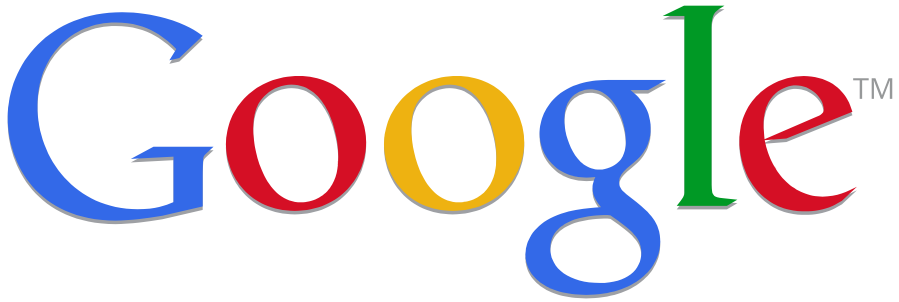

Coreboot Mainboard Porting

Firmware Summit
February 20, 2014

Reference Design Model

- Google develops FW for new reference board
- OEMs base clone designs on reference board
- Ex. Slippy (Haswell) → Falco, Peppy, Leon
- Ex. Rambi (BYT) → Clapper, Glimmer, Squawks

Clone Board Firmware

- Clone board FW derived from reference FW
- Clone FW lives in branch only (not ToT)
- Must be kept up-to-date with ref board fixes

coreboot

- Board-level Coreboot folders
  - .../coreboot/src/mainboard/$VENDOR
  - .../coreboot/src/mainboard/google: CrOS in-dev projects
  - .../coreboot/configs, .../payloads/libpayload/configs
- All board-level code lives here - nowhere else!
- Changes outside board-level go to ToT first

CB Board Folder

- Example: .../coreboot/src/mainboard/google/rambi

acpi / acpi_tables.c

chromeos.c: WP, recovery, etc. switch handling

devicetree.cb: port / peripheral config

gpio.c: GPIO config tables

irqroute.{c,h}: IRQ routing + config

Kconfig: Config flag options

mainboard.c: Board-level callbacks

onboard.h: Peripheral defs

romstage.c: DRAM / SPD strapping

CB Porting - Cloning

- First step: Copy reference board folder
- Copy board config files
- Change strings, ex. s/Rambi/Clapper/
- Commit CL at this point as best practice

Reference CLs:

https://chromium-review.googlesource.com/#/c/176546/

https://chromium-review.googlesource.com/#/c/184089/

CB Porting - GPIOs

- Compare GPIOs in ref. schematic vs clone
  - Pay attention to changed peripherals, control signals, etc
- Modify cloned gpio.{c,h} to reflect clone board
- Reference chipset/soc-level code for GPIO types

Reference CLs:

https://chromium-review.googlesource.com/#/c/56869/

https://chromium-review.googlesource.com/#/c/181034/

CB Porting - SPD Strapping

- No SPD ROM for soldered-down DRAM
- Instead use strap GPIOs to indicate DRAM config
- Read GPIOs and load correct SPD data
- See romstage.c / spd folder

Reference CLs:

https://chromium-review.googlesource.com/#/c/182426/

https://chromium-review.googlesource.com/#/c/184085/

CB Porting - Device Tree

- Specified in devicetree.cb
- Check enabled ports -- I2C, PCIe, SATA, etc
- Also specify custom device / port configs

Reference CLs:

https://chromium-review.googlesource.com/#/c/65717/

https://chromium-review.googlesource.com/#/c/180084/

CB Porting - ACPI

- acpi_tables.c / onboard.h / acpi folder
- Inform OS of I2C devices / config
- Thermal settings - define thermal zones

Reference CLs:

https://chromium-review.googlesource.com/#/c/182366/

https://chromium-review.googlesource.com/#/c/182936/

CB Porting - Misc. Peripherals

- LAN: Set config registers
- Ex. LED behavior, MAC address?
- Codec: Reference codec module datasheet
- Define verb table according to audio config

Reference CLs:

https://chromium-review.googlesource.com/#/c/173581/

https://chromium-review.googlesource.com/#/c/182936/

It’s Firmware Porting Time!

- New board: *urgot*
  - urgot is a rambi-class BYT platform
  - It has NO WLAN card!
  - Two I2C devices: TP (I2C0) and codec (I2C2) (no ALS)
  - No SATA support (uses eMMC)
  - Only one RAM_ID GPIO (2GB / 4GB single/dual channel Micron DDR)

Step 1 - Clone

- Copy rambi/ to urgot/ and change strings
- Copy config files and rename
- Test compile and commit NOW

Actual CL:

https://chromium-review.googlesource.com/187150

Step 2 - GPIOs

- Remove WLAN-related pins
- Disable GPIOs for unused I2C ports
- Remove unneeded extra 2 strap GPIOs

Actual CL:

https://chromium-review.googlesource.com/187161

Step 3 - SPD Strapping

- Modify spd/ folder + Makefile
- Modify romstage.c to use single strap GPIO
- Note dual channel mask

Actual CL:

https://chromium-review.googlesource.com/187164

Step 4 - Device Tree

- Remove SATA-related settings
- Enable proper I2C ports
- Disable WLAN PCIe port

Actual CL:

https://chromium-review.googlesource.com/187173

Step 5 - ACPI

- Remove tables for missing deviecs
- Note the specified I2C ports (base 0 vs base 1)

Actual CL:

https://chromium-review.googlesource.com/#/c/187154/

Step 6 - Ship It!

Any questions?