

Grundlagen der IT, klausurrelevante Themen 1. Semester

Darstellung von Daten

Das Dualsystem

(Binär \rightarrow Decimal)

$$\cdot 101101 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 8 + 4 + 1 = 45$$

(Decimal - Binär)

$$\cdot 42 = 42 : 2 = 21 \text{ Rest } 0 \rightarrow 21 : 2 = 10 \text{ Rest } 1 \rightarrow 10 : 2 = 5 \text{ Rest } 0 \rightarrow 5 : 2 = 2 \text{ Rest } 1 \rightarrow 2 : 2 = 1 \text{ Rest } 0 \rightarrow 1 : 2 = 0 \text{ Rest } 1$$

$$= 101010 \quad \begin{matrix} \leftarrow \text{als 6.} \\ \downarrow 0 \end{matrix} \quad \begin{matrix} \leftarrow \text{als 5.} \\ \downarrow 1 \end{matrix} \quad \begin{matrix} \leftarrow \text{als 4.} \\ \downarrow 0 \end{matrix} \quad \begin{matrix} \leftarrow \text{als 3.} \\ \downarrow 1 \end{matrix} \quad \begin{matrix} \leftarrow \text{als 2.} \\ \downarrow 0 \end{matrix} \quad \begin{matrix} \leftarrow \text{als 1.} \\ \downarrow 1 \end{matrix}$$

Rechnen im Dualsystem

(Alternativ kann man immer in Dezimalzahlen umwandeln
(und dann das Ergebnis zurück in Binär wandeln))

- Addition:

$$\begin{array}{r} 1 & 0 & 1 & 0 \\ + & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array} \quad \left\{ \begin{array}{l} 0+0=0 \\ 1+0=1 \\ 1+1=0 \text{ aber } 1 \text{ weitergeleitet} \\ 1+1+1=1 \end{array} \right. \quad \begin{array}{l} \rightarrow \text{In dem Fall wird so oft ein } 1 \text{ weitergeleitet wie sie addiert} \\ \text{wurde } (1+1)+(1+1)=0 \rightarrow \text{wie leiten } 2 \times 1 \text{ weiter} \end{array}$$

- Subtraktion:

$$\begin{array}{r} 1 & 0 & 1 & 0 \\ - & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 \end{array} \quad \left\{ \begin{array}{l} 0-0=0 \\ 1-0=1 \\ 0-1=1 \text{ aber } 1 \text{ weitergeleitet} \\ 1-1=0 \end{array} \right.$$

! Positive Dualzahlen bis 7 immer
0 vorne: $3 = 0011$ und
nicht 11

- Multiplikation:

$$\begin{array}{r} 1 & 0 & 0 & 0 \cdot 1 & 0 & 1 & 0 \\ \times & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{array} \quad \left\{ \begin{array}{l} 0 \cdot 0=0 \\ 0 \cdot 1=0 \\ 1 \cdot 1=1 \end{array} \right.$$

Addition	Subtraktion	Multiplikation	Division
$0+0=0$	$0-0=0$	$0 \cdot 0=0$	Wenn Zahl
$1+0=1$	$0-1=1\ddot{\cup}1$	$0 \cdot 1=0$	passt dann 1
$1+1=0\ddot{\cup}1$	$1-0=0$	$1 \cdot 1=1$	sonst 0
	$1-1=0$		bis zum Ende

- Division: $\begin{array}{r} 100 : 10 = 10 \\ \hline 010 \quad \text{passt also } \swarrow \\ 00 \quad \text{passt nicht, kein dahter } 0 \\ 10 \end{array}$

Das Hexadezimalsystem

$$\text{Bsp. } 2A8FC = 2 \cdot 16^4 + 10 \cdot 16^3 + 8 \cdot 16^2 + 15 \cdot 16^1 + 12 \cdot 16^0 = 174330$$

Prinzip: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, \dots\}$ Also $A=10$

$$\begin{array}{l} B=11 \\ C=12 \\ \vdots \end{array}$$

→ Diese Werte verwenden
wie Binärzahlen auch,
nur immer mal 16 und
nicht mal 2

Das Einerkomplement

→ kann negative Zahlen darstellen

$$1 = \underline{0}0001 \xrightarrow{\text{umdrehen}} -1 = \underline{1}110$$

$$\text{Berechnung } -3+2 = \underline{\underline{1}}110$$

$$\begin{array}{r} + 0010 \\ \hline 1110 \end{array} \rightarrow -1 \checkmark$$

- Probleme des Einerkomplements sind, dass es nur für Werte von $-7 - 7$ gilt und dass die 0 zwei Binärwerte annehmen kann.

$$\begin{array}{r} -3+4 = \underline{\underline{1}}100 \\ + 0100 \\ \hline 0000 \end{array} \quad \begin{array}{r} \xrightarrow{\text{Übertrag muss noch beim hinzugefügten Ergebnis addiert werden}} \\ \underline{\underline{1}}0001 \\ + 0001 \\ \hline 0001 \end{array} \rightarrow 1 \checkmark$$

Das Zweierkomplement

$$6 = 0110$$

$$-6 = 1001 \text{ (Einerkomplement)}$$

(umdrehen und +1)

$$\begin{array}{r}
 1001 \\
 + 0001 \\
 \hline
 1010
 \end{array}$$

$1010 \rightarrow \text{Zweierkomplement} = \text{Einerkomp.} + 1$

- Hier hat die 0 nicht mehr 2 Binärwerte und man muss wenn man beim addieren die 0 überschreitet kein Übertrag mehr rechnen

Es
kommen keine
negativen
Zahlen dran
(denn lernen)

Rechnerarchitektur

- Bezieht sich auf die Struktur und das Design eines Computers. Das umfasst Hardwarekomponente wie z.B. Prozessor, sowie die Art und Weise, wie diese Komponente miteinander kommunizieren. Die Rechnerarchitektur beeinflusst die Leistungsfähigkeit des Computers.

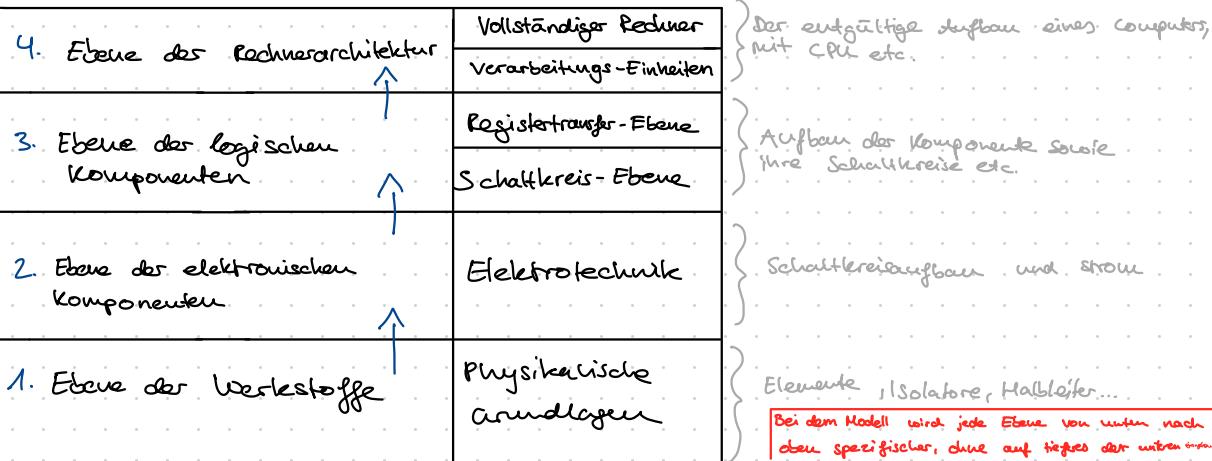
Rechnerorganisation

- Beschäftigt sich damit, wie ein Computer auf Ebene der Hardware tatsächlich funktioniert und wie seine Komponente miteinander interagieren. Sie bezieht sich weniger auf Design, sondern eher darauf, wie der Computer tatsächlich arbeitet und wie Daten und Befehle verarbeitet.

