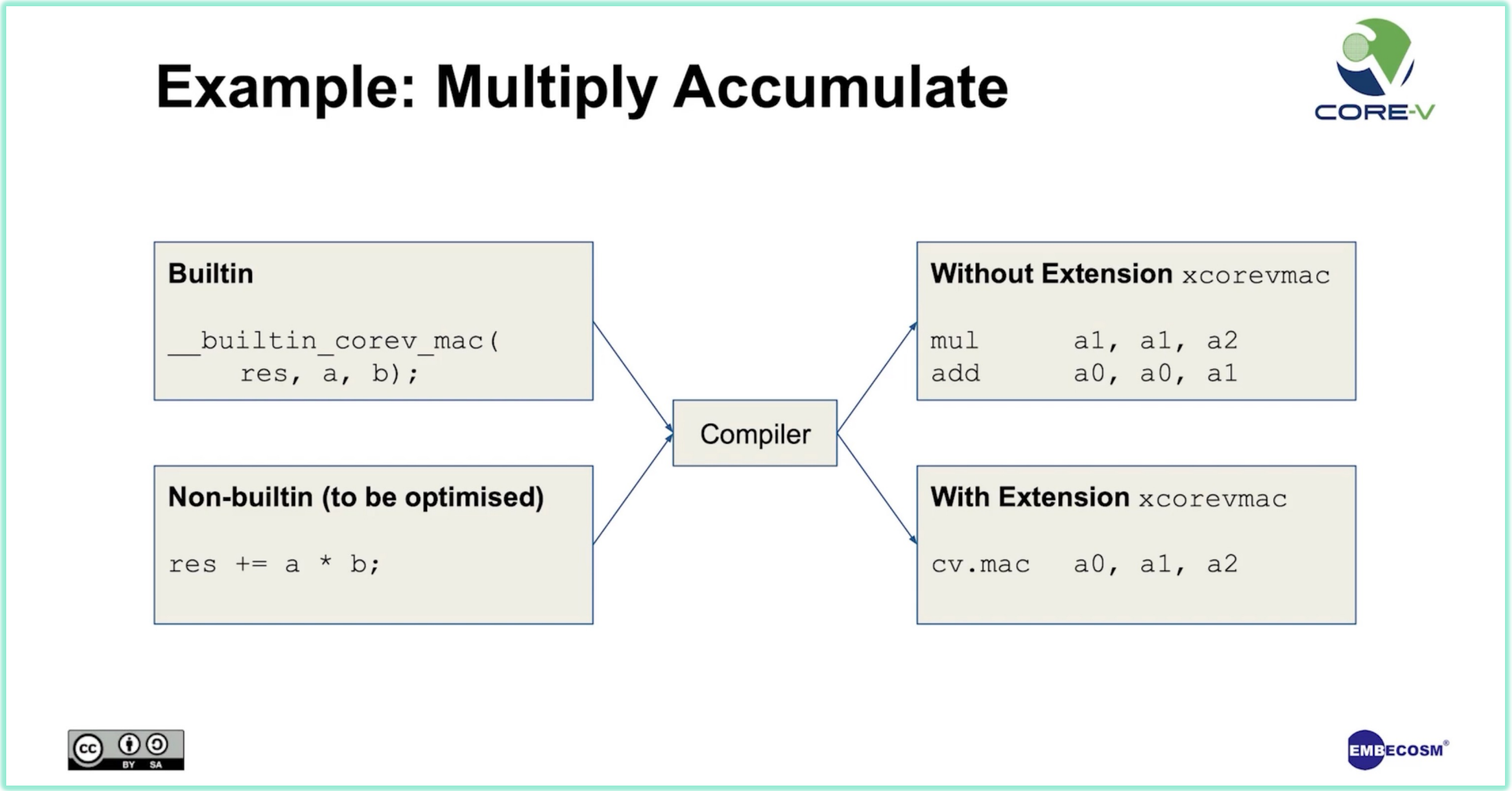
如何启用

- target riscv32-corev-elf
- additional -march architecture sepcificatiosn Xcorev and Xcorevyyy
- 扩展指令集一般以 cv. 作为前缀

  cv.starti, cv.mac, cv.abs etc.

