

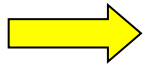
Computer Engineering WS 2012

Speicher

HTM – SHF - SWR



Übersicht



- ▶ Random Access Memory (RAM)
 - ▶ Statisch
 - ▶ Dynamisch
- ▶ Flashspeicher
- ▶ SPI-Flashspeicher



Klassifizierung

- ▶ **Flüchtige Speicher:**
 - ▶ **Verlieren Inhalt, wenn Spannung abgeschaltet wird**
 - ▶ **Beispiele:**
 - ▶ **Statisches RAM** **SRAM**
 - ▶ **Dynamisches RAM** **DRAM**

- ▶ **Nicht flüchtige Speicher:**
 - ▶ **Speicherinhalt bleibt auch nach Abschalten der Versorgungsspannung erhalten**
 - ▶ **Beispiele:**
 - ▶ **Nicht löschbar:** **PROM**
 - ▶ **Löschbar:** **EPROM, Flash**
 - ▶ **Batteriegepuffertes RAM** **NVRAM**



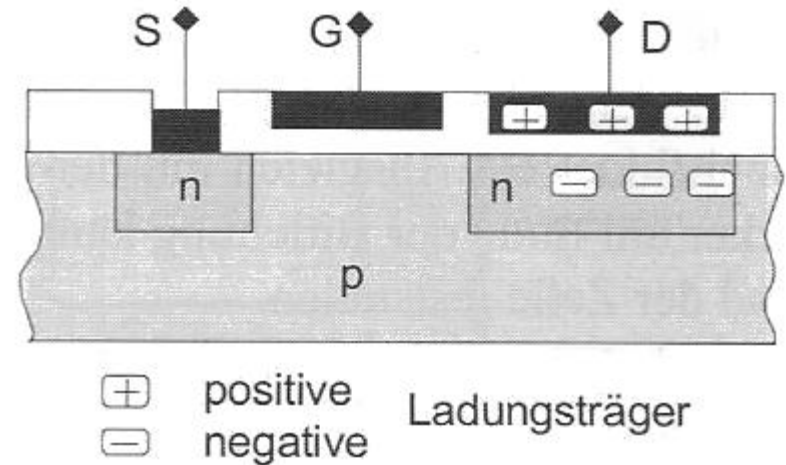
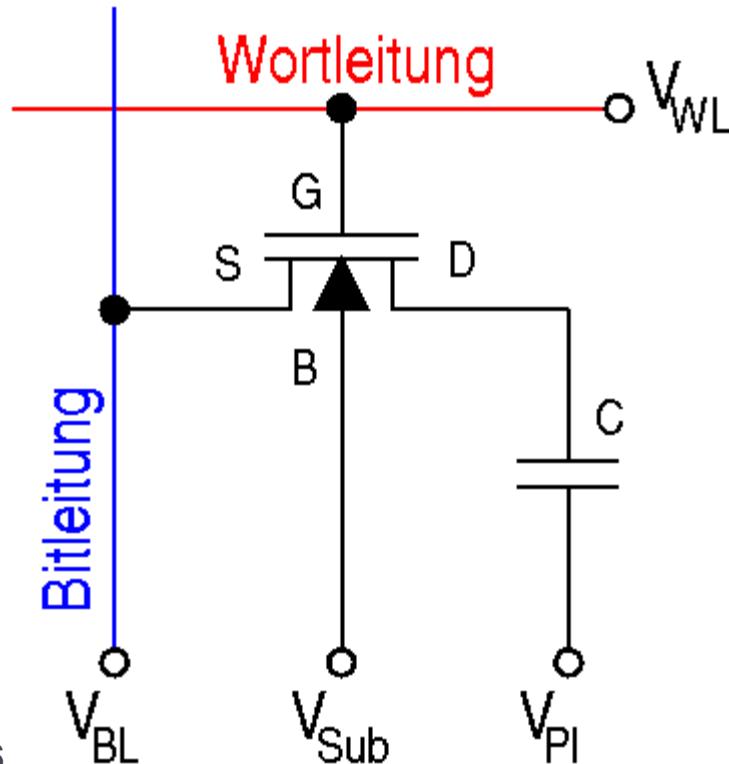
Statisches RAM (SRAM)

- ▶ **Wahlfreier Zugriff auf alle Speicheradressen**
➔ „Random Access Memory“.
- ▶ **Speicherelemente mit Flipflops realisiert:**
 - ▶ Bei anliegender Versorgungsspannung bleibt Zustand dauerhaft erhalten.
- ▶ **Sehr schnell**
- ▶ **Hoher Platzbedarf, dadurch teuer.**
- ▶ **Datenerhaltung benötigt sehr wenig Energie:**
 - ▶ Mit Pufferung über Batterie:
Speicherung bis zu 10 Jahre möglich
- ▶ **Verwendung:**
 - ▶ **Schnelle Speicher, Caches**
 - ▶ **Register**

-

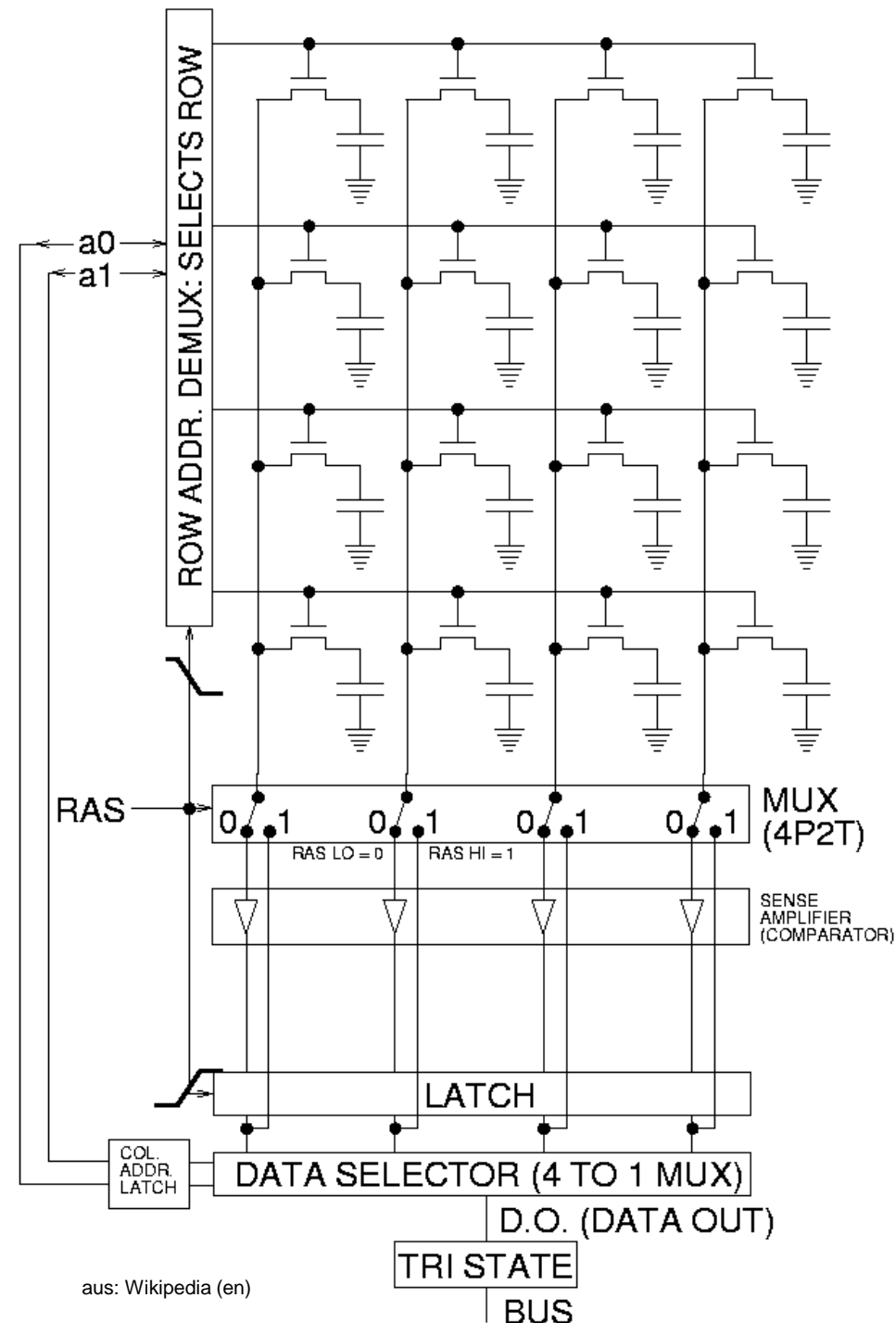
Dynamisches RAM (DRAM)

- ▶ **Speicherung in Kondensatoren**
 - ▶ z.B.: geladen: 1, entladen: 0
- ▶ **Kondensator entlädt sich mit der Zeit**
 - ▶ Refresh notwendig.