Abstraktionsebenen eines Rechners

nicht lernen

- Der Begriff Abstraktion beschreibt die Trennung zwischen Konzept und Umsetzung. Es bedeutet, dass man sich auf wesentliches konzentriert und unnötige Details ausblendet, um komplexe Systeme besser zu verstehen.
- Bei der Betrachtung eines Computers kann man verschiedene Abstraktionsebenen etablieren. Daniel T. Siewiorek, C. Gordon Bell und Allen Newell haben dazu ein schichtenmodell entworfen, was in Abstraktionsebenen schichten bis zur Hardware Software Schnittstelle zeigt:



Grundlagen elektronischer Schaltkreise / Flederintegration

Grundlegende Begriffe

Elektrischer Strom Fließt, wenn Elektronen (e^-) sich bewegen. Die Stromstärke beschreibt die Menge an elektrischer Ladung, durch einen Leiter fließt. Maßeinheit = Ampere [A]

Ohm'scher Widerstand = Der Widerstand, welcher auftritt, wenn Strom durch einen Gegenstand fließt. Wird in Ohm (Ω) gemessen. Je mehr es etwas hat, desto schlechter fließt der Strom dadurch. Er wird beeinflusst durch Material, Temperatur und Größe. Der spezifische Widerstand ist der Widerstand eines bestimmten Materials, wenn $l=1\text{cm}$, $q=1\text{mV}$ und $T=0^\circ\text{C}$ gegeben ist.

Spannung = Sie gibt an, wieviel Energie nötig ist, um Elektronen den bewegen. Eine Spannungsquelle hat immer einen pluspol mit Elektronenmangel und einen Minuspol mit Elektronenüberschuss. Je höher die Elektrondifferenz, desto größer die Spannung. Sie sorgt dafür, dass sich die Elektronen in Richtung + pol bewegen und somit Strom fließt. Gemessen wird sie in Volt [V]. ++ oder -- wirken abstoßend aneinander.

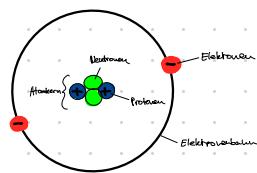
Das ohm'sche Gesetz = Beschreibt den Zusammenhang von Widerstand, Spannung und Strom. $R = \frac{U}{I}$



Elektrisches Feld = Wie ein Magnetfeld für elektrische Dinge. Ein unsichtbares Gebiet, das um eine elektrische Sache herum existiert und eine Kraft auf andere elektrische Sachen ausübt.

Ladungsträger = elektrisch geladene Teilchen (+/-) (bsp. Elektron e^-)

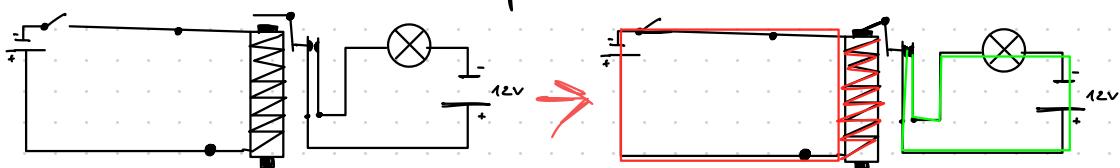
Das Atom =



- Nach außen ist ein Atom neutral, da die Anzahl von Protonen und Elektronen gleich sind
- Im ersten Raum können höchstens 2 Elektronen sein, im zweiten bis am 8, im 3 und 3...

Die elektrische Leitfähigkeit beruht auf beweglichen Ladungsträgern (Elektronen und Ionen). Wir betrachten die Elektronen! Isolatoren haben keine beweglichen Ladungsträger, daher isolieren sie. Kupfer hat viele, daher leitet es gut.

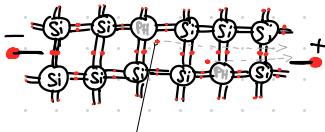
Beispiel:



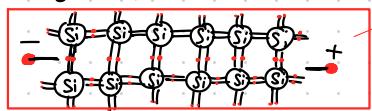
Halbleiter

Dotieren = Ist die "Verschmutzung" von Edelgassintern durch das einfügen von Atomen mit zu vielen Valenzelektronen (n -dotierung) oder mit zu wenigen (p -dotierung). Als Halbleiter verwenden wir Elemente der 4. Hauptgruppe, da sie ein Edelgassintern ergeben. Je meist wird hier Silizium verwendet.

(n -dotiert)
 n -dotierung
↓ Donator
(mit z.B. Phosphor)
hat 5 Valenzelektronen, also 1 zu viel

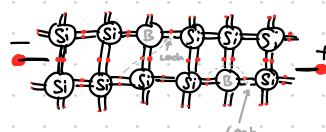


Überschüssige Elektronen fließen zum + pol und wir haben einen leitenden Halbleiter



Edelgassintern fehlen 1 Valenzelektronen.
 p -dotierung

(mit z.B. Bor)
↓ Akzeptor



hat 4 Valenzelektronen, also 1 zu wenig

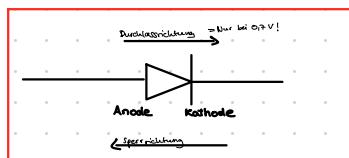
Nun stoppen elektronen die Löcher, wodurch neue entstehen. Dabei bewegen sich die elektronen zum + pol und wir haben einen leitenden Halbleiter

(Alles bei Zimmertemperatur, unter Hitzeangabe lösen sich Elektronen von alleine, da sie stark schwingen. Es entstehen Löcher und wenn der Halbleiter an eine Spannung angeschlossen wird, verhält er sich wie bei der p -dotierung)

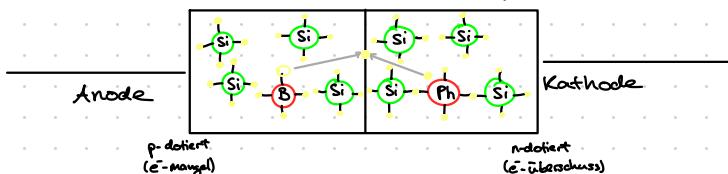
Der Vorgang oben heißt Störstellenleitung, da Störstellen erzeugt werden

Die Diode

Die Diode ist ein Halbleiterelement, welches im Stromkreis mit einem Pfeil gekennzeichnet wird. Die Besonderheit der Diode ist, dass sie Strom nur in eine Richtung durchlässt