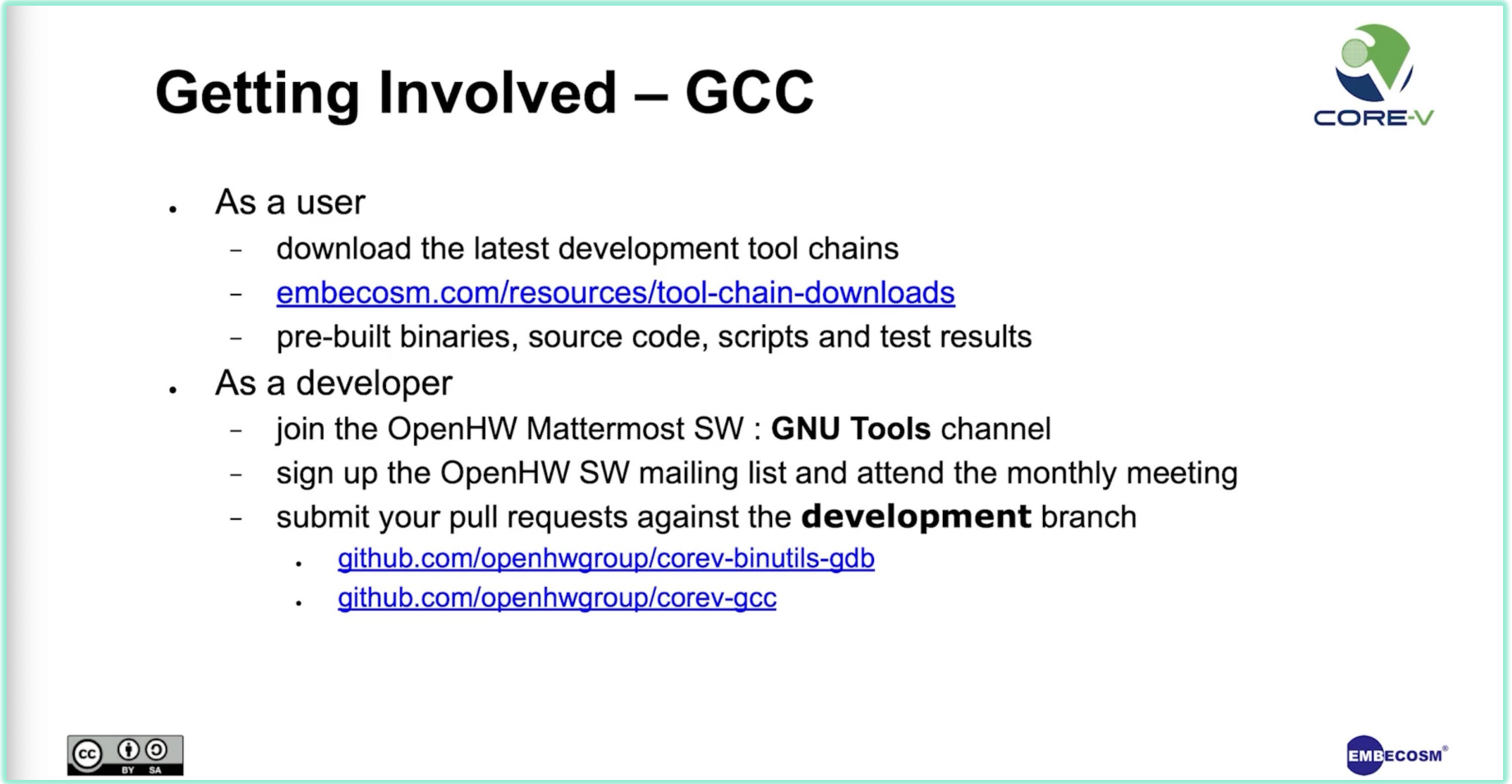
Source & Credit: OpenHW TV S2 E04: Software Task Group Project Updates, GNU Toolchain part
https://www.openhwgroup.org/resources/openhwtv/s02e04/

