

Rechnerarchitekturen Semester 4

Inhaltsverzeichnis

- [Rechnerarchitekturen Semester 4](#)
- [Inhaltsverzeichnis](#)
 - [Klausurrelevante Kapitel](#)
 - [Kapitel 1.4 Röhren](#)
 - [Kapitel 2.2 Klassifizierung BS](#)
 - [2.2.1 Bitbreite](#)
 - [2.2.2 64-Bit-Verwechslung](#)
 - [2.2.3 NX-Bit](#)
 - [2.3.4 CPU](#)
 - [2.3.4.2 Aktueller CPU Aufbau](#)
 - [2.3.4.3 Fehlende Adressleitungen](#)
 - [2.3.4.4 Takt und Timing](#)
 - [2.3.6.2 Magnetisch](#)
 - [Kapitel 2.4 Architekturen](#)
 - [2.4.1 Von-Neumann](#)
 - [2.4.2 Harvard-Architektur](#)
 - [Kapitel 2.5 PC-Bussystem](#)
 - [2.5.1 ISA](#)
 - [2.5.2 PCI](#)
 - [2.5.3 PCI-Express](#)
 - [Kapitel 3.3 Rechenwerk](#)
 - [3.3.1 Addition](#)
 - [3.3.1.1 Halbaddierer](#)
 - [3.3.1.2 Volladdierer](#)
 - [3.3.1.3 Paralleladdierwerk](#)
 - [3.3.1.4 Inkrement](#)
 - [3.3.6 Faktor 2 hoch x](#)
 - [3.3.7 Faktor 256 hoch x](#)
 - [3.3.8 MAC](#)
 - [3.3.9 SIS / SIMD](#)
 - [3.3.10 Sättigungsarithmetik \(MMX\)](#)
- [3.4 Steuerwerk](#)
 - [3.4.4 Pipelining](#)
 - [3.4.4.1 Grundprinzip](#)
 - [3.4.4.2 Superskalar](#)
 - [3.4.4.3 Out Of Order Execution](#)
 - [3.4.4.4 Branch Prediction](#)
 - [3.4.4.5 Fazit](#)
 - [5.2.2 Dynamisch \(DRAM\)](#)
 - [5.2.2.1 Standard-DRAM](#)

- 5.2.2.2 EDO
- 5.2.2.3 SDR/DDR/QDR
- 5.3 Nichtflüchtige Speicher (ROM)
- 5.3.1 Allgemeines
 - 5.3.1.1 Maskenprogrammiert (Fuse)
 - 5.3.1.2 Elektrisch Programmierbar
- 5.3.2 EPROM: Bezeichnung 27xxx
 - 5.3.3 EEPROM
 - 5.3.3.2 Parallel: Bezeichnung 28xxx
 - 5.3.3.3 Seriell
- 5.3.4 Flash
 - 5.3.4.1 Allgemeines
 - 5.3.4.2 NOR-Flash
 - 5.3.4.3 NAND-Flash
 - 5.3.4.4 Vergleich NAND-NOR-Flash
 - 5.3.4.5 SLC MLC TLC QLC
 - 5.3.4.6 3D-NAND Flash
- 5.3.5 Modernere Speicherentwicklungen
 - 5.3.5.1 Überblick
 - 5.3.5.2 FRAM
 - 5.3.5.3 MRAM
 - 5.3.5.4 Fazit
- 5.4 Fehlerkorrektur
 - 5.4.1 Softerror
 - 5.4.2 Parity
 - 5.4.3 ECC

Klausurrelevante Kapitel

1.4, 2.2, 2.3.4, 2.3.6.2, 2.4, 2.5, 3.3.1, 3.3.6 bis 3.3.10, 3.4.4, 5.2.2, 5.3, 5.4, 6.2.2, 6.2.8.4, 6.2.9.1, 7.2.1, 7.2.2, 7.3.3

Kapitel 1.4 Röhren

Röhren: Versuch der 40er/50er Jahre, einen PC zu bauen. Bekanntes Modell der US Armee ENIAC, Verwendung von 18.000 Röhren mit einer gesamten Leistungsaufnahme von 174 kW. Jedoch unzuverlässig, da immer Röhren defekt. GB funktionsfähige Maschine in WW2 mit 1500 Röhren, Leistungsaufnahme 4,5 kW

Kapitel 2.2 Klassifizierung BS

2.2.1 Bitbreite

Bitbreite wird durch die Menge des adressierbaren Speichers entschieden. CPU könnte theoretisch mehr adressieren, jedoch Einschränkung durch Busbreite (z.B. 32 Bit).

Gängige Systeme auf dem Markt:

- 16 Bit (max. 64 Kilobyte RAM)
- 20 Bit (max 16 Bit + Segmentierung 1 MB)

- 32 Bit (max 4 GB)
- 64 Bit (max 64 ExaByte = 18 Millionen Terrabyte)

2.2.2 64-Bit-Verwechslung

Jede Speicheradresse ist 64 Bit breit. Adressierung eines Datums dauert z.B. doppelt so lang gegenüber 32 Bit. Vorteil ergibt sich erst, wenn mehr als 4 GB RAM verwendet werden.

Aktuelle Prozessoren verwenden zudem maximal 45 echte Adressleitungen, somit limitiert auf 45 Bit. Maximal anprechbarer Speicher sind damit 256 TB.

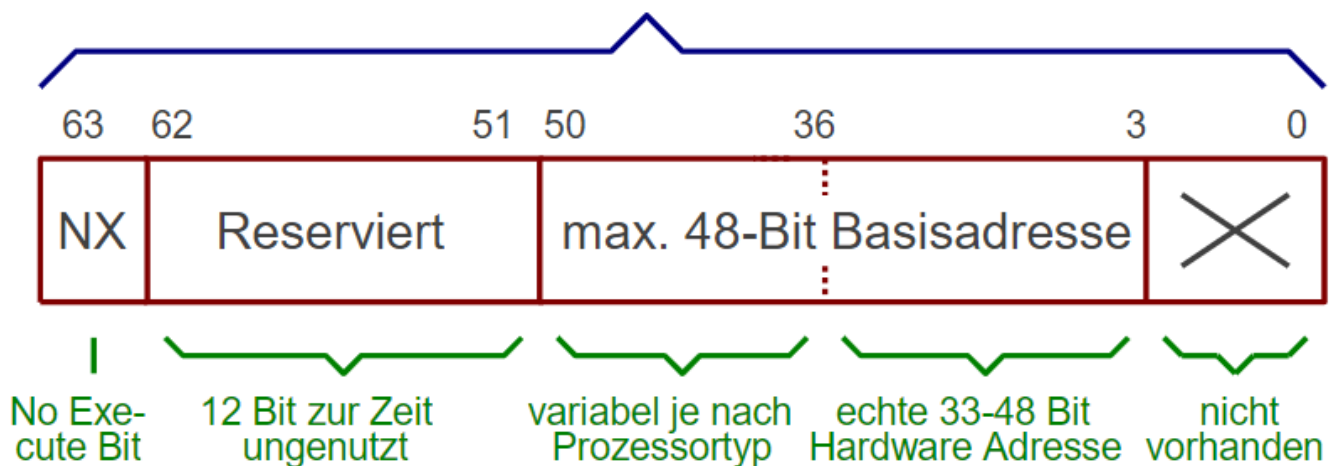
2.2.3 NX-Bit

No-eXecute-Bit

Höchstens Bits einer 64 Bit Adresse selten verwendet. Deshalb Einführung von Sonderbefehle auf Bit 63 einiger Prozessorhersteller.

Verwendung um zu speichern, ob an Adresse Daten oder Programmcode abgelegt ist. Dient dazu es zu erschweren, dass Schadcode in Speicher eingeschleust wird.

Betriebssystem 64-Bit Adresse



2.3.4 CPU

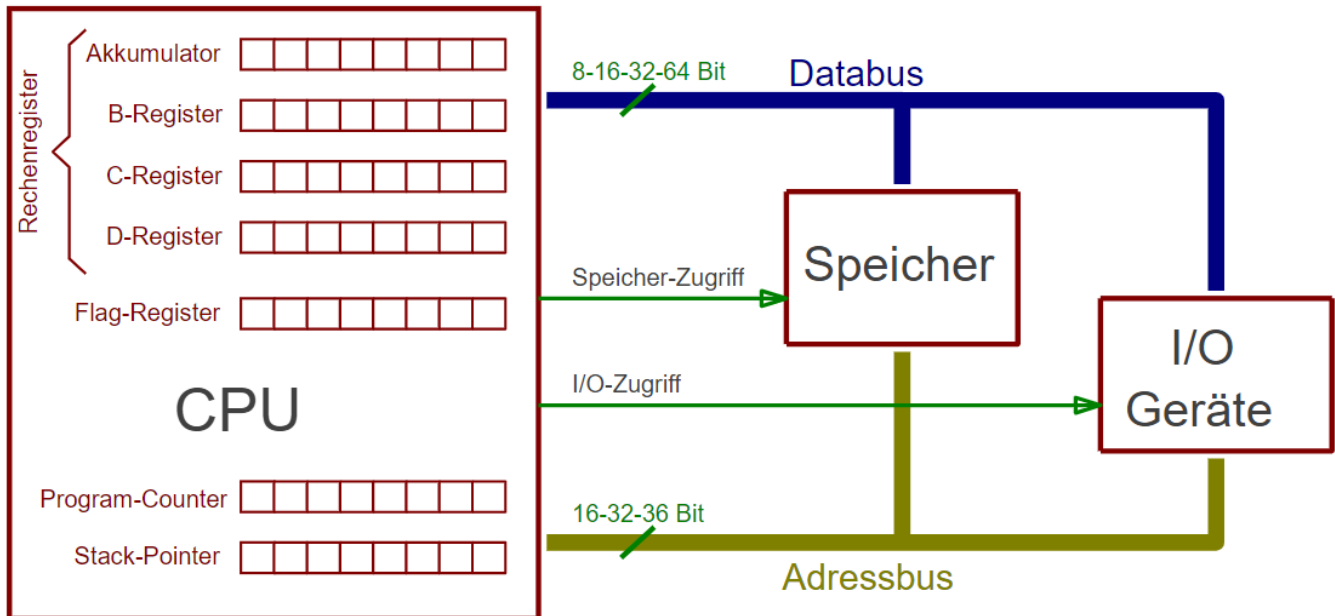
Die CPU (Central Processing Unit) ist zentrales Element in einem Computer.

