How FreeBSD Boots: a soft-core MIPS perspective

Brooks Davis
SRI International

SCALE x13
February 21, 2015
Bluespec
Extensible RISC Implementation
MIPS64-ISA
Soft-core CPU
Multi-core & Multi-threaded
Why BERI?
Banks lose over $300m

Sony hackers

Anthem

80 million customer records
40 million cards compromised

Total damage to banks and retailers could exceed $18 billion. — NYTimes
CVE-2012-5445:

...allows attackers to execute arbitrary code or cause a denial of service...
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>366</td>
<td>11:57:520</td>
<td>192.168.0.31</td>
<td>192.168.0.28</td>
<td>SNMP</td>
<td>get-response SNMPv2-SMI::enterprises.11.3.9.4.2.1.4.1.5.7.1</td>
</tr>
<tr>
<td>367</td>
<td>11:57:520</td>
<td>192.168.0.31</td>
<td>192.168.0.28</td>
<td>SNMP</td>
<td>get-request SNMPv2-SMI::enterprises.11.3.9.4.2.1.4.1.5.8.1</td>
</tr>
<tr>
<td>369</td>
<td>11:57:520</td>
<td>192.168.0.31</td>
<td>192.168.0.28</td>
<td>SNMP</td>
<td>get-request SNMPv2-SMI::enterprises.11.3.9.4.2.1.4.1.5.8.1</td>
</tr>
<tr>
<td>384</td>
<td>12:31:862</td>
<td>192.168.0.1</td>
<td>192.168.0.28</td>
<td>DNS</td>
<td>Standard query response A 64.236.91.21 A 64.236.91.23 A 64.236.91.23</td>
</tr>
<tr>
<td>385</td>
<td>12:31:727</td>
<td>192.168.0.28</td>
<td>64.236.91.21</td>
<td>TCP</td>
<td>http &gt; 56606 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2</td>
</tr>
<tr>
<td>386</td>
<td>12:31:727</td>
<td>192.168.0.28</td>
<td>64.236.91.21</td>
<td>TCP</td>
<td>http &gt; 56606 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460</td>
</tr>
<tr>
<td>387</td>
<td>12:31:953</td>
<td>64.236.91.21</td>
<td>64.236.91.21</td>
<td>TCP</td>
<td>http &gt; 56606 [ACK] Seq=1 Ack=1 Win=17520 Len=0</td>
</tr>
<tr>
<td>388</td>
<td>12:31:727</td>
<td>64.236.91.21</td>
<td>64.236.91.21</td>
<td>TCP</td>
<td>[TCP segment of a reassembled PDU]</td>
</tr>
<tr>
<td>389</td>
<td>12:31:727</td>
<td>64.236.91.21</td>
<td>64.236.91.21</td>
<td>TCP</td>
<td>[TCP segment of a reassembled PDU]</td>
</tr>
</tbody>
</table>

Frame 384 (167 bytes on wire, 167 bytes captured)

Ethernet II, Src: Sparklan_04:04:0e (00:0e:8e:04:04:0e), Dst: HonHaiPr_26:66:a2 (00:1c:26:26:66:a2)

Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.28 (192.168.0.28)

User Datagram Protocol, Src Port: domain (53), Dst Port: 62872 (62872)

DNS (response)

[Request In: 381]
[Time: 0.025771000 seconds]
Transaction ID: 0xcf1f
Flags: 0x8180 (Standard query response, No error)
Questions: 1
Answer RRs: 6
Authority RRs: 0
Additional RRs: 0

Queries
- www.cnn.com: type A, class IN
  Name: www.cnn.com
  Type: A (host address)
  Class: IN (0x0001)

Answers
- www.cnn.com: type A, class IN, addr 64.236.91.21
  Class: IN (0x0001)
Vulnerabilities by year

2006  2007  2008  2009  2010  2011  2012  2013
Clean-slate design of Resilient, Adaptive, Secure Hosts
The FreeBSD Project

FreeBSD is an advanced computer operating system used to power modern servers, desktops and embedded platforms. A large community has continually developed it for more than thirty years. Its advanced networking, security and storage features have made FreeBSD the platform of choice for many of the busiest web sites and most pervasive embedded networking and storage devices.

