COLLABORA

Getting the RK3588 SoC supported upstream

Sebastian Reichel

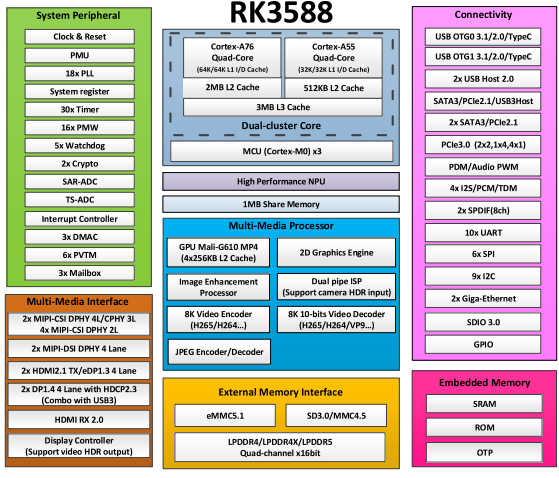
Collabora

COLLABORA

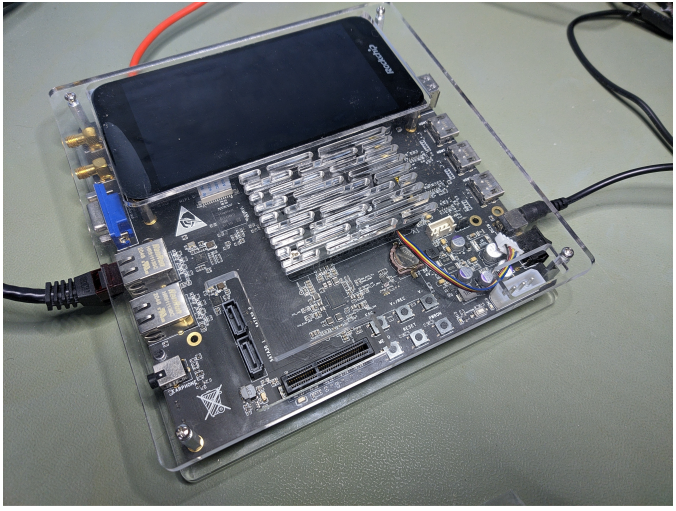
Who am I?

- Kernel Engineer at Collabora
- Maintainer of kernel's power-supply subsystem
- Debian Developer
- Living in Oldenburg, Germany
  - Co-Founder of the local hackerspace
  - Deputy Lead of the Fire Brigade Diver Squad

What is Rockchip RK3588?



Rockchip RK3588 EVB1



First steps

- Downstream/Vendor kernel source was/is available
  - e.g. https://github.com/radxa/kernel/tree/stable-5.10-rock5/
- Create minimal DT using info from vendor tree
- Rely on bootloader to have initialized PMIC, pinctrl, clocks
- Aim: Get Kernel boot messages on UART

COLLABORA

Minimal DT

```
/dts-v1/;

/ {
    compatible = "rockchip,rk3588";

    #address-cells = <2>;
    #size-cells = <2>;

    aliases { serial2 = &uart2; };
    chosen { stdout-path = "serial2:1500000n8"; };

    uart2: serial@feb50000 {
        compatible = "rockchip,rk3588-uart", "snps,dw-apb-uart";
        reg = <0x0 0xfeb50000 0x0 0x100>;
        reg-io-width = <4>;
        reg-shift = <2>;
    };
};
```

Booting

```
[    0.000000] Booting Linux on physical CPU 0x0000000000 [0x412fd050]
...
[    0.000000] Machine model: rockchip,rk3588
...
[    0.000000] Kernel command line: earlycon=uart8250,mmio32,0xfeb50000 loglevel=8
...
[    0.000000] timer_probe: no matching timers found
[    0.000000] Kernel panic - not syncing: Unable to initialise architected timer.
```

Add Timer (and IRQ)

```
/ {
    interrupt-parent = <&gic>;
    timer {
        compatible = "arm,armv8-timer";
        interrupts = <GIC_PPI 13 IRQ_TYPE_LEVEL_HIGH 0>,
                     <GIC_PPI 14 IRQ_TYPE_LEVEL_HIGH 0>,
                     <GIC_PPI 11 IRQ_TYPE_LEVEL_HIGH 0>,
                     <GIC_PPI 10 IRQ_TYPE_LEVEL_HIGH 0>,
                     <GIC_PPI 12 IRQ_TYPE_LEVEL_HIGH 0>;
        interrupt-names = "sec-phys", "phys", "virt", "hyp-phys", "hyp-virt";
    };
    gic: interrupt-controller@fe600000 {
        compatible = "arm,gic-v3";
        reg = <0x0 0xfe600000 0 0x10000>, <0x0 0xfe680000 0 0x100000>;
        interrupts = <GIC_PPI 9 IRQ_TYPE_LEVEL_HIGH 0>;
        interrupt-controller;
    };
}
```

