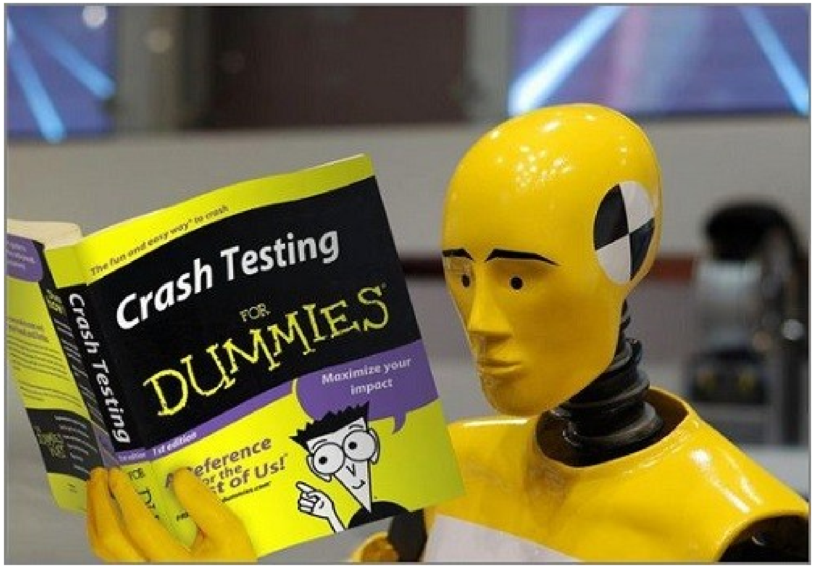
Coreboot for Dummies

By Youness Alaoui



This adventure has been sponsored by :

Purism

What are we gonna talk about?

- Who am I ?
- Getting started with coreboot!
- Getting an existing port to build and work
- Testing and finishing the Librem 13 v1 port
- Starting a new port from scratch
- Debug output, how hard can it be?
- Summary of doing a port
- It's question time!

Who am I ?

- Youness Alaoui, a.k.a **KaKaRoTo**
- aMSN developper
- libnice, Farstream, GStreamer, Meego
- PS3 reverse-engineer
- Freelance consultant
- Most importantly: a coreboot newbie

Getting started with coreboot!

- What is coreboot? How does it work ?
- Looking at the entry point… Bad ideas
- The Three Stooges
- Lack of documentation
- Excessive documentation
- Getting started tutorial

Purism

How to brick a laptop quickly and painlessly!

- Just kidding, it will be painful to the laptop.
- At first glance, most wiki information is about desktop motherboards
- Backup the rom before doing

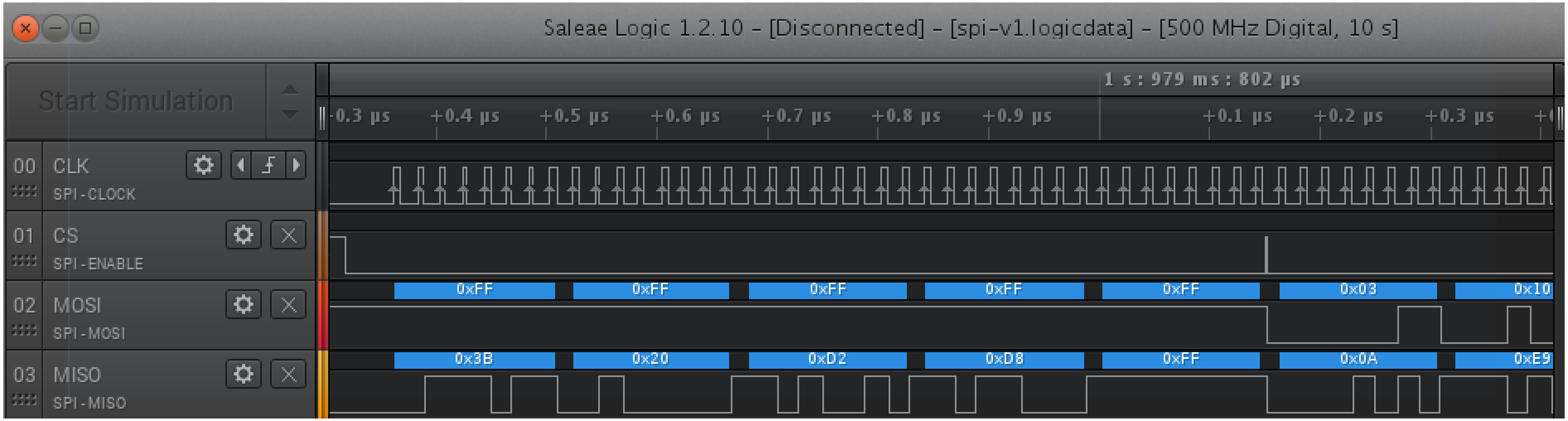
  anything else!
- Don’t solder to the motherboard!