Mittlerweile in modernen PCs unter CPU noch kleinere Mikrocontroller, die z.B. aktiv sind, bevor eigentliche CPU gestartet ist.

CPU bedient alle Busse:

- Adressbus
- Datenbus
- Steuerbus

je nachdem, welche Aktionen benötigt werden.



Interer Aufbau einer CPU normalerweise mit Registern dargestellt.

Bei alten CPUs immer bestimmtes Register bei arithmetischen oder logischen Funktionen involviert (=Akkumulator). Rechenergebnisse landen immer bei Akkumulator. Bei heutigen CPUs geschieht dies nur noch bei speziellen Befehlen. Mittlerweile können mit jedem Register alle Operationen durchgeführt werden.

2.3.4.2 Aktueller CPU Aufbau

Langsamer Speicherzugriff auf Cache ist trotz DDR3 weiterhin Problem. Lösung durch Versuch des Einbaus nach außen eines 64 Bit Datenbusses. ➡ Dadurch doppelt so schnelle Datenübertragung zwischen CPU und Hauptspeicher.

Bei modernen CPUs ab 2010 existiert Bus Struktur nur noch innerhalb der CPU. Nach außen gibt es jetzt mehrere Datenbusse und Adressbusse. Adressleitungen sind jetzt multiplexed.

2.3.4.3 Fehlende Adressleitungen

Besonderheit bzgl. Adressbussen ab 32-Bit:

- Speicheradressierung immer byteweise
- Bei 32 Bit immer gleich 4 Bytes


➡ die untersten Adressleitungen werden bei 32-Bit Prozessor nicht mehr benötigt, da immer 3 Bytes übersprungen werden

Für einzelne Adressierung der Bytes: Busleitung "Byte Enable" BE0# - BE3# Bei 64-Bit BE0# - BE7#

2.3.4.4 Takt und Timing

Erste Computer genau ein Systemtakt durch Taktoszillator. Takt in CPU von außerhalb entkoppeln.

Takt Steigerung in CPU mittels PLL umgesetzt. Speichercontroller (Cache Controller) in CPU kümmert sich um Entkopplung der unterschiedlich schnellen Busse.

Maximale Taktfrequenz bereits 2000 erreicht mit 200-300 MHz. Für schnellere Übertragung:  Übertragung von 4 statt einem Datenwort

2.3.6.2 Magnetisch

Band Erste Massenspeicher waren Bänder. In Anfangszeit Zweckentfremdung von Audio-Tonspeichern.

Heutzutage immernoch bewährtes Medium zur Speicherung von Backups.

Zugriff auf Bänder findet sequentiell statt.

- Nachteil: Zugriffsgeschwindigkeit gering, im Minutentbereich
- Vorteil: Bei Virenbefall nicht alle Daten direkt verfügbar: Verbreitung verlangsamt.

Laufwerkstyp	Speichergroße
LTO-1	100 GB
...	...
LTO-8	12 TB

Diskette

Zwischenlösung zwischen Magnetbandspeicher und Magnetplatte. Ähnlichkeiten mit Magnetband jedoch Vorteil des wahlfreien Zugriffs.

- Direkter Kontakt Schreiblesekopf und Medium
- starker Verschleiß und geringe Lebensdauer
- Entwicklung zu immer kleiner und höherer Speicherdichte

Übliche Größe	Eingeführt	Speicher-Kapazität
8 Zoll	1970er	80 kByte bis 256 kByte
5,25 Zoll	1980er	360 kByte bis 1,2 MByte
3,5 Zoll	1980er	720 kByte bis 1,4 MByte

Platte

Funktionsweise wie Bänder oder Disketten jedoch Schreiblesekopf schwebend über Medium. Der dadurch entstehende Luftwirbel der Rotation der Platte herrscht sorgt hierfür. Wenn Platte ausgeschaltet wird, fällt Kopf in eine Landing Zone, in der keine Daten gespeichert sind.

Vorteile:

- hohe Rotation und dadurch kurze Zugriffszeiten
- hohe Übertragungsrate

Hauptwartezeit hängt von Umdrehungsgeschwindigkeit ab, da Kopf im Mittel eine halbe Umdrehung warten muss, bis gesuchte Daten vorbei kommen.

Innenraum einer Festplatte ist mit staubfreiem Gas gefüllt. Staubpartikel würde zu Headcrash führen.

Speicherentwicklung kontinuierlich verbessert.

- Große Verbesserung durch GMR-Effekt (Giant Magneto Resistance): quantenmechanischer Effekt mit dem Zweck einen kleineren Lesekopf zu konstruieren. Ab 1995 konnte diese Technik in Platten genutzt werden.
- Weitere Steigerung 2008 durch Magnetisierung des Schreibkopfes
- 2013 SMR (Shingled Magnetic Recording) um mehr magnetische Bits auf eine Platte zu bekommen. Bei Magnetisierung einzelner Bits musste immer Sicherheitsabstand eingehalten werden. Schreibkopf ist größer als Lesekopf. Beim Lesen der nun kleineren Magnetzonen macht dies kein Problem, weil der wesentlich kleinere GMR-Lesekopf keine Probleme damit hat.

Aktuelle Festplatten speichern zwischen 2 Lücken aktuell ca. 40 MByte an Daten.

Ausblick auf zukünftige Generationen

Ausnutzung aller in der Vergangenheit eingeführten Technologien liefern maximale Größe von 18 TB.

- HAMR (Heat Assisted Magnetic Recording): Laser erhitzt zu schreibendes Material. Dadurch sinkt benötigte magnetische Feldstärke.
- MAMR (Microwave Assisted Magnetic Recording): direkt bei Schreibkopf Material aufgeweicht durch elektromagnetische Mikrowellenstrahlung.

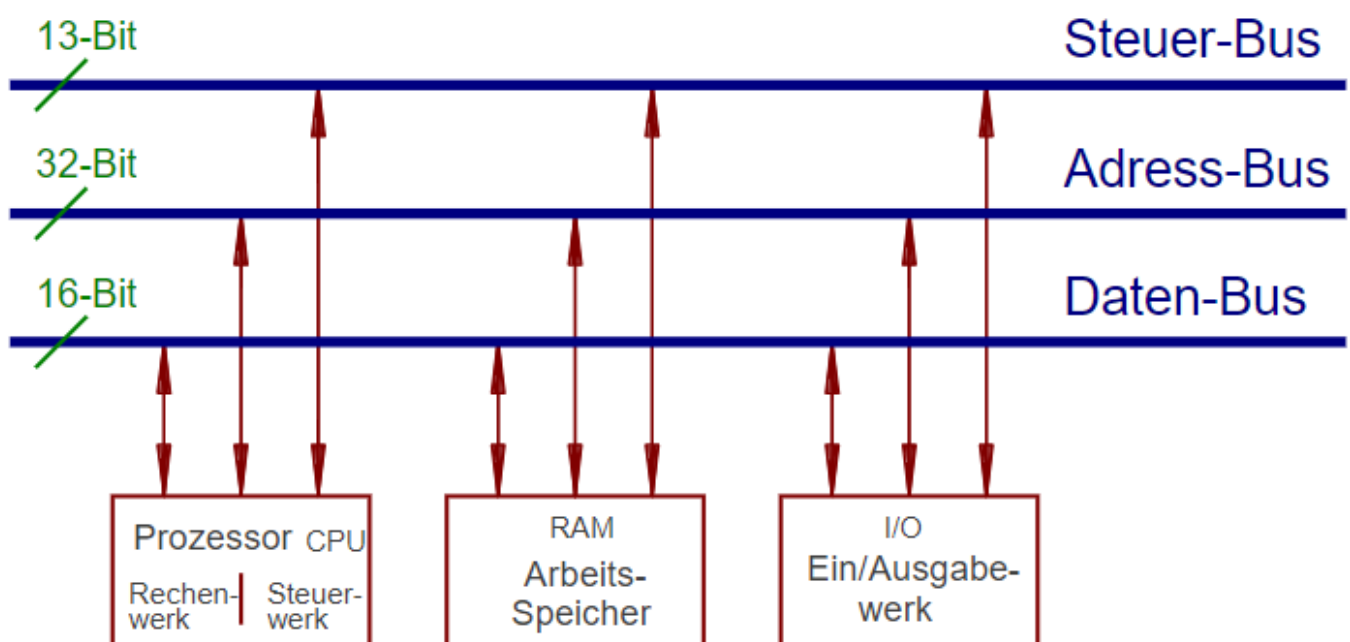
Kapitel 2.4 Architekturen

2.4.1 Von-Neumann

Architektur, nach der fast alle modernen PCs aufgebaut sind. Erster Computer (Zuse Z1) hatte Von-Neumann Struktur.

Prinzip: es existiert nur ein einzelnes Bussystem. Dies können Daten oder Programmcode sein.

Nachteil: Bus wird abwechselnd für Speicher und Daten verwendet. Es wird nur eine Sorte Arbeitsspeicher benötigt, der für alles zuständig ist.



Problem damals: Daten und Programmcode haben sich gegenseitig aus Cache "geworfen" (Cache-Trashing).

Lösung durch Einführung von:

- Cache für Daten
- Cache für Programmcode

Vorgehen ähnelt CPU intern der [Harvard-Architektur](#)

Nachteil der Von-Neumann Architektur:

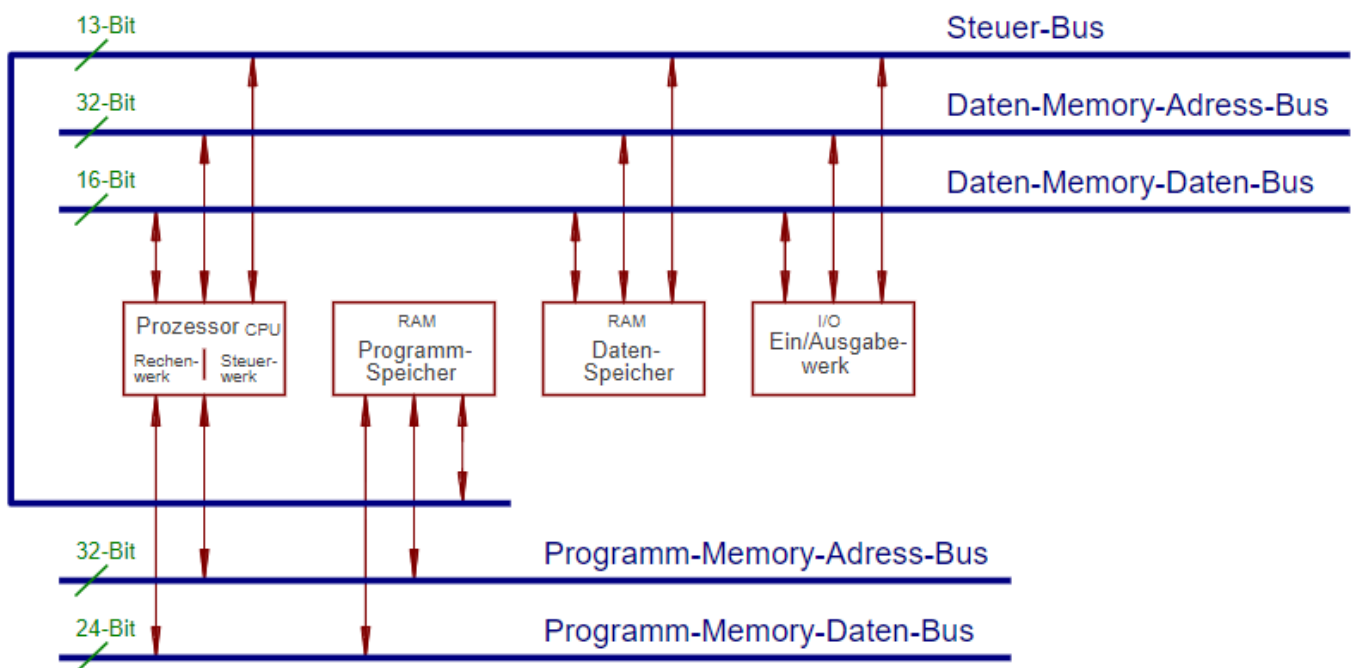
Software kann ihren eigenen Programmcode verändern. Schadsoftware kann dies als Möglichkeit der Ausbreitung nutzen.

2.4.2 Harvard-Architektur

Entwicklung durch IBM und Harvard-Universität 1944.

Wichtigstes Merkmal:

Physikalisch getrennte Speicher und Busse für Programmcode und Daten.



Dadurch 2 Vorteile:

- Zugriff auf Speicher bei Befehlsabarbeitung doppelt so schnell
- Nahezu unmöglich, dass Programm eigenen Code überschreibt.

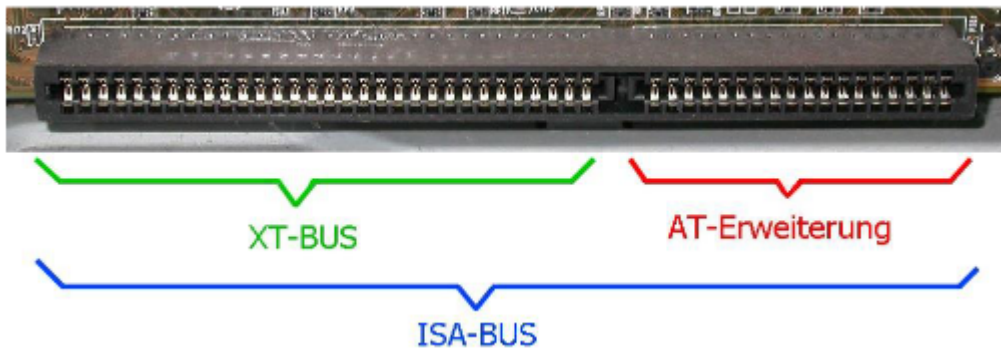
Programmspeicher ist bei normaler Ausführung Nur-Lese-Speicher. Bei Embedded Systemen oft sogar in ROM-Speicher abgelegt.

Modifizierte Version (Modified-Harvard) z.B. als Cache in modernen Systemen implementiert.

Kapitel 2.5 PC-Bussystem

2.5.1 ISA

Erster Standard Bus für PCs war der von IBM in den 1980ern entwickelte ISA-Bus (Industry Standard Architecture). Vorerst als XT-Bus mit 8-Bit-Datenbusbreite und 4,7 MHz. Wenig später 16 Bit mit 8,33 MHz.



Der gesamte Systembus wird direkt auf den ISA-Stecker geführt. Spätere Entwicklung ISA-BUS mit 32 Bit hat sich nicht durchgesetzt.

Für Industrieanwendungen werden noch ISA-Slot-Systeme verwendet (Altlasten) durch ISA-Hardware-Emulation.

2.5.2 PCI

Einführung PCI-Bus (Peripheral Component Interconnect) Mitte 90er Jahre. BUS hat keine direkte Verbindung mehr zu CPU. Verbindung stattdessen über Chipsatz des Prozessors.

PCI-Bus trennt Daten und Adressleitungen nicht, sondern Multiplexing. Adresse und Daten haben beide 32-Bit, Taktfrequenz 33 MHz. Übertragungsrate im [Burst-Modus](#) maximal 133 Mbyte/s.

Später Entwicklung einer 64-Bit-Variante (PCI-X) mit Stecker doppelter Größe, hauptsächlich für Serverbetrieb, jedoch kein Erfolg auf dem Markt.

2.5.3 PCI-Express

Bei Entwicklung paralleler Bussysteme entsteht Problem, dass alle Signale auf allen Leitungen zwischen allen Komponenten gleich lang unterwegs sein müssen. Abhilfe durch künstliche Verlängerung von Leiterbahnen mittels Mäander (Schlaufen zur künstlichen Verlängerung).



Diese Methode begrenzt jedoch Datendurchsatz. Steigerung nur möglich durch Aufgabe des parallelen Datendurchsatzes zu serielllem Durchsatz.

PCIe (ca. ab 2003) Ersetzung des Busses durch serielle Punkt zu Punkt Verbindung (Lane).

Taktfrequenz bei PCIe-1 2,5 GHz, Verdopplung der Übertragungsrate von 32-Bit-PCI.

Zusammenschalten von bis zu 16 Lanes um noch mehr Daten gleichzeitig zu übertragen. Die so übertragenen Daten treffen nicht gleichzeitig an Ziel ein und müssen wieder korrekt Zusammengesetzt werden.

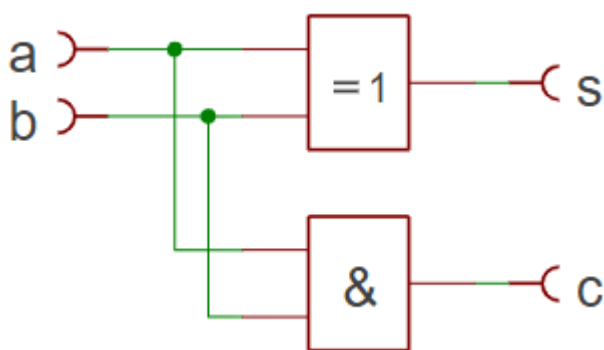
PCIe ist für Software nicht sichtbar, da die parallel-seriell-parallel Wandlung direkt von Hardware übernommen wird.