DRAM: Prinzip

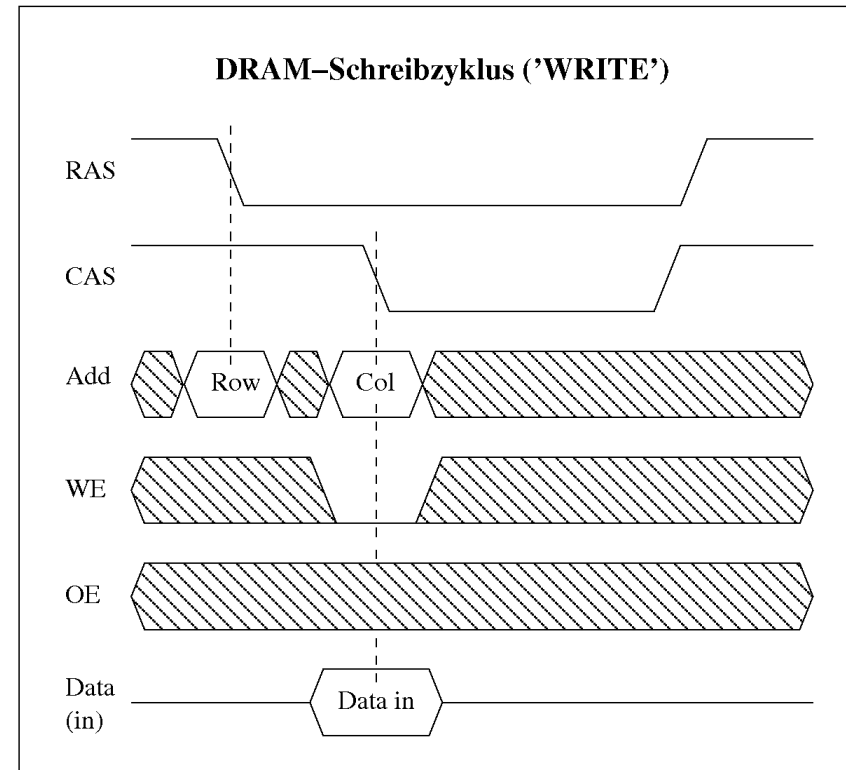
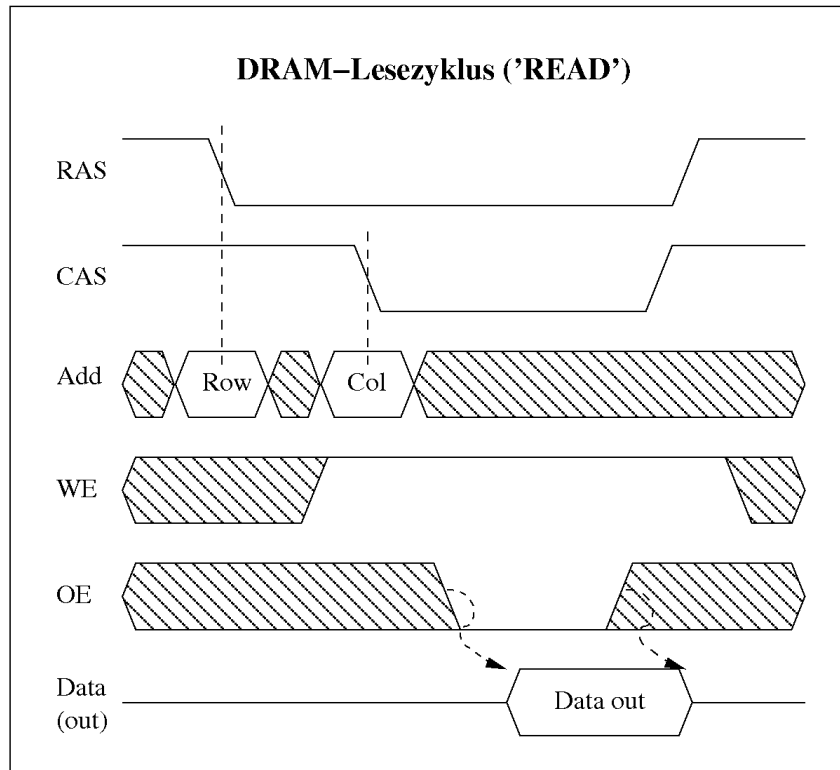
- ▶ Aufteilung der Adressen in
 - ▶ Spaltenadresse:
 - ▶ Auswahl mit CAS
„Column Address Select“
 - ▶ Reihenadresse:
 - ▶ Auswahl mit RAS
„Row Address Select“
- ▶ Lesen:
 - ▶ Übertragen einer vollständigen Reihe ins Latch
 - ▶ Dabei Zerstörung der Daten:
 - ▶ Zurückschreiben erforderlich.
 - ▶ Zugriff auf Bits im Latch sehr schnell:
 - ▶ Unterstützt Burst-Zugriff.





DRAM: Lese- und Schreibzyklus

- ▶ Adressleitungen liefern nacheinander Zeilen- und Spaltenadressen.
- ▶ Zykluszeit beachten:
Zusätzliche Zeit zum Zurückschreiben der Zeile notwendig.





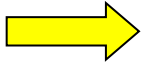
DRAM: Weiterentwicklungen

- ▶ **Fast Page Mode (FPM)**
 - ▶ Zeile bleibt aktiv, es muss nur noch die Spalte gesetzt werden:
 - ▶ Mehrfaches aktivieren von CAS: Burst Mode.
- ▶ **Extended Data Output (EDO)**
 - ▶ Ausgang bleibt aktiv, bis neue Daten vorliegen.
 - ▶ Im Hintergrund kann schon der nächste Zugriff bearbeitet werden.
- ▶ **Synchronous DRAM (SDRAM)**
 - ▶ Zusätzlicher Systemtakt:
 - ▶ Alle Komponenten arbeiten synchron.
- ▶ **Double Data Rate Synchronous DRAM (DDR-SDRAM)**
 - ▶ Übertragung bei steigender und fallender Flanke des Systemtakts
 - ▶ Dadurch doppelter Durchsatz möglich



Übersicht

- ▶ Random Access Memory (RAM)
 - ▶ Statisch
 - ▶ Dynamisch
- ▶ Flashspeicher
- ▶ SPI-Flashspeicher





Technologien

- ▶ **Maskenprogrammiert (Read Only Memory, ROM)**
 - ▶ Programmierung erfolgt bei Herstellung.
 - ▶ Einfach, kostengünstig.
 - ▶ Kann nicht mehr verändert werden.
 - ▶ Gut geeignet für Großserienprodukte
- ▶ **Benutzerprogrammiert (Programmable Read Only Memory, PROM)**
 - ▶ Benutzer kann Speicher einmalig programmieren.
 - ▶ Wegen Programmierlogik Aufwand höher als ROM.
- ▶ **Löschbar (Erasable Programmable Read Only Memory, EPROM)**
 - ▶ Benutzer kann Speicher mehrfach programmieren.
 - ▶ Programmierung nur möglich von „1“ nach „0“.
 - ▶ „1“ nur durch Löschen des gesamten Speichers.
 - ▶ Löschen mit UV-Licht oder elektrisch (**FLASH**)
 - ▶ Geringe garantierte Schreibzyklenzahl (z.B.: 10.000).
- ▶ **Elektrisch löschbar (Electrically Erasable Programmable Read Only Memory, EEPROM)**
 - ▶ Ermöglicht byteweises Löschen.
 - ▶ Hoher Logikaufwand, teuer.
 - ▶ Höhere garantierte Schreibzyklenzahl (z.B.: 1.000.000)
 - ▶ Größerer erlaubter Temperaturbereich zum Programmieren.
 - ▶ Gut für kleine Datenmengen, z.B.: Konfigurationsdaten Microcontroller



Performance

- ▶ Flash-Speicher ist deutlich langsamer als RAM.
- ▶ Wird schnell zum Flaschenhals in einem Microcomputersystem.