一个编译器优化的例子



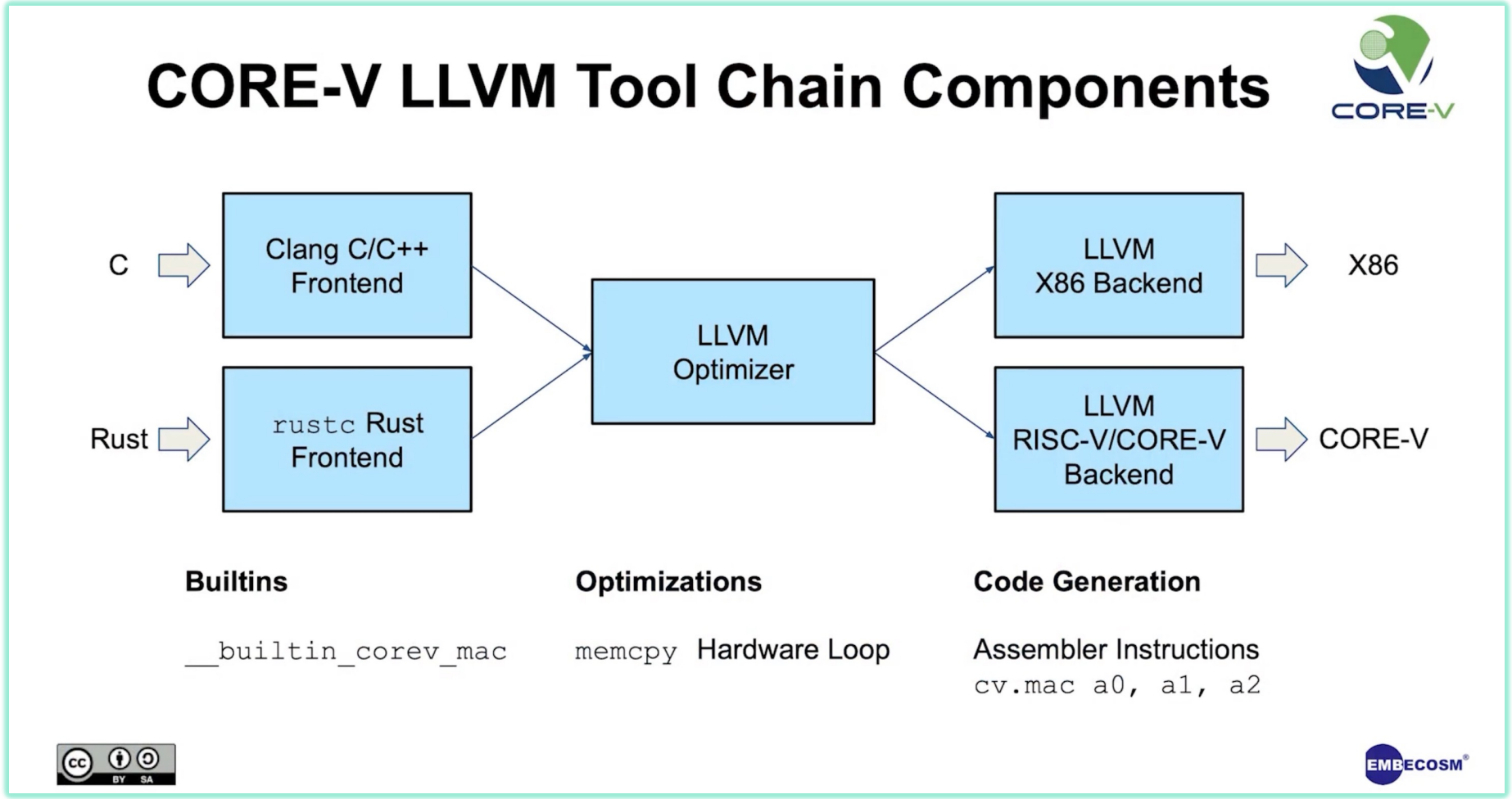
Source & Credit: OpenHW TV S2 E04: Software Task Group Project Updates, GNU Toolchain part
https://www.openhwgroup.org/resources/openhwtv/s02e04/

随时可以下载尝试，工具链都是开源的. 同时也欢迎一起开发完善CORE-V工具链（亚洲工作组的成员单位PLCT实验室已经是CORE-V工具链的贡献单位之一 ☺）



Source & Credit: OpenHW TV S2 E04: Software Task Group Project Updates, GNU Toolchain part
https://www.openhwgroup.org/resources/openhwtv/s02e04/