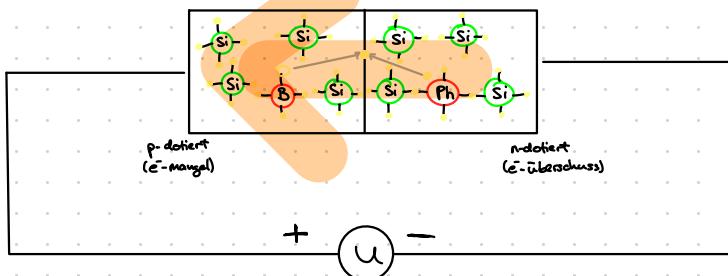
Silizium-Diode Aufbau



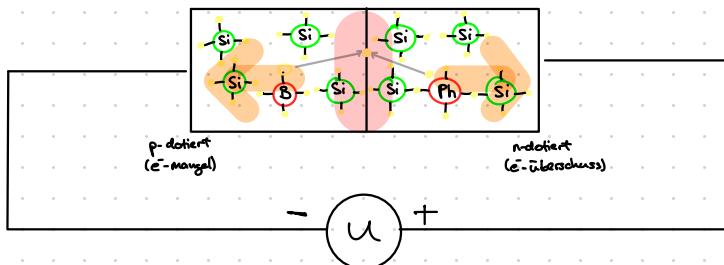
- Das Loch links wird von dem elektronenüberschuss angezogen und das freie Elektronen von elektronenmangel. In der Mitte treffen sie sich dann und sind glücklich miteinander

- { Durch die Ladungverschiebung entsteht ein elektrisches Feld über den verschobenen Ladungen

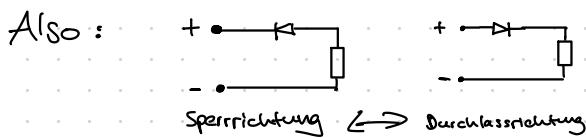
Dieses hat seinen $+$ -pol auf der n-Seite und seinen $-$ -pol auf der p-Seite. Dies ist, da auf der d-Seite nun mehr Elektronen sind, als es sein sollte, da ein Elektron kann \rightarrow also ein Elektronenüberschuss \rightarrow $-$ -pol. Auf der n-Seite ist ein Elektron gegenwärtig, also weniger als davor, daher Elektronenmangel \rightarrow $+$ -pol.



Legen wir den positiven pol nun an die p-Seite können die Elektronen durch, zu ihm und wir haben einen leiter



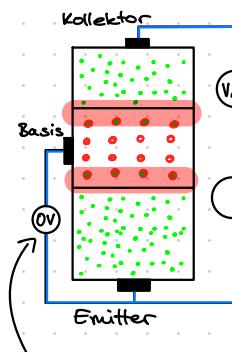
Gegen wir den positiven pol an die n-Seite, so fließen die Elektronen nicht durch die Diode und sie wirkt als Isolator



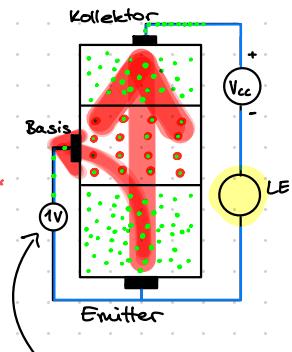
Info: Bei Si-Diode braucht man mind. 0,7 V, damit fließt

Der Transistor

Aufbau:

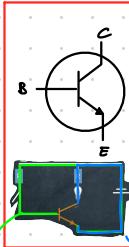


Hier liegt keine Spannung vor



Hier liegt eine weinig Spannung vor

Schaltbild



Basis-Emitter-Schaltkreis
Kollektor-Emitter-Schaltkreis

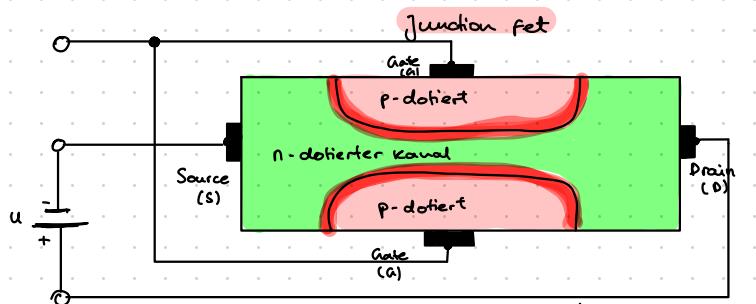
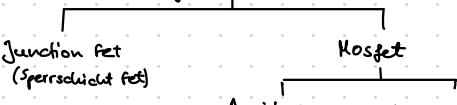
Erklärung: Solange am Basis-Emitter Kreis keine Spannung herrscht, befinden sich zwei Sperrschichten im Transistor (zwischen Basis, Emitter und Basis, Kollektor). Dies lässt sich mit dem Prinzip der Diode erklären. Wenn nun eine Spannung > 0.7 an der Basis herrscht geben die Elektronen bei der Basis zu einer kleinen Anzahl raus und zurück zum Emitter. Der Basis-Emitter Stromkreis leitet nun wenige Elektronen. Dies bewirkt aber dass die Elektronen im Emitter nicht alle bei der Basis rausgehen, sondern nun mit großer Anzahl in den Kollektor wandern. Dieses hat nun zu viele Elektronen und sie fließen aus ihm raus. Nun haben wir einen noch stärkeren Stromkreis im Kollektor-Emitter Stromkreis. Transistoren haben also die Eigenschaft, dass sie einen Stromkreis verstärken und keine Abmilderung verursachen. Der Transistor wird so aber nicht mehr verwendet, da die Elektronen von Emitter zu Kollektor zu lange brauchen. Außerdem wäre der Stromverbrauch zu hoch, da auch wenn für die Steuerung des Basis-Emitter Stromkreis nur eine kleine Kleine Strom gebraucht wird, es allein bei einem Chip mit 1Mio Transistoren zu viel wäre.

Daher ↴

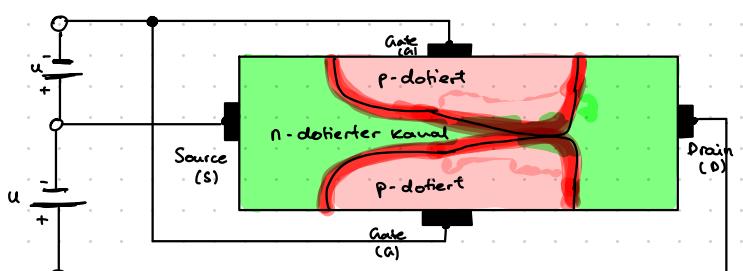
Der Feldeffekttransistor

Lehr Mosfet kommt
Feldeffekttransistoren

Es gibt verschiedene Typen von Feldeffekttransistoren:



- Durch den n-Kanal fließen problemlos Elektronen und der Source-Drain Kreis wird von Strom durchflossen

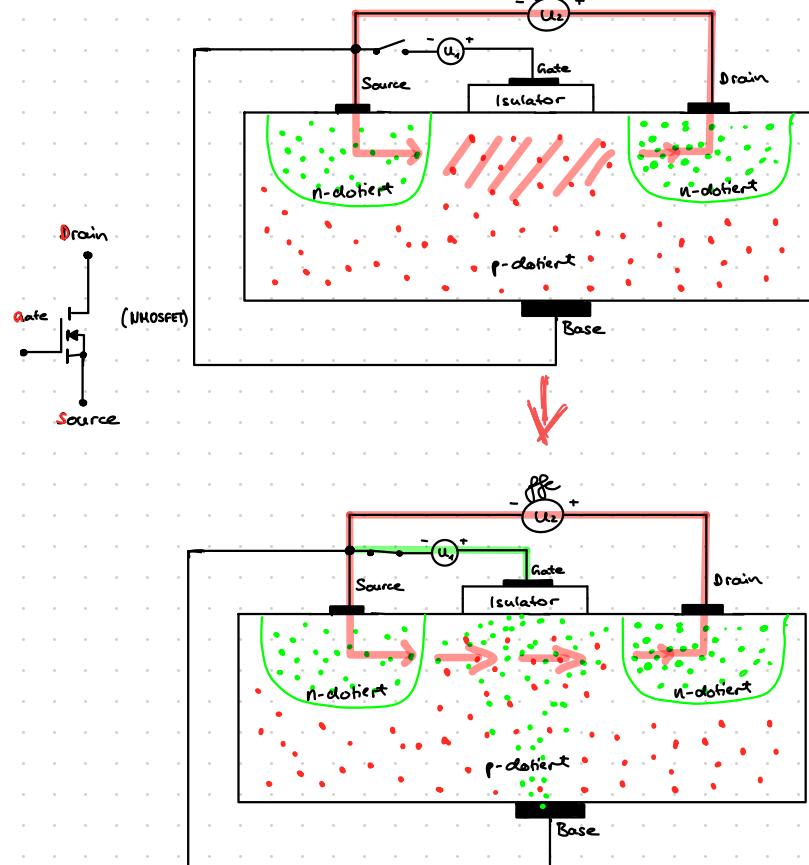


- Erhöht man die Spannung zwischen den Gates, breite sich die Sperrschicht aus und verengt oder gar schließt den n-Kanal, so wird also der Strom im Kanal leistungsfähiger gesteuert.

