IT ABI Zusammenfassung

Jannis Müller, Robin Rausch

03 April 2021

Inhaltsverzeichnis

1 Zahlensysteme

1.1 Dezimalsystem

Nennwerte:	0,1,[. . .],8,9
Basis:	10
Größter Nennwert:	9

10^1	10^0
10	1
0	0
0	1
0	2
0	3
0	4
0	5
0	6
0	7
0	8
0	9
1	0
1	1
1	2
1	3
1	4
1	5

1.2 Dualsystem

Nennwerte:	0,1
Basis:	2
Größter Nennwert:	1

2^3	2^2	2^1	2^0
8	4	2	1
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

1.3 Oktalsystem

Nennwerte:	0,1,[. . .],6,7
Basis:	8
Größter Nennwert:	7

8^1	8^0
8	1
0	0
0	1
0	2
0	3
0	4
0	5
0	6
0	7
1	0
1	1
1	2
1	3
1	4
1	5
1	6
1	7

1.4 Hexadezimalsystem

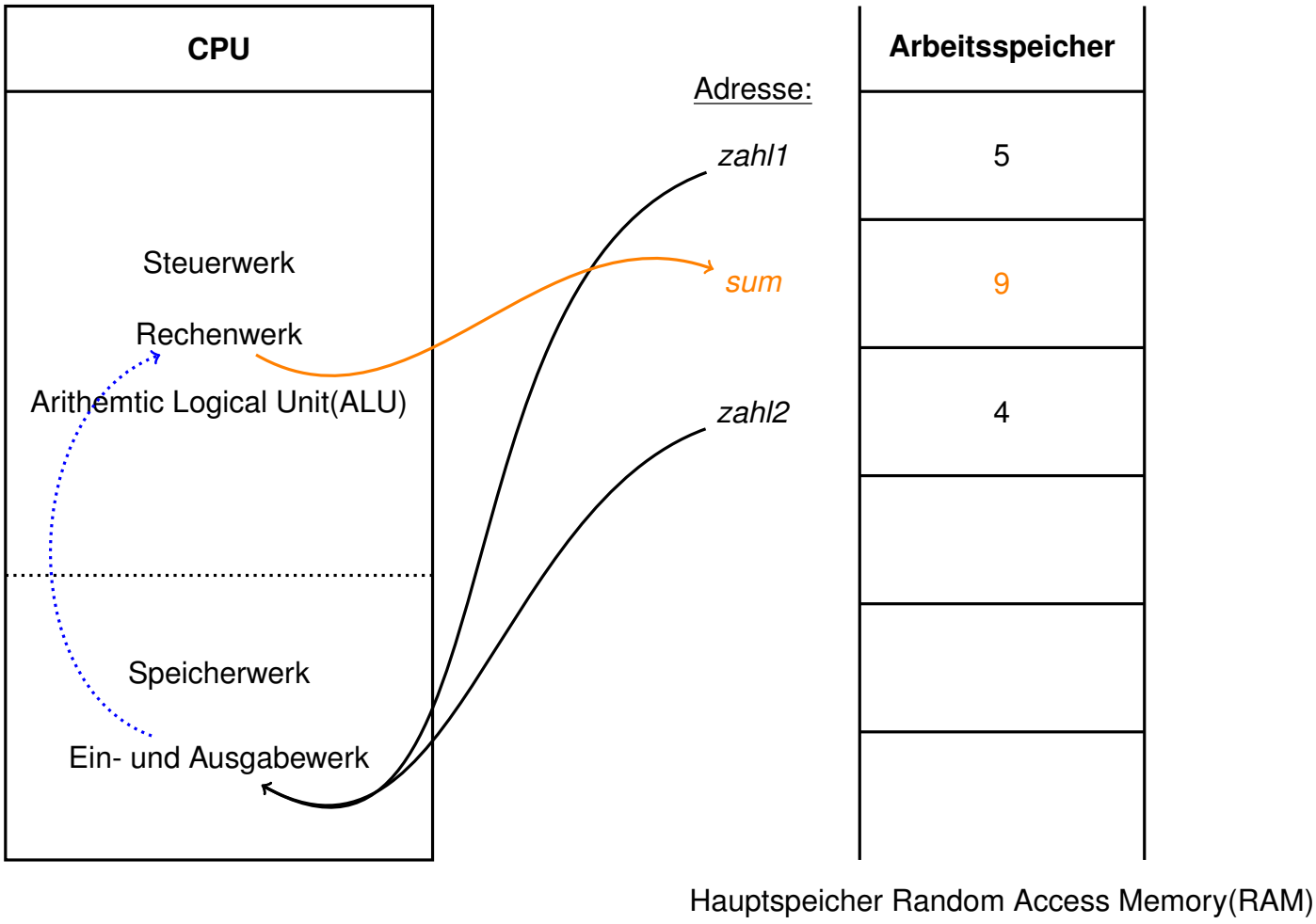
Nennwerte:	0,1,[. . .],8,9,A,B,C,D,E,F
Basis:	16
Größter Nennwert:	F

