

Grundlagen und Anwendungen von Flash Speichern

Fachkonferenz
Software-Flashen im KFZ

Hubert Högl
Hubert.Hoegl@fh-augsburg.de

München, 19. Mai 2003

Der Flash-Speicher Markt

Nr.	Unternehmen	2002 Umsatz in Mio. \$	2002 Marktanteil
1	Intel	2058	26,3 %
2	Samsung	1181	15,1 %
3	Toshiba	918	11,7 %
4	AMD	727	9,3 %
5	Fujitsu	665	8,5 %
6	STMicro	616	7,9 %
7	Sharp	403	5,2 %
8	Mitsubishi	392	5,0 %
9	SST	275	3,5 %
10	Hitachi	249	3,2 %

Gesamter Umsatz: 7818 Mio. \$

[Quelle: Computerwoche/iSuppli]

Vom UV- zum Flash-EPROM

- Der **UV-EPROM** (1983)



2716 (2K × 8 = 16 KBit)

- Flash- und EEPROM Speicher als **Weiterentwicklung** des UV-EPROM Speichers
- Erster Flash-Speicher im Jahr **1988** von Intel (ETOX)
- Heute in der **achten Generation**
- **Zellengrösse** um Faktor **233** geschrumpft: von $36\mu m^2$ (1.5μ) auf $0.154\mu m^2$ (0.13μ)

Grundlegende Eigenschaften

- **Nicht-flüchtiger**, (begrenzt) **reprogrammierbarer** Speicher
- **Robust, schnell, zuverlässig, klein, leicht**
- Beschreiben und Löschen (Zyklus) durch **elektrische Signale**
- Nur **blockweise** Operationen (z.B. **64 × 16K Sektoren**)
- **Löschen: 0 → 1; Programmieren: 1 → 0**
- Vor dem Löschen **alles auf 0** programmieren
- Zwei verschiedene **Flash-Array Strukturen: NOR und NAND**
- Vorsicht: **Speicherfähigkeit** nimmt mit Anzahl Zyklen ab, z.B. Programmierzeit je Byte $6 \mu\text{s}$ (Minimum) . . . **300 μs !**
- **EEPROMs** sind enge Verwandte mit **byte-weiser** Programmierung und Optimierung für hohe Zykluszahl (Parameterspeicher)

