```
SiFive FSBL:       2018-03-20
HiFive-U serial #: 000001e0

OpenSBI v0.4 (Sep 18 2019 22:56:42)
   ____                    _____ ____ _____
  / __ \                  / ____|  _ \_   _|
 | |  | |_ __   ___ _ __ | (___ | |_) || |
 | |  | | '_ \ / _ \ '_ \ \___ \|  _ < | |
 | |__| | |_) |  __/ | | |____) | |_) || |_
  \____/| .__/ \___|_| |_|_____/|____/_____|
        | |
        |_|

Platform Name          : SiFive Freedom U540
Platform HART Features : RV64ACDFIMSU
Platform Max HARTs     : 5
Current Hart           : 2
Firmware Base          : 0x80000000
Firmware Size          : 92 KB
Runtime SBI Version    : 0.1

PMP0: 0x0000000080000000-0x000000008001ffff (A)
PMP1: 0x0000000000000000-0x0000007fffffffff (A,R,W,X)

U-Boot 2020.01-rc1-00217-g10aa74cb53-dirty (Nov 09 2019 - 17:12:46 +0530)

CPU:   rv64imafdc
Model: SiFive HiFive Unleashed A00
DRAM:  8 GiB
MMC:   spi@10050000:mmc@0: 0
In:    serial@10010000
Out:   serial@10010000
Err:   serial@10010000
Net:   eth0: ethernet@10090000
Hit any key to stop autoboot:  0
=>
```

An Introduction to RISC-V Boot flow: Overview, Blob vs Blobfree standards

Jagan Teki, Amarula Solutions

China RISC-V Forum - 2019, ShenZhen

Jagan Teki

- CEO, Embedded Linux Architect at **Amarula Solutions India**
  - *Bootloader*: BootROM, bootloaders, U-Boot, boot bsps, chip/board bring ups, devicetrees, device drivers, boottime, secure boot, atf, optee and etc.
  - *Embedded Linux*: Linux bsps, devicetrees, device drivers, multimedia, optimizations, integrations and etc.

- Mainline contributions
  - **Linux**
    - Contributor of Allwinner, Rockchip, i.MX platforms, bsps, device drivers.
    - Maintainer of few **DSI** LCD panels.
  - **U-Boot**
    - Contributor of Xilinx Zynq, Allwinner, Rockchip, i.MX platforms, bsps, device drivers.
    - Maintainer of Allwinner **sunXi** SoCs
    - Maintainer of **SPI/SPI-NOR** Subsystems
  - Contributor of **Buildroot**, **Yocto**