16^1	16^0
8	1
0	0
0	1
0	2
0	3
0	4
0	5
0	6
0	7
0	8
0	9
0	A
0	B
0	C
0	D
0	E
0	F

2 Aufbau Computer

2.1 Von Neumann Prinzip

Quellcode:
zahl1 = 5;
zahl2 = 4;
sum = zahl1 + zahl2;



3 Betriebssysteme

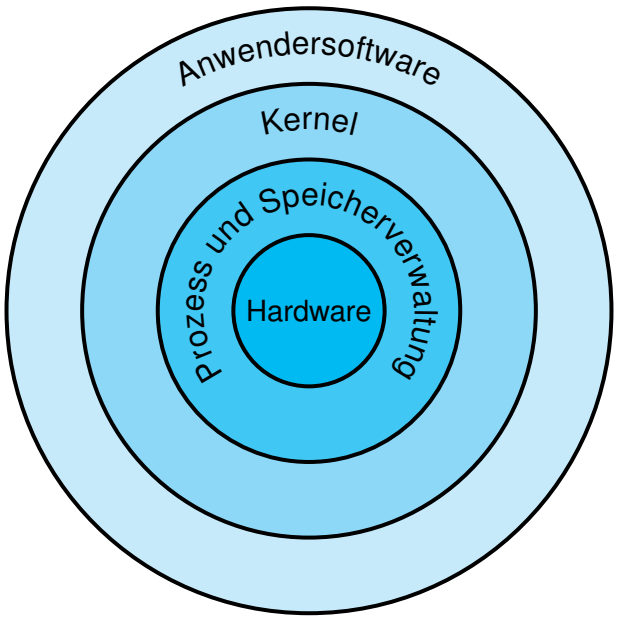
3.1 Kernel

Der Kernel ist die Schnittstelle zwischen der Anwendersoftware und der Speicher- und Prozessverwaltung. Er kann auch als Betriebssystem-Kern bezeichnet werden.

3.1.1 Monolitischer Kernel

+ sehr schnell, da alles nah bei einander liegt

- wenn eine Komponente defekt ist oder einen Fehler hat, ist der Kernel nichtmehr funktionsfähig



- Updates müssen im Gesamten direkt erfolgen

Der Monolitische Kernel wird bei Betriebsystemen wie z.B. DOS, Linux und Android verwendet.

3.1.2 Mirkokernel

+ wenn eine Komponente defekt ist oder einen Fehler hat, kann man den Kernel trotzdem weiterhin verwenden

- Prozesse dauern länger, da nicht Alles bei einander liegt

Der Mirkokernel wird bei Betriebsystemen wie z.B. Echtzeitsbetriebssystemen oder RISC OS verwendet.

3.1.3 Hybridkernel

+ Nutzung beider Vorteile

- Nutzung beider Nachteile

Der Hybridkernel wird bei Betriebsystemen wie z.B. Windows oder MAC OS verwendet.

3.2 Master Boot Record (MBR)

Der Master Boot Record (kurz MBR) enthält ein Startprogramm für BIOS-basierte Computer und eine Partitionstabelle. Außerdem werden durch den MBR die Partitionen einer Festplatte eingeteilt. Zudem kann man die Betriebssysteme und Dateisysteme der einzelnen Partitionen herauslesen.

Tabelle 1: Partitionstabelleneinträge

Adresse	Inhalt
00	80hex = bootfähige Partition 00hex = nicht bootfähige Partition
01	CHS-Eintrag des ersten Sektors
04	Typ der Partition (Dateisystem/Partitionsytp)
05	CHS-Eintrag des letzten Sektors
08	Startsektor nach LBA
0C	Anzahl der Sektoren in der Partition

Tabelle 2: Master Boot Record

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	eb	48	90	10	8e	d0	bc	00	b0	b8	00	00	8e	d8	8e	c0
0010	fb	be	00	7c	bf	00	06	b9	00	02	f3	a4	ea	21	06	00
...																
01a0	10	ac	3c	00	75	f4	c3	00	00	00	00	00	00	00	00	00
01b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	80	01
01c0	01	00	83	fe	ff	ff	3f	00	00	00	41	29	54	02	00	fe
01d0	ff	ff	82	fe	ff	ff	80	29	54	02	fa	e7	1d	00	00	fe
01e0	ff	ff	83	fe	ff	ff	7a	11	72	02	fa	e7	1d	00	00	fe
01f0	ff	ff	05	fe	ff	ff	74	f9	8f	02	0c	83	6c	04	55	aa

3.3 GPT

Die Guide Partition Table ist der bessere MBR und hat folgende Vor- und Nachteile gegenüber dem MBR:

+ Funktioniert auch bei 64-Bit Systemen
+ beliebig viele primäre Partitionen
+ verfügt über Schutz- und Sicherheitsmerkmale(Backups & Prüfsummen)
- Nur mit UEFI kompatibel

3.4 UEFI und BIOS

UEFI ist das modernere und bessere BIOS. UEFI verfügt gegenüber BIOS folgende Vorteile:

- schnelleres Starten durch paralleles Laden der Treiber
- Datenträger können auch mit mehr als 2TB booten
- Funktioniert auch bei modernen 64-Bit Systemen
- Netzwerke können auch gebootet werden

4 Dateisysteme

4.1 FAT

4.2 EXT

4.3 NTFS

4.3.1 Dataruns

5 Prozessverwaltung

5.1 Prozesse

5.2 Threads

5.3 Scheduling

5.4 Prozesssynchronisation

5.4.1 Shortest Remaining Time

5.4.2 First Come first Serve

5.4.3 Shortest Job First

5.4.4 Round-Robin

6 Speicherkonzepte

6.1 Speicherhierarchie

Speicher werden nach ihrer Schnelligkeit in eine Hierarchie eingeordnet, von kurzer zu langer Zugriffszeit.

1. Prozessorregister
2. Prozessorcache
3. Hauptspeicher
4. Festplatte
5. Wechseldatenträger

7 Speicherverwaltung

7.1 Direkte Speicherverwaltung

Beim Einprogrammbetrieb verwaltet das Betriebssystem den Hauptspeicher als einen großen Speicherblock. Dieser besteht aus System und Benutzerbereich.

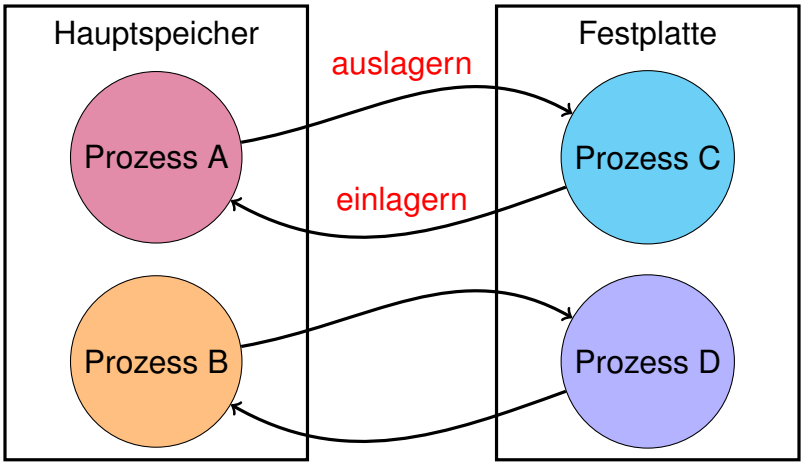
Betriebssystem	Programm	freier Speicher

Problem:
Die Programmgröße ist durch die Speichergröße begrenzt. Der Hauptspeicher ist somit zu klein für alle aktiven Prozesse.
Lösung: Swapping

7.2 Swapping

Beim Swapping wird ein Bereich der Festplatte als sogenannter *'Swap-Space'* deklariert. In diesen können Prozesse ausgelagert werden. Daten und Programmcode des Prozesses werden vollständig in den Swap Space ausgelagert.

Ein ausgelagerter Prozess kann nicht ausgeführt werden.



Demnach wären hier nur Prozess A und Prozess B ausführbar.

Ein Prozess wird ausgelagert wenn:

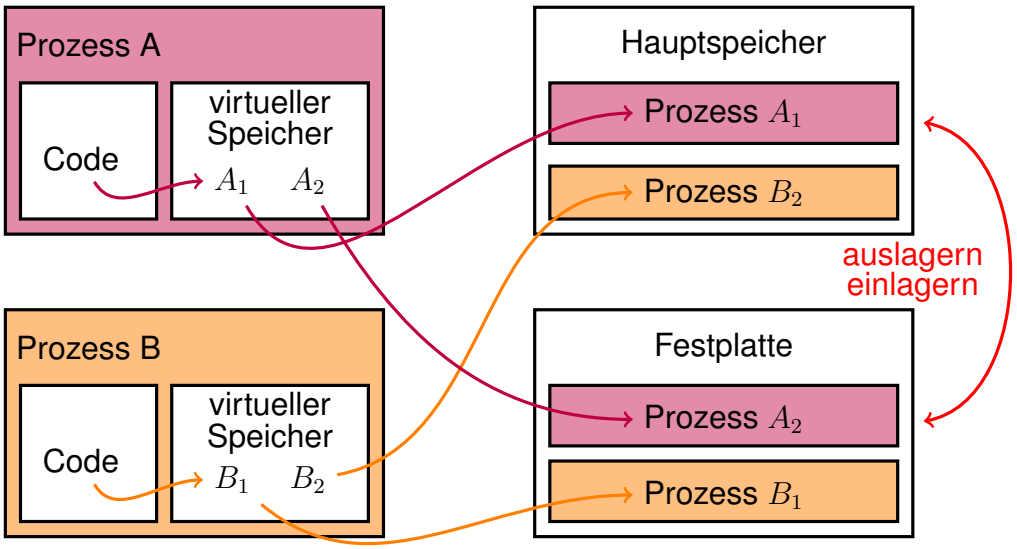
- Ein Prozess, der neu entsteht nicