Porting coreboot to the HP ProLiant MicroServer Gen8 (Part 1)

Alexander Couzens , Felix Held

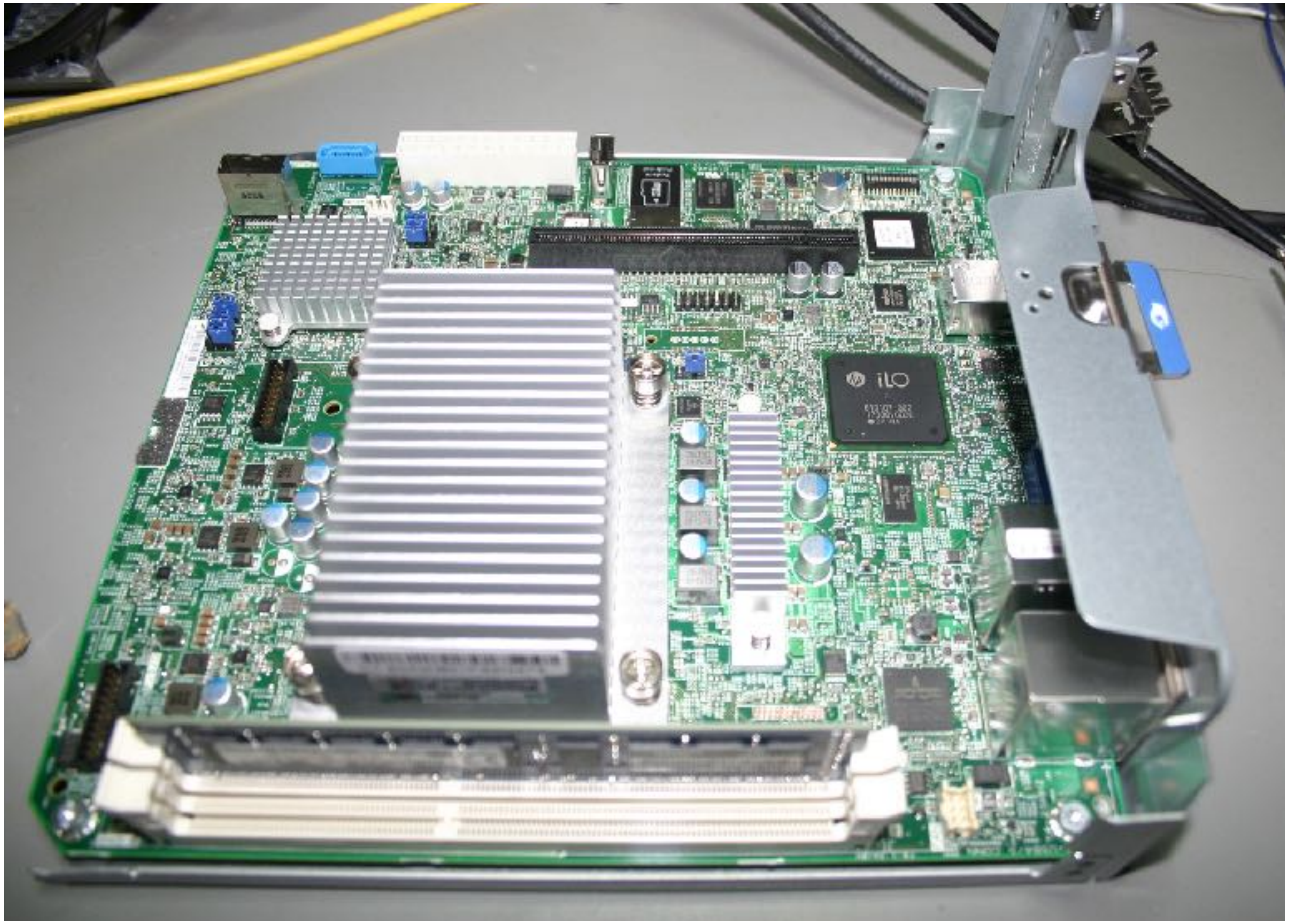
Why this platform?

- good value (~200 Euro for system including mainboard, dual core Ivy Bridge CPU, power supply, chassis with 4 3,5” HDD trays, 4GB RAM included, one RAM slot still empty)
- nice form factor
- ECC RAM
- native RAM init exists for this generation

Gathering information

- Have a look at the board
- Search for board schematics
- lspci, lsusb, …
- Dump all flash/EEPROM chip contents
- Finding out pinouts of possible debug connectors