Bei niedriger Taktfrequenz:

- ▶ Programme können direkt aus dem Flash heraus ausgeführt werden: XIP (eXecute in Place)
- ▶ Beschleunigung:
 - ▶ Verwendung größerer Wortbreiten:
 - z.B.: 128-Bit bei einer 32-Bit CPU.
Pro Speicherzugriff werden gleichzeitig $4 * 32\text{-Bit}$ Worte gelesen.
 - ▶ Verwendung von Cache Speicher

Bei höherer Taktfrequenz:

- ▶ Programm wird vom Flash ins RAM kopiert und von dort ausgeführt.

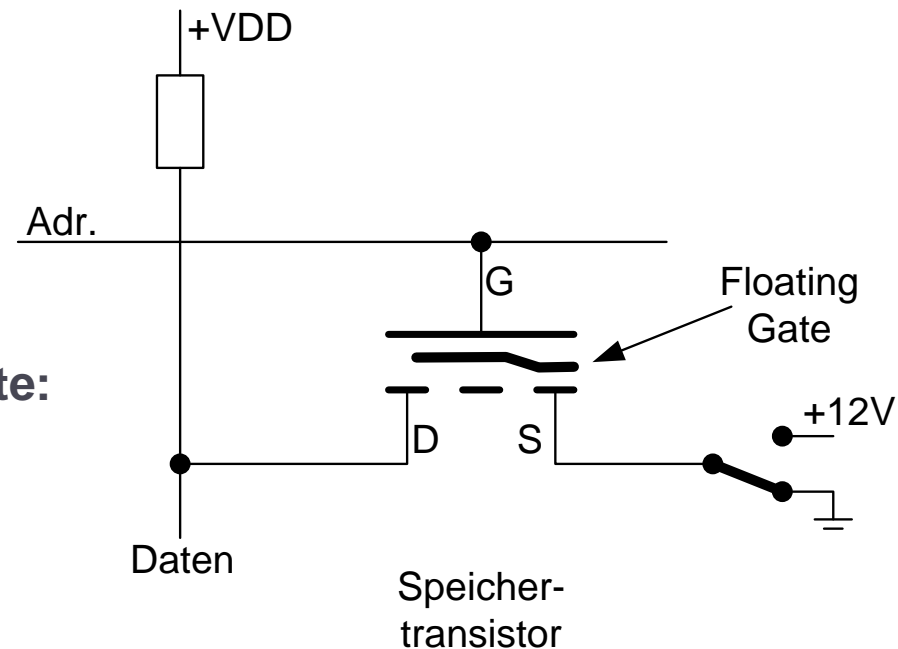


Flash-Speicher Architekturen

- ▶ **NOR-Architektur**
 - ▶ Inhalt kann wahlfrei adressiert werden.
 - ▶ Lesegeschwindigkeit höher.
 - ▶ Gut für Programmspeicher in Mikrocomputersystemen.
- ▶ **NAND-Architektur**
 - ▶ Einfachere Struktur, dadurch kleinere Bitzellen:
 - ▶ höhere Kapazitäten möglich,
 - ▶ preiswerter.
 - ▶ Zugriff nur in Burst von z.B. 512 Bytes möglich.
 - ▶ Schreib- und Löschvorgänge sind wesentlich schneller.
 - ▶ Löscheinheiten.
 - ▶ Programme können nicht direkt aus NAND-Speicher ausgeführt werden.
 - ▶ Gut für Massenspeicher: USB-Stick Memory-Card, Solid State Disk.
 - ▶ Können bereits bei Auslieferung verstreute „Bad Blocks“ haben.

Aufbau von NOR-Bausteinen

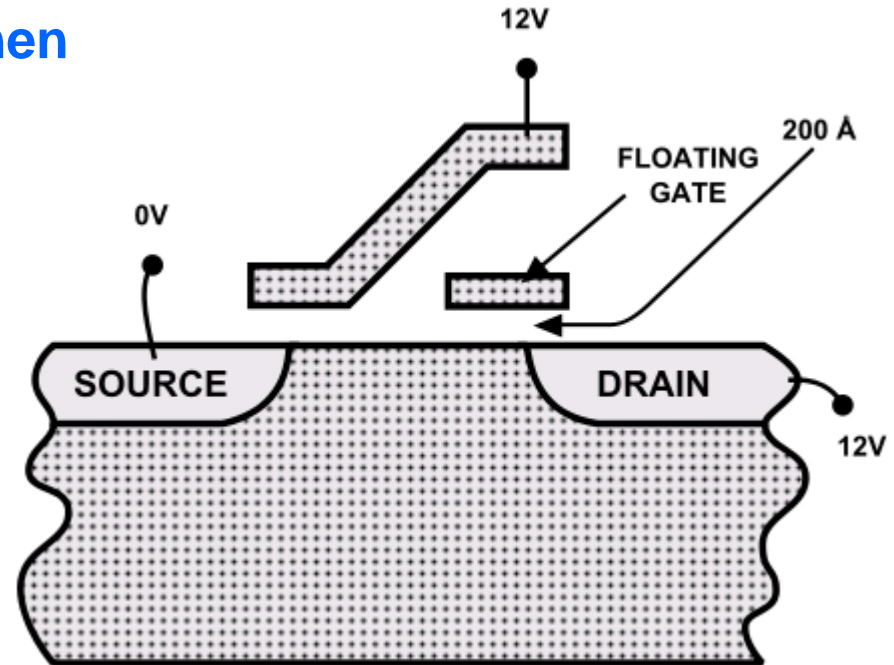
- ▶ Datenspeicherung durch Ladungsträger auf Floating Gate
- ▶ Lesen durch Aktivierung des Gates:
 - ▶ Keine Ladung auf Floating Gate:
 - ▶ Transistor schaltet durch
 - ▶ Datenleitung = 0 V
 - ▶ Ladung auf Floating Gate:
 - ▶ Transistor sperrt
 - ▶ Datenleitung = +VDD



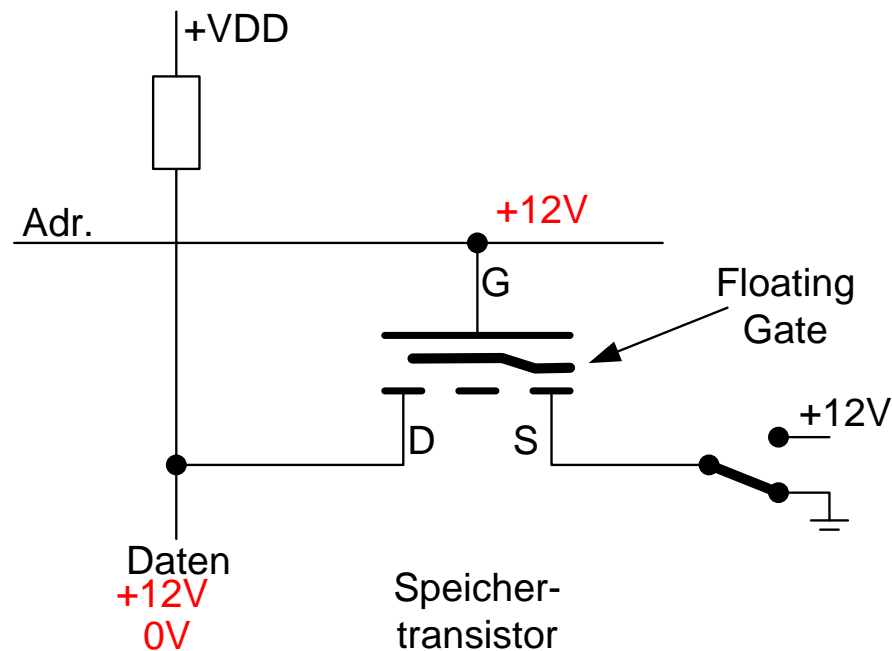
Programmieren von NOR-Bausteinen

Channel Hot Electron Injection

- ▶ Spannung am Steuergate macht Tunnel leitend.
- ▶ Zwischen Source und Drain wird eine hohe Spannung geschaltet.
- ▶ Dadurch hohe Beschleunigung der Elektronen:
 - ▶ Einige Elektronen gelangen dabei auf das Floating Gate.
- ▶ Diese negative Ladung sorgt dann im Normalbetrieb dafür, dass der Tunnel immer gesperrt ist.



