Reversing from the other end: hardware debugging for fun and no profit

Kamilla Magomedova, Andrey Korolyov

October 28, 2017

porting coreboot to the new board

- northbridge/southbridge support already exists
- southbridge variant is supported (bcm5785 vs bcm5780+bcm5785)
- look on the boot log...

common corner cases

- unsupported superio chip/non-common superio
- clock generator requires setup from firmware
- peripheral control wired to the EC
- something standard does not work as expected
  (PIT 0x61 ch 2 on H8SSL, buzz always on)

trying to fix things with peripherals

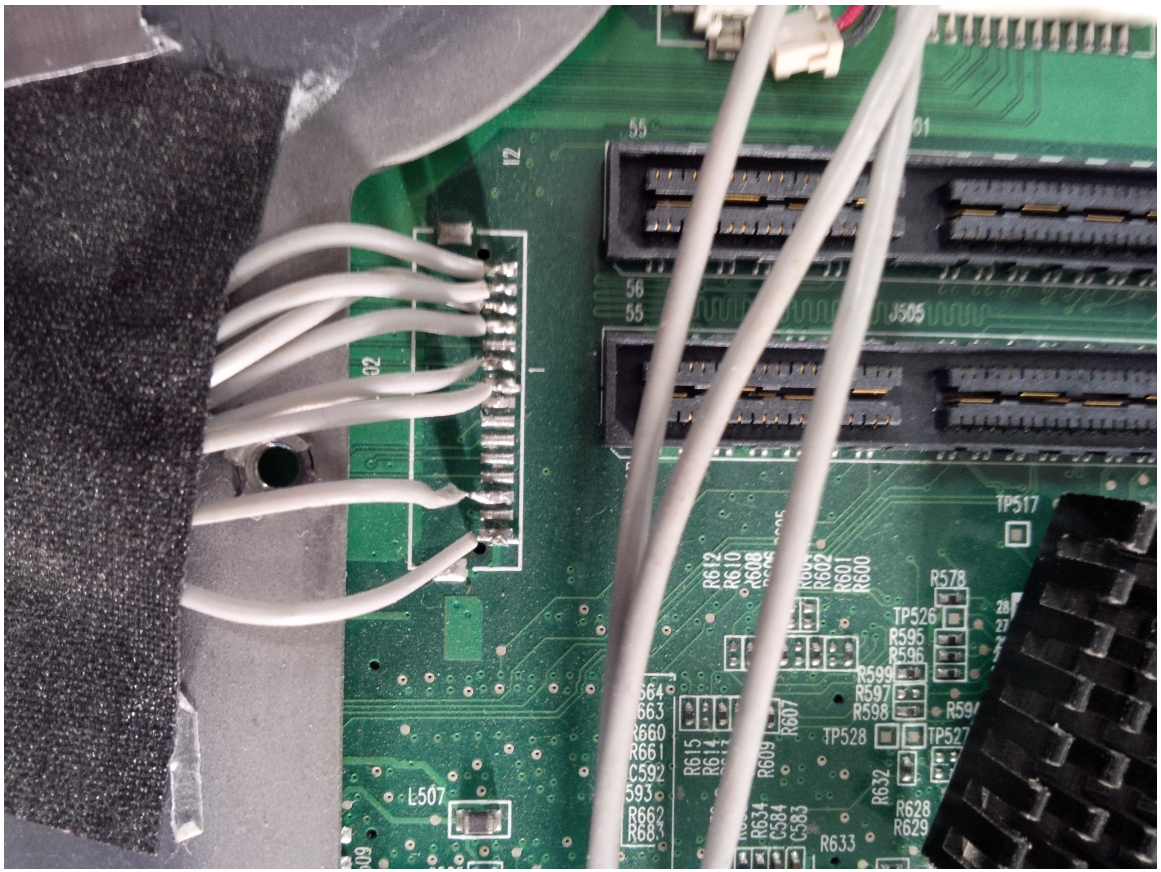
- EC always sits on LPC, just sniff it for f/w interaction
- LPC pins are not always populated separately
- keyboard is wired to EC directly == some work
- i2c peripherals or southbridge gpio settings

getac A790

- superio with two chips (same as in P470)
- actions from special keys and display backlight are wired to EC
- undescribed g-sensor - LIS3L02DQ
- custom/mil laptops bearing some unique stuff with each model