Agenda

RISC-V Overview

Boot flow

- ➔ Processor modes
- ➔ In a Nutshell
- ➔ ARM64 Boot flow
- ➔ RISC-V Boot flow

OpenSBI

- ➔ SBI
- ➔ OpenSBI
- ➔ OpenSBI, firmware implementations

Summary

- ➔ Demo on SiFive HiFive Unleashed A00 board
- ➔ Future plans

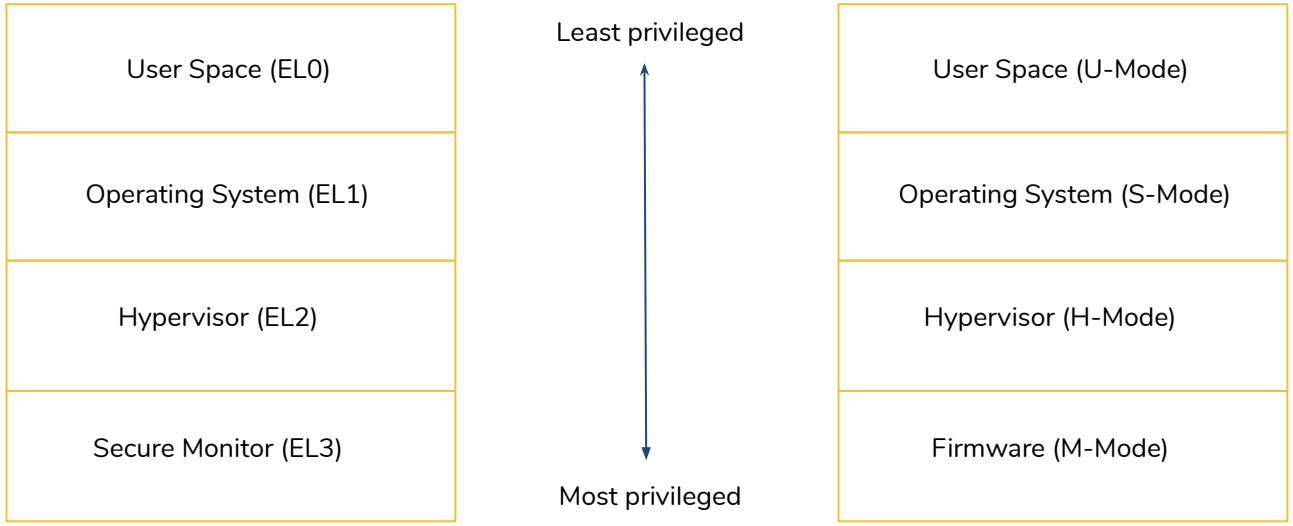
RISC-V

- pronounced "risk-five"
- Open Source Instruction Set Architecture (ISA) for Reduced Instruction Set Computer(RISC).
- Typical load-store instruction architecture.
- Targeted for low/high-end embedded systems to high-end super computers.
- *Several CPU, SoC and Research groups*: SiFive, Syntacore, Andes Technology, Ariane, Greenwaves Technology, Kendryte, Shakti, Hex Fivel, Western Digital, Alibaba Group.
- *Deterministic ecosystem* includes universities, summits, forums, alliances, meetup like Berkeley, Tsinghua-Berkeley, IIT-M, RISC-V Summit, China RISC-V Forum, Chip Alliance, China RISC-V Alliance, Taiwan RISC-V Alliance, LF RISC-V summit and much more.
- Software ports:
  - Bootloaders: U-Boot, Coreboot, EDK2, Oreboot, EFI
  - Linux kernel
  - Build Systems/distros: Buildroot, yocto, Fedora
- Hardware ports:
  - QEMU: RISC-V 32/64-bit
  - HiFive1 Freedom E310
  - HiFive Unleashed
  - IGLOO2 RISC-V

Boot flow

- ➔ Processor modes
- ➔ Boot flow, In a Nutshell
- ➔ ARM64 Boot flow
- ➔ RISC-V Boot flow