Kapitel 3.3 Rechenwerk

3.3.1 Addition

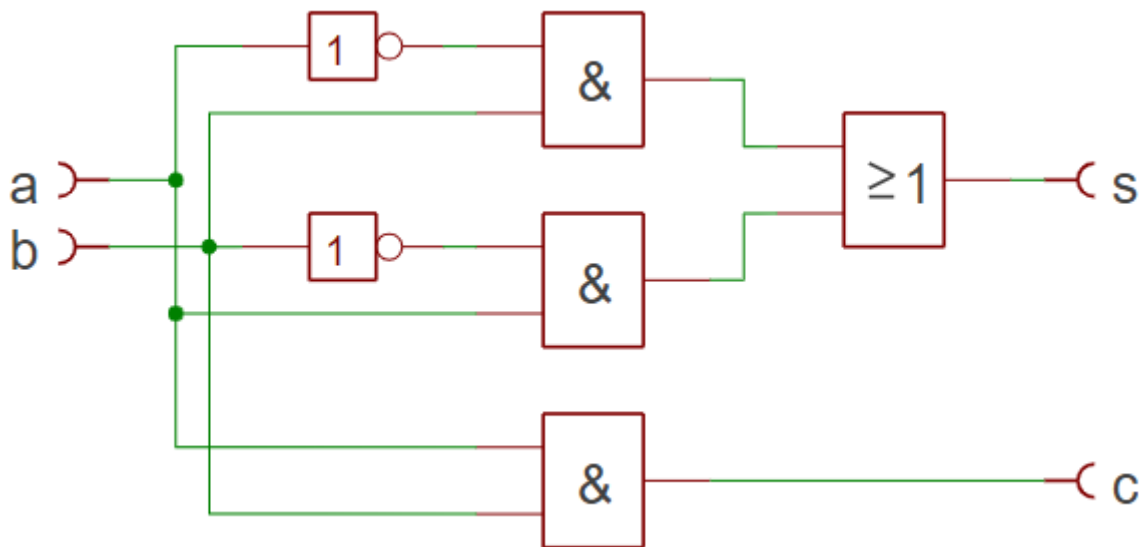
3.3.1.1 Halbaddierer

Aufgabe ist Addition von einstelligen Dualzahlen



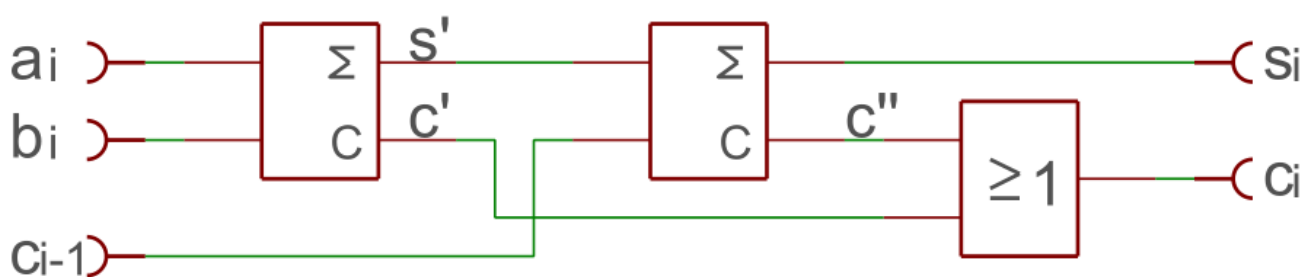
Wahrheitstabelle

a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

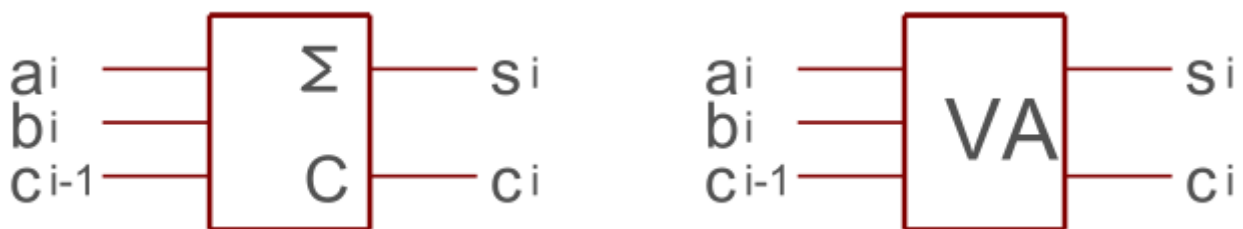


Formel für S: $s = (\text{not } a \text{ and } b) \text{ or } (a \text{ and not } b)$ Formel für C: $c = a \text{ or } b$

3.3.1.2 Volladdierer



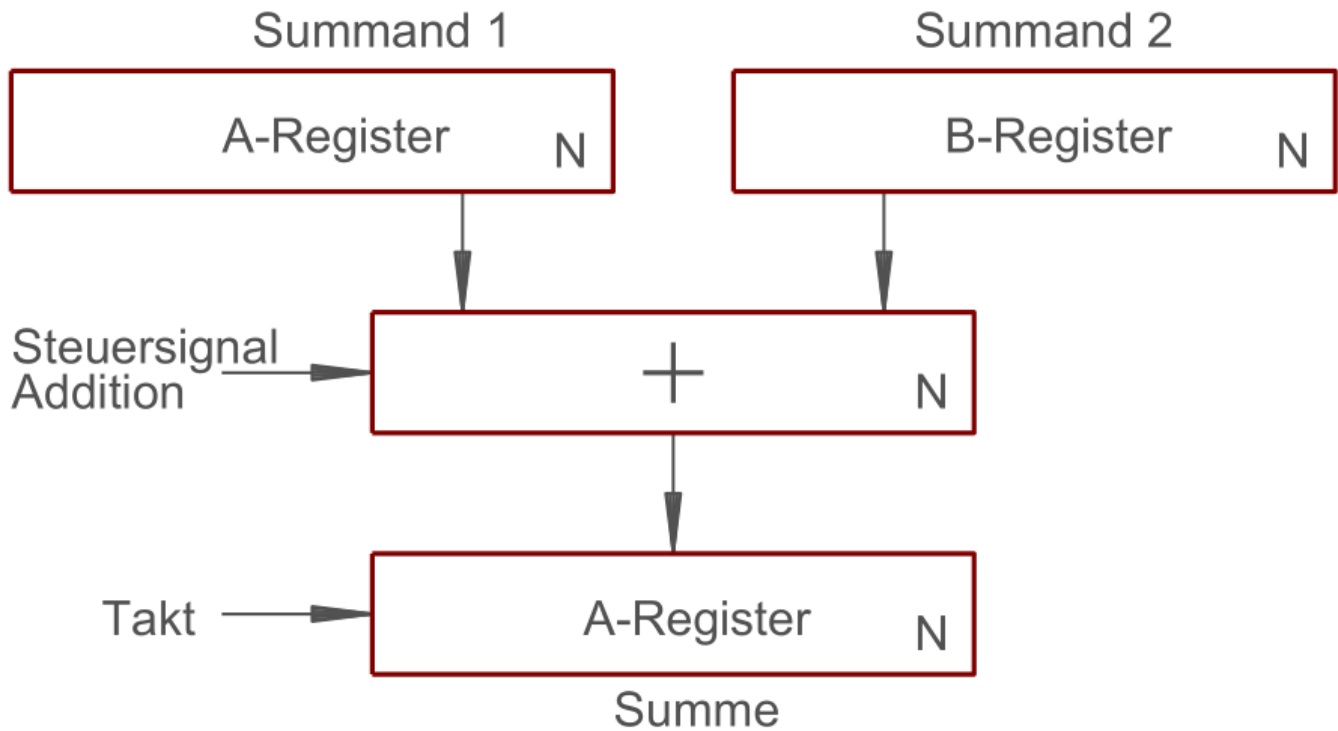
Gatter Symbol



3.3.1.3 Paralleladdierwerk

Kompletzte Addition zweier Register:

- Sumanden in Registern mit der Breite N
- Steuersignal in Addition aktiviert das Paralleladdierwerk
- Mit nächster Taktflanke wird Ergebnis dann in das A-Register übernommen



Für eine Subtraktion, wird bei Subtrahend (B-Register) noch ein Zweierkomplement zwischengeschaltet

Ripple Carry Addition mehrstelliger Zahlen, wird pro Dualstelle ein Volladdierer benötigt

Carry Look Ahead

3.3.1.4 Inkrement

Inkrementierung ist eine sehr häufig benötigte arithmetische Funktion.

- Bei Schleifenberechnung ist das Inkrementieren der Zählervariable eine typische Aktion

Wenn nur um 1 inkrementiert wird, kann jeder Volladdierer durch einen Halbaddierer ersetzt werden, außer der des Least Significant Bit (LSB).

3.3.6 Faktor 2 hoch x

Division oder Multiplikation mit dem Faktor 2^x

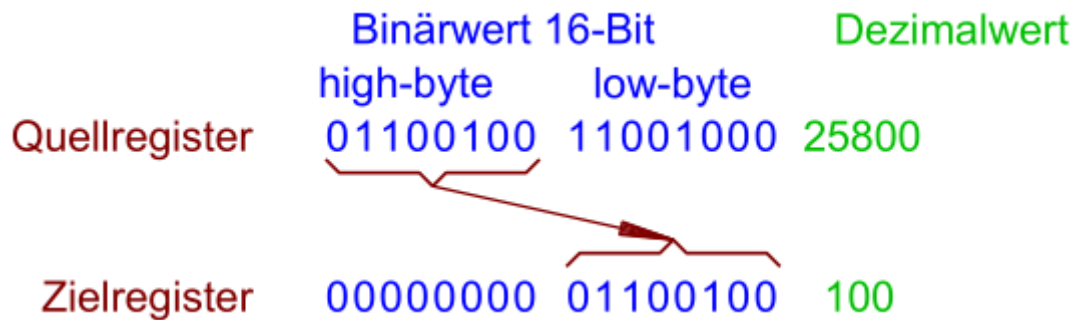
- Entspricht Shift-Operation um 1 Bit

	Binärwert	Dezimalwert
Links Shift ↑	1 100 1000	200
	0 1100 100	100
	0 01100 10	50
	0 001100 1	25
Rechts Shift ↓		

Viele COMpiler erkennen diese Funktion und setzen 2^x damit automatisch durch SHift-Befehl um. Dauer üblicherweise nur 1 Taktzyklus

3.3.7 Faktor 256 hoch x

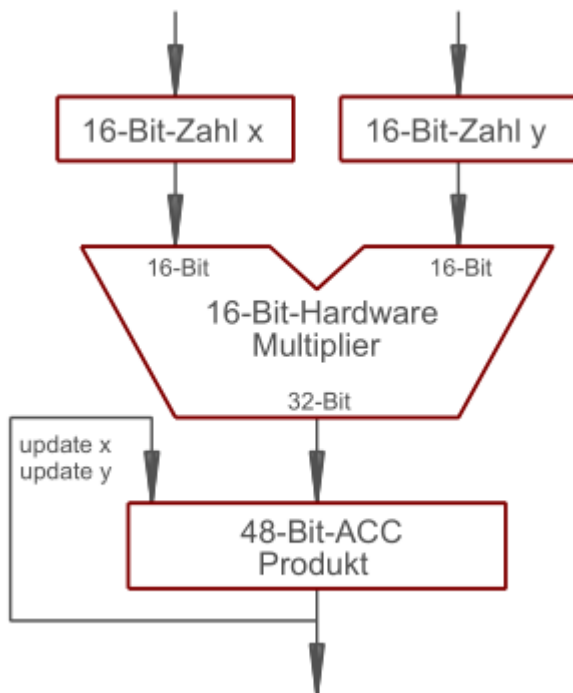
Fast alle modernen Compiler führen in diesem Fall keine Operation aus