Board overview



Board overview

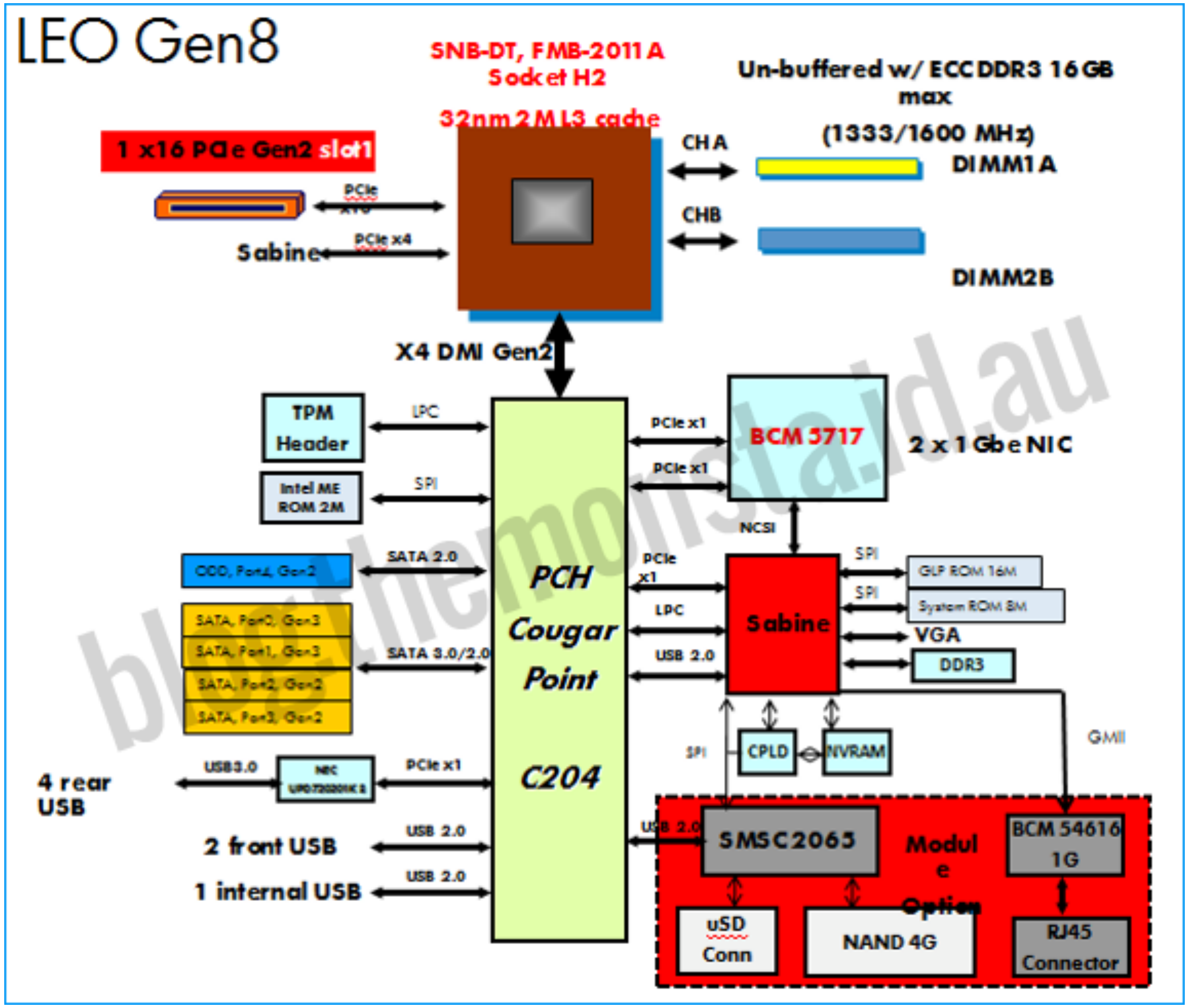
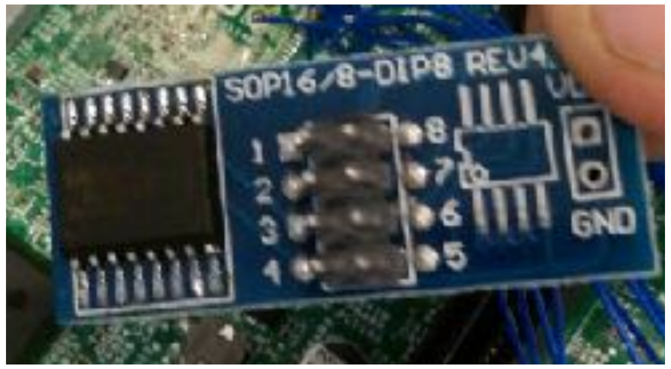
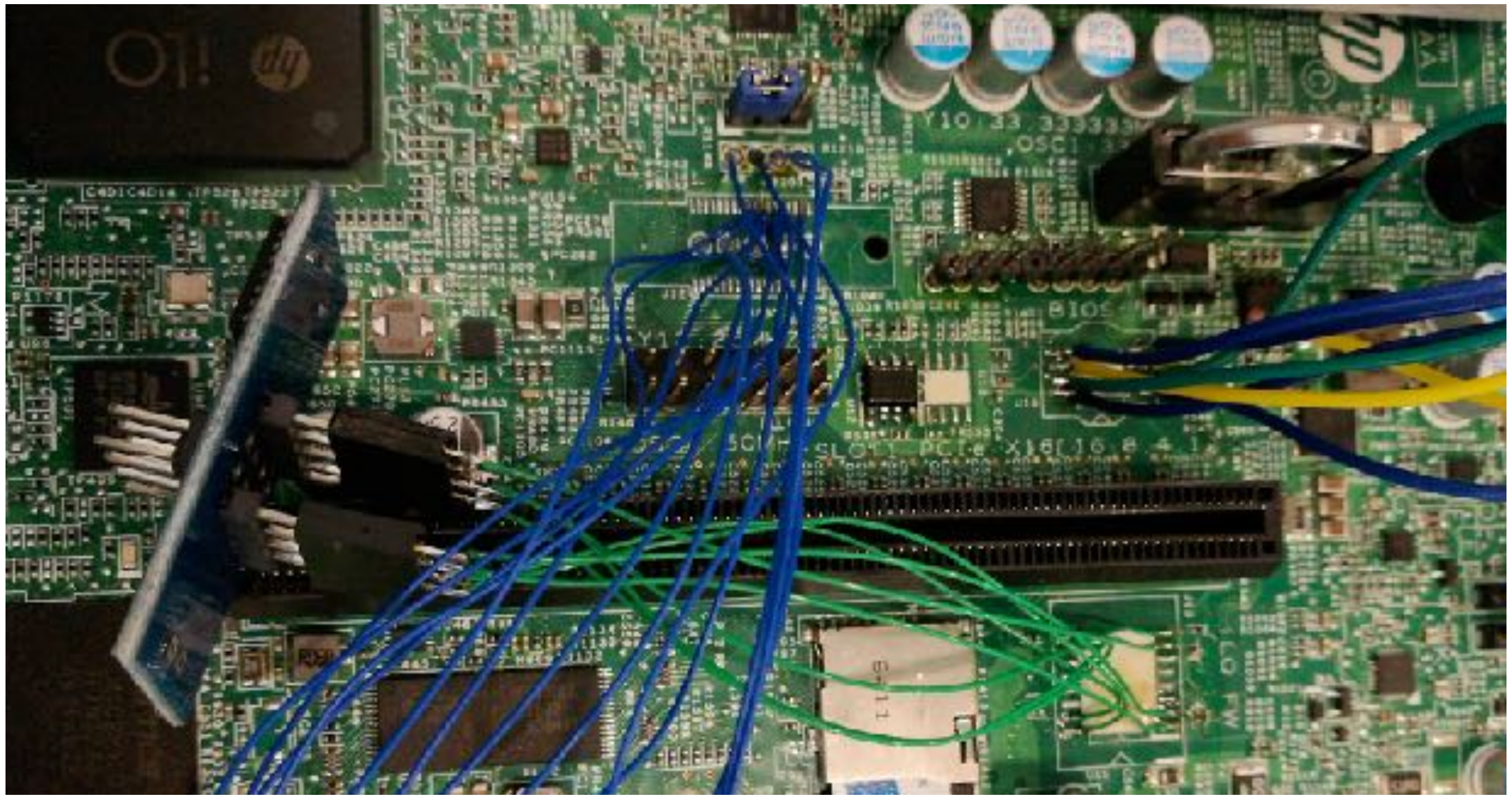


diagram source: http://blog.themonsta.id.au/wp-content/uploads/2013/06/G8_Microserver_OriginalSystemDesign.png
(the diagram seems to be from an early development version, but is quite similar to the real product)