» Learn More

LATEST NEWS
2014-02-10  FreeBSD Journal First Edition Available
2014-01-23  New committer: Rodrigo Osorio (ports)
2014-01-21  New committer: Tycho Nightingale (src)
2014-01-21  New committer: Michael Gmelin (Ports)
More News  RSS

UPCOMING EVENTS
2014-03-12 - 2014-03-12  bhyvecon 2014 (Tokyo, Japan)
2014-03-13 - 2014-03-16  AsiaBSDCon 2014 (Tokyo, Japan)
2014-05-14 - 2014-05-17  BSDCan 2014 (Ottawa, Canada)
More Events
The FreeBSD Project

FreeBSD is an advanced computer operating system used to power modern servers, desktops and embedded platforms. A large community has continually developed it for more than thirty years. Its advanced networking, security and storage features have made FreeBSD the platform of choice for many of the busiest web sites and most pervasive embedded networking and storage devices.

Learn More

LATEST NEWS
2014-02-10
FreeBSD Journal First Edition Available

2014-01-25
October-December, 2013 Status Report

2014-01-23
New committer: Rodrigo Osorio (ports)

2014-01-21
New committer: Tycho Nightingale (src)

2014-01-21
New committer: Michael Gmelin (ports)

More News

UPCOMING EVENTS
2014-03-12 - 2014-03-12
bhyvecon 2014
(Tokyo, Japan)

2014-03-13 - 2014-03-16
AsiaBSDCon 2014
(Tokyo, Japan)

2014-05-14 - 2014-05-17
BSDCan 2014
(Ottawa, Canada)

More Events

PRESS
2014-01
McKusick Denies FreeBSD Lagging on Security

2013-06
Fixing Network Attached Storage with commodity hardware and BSD

2013-02
2012 - A BSD Year in Retrospective

2013-01
A Decade of OS Access-control Extensibility

2012-11
A world without Linux: Where would Apache, Microsoft — even Apple be today?

More Media

SECURITY ADVISORIES
2014-01-14
FreeBSD-SA-14-04.blind

2014-01-14
FreeBSD-SA-14-03.openssl

2014-01-14
FreeBSD-SA-14-02.ntd

2014-01-14
FreeBSD-SA-14-01.bsm

More

ERRATA NOTICES
2014-01-14
FreeBSD-EN-14-02.mmap

2014-01-14
FreeBSD-EN-14-01.random

More

Site Map | Legal Notices | © 1995–2014 The FreeBSD Project. All rights reserved. The mark FreeBSD is a registered trademark of The FreeBSD Foundation and is used by The FreeBSD Project with the permission of The FreeBSD Foundation. Contact.
Capability
Hardware
Enhanced
RISC
Instructions
BERI
+ Memory Capabilities
= CHERI
Memory Capabilities

- Unforgeable references to memory
- Implemented as a MIPS coprocessor
- All memory access via a capability
- Regular MIPS instructions access via a default capability
Other BERI research topics

- Novel multi-threaded CPUs
- Multi-core architectures
- Tight integration of CPUs and switches
- Validation of formal models
Where does BERI run?
Simulation
NetFPGA-10G
Terasic DE4
1GB DDR2 DRAM
64GB NOR flash

CFI(4)
LEDs

terasic_de4led(4)
Buttons & Switches (/dev/de4bsw)
CPU temperature
/dev/de4tempfan
Gigabit Ethernet

atse(4)
HSMC connector
HDMI out card
GPIO connector
Touch screen
terasic_mtl(4)
RS232
uart(4)
SD card (2GB)
altera_sdcard(4)
Booting CheriBSD
Please Wait
Booting the DE4
Booting the DE4

**DRAM**

- **Flash**
  - 128k FPGA 1
  - 12MB FPGA2
  - 12MB Operating system (UFS)
  - 39.75MB boot2

