Subject: FLASH storage devices
Posted by glevand on Wed, 07 Mar 2012 16:58:51 GMT
View Forum Message <> Reply to Message

Here i will write down anything i will found about FLASH devices.

Subject: Re: FLASH storage devices
Posted by glevand on Wed, 07 Mar 2012 21:15:34 GMT
View Forum Message <> Reply to Message

FLASH storage device object in HV has a flag at offset 0x1ECC of size 4 bytes.

This flag tells if the FLASH device is NOR or NAND (Starship).

The offset is the same even on 3.55.

Value 0x1 means NAND FLASH.
Value 0x2 means NOR FLASH.

HV checks this when the next storage request is processed and branches accordingly either to NOR or NAND subroutines.

Subject: Re: FLASH storage devices
Posted by glevand on Fri, 09 Mar 2012 21:31:04 GMT
View Forum Message <> Reply to Message

ENCDEC debug level on 3.55 is at address 0x35af3c.

```
glevand@debian:~$ sudo dd if=/dev/ps3physmem bs=1 count=4 skip=$((0x35af3c)) | hexdump -C
00000000  00 00 00 00                                       |....|
00000004
```

Set level to 1.

```
glevand@debian:~$ echo "0: 00 00 00 01 " | xxd -c 256 -r | sudo dd of=/dev/ps3physmem bs=1
count=4 seek=$((0x35af3c))
```

LV1 debug buffer is at 0x684360.

Dump debug buffer.

```
sudo dd if=/dev/ps3physmem bs=1 count=4080 skip=$((0x684360)) | strings
```

Clear debug buffer.

sudo dd if=/dev/zero of=/dev/ps3physmem bs=1 count=4080 seek=$((0x684360))

Subject: Re: FLASH storage devices
Posted by glevand on Fri, 09 Mar 2012 21:49:07 GMT
View Forum Message <> Reply to Message

Read sector 0 from ps3flashb.

sudo dd if=/dev/ps3nflashb skip=0 bs=512 count=1 of=/dev/null

Debug messages.

EdecSS start.
SetStgSsDbufEncdec ENC OR DEC:1 lbn:8 num:20 secsize:200 key:1 usr_sb_addr:c0040000 sp_sb_addr:1f001000
EncDec Interrupt Reason:6 ret:0
Encdec decsec.
SetStgSsDbufEncdec ENC OR DEC:0 lbn:8 num:20 secsize:200 key:1 usr_sb_addr:c0040000 buf_sb_addr:c0040000
EncDec Interrupt Reason:6 ret:0
Encdec decsec.

Hmm, it looks like for one read 2 encdec requests are processed.
First decrypt and then encrypt. Interesting.
But we know that NOR flash is NOT encrypted.
It makes sense. ENCDEC can either encrypt or decrypt.
First ENCDEC decrypts a NOR sector and then encrypts it.
In the end we have an unencrypted NOR sector because decrypting and encrypting it didn't change the NOR sector.

Another thing is that changing the key should not do any harm to our NOR sectors. I have to patch LV1 and test it.

Subject: Re: FLASH storage devices
Posted by glevand on Fri, 09 Mar 2012 22:34:58 GMT
View Forum Message <> Reply to Message

ENCDEC is really interesting stuff. I want it really bad to decrypt/encrypt data i pass to encdec.

Subject: Re: FLASH storage devices
Posted by glevand on Fri, 09 Mar 2012 22:41:32 GMT
View Forum Message <> Reply to Message

Hehe, tested with key 2 but didn't work because key 2 was used for decryption and key 1 for encryption. I didn't patch all LV1 places which need to be patched. Data was corrupted :)

EdecSS start.
SetStgSsDbufEncdec ENC OR DEC:1 lbn:8 num:20 secsize:200 key:2 usr_sb_addr:c0040000 sp_sb_addr:1f001000
EncDec Interrupt Reason:6 ret:0
Encdec decsec.
SetStgSsDbufEncdec ENC OR DEC:0 lbn:8 num:20 secsize:200 key:1 usr_sb_addr:c0040000 buf_sb_addr:c0040000
EncDec Interrupt Reason:6 ret:0
Encdec decsec.

That proves at least the theory that NOR sectors are actually read through ENCDEC device first by decrypting NOR sectors and then encrypting them :d Let me patch all places in LV1 and we should get correct data with key 2 too.