Adresszugriff der multiplizierenden variable um 1 Byte

3.3.8 MAC

Für Berechnung von Fourir-Transformation $z = z + (x \cdot y)$



MAC-Einheit (Multiply ACcumulate)

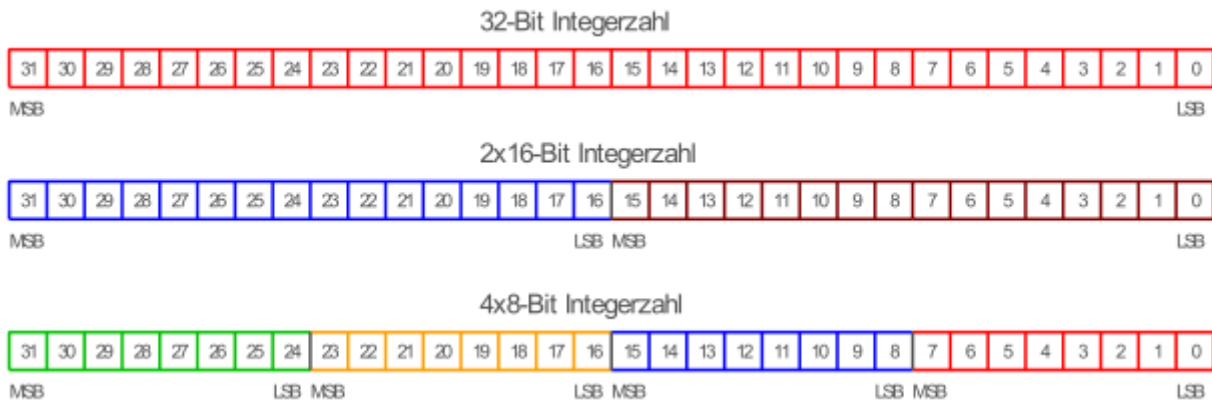
- Multiplikation ist oft breiter als die Ergebnisbreite des Multipliers
- Die Eingangswerte x, y des Multipliers können bei den meisten Signalprozessoren mit automatischem Zeiger auf nächsten Multiplikator weitergeschaltet werden.

3.3.9 SIS / SIMD


Normal kann in einem Register der CPU nur eine Operation auf den gesamten Registerinhalt angewandt werden SISD (Single Instruction Single Data).

- Bei 16-Bit-Operationen 75% des 64-Bit-Registers ungenutzt

- Um Registerbreite voll zu nutzen: (Single Instruction Multiple Data): Register in mehrere Teile geteilt, Einzelteile werden wie separate Register behandelt
- Bei 64-Bit bis zu 8x 8-Bit Berechnung



3.3.10 Sättigungsarithmetik (MMX)

MMX  (Matrix Math Extensions oder Multi Media Extensions) Zuständig für Verschnellerung von Signalverarbeitungen

- Verhinderung mögliches Über-/unterlaufs
- Ergebnis bleibt auf dem Größt-/Kleinstmöglichen Registerwert stehen

Fallbeispiel 8-Bit-Sättigungsarithmetik

ohne Sättigung $\$187 + 175 = 106 + \$\text{Carry-Bit} = 106 + 256 = 362\$$




mit Sättigung $\$187 + 175 = 255\$$

3.4 Steuerwerk

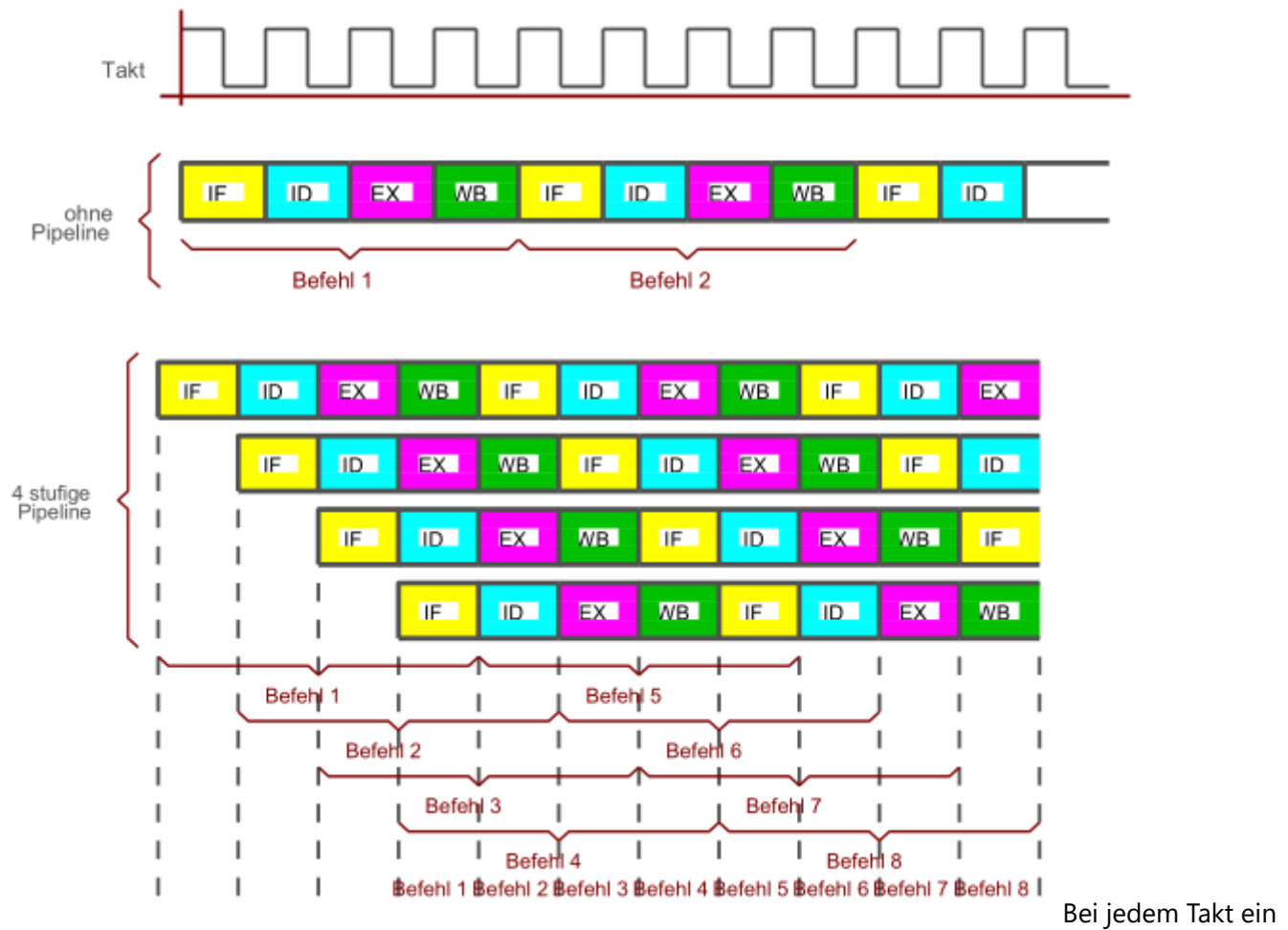
3.4.4 Pipelining

3.4.4.1 Grundprinzip

Prozessoren in unterschiedliche Verarbeitungsklassen unterteilen

- nicht skalar  weniger als 1 Assemblerbefehl pro Taktzyklus
- skalar  1 Assemblerbefehl pro Taktzyklus
- superskalar  mehr als 1 Assemblerbefehl pro Taktzyklus

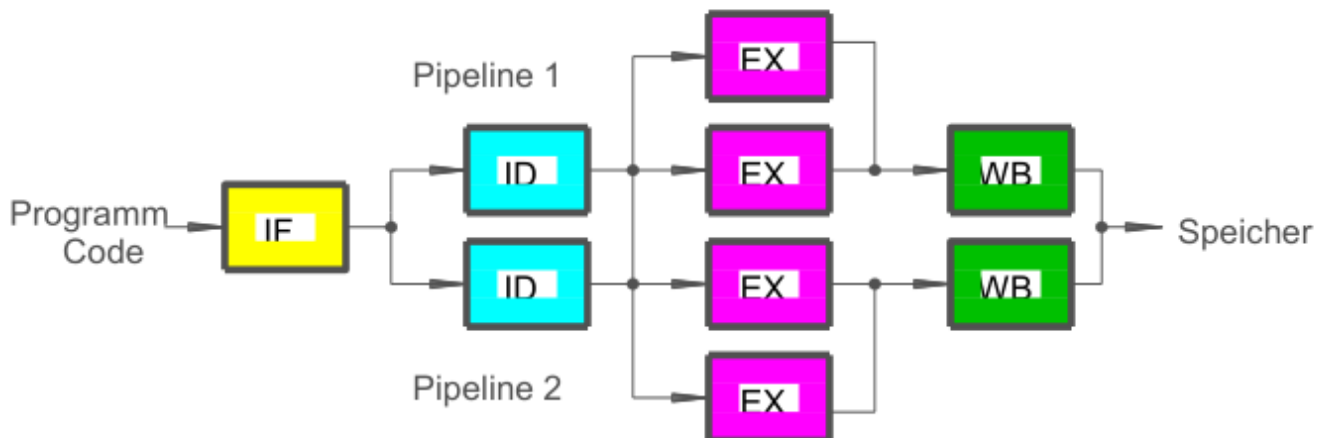
1. Befehl holen (IF, Instruction Fetch)
2. Befehl dekodieren (ID, Instruction Decoding)
3. Befehl ausführen (EX, EXecution)
4. Ergebnis wegschreiben (WB, Write Back)



neuer Befehl

3.4.4.2 Superskalar

Weiterer Entwicklungsschritt: Gesamte Pipeline kann doppelt ausgeführt werden.