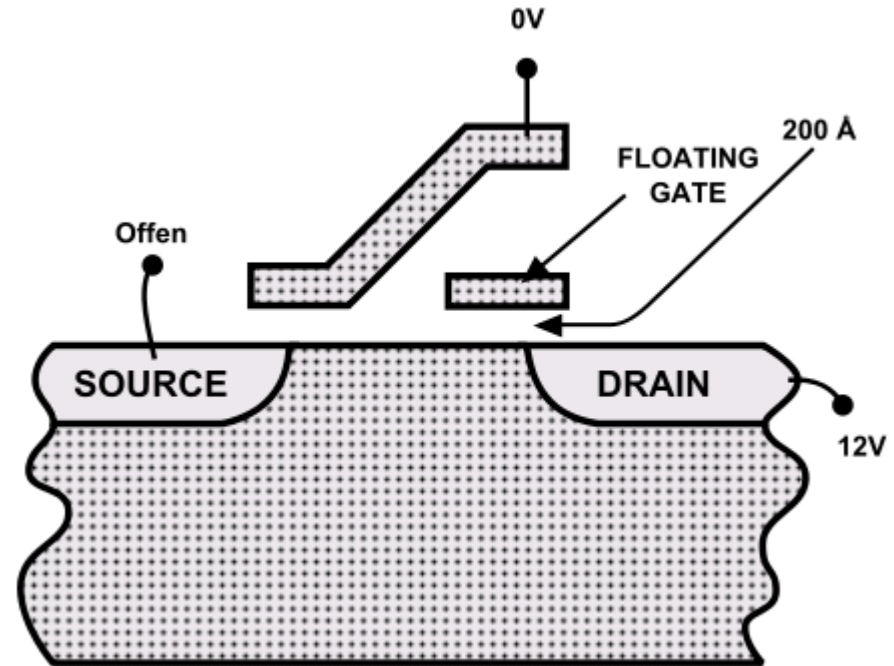
Programmieren von NOR-Bausteinen



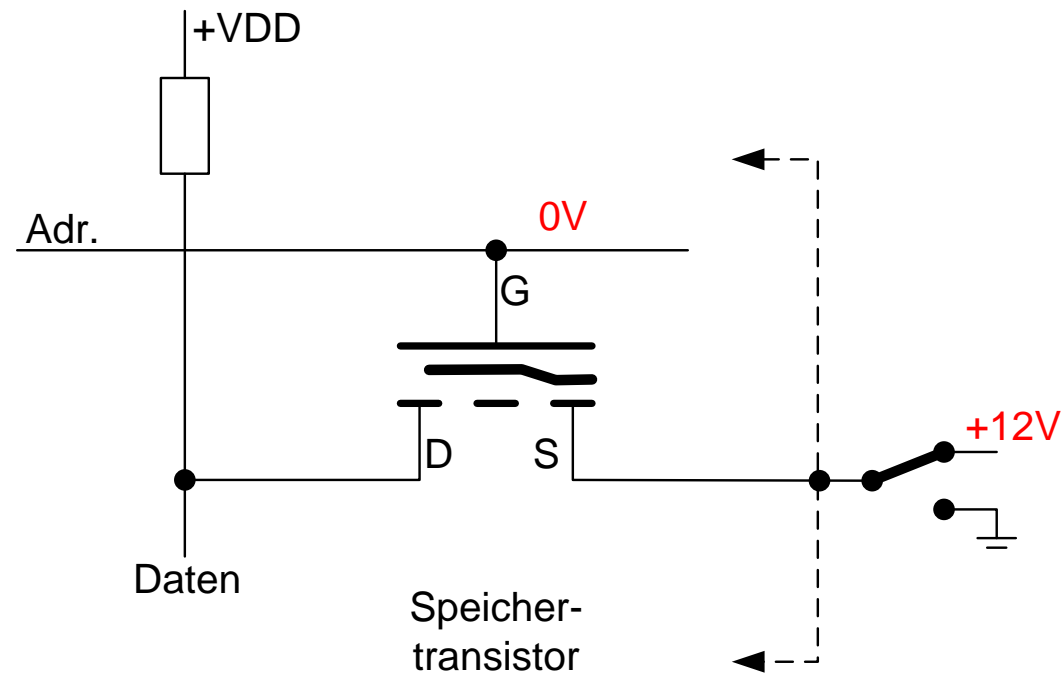
Löschen von NOR-Bausteinen

Fowler-Nordheim tunneling

- ▶ Quantenmechanischer Effekt:
 - ▶ Elektronen „tunneln“ vom Floating Gate zum Drain
- ▶ Negative Ladung auf dem Gate wird abgebaut.
- ▶ Damit kann im Normalbetrieb wieder Strom zwischen Source und Drain fließen.



Löschen von NOR-Bausteinen





Beispiel: AT29LV040A (8 * 512kBits)

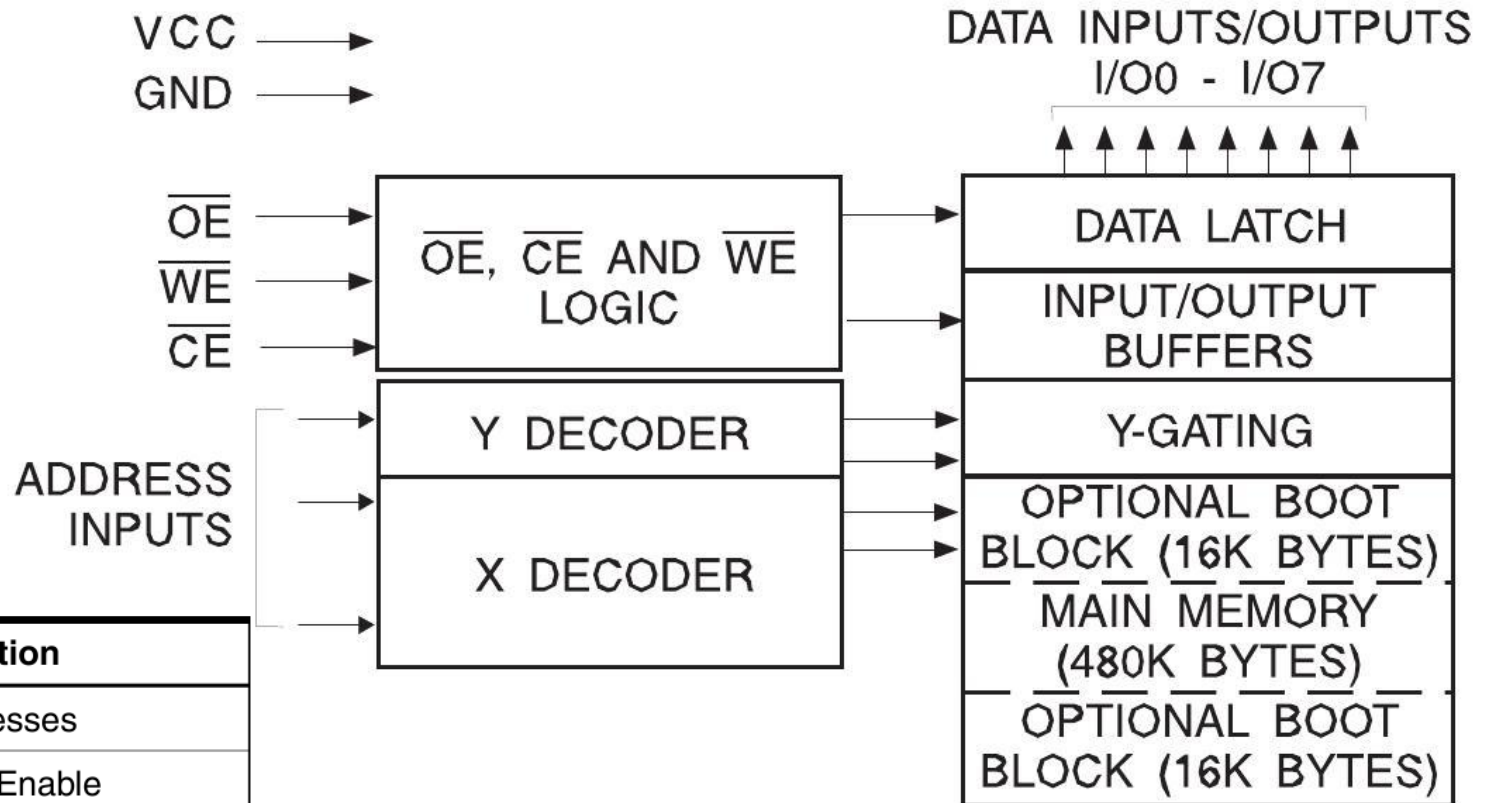
► Features

- **Single Voltage, Range 3V to 3.6V Supply**
- **Software Protected Programming**
- **Fast Read Access Time – 150 ns**
- **Low Power Dissipation**
 - 15 mA Active Current
 - 50 µA CMOS Standby Current
- **Sector Program Operation**
 - Single Cycle Reprogram (Erase and Program)
 - 2048 Sectors (256 Bytes/Sector)
 - Internal Address and Data Latches for 256 Bytes
- **Two 16K Bytes Boot Blocks with Lockout**
- **Fast Sector Program Cycle Time – 20 ms Max**
- **Internal Program Control and Timer**
- **DATA Polling for End of Program Detection**
- 19 ► **Typical Endurance > 10,000 Cycles**