Socketing flash chip





- cheap solution
- does not interfere with the other chips
- writing flash chip is easy

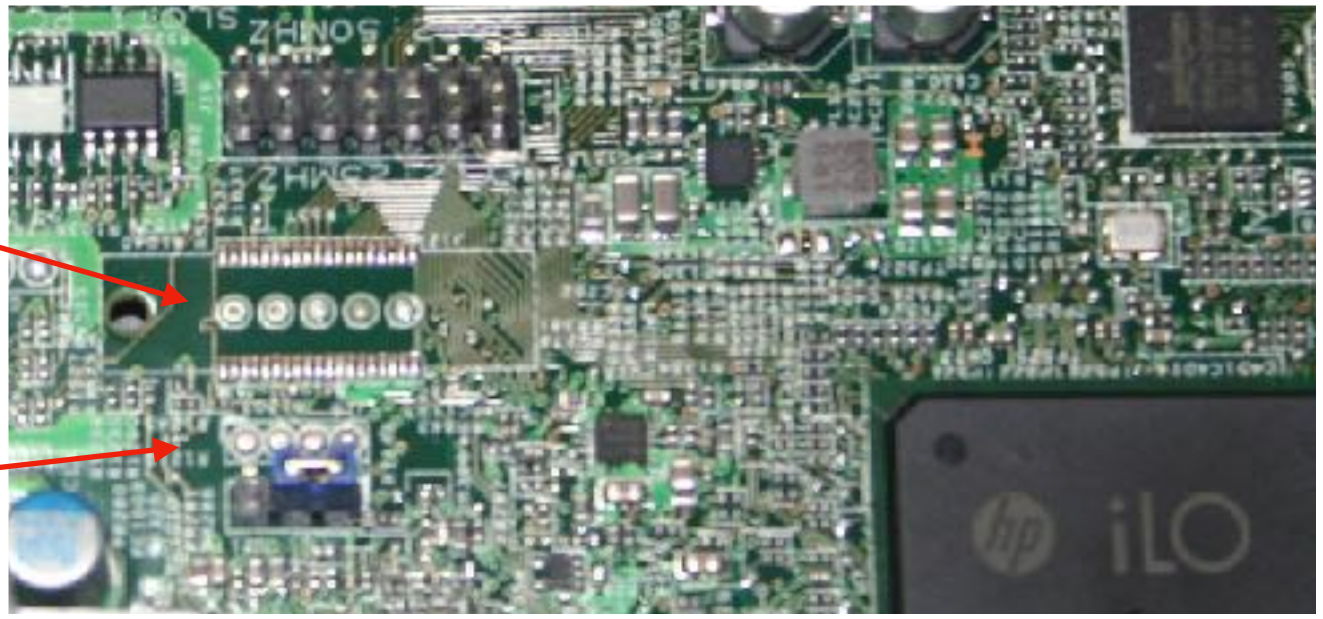
Flash/EEPROM chips

- 16 MB ILO (SOIC16)
- 8 MB BIOS
- 2 MB IFD & ME
- Broadcom 256 kbytes firmware + config
- U12: 256 bytes: serial number, iLO username & password, asset tag
- U74: 8192 bytes: mac address (6 bytes used)
- U75: 256 bytes serial number again (?)
- eMMC

iLO4 JTAG + serial

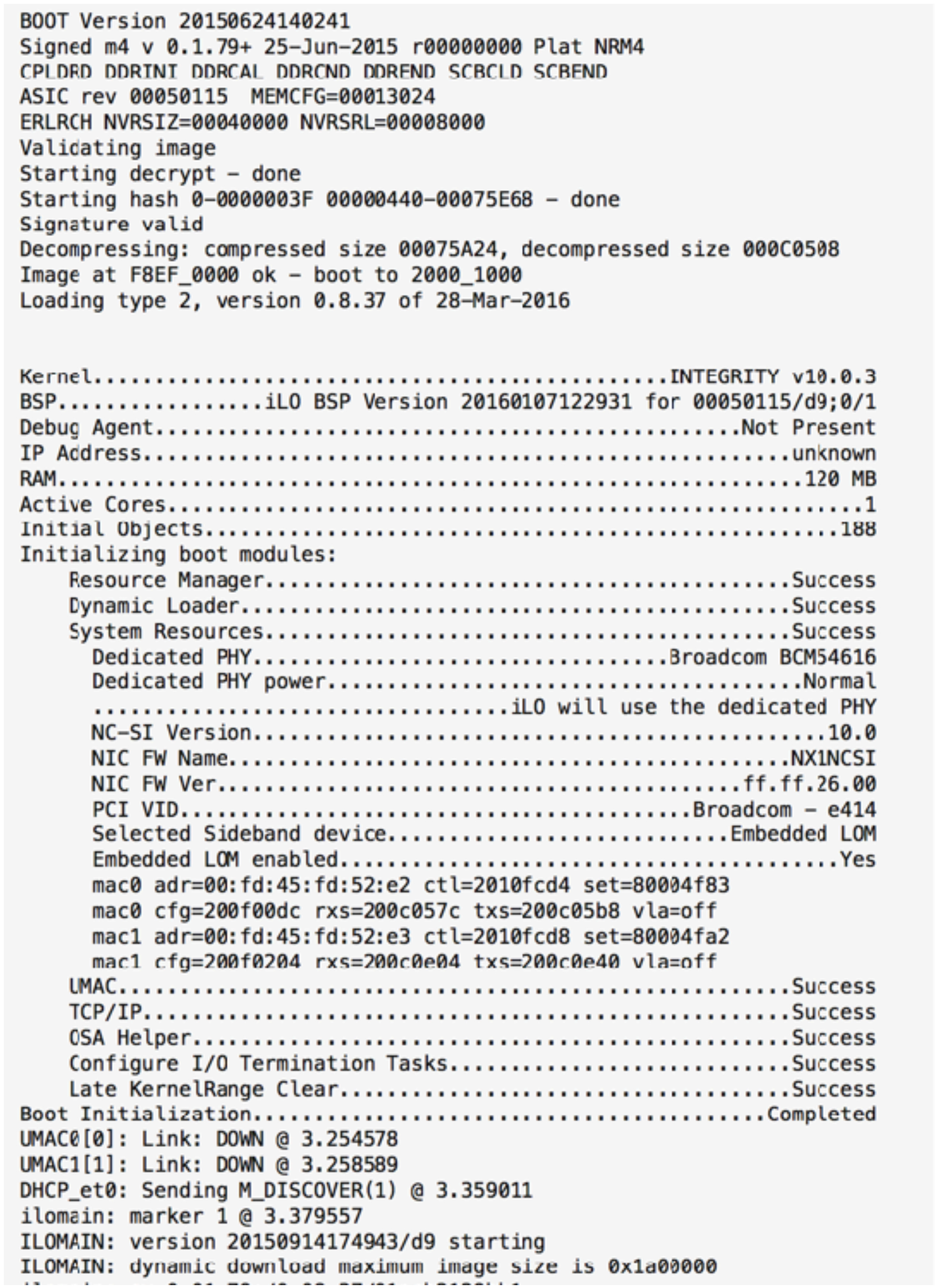
38 pin MICTOR
ARM ETM connector
(JTAG + trace)

iLO serial port



iLO4 serial port

- use multimeter to find ground pin
- use logic analyzer to find the TX pin
- Connect USB-UART chip