Verwendung mehrerer Pipelines und Execution Units

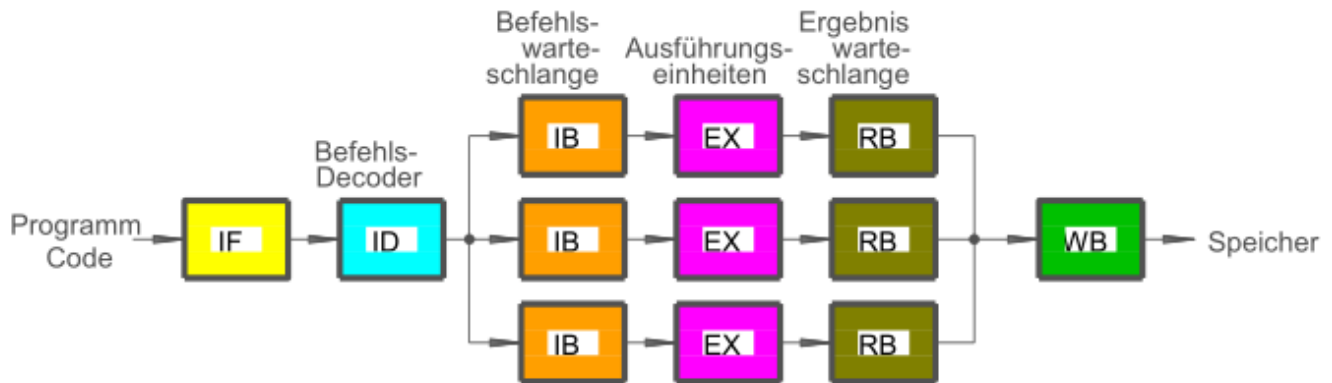
(bei Intel ab 1993 mit dem Pentium ➡ U und V-Pipe)

3.4.4.3 Out Of Order Execution

Gewisse Befehlsabfolgen lassen sich schlecht mit einer oder mehreren Pipelines parallel verarbeiten Lösung

➡ Vertauschen von Befehlsreihenfolgen ➡ bessere Ausnutzung von Pipelines

Befehle werden in Warteschlange (Instruction Buffer) eingereiht. Befehlsdecoder prüft mit Heuristik, ob Befehl vorgezogen werden kann, wenn Execution Unit frei ist



Nach Ausführung werden Ergebnisse in Re-order-Buffer eingereiht und in Ursprungs-Reihenfolge zurück an Speicher oder Ergebnisregister übergeben.

3.4.4.4 Branch Prediction

Zur Optimierung von urspr. Programmcode ➡ in modernen Prozessoren weitere Einheit: Sprungvorhersage-Einheit (Branch-Prediction-Unit)

- in Zusammenarbeit mit der Out-Of-Order-Unit Vorhersage für wahrscheinlichsten Zweig
- Bei Vorhersage Ausführung von spekulativem Code. Bei Fehler muss er wieder verworfen werden
- Sehr aufwändige Technologie ➡ Verbrauch von viel Energie

3.4.4.5 Fazit

Zusammenspiel von:

- mehrfachen Pipelines
- mehrfachen pipelineübergreifenden Execution-Units
- Out-Of-Order-Execution
- Branch-Prediction-Unit
- Speculative Execution

sorgt für Performance Steigerung

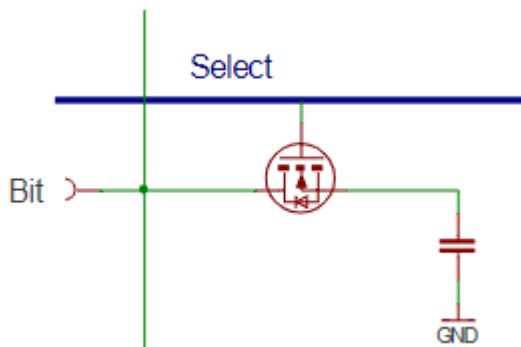
Alle Optimierungen Zusammen bilden Sicherheitslücke (Stichwörter: Spectre, Meltdown).

5.2.2 Dynamisch (DRAM)

5.2.2.1 Standard-DRAM

DRAM-Speicher (Dynamic Random Access Management) basiert auf einem kleinen Kondensator, in dem Bit Wert mittels elektrischer Ladung gespeichert wird.

- Vorteil: nur ein Transistor zur Bit Speicherung nötig



- Größe heutzutage im Gigabit Bereich
- Kondensator entlädt sich von selbst
- Vor jedem Verlorengehen muss Inhalt ausgelesen werden
- Zyklisches Auslesen im Refresh Zyklus

Darstellung des DRAM-Speichers in einem 2-Dimensionalen Array


- Wortleitungen
- Bit-Leitungen

Zugriff auf Kondensator erfolgt über:

- RAS (Row-Adress-Select, Zeilenadresse)
- CAS (Column-Adress-Select, Spaltenadresse) unterteilt



Zugriffdauer heutzutage im Bereich 40-60 nS mit Zugriffstaktfrequenz von 16-15 MHz


Wen nur ein Adressteil verwendet wird  Zugriff doppelt so schnell

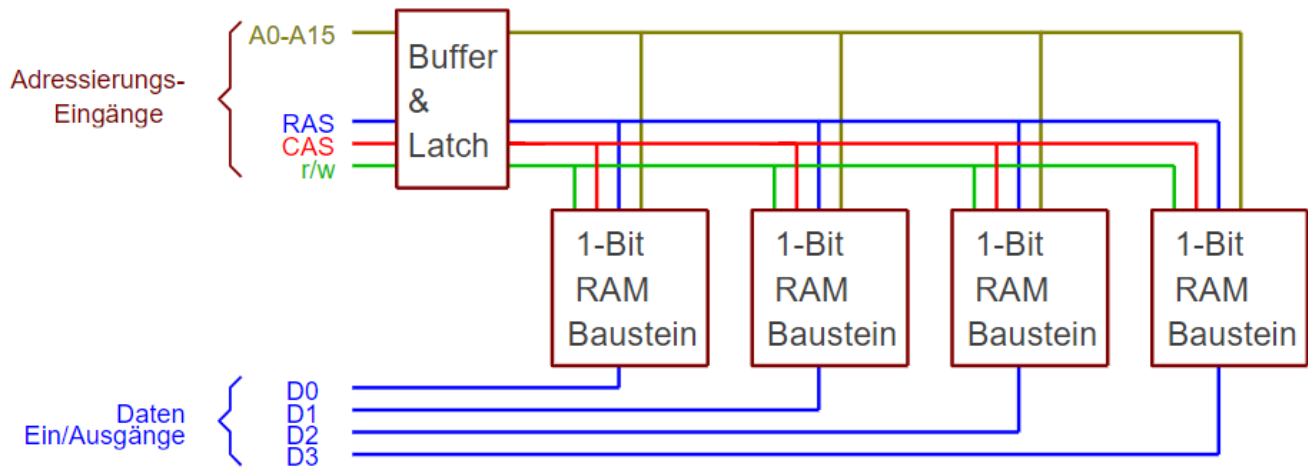
RAM-Module

- Parallelschaltung der Adressierungseingänge werden bei mehreren DRAM-Bausteinen eingesetzt
- Für jedes Bit des RAM-Moduls war eigener Baustein zuständig

Heutzutage 4-Bit

Parallelschaltung

- Führt bei manchen Rechner-Designs zu einer Überlastung des Adressbusses
 - Kann verhindert werden, indem vor Adressleitungen der Bausteine auf RAM-Modul ein zusätzlicher Buffer geschaltet wird  Bufferd-RAM
 - Wenn zusätzlich noch Zwischenspeicher (Latch) für Adresse vorgeschaltet wird, spricht man von Registered-Ram



5.2.2.2 EDO

- Adressierungsphase dauert gewisse Zeit
- Bis nach Abschluss dieser Phase sind Daten auf dem Adressbus ungültig
- Diese Wartezeit kann anderweitig genutzt werden:
 - Einführung eines extra Buffers in die Datenausgänge der RAM-Bausteine (Enhanced Dynamic Output)
- Während Adressierungsphase ist in Datenbuffer noch das vorherige Datenwort gespeichert und liegt auf Datenbus
 - ➡ Verschachtelter Adressierung möglich

Nachteil von EDO-RAM:

- Asynchrone Arbeitsweise
- Zeiten zwischen den RAS und CAS nicht gleichmäßig, jeder RAM-Baustein ist anders
 - ➡ der langsamste Baustein bestimmt Schreib-/Leserate
 - Einbeziehung von Sicherheitsreserve für Schwankungen durch z.B. Temperaturunterschiede

5.2.2.3 SDR/DDR/QDR

SDR

- Einführung eines gemeinsamen Takts für gesamten RAM-Baustein
 - Alle Steuersignale beziehen sich auf Taktsignal

Prozessoren schreiben/lesen durch zwischengeschalteten Cache Speicher im Burst Mode

- Lesen von immer 16 Bytes hintereinander

Speicherbausteine wurden insofern optimiert, dass Speicherzugriffe auf aufeinanderfolgende Adressen schnell stattfinden


Durch:

- Pipelining
- Vervielfachung der Speicherbänke
- mehrere Buffer innerhalb des RAM-Bausteins

Verdopplung Geschwindigkeit gegenüber EDO-RAM

DDR/QDR

Synchrone Ausgabe von SDRAM schnell wieder zu langsam. Deshalb:

- Pro Taktperiode Ausgabe von 2x Datenworten statt 1x  DDR (Double Data Rate)

Standard	Zugriffe
DDR2	2 Zugriffe
DDR3	3 Zugriffe
DDR4	4 Zugriffe

 Daten im Voraus bereithalten (Prefetch) QDR-SDRAM (Quad Data Rate Synchronous Dynamic Random Access Memory)

5.2.2.4 Fazit


Zugriffszeit des ersten wahlfreien Zugriffs heute immer noch 40-60 nS. Maximale Zugriffstaktfrequenz 16-25 MHz.