RISC Processor modes



ARM64 Exception Levels

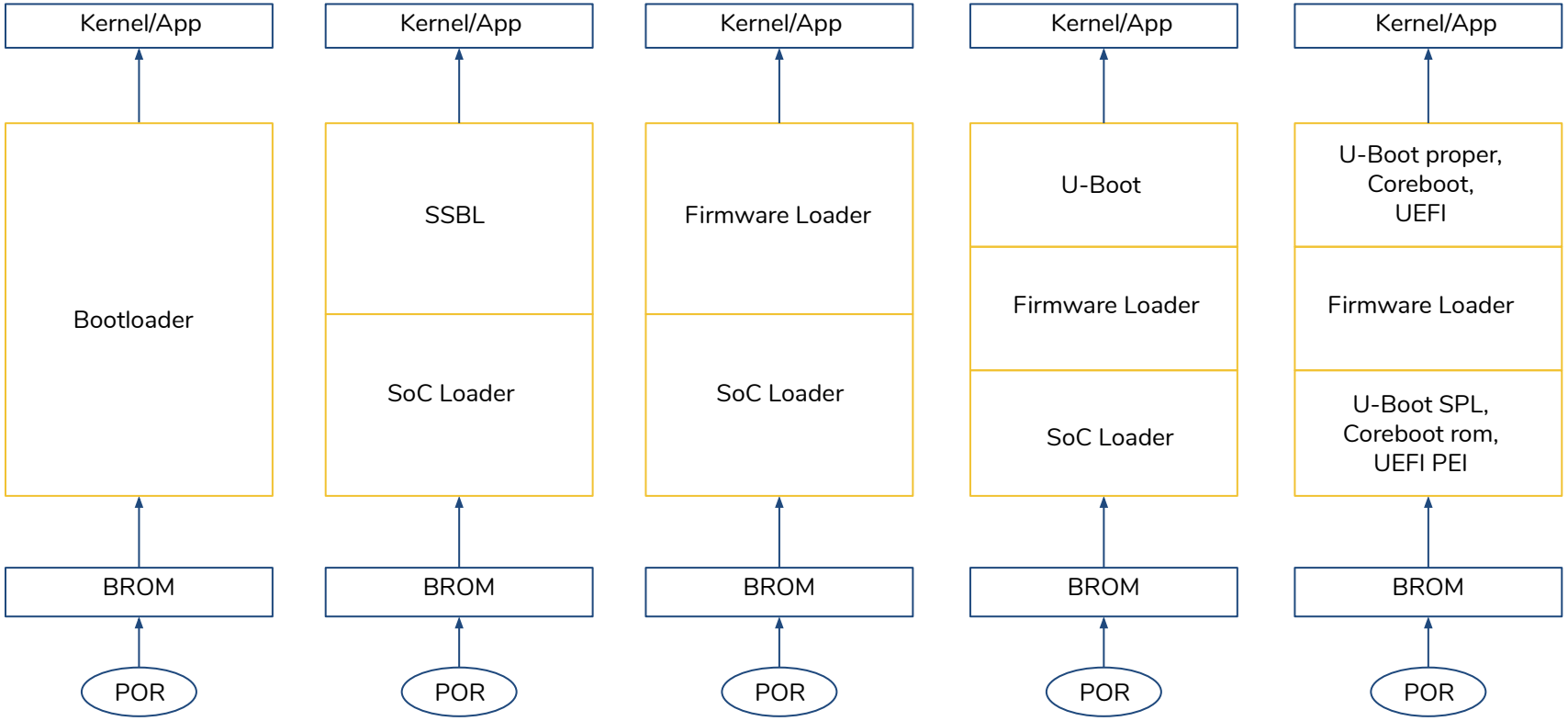
- ➔ EL3 has platform specific runtime firmware.
- ➔ EL3 has secure privileges.
- ➔ ARM64 start from EL3, means in secure world
- ➔ Bootloaders(non-secure) uses **ARM Trusted firmware (TF-A)** switch normal world EL2 since system boot from secure EL3.

RISC-V Privilege Modes

- ➔ M-Mode has platform specific runtime firmware(only).
- ➔ M-Mode does have secure privileges.
- ➔ RISC-V start from M-Mode, A bare metal machine mode.
- ➔ Bootloaders uses **OpenSBI** switch into S-Mode from M-Mode for non-hypervisor world.