Booting

```
[    0.000000] Kernel command line: earlycon=uart8250,mmio32,0xfeb50000 loglevel=8 keep_bootcon
...
[    0.001337] printk: console [tty0] enabled
...
[    0.006691] No CPU information found in DT
[    0.007111] cacheinfo: Failed to find cpu0 device node
[    0.007592] cacheinfo: Unable to detect cache hierarchy for CPU 0
[    0.008174] Unable to handle kernel NULL pointer dereference at virtual address 00000000000003f0
...
[    0.030513] Kernel panic - not syncing: Attempted to kill init! exitcode=0x0000000b
```

Add CPU node to DT

```
/ { cpus {
    #address-cells = <1>; #size-cells = <0>;

    cpu-map { cluster0 { core0 { cpu = <&cpu_l0>; }; }; };

    cpu_l0: cpu@0 {
        device_type = "cpu"; compatible = "arm,cortex-a55";
        reg = <0x0>; enable-method = "psci";
        next-level-cache = <&l2_cache_l0>; ... // L1 cache info
    };

    l2_cache_l0: l2-cache-l0 {
        compatible = "cache"; next-level-cache = <&l3_cache>;
        ... /* L2 cache info */
    };

    l3_cache: l3-cache { compatible = "cache"; ... // L3 cache info };
}; };
```

Booting

```
...
[    1.747049] /dev/root: Can't open blockdev
[    1.747486] VFS: Cannot open root device "" or unknown-block(0,0): error -6
[    1.748138] Please append a correct "root=" boot option; here are the available partitions:
...
[    1.750916] Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(0,0)
[    1.751688] CPU: 0 PID: 1 Comm: swapper/0 Not tainted 6.6.0-rc1-00038-gbf75d6f0324e #1110
[    1.752453] Hardware name: rockchip,rk3588 (DT)
```

Boot device

- RK3588 has a bunch of options
  - PCIe / NVMe - PCIe, PHY, pmdomain, clocks, reset
  - SATA - PCIe, PHY, genpd, clocks, reset
  - USB - USB, PHY, genpd, clocks, reset
  - SD card - n/a
  - SDIO - n/a
  - eMMC - eMMC, clocks, reset
  - Network - GMAC, genpd, clocks, reset

How to proceed?

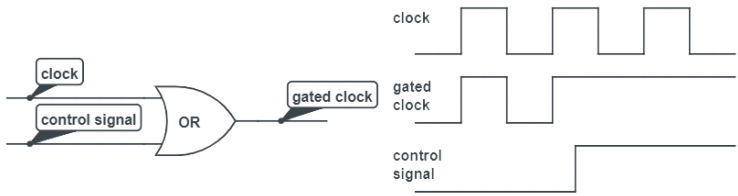
- Goal: Boot Debian userspace

1. Clocks / Resets
2. Pinctrl
3. eMMC
4. Normal console

Clocks & Reset

- https://lore.kernel.org/linux-clk/20221018151407.63395-1-sebastian.reichel@collabora.com/
- Nothing fundamentally new compared to previous generations
- DT maintainers asked for continuous RESET IDs
  - so far Rockchip just used register offsets (i.e. RK3568)
  - had to introduce a lookup table in the kernel
- Now makes exhaustive use of special clock gates (ca. 20)
  - also known as NIU (native interface unit)
  - badly documented

Linked Clock Gates



- Rockchip has special variant
- Gated clock is only enabled when all apply:
  - parent clock is enabled
  - control signal is enabled
  - linked clocks are also enabled

Linked Clock Gates

- Clock Framework only supports one active parent
- Rockchip solved this by introducing new driver
  - driver had one DT node per clock
  - registers it as normal gates clock
  - enables the extra required clocks via runtime PM
- For RK3568 upstream choose to just mark the “linked” clocks critical
  - this obviously wastes power
  - but does not affect ABI
  - idea is to fix the behaviour once clock framework is ready

Linked Clock Gates

- https://lore.kernel.org/all/20230807065942.9937-1-zhangqing@rock-chips.com/
- Recent contribution from Rockchip
  - Create a Rockchip clk_gate_link type
  - Extension from normal “clk_gate”
  - enable() and disable() call existing clk_gate_endisable()
  - additionally enable()/disable() also handle the linked clock
- (Still pending)

Pinmux / Pinctrl / GPIO

- RK3588 has V2.1 GPIO controller - fully compatible with V2.0
- Pinmux registers are slightly different (like always. . .)
- Otherwise nothing particular interesting
- Not initially required (rely on bootloader)
- But wanted for initial upstreaming and quite trivial

eMMC

- First RK3588 change that made it upstream
- Basically needed two changes
  - reset handling to workaround hardware issues
  - slightly different clock handling compared to RK356x
- v6.4 had a regression breaking HS200/HS400
  - checking max. allowed clock speed overflowed the clock driver
  - resulting in a very small clock speed being returned
  - https://lore.kernel.org/linux-clk/20230526171057.66876-2-sebastian.reichel@collabora.com/
  - https://lore.kernel.org/linux-clk/20230630183835.464216-1-sebastian.reichel@collabora.com/

Time to upstream

- Board boots far enough, that it becomes usable
- It's fine to upstream incomplete DT
- Reviewing needs time, so upstreaming and adding more features can be done in parallel
- Working too long behind closed doors means other people might do the same work