CORE-V LLVM 工具链：同样开箱即用



Source & Credit: OpenHW TV S2 E04: Software Task Group Project Updates, GNU Toolchain part
https://www.openhwgroup.org/resources/openhwtv/s02e04/

FreeRTOS

https://github.com/openhwgroup/core-v-freertos

CORE-V-MCU FREERTOS

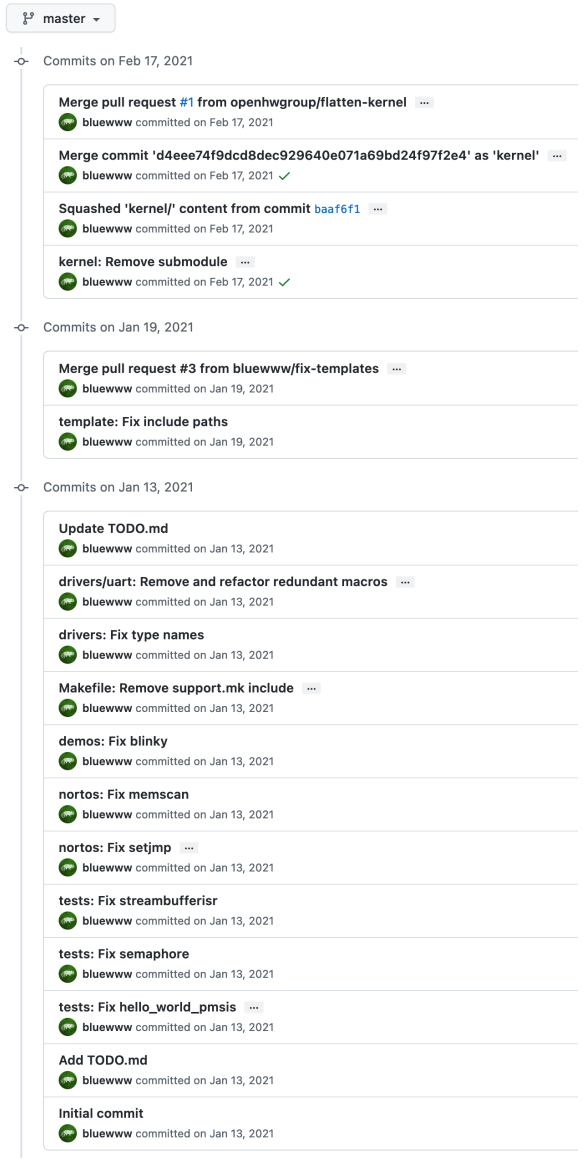
This project provides FreeRTOS and drivers for development of real-time applications on the core-v-mcu.

Programs can be run using RTL simulation (simulating the hardware design) or the virtual platform (called gvsoc, software emulation of the hardware design).

A book about FreeRTOs can be found here and the official documentation is available on this website.