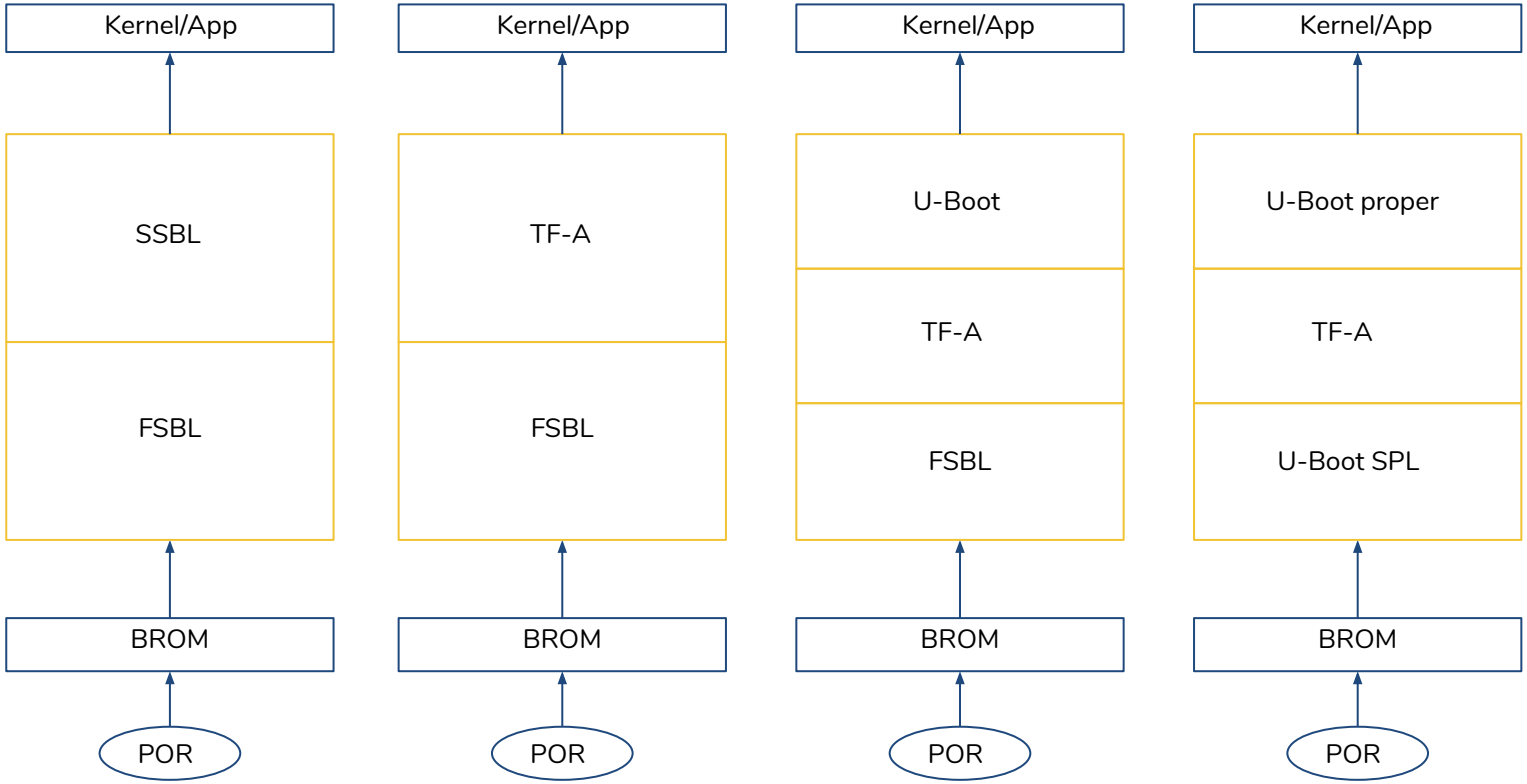
Note: Comparing processor modes here is for the sake of understanding but the actual modes of operations are purely platform specific.

Embedded Boot flow, In a Nutshell



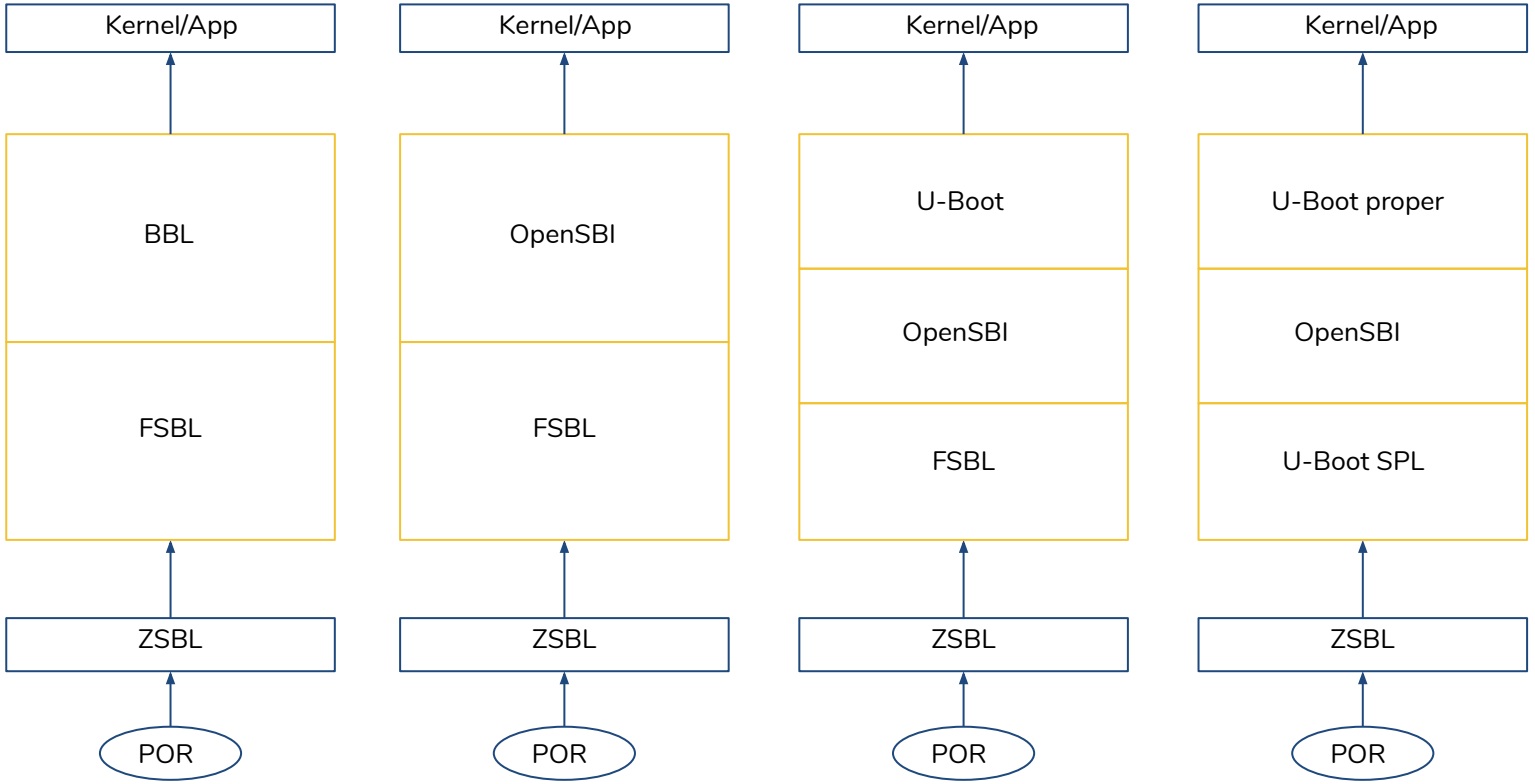
BROM: BootROM, **SoC Loader**: MLO, FSBL, **SSBL**: Second stage bootloader U-Boot, Coreboot, UEFI, **Firmware Loader**: TF-A, OpenSBI