Subject: Re: FLASH storage devices
Posted by glevand on Fri, 09 Mar 2012 23:00:34 GMT
View Forum Message <> Reply to Message

Yeah, it worked with key 2 :)

EdecSS start.
SetStgSsDbufEncdec ENC OR DEC:1 lbn:8 num:20 secsize:200 key:2 usr_sb_addr:c0040000 sp_sb_addr:1f001000
EncDec Interrupt Reason:6 ret:0
Encdec decsec.
SetStgSsDbufEncdec ENC OR DEC:0 lbn:8 num:20 secsize:200 key:2 usr_sb_addr:c0040000 buf_sb_addr:c0040000
EncDec Interrupt Reason:6 ret:0
Encdec decsec.

And guess what, NOR sectors are correct again :d

Subject: Re: FLASH storage devices

Posted by glevand on Sat, 10 Mar 2012 09:52:46 GMT
View Forum Message <> Reply to Message

Hmm, found interesting stuff about writing NOR flash.
ENCDEC is not used for writing sectors.
Before writing data to NOR flash, LV1 checks if the address where data should be written is protected or not.

Subject: Re: FLASH storage devices
Posted by glevand on Sat, 10 Mar 2012 10:13:04 GMT
View Forum Message <> Reply to Message

It seems that LV1 has an array of FLASH configurations.
Each entry is 0x60 bytes. On 3.55 the entry size is 0x68.
NOR flash is entry with index 0x4 on PHAT. On SLIM it's entry with index 0x4 too.
Each entry stores total size of FLASH.
And LV1 checks requests for correct byte offset nd size by using this table.
FLASH block size is stored here too.
NOR flash has 128kb block size. Total number of blocks is stored here too: 0x80 blocks.

Subject: Re: FLASH storage devices
Posted by glevand on Sat, 10 Mar 2012 10:56:58 GMT
View Forum Message <> Reply to Message

To get mmaped access to NOR flash on Linux 3 with ps3physmem do this:

```
sudo modprobe ps3physmem ps3physmem_start=0x2401F000000
sudo hexdump -C /dev/ps3physmem | less
```

With the new ps3physmem driver you can map any physical address and not only physical RAM.
No need for ps3sbmmio and ps3rsxmmio drivers on Linux 3.

Subject: Re: FLASH storage devices
Posted by glevand on Sat, 10 Mar 2012 11:00:21 GMT
View Forum Message <> Reply to Message

Hmm, to enable writing to FLASH, LV1 writes some magic values to flash controller first by using memory mapped access.
Let's see if we can reproduce it.

Subject: Re: FLASH storage devices
Posted by glevand on Sat, 10 Mar 2012 11:08:41 GMT
View Forum Message <> Reply to Message

1. LV1 writes 0xAAAA at offset x+0xAAA.
2. LV1 writes 0x5555 at offset x+0x554.
3. LV1 writes 0xE0E0 at offset x+0xAAA.
4. LV1 reades 2 bytes at offset x+0x0.
5. LV1 writes 0x9090 at offset x+0x0.
6. LV1 writes 0x0000 at offset x+0x0.

Copy data from flash to LV1 buffer.

1. LV1 writes 0xF0F0 at offset x+0x0.
2. LV1 writes 0xAAAA at offset x+0xAAA.
3. LV1 writes 0x5555 at offset x+0x554.
4. LV1 writes 0x8080 at offset x+0xAAA.
5. LV1 writes 0x5555 at offset x+0x554.
6. LV1 writes 0x3000 at offset x+???.

x = 0x2401F000000 + offset of flash block in bytes where data is written

E.g. x would be 0xf20000 for ps3flashc.

Subject: Re: FLASH storage devices
Posted by MoKoffman on Sat, 10 Mar 2012 19:34:34 GMT
View Forum Message <> Reply to Message

That magic looks awfully familiar to the NOR chip commands to me :)

http://www.spansion.com/Support/Datasheets/S29GL-P_00.pdf

Subject: Re: FLASH storage devices
Posted by glevand on Sat, 10 Mar 2012 19:43:51 GMT
View Forum Message <> Reply to Message

Thanks for the tip

Subject: Re: FLASH storage devices
Posted by sktsqrl on Fri, 16 Mar 2012 23:57:28 GMT

lv0 (old version, I don't have any new ones) is doing pretty much the same commands.