Directory structure

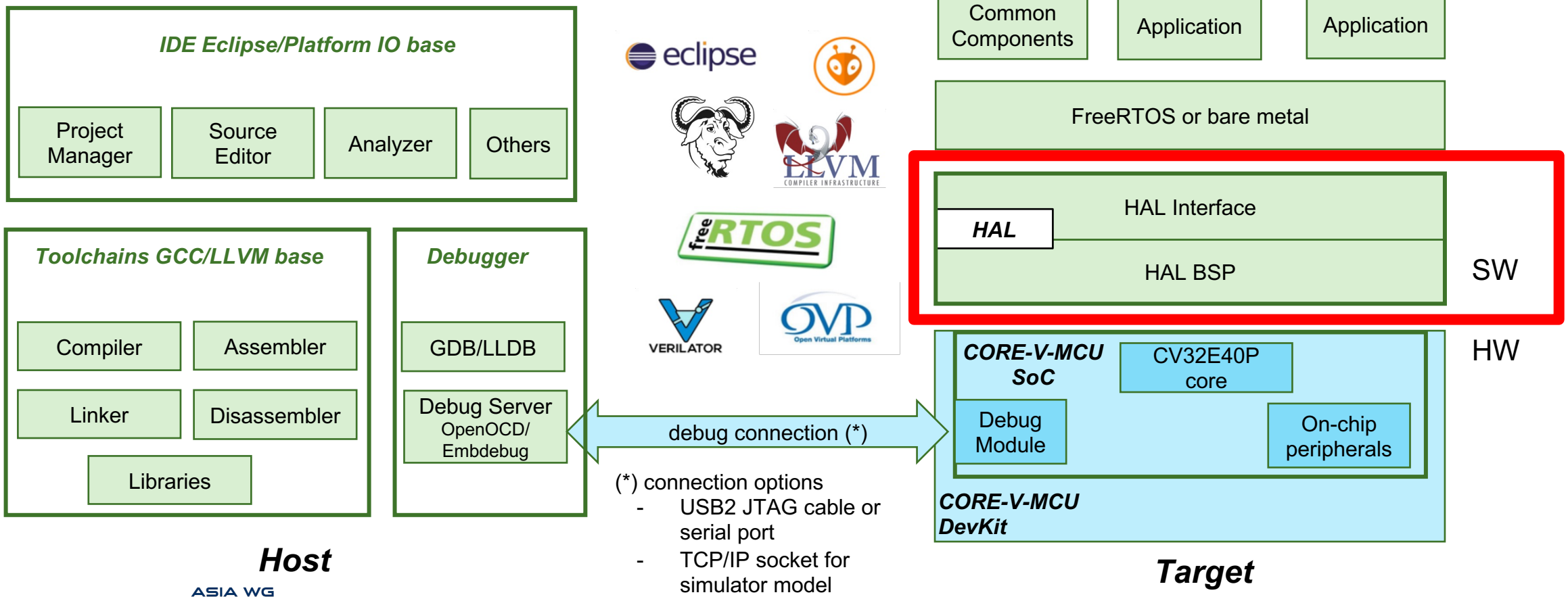
```
.
├── target             Platform specific code
├── drivers            Driver code and build system
├── CONTRIBUTING.md    How to contribute to this repository
├── demos              Classic FreeRTOS blinky demo
├── env                Sourcable configuration files to target the desired platform
├── kernel             FreeRTOs kernel code with PULP specific patches
├── nortos             Simple programs that don't need FreeRTOS
├── README.md          Read this
├── scripts            Various analysis scripts
├── template           Template projects to get started
└── tests              Advanced tests using FreeRTOS primitives
```



HAL 由OpenHW及AWG的成员的阿里巴巴平头哥提供开发支持

Abstraction Layer enabling SDK to be easily configured for each 32 bit core

Overview of Development Environment



Source & Credit: OpenHW TV S2 E04: Software Task Group Project Updates, GNU Toolchain part https://www.openhwgroup.org/resources/openhwtv/s02e04/

