Building a low-key RISC-V Linux XIP system

Vitaly Wool, Konsulko Group

About me

- ❑ Has been with embedded Linux since 2003
- ❑ Worked for MontaVista
- ❑ Currently living in Sweden (Skåne)
- ❑ Staff Engineer at Konsulko Group
- ❑ Managing Director at Konsulko AB

About this presentation

- ❑ A few words on RISC-V
- ❑ A few words on XIP
- ❑ Linux XIP on RISC-V: why and how
- ❑ Future work on RISC-V XIP
- ❑ Demo (BeagleV)
- ❑ Conclusions

RISC-V

RISC-V

- open source hardware Instruction Set Architecture (ISA)
  - RISC (reduced instruction set)
  - Royalty free for any chip manufacturer
  - Developed by UC Berkeley
    - V postfix because it's fifth UC Berkeley RISC design
  - Standard maintained by non-profit RISC-V foundation
- It's not a CPU implementation nor a company
  - But you can implement a CPU using RISC-V ISA

RISC-V vs ARM

- ...in a nutshell

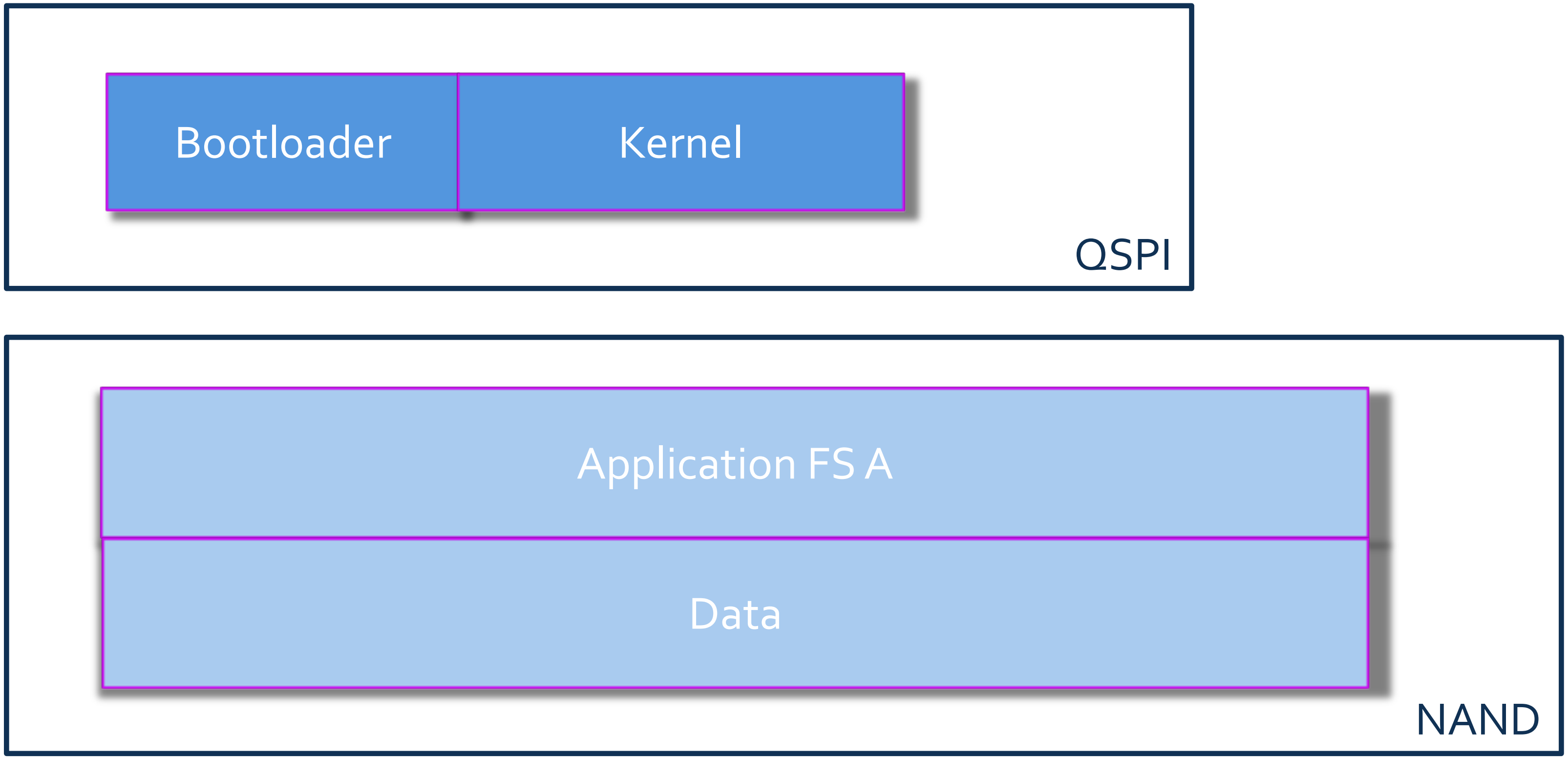
RISC-V	ARM
RISC instruction set (load/store)	RISC instruction set (load/store)
Can be 32-, 64- and 128-bit	Can be 32- and 64-bit
Open and royalty free ISA	Proprietary, licensed
Relatively new, community is emerging	Leading architecture, well-established community
Linux XIP supported	Linux XIP supported