Network & PMdomain

- RK3588 GMAC need PM domain
  - 6541b424ce1d: Adding basic PM domain support
  - 2f2b60a0ec28: Adding GMAC support
- colleague received Radxa Rock 5A and GMAC didn't work :(
  - 88619e77b33d: Fix support for GMAC1
- next colleague received Radxa Rock 5B
  - uses PCIe ethernet instead of GMAC
  - PCIe needs lots of things :(
  - among others: GIC ITS support

GIC Interrupt Translation Service (ITS)

- PCIe uses MSI (message signaled interrupts)
- ARM GIC (Generic Interrupt Controller) has ITS for this
- RK3588 has a design flaw
  - GIC600 is only connected to AXI, but not to ACE/ACE-lite
  - Thus it's not cache coherent
  - Registers may not be configured shareable
- 32-bit limitation issue from RK356x got fixed in hardware
- Marc Zyngier: Either Errata # or rework all of GIC-ITS driver

GIC Interrupt Translation Service (ITS)

- After lot's of discussion got Errata #3588001 from Rockchip
  - https://lore.kernel.org/all/20230418142109.49762-1-sebastian.reichel@collabora.com/
- Generic solution is now worked on by Lorenzo Pieralisi
  - https://lore.kernel.org/all/20230905104721.52199-1-lpieralisi@kernel.org/

PMIC

- All known RK3588 boards use one of the following
  - 2x RK806 (SPI)
  - 1x RK806 (SPI) + multiple RK8602 (I2C)
- RK8602 turned out to be a Fairchild FAN53555 clone
- RK806 is similar to RK808, but connected via SPI
  - RK808 driver is MFD using regmap
  - Subdrivers did things like
    - `struct rk808 *rk808 = dev_get_drvdata(pdev->dev.parent);`
    - `struct i2c_client *client = rk808->i2c;`
  - Reworking was a pain because it involves a lot of subsystems

More SoC hardware

- Many IP blocks are very similar to previous generation
  - PWM, I2C, SPI, I2S only got a new compatible string
  - USB 2 needed new compatible + extra clocks
- Some of the existing DT bindings were broken
  - PCIe, SATA
- Some peripherals were low hanging fruits
  - ADC, OTP, Watchdog
- AV1 codec work started very early
  - needed as reference implementation in V4L2

Kernel Status

- DONE: UART, eMMC, SD, GPIO, Pinctrl, PMdomain, IRQ, Clocks, SATA, USB2, PCIe, Ethernet, PWM, I2C, SPI, I2S, ADC, OTP, Watchdog, AV1
- WIP
  - HDMI Out - Cristian Ciocaltea (Collabora)
  - HDMI In - Shreeya Patel (Collabora)
  - USB3 - Sebastian Reichel (Collabora)
  - GPU - Panthor/Panfrost team (Collabora, ARM)
  - Crypto - Corentin Labbe (Baylibre)
  - DFI - Sascha Hauer (Pengutronix)
- TODO: Displayport, DSI, CSI, ISP, Video Codecs, SPDIF, CAN, RNG

U-Boot

- RK3588 EVB1: low priority
- Radxa Rock 5B
  - Downstream U-Boot does support PCIe and USB
  - Ethernet is required for KernelCI
  - PCIe ethernet has no driver
  - USB ethernet dongle does not work

U-Boot Downstream

- Eugen Hristev found the bug resulting in failing USB ethernet
- CONFIG_SET_ETHADDR was set
  - this auto-generates the MAC from the serial number
  - this MAC also applies to USB Ethernet dongle
  - USB eth dongle filters incoming packets based on its own MAC
- => Downstream is usable for KernelCI (and our developers)

U-Boot Upstream

- Step 1: Add basic RK3588 support
  - Rockchip sent their vendor patches at the beginning of the year
  - Jagan Teki sent a series based on the kernel DT changes a few days later
  - Eugen Hristev started working on Rock 5B U-Boot around the same time
- Step 2: Add basic Rock 5B support
  - Surprise: Kernel crashes
  - Memory Gaps needed
  - Vendor U-Boot gets them via ATAGs from previous loader
- Step 3: Add PCIe Ethernet support to upstream solution
  - Port PHY and PCIe driver
  - Add RTL8125B support to rtl8169

U-Boot Upstream

- Step 4: Create Upstream U-Boot on-par with Downstream
  - Support SD, eMMC & SPI Flash
  - Support USB2, USB3 host
  - Support USB3 DFU
- Upstreaming DFU is blocked
  - dwc3 needs sync against Linux

U-Boot Upstream

- Rock 5A
  - Bringing Rock 5A on par with Rock 5B
    - basically just create a DT
  - Adding Ethernet support
    - Jonas Karlman got it working
    - Received Reviewed-by from Kever Yang (Rockchip) tonight!

COLLABORA

Questions?

Links

Links:

- Kernel Status Matrix
- U-Boot Status Matrix
- Rock 5B SD card images
- Trusted Firmware-A
- Looking for a Job?