(Das ganze geht auch mit p-Kanal und n-Mosfet, hier ist das Prinzip gleich und nur +/- müssen vertauscht werden)

Mos - Feldeffekttransistor

(Youtube, Thomas Schwenkel)



- Wir wollen einen Strom zwischen Source und Drain, doch dazwischen sind sperrschichten und wir haben kein Krets

- Legen wir positive Spannung an die Gate an, so entsteht ein elektrisches Feld, das Elektronen in dem p-dotierten Feld anzieht, die körner jedoch abstößt. Die Elektronen bilden nun einen n-dotierten Kanal. Stromstärke zwischen Source und Drain kann durch erhöhen der Spannung U_D verstärkt werden, da mehr Elektronen in der p-Schicht sind, wenn mehr positive Spannung ausgeübt wird.

Dies ist ein Mosfet als Anreicherungstyp, der Ladungsträger (Elektronen) herangeschafft werden müssen.

für einen Verarmungstyp wären p und n, sowie die Ladungen jeweils andersrum. Somit würden Löcher in die Leitung rücken und leiten } pmosfet
→ Verarmungstyp

C-MOSFET

NMOSFET und PMOSFET verschwenden beim Umschalten ein wenig Strom. Addiert ist das jedoch ganz schön viel.

Der CMOSFET ist eine Kombination (Aneinanderreihung) von NMOSFET und PMOSFET. Er verbraucht kein Strom beim Umschalten und wird in fast allen Hochleistungs-Mikroprozessoren verwendet.

Hier muss man nicht wissen → würde Fahrzeuge sprengen

Kurz zu FinFET: 3D-MOSFETs, bieten kürzere Schaltzeit etc., Weiterentwicklungen: GAAFET, HBCFET

Technische Komponenten und Realisierung von Logikbausteinen

Logikbausteine

- Digitale Schaltung hat die Werte 0 und 1
- TTL-Technik verwendet Spannungsbereiche von 0-0,4V für 0 und 2,4-5,0V für 1
- Die wichtigsten Technologien sind bipolare und MOS-Bausteine und BiCMOS-Gatter
- MOS-Transistoren können als Schalter fungieren
- BiCMOS vereinigt bipolare und bipolare Technologien
- CMOS haben die meisten modernen CPUs und Speicher

Die Bausteine heißen Gatter

Für uns relevant sind folgende Gatter:

Inverter: Wenn V_{in} auf low liegt, liegt V_{out} auf high und umgekehrt
 $\begin{pmatrix} V_{in} & V_{out} \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$ Logische 0 wird 1 und 1 wird 0.
 Der Widerstand ist nötig um den vom Transistor gezogenen Strom zu begrenzen

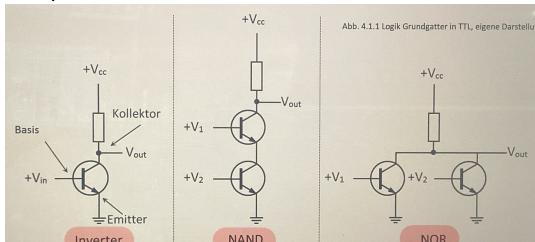


Abb. 4.1.1 Logik Grundgatter in TTL, eigene Darstellung

NAND: Zwei in Reihe geschaltete Transistoren. Wenn V_1 und V_2 auf High liegen, dann ist V_{out} low. Ist einer oder beiden Transistoren auf low, ist V_{out} high
 $\begin{pmatrix} V_1 & V_2 & V_{out} \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

NOR: Zwei parallel geschaltete Transistoren. Wenn nur V_1 oder nur V_2 auf high, ist V_{out} low. Sind V_1 und V_2 low, bleibt der Ausgang (V_{out}) auf high
 $\begin{pmatrix} V_1 & V_2 & V_{out} \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$

↳ Bei AND und OR braucht man ein Transistor mehr

Wichtig: Fertigungsprobleme: Größe ...

Von Neumann Architektur

Eigenschaften eines von Neumann Computers

1. Hardware hat die 4 Komponenten: -Haupt- bzw. Arbeitsspeicher für Programme und Daten
 - Leitwerk, dass das Programm interpretiert
 - Rechenwerk, dass arithmetische Operationen ausführt
 - Ein/Ausgabewerk, welches mit der Umwelt kommuniziert
2. Struktur des Rechners ist unabhängig vom bearbeiteten Programm: Programmsteuerung
3. Programm und Daten verwenden denselben Speicher (Hauptspeicher) und können durch die CPU verändert werden
4. Hauptspeicher ist in Zellen gleicher Größe geteilt, die durch fortlaufende Nummern bezeichnet werden
5. Programm besteht aus einer Folge von Befehlen, die im Allgemeinen nacheinander ausgeführt werden
6. Abfolge von Maschinenbefehle kann nur durch spezielle Sprungbefehle verändert werden
7. Die Maschine benutzt Binärcode, Zahlen werden dual dargestellt

Nach von Neumann darf der Prozessor erst einen neuen Befehl auffangen, wenn der alte abgearbeitet ist.

Architektur der von Neumann Architektur

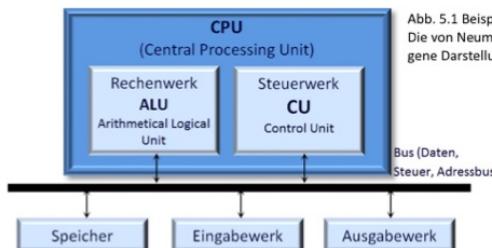
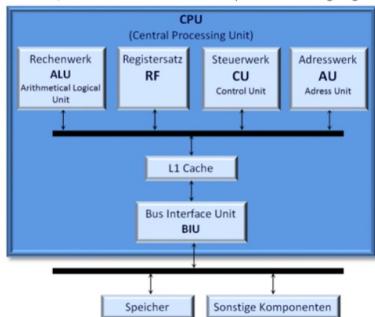


Abb. 5.1 Beispiel
Die von Neumannsche Darstellung

Probleme: Es dauert viel zu lang

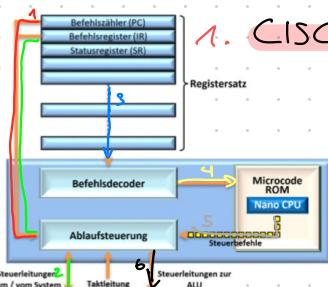
Erweiterte von Neumann Architektur



Erstmal: Programmablauf im Prozessor

1. Holphase (Fetch): Zunächst muss ein Befehl aus dem Speicher (eventuell in eines der Register) geladen werden. Wenn sich die gesuchte Information noch nicht im L1-Cache befindet, muss die BIU dafür den Speicher ansprechen.
2. Dekodierphase (Decode): Das Steuerwerk untersucht den Befehl hinsichtlich der vorzunehmenden Arbeitsschritte. Es entscheidet, welche Operation (Addition, Subtraktion, logische Verknüpfung und so weiter) die ALU ausführen hat.
3. Ausführungsphase (Execute): Die ALU wird vom Steuerwerk nach angewiesen, die gewünschte Operation mit den geladenen Daten auszuführen.
4. Schreibphase (Write Back): Das Ergebnis der Operation landet (abhängig vom Befehl) in einem der Register oder im Speicher. Um langwierige Zugriffe auf das externe RAM zu vermeiden, sammelt die CPU die Daten eventuell im L1-Cache, damit sie später in einem schnellen Blockschreibvorgang übertragen werden können.