XIP

XIP: execute in place

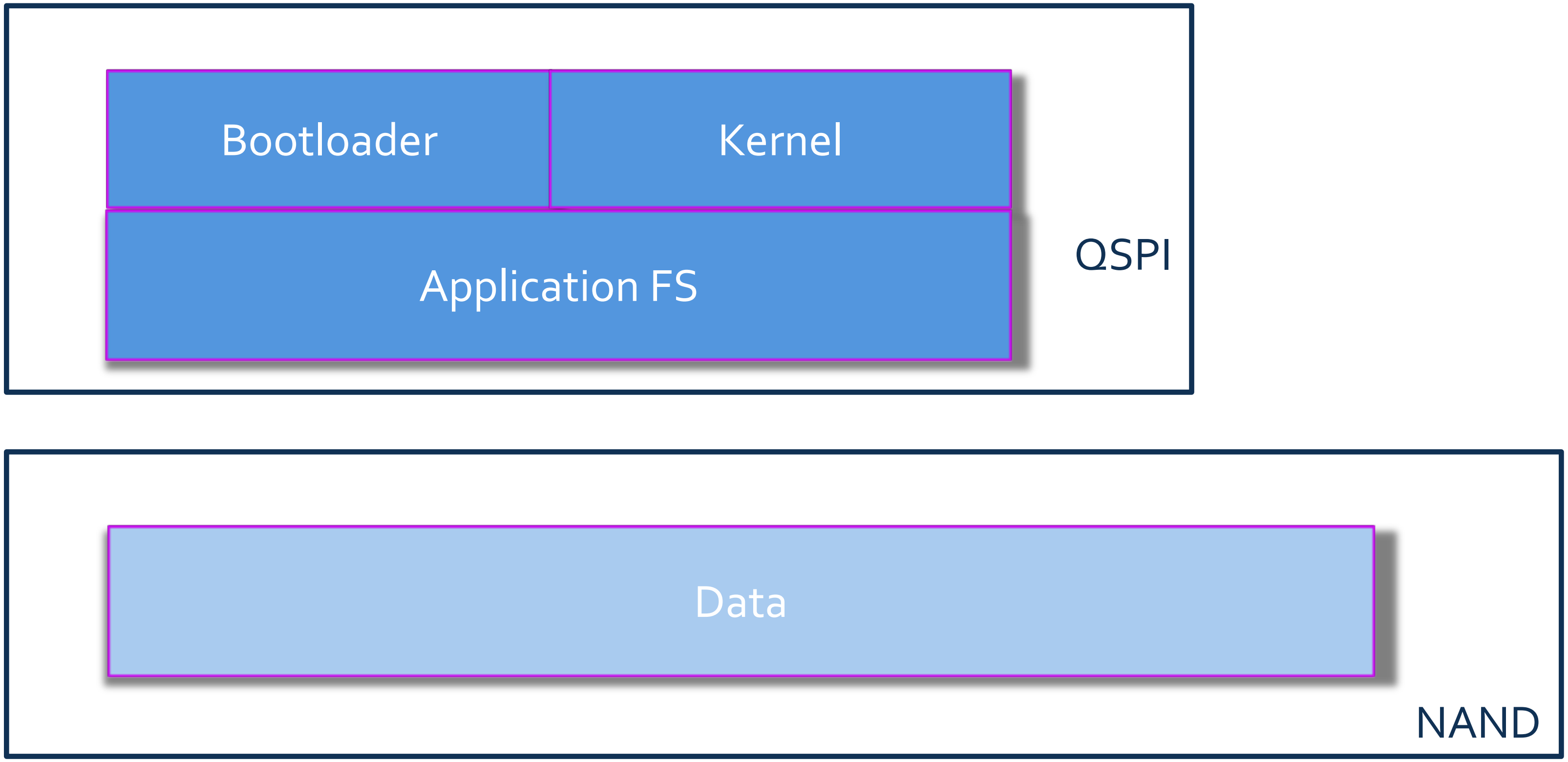
- Code executed directly from persistent storage
  - Typically NOR flash
  - QSPI
- XIP kernel
  - Option selected at compile time
- XIP userspace
  - Requires a special filesystem
    - Cramfs (legacy), AXFS