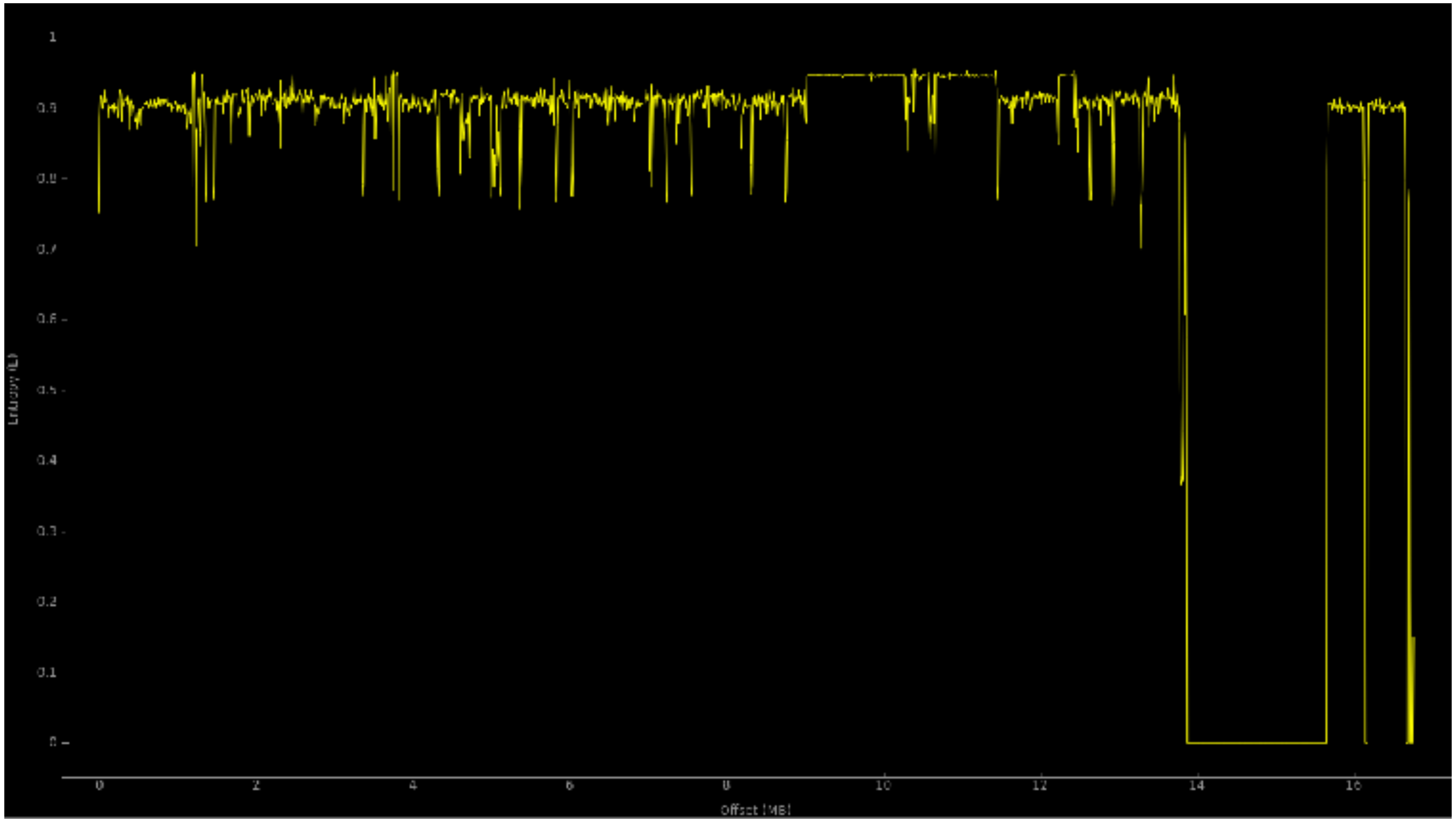
```
BOOT Version 20150624140241
Signed m4 v 0.1.79+ 25-Jun-2015 r00000000 Plat NRM4
CPLDRD DDRINI DDRCAL DDRCND DDREND SCBCLD SCBEND
ASIC rev 00050115  MEMCFG=00013024
ERLRCH NVRSIZ=00040000 NVRSRL=00008000
Validating image
Starting decrypt - done
Starting hash 0-0000003F 00000440-00075E68 - done
Signature valid
Decompressing: compressed size 00075A24, decompressed size 000C0508
Image at F8EF_0000 ok - boot to 2000_1000
Loading type 2, version 0.8.37 of 28-Mar-2016


Kernel..............................................INTEGRITY v10.0.3
BSP................iLO BSP Version 20160107122931 for 00050115/d9;0/1
Debug Agent.................................................Not Present
IP Address......................................................unknown
RAM..............................................................120 MB
Active Cores..........................................................1
Initial Objects.....................................................188
Initializing boot modules:
    Resource Manager............................................Success
    Dynamic Loader..............................................Success
    System Resources............................................Success
      Dedicated PHY.................................Broadcom BCM54616
      Dedicated PHY power..........................................Normal
      ................................iLO will use the dedicated PHY
      NC-SI Version..................................................10.0
      NIC FW Name...............................................NX1NCSI
      NIC FW Ver..............................................ff.ff.26.00
      PCI VID......................................Broadcom - e414
      Selected Sideband device............................Embedded LOM
      Embedded LOM enabled..............................................Yes
      mac0 adr=00:fd:45:fd:52:e2 ctl=2010fcd4 set=80004f83
      mac0 cfg=200f00dc rxs=200c057c txs=200c05b8 vla=off
      mac1 adr=00:fd:45:fd:52:e3 ctl=2010fcd8 set=80004fa2
      mac1 cfg=200f0204 rxs=200c0e04 txs=200c0e40 vla=off
    UMAC........................................................Success
    TCP/IP......................................................Success
    OSA Helper..................................................Success
    Configure I/O Termination Tasks.............................Success
    Late KernelRange Clear......................................Success
Boot Initialization...........................................Completed
UMAC0[0]: Link: DOWN @ 3.254578
UMAC1[1]: Link: DOWN @ 3.258589
DHCP_et0: Sending M_DISCOVER(1) @ 3.359011
ilomain: marker 1 @ 3.379557
ILOMAIN: version 20150914174943/d9 starting
ILOMAIN: dynamic download maximum image size is 0x1a00000
```

