



# Flash subsystem for NetBSD<sup>®</sup>

Device drivers for NAND flash memory

Adam Hoka  
The NetBSD Foundation

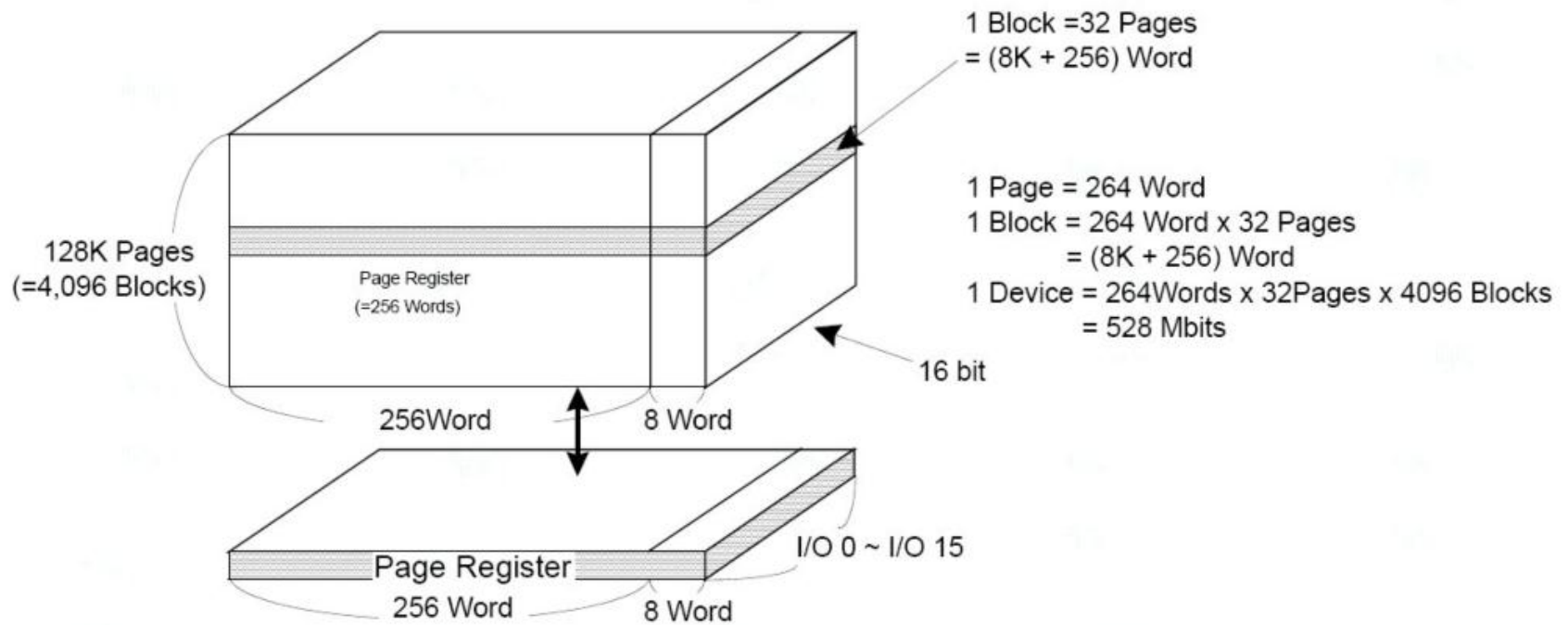
# Flash memory

- NOR flash
  - Random r/w access, execute-in-place capability
  - Small storage size, expensive
- NAND flash
  - Cheap, high storage capacity
  - No random access possible, unreliable
  - Very common in embedded devices

# NAND flash memory

- Read and write are on a per page basis
  - e.g. 2048 bytes for large page NAND
- Writing needs an erase operation first
- Erase is on a per block basis
  - e.g. 64 pages is a block and thus 128kbytes
- Blocks age, and become unusable over time
  - “Bad blocks” need to be marked
- Needs ECC because of this aging

# NAND flash architecture



# Where do we find NAND?

- Embedded boards (ARM, MIPS, etc)
  - Smart phones
  - Portable music players
  - PC motherboards (BIOS)
- The first two is important for us, because we want to support NetBSD on those devices.



# “Prior art”

- Linux MTD
  - Supports many devices, but horrible code
- U-boot MTD
  - Almost the same as Linux, except better quality
- Andrew Turner's NAND Driver (FreeBSD)
  - 8bit devices only, limited feature support

# Flash device hierarchy in NetBSD

- Device specific driver
  - implements the NetBSD nand(4) API
- nand(4) driver
  - Uses functions exported from the device specific driver to communicate with the NAND chip using ONFI standard commands
  - Implements the flash(4) API
- flash(4) driver
  - Provides a high level API for the flash file system and a block device interface through /dev/flash\*



# Why is it better than Linux MTD?

- Implemented using industry standard ONFI NAND commands and specifications
- Clear interface and understandable code path (mtd is a spaghetti monster)
- Designed for modern devices, no legacy code

# Configuration example

```
# NAND controller
```

```
omapnand0      at gpmc? addr 0x30000000
```

```
# NAND layer
```

```
nand0      at omapnand0
```

```
# Define FLASH partitions for board
```

```
flash0     at nand0 offset 0x0 size 0x80000 readonly 1
```

```
flash1     at nand0 offset 0x80000 size 0x80000 readonly 1
```

```
flash2     at nand0 offset 0x260000 size 0x20000
```

```
flash3     at nand0 offset 0x280000 size 0x400000
```

```
flash4     at nand0 offset 0x680000 size 0x0
```

# The state of implementation

- What's finished?
  - NAND commands (standard ONFI 2.3)
  - Block device driver
  - Partition support
  - ECC error checking support
  - flashctl(8), a tool to manage flash devices
- What needs to be done?
  - Bad block handling needs improvement
  - Test on more hardware and bugs to find and fix

The Project named „TÁMOP-4.2.1/B-09/1/KONV-2010-0005 – Creating the Center of Excellence at the University of Szeged” is supported by the European Union and co-financed by the European Regional Fund.

[www.nfu.hu](http://www.nfu.hu)

*Investing in your future*



*New Hungary Development Plan*





**Thank you for listening!**

Visit <http://chewiefs.sed.hu/> for more information

**Any questions?**