CE WS12

Beispiel: AT29LV040A (8 * 512kBits)

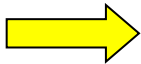


Pin Name	Function
A0 - A18	Addresses
\overline{CE}	Chip Enable
\overline{OE}	Output Enable
\overline{WE}	Write Enable
I/O0 - I/O7	Data Inputs/Outputs
NC	No Connect



Übersicht

- ▶ Random Access Memory (RAM)
 - ▶ Statisch
 - ▶ Dynamisch
- ▶ Flashspeicher
- ▶ SPI-Flashspeicher





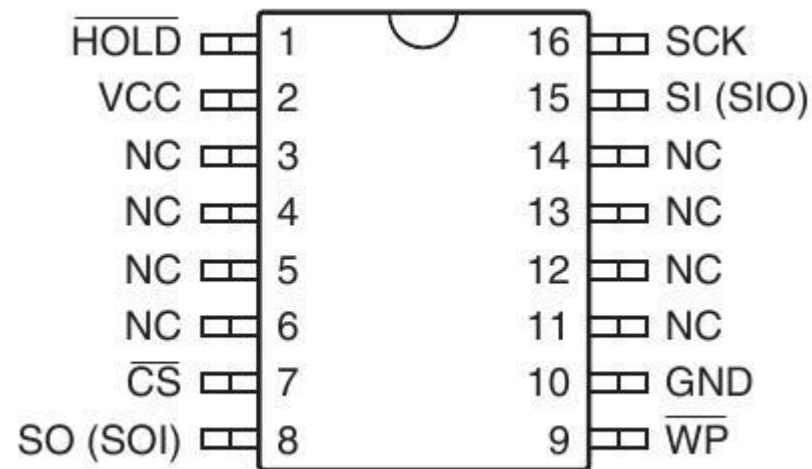
Beispiel: AT25DF641

- ▶ **Eigenschaften**
 - ▶ **Serial Peripheral Interface (SPI) Compatible**
 - ▶ **Very High Operating Frequencies (85 MHz for SPI)**
 - ▶ **Flexible, Optimized Erase Architecture for Code + Data Storage Applications**
 - Uniform 4-Kbyte Block Erase
 - Uniform 32-Kbyte Block Erase
 - Uniform 64-Kbyte Block Erase
 - Full Chip Erase
 - ▶ **Individual Sector Protection with Global Protect/Unprotect Feature**
 - 128 Sectors of 64-Kbytes Each
 - ▶ **Hardware Controlled Locking of Protected Sectors via WP Pin**
 - ▶ **Sector Lockdown : Make Any Combination of 64-Kbyte Sectors Permanently Read-Only**
 - ▶ **128-Byte Programmable OTP Security Register**
 - ▶ **Flexible Programming: Byte/Page Program (1 to 256 Bytes)**
 - ▶ **Fast Program and Erase Times**
 - 1.0 ms Typical Page Program (256 Bytes) Time
 - 50 ms Typical 4-Kbyte Block Erase Time
 - 250 ms Typical 32-Kbyte Block Erase Time
 - 400 ms Typical 64-Kbyte Block Erase Time
 - ▶ **Program and Erase Suspend/Resume**
 - ▶ **Automatic Checking and Reporting of Erase/Program Failures**
 - ▶ **Endurance: 100,000 Program/Erase Cycles**
 - ▶ **Data Retention: 20 Years**