ARM64 Boot flow



BROM: BootROM, **FSBL**: First stage bootloader, MLO, FSBL, **SSBL**: Second Stage bootloader, U-Boot, Coreboot, UEFI, **TF-A**: ARM Trusted Firmware

RISC-V Boot flow



ZSBL: Zero Stage Bootloader(BROM), **BBL**: Berkeley Bootloader, **FSBL**: First Stage Bootloader, **OpenSBI**: RISC-V Open Source Supervisory Binary Interface

OpenSBI

- ➔ SBI
- ➔ OpenSBI
- ➔ OpenSBI, firmware implementations

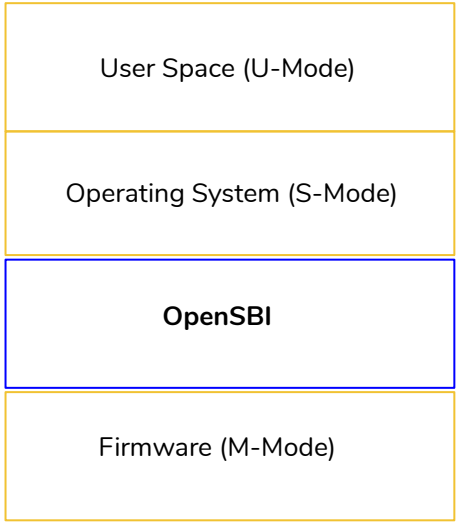
OpenSBI

SBI

- ➔ RISC-V Supervisor Binary Interface
- ➔ System call type interface layer between Firmware runtime, M-Mode to Operating system, S-Mode.
- ➔ Avoid fragmentation of various OEM silicon providers specific runtime firmware implementations.
- ➔ Standard, generic runtime firmware interface specification across all OSes, different cpu and silicon platforms.
- ➔ Specification in SBI v0.1 in usage (v0.2 in draft)

OpenSBI

- ➔ RISC-V Open Source Supervisory Binary Interface
- ➔ An Open Source implementation of SBI specification, BSD-2 license
- ➔ Modular, Scalable and Extendable between all CPU and Silicon specific hardware configurations.
- ➔ Contains platform-independent and platform-dependent libraries like libsbi.a, libplatsbi.a
- ➔ Platforms supports like SiFive U540, Andes AE350, Ariane FPGA, Kendryte K210, QEMU