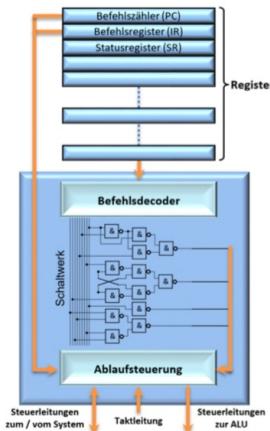
- Das Steuerwerk:** Koordiniert den Befehls- und Datenfluss der CPU und bestimmt die Geschwindigkeit des Programmablaufs
- Zum Steuerwerk werden **Befehlszähler**, **Befehlsregister** und **Statusregister** zugeordnet
 - **Befehlszähler:** Enthält die Adresse des zunächst auszuführenden Befehls im Arbeitsspeicher. Nach der Ausführung eines Befehls wird er um die Länge des Befehls erhöht, um die Adresse des nächsten Befehls zu ermitteln. Bei Sprungoperationen wird der Zähler auf die Zieladresse des Sprungs eingestellt.
 - **Befehlsregister:** Hier wird der Befehl, der aus dem Arbeitsspeicher geholt wird, abgespeichert. Er enthält immer den Befehl, der gerade ausgeführt wird.
 - Ein dem Steuerwerk zugeordneter "Decoder" entschlüsselt die im Operatorkommando angegebene Bitkombination und setzt diese in steuersignale um. Die Signale und die errechnete Operandenadresse werden dann je nach Befehltyp an die zuständigen Teile weitergeleitet
- Nun unterscheidet das Steuerwerk (die CU) je nach Art der Befehlscodierung in sogenannte **RISC** und **CISC** Strukturen



1. CISC (Complex Instruction Set Computer)

- CISC ist eine CU Struktur, die eine Vielzahl von komplexen und mächtigen Befehlen bereitstellt, die in der Lage sind, komplexe Operationen in einem Schritt auszuführen. Sie sind im Vergleich zu RISC langwamer, aber leichter zu programmieren.

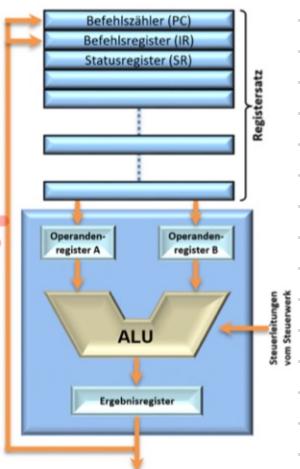
2. RISC (Reduced Instruction set computer)



- CU - Struktur mit einem einfachen und gleichbleibenden Befehlssatz, der hauptsächlich aus arithmetischen und logischen Operation besteht.
- Eigenschaften:
 - Wenige und einfache Befehle
 - Fester Befehlssatz (keine oder wenige komplexe Befehle)
 - Befehle haben gleiche Länge
 - Einfache Adressierungsarten
 - Optimierung für Pipelining und Parallelverarbeitung
 - Registerbasierte Architektur
- Ist schneller als CISC

Heute sind CISC CPUs nicht mehr nur mikroprogrammiert. CISC und RISC sind heute vereint von beiden Vorteile zu haben.

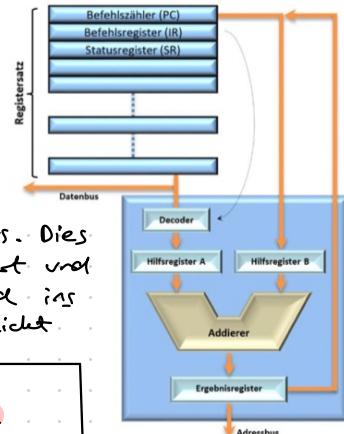
Das Rechenwerk (ALU)



Die ALU (Arithmetic Logical Unit) wird von der CU gesteuert. Sie holt auf Anweisung die beiden Operanden aus dem Registersatz. Im nächsten Schritt führt sie mit den beiden eine Rechenoperation aus (+, -, ...). Das Resultat wird im Ergebnisregister zwischengespeichert sodass die ALU mit den nächsten Operanden rechnen kann.

Das Adresswerk:

In Hilfsregister A befindet sich die vom Decoder extrahierte Adressinformation und im Hilfsregister B der Inhalt des Befehlszählers. Dies wird im Addierer addiert und die neue Adresse wird ins Befehlsregister zurückgeschickt.



Der Registersatz: Er hat die Aufgabe Angaben, welche direkt wieder gebraucht werden kann zwischenzuspeichern. Er fasst nur wenige Bytes.

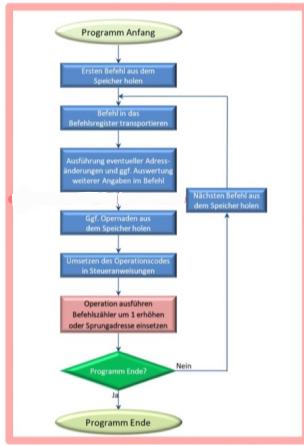
↓
Registers erlauben einen schnelleren Zugriff auf den Arbeitsspeicher

Die Busschnittstelle (BIU):

Die „Bus Interface Unit“ verbindet die internen Busse eines Prozessors mit der Außenwelt.

Außerdem sorgt sie für eine Pegelaufpassung zwischen dem CPU-Kern und dem externen Bussystem. (Dort nutzt es heiss wird)

Von Neumann Ablauf



→ Nun haben wir aber immer noch das Problem des von Neumann Flaschenhalses (Abläufe nacheinander)

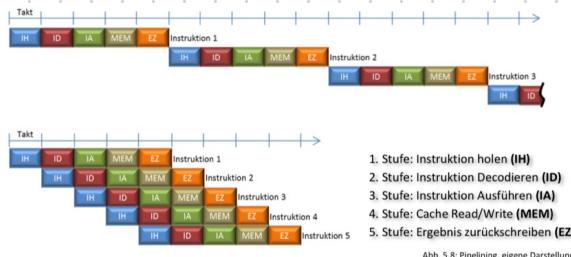


Das lösen wir durch Parallelität. Also Prozessen die nun gleichzeitig ausführen kann.

