RISC-V Platform Firmware Implementation with UEFI

Spike Yuan, Alibaba Damo Academy

Introduction

UEFI Platform Specification provides a mature definition on support for silicon and platform initialization and major external devices, while RISC-V architecture faces a lot of challenges to be compatible with these industry specifications, external devices and operating systems. So, adopting UEFI framework and its implementation is an expressway.

We successfully implement UEFI in a RISC-V laptop product, The RuyiBook (JiaChen Version) powered by the TH1520 platform, which boots to Linux using a simplified UEFI solution.

Methodology

Pre-PI Phase

Pre-PI phase is designed to prepare UEFI DXE environment and work as DXE-IPL. It uses the FDT to configure an UEFI style PEI environment, including MemoryInit-PeiLib and PlatPeiLib etc. Within these Libs, UEFI HOBs are produced to describe UEFI memory map, Firmware volunm and CPU policy etc. To invoke DXE IPL, in the end, it decompresses the DXE main FV and jumps to the DXE main dispatcher.

DXE Phase

In DXE phase, DXE still utilizes a FdtBusDxe driver to scan external devices. For PCIe devices emuneration, the implementation is simple as only discovering a bootable SD card. There is a set of platform specific DXE drivers for TH1520 specific devices initialization.

BDS Phase

The BDS phase is total UEFI style but there is no Boot Option as for embedded system design. Here uses UEFI shell script to do LinuxBoot with grub solution.

The kernel image within the SD card must contain the PE/COFF header, an UEFI compatible header, to be recognized and booted from. Bootargs configuration could be added into grub config file, too.
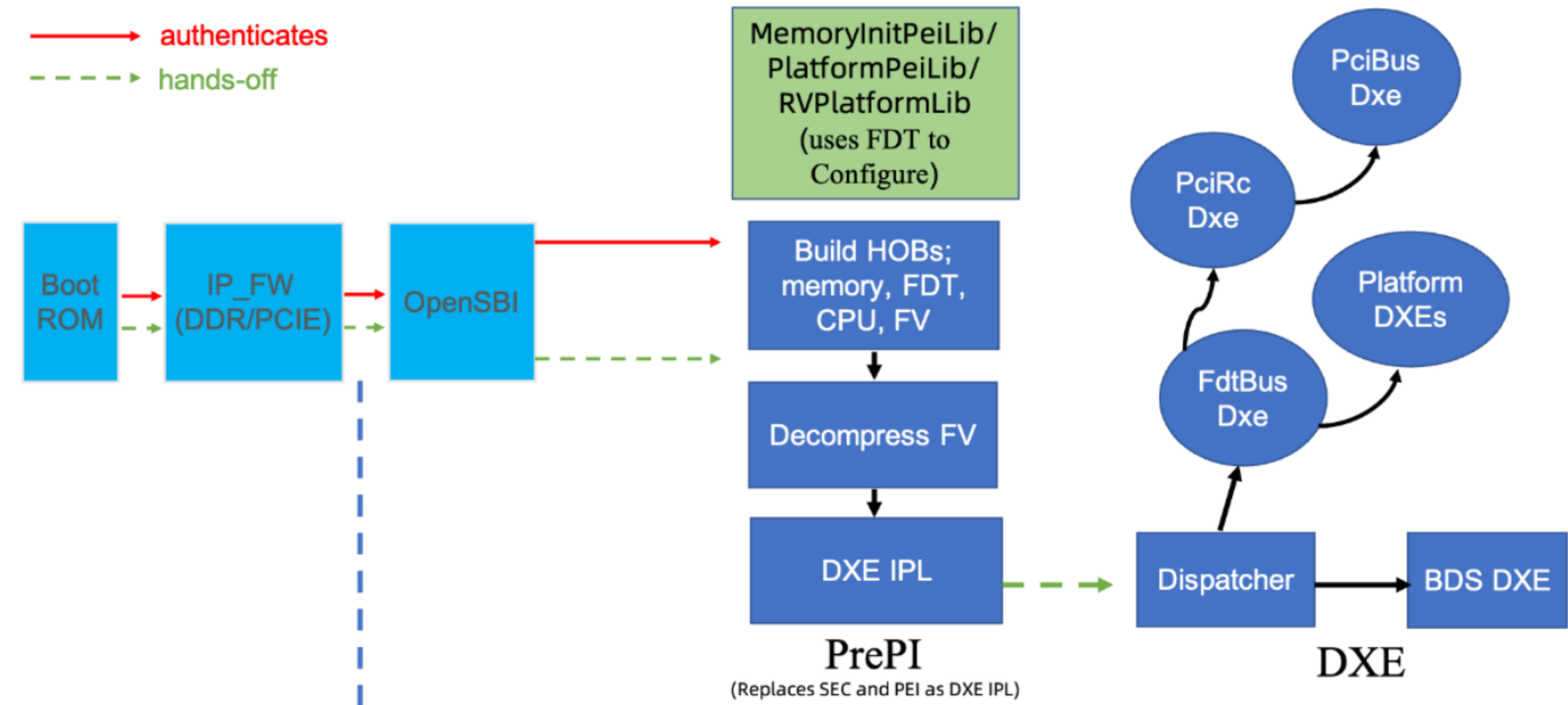
Collaboration with OS and Product