 Problem umgehen durch Einführen eines schnelleren Cache Speichers zwischen RAM und CPU

5.3 Nichtflüchtige Speicher (ROM)

5.3.1 Allgemeines


5.3.1.1 Maskenprogrammiert (Fuse)

Speicherbausteine erster Computer war ROM (Read-Only-Memory), mittlerweile fast ausgestorben  Vorteil durch schnelle Zugriffszeit

5.3.1.2 Elektrisch Programmierbar

- Information eines Bits wird in Floating Gate eines Feldeffekttransistors gespeichert
 - mittels Glas-Elektroden vom Rest isoliert und elektrisch nicht angeschlossen

Schreiben

- Anlegen einer höheren Spannung (ca. 12-25 V) an Control-Gate und den Drain
- Source liegt auf 0 V
 -  Transistor wird leitend und hoher Strom fließt zwischen Source und Drain
 - Elektronen wandern mittels quantenmechanischem Tunneleffekt durch untere 50nm Glas-Elektrode auf Floating Gate
 - Bis Abschalten der Spannung bleiben sie dort

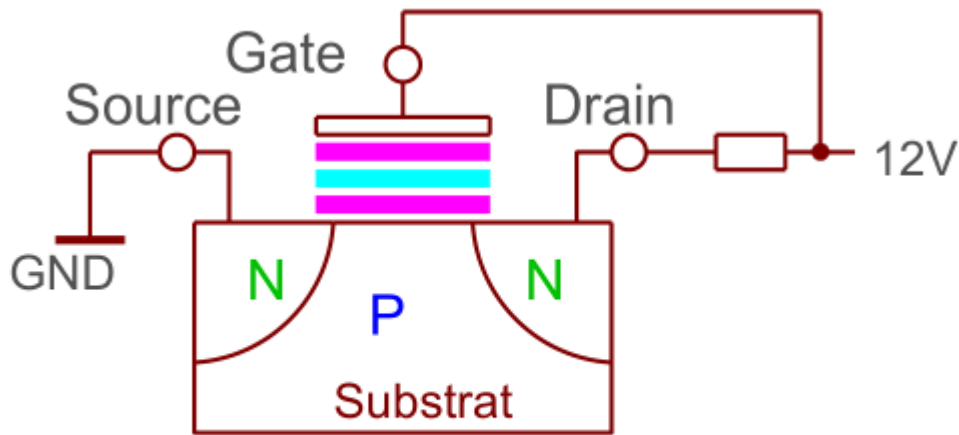


Abbildung 132 Floating Gate Schreiben

Lesen 0

- Lesevorgang ebenfalls über Control-Gate
- Jedoch kleinere Spannung als beim Schreiben
- Wenn Transistor ungeladen ist (0), verhält sich Floating-Gate anders als bei geladenem
 - Wie normaler MOSFET ➡ Durchsteuerung und an Ausgang und Leseverstärker liegen 0 V

Lesen 1

- Durch geladenes Floating-Gate (1) wird Durchsteuern verhindert
- Transistor ist gesperrt: ➡
 - Ausgang und Leseverstärker 3 V

5.3.2 EPROM: Bezeichnung 27xxx

Nichtflüchtiger löschbarer Speicher: (Erasable Programmable Read Only Memory)

- Löschen nach Beschreibung mittels UV-Licht
 - Dauer ~20min. Nur 100 mal möglich da Kristallstruktur durch UV- Strahlung zerstört wird
- Quarzglas-Fenster eingebaut, damit UV-Licht durchdringend kann: Wellenlänge 254nm.
 - Restliches Gehäuse aus Keramik gefertigt
- Löschen mittel Photoeffekt

EPROM Werte:

- Datenbusbreite 8-Bit
- Größen zwischen 64 kBit und 8Mbit

5.3.3 EEPROM

Nichtflüchtiger elektrisch löschbarer Speicher: (Electrically Erasable Programmable Read Only Memory)

- Erzeugung der Spannung für Schreibvorgang intern
- Verschiedene Verfahren zum Löschen der Daten
 - größere negative Spannung an Control-Gate anlegen

- oder positive Spannung am Drain Anschluss
- oder MOSFET

5.3.3.2 Parallel: Bezeichnung 28xxx

Ersatz für EPROMS da Pinkompatibel.

- Löschung einzeln oder durch Blöcke möglich
- Ausgestorben ➡ durch Flash-Bausteine ersetzt

5.3.3.3 Seriell

Heute oft in Embedded-Systemen verwendet

- Löschung fast jeder Speicherzellen einzeln möglich: mehrere 10.000 mal möglich
- Dauer von Speichern inkl. Löschung im Millisekunden Bereich
- 8-Polige Bauform

Nutzung z.B. auf DRAM-Moduk als Speicher für Referrnzdaten (Versorgungsspannung, Speicherkapazität, Timing, Refresh-Daten)

5.3.4 Flash

5.3.4.1 Allgemeines

Bit SPEicherung ebenfalls in Floating-Gate von Feldeffektransistor gespeichert.

Unterscheidung in NOR-Flash und NAND-Flash

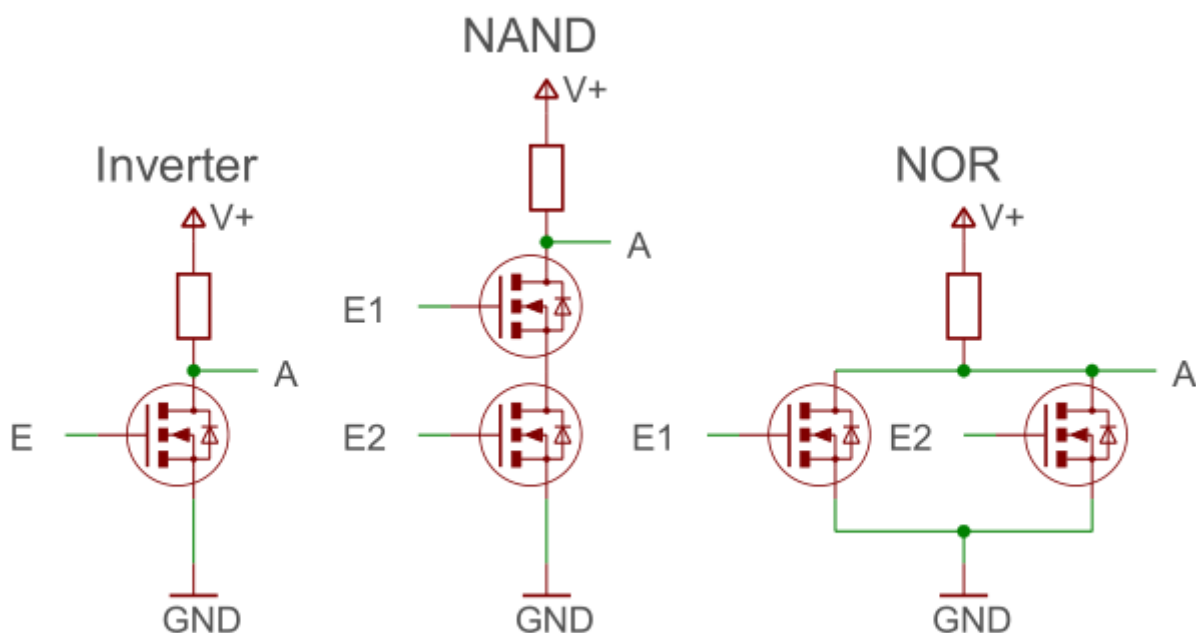


Abbildung 138 NOT NAND NOR

5.3.4.2 NOR-Flash

Ausführung der einzelnen Zellen als Matrix mit wahlfreiem Zugriff

- Zugriff überra Adress-Wortleitung

5.3.4.3 NAND-Flash

Hintereinanderschaltung von Gruppen von Zellen.