- **Reserved** 128k

- **FPGA 1**
  - 12MB

- **FPGA 2**
  - 12MB

- **Operating system (UFS)**
  - 39.75MB

- **boot2**
  - 128k

**/boot/loader**

**/boot/kernel/kernel**
Booting the DE4

Flash

<table>
<thead>
<tr>
<th></th>
<th>FPGA 1</th>
<th>FPGA2</th>
<th>Operating system (UFS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-segment Displays</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 User LEDs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-Position DIP Switch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USB Type mini-AB Port</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50MHz Oscillator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSIC Port A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXII CPLD EP2C20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Controller</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Re-Configuration Push-button</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two 40-pin GPIO Connectors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External PLL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+12V Fan Connector</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clock Input/Output SMA Connectors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100MHz Oscillator</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12V and 3.3V Power Supply Connector</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power Switch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RS-232 Port</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JTAG 3-Position DIP Switch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI Express x8 Edge Connector</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Gigabit Ethernet Ports</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU Reset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SunDisk 2.0GB SD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

/boot/loader

/boot/kernel/kernel

miniboot

128k 12MB 12MB 39.75MB 128k
Booting the DE4

Flash
- FPGA 1: 128k, 12MB
- FPGA 2: 128k, 12MB
- Operating system (UFS): 39.75MB

DRAM
-Reserved 128k

FPGA 1: 12MB
FPGA 2: 12MB

/boot/loader
/boot/kernel/kernel

Operating system (UFS): 39.75MB
128k
### Booting the DE4

#### DRAM

<table>
<thead>
<tr>
<th>Component</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>128k</td>
</tr>
<tr>
<td>FPGA2</td>
<td>12MB</td>
</tr>
<tr>
<td>Flash</td>
<td>128k</td>
</tr>
<tr>
<td>Flash</td>
<td>128k</td>
</tr>
<tr>
<td>Flash</td>
<td>128k</td>
</tr>
<tr>
<td>Flash</td>
<td>128k</td>
</tr>
<tr>
<td>Flash</td>
<td>128k</td>
</tr>
<tr>
<td>Flash</td>
<td>128k</td>
</tr>
</tbody>
</table>

#### Flash Memory

- **FPGA 1**: 128k
- **FPGA 2**: 12MB
- **Operating system (UFS)**: 39.75MB

#### Boot Files

- `/boot/loader`
- `/boot/kernel/kernel`
Booting the DE4

Flash

<table>
<thead>
<tr>
<th></th>
<th>FPGA 1</th>
<th>FPGA2</th>
<th>Operating system (UFS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>128k</td>
<td>12MB</td>
<td>39.75MB</td>
</tr>
<tr>
<td>Reserved</td>
<td>128k</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DRAM

- 2.0GB SD Card
- 7-segment Displays
- 8-pin DIP Switch
- 8-Position DIP Switch
- Type A Ports
- USB Type mini-AB Port
- 8-user LEDs
- 4 Serial ATA Ports
- 12V and 3.3V Power Supply Connector

FPGA 1
- 12MB
- 2MB SRAM
- 64MB FLASH
- 4Serial ATA Ports
- 4 Gigabit Ethernet Ports
- JTAG 3-Position DIP Switch
- PCI Express x8 Edge Connector

FPGA 2
- 12MB
- 4 Serial ATA Ports
- 12MB
- 4 Serial ATA Ports

boot loader

/kernel/kernel

/boot
Early kernel boot

• Enter kernel in `_locore()` at `_start`
  • Calls BERI specific `platform_init()`
  • Finally calls `mi_startup()`
SYSINITs

- Initializer functions
  - Declared with `SYSINIT()` macro
  - Sorted by `subsystem` and `order`
  - Run in `mi_startup()`
Copyright (c) 1992-2013 The FreeBSD Project.
  The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.
SI_SUB_CONFIGURE

- `configure_first()` @ SI_ORDER_FIRST
  - Attaches root bus

- `configure()` @ SI_ORDER_THIRD
  - Attaches all devices

- BERI uses a mix of FDT and hints
Flat Device Tree

cpus {
    cpu@0 {
        device-type = "cpu ";
        compatible = "sri-cambridge,beri";
    };
};

soc {
    beripic : beripic@7f804000 {
        compatible = "sri-cambridge,beri-pic";
        interrupt-controller;
        reg = <0x7f804000 0x400 0x7f806000 0x10
             0x7f806080 0x10 0x7f806100 0x10>;
    };
    ...
};
static Elf64_Brandinfo freebsd_brand_info = {
    .brand = ELFOSABI_FREEBSD,
    .machine = EM_MIPS,
    .compat_3_brand = "FreeBSD",
    .emul_path = NULL,
    .interp_path = "/libexec/ld-elf.so.1",
    .sysvec = &elf64_freebsd_sysvec,
    .interp_newpath = NULL,
    .flags = 0
};