Achievements

- Applied the UEFI solution to LicheePi 4A / RuyiBook successfully
- Connected two communities(UEFI and OpenEuler) in TH1520
- Embedded product can harmoniously blend with UEFI firmware

UEFI boot to OpenEuler
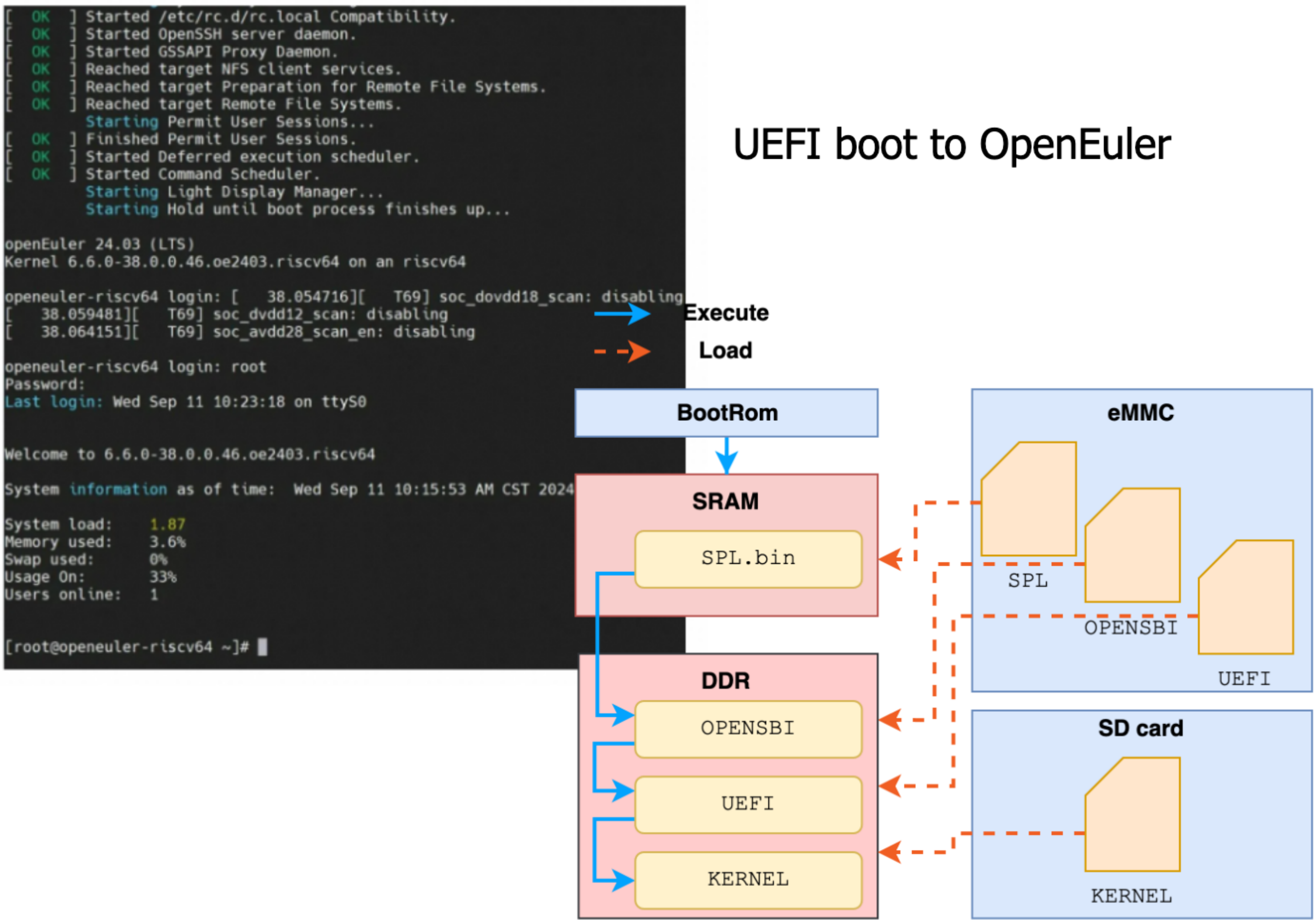
Future Work

- Simplify Complexity: Reduce firmware bloat and vendor-specific dependencies.
- Accelerate Innovation: Leverage Linux/open-source agility for cloud-native workloads.
- Unify Standards: Align RISC-V, x86, and Arm ecosystems under USF/UPL frameworks.
- Secure Market Leadership: Position RISC-V as a low-barrier, high-performance firmware solution for next-gen servers.