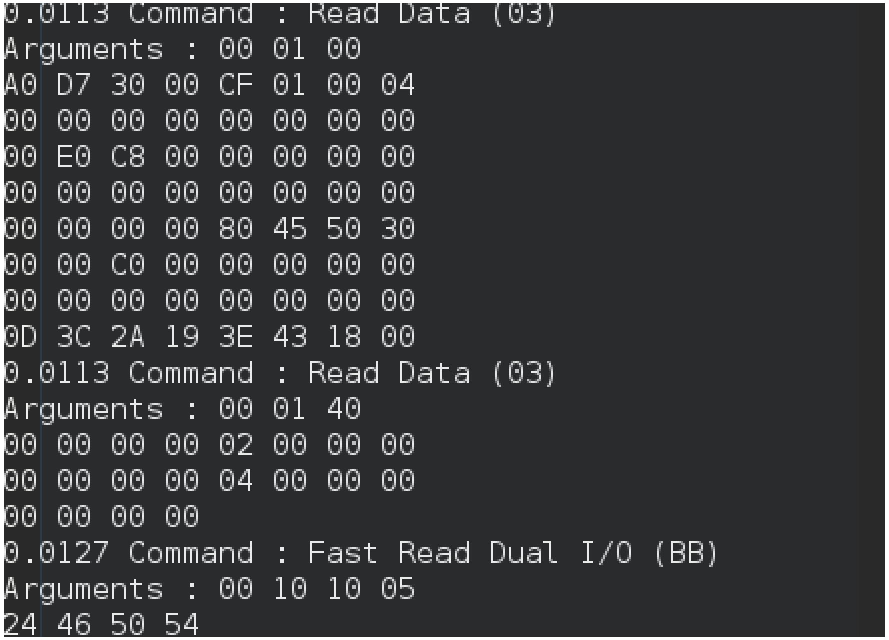
Purism

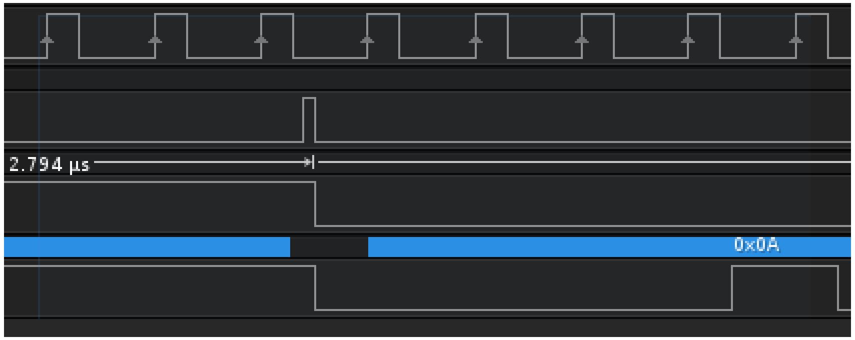
Dumping the flash on v2 hardware

- Used a Logic Analyzer
- Dump trace data into CSV
- Script to analyze SPI commands and reconstitute the image from reads
- Realize the image is corrupted
- Adjust for <2ns spikes to ignore cross talk
- Give up

Dumping the flash on v2 hardware

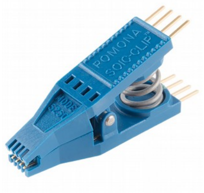



```
0.0113 Command : Read Data (03)
Arguments : 00 01 00
A0 D7 30 00 CF 01 00 04
00 00 00 00 00 00 00 00
00 E0 C8 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 80 45 50 30
00 00 C0 00 00 00 00 00
00 00 00 00 00 00 00 00
0D 3C 2A 19 3E 43 18 00
0.0113 Command : Read Data (03)
Arguments : 00 01 40
00 00 00 00 02 00 00 00
00 00 00 00 04 00 00 00
00 00 00 00
0.0127 Command : Fast Read Dual I/O (BB)
Arguments : 00 10 10 05
24 46 50 54
```



Dumping the flash on v1 hardware

- Don't trust AFULNX, AFUDOS, AFUWIN
- Flashrom to the rescue!
- Laptops and EC
- Use a SOIC clip, instead of a chip socket
- Understanding the Intel Flash Descriptor
- Powering the flash chip and hardware magical nonsense



First coreboot build

- The first build tutorial is an excellent start
- Missing microcode
- Missing blobs and descriptors

Binary blobs, gotta catch them all!

- How to dump the VGA Bios properly
- Where to get the MRC.bin file ?
- What about the refcode.bin ?
- IFD Descriptor and ME binaries

Purism

The importance of debugging

- Getting USB debug to work was easy, thank you!
- With no debug output, you can’t properly do a port.
- M.2 issues and IOBP registers

Getting the bootsplash to work

- SeaBIOS and coreboot fail to decode bootsplash.jpg, coreboot silently…
- Coreboot options to display bootsplash is not intuitive
- Only YCBR:22:11:11 colorspace is supported
- Sequential DCT (non-Progressive) only

Purism

Interference from AMI

- Poweroff vs reboot
- NVRam storage from AMI
- Only mentioned in *Infrastructure Projects[1]* page on the wiki

- [1] https://www.coreboot.org/Infrastructure_Projects#Handle_default_boot_firmware_settings_saving_at_shutdown

To NVMe or not to NVMe

- NVMe drives are PCIe, not SATA
  - Proper PCIe root port needs to be enabled
- SeaBIOS doesn’t support NVMe drives (not really)
- Issues with D3 power state
- AMI doesn’t handle NVMe drives that well either

Starting a new port from scratch

- Very scary, but it’s actually quite straightforward
- As long as it’s to a platform that already has boards ported to it
- Not much documentation about the process
- ACPI and ASL can be daunting to newbies

Purism

Flashrom, GPIOs, configs

- First step: Dump existing ROM
- Second step: Dump your GPIO configuration
- No Skylake support in flashrom or inteltool
- Thank you Nico Huber for doing the Skylake work!
- GPIO data+datasheet+code = easy

Debugging is really important

- Without debugging, you will be stuck after your first (inevitably failing) first test.
- Skylake has no USB/EHCI debug capabilities
- Without UART, you’re stuck
- Introducing flashconsole!
- NOR flash can only write 0s
- Erasing an empty sector is a bad idea

Debugging is really important (continued)

- Romstage and global variables
- Use car_sync_var when using CAR pointers
- SPI drivers, SPI controller, Fast SPI
- No C env for bootblock on broadwell
- Thank you Matt DeVillier and Aaron Durbin for making it happen!

Purism

Memory init, devicetree configs

- FspTempRamInit requires microcode (bad FSP documentation)
- Static SPD data didn't work… and get_spd_smbus requires DIMM_MAX
- FSP binaries and versions (FspUpdVpd.h)
- VBT binary is required for FSP
- Select SERIRQ_CONTINUOUS_MODE
- ACPI is a horrible mess

Purism

Conclusion

- A port is relatively easy, but daunting :
  - Copy an existing directory and rename board
  - Configure GPIO
  - Configure Memory init
  - Configure FSP/devicetree
  - Ignore ACPI and close your eyes
  - Test and fix features (audio, PCI devices, USB, etc..)
  - Invest in a stress ball
- Checkout my more detailed, technical blog posts about this adventure on Purism website (all listed in the side bar of http://puri.sm/coreboot/timeline)

Questions ?

Purism