Kernel XIP



Traditional XIP design (userspace can be anywhere)

Kernel/Userspace XIP



Possibly more expensive design but we save RAM

XIP advantages

- Less RAM needed
  - Usually up to 10x smaller RAM footprint
  - Sometimes no RAM at all is needed
- Lower idle power consumption
  - May be crucial for IoT running on battery
- Shorter boot time
  - No copy on boot
- Faster execution
  - QSPI flash

XIP obstacles

- You can’t write to flash and execute from it at the same time
- However, you can write to flash using special tricks
  - Code copied/executed from RAM
  - No other code may be executed during that time
- XIP requires more space on flash storage
  - At least kernel code can not be compressed
- All addresses are defined at compile time
  - Which may be a security compromise

XIP on RISC-V

Why do XIP on RISC-V?

- QSPI flash in most designs
  - Medium to XIP from is likely to be there
- Many cheaper RISC-V boards target IoT
  - XIP reduces power consumption
- RISC-V SBI firmware allows to bypass u-boot completely
  - Works for an XIP kernel but not for a usual kernel
  - Significantly reduces boot-up time
- Many RISC-V boards have small RAM
  - XIP allows to run Linux on these

XIP on RISC-V : current status

- ❑ XIP support for RISC-V is in the mainline!
  - Only 64-bit RISC-V designs are supported now
  - Since 5.13
  - Created by the author of this presentation with some great help from Alexander Ghiti
- ❑ Just the second platform to get XIP support
  - ARM (32-bit) is the other one
- ❑ Developed and tested mostly on PolarFire Icicle Kit
- ❑ Verified to work on SiFive and Beagle-V

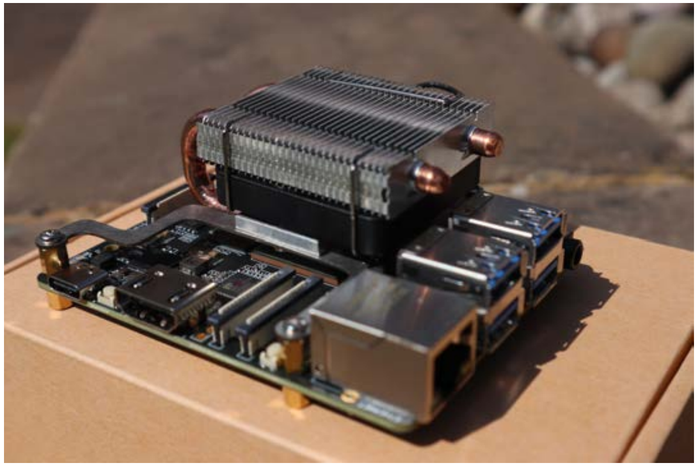
XIP on RISC-V : work to be done

- ❑ Add support for MMU-less boards
  - ▪ Boards like Kendryte K210 are the main target for XIP
- ❑ Add .data/.sdata section compression
  - ▪ Will leave more space for XIP userspace
- ❑ Add support for 32-bit RISC-V designs
- ❑ Optimize address translations
  - ▪ Branching on “NOR flash vs RAM” slows down things
- ❑ Test on more boards ☺