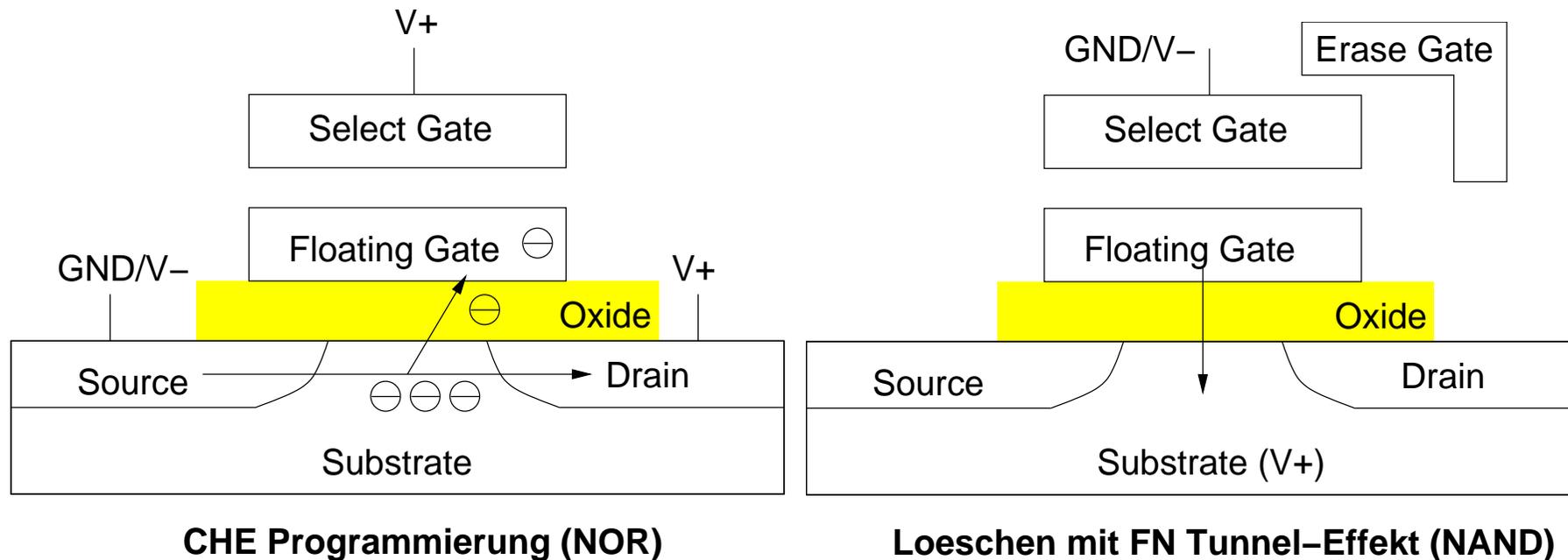
Physik I

Folgende physikalische Effekte bestimmen die grundlegenden Eigenschaften des Flash-Speichers:

- **Channel Hot Electron Injection (CHE)**
- **Fowler/Nordheim Tunnel-Effect (FN)**

Techn.	Programmieren	Löschen	Verwendung	Erfinder
NOR	FN oder CHE	FN	Code/Daten	Intel, 1988
NAND	FN	FN	Daten	Toshiba, 1989

Physik II



- CHE: langsam, sanft, zuverlässig
- FN: schnell, effizient durch fokussierten Elektronentransfer, jedoch fehleranfälliger als CHE
- Güte der Oxidschicht wichtig für schnelles Schreiben/Löschen

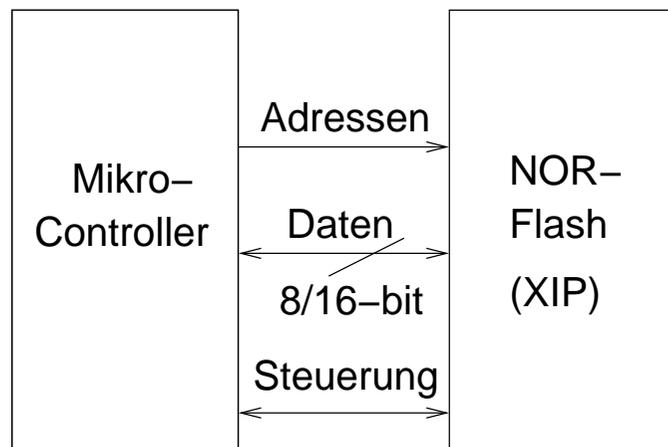
NOR- vs. NAND-Flash: Technische Daten

	NOR†	NAND‡
Prog./Löschen	CHE/FN	FN/FN
Zellengrösse	1	ca. 1/3 NOR
Sektorgrösse	gross (64K)	klein (8K)
Bus	Adr/Daten	I/O (sequentiell)
Zugriffszeit	ca. 50 ns	ca. 50 ns
Progr.-Zeit/Byte	ca. 6 μ s	ca. 430 ns
Lösch-Zeit/Sektor	ca. 600 ms	ca. 2 ms
Kapazität (zur Zeit)	bis 8 MB	bis 128 MB
Datenerhaltung	10-20 Jahre	10-20 Jahre
Lebensdauer (\approx Zyklen)	100.000	250.000 – 1 Mio
Fehleranfälligkeit	gering	grösser
Kosten	hoch	gering (1/3 NOR)

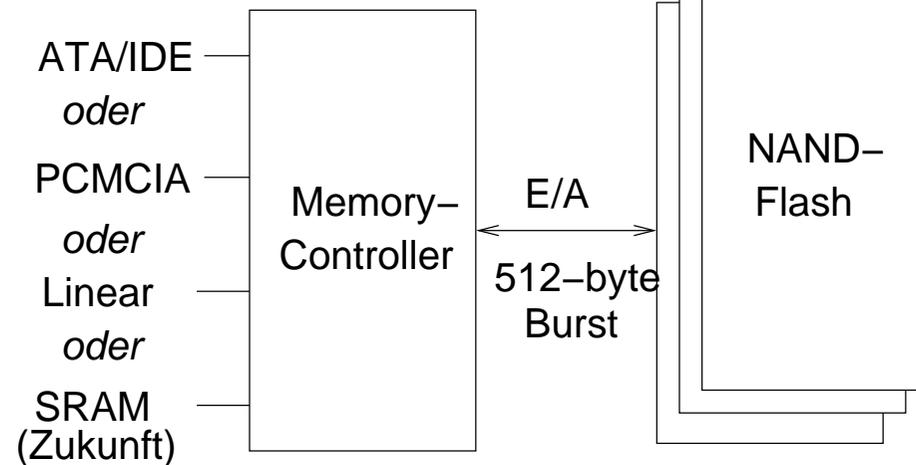
†: AMD Am29F400B (4 MBit)