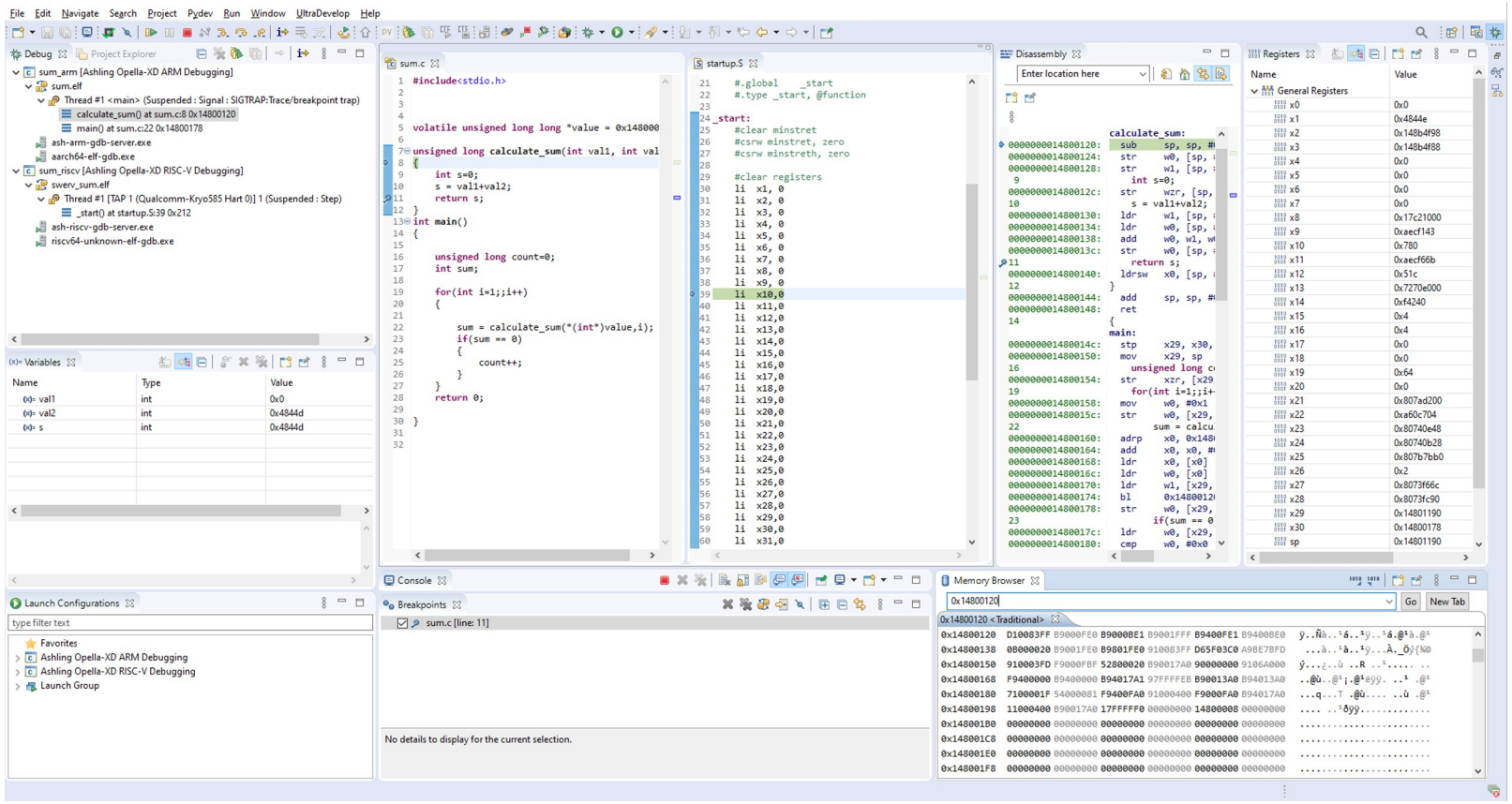
CORE-V™ IDE

ASHLING

EMBECOSM®

OPENHW™

- CORE-V IDE 是 OpenHW Group 软件工作组管理的开源开发项目
- 基于 Eclipse 和 PlatformIO 二次开发，用于 CORE-V 系列处理器核的开发
- 内置了 CORE-V 的 GCC 工具链
- 内置 OpenOCD 调试支持
- 内置 Digilent FPGA 板的“即用型”示例
- 文档：入门指南



ECLIPSE FOUNDATION

PlatformIO labs

ASIA WG OPENHW™ PROVEN PROCESSOR IP

Source & Credit: OpenHW TV S2 E04: Software Task Group Project Updates, GNU Toolchain part
https://www.openhwgroup.org/resources/openhwtv/s02e04/

SDK

https://github.com/openhwgroup/core-v-sdk/tree/main/SDK
https://docs.google.com/document/d/1pdm5ZwH6GKYAabQpQ8HwzvLy4g0O1JPzRHZVA8GHVc4/edit

SDK Components	Development Tasks	Contributors & Comments
IDE & Debugger	Eclipse based including CDT and Embedded CDT with OpenOCD debug support. Main tasks: • IDE titled **CORE-V™ IDE** • Include and integrate GCC Toolchain with CORE-V support and add IDE UI support for new CORE-V specific GCC switches (if any) ○ Include command-line build tools (make etc.) for Windows hosts • OpenOCD debug support for the DevKit, Nexys A7, Genesis 2 boards including on-board probes and standalone OpenOCD compliant probes (e.g. Ashling Opella-LD) with ready-to-run Debugger Launches • Include example projects links/details (see below) in IDE welcome page allowing easy import to Workspace ready to run • Debugger Register View support for CORE-V Peripheral and CSR Registers • Debugger FreeRTOS Task and Queue Views (nice-to-have for now) • Installer to create a **License** subdirectory containing all open-source licenses used (as text files) • Include QEMU simulator (and an example program with an IDE QEMU Simulator Debug Launch). Should we include another simulator (e.g. Verilator, Imperas/OVPSim or GVSoC) or at least support for a separately installed simulator ?. On-hold for now • Splash-screen and About graphics • Built-in links to SDK documentation via IDE Help menu and IDE Landing page • Release notes which are shown during the install process	Ashling. ArSysOp and others? Ashling proposes to base the IDE on their ***RiscFree*** IDE. It will be available free-of-charge and free-to-reuse with all component sources uploaded to the OpenHW github. CORE-V register definitions will be needed in SVD or IP-XACT format to populate Debugger Peripheral and CSR Register Views.