BeagleV demo

BeagleV board

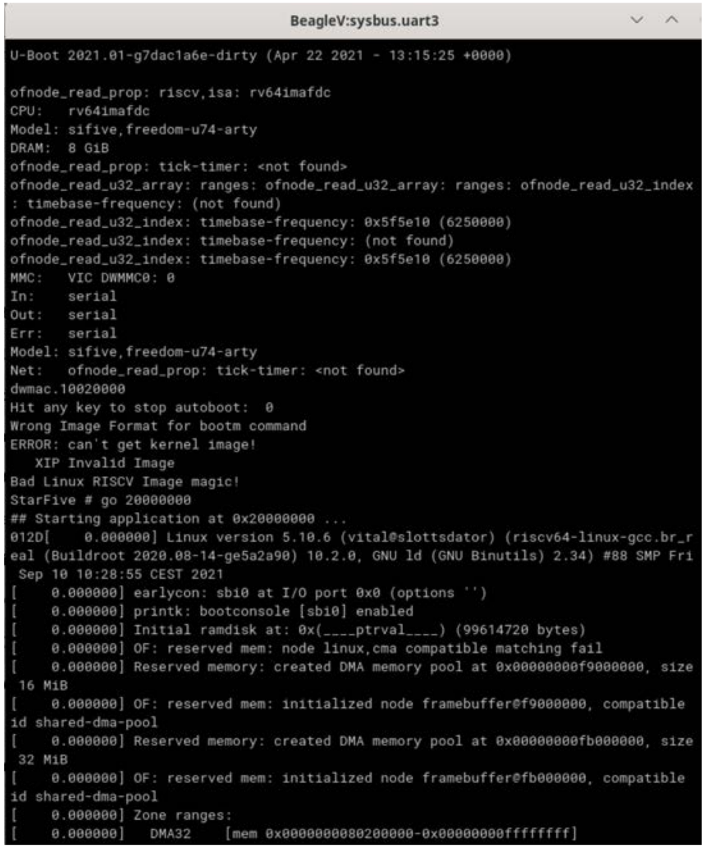
- ❑ Raspberry Pi like board with a 64-bit RISC-V CPU
- ❑ Designed by BeagleBoard® Foundation
- ❑ Incorporates SiFive U74 2-core CPU
  - ▪ Roughly on par with ARM Cortex-A55
- ❑ 8GB LPDDR4 RAM
- ❑ 32 MB QSPI NOR flash
- ❑ uSD, HDMI, Ethernet, audio, and USB ports
- ❑ ...and more



BeagleV software

- Complete Linux build HOWTO: https://bootlin.com/blog/buildroot-beagle-v/
  - Thanks Bootlin!
- This build is using 5.10 kernel which doesn’t support XIP on RISC-V
  - I was using earlier version of my RISC-V XIP patch (available on demand of course)
- Some configuration changes are to be made too
  - STRICT_RWX must be turned off
  - Memory model should be SPARSEMEM
  - QSPI flash is at 0x20000000, not 0x21000000

Beagle-V demo



```
U-Boot 2021.01-g7dac1a6e-dirty (Apr 22 2021 - 13:15:25 +0000)

ofnode_read_prop: riscv,isa: rv64imafdc
CPU:   rv64imafdc
Model: sifive,freedom-u74-arty
DRAM:  8 GiB
ofnode_read_prop: tick-timer: <not found>
ofnode_read_u32_array: ranges: ofnode_read_u32_array: ranges: ofnode_read_u32_index
: timebase-frequency: (not found)
ofnode_read_u32_index: timebase-frequency: 0x5f5e10 (6250000)
ofnode_read_u32_index: timebase-frequency: (not found)
ofnode_read_u32_index: timebase-frequency: 0x5f5e10 (6250000)
MMC:   VIC DWMMC0: 0
In:    serial
Out:   serial
Err:   serial
Model: sifive,freedom-u74-arty
Net:   ofnode_read_prop: tick-timer: <not found>
dwmac.10020000
Hit any key to stop autoboot:  0
Wrong Image Format for bootm command
ERROR: can't get kernel image!
   XIP Invalid Image
Bad Linux RISCV Image magic!
StarFive # go 20000000
## Starting application at 0x20000000 ...
012D[    0.000000] Linux version 5.10.6 (vital@slottsdator) (riscv64-linux-gcc.br_r
eal (Buildroot 2020.08-14-ge5a2a90) 10.2.0, GNU ld (GNU Binutils) 2.34) #88 SMP Fri
 Sep 10 10:28:55 CEST 2021
[    0.000000] earlycon: sbi0 at I/O port 0x0 (options '')
[    0.000000] printk: bootconsole [sbi0] enabled
[    0.000000] Initial ramdisk at: 0x(____ptrval____) (99614720 bytes)
[    0.000000] OF: reserved mem: node linux,cma compatible matching fail
[    0.000000] Reserved memory: created DMA memory pool at 0x00000000f9000000, size
 16 MiB
[    0.000000] OF: reserved mem: initialized node framebuffer@f9000000, compatible
id shared-dma-pool
[    0.000000] Reserved memory: created DMA memory pool at 0x00000000fb000000, size
 32 MiB
[    0.000000] OF: reserved mem: initialized node framebuffer@fb000000, compatible
id shared-dma-pool
[    0.000000] Zone ranges:
[    0.000000]   DMA32    [mem 0x0000000080200000-0x00000000ffffffff]
```

Conclusions

- ❑ XIP is not a yesterday’s technology
- ❑ XIP goes very well with what RISC-V designs have to offer
- ❑ Important extra work is to be done to get the best out of XIP/RISC-V combo
- ❑ RISC-V designs also have to mature to get the best of that combo

Questions?

Vitaly.Wool@konsulko.com