‡: AMD Am30LV0064D (64MBit „UltraNAND“)

NOR- vs. NAND-Flash: Buszugriff



Wahlfreier Zugriff
(Programmspeicher bei eingebetteten Rechnern)



Blockweiser Zugriff
(Massenspeicher)

Fehlerquellen

- **Überlöschung**, d.h. unsymmetrischer Elektronentransfer bei Programmier-/Lösch Zyklen
- **Bit-Flipping**
- Überschreitung der **maximalen Zykluszahl**
- Zu hohe **Umgebungstemperatur**

Massnahmen gegen Fehler

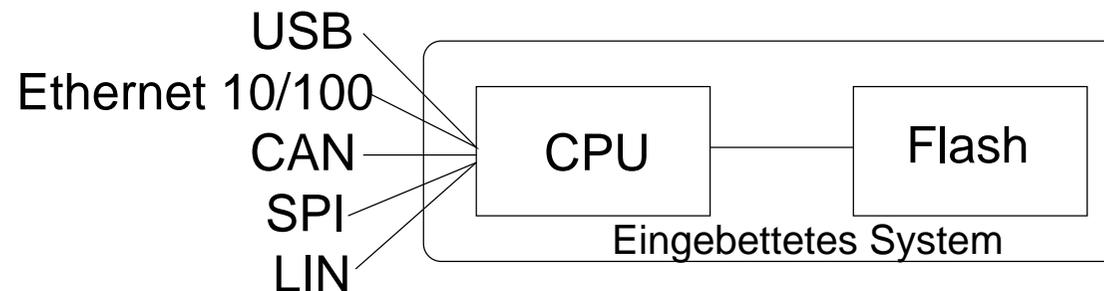
- Penible Einhaltung der **Programmialgorithmen**
- **Prüfsummen**
- Gleiche **Schreib-/Löschbelastung** der Sektoren
- Verwende Bauteile mit erweiterten **Temperaturbereich** (-40 . . . 145 Grad)

Maximale Datenraten bei der Programmierung

- Traditionelles NOR-Flash, z.B. Am29F400: ca. $6 \mu\text{s}/\text{Byte}$ (best-case) → ca. **1.33 MBit/s** bzw. 6.3 sec für 1 MByte
- Modernes NAND-Flash, z.B. Am30LV0064D: ca. $430 \text{ nsec}/\text{Byte}$ (best-case) → ca. **18.6 MBit/s** bzw. 0.43 sec für 1 MByte
- *In der Praxis wird die maximale Programmiertrate häufig wegen beschränkter Busrate nicht erreicht. Im Auto z.B. CAN-Bus mit max. 1 MBit/sec oder LIN-Bus mit 20 KBit/s.*

Programmiermethoden

- **Mit CPU** – Programmierdaten gelangen über Schnittstelle zur CPU, die das Flash im System programmiert



⇒ Meist kann maximale Programmierrate erreicht werden, falls Zufuhr der Programmierdaten kein Engpass ist

- **Ohne CPU** – meist über den **JTAG** Test Access Port (TAP)
 - ⇒ Bei 25 MHz JTAG-TCK und 132 Scan-Bits kann noch etwa mit 300 KBit/sec programmiert werden
- Weiterhin: Automatische mechanische Programmieradapter, Gang Programmer, Einzelprogrammer

Optimierung von NOR-Flash Bausteinen

Am Beispiel des Am29BDD160 (16 MBit)

- **16-Bit/32-Bit** Bus (unterstützt von modernen 32-Bit Controller)
- **Burst Modus** (54 nsec + 8 + 8 + 8 . . . , bei 66 MHz Datenrate)
- **Gleichzeitiges Schreiben/Lesen** auf unterschiedliche Sektoren
- **Sektoren:** 16 x 8 KByte, 30 x 64 KByte
- **Sektor-Löschzeit** 0.5 sec bis 5 sec
- **Programmierzeit** je Wort: 11.5 μ sec bis 210 μ sec
- **Zugriffsschutz** je Sektor mit 8-byte Passwort
- Elektronische **Seriennummer** (16 byte)
- Minimum 1 Mio **Schreibzyklen** je Sektor
- Temperaturbereich -40 . . . 145 Grad
- 20 Jahre **Datenerhaltung** (Retention) bei 125 Grad
- Gehäuse 8 x 9 mm, 64-ball FBGA