Man kann Parallelität in zwei Hauptformen auffinden

1. Auf Instruktionsebene
2. Auf Prozessorebene

Pipelining

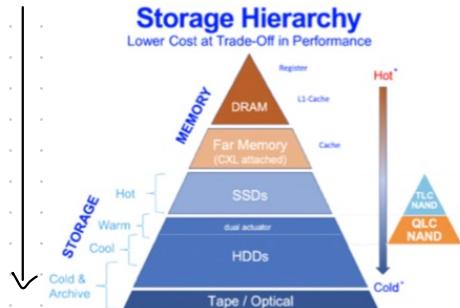


Probleme von Pipelining:

- Ressourcen-Konflikte; wenn Zugriffe belegt sind
- Daten-Konflikte - Daten stehen nicht zur Verfügung
 - Registerinhalte stehen nicht zur Verfügung
- Kontrollfluss-Konflikte; Pipeline wartet Sprung ab

Halbleiterspeicher

Speicher-Kapazität



↑
Zugriffs-geschwindigkeit

Preis

Primär Speicher: Teil des Computers, in dem Programme und Daten gehalten werden

Bits: Grundeinheit des Speichers. Ein Bit enthält entweder 0 oder 1.

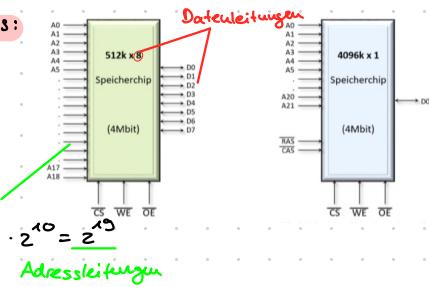
Speicheradressen

Fragen: Wenn der Computer eine Information speichert, gibt er ihr eine eindeutige Speicheradresse, um die Information später identifizieren zu können. Sie besteht aus einer beliebigen Anzahl von Bits.

Ein Computer mit 8-Bit-Speicheradressen kann 256 verschiedene Adressen haben, da $2^8 = 256$.

Byteanordnung: Probleme wenn man Daten von Computern die verschiedene Anordnungen benutzen austauscht

Speicherchips:



$$512 \cdot 1024 = 2^9 \cdot 2^{10} = 2^{19}$$

Adressleitungen

Berechnung: $\frac{512 \cdot 1024 \cdot 8}{1024 \cdot 1024 \cdot 8} = \frac{512}{1024} = 0,5 \text{ MB}$ Speicherkapazität
(Abb. 1) (kB) (kB) (Bit)

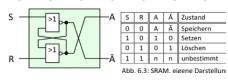
RAM

Statistisch (SRAM)

Diese Speicher haben die Fähigkeit ihre Inhalte so lange zu behalten, wie Versorgungsspannung anliegt. Er besteht aus 6 Transistoren pro Bit

Vorteile: extrem schnell,
Nachteile: teuer, geringe Packungsdichte

↓
eher cache geeignet



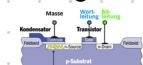
dynamisch (DRAM)

Diese Speicher bestehen aus einer Reihe von Zellen, wobei jede Zelle ein Kondensator und ein Transistor beinhaltet. Er braucht keine Versorgungsspannung.

Vorteile: Braucht weniger Transistoren und hat somit mehr Bits pro Chip

Nachteile: langsam

↓
Hauptspeicher geeignet

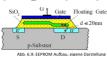
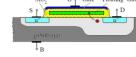


ROM: Read only Memory, also vom Hersteller programmierte Speicher, welchen man nicht verändern oder löschen kann.
- In Roll Form ist diese Speicherart allerdings sehr teuer und lohnt nur bei Massenprodukten

PROM: Programmable Read Only Memory, ist ein Speicherbaustein, der einmalig programmiert werden kann.
- PROM kann man "brennen"

Fragen

E PROM: Eraseable PROM, wie PROM, nur dass aufgrund des versteckten FETs UV-Strahlen den Speicherinhalt löschen können (FAMOS-Transistor)



EEPROM: Electrical EEPROM ist ein durch das Anlegen von elektrischen Spannungen löscharbarer Speicherbaustein. → Aber keine Lösung mit UV-Strahlen

FlashROM: Wie EEPROM nur das die Daten nicht Bit oder Byte-Schreibe sondern Blockweise gelesen und geschrieben werden → in Handys, den kleinen Schreib/Lesegeschwindigkeit

Primärer Cache:
(first level cache)

getrennt als 2 Caches zum Speichern von Befehlen und Daten. Schnell aber teuer
Kapazität: paar dutzend KB

Cache-Speicher

= schneller Puffer-Speicher; schneller Zugriff; zwischen Komponenten im PC angeordnet

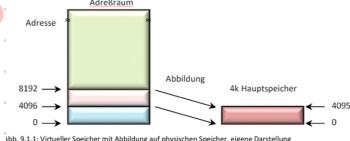
Sekundärer Cache:
(second level cache)

Daten und Befehle werden gemeinsam gespeichert. Ist langsamer und wird meist außerhalb des Prozessors mit SRAM-Bausteinen realisiert

Sonstiger Cache: Auf Festplatten, Grafikkarte etc...

Speichertechnik

Paging:



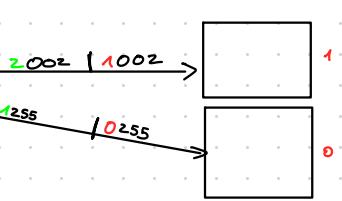
Beispiel:

Virtueller Speicher

255	Insta
255	Spotify
255	Youtube
255	WhatsApp

MMU

Physischer Speicher



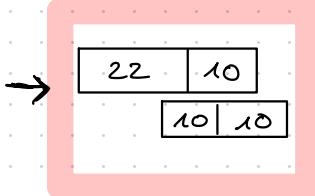
Rednung:

$$\begin{aligned}32 \text{ Bit} &= 2^{32} \\1 \text{ MB} &= 2^{20} \\1 \text{ kB} \text{ Pagesize} &= 2^{10}\end{aligned}$$

$$\text{Virtueller Speicher: } 2^{32} : 2^{10} = 2^{22}$$

$$\text{Hauptspeicher: } 2^{20} : 2^{10} = 2^{10}$$

Aufteilung



RAIDs

Zusammenschaltung unabhängiger Festplatten

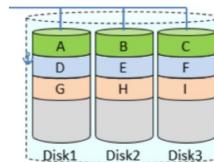
Löst das Problem, dass SLEDs zum Beispiel zu teuer sind

Raidarten:

RAID 0: Data Striping

Zusammen schaltung von Festplatten durch Aufteilung in kleine Blöcke

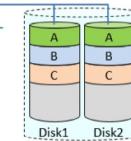
- schneller Zugriff
- keine Redundanzinformationen: Bei Ausfall alles weg



RAID 1: Drive-Mirroring

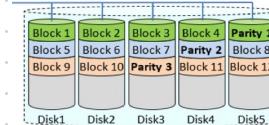
Speicher mit identischen Informationen (100% Redundanz)

- teuer
- Kein Dataverlust wenn 1 Speicher ausfällt



RAID 5: Block Striping mit verteilter Parity

Parity über alle Festplatten verteilt
Parity ist wie Prüfsumme mit den sich Blöcke
Wiederherstellen lassen



Kommunikationssysteme

Grundlagen: Kommunikationssysteme sind Technologie, die Informationen zwischen Sendern und Empfängern übertragen, um eine verständliche und zuverlässige Kommunikation zu gewährleisten.

