



ACPI-DT hybrid mode

Hans de Goede

Senior Staff Engineer, Qualcomm Technologies NL BV

Current situation

- Currently as soon as the kernel boots with a populated DT provided then the arch/arm64 code sets `acpi_disabled=1` and the complete ACPI subsystem gets disabled.
- On WoA laptops where the factory Windows OS actually boots using these tables this is not necessarily desirable

Idea / proposal

- In DT mode still parse the ACPI tables and make the ACPI fwnodes available for device-drivers to use.
- By default no devices will be instantiated from ACPI and no other actions will be taken.
- Devicetree can opt-in to have some devices be instantiated from ACPI, e.g. thermal-zone and EC devices

Possible uses

- This populates `/sys/firmware/acpi` and `/sys/bus/acpi` with tables/fwnodes allowing *acpidump* to work and can be used to investigate the ACPI tables on the system.
- Allows using ACPI info from drivers for DT described devices, e.g. MIPI SDCA tables for audio.
- Go truly hybrid and use ACPI to instantiate some of the kernel device objects representing the hardware. For example instantiate I2C clients described in ACPI under an I2C adapter described in DT.
- The dream: have a single DTS for compute SoCs and grab all laptop-model specific information from the ACPI tables

Implementation

- Add a new `acpi_dt_hybrid` ACPI subsys flag to enable this new mode
- ACPI glue changes to support devices with an `of_node` as primary fwnode having a "acpi-path" string property and when present look up the ACPI fwnode for that path, e.g. "`\\_SB.GIO0`", "`\\_SB.I2C9`" and set that fwnode as secondary node.
- I2C and GPIO ACPI integration code changes to:
 - Make ACPI I2C/GPIO lookups work on secondary fw-nodes
 - Install ACPI I2C/GPIO Oregion handlers for secondary fw-nodes
 - Install ACPI GPIO event-handlers on OF described GPIO chips (opt-in)
 - Determine I2C bus-speed for OF described I2C-adapter using ACPI
 - Handle `AttribRawBytes` accessor-type I2C Oregion accesses (upstreamed)

Implementation (2)

- Allow devicetree to opt-in to instantiate ACPI described devices as platform-devices through a new “acpi-hybrid-allowed-hids” property in the chosen DT node.
- Handle ACPI _DEP method for thermal-zone devices
- Make Qualcomm TLMM GPIO driver register “virtual” GPIOs to match how the ACPI tables use out of range numbers on the TLMM to indicate that the GPIO should be handled through the

Testing on T14s gen 6

- Setting the new *“acpi-path”* DT property on the keyboard and touchpad I2C adapter + the main TLMM GPIO chip results in:
 - I2C clients successfully getting instantiated for keyboard and touchpad

```

@@ -816,57 +816,8 @@
816 816     };
817 817
818 818     &i2c0 {
819 -        clock-frequency = <400000>;
820 -
821 -        pinctrl-0 = <&qup_i2c0_data_clk>, <&tpad_default>;
822 -        pinctrl-names = "default";
823 -
819 +        acpi-path = "\\_SB.I2C1";
824 820         status = "okay";
825 -
826 -        /* ELAN06E2 or ELAN06E3 */
827 -        touchpad@15 {
828 -            compatible = "hid-over-i2c";
829 -            reg = <0x15>;
830 -
831 -            hid-descr-addr = <0x1>;
832 -            interrupts-extended = <&tlmm 3 IRQ_TYPE_LEVEL_LOW>;
833 -
834 -            vdd-supply = <&vreg_misc_3p3>;
835 -            vddl-supply = <&vreg_l12b_1p2>;
836 -
837 -            wakeup-source;
838 -        };
839 -
840 -        /* SYNA8022 or SYNA8024 */
841 -        touchpad@2c {
842 -            compatible = "hid-over-i2c";
843 -            reg = <0x2c>;
844 -
845 -            hid-descr-addr = <0x20>;
846 -            interrupts-extended = <&tlmm 3 IRQ_TYPE_LEVEL_LOW>;
847 -

```

```

847 -
848 -     vdd-supply = <&vreg_misc_3p3>;
849 -     vddl-supply = <&vreg_l12b_1p2>;
850 -
851 -     wakeup-source;
852 - };
853 -
854 - /* ELAN06F1 or SYNA06F2 */
855 - keyboard@3a {
856 -     compatible = "hid-over-i2c";
857 -     reg = <0x3a>;
858 -
859 -     hid-descr-addr = <0x1>;
860 -     interrupts-extended = <&tlmm 67 IRQ_TYPE_LEVEL_LOW>;
861 -
862 -     vdd-supply = <&vreg_misc_3p3>;
863 -     vddl-supply = <&vreg_l15b_1p8>;
864 -
865 -     pinctrl-0 = <&kybd_default>;
866 -     pinctrl-names = "default";
867 -
868 -     wakeup-source;
869 - };
870 821 };
871 822
872 823 &i2c3 {
873 824 @@ -1383,6 +1334,8 @@
874 825
875 826 };
876 827
877 828 &tlmm {
878 829 +     acpi-path = "\\_SB.GI00";
879 830 +
880 831
881 832 gpio-reserved-ranges = <34 2>, /* Unused */
882 833 <44 4>, /* SPI (TPM) */
883 834 <72 2>, /* Secure EC I2C connection (?) */