iLO4 JTAG

- openOCD
- Info : JTAG tap: auto0.tap tap/device found: 0x07926021 (mfg: 0x010 (NEC), part: 0x7926, ver: 0x0)
    - iLO4 chip (probably a custom multi-die package)
- Info : JTAG tap: auto1.tap tap/device found: 0x0128d043 (mfg: 0x021 (Lattice Semi.), part: 0x128d, ver: 0x0)
    - Lattice LCMXO2280C (FPGA)

iLO4 flash: binwalk -E

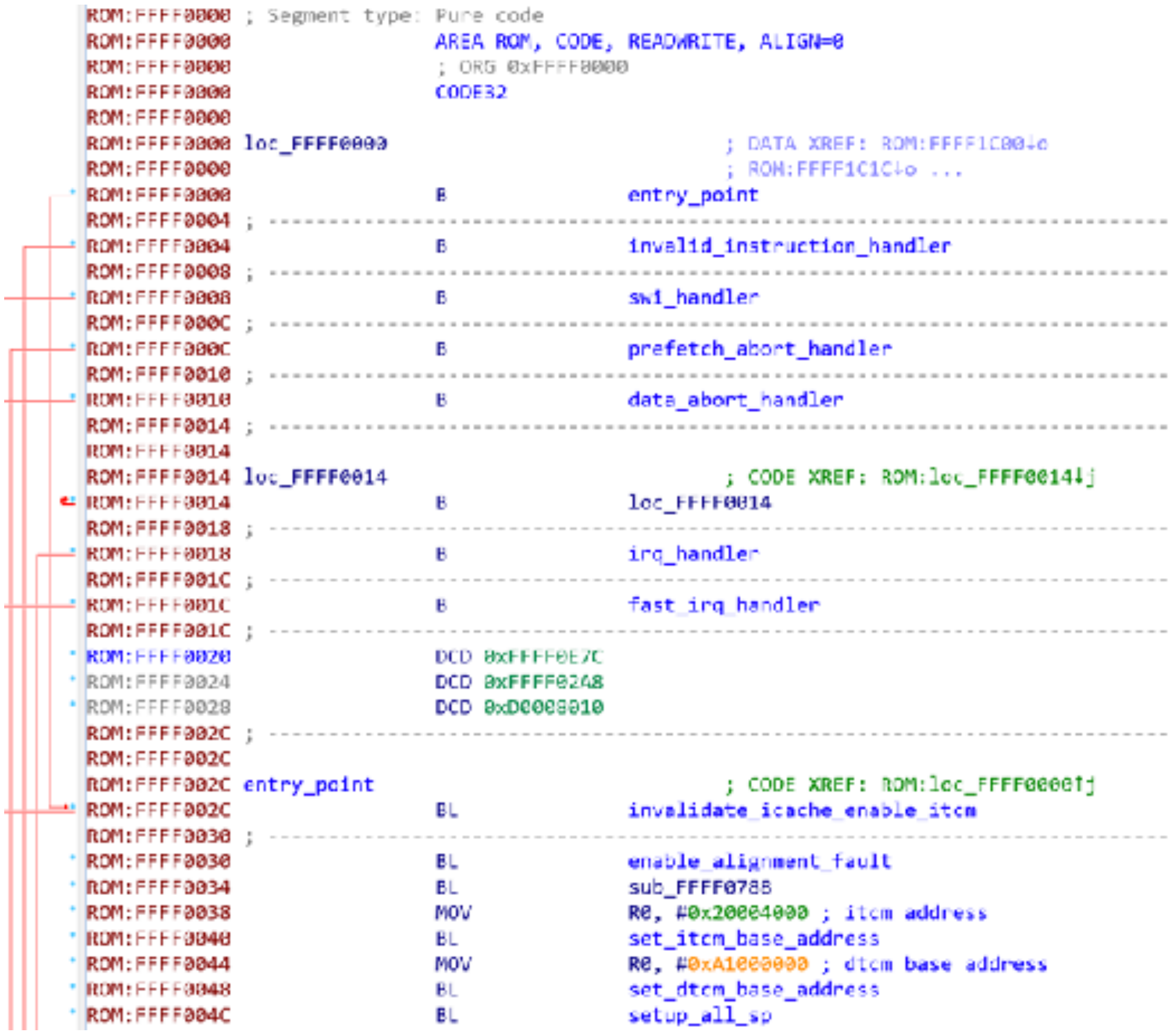


- different sections: normal code, compressed, scrambled/ encrypted, empty

iLO4 - Firmware

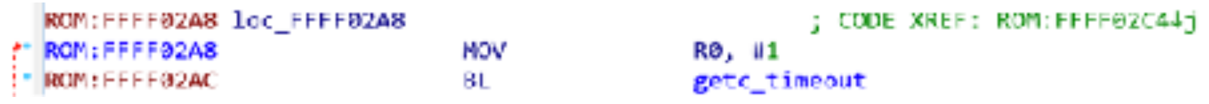
- 16MB SPI flash
- Flash image consists of 2 parts:
  - Flash boot loader (last 64kB of the flash)
  - Main firmware sections
- ARM9 (or newer)
- Reset vector either at 0x00000000 or 0xFFFF0000