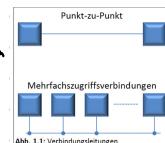
Kommunikativität: • Netzwerke verbinden Computer

- Begrenzte Netzwerke können ausreichend sein
- Internet verbindet alle Rechner der Welt
- Private Netzwerke beschränken die Anzahl der angeschlossenen Rechner
- Skalierbarkeit ist ein wichtiges Kriterium im Netzwerksdesign

Links, Knoten und Wölken: • Die einfachste Ebene auf dem Netzwerk

Direkte Kommunikativität:

Sich zusammensetzen können ist die Verbindung zweier Knoten (Rechner) über ein Physisches Medium (Kabel)
↳ Diese Variante reicht jedoch nicht für große Rechner, da Kabel etc. irgendwann zu viel werden würden

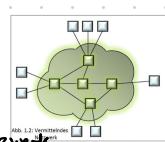


Indirekte Kommunikativität:

Leitungsvermittlung

Paketvermittlung

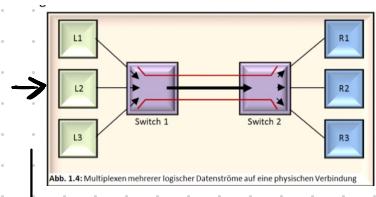
• Auf den Knoten, die an minst. 2 Verbindungsleitenden angeschlossen sind, läuft Software, welche die über eine Verbindungsleitung empfangenen Daten auf einer anderen weiterleitet. In einer systematischen Anordnung bilden diese weiterleitenden Knoten ein vermitteltes Netzwerk



Kostengünstige gemeinsame Ressourcennutzung

Wichtigste Anforderungen an Rechnernetze - Effizienz:

Multiplexen: Bedeutet, dass mehrere Benutzer eine Systemresource gleichzeitig nutzen können



Methoden: 1. STDM: Im Synchronous Time-Division Multiplexing bekommt jeder Nutzer im Netzwerk nacheinander in einem Zyklus Zeit bereitgestellt, in welcher er Daten übertragen kann
Problem: Auch wenn 1 Nutzer gerade mit überträgt bekommt er Zeit, das ist unnötig

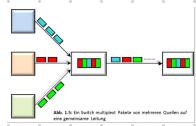
Datenströme werden gemultipliert in Switch 1 und dann in Switch 2 entmultipliziert und korrekt verteilt

2. FDM: Im Frequency Division Multiplexing wird jeder Datenstrom über die gemeinsame Verbindungsleitung in einer anderen Frequenz übertragen
Problem: Auch Nutzer die keine Daten übertragen bekommen eine Frequenz und die Anzahl dieser ist begrenzt

Lösung: Statistisches Multiplexing: Hier muss jeder Nutzer anfragen bevor er Daten teilen kann. Somit werden keine Frequenzen unnötig vergeben und keine Zeit verschwendet

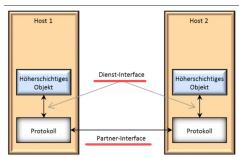
Paketübertragung beim Statistischen Multiplexing:

Wenn Daten zu groß sind, werden sie in Pakete aufgeteilt. Pakete werden abwechselnd verschickt



In meinen Lernzetteln beziehe ich mich nur auf das OSI-Schichtenmodell und Protokolle, da laut @Amira nur das dran kommt.

Was ist ein Protokoll?: Abstrakte Objekte, aus den sich die Schichten eines Netzwerksystems zusammensetzen. Es bietet einen Kommunikationsdienst, den Höherschichtige Objekte verwenden.



Der Begriff Protokoll kann auf 2 Arten verwendet werden:

1. Protokollspezifikation bezieht sich auf die abstrakten Interfaces
2. Protokoll bezieht sich auf das Modell, das diese beiden Interfaces implementiert

OSI-Schichtenmodell

Schicht	Bezeichnung	Beschreibung
7	Anwendungsschicht	besteht aus Anwendungen, mit denen man das Netz nutzen kann
6	Darstellungsschicht	Standardisiert das Format der Daten auf dem Netz
5	Kommunikationsschicht	Verwaltet die Verbindungen zwischen den Anwendungen
4	Transportschicht	garantiert die fehlerfreie Datenübertragung durch Fehlererkennung und Korrektur
3	Vermittlungsschicht	vermittelt die Verbindungen zwischen den Rechnern im Netz für höhere Schichten
2	Sicherungsschicht	sorgt für die zuverlässige Übertragung der Daten über die physikalische Verbindung
1	Bitübertragungsschicht	definiert die physikalischen Eigenschaften der Übertragung

Definition: Das OSI - Schichtenmodell ist ein Referenzmodell für Netzwerkeprotokolle als Schichtenarchitektur.

Kapselungen anhand des Modells: Die Schicht fügt ein Kopfteil und ein Endteil ein um mit dem Partner zu kommunizieren!

OSI-Layer 7 (Application Layer = Anwendungsschicht)

Interpretiert die Inhalte der Datenpakete.

OSI-Layer 6 (Presentation Layer = Darstellungsschicht)

Beschreibt die Codierung der Nachricht.

OSI-Layer 5 (Session Layer = Sitzungsschicht)

Hält virtuelle Verbindungen zwischen zwei Computern im Internet.

OSI-Layer 4 (Transport Layer = Transportschicht)

Regelt die komplette Vermittlung der Datenpakete vom Ursprung bis zum Ziel.

„Vorfahrtsregelung“: Internet-Telefonie vor WWW-Seite

OSI-Layer 3 (Network Layer = Netzwerkschicht oder Paketschicht)

Sorgt für die Vermittlung der Nachrichten im Internet.

OSI-Layer 2 (Data-link Layer = Datenverbindungsschicht)

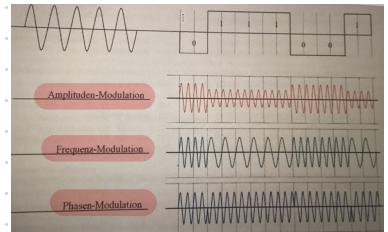
Legt fest, wie die Anforderung einer vermissten Nachricht funktioniert und wie die Datenpakete prinzipiell aufgebaut sind.

OSI-Layer 1 (Physical Layer = Physikalische Schicht)

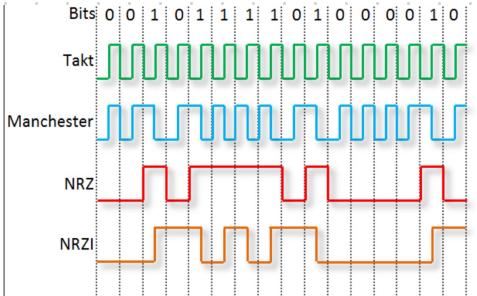
Beschreibt die Signale des Übertragungsmediums.

Modulationsverfahren:

gegner Code in
binär: 011001 →



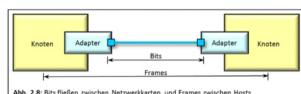
Verschiedene Kodiermethoden



Die Probleme von NRZ lassen sich lösen durch:

- Manchester-Kodierung: Mischt Takt und Signal durch XOR. Daher führen 0 und 1 zum Übergang. Dies ist dann aber nur 50% effektiv.
- NRZI-Kodierung: Aufeinanderfolgende 1 sorgen für Übergang aber 0 nicht!

Erzeugung von Frames: Sie entstehen aus Datenblöcken, werden in eine Bitfolge umgewandelt und von der Empfängerkarte identifiziert und gespeichert



Fehlererkennung

Es gibt 2 Methoden zur Korrektur von erkannten Fehlern. Bei der ersten wird der Sender benachrichtigt und aufgefordert die Nachricht erneut zu senden. Die zweite ist die Fehlererkennung:

Zur Fehlererkennung werden in Netzwerken verschiedene Techniken eingesetzt. Eine davon ist der CRC (Cyclic Redundancy Check).