SYSINIT(elf64, SI_SUB_EXEC, SI_ORDER_ANY,
    (sysinit_cfunc_t) elf64_insert_brand_entry,
    &freebsd_brand_info);
Starting /sbin/init

- `create_init()` at SI_SUB_CREATE_INIT
  - Creates process

- `kick_init()` at SI_SUB_KTHREAD_INIT
  - Makes process runnable

- `scheduler()` at SI_SUB_RUN_SCHEDULER
  - Schedules processes
SMP
SI_SUB_TUNABLES

Sets `mp_cpus` and `mp_maxid` from `platform_cpu_mask()`
start_ap(n) {
  ...
  cpus = mp_naps;
  platform_start_ap(n);
}
start_ap(n) {
    ...
    cpus = mp_naps;
    platform_start_ap(n);
    mpentry() →
    while (mp_naps <= cpus)
        DELAY(1000);
SI_SUB_CPU

BSP

start_ap(n) {
  ...
  cpus = mp_naps;
  platform_start_ap(n);

  while (mp_naps <= cpus) {
    DELAY(1000);
  }
}

APn

mpentry() →

smp_init_secondary() {
  ...
}
SI_SUB_CPU

BSP

start_ap(n) {
  ...
  cpus = mp_naps;
  platform_start_ap(n);
  while (mp_naps <= cpus)
    DELAY(1000);

APn

mpentry() →
  smp_init_secondary() {
    ...
    mp_naps++;
    ...
    while (!aps_ready)
      ;
**SI_SUB_CPU**

**BSP**

```c
start_ap(n) {
    ...
    cpus = mp_naps;
    platform_start_ap(n);
    while (mp_naps <= cpus)
        DELAY(1000);
}
```

**APn**

```c
mpentry() →
smp_init_init_secondary() {
    ...
    mp_naps++;
    ...
    while (!aps_ready)
        ;
```
Spin table

struct {
    uint64_t entry_addr;
    uint64_t a0;
    uint32_t rsvd1;
    uint32_t pir;
};
SI_SUB_SMP

BSP

APn

smp_init_secondary() {
    ...
    while (!aps_ready)
        ;
SI_SUB_SMP

BSP

release_aps()
{
    /* IPI setup */
    ...

APn

smp_init_secondary()
{
    ...
    while (!aps_ready)
    {
    }

release_aps() {
    /* IPI setup */
    ...
    aps_ready = 1;

    while (!smp_started) {
        ...
    }

    smp_init_secondary() {
        ...
        while (!aps_ready) {
            ...
        }
    }
**SI_SUB_SMP**

**BSP**

```c
release_aps() {
    /* IPI setup */
    ...
    aps_ready = 1;

    while (!smp_started)
        ;
}
```

**APn**

```c
smp_init_secondary() {
    ...
    while (!aps_ready)
        ;
    ...
    if (/* last AP */)
        smp_started = 1;
```
SI_SUB_SMP

BSP

release_aps() {
    /* IPI setup */

    ...}

    aps_ready = 1;

    while (!smp_started) ;

APn

smp_init_secondary() {

    ...}

    while (!aps_ready) ;

    ...

    if (/* last AP */) {
        smp_started = 1;

        while (!smp_started) ;
    }
release_aps() {
    /* IPI setup */
    ...
    aps_ready = 1;

    while (!smp_started)
        ;
}

smp_init_secondary() {
    ...
    while (!aps_ready)
        ;
    ...
    if /* last AP */
        smp_started = 1;

    while (!smp_started)
        ;
    ...
    /* enter scheduler */
}
FreeBSD Journal
http://freebsdjournal.org

Porting FreeBSD to a new CPU, even within a previously supported family, is a significant undertaking.