Upstreaming Qualcomm SoC baseport

ELC-NA
June 2020
Vinod Koul

Linaro

whoami

- Doing embedded linux kernel work since 2007
- Worked on Intel Audio for phone foray!
- Maintainer of Dmaengine, SoundWire, ALSA compressed Audio
- Co-maintainer for Generic-Phy
- @ Linaro: Qualcomm Landing Team

Upstreaming & Qualcomm !!!

Scope

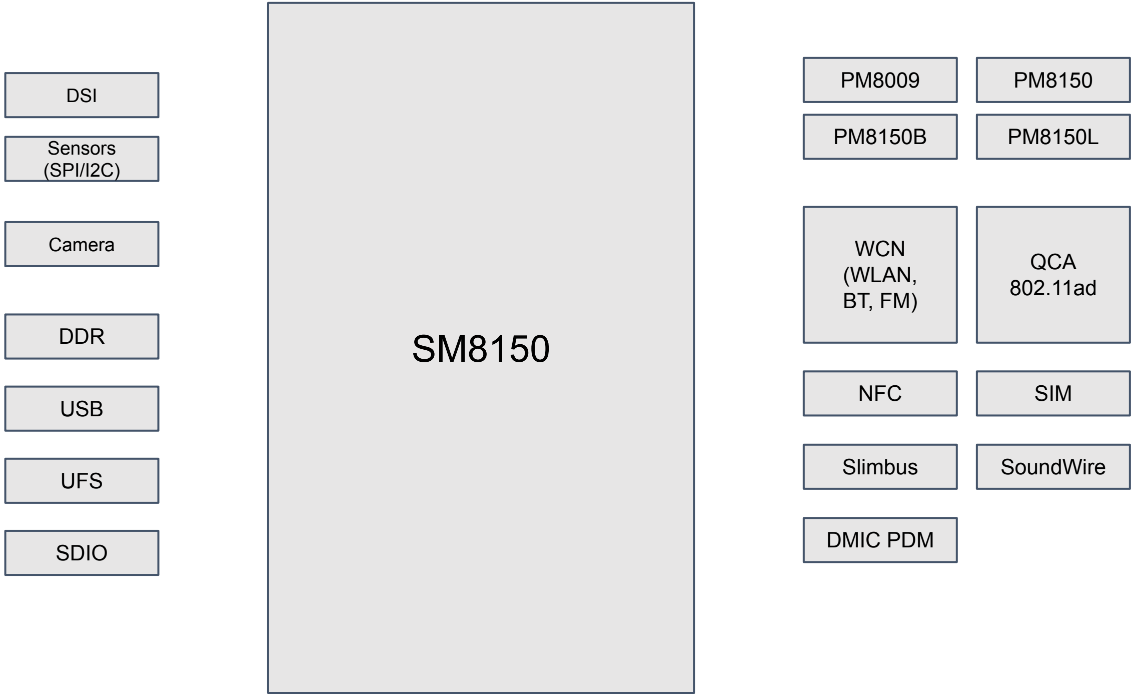
- How to go about baseport upstreaming
- Starter: Get serial console
- Pin control, clocks
- Regulators
- UFS
- USB
- Out of scope
  - Modem, multimedia, ...

Kickstart

- Downstream Source (msm-4.14 / msm-4.19 on CAF - Code Aurora)
- https://source.codeaurora.org/quic/la/kernel/
- Board schematics, if available!
- And a board :-)

SM8150 Platform

- Snapdragon Mobile
- Premier Tire Mobile SoC
- Announced: July 2019
- Pixel 4 and other premier phones features this chipset



Boot to console

- Serial driver upstream
- Use QCOM GENI
    - Compatible: Use qcom,geni-debug-uart for debug serial port
    - Do NOT use qcom,geni-uart
  - Need reduced Clock driver (describe UART clks only)
- Options: earlycon=qcom_geni,0xa90000 console=ttyMSM0,115200n8

Boot to console... DT

- Need Basic DT description for boot
- Use downstream description
- Modify & tidyup for upstream
    - Describe CPUs
        - Kryo 485 Cores
        - 1 Gold @ 3.6GHz
        - 3 Gold @ 2.7GHz
        - 4 Silver @ 2.3 GHz
        - Add new compatible

Boot to console... DT

- GCC
    - New driver and compatible
- Timer
    - Upstream
    - compatible: arm,armv7-timer-mem
- Serial
    - Upstream
    - compatible: qcom,geni-debug-uart

Pincontrol

- Downstream driver needs decent tidyup
- Bjorn Andersson added Tile support for disjoint tiles
    - Tip: Use tiles even for joint tiles
    - Bonus: Get free handing of XPU, they won’t be mapped
- SM8150 has 4 tiles: West, East, North & South
    - Tip: Add UFS reset after pins
    - SD pins last

GCC

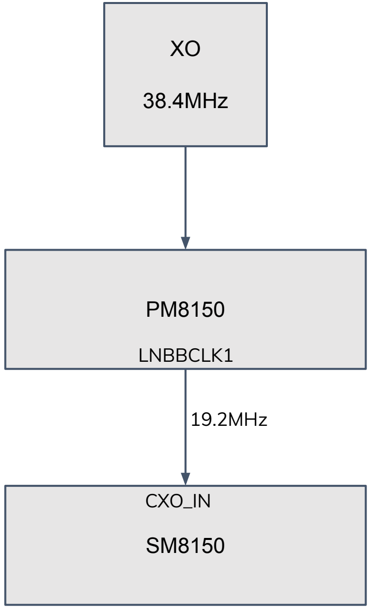
- Add new compatible
- Downstream driver needs tidyup
- Upstream requires parent data scheme
- Describe parent clocks as .parent_data
- Reference to parent clocks, no more arrays of global names!
- External clocks (xo, sleep_clk, rpmcc) described as parents in DT
- Helps resolve namespace issues

Porting GCC

- Port downstream driver with *changes*
  - Parent Data scheme (new one now!)
  - Describe parents in DT
  - Remove downstream VDD fields for clks
  - Move some ops to use upstream ones:
    - Clk_branch2_hw_ctl_ops -> clk_branch_simple_ops
    - clk_gate2_ops -> clk_branch2_ops
  - Many clocks don't have parents
    - Shared clocks, Linux doesn't manage parent

Word on Clocks

- XO generates clock @38.4MHz
- Feeds to PMIC PM8150
- PMIC generates clocks
- RPM configures and controls these (**rpmhcc**)
- LNBBCLK1 aka RPMH_CXO_CLK is xo for SoC



DT Description

```
clocks {
        xo_board: xo-board {
                compatible = "fixed-clock";
                #clock-cells = <0>;
                clock-frequency = <38400000>;
                clock-output-names = "xo_board";
        };

        sleep_clk: sleep-clk {
                compatible = "fixed-clock";
                #clock-cells = <0>;
                clock-frequency = <32764>;
                clock-output-names = "sleep_clk";
        };
 };
```

DT Description

```
rpmhcc: clock-controller {
        compatible = "qcom,sm8150-rpmh-clk";
        #clock-cells = <1>;
        clock-names = "xo";
        clocks = <&xo_board>;
};

gcc: clock-controller@100000 {
        compatible = "qcom,gcc-sm8150";
        reg = <0x0 0x00100000 0x0 0x1f0000>;
        #clock-cells = <1>;
        #reset-cells = <1>;
        #power-domain-cells = <1>;
        clock-names = "bi_tcxo”, "sleep_clk";
        clocks = <&rpmhcc RPMH_CXO_CLK>, <&sleep_clk>;
};
```

RPMHCC

- RPM manages PMIC clock controller (rpmhcc)
- Driver Upstream!
  - drivers/clk/qcom/clk-rpmh.c
- Update the driver for platform
  - Add new compatible
  - Describe rpmh clocks

cmd-db

- Shared mem SoC driver
- Helps find SoC specific identifier and information
- compatible: qcom,cmd-db
- Find from memory map!

Regulators

- Downstream not much reuse
- RPMH controls PMICs
  - Update qcom-rpmh-regulator driver for PMIC
  - drivers/regulator/qcom-rpmh-regulator.c
- Downstream tells “pmic-id”
  - Used to get ‘addr’ from cmd_db
- Describe PMIC supplies, SMPS and LDOs in board DTS
- Need schematics!

SoC Infra

- Upstream!
- Need DT description
    - PMU
        - compatible: arm,armv8-pmuv3
    - PSCI
        - compatible: arm,psci-1.0
    - SMEM
        - compatible: qcom,smem

SoC Infra

- hw_mutex
  - compatible: qcom,tcsr-mutex
- AOSS_QMP
  - Add new platform compatible
  - drivers/soc/qcom/qcom_aoss.c
- Mailbox
  - Add new platform compatible and data
  - drivers/mailbox/qcom-apcs-ipc-mailbox.c
- Apps RSC (Resource State Coordinator)
  - compatible = "qcom,rpmh-rsc

UFS

- UFS Controller Upstream!
    - compatible = "qcom,ufshc
- Describe DT
- UFS ICE (Integrated Crypto Engine) on mailing list atm!

UFS PHY

- Support required for new PHY
- Many PHY drivers!
  - common QMP phy driver for QMP PHYs
    - UFS, USB, PCIe
  - Use sequences from downstream as reference
  - Bit of trial and error!
- Need different sequences for UFS, PCIe, USB

USB

- Controller upstream!
    - compatible: qcom,dwc3
    - Needs child node for core DWC3 IP block
        - compatible: snps,dwc3
    - Supports both SS and HS

USB PHY

- Check phy for SS/HS
- QMP Phy
  - Add sequences for USB
- May need new driver for non QMP Phy
  - Example: SM8150 uses snps phy for hs usb

Upstream Status on SM8150

Component	State
GCC	Upstream
PIN control	Upstream
Regulators	Upstream
DTS	Upstream
RPMHCC	Upstream
Remote procs	Upstream
Cpu freq	Upstream
UFS	Upstream
USB	Phy upstream DT on list

Additional Resources

- Qualcomm BOF at Linaro Connect
- Linaro QC landing team tree

  https://git.linaro.org/landing-teams/working/qualcomm/kernel.git/log/?h=integration-linux-qcomlt
- 96boards https://www.96boards.org/product/rb3-platform/

Thank You

contact@linaro.org

vkoul@kernel.org
@vkoulk