iLO4 - Firmware

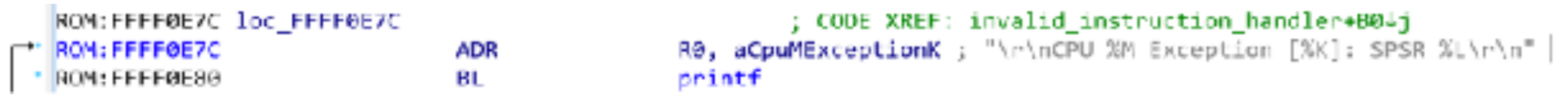


```
ROM:FFFF0000 ; Segment type: Pure code
ROM:FFFF0000                 AREA ROM, CODE, READWRITE, ALIGN=0
ROM:FFFF0000                 ; ORG 0xFFFF0000
ROM:FFFF0000                 CODE32
ROM:FFFF0000
ROM:FFFF0000 loc_FFFF0000                            ; DATA XREF: ROM:FFFF1C80↓o
ROM:FFFF0000                                         ; ROM:FFFF1C1C↓o ...
ROM:FFFF0000                 B               entry_point
ROM:FFFF0004 ; ---------------------------------------------------------------
ROM:FFFF0004                 B               invalid_instruction_handler
ROM:FFFF0008 ; ---------------------------------------------------------------
ROM:FFFF0008                 B               swi_handler
ROM:FFFF000C ; ---------------------------------------------------------------
ROM:FFFF000C                 B               prefetch_abort_handler
ROM:FFFF0010 ; ---------------------------------------------------------------
ROM:FFFF0010                 B               data_abort_handler
ROM:FFFF0014 ; ---------------------------------------------------------------
ROM:FFFF0014
ROM:FFFF0014 loc_FFFF0014                            ; CODE XREF: ROM:loc_FFFF0014↓j
ROM:FFFF0014                 B               loc_FFFF0014
ROM:FFFF0018 ; ---------------------------------------------------------------
ROM:FFFF0018                 B               irq_handler
ROM:FFFF001C ; ---------------------------------------------------------------
ROM:FFFF001C                 B               fast_irq_handler
ROM:FFFF001C ; ---------------------------------------------------------------
ROM:FFFF0020                 DCD 0xFFFF0E7C
ROM:FFFF0024                 DCD 0xFFFF0248
ROM:FFFF0028                 DCD 0xD0008010
ROM:FFFF002C ; ---------------------------------------------------------------
ROM:FFFF002C
ROM:FFFF002C entry_point                             ; CODE XREF: ROM:loc_FFFF0000↑j
ROM:FFFF002C                 BL              invalidate_icache_enable_itcm
ROM:FFFF0030 ; ---------------------------------------------------------------
ROM:FFFF0030                 BL              enable_alignment_fault
ROM:FFFF0034                 BL              sub_FFFF0788
ROM:FFFF0038                 MOV             R0, #0x20004000 ; itcm address
ROM:FFFF0040                 BL              set_itcm_base_address
ROM:FFFF0044                 MOV             R0, #0xA1000000 ; dtcm base address
ROM:FFFF0048                 BL              set_dtcm_base_address
ROM:FFFF004C                 BL              setup_all_sp
```

- vector table
- Reset vector either at 0x00000000 or **0xFFFF0000**

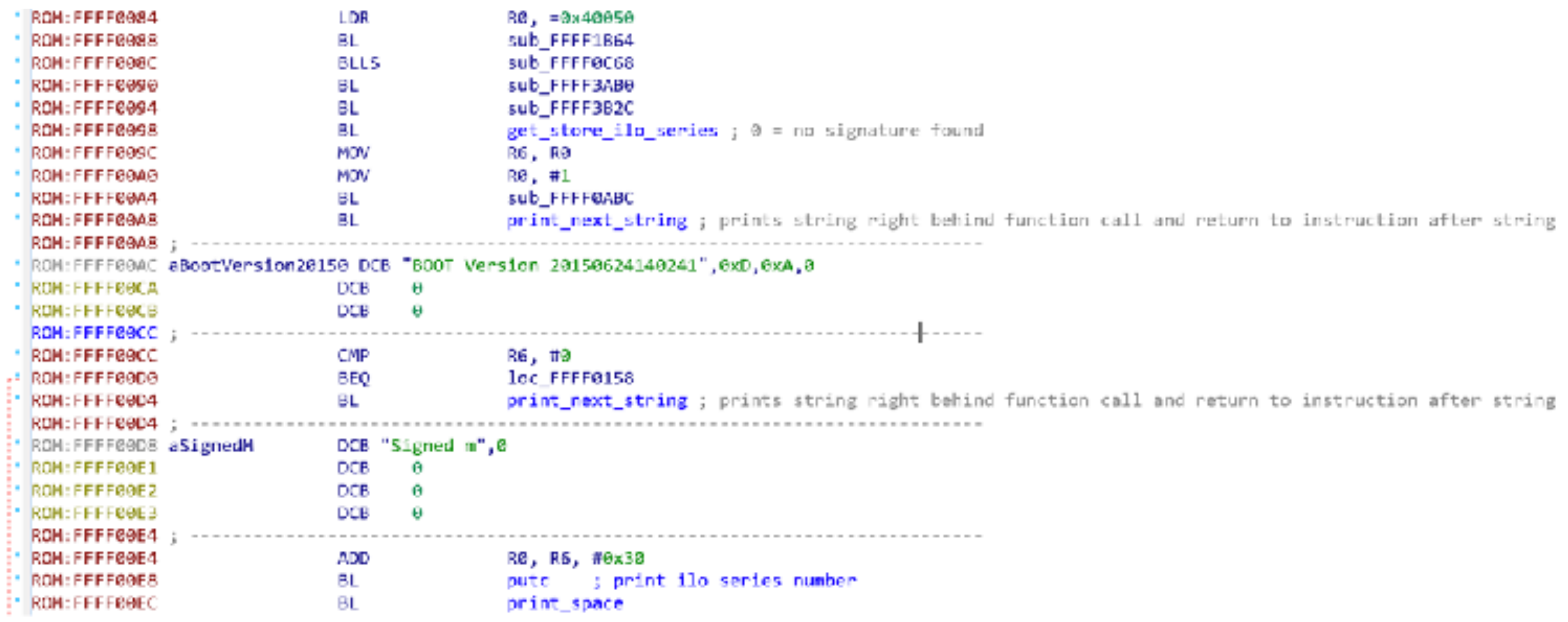


```
ROM:FFFF02A8 loc_FFFF02A8                            ; CODE XREF: ROM:FFFF02C4↓j
ROM:FFFF02A8                 MOV             R0, #1
ROM:FFFF02AC                 BL              getc_timeout
```