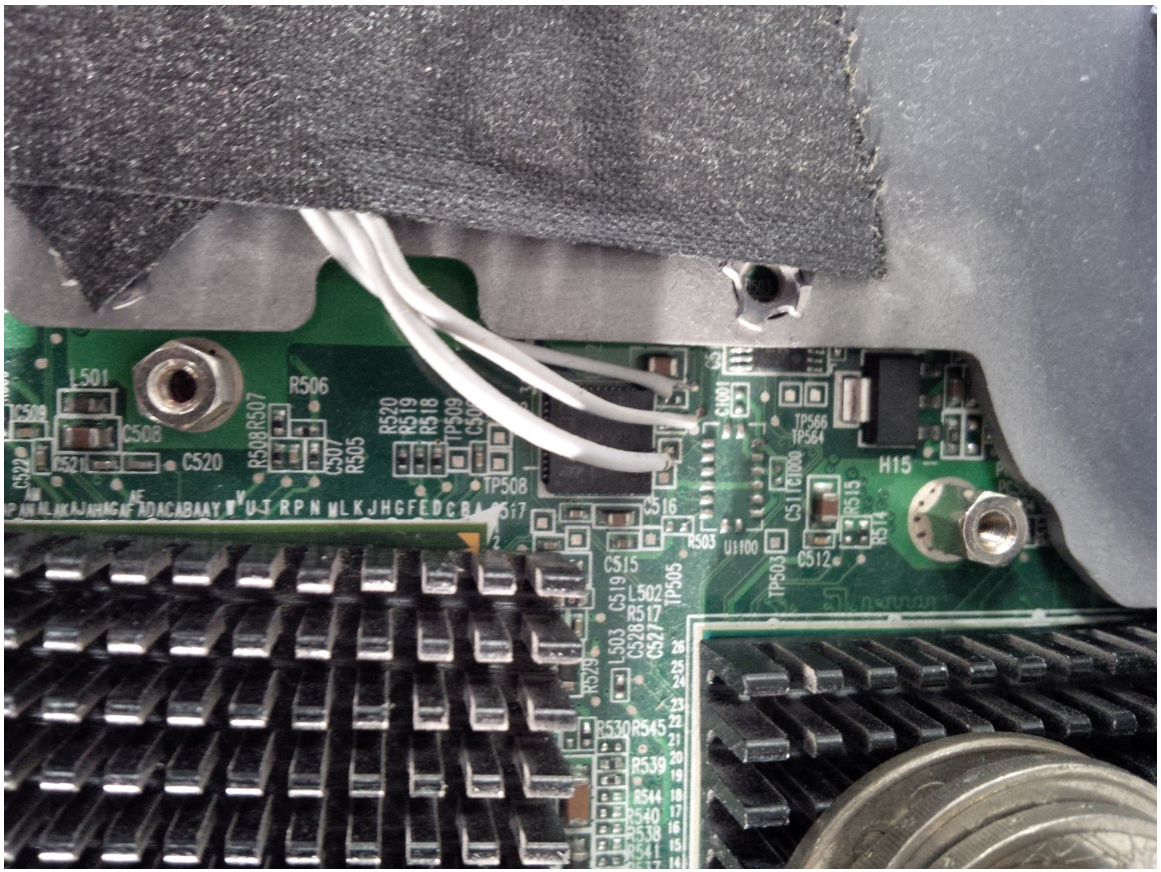
getac A790, acquiring EC interaction bits

umm. . . almost anything except RST/CLK, take them from PCI bus



getac A790, probing for g-sensor

sparse/nonconditional signals like i2c exchange from g-sensor are difficult to trace



looking for pins across the board

- check endpoint connections with multimeter
- tap in logic analyzer, run its output via awk and feed to protocol dissector
  sigrok is very good on parsing most stuff
- most signals are relatively slow but
  50+ Mhz resolution on LA will never harm

study and repeat

- while moving out from vendor firmware, most peripheral initialization sequences need not to be modified
- two-way interaction on a busy bus could be problematic to follow (memory depth limitations on LA)

programmatic methods could be unreliable

clock generator initialization on x60:

```
2e f7 3c 20 01 00 1b 01 54 ff ff 07
```

programmatic methods could be unreliable

clock generator initialization on x60:

`2e f7 3c 20 01 00 1b 01 54 ff ff 07`

z61t: cut down to 8 bytes and reading 0x69 register back:

`6d ff ff 20 41 7f 1b 01`

programmatic methods could be unreliable

clock generator initialization on x60:

```
2e f7 3c 20 01 00 1b 01 54 ff ff 07
```

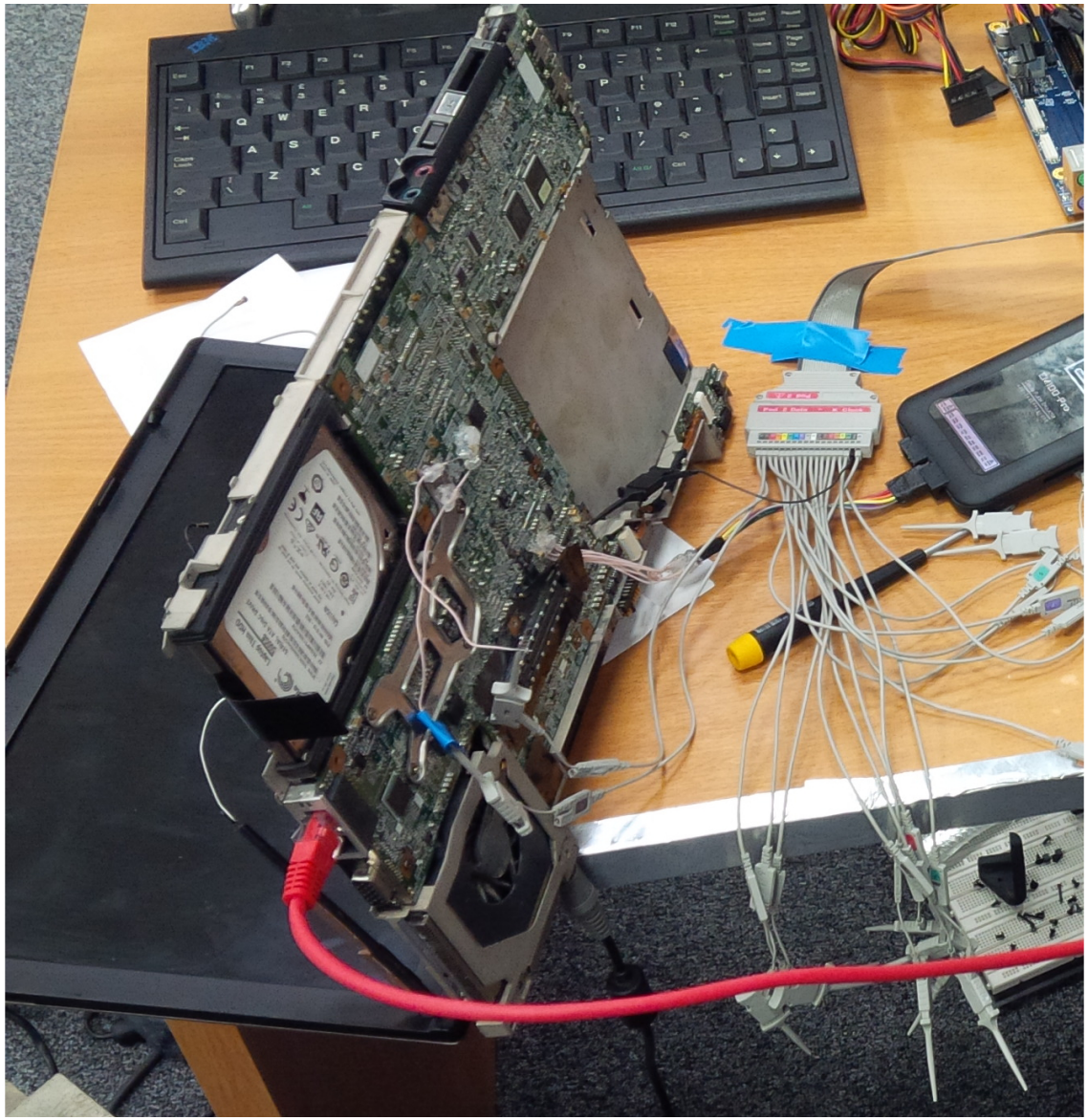
z61t: cut down to 8 bytes and reading 0x69 register back:

```
6d ff ff 20 41 7f 1b 01
```

z61t: (vendor) setup which actually works:

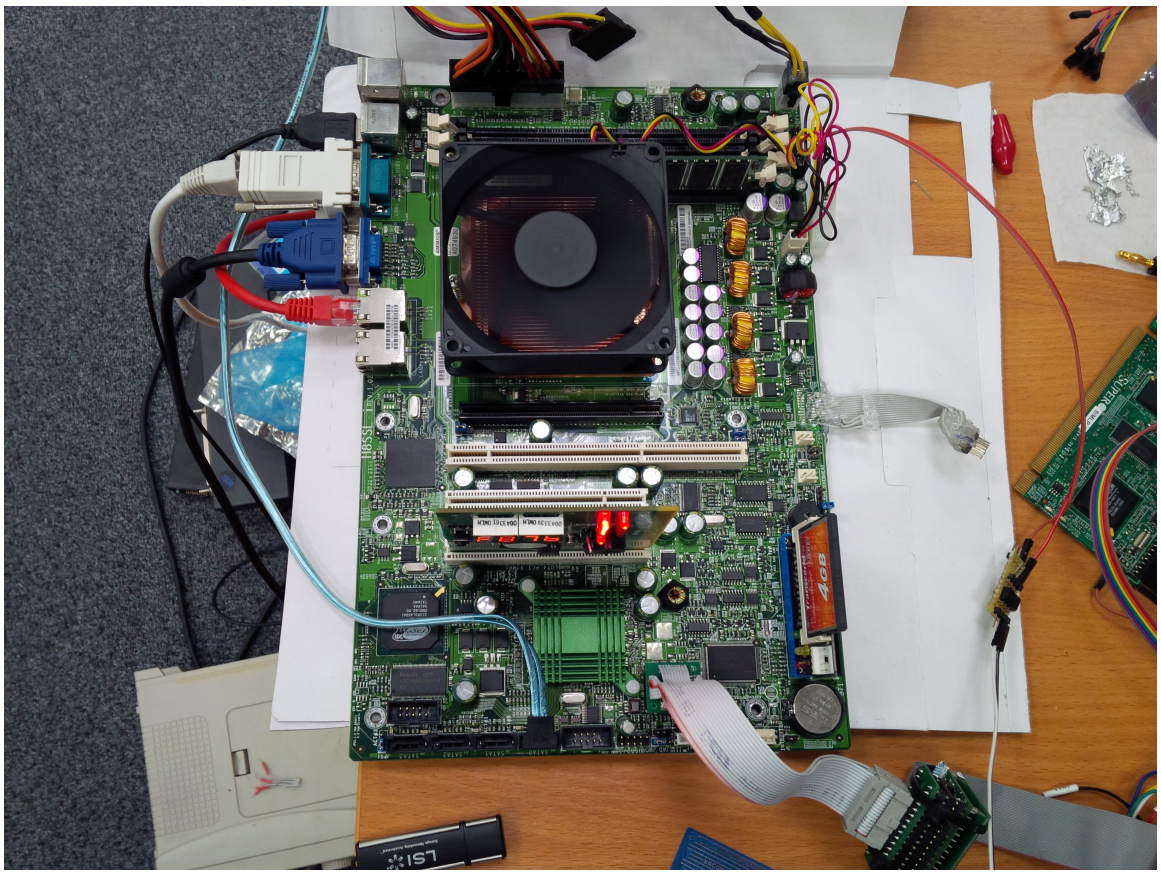
```
6d ff ff 20 41 7f 18 00
```

spend few hours to set correctly few bytes



desktop: standard layout, easy to work with

preparing to play with HDT on H8-SSL, coreboot inside



that‘s all

Thanks!