```

More testing on T14s gen 6

- Setting *“acpi-path”* on more I2C adapters, enabling GPIO event handlers and setting *“acpi-hybrid-allowed-hids”* to instantiate thermal-zones, LID device and EC ACPI nodes results in:
 - Working ACPI LID device
 - Working thermal-zones (keyboard, bottom + others)
 - GPIO event handler for EC firing and AML code successfully reading EC events
- Modifying the thinkpad_acpi driver driver to work on aarch64 results in working:
 - hotkeys
 - LEDs (lid_logo_dot, power, speaker-mute, mic-mute)
 - Keyboard backlight support
 - Platform profile support

▼ ...4/boot/dts/qcom/x1e78100-lenovo-thinkpad-t14s.dts



+4



@@ -10,6 +10,10 @@

```
10 10      compatible = "lenovo,thinkpad-t14s-lcd", "lenovo,thinkpad-t14s",
11 11          "qcom,x1e78100", "qcom,x1e80100";
12 12
```

```
13 +      chosen {
14 +          acpi-hybrid-allowed-hids = "LNXTHERM", "PNP0C0D";
15 +      };
16 +
```

```
13 17      backlight: backlight {
14 18          compatible = "pwm-backlight";
15 19          pwms = <&pmk8550_pwm 0 4266537>;
```



880	862	&i2c5 {
881	863	clock-frequency = <400000>;
882	864	
	865	+ acpi-path = "_SB.I2C6";
883	866	status = "okay";
884	867	
885	868	eusb5_repeater: redriver@43 {
***** ↓ ↑ *****		
		@@ -927,20 +910,8 @@
927	910	
928	911	&i2c6 {
929	912	clock-frequency = <400000>;
930		-
	913	+ acpi-path = "_SB.I2C7";
931	914	status = "okay";

```
1335 1306
1336 1307     &tlmm {
1337 1308         acpi-path = "\\_SB.GI00";
1309 +     acpi-enable-gpio-events;
1338 1310
1339 1311         gpio-reserved-ranges = <34 2>, /* Unused */
1340 1312             <44 4>, /* SPI (TPM) */
1341 1313             <72 2>, /* Secure EC I2C connection (?) */
1342 1314             <238 1>; /* UFS Reset */
1343 1315
```

```

hans@t14s:~$ ls -d /sys/bus/i2c/devices/i2c-[A-Z]*
/sys/bus/i2c/devices/i2c-ELAN06F1:00 /sys/bus/i2c/devices/i2c-SYNA8022:00
hans@t14s:~$ cat /sys/bus/acpi/devices/ELAN06E?:00/status /sys/bus/acpi/devices/SYNA8022\:
00/status
0
0
15
hans@t14s:~$ cat /sys/class/hwmon/hwmon0/name /sys/class/hwmon/hwmon0/temp?_input
acpitz
39000
37000
38000
38000
37000
28000
hans@t14s:~$ ls /sys/class/leds/
input1::capslock  input1::numlock  platform::mute  tpacpi::power
input1::compose  input1::scrolllock  tpacpi::kbd_backlight  tpacpi::standby
input1::kana      platform::micmute  tpacpi::lid_logo_dot  tpacpi::thinkvantage
hans@t14s:~$ ls /sys/bus/platform/devices/thinkpad_acpi/
doubletap_enable  hotkey_enable  platform-profile
driver            hotkey_mask   power
driver_override  hotkey_recommended_mask  subsystem
dytc_lapmode     hotkey_tablet_mode  uevent
hotkey_adaptive_all_mask  input  wakeup_hotunplug_complete
hotkey_all_mask  keyboard_lang  wakeup_reason

```

Questions / discussion point

- Demo: <https://github.com/jwrdegoede/linux-sunxi/commits/acpi-dt-hybrid-v0.4/>
- Is this hybrid mode a good idea, something to continue working on?
- Next steps

Thank you !

Děkuji!

Dank je wel !

Nothing in these materials is an offer to sell any of the components or devices referenced herein.

© Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved.

Qualcomm and Snapdragon are trademarks or registered trademarks of Qualcomm Incorporated. Other products and brand names may be trademarks or registered trademarks of their respective owners.

References in this presentation to “Qualcomm” may mean Qualcomm Incorporated, Qualcomm Technologies, Inc., and/or other subsidiaries or business units within the Qualcomm corporate structure, as applicable. Qualcomm Incorporated includes our licensing business, QTL, and the vast majority of our patent portfolio. Qualcomm Technologies, Inc., a subsidiary of Qualcomm Incorporated, operates, along with its subsidiaries, substantially all of our engineering, research and development functions, and substantially all of our products and services businesses, including our QCT semiconductor business.

Snapdragon and Qualcomm branded products are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm patents are licensed by Qualcomm Incorporated.