RISC-V Privilege Modes, non-hyp

Source: *SBI*; https://github.com/riscv/riscv-sbi-doc *OpenSBI*; https://github.com/riscv/opensbi

OpenSBI, firmware implementations

FW_PAYLOAD

- ➔ Pack the firmware with next level boot stage as payload, fw_payload.bin
- ➔ Can be packable with U-Boot, Kenel

FW_DYNAMIC

- ➔ Pack the firmware with runtime accessible to the next level boot stage, fw_dynamic.bin
- ➔ Can be packable in U-Boot SPL, Coreboot

FW_PAYLOAD

➔ Build Linux mainline repository [1]

```
₹ ARCH=riscv CROSS_COMPILE=riscv64-buildroot-linux-gnu- make defconfig
₹ ARCH=riscv CROSS_COMPILE=riscv64-buildroot-linux-gnu- make Image dtbs
```

➔ Build U-Boot mainline repository [2]

```
₹ ARCH=riscv CROSS_COMPILE=riscv64-buildroot-linux-gnu-  \
  > make sfive_fu540_defconfig \
₹ ARCH=riscv CROSS_COMPILE=riscv64-buildroot-linux-gnu- make
```

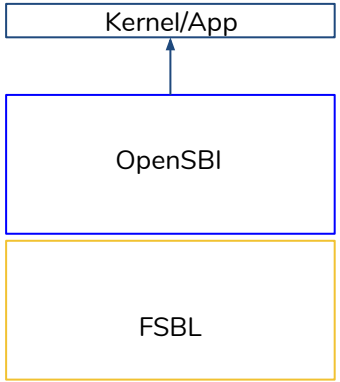
➔ Build OpenSBI[3], with Linux as payload

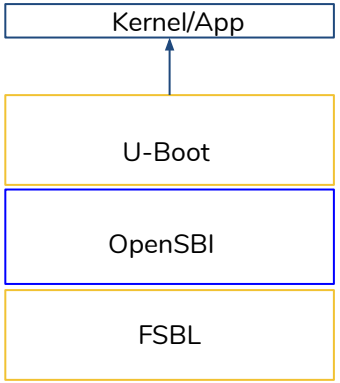
```
₹ CROSS_COMPILE=riscv64-buildroot-linux-gnu- make \
  > PLATFORM=sifive/fu540 \
  > FW_PAYLOAD_PATH=</path/to/linux/arch/riscv/boot/Image>
```

➔ Build OpenSBI[3], with U-Boot as payload

```
₹ CROSS_COMPILE=riscv64-buildroot-linux-gnu- make \
  > PLATFORM=sifive/fu540 \
  > FW_PAYLOAD_PATH=</path/to/u-boot/u-boot-dtb/bin>
```

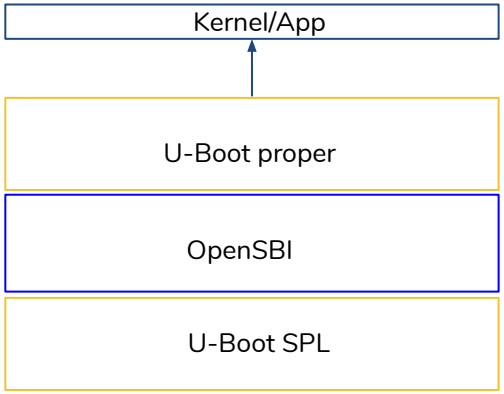
Output:
fw_payload.bin





[1] https://git.kernel.org/pub/scm/linux/kernel/git/next/linux-next.git *[2]* https://gitlab.denx.de/u-boot/u-boot *[3]* https://github.com/riscv/opensbi

FW_DYNAMIC



Output:
u-boot-spl-dtb.bin
u-boot.itb

➔ Build OpenSBI from [1]

```
₹ CROSS_COMPILE=riscv64-buildroot-linux-gnu- \
  > make PLATFORM=sifive/fu540
```

