The Common RISC Platform

Desktop Edition

Book I: System Overview

Pacific Semiconductor

January 23, 2021

Contents

Licensing Terms

The Common RISC Platform is availed to users under the terms of the Common RISC Platform License (CRPL).

The full text of this license is contained in `license.txt`

Document Purpose

This document provides a brief overview of each component of the Common RISC Platform. More in-depth documentation is avaiable in Books II through VII.

Chapter 1

Introduction

1.1 Preface

Welcome to the Common RISC Platform documentation. Here you will find the full library of specifications that comprise the Platform. This project is a large undertaking, with the intent of producing a PC-like compute platform that promotes user freedom, openness and flexibility.

The PC platform has, for years, been threatened by challenges both from within the ecosystem and without. The relentless pace at which ARM-based platforms have increased in both performance and efficiency has led to many consumers abandoning their PC devices with devices like the M1 Macs and Microsoft Surface. This should be of great concern to all users, as vendors seem to be using this paradigm shift as an opportunity to do away with such silly notions as 'expandibility' and 'user freedom'. Most (if not all) PC-replacement platforms are non-free, locked down and inflexible. It is impossible to even so much as install a different operating system than the manufacturer standard on many of these devices.

The PC's greatest strength in the past has been its extensibility and openness. The almost-universal platform gave software developers a target that guaranteed their software would have the largest possible addressable market, and the open industry standards upon which the platform is built ensured competing hardware stood mostly on its own merits.

By the turn of the century however, this had begun to change. Chip vendors found themselves locked in legal battles, attempting to sue each other into oblivion and wrest control over the PC market. Intel stopped allowing other vendors to use their sockets, ending an era of almost universal cross-compatibility on the PC. This has given rise to the market we know today, where both remaining x86 vendors produce their own hardware platforms. This is a terrible outcome for the consumer. Vendors now, in a large way, dictate to users what hardware they can and cannot use with their machines. In Intel's case, users are not permitted to enable DDR4 XMP profiles on any chipset save for the high-end Z-series chipsets. This is not a technical limitation, as prior generations allowed this functionality on low end motherboards. This is merely an attempt by Intel to force users into paying for a higher margin product, often with unwanted or unneeded features, just to enable the single feature they actually *do* want.

The death of the PC should not mean the death of user freedom. There is a place in the market for a modern, PC-like platform with all the benefits of the latest advances in system architecture that still respects the user and gives them control over their computer. The Common RISC Platform aims to be exactly that.

By unifying existing open hardware standards, we aim to create a PC-like platform that offers a universal, free and open competitive field upon which products can stand on their engineering merits alone, and where users are afforded total control over the products they own.

1.2 Component Standards

The Common RISC Platform at its heart represents the unification of many existing de-facto industry standards. The rest of this document assumes the reader is at least conceptually familiar with:

- The RISC-V group of instruction set architectures
- Advanced Configuration and Power Interface
- PCI Express
- JEDEC DDR memory standards
- The ATX12VO power delivery standard
- NVM Express
- Coreboot
- eSPI
- I2C

An in-depth knowledge of these standards is not required to understand the platform at an architectural level.

1.3 Glossary of Terms

To be populated

1.4 Design Goals

The development of the Common RISC Platform is informed and guided by this set of common design goals.

- The user must, at all times, retain **full control** over their system
- The platform must make use of established free and open industry standards wherever possible
- The platform must be flexible and extensible to meet the computing needs of the mid-term future
- Anyone should be able to develop and produce a CRP-compliant system
- Any CRP system must be able to boot a fully vanilla Linux kernel with no proprietary firmware blobs either in the kernel or in the platform firmware
- All CRP hardware should be as interoperable and cross-compatible as possible
- The platform should be suitable as a replacement for the PC in all form factors
- The platform design should be as simple as humanly possible
- Software targeted at the platform must be able to run on all systems with no errors (i.e. a common ISA)

Chapter 2

Main System Package

2.1 Nomenclature

The Main System Package replaces the CPU in the Common RISC Platform. This change in nomenclature is driven by a belief that the term 'CPU' is no longer reflective of the device's role in the modern computer system. Modern CPU packages not only contain general purpose logic cores, but also fixed-function accelerators, I/O controllers and other non-core devices.

Additionally, given the rapid adoption of heterogeneous compute, the CPU is no longer 'central' to many compute tasks, which are often offloaded to external accelerator FPGAs and ASICs.

Hence, we believe the term 'Main System Package' represents a more accurate descriptor of the device. The MSP combines the main RISC-V logic cores and I/O controllers, without which the system cannot function, but we predict that it will become increasingly irrelevant in the actual processing of data as add-in accelerators gain popularity.

2.2 Logic

A Common RISC Platform compliant MSP **must** implement **at least one** RV64GC logic core. RV64GC is the set of RV64I, M, A, F, D, Zicsr, Zifencei and C RISC-V instruction sets. Microarchitectural decisions are at the discretion of the hardware vendor.

MSPs making use of microcode in any way must store this on local ROM. In the interests of transparency, vendors are encouraged to use EEPROM, open source their microcode, and allow users to flash their ROM.

The MSP must incorporate a JEDEC DDR5 memory controller for main system memory. Implementation details are left to the imagination of architects.

The MSP must expose to the system a PCI Express Root Complex of 40 lanes. 32 of these lanes are general purpose, and may be assigned at random by mainboard manufacturers for any PCIe device. 8 of these lanes are exclusively for communicating with the Platform Hub. All communication with off-package devices must be done through PCI Express. Board manufacturers are given the freedom (and responsibility) of interfacing other controllers of their choice with the MSP through the PH.

The MSP must also implement a full USB4 Host Controller. MSPs incorporating graphics output must use USB4 Alternate Modes (or direct tunnelling) for display output. Mainboard vendors are required to provide at least one USB4 Type-C connector on the rear I/O to facilitate display output.

Out-of-band signalling for the purposes of platform initialisation, etc. must be carried out via the eSPI bus.

2.3 Socket

MSPs will interface the rest of the system through a universal socket, CRP-1, yet to be fully described. All implementaions must conform to the electrical and physical specifications of CRP-1.

2.4 Power Delivery

The MSP shall accept two voltage rails from the bulk VRMs on the mainboard, V_{D0} and V_{D1}. In order to maintain full cross-compatibility, MSPs must implement FIVRs for any minor rails that cannot be provided by V_{D0} or V_{D1}. Both rails can be controlled using SVID.

The MSP shall not exceed 120W total power draw at stock settings.

Chapter 3

Mainboards

3.1 Overview

Mainboard vendors are given essentially a blank canvas with