- Bei Zugriff auf einzelnen Zelle müssen alle Nachbarzellen in Kette durchgesteuert werden
- Wegen Signalübertragung durch Nachbartransistoren weniger zuverlässig als NOR-Flash
- Haltbarkeit gegenüber NOR-Flash 1/10

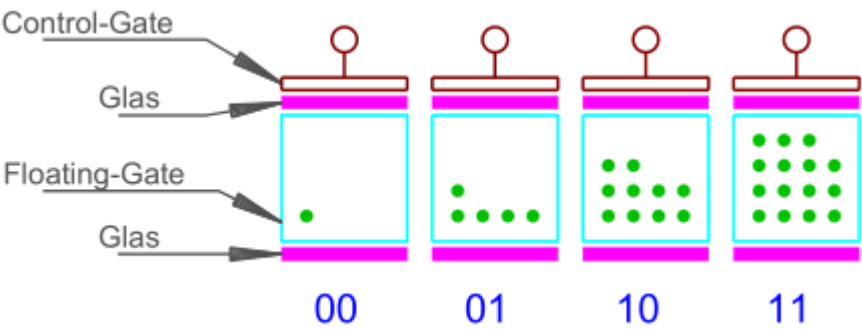
5.3.4.4 Vergleich NAND-NOR-Flash

Eigenschaft	NOR	NAND
wahlfreier Zugriff	ja	nein
Löschgeschwindigkeit	langsam	schnell
Fläche/Zelle	groß	klein
Zuverlässigkeit	hoch	niedrig
Verwendung	Programmspeicher in Mikrocontrollern	USB-Sticks, Speicherkarten, Festplatten

5.3.4.5 SLC MLC TLC QLC

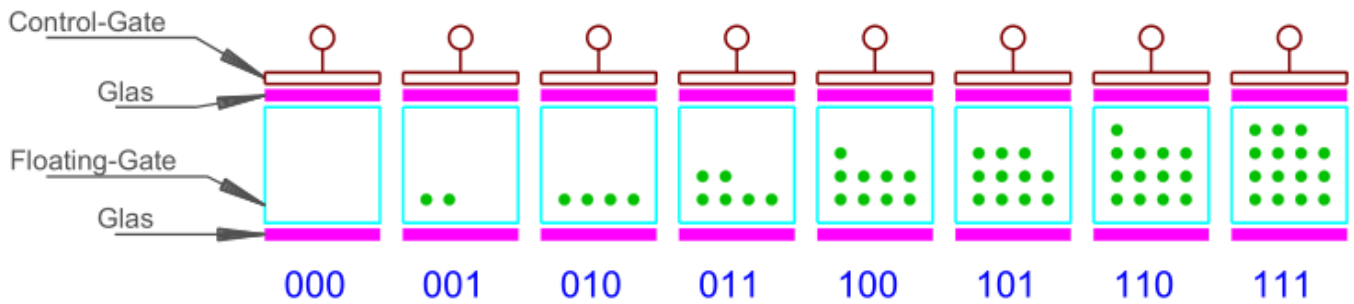
Für Speicherung von mehr Informationen pro Fläche, Entwicklung, mehr als eine 1 Bit information in Flash-Zelle zu speichern

- Menge der in das Floating-Gate fließenden Elektronen über zeit oder angelegte Spannung gesteuert
 - 3 Füllstufen: Speicherung von 2 Bit in Zelle



MLC-Zelle

Mit 7 Füllstufen Speicherung von 3-Bit



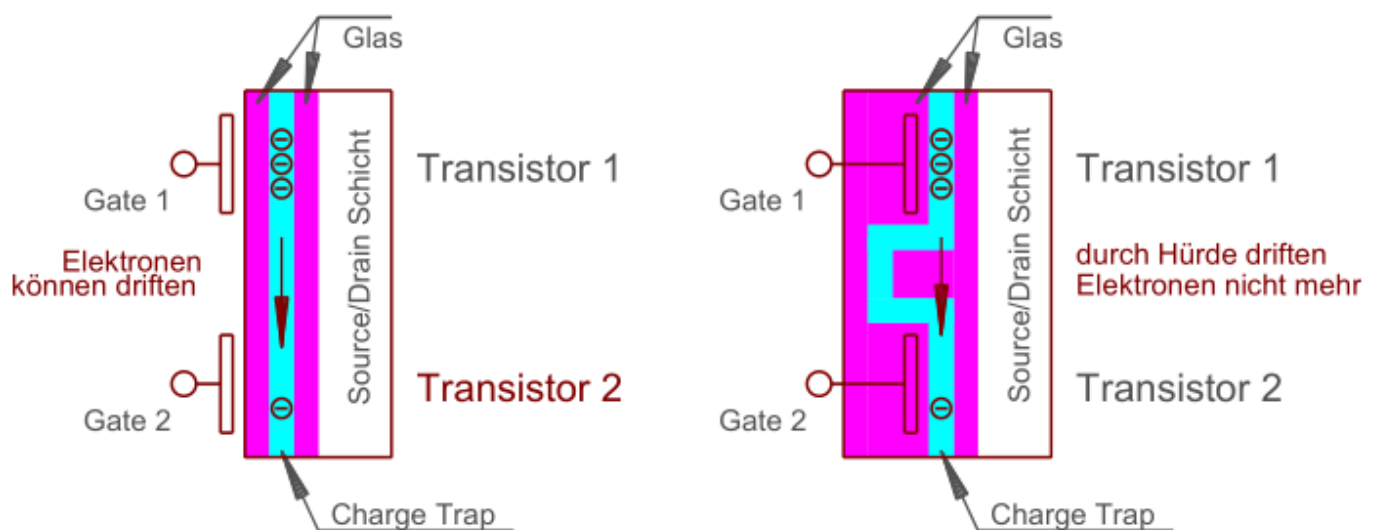
Beim Auslesen muss eine Trennung der Füllstufen ermittelt werden. Schon ab TLC Fehlerkorrektur aufwändig

- Je mehr Füllstufen, desto weniger Löschkzyklen

5.3.4.6 3D-NAND Flash

Speicherung an Grenzen gekommen, Elektronen können nicht ohne Fehlererzeugung weiter verringert werden

- Möglichkeit der Aufeinanderstaplung in der 3. Dimension
- Floating Gate aus Siliziumnitrid jetzt Charge Trap
- Damit Elektronen nicht driften, einbauen von Hürden



5.3.5 Modernere Speicherentwicklungen

5.3.5.1 Überblick

Moderne nicht-flüchtige Speicher sind deutlich schneller als FLASH oder EEPROMs.

- Information wird im Gegensatz zu FLASH oder EEPROM nicht in elektrischem Feld, sondern mittels ferroelektrischer Materialien in Magnetfeld gespeichert
- Robust 10^{10} Schreib-Lese-Zyklen

5.3.5.2 FRAM


FRAM-Speicherzellen (Ferroelectric Random Access Memory, auch FeRAM)

- physikalisch ähnlich wie Feldeffekttransistor mit Floating-Gate

- Speicher- und Löschvorgang ebenfalls durch Polarisationsänderung umgesetzt (ferroelektrische Schicht)

5.3.5.3 MRAM

MRAM-Speicherbausteine (Magnetoresistive Random Access Memory)

- Ähnlichkeiten zu GMR-Effekt
- Phsyikalisch ähnlich zu Feldeffekttransistor mit Floating-Gate
- unempfindlich gegenüber elektromagnetischer Strahlung
 -  Luft und Raumfahrt
 - 50x teurer als FLASH Speicher

5.3.5.4 Fazit

- Vermehrte Weiterentwicklung FRAM MRAM
 - Speicherdichte geringer als FLASH und DRAM
 - Schwierigkeit in der Integration ferroelektrischer Materialien in konventionelle Chip-Produktion

5.4 Fehlerkorrektur

5.4.1 Softerror

In Anfangszeit der Speicherbausteine ist es häufig vorgekommen, dass einzelne Bits ihren Wert von alleine geändert haben

- Ursache energiereiche Strahlung radioaktiver Isotope in Gehäusematerialien
- Gehäuse damals viel größer als heute, da viele Bausteine auf Leiterplatte benötigt wurden

Diese Softerrors wurden mit der Zeit immer weniger da:

- Vergussmaterialien immer ebsser wurden
- immer weniger Matieral verbaut wurde, das radioaktive Isotope enthält
- Thema heute wieder relevanter, aufgrund der vielen Gigabyte an Speicher, die verbaut werden

5.4.2 Parity

Beeinflussung der Softerrors, erste Maßnahme einfache Paritätsprüfung. Für Gruppe von 8-Bit wurde ein 9. Bit als Paritary-Bit eingeführt.

Bei ungerader Anzahl an Bits: Parity-Bit: 0 Bei gerader Anzahl an Bits: Parity-Bit: 1

<i>ASCII Buchstabe</i>	<i>Bit-7</i>	<i>Bit-6</i>	<i>Bit-5</i>	<i>Bit-4</i>	<i>Bit-3</i>	<i>Bit-2</i>	<i>Bit-1</i>	<i>Bit-0</i>	<i>Parity-Bit</i>
1	0	0	1	1	0	0	0	1	0
2	0	0	1	1	0	0	1	0	0
3	0	0	1	1	0	0	1	1	1
4	0	0	1	1	0	1	0	0	0
5	0	0	1	1	0	1	0	1	1
6	0	0	1	1	0	1	1	0	1
7	0	0	1	1	0	1	1	1	0
8	0	0	1	1	1	0	0	0	0

Nachteil: bei Umkippen von 2 Bits in einem Datenwort stimmt Paritäts-Prüfung nicht mehr

5.4.3 ECC

Weiterentwicklung von Paritätsprüfung EEC (Error Correction Code) Prüfung.

Dezimalwert	Bits								YP
170	1	0	1	0	1	0	1	0	1
85	0	1	0	1	0	1	0	1	1
20	0	0	0	1	0	1	0	0	1
250	1	1	1	1	1	0	1	0	1
127	0	1	1	1	1	1	1	1	0
128	1	0	0	0	0	0	0	0	0
33	0	0	1	0	0	0	0	1	1
175	1	0	1	0	1	1	1	1	1
XP 1 0 0 1 1 1 1 1 1									

Abbildung 149 2D-Parity

Dezimalwert	Bits								YP
170	1	0	1	0	1	0	1	0	1
85	0	1	0	1	0	1	0	1	1
20	0	0	1	1	0	1	0	0	1
250	1	1	1	1	1	0	1	0	1
127	0	1	1	1	1	1	1	1	0
128	1	0	0	0	0	0	0	0	0
33	0	0	1	0	0	0	0	1	1
175	1	0	1	0	1	1	1	1	1
XP 1 0 0 1 1 1 1 1 1									

Y-Parity Error

X-Parity Error

Abbildung 150 ECC Fehlererkennung

Prüfung von umgedrehten Bits in 2-Dimensionalem Array