Beispiel: AT25DF641

Anschlüsse





Blockschaltbild SPI-Flashspeicher

► **SCK:**
Serieller Takt

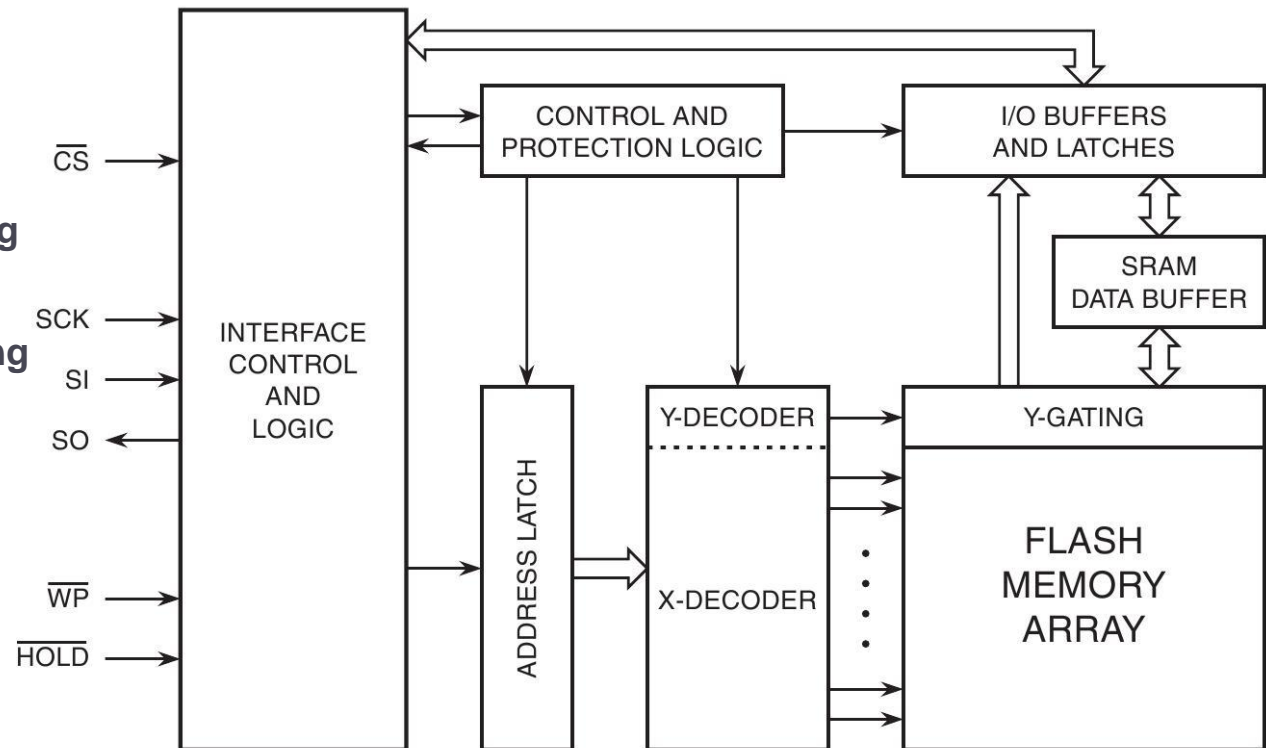
► **SI:**
Serieller Dateneingang

► **SO:**
Serieller Datenausgang

► **CS:**
Chip select

► **WP:**
Write Protect

► **Hold:**
Einfrieren des
Datentransfers





CE WS12

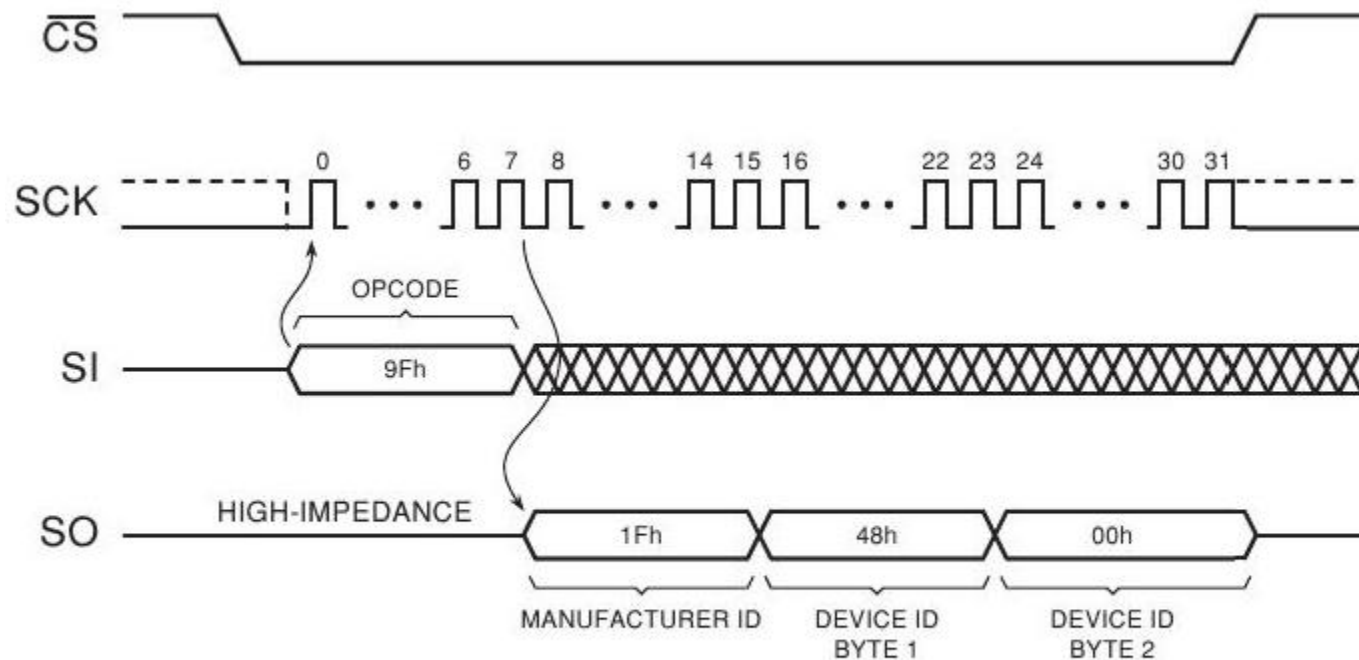
Beispiel: AT25DF641


Kommandos

Command	Opcode		Clock Frequency	Address Bytes	Dummy Bytes	Data Bytes
Read Commands						
Read Array	1Bh	0001 1011	Up to 100 MHz	3	2	1+
	0Bh	0000 1011	Up to 85 MHz	3	1	1+
	03h	0000 0011	Up to 50 MHz	3	0	1+
Dual-Output Read Array	3Bh	0011 1011	Up to 85 MHz	3	1	1+
Program and Erase Commands						
Block Erase (4-KBytes)	20h	0010 0000	Up to 100 MHz	3	0	0
Block Erase (32-KBytes)	52h	0101 0010	Up to 100 MHz	3	0	0
Block Erase (64-KBytes)	D8h	1101 1000	Up to 100 MHz	3	0	0
Chip Erase	60h	0110 0000	Up to 100 MHz	0	0	0
	C7h	1100 0111	Up to 100 MHz	0	0	0
Byte/Page Program (1 to 256 Bytes)	02h	0000 0010	Up to 100 MHz	3	0	1+
Dual-Input Byte/Page Program (1 to 256 Bytes)	A2h	1010 0010	Up to 100 MHz	3	0	1+
Program/Erase Suspend	B0h	1011 0000	Up to 100 MHz	0	0	0
Program/Erase Resume	D0h	1101 0000	Up to 100 MHz	0	0	0
Protection Commands						
Write Enable	06h	0000 0110	Up to 100 MHz	0	0	0
Write Disable	04h	0000 0100	Up to 100 MHz	0	0	0
Protect Sector	36h	0011 0110	Up to 100 MHz	3	0	0
Unprotect Sector	39h	0011 1001	Up to 100 MHz	3	0	0
Global Protect/Unprotect	Use Write Status Register Byte 1 Command					
Read Sector Protection Registers	3Ch	0011 1100	Up to 100 MHz	3	0	1+
Security Commands						
Sector Lockdown	33h	0011 0011	Up to 100 MHz	3	0	1
Freeze Sector Lockdown State	34h	0011 0100	Up to 100 MHz	3	0	1
Read Sector Lockdown Registers	35h	0011 0101	Up to 100 MHz	3	0	1+
Program OTP Security Register	9Bh	1001 1011	Up to 100 MHz	3	0	1+
Read OTP Security Register	77h	0111 0111	Up to 100 MHz	3	2	1+
Status Register Commands						
Read Status Register	05h	0000 0101	Up to 100 MHz	0	0	1+
Write Status Register Byte 1	01h	0000 0001	Up to 100 MHz	0	0	1
Write Status Register Byte 2	31h	0011 0001	Up to 100 MHz	0	0	1
Miscellaneous Commands						
Reset	F0h	1111 0000	Up to 100 MHz	0	0	1
Read Manufacturer and Device ID	9Fh	1001 1111	Up to 85 MHz	0	0	1 to 4
Deep Power-Down	B9h	1011 1001	Up to 100 MHz	0	0	0
Resume from Deep Power-Down	ABh	1010 1011	Up to 100 MHz	0	0	0

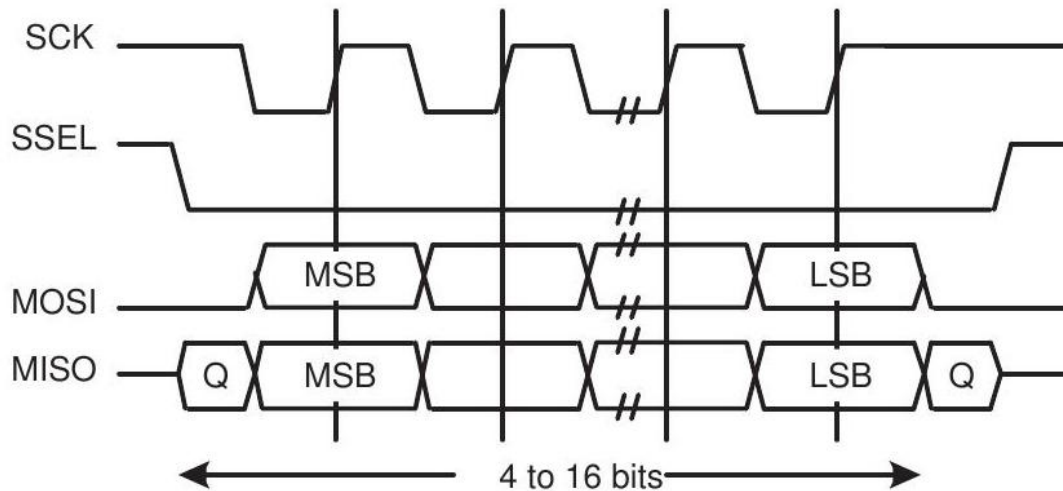
Beispiel: AT25DF641

Kommando: „Read Manufacturer and Device ID“



Note: Each transition  shown for SI and SO represents one byte (8 bits)

SPI-Flashspeicher: Auswahl der SPI-Betriebsart in der CPU



SPI Frame Format with CPOL = 1 and CPHA = 1



Beispiel: AT25DF641

Kommando: „Read Manufacturer and Device ID“

```
GPIO0_IOCLR  = (1<<16);  //SSEL

SSP0DR = 0x9f;           //Opcode
SSP0DR = 0;              //Send 3 dummy bytes
SSP0DR = 0;
SSP0DR = 0;

id=0;
for( j=0; j<4; j++ ){
    while( (SSP0SR & (1<<2)) == 0 ){
    }
    id = (id<<8) | SSP0DR;
}

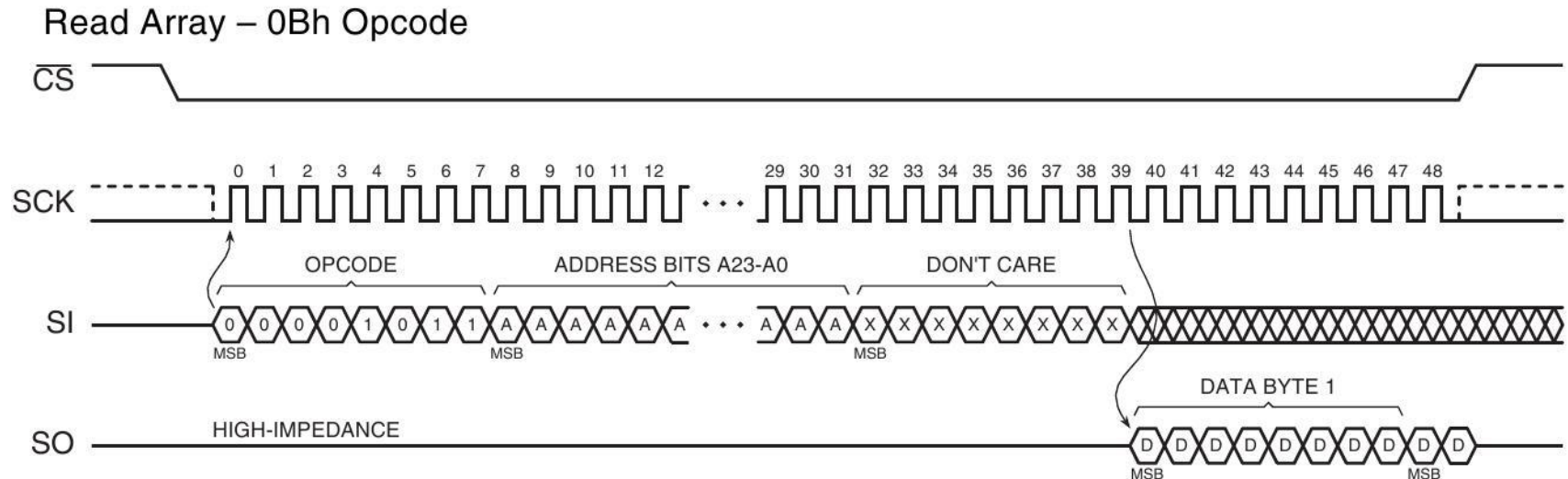
GPIO0_IOSET  = (1<<16);  //SSEL

printf( "id=%08X\n", id & 0xffffffff ); //first byte unused
```