SDK

https://github.com/openhwgroup/core-v-sdk/tree/main/SDK
https://docs.google.com/document/d/1pdm5ZwH6GKYAabQpQ8HwzvLy4g0O1JPzRHZVA8GHVc4/edit

GNU GCC Toolchain	RISC-V GCC Toolchain. Main tasks: • CORE-V GCC compiler, binutils and GDB • OpenOCD (standard RISC-V version) • newlib support for CORE-V bare-metal • Semi-hosting support • UART in bare-metal newlib. Is this needed?	Embecosm.
FreeRTOS	FreeRTOS kernel and drivers. Main tasks: • Port to CORE-V and validate on DevKit board • Drivers & support for all DevKit board peripherals • Do we include FreeRTOS source in SDK or link to on-line repo? • FreeRTOS examples x2 as below	Who is the component leader ? Amazon, CMC, ETH and QuickLogic all have some involvement to date.
SymbiFlow	SymbiFlow FPGA toolchain. Main tasks: • Allow invocation from the IDE (or install standalone?) • Support for targeting on-board CORE-V FPGA (include any necessary ancillary files) and programming • Example as below section	Who is the component leader ? OpenFPGA group role to be clarified…maybe a partnership which will replace the need for any SDK work by OpenHW SDK WG? Contributors as per IDE with examples, integration and usage help provided by Quicklogic?
Documentation	User manual, IDE help, wiki, release notes and video: • Hosted on OpenHW site (https://docs.openhwgroup.org/) and linked to by IDE using **Read the Docs** • IDE Links to CORE-V and FreeRTOS online references • Use examples as the basis for getting-started videos	Who is the documentation leader ? Do component owners provide their own documentation ?

SDK

https://github.com/openhwgroup/core-v-sdk/tree/main/SDK
https://docs.google.com/document/d/1pdm5ZwH6GKYAabQpQ8HwzvLy4g0O1JPzRHZVA8GHVc4/edit

Examples	Include example projects links in the IDE welcome page allowing easy import of projects to Workspace ready to run. Projects include all sources, build files and debug launches ready-to-run for DevKit on-board debugger). Examples targeted at CORE-V and need to include: • Bare-metal DevKit blinking LED • Examples of driving all the devices/peripherals on the DevKit (bare metal) • Minimal FreeRTOS example • FreeRTOS example exercising all devices (with drivers) including the WiFi driver • SymbiFlow example • QEMU example program	TBC. Suggest: 1. Bare-metal blinking:Ashling. 2. Bare-metal devices/peripherals:QuickLogic 3. FreeRTOS (x2) examples:Amazon? 4. SymbiFlow:QuickLogic 5. QEMU: Ashling Look at other examples in the DevKit requirements…we cannot accommodate everything and maybe best to have a strong IDE "import example" feature from a OpenHW github examples page
SDK Installer	Single, installer binary executable which contains and installs ALL components ready to run. • OS support needed for Windows (10/11) and Linux (RH 7.9/8.4 & Ubuntu 18.04/20.04) • Upgrades will be done by new releases of the installer binaries which must be downloaded and installed by the end-use • Automated component updating is something that can be considered for future releases • macOS not covered for now	Ashling
ExpressLink	AWS IoT cloud connection to DevKit board over Wi-Fi interface. Details and contents to be clarified. Packaged as an example or more needed?	Amazon and others (e.g. Espressif) ?
SDK Deployment	OpenHW SDK download page with release notes	TBC

谢谢大家

更多技术细节，欢迎访问

https://github.com/openhwgroup