➔ Build U-Boot from [2]

```
₹ CROSS_COMPILE=riscv64-buildroot-linux-gnu-  \
  > make sfive_fu540_spl_defconfig \
₹ export OPENSBI=</path/to/fw_dynamic.bin>
₹ CROSS_COMPILE=riscv64-buildroot-linux-gnu- make
```

```
/dts-v1/;
/ {
    description = "Configuration to load OpenSBI before U-Boot";
    images {
            uboot {
                        description = "U-Boot";
                        data = /incbin/("u-boot-nodtb.bin");
                        type = "standalone";
                        os = "U-Boot";
                        arch = "riscv";
                        compression = "none";
                        load = <0x81200000>;
            };
            opensbi {
                        description = "RISC-V OpenSBI";
                        data = /incbin/("/home/jagan/work/code/riscv/fw_dynamic.bin");
                        type = "firmware";
                        os = "opensbi";
                        arch = "riscv";
                        compression = "none";
                        load = <0x81000000>;
                        entry = <0x81000000>;
            };
            fdt_1 {
                        description = "hifive-unleashed-a00";
                        data = /incbin/("arch/riscv/dts/hifive-unleashed-a00.dtb");
                        type = "flat_dt";
                        compression = "none";
            };
    };
    configurations {
            default = "config_1";
            config_1 {
                        description = "hifive-unleashed-a00";
                        firmware = "opensbi";
                        loadables = "uboot";
                        fdt = "fdt_1";
            };
    };
};
```

[1] https://github.com/riscv/opensbi *[2]* https://github.com/amarula/u-boot-amarula

Summary

- ➔ Demo, runs
- ➔ Future plans

Demo - SiFive HiFive Unleashed A00

- ➔ Complete information including Building, flashing and setting bootmode is available at https://wiki.amarulasolutions.com/bsp/riscv/hifive-unleashed.html#booting
- ➔ Demo link has information about OpenSBI using fw_payload.
- ➔ fw_dynamic is still working under progress, but the same link will update once done in few weeks.

```
SiFive FSBL:       2018-03-20
HiFive-U serial #: 000001e0

OpenSBI v0.4 (Sep 18 2019 22:56:42)
   ____                    ____ ____ _____
  / __ \                  / ___|  _ \_   _|
 | |  | |_ __   ___ _ __ | (___ | |_) || |
 | |  | | '_ \ / _ \ '_ \ \___ \|  _ < | |
 | |__| | |_) |  __/ | | |____) | |_) || |_
  \____/| .__/ \___|_| |_|_____/|____/_____|
        | |
        |_|

Platform Name          : SiFive Freedom U540
Platform HART Features : RV64ACDFIMSU
Platform Max HARTs     : 5
Current Hart           : 2
Firmware Base          : 0x80000000
Firmware Size          : 92 KB
Runtime SBI Version    : 0.1

PMP0: 0x0000000080000000-0x000000008001ffff (A)
PMP1: 0x0000000000000000-0x0000007fffffffff (A,R,W,X)


U-Boot 2020.01-rc1-00217-g10aa74cb53-dirty (Nov 09 2019 - 17:12:46 +0530)

CPU:   rv64imafdc
Model: SiFive HiFive Unleashed A00
DRAM:  8 GiB
MMC:   spi@10050000:mmc@0: 0
In:    serial@10010000
Out:   serial@10010000
Err:   serial@10010000
Net:   eth0: ethernet@10090000
Hit any key to stop autoboot:  0
=>
```

Future plans

- ➔ U-Boot SPL (WIP)
- ➔ EDK2 Mainline (WIP)
- ➔ Linuxboot ?
- ➔ More CPU/SoC support in bootloaders, Linux Kernel
- ➔ More distributions Debian, ArchLinux ?

References

- Working experience on RISC-V
- Wiki - https://wiki.amarulasolutions.com/bsp/riscv/hifive-unleashed.html
- OpenSBI - https://github.com/riscv/opensbi
- Atish Patra “RISC-V Boot Process: One step at a time”
  https://static.sched.com/hosted_files/osseu19/c3/ELCE_2019_final_upload.pptx
- Anup Patel “Xvisor: Embedded Hypervisor for RISC-V”
  https://static.sched.com/hosted_files/osseu19/4e/Xvisor_Embedded_Hypervisor_for_RISCV_v5.pdf

Questions??

Thank you

Jagan Teki <jagan@amarulasolutions.com>