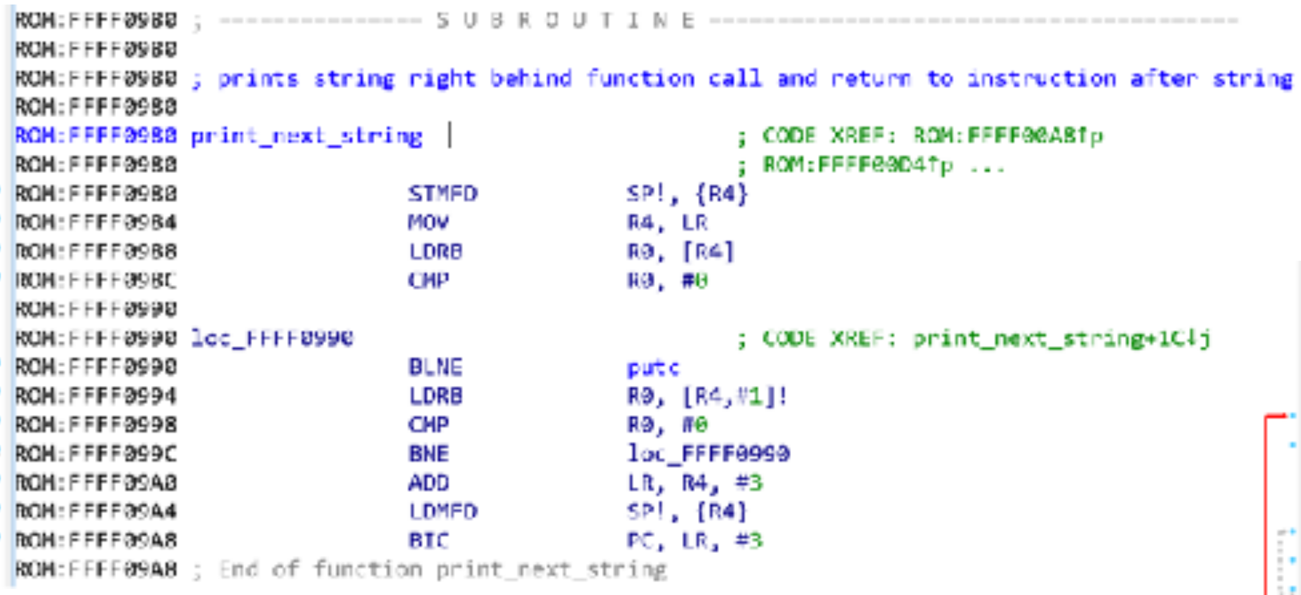


```
ROM:FFFF0E7C loc_FFFF0E7C                            ; CODE XREF: invalid_instruction_handler+B0↓j
ROM:FFFF0E7C                 ADR             R0, aCpuMExceptionK ; "\r\nCPU %M Exception [%K]: SPSR %L\r\n"
ROM:FFFF0E80                 BL              printf
```

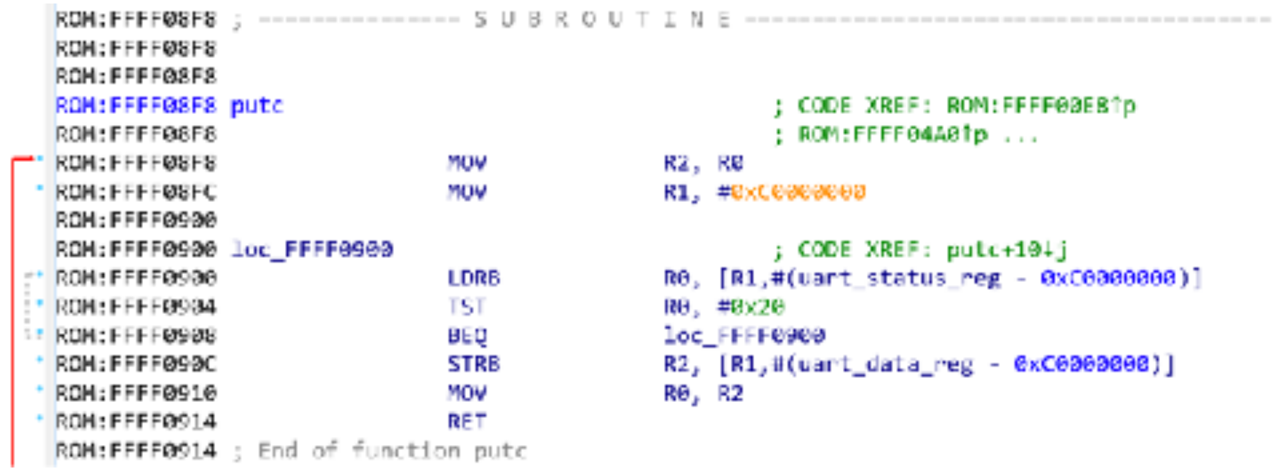
iLO4 - Firmware



```
ROM:FFFF0084                 LDR             R0, =0x40050
ROM:FFFF0088                 BL              sub_FFFF1B64
ROM:FFFF008C                 BLLS            sub_FFFF0C68
ROM:FFFF0090                 BL              sub_FFFF3AB0
ROM:FFFF0094                 BL              sub_FFFF3B2C
ROM:FFFF0098                 BL              get_store_ilo_series ; 0 = no signature found
ROM:FFFF009C                 MOV             R6, R0
ROM:FFFF00A0                 MOV             R0, #1
ROM:FFFF00A4                 BL              sub_FFFF0ABC
ROM:FFFF00A8                 BL              print_next_string ; prints string right behind function call and return to instruction after string
ROM:FFFF00A8 ; ---------------------------------------------------------------------------
ROM:FFFF00AC aBootVersion20150 DCB "BOOT Version 20150624140241",0xD,0xA,0
ROM:FFFF00CA                 DCB    0
ROM:FFFF00CB                 DCB    0
ROM:FFFF00CC ; ---------------------------------------------------------------------------
ROM:FFFF00CC                 CMP             R6, #0
ROM:FFFF00D0                 BEQ             loc_FFFF0158
ROM:FFFF00D4                 BL              print_next_string ; prints string right behind function call and return to instruction after string
ROM:FFFF00D4 ; ---------------------------------------------------------------------------
ROM:FFFF00D8 aSignedM        DCB "Signed m",0
ROM:FFFF00E1                 DCB    0
ROM:FFFF00E2                 DCB    0
ROM:FFFF00E3                 DCB    0
ROM:FFFF00E4 ; ---------------------------------------------------------------------------
ROM:FFFF00E4                 ADD             R0, R6, #0x30
ROM:FFFF00E8                 BL              putc    ; print ilo series number
ROM:FFFF00EC                 BL              print_space
```



```
ROM:FFFF0980 ; --------------- S U B R O U T I N E ---------------------------------------
ROM:FFFF0980
ROM:FFFF0980 ; prints string right behind function call and return to instruction after string
ROM:FFFF0980
ROM:FFFF0980 print_next_string |                     ; CODE XREF: ROM:FFFF00A8↑p
ROM:FFFF0980                                         ; ROM:FFFF00D4↑p ...
ROM:FFFF0980                 STMFD           SP!, {R4}
ROM:FFFF0984                 MOV             R4, LR
ROM:FFFF0988                 LDRB            R0, [R4]
ROM:FFFF098C                 CMP             R0, #0
ROM:FFFF0990
ROM:FFFF0990 loc_FFFF0990                            ; CODE XREF: print_next_string+1C↓j
ROM:FFFF0990                 BLNE            putc
ROM:FFFF0994                 LDRB            R0, [R4,#1]!
ROM:FFFF0998                 CMP             R0, #0
ROM:FFFF099C                 BNE             loc_FFFF0990
ROM:FFFF09A0                 ADD             LR, R4, #3
ROM:FFFF09A4                 LDMFD           SP!, {R4}
ROM:FFFF09A8                 BIC             PC, LR, #3
ROM:FFFF09A8 ; End of function print_next_string
```