Im Trend: Flash-Speicher im Mikrocontroller

System-On-Chip häufig mit integriertem (NOR) Flash-Speicher

Beispiel: **MPC565** mit 2 Bänken zu je 512 KByte

- 10 Jahre garantierter Datenerhalt
- ca. 100 . . . 1000 Schreib/Lösch-Zyklen
- Programmierzeit ist 15 bis 20 μsec je **32-Bit** Wort



NAND-Flash Datenspeicher

- Meist **Flash-Controller** plus **mehrere NAND-Flash** Bausteine in verschiedenen Bauformen (IC-Gehäuse oder Steckkarten)
- Benötigte Techniken im Controller
 - **Bit-Error Correction** (EDC/ECC)
 - **Bad Block Scanning/Handling** – Abschalten defekter Sektoren und Nutzung **redundanter** Speicherzellen
 - **Wear-Leveling** – Gleichmässiges „Abtragen“ der Sektoren



NAND-Flash von der Firma SunDisk

Produkte mit NAND-Flash Speichern



PC-Card (PCMCIA)
Linear/ATA



SD Card



Compact Flash



USB – Drive



Smart Media Card



MultiMedia Card

Die gezeigten Produkte sind von den Firmen *SanDisk* (<http://www.sundisk.com>) und *Hama* (USB-Drive)

Flash Filesysteme

- Aufgabe: Illusion einer **Festplatte**
- Mittler zwischen **Betriebssystem** und meist **NAND-Flash** Bausteinen (auch NOR möglich)
- Software übernimmt **Bad Block Handling** und **Wear-Leveling**
- Beispiele: **TrueFFS** (M-Systems), **FlashFX** (Datalight), **JFFS** (GNU/Linux)



DiskOnChip Bausteine der Firma M-Systems bestehen aus einem Controller und mehreren NAND-Flash Chips. Mit dem Flash-Filesystem TrueFFS kann auf diese Bausteine wie auf eine Festplatte zugegriffen werden.

Zusammenfassung

- **Integrationsdichte** und **Leistungsvermögen** von NOR- und NAND Flash-Speicherbausteinen steigen stetig.
- Wichtige Parameter: **max. Zyklenzahl, Datenerhaltungszeit, Kapazität, Temperaturbereich.**
- **System-On-Chip** mit integriertem Flash erlauben extrem kompakte und kostengünstige Rechner.
- Tendenz zu immer mehr **NAND-Flash** Einsatz, da preiswerter in der Herstellung. Erobert zunehmend klassische NOR-Flash Einsatzgebiete.

Literatur

- [1] Brian Dipert, *Data Storage in a Flash*, EDN Magazine, July 3 1997.
- [2] Brian Dipert, *Inside Flash Memory*, Dr. Dobbs Journal, October 1995.
- [3] Brian Dipert, *Designing with Flash Memory*, Annabooks, San Diego, 1993, ISBN 0-929392-17-5.
- [4] Peter Nangle, *Flash - aber nicht blitzschnell! Ein Vergleich der beiden Flash Technologien NAND und NOR*. Elektronik 6, 1996.

World Wide Web

Advanced Micro Devices (AMD) – <http://www.amd.com>

Intel – <http://developer.intel.com/design/flash>

M-Systems – <http://www.m-sys.com> – NAND-Flash Speicher,
DiskOnChip, File-System Software *TrueFFS*

Datalight – <http://www.datalight.com>

SanDisk – <http://www.sandisk.com>

Silicon Storage Technology – <http://www.sst.com>

Motorola – <http://e-www.motorola.com>

Über den Autor

Hubert Högl ist Diplom-Physiker und promovierte an der Universität Mannheim im Fach Informatik. Nach einigen Jahren als wissenschaftlicher Mitarbeiter am Deutschen Zentrum für Luft- und Raumfahrt (DLR) und anschliessend mehreren IT-Projekten als Freiberufler vertritt er nun an der Fachhochschule Augsburg das Fachgebiet Technische Informatik mit den folgenden Schwerpunkten:

- **Embedded Systems** Hard- und Software
- **Linux** auf Embedded Systems
- **Rechnertechnik** von Microcontrollern und -computern
- Rechnen mit **rekonfigurierbarer Logik**
- **Systemprogrammierung**
- **Open Source** Software

Kontakt

E-mail: Hubert.Hoegl@fh-augsburg.de

Weitere Unterlagen zum Vortrag finden Sie unter

<http://www.fh-augsburg.de/~hhoegl/doc/doc.html>