CRC Berechnung Beispiel:

Hier ein Beispiel:

Die Nachricht $M(x) = x^7 + x^6 + x^3 + x^1$ bzw. **10011010**

und Divisor-Polynom $C(x) = x^3 + x^1$ bzw. **1101**.

Wir beginnen mit der Multiplikation von $M(x)$ mit x^3 , da unser Divisor-Polynom Grad 3 hat. Das ergibt

$T(x) = \text{1001101000}$. Wir dividieren dies durch den Generator $C(x) = 1101$.

Abb. 2.17 zeigt die lange

polynomiale Division. Mit den oben beschriebenen

Polynom-Arithmetik-Regeln verläuft die lange Division

mehr oder weniger genau als würden wir Ganzzahlen dividieren. Im ersten Schritt unseres Beispiels sehen

wir also, dass der Divisor, auch Generator genannt, 1101 einmal in die ersten vier Bit der Nachricht (1001) passt, weil sie den gleichen Grad haben; es ergibt sich ein Rest von 100 ($1101 \text{ XOR } 1001$).

Im nächsten Schritt bringen wir ein Bit vom Nachrichten-Polynom herunter, bis wir ein weiteres Polynom mit dem

gleichen Grad wie $C(x)$ erhalten - in diesem Fall 1001. Wir berechnen wieder den Rest (100) und fahren fort,

bis die Berechnung vollständig ist. Man beachte, dass das »Ergebnisse« der langen Division, das in der

Berechnung nicht erscheint, nicht von Interesse ist; von Belang ist nur der Rest am Ende der Division.

In unserer Beispielberechnung beträgt der Rest 101. Folglich wissen wir, dass **1001101000** minus **101**

genau durch $C(x)$ teilbar ist, und genau das senden wir. Die Minusoperation in der Polynom-Arithmetik ist

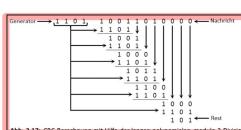


Abb. 2.17: CRC-Berechnung mit Hilfe der langen polynomischen modulo-2-Division

Subnetting

Definition: Beim Subnetting handelt es sich um die Aufteilung eines IP-Netzwerkes in kleinere Netzwerke, um effizienter mit IP-Adressen umzugehen.

Statistisches Subnetting: Hier wird die Anzahl der Subnetze, die Größe der Subnetze und die Anzahl der Hosts in jedem Subnetz im Voraus geplant.
Es ist eine manuelle Methode zur Erstellung von Subnetzen.

Nachteil: Weniger flexibel

Dynamisches Subnetting: Ermöglicht die automatische Zuweisung von IP-Adressen und die dynamische Erstellung von Subnetzen, um auf Änderungen im Netzwerksbedarf zu reagieren.

Nachteil: Komplexer, schwerer

Netz	Mask	Bitmask	Präfix
Klasse A Netz	255.0.0.0	11111111.00000000.00000000.00000000	/8
Klasse B Netz	255.255.0.0	11111111.11111111.00000000.00000000	/16
Klasse C Netz	255.255.255.0	11111111.11111111.11111111.00000000	/24

Wie geht das ???

SIE MÖCHTEN DAS NETZ 192.100.2.0 IN 4 SUBNETZE UNTEREILEN. WIE LAUTEN DIE SUBNETZMASKE UND DIE ECKDATEN DER SUBNETZE? (6P)

(Berechnung der Subnetzmasks, Subnetze müssen nachvollziehbar sein!)

192.100.2.0 ist ein Class C Netz -> Standard Subnetmask = 255.255.255.0

Für 4 Subnetze werden 2 Bit benötigt. Somit das 4. Tupel der Subnetmask binär/dezimal: 1100 0000 = 192

255.255.255.192 als Subnetmask für 4 Netze

Netz	Netz	Netz	Host	
11111111.11111111.11111111.11000000				
11.000000.01100100.00000010.00000001	.1			Subnetznummer 1 / .2. erste .62 letzte Hostadresse
11.000000.01100100.00000010.00111111	.63			Subnet Broadcast
11.000000.01100100.00000010.01000001	.64			Subnetznummer 3 / .65 erste .126 letzte Hostadresse
11.000000.01100100.00000010.01111111	.127			Subnet Broadcast
11.000000.01100100.00000010.10000001	.128			Subnetznummer 3 / .129 erste .190 letzte Hostadresse
11.000000.01100100.00000010.10111111	.191			Subnet Broadcast
11.000000.01100100.00000010.11000001	.192			Subnetznummer 4 / .129 erste .253 letzte Hostadresse
11.000000.01100100.00000010.11111111	.254			Subnet Broadcast

Da per Definition Host=0 die Netznummer und Host=255 die Broadcastadresse des Gesamtnetzes ist, beginnt das erste Subnetz bei 1 und das letzte endet bei 254

Latenz: Ausbreitungsverzögerung + Übertragungsverzögerung + Wartezeit

Ausbreitungsverzögerung: Entfernung / Lichtgeschwindigkeit

Übertragungsverzögerung: Paketgröße / Bandbreite

Wartezeit: Wartezeit durch Zwischenspeicherungen in den Routers/Switches

Betriebssysteme

Schichtenmodell



- Jede Schicht bildet eine abstrakte virtuelle Maschine, die mit ihrer benachbarten Schicht über Schnittstellen kommuniziert
- Die Funktionen, die die Schichten kommunizieren heißen Dienste
- Jede Schicht kann Dienste der nächst niedrigeren Schicht abrufen und an die nächsthöhere Schicht weitergeben
- Die Gesamtheit der Vorschriften, die bei der Nutzung der Dienste einzuhalten sind heißen Protokoll
- Das UEFI (unified extensible firmware interface) stellt die Firmware-Schnittstelle dar

Was ist der Unterschied zwischen einem Programm und einem Prozess?

Programm: Statische Handlungsvorschrift

Prozess: In Aktion befindes Programm

Zeitlicher Charakter ist Unterschied

Nebentätigkeiten:

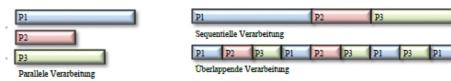
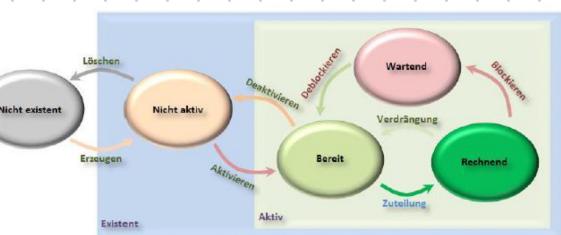


Schaubild der Prozesszustände

Prozesssynchroisation

Die zwei Klassen der Abhängigkeit:



- Die Prozesse konkurrieren über die Nutzung gemeinsamer, exklusiv nutzbarer Betriebsmittel

- Die Prozesse sind voneinander datenabhängig

Prozess-Kommunikation

- Kommunikation über gemeinsame Speicherbereiche: Prozesse können gemeinsame Datenbereiche etc. nutzen
- Kommunikation über gemeinsame Dateien: Prozesse schreiben in Dateien, die von anderen Prozessen gelesen werden. Dateisystem muss dies jedoch erlauben
- Kommunikation über Streams: Streams ermöglichen die Kommunikation über Rechnernetze.

• Kommunikation über Signale: Signale sind asynchron auftretende Ereignisse, die eine Unterbrechung bewirken.

Scheduling:

ausführen!

- FCFS (first come first serve)
- LCFS (last come first serve)
- round robin = jeder Prozess bekommt eine feste Zeitspanne zugewiesen
- SJN (shortest job next)
- SRTN (shortest remaining time next)
- Prioritätssteuerung