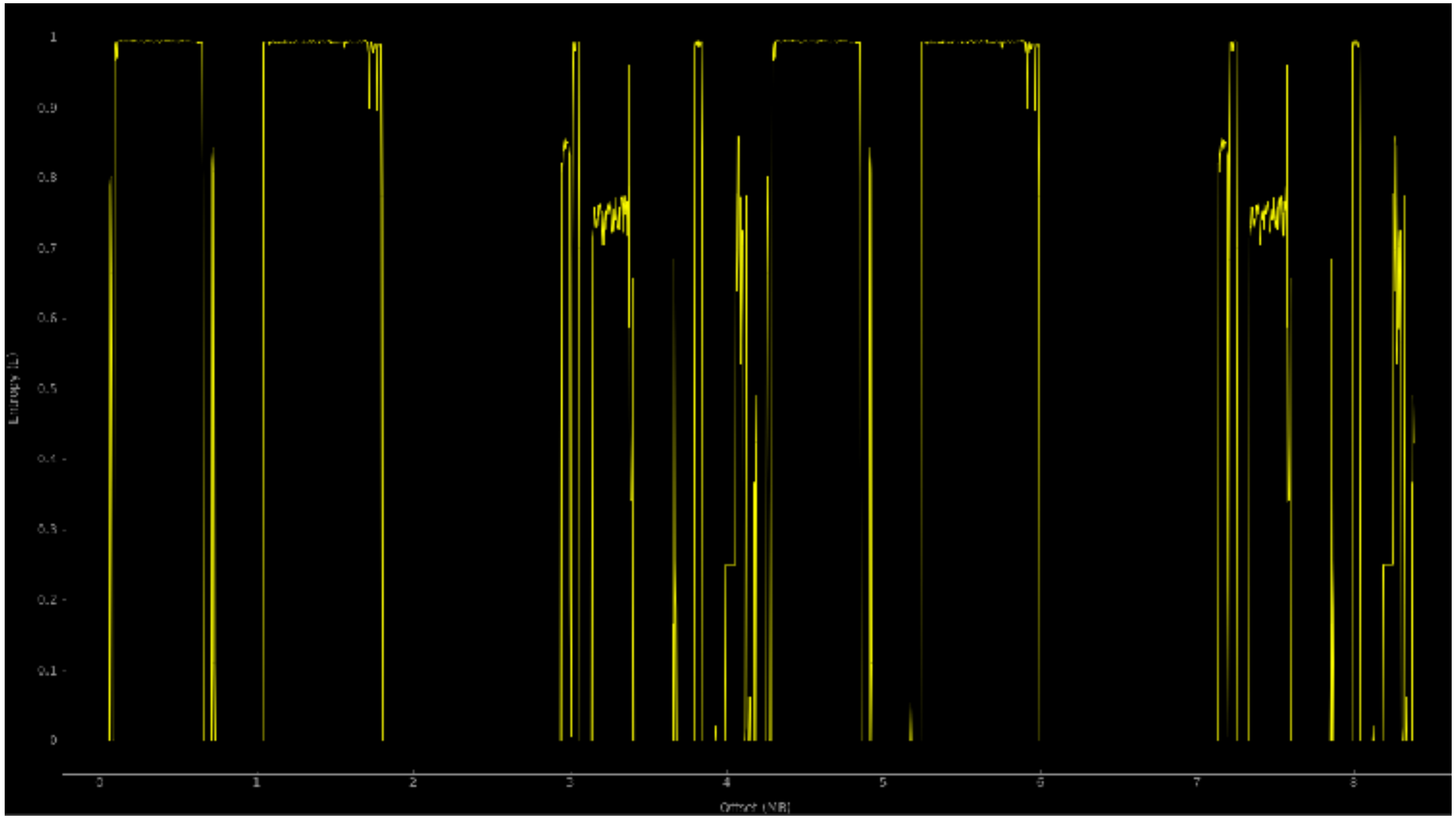


```
ROM:FFFF08F8 ; --------------- S U B R O U T I N E ---------------------------------------
ROM:FFFF08F8
ROM:FFFF08F8
ROM:FFFF08F8 putc                                    ; CODE XREF: ROM:FFFF00E8↑p
ROM:FFFF08F8                                         ; ROM:FFFF04A0↑p ...
ROM:FFFF08F8                 MOV             R2, R0
ROM:FFFF08FC                 MOV             R1, #0xC0000000
ROM:FFFF0900
ROM:FFFF0900 loc_FFFF0900                            ; CODE XREF: putc+10↓j
ROM:FFFF0900                 LDRB            R0, [R1,#(uart_status_reg - 0xC0000000)]
ROM:FFFF0904                 TST             R0, #0x20
ROM:FFFF0908                 BEQ             loc_FFFF0900
ROM:FFFF090C                 STRB            R2, [R1,#(uart_data_reg - 0xC0000000)]
ROM:FFFF0910                 MOV             R0, R2
ROM:FFFF0914                 RET
ROM:FFFF0914 ; End of function putc
```

iLO4 code execution

- Patch iLO flash bootblock with custom message and infinite loop
- flash socketed chip
- serial output “Hello ECC2017!” \o/
- No code signing
- at least theoretical possible to port openBMC

BIOS flash: binwalk -E



- 2 images (A/B image)

x86 side

- SPI flash connected to PCH only contains flash descriptor and ME image
- BIOS flash connected to iLO4 or FPGA
- BIOS flash mapped into x86 side via LPC

x86 side: autoport

- board support auto generated
- hacked up version of pilot 2 SIO
- serial over LAN via iLO4
- DDR3 SPD EEPROM not found
- hardcode SPD image for now
- currently still hangs in CAR migration

LPC trace

- LPC bus on TPM connector
  - accessible without soldering
- iCEstick FPGA board with iCE40 FPGA
  - supported by yosys and arachne-pnr
- still work in progress

Thank you for your attention.

Questions?