RISC-V

Overview

This document outlines the U-Boot boot process for the RISC-V architecture. RISC-V is an open-source instruction set architecture (ISA) based on the principles of reduced instruction set computing (RISC). It has been designed to be flexible and customizable, allowing it to be adapted to different use cases, from embedded systems to high performance servers.

Typical Boot Process

U-Boot can run in either M-mode or S-mode, depending on whether it runs before the initialization of the firmware providing SBI (Supervisor Binary Interface). The firmware is necessary in the RISC-V boot process as it serves as a SEE (Supervisor Execution Environment) to handle exceptions for the S-mode U-Boot or Operating System.

In between the boot phases, the hartid is passed through the a0 register, and the start address of the devicetree is passed through the a1 register.

As a reference, OpenSBI is an SBI implementation that can be used with U-Boot in different modes, see the OpenSBI firmware document for more details.

M-mode U-Boot

When running in M-mode U-Boot, it will load the payload image (e.g. fw_payload) which contains the firmware and the S-mode Operating System; in this case, you can use mkimage to package the payload image into an uImage format, and boot it using the bootm command.

The following diagram illustrates the boot process:

```
<-----------( M-mode )---------><--( S-mode )-->
+----------+   +--------------+    +------------+
|  U-Boot  |-->| SBI firmware |--->|     OS     |
+----------+   +--------------+    +------------+
```

To examine the boot process with the QEMU virt machine, you can follow the steps in the “Building U-Boot” section of the following document: QEMU RISC-V.

S-mode U-Boot

RISC-V production boot images may include a U-Boot SPL for platform-specific initialization. The U-Boot SPL then loads a FIT image (u-boot.itb), which contains a firmware (e.g. fw_dynamic) providing the SBI, as well as a regular U-Boot (or U-Boot proper) running in S-mode. Finally, the S-mode Operating System is loaded.

The following diagram illustrates the boot process:

```
<-------------( M-mode )---------><----------( S-mode )------->
+------------+   +--------------+    +----------+   +----------+
| U-Boot SPL |-->| SBI firmware |--->|  U-Boot  |-->|    OS    |
+------------+   +--------------+    +----------+   +----------+
```

To examine the boot process with the QEMU virt machine, you can follow the steps in the “Running U-Boot SPL” section of the following document: QEMU RISC-V.

Toolchain

You can build the RISC-V GNU toolchain from scratch, or download a pre-built toolchain from the releases page.