Daten Lesen

- ▶ Unterschiedliche Kommandos in Abhängigkeit von der verwendeten Taktfrequenz und Übertragungsart:
 - ▶ Unterschiedliche Anzahl von „Don't care“-Bytes (Wartezeit).
 - ▶ Single- und Dual-Output (Datenausgabe parallel über SI und SO).
- ▶ Kommando und Adresse müssen nur einmal übertragen werden.
- ▶ Danach beliebige Anzahl Daten.





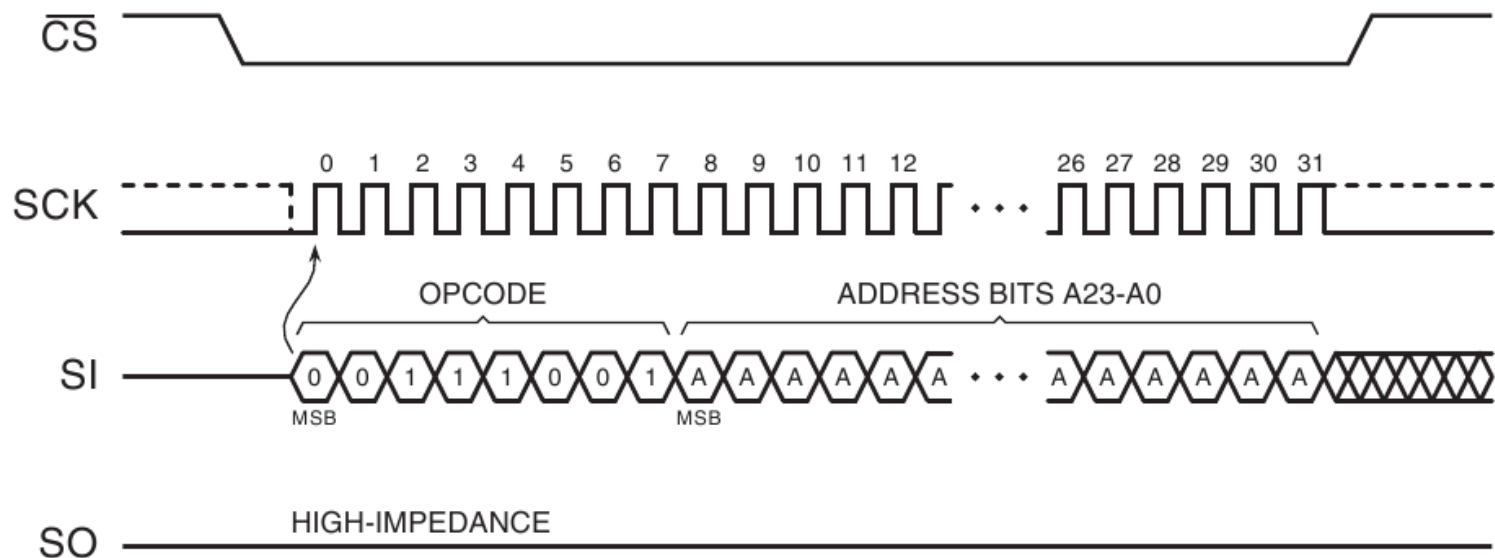
Löschen

- ▶ Löschen kann nur blockweise erfolgen.
- ▶ Blockgröße variabel:
 - ▶ 128 Blöcke à 64 kB oder
 - ▶ 256 Blöcke à 32 kB oder
 - ▶ 2048 Blöcke à 4 kB
- ▶ Ablauf:
 1. Unprotect Sectors:
 - Write Enable
 - Unprotect
 2. Erase Sectors:
 - Write Enable
 - Erase Sectors
 3. Warten:
 - Read Status Register und prüfe RDY/BSY

Internal Sectoring for Protection, Lockdown, and Suspend Functions	64KB Block Erase (D8h Command)	32KB Block Erase (52h Command)	4KB Block Erase (20h Command)	Block Address Range
64KB (Sector 127)	64KB	32KB	4KB	7FFFFFFh – 7FF000h
			4KB	7FEFFFFh – 7FE000h
			4KB	7FDFFFFh – 7FD000h
			4KB	7FCFFFFh – 7FC000h
			4KB	7FBFFFFh – 7FB000h
			4KB	7FAFFFFh – 7FA000h
			4KB	7F9FFFFh – 7F9000h
			4KB	7F8FFFFh – 7F8000h
			4KB	7F7FFFFh – 7F7000h
			4KB	7F6FFFFh – 7F6000h
			4KB	7F5FFFFh – 7F5000h
			4KB	7F4FFFFh – 7F4000h
			4KB	7F3FFFFh – 7F3000h
			4KB	7F2FFFFh – 7F2000h
			4KB	7F1FFFFh – 7F1000h
			4KB	7F0FFFFh – 7F0000h
			4KB	7EFFFFh – 7EF000h
			4KB	7EEFFFFh – 7EE000h
			4KB	7EDFFFFh – 7ED000h
			4KB	7ECFFFFh – 7EC000h
			4KB	7EBFFFFh – 7EB000h
			4KB	7EAFh – 7EA000h
			4KB	7E9FFFFh – 7E9000h
			4KB	7E8FFFFh – 7E8000h
			4KB	7E7FFFFh – 7E7000h
			4KB	7E6FFFFh – 7E6000h
			4KB	7E5FFFFh – 7E5000h
			4KB	7E4FFFFh – 7E4000h
			4KB	7E3FFFFh – 7E3000h
			4KB	7E2FFFFh – 7E2000h
			4KB	7E1FFFFh – 7E1000h
			4KB	7E0FFFFh – 7E0000h
⋮	⋮	⋮	⋮	
64KB (Sector 0)	64KB	32KB	4KB	00FFFFFFh – 00F000h
			4KB	00EFFFFh – 00E000h
			4KB	00DFFFFh – 00D000h
			4KB	00CFFFFh – 00C000h
			4KB	00BFFFFh – 00B000h
			4KB	00AFFFFh – 00A000h
			4KB	009FFFFh – 009000h
			4KB	008FFFFh – 008000h
			4KB	007FFFFh – 007000h
			4KB	006FFFFh – 006000h
			4KB	005FFFFh – 005000h
			4KB	004FFFFh – 004000h
			4KB	003FFFFh – 003000h
			4KB	002FFFFh – 002000h
			4KB	001FFFFh – 001000h
			4KB	000FFFFh – 000000h

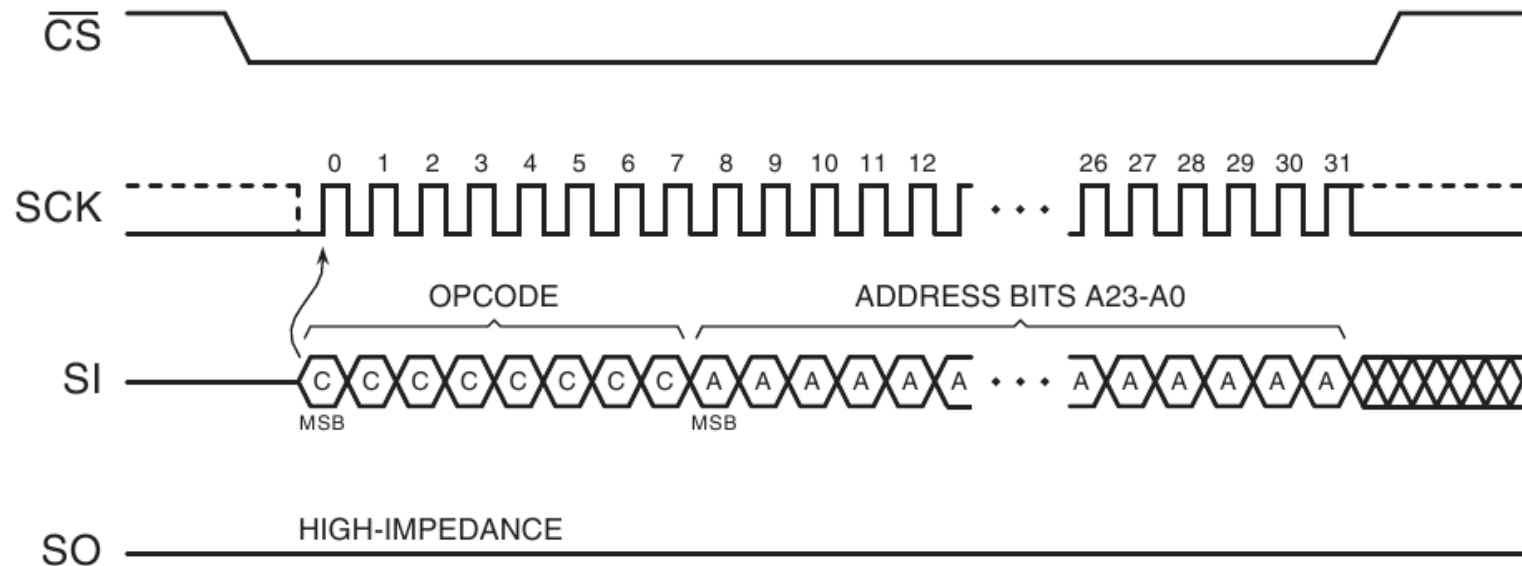
Unprotect Sector

- ▶ Vor Ausführung: „Write Enable“ senden
- ▶ Opcode 0x39



Block Erase

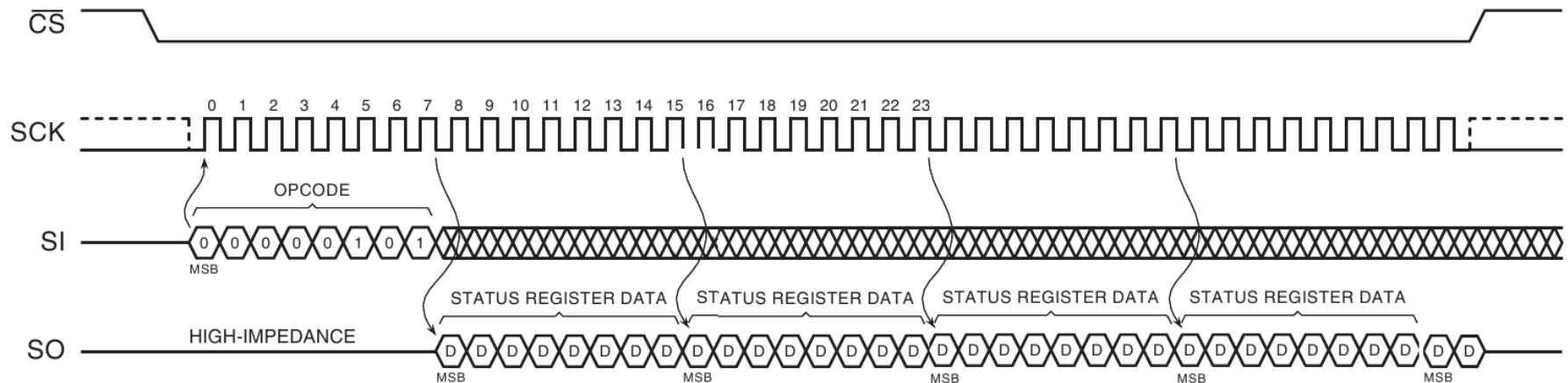
- ▶ Vor Ausführung: „Write Enable“ senden
- ▶ Opcode: 0x20 (4-KBytes), 0x52 (32-kBytes), 0xD8 (64-kBytes)





Auf „Fertig“ warten

- ▶ Kontinuierlich Statusregister lesen.
- ▶ Opcode: 0x05.
 - ▶ Muss nur einmal gesendet werden.
 - ▶ Danach kann der Inhalt des Statusregisters solange ausgelesen werden, bis RDY/BSY == 0.



Programmieren

- ▶ Programmieren von einzelnen Bytes möglich.
- ▶ Pro Programmiervorgang können aber nur Bytes innerhalb eines 256 Bytes Block programmiert werden.
- ▶ Bytes müssen zuvor gelöscht werden:
 - ▶ Gelöschte Bytes haben den Wert 0xFF.
- ▶ Ablauf:
 1. **Unprotect Sectors:**
 - Write Enable
 - Unprotect
 2. **Program Sectors:**
 - Write Enable
 - Program Sectors:
Notwendiger Zeitablauf wird automatisch vom Chip generiert.
 3. **Warten:**
 - Read Status Register und prüfe RDY/BSY

1-256 Byte Page Program (02h Command)	Page Address Range
256 Bytes	7FFFFFFh – 7FFF00h
256 Bytes	7FFEFFh – 7FFE00h
256 Bytes	7FFDFFh – 7FFD00h
256 Bytes	7FFCFFh – 7FFC00h
256 Bytes	7FFBFFh – 7FFB00h
256 Bytes	7FFAFFh – 7FFA00h
256 Bytes	7FF9FFh – 7FF900h
256 Bytes	7FF8FFh – 7FF800h
256 Bytes	7FF7FFh – 7FF700h
256 Bytes	7FF6FFh – 7FF600h
256 Bytes	7FF5FFh – 7FF500h
256 Bytes	7FF4FFh – 7FF400h
256 Bytes	7FF3FFh – 7FF300h
256 Bytes	7FF2FFh – 7FF200h
256 Bytes	7FF1FFh – 7FF100h
256 Bytes	7FF0FFh – 7FF000h
256 Bytes	7FEFFh – 7FEF00h
256 Bytes	7FEFFh – 7FEE00h
256 Bytes	7FEDFFh – 7FED00h
256 Bytes	7FECFFh – 7FEC00h
256 Bytes	7FEBFFh – 7FEB00h
256 Bytes	7FEAFFh – 7FEA00h
256 Bytes	7FE9FFh – 7FE900h
256 Bytes	7FE8FFh – 7FE800h
:	
256 Bytes	0017FFh – 001700h
256 Bytes	0016FFh – 001600h
256 Bytes	0015FFh – 001500h
256 Bytes	0014FFh – 001400h
256 Bytes	0013FFh – 001300h
256 Bytes	0012FFh – 001200h
256 Bytes	0011FFh – 001100h
256 Bytes	0010FFh – 001000h
256 Bytes	000FFFh – 000F00h
256 Bytes	000EFFh – 000E00h
256 Bytes	000DFFh – 000D00h
256 Bytes	000CFFh – 000C00h
256 Bytes	000BFFh – 000B00h
256 Bytes	000AFFh – 000A00h
256 Bytes	0009FFh – 000900h
256 Bytes	0008FFh – 000800h
256 Bytes	0007FFh – 000700h
256 Bytes	0006FFh – 000600h
256 Bytes	0005FFh – 000500h
256 Bytes	0004FFh – 000400h
256 Bytes	0003FFh – 000300h
256 Bytes	0002FFh – 000200h
256 Bytes	0001FFh – 000100h
256 Bytes	0000FFh – 000000h

Programmierkommando

- ▶ **Bestandteile des Kommandos:**
 - ▶ **Opcode 0x02**
 - ▶ **3-Bytes Adresse**
 - ▶ **N Daten**
- ▶ **Falls Datenmenge 256-Bytes Blockgrenze überschreitet:**
 - ▶ **„Wrap-around“ auf Anfang des gleichen Blocks!**
- ▶ **Programmiervorgang beginnt automatisch nach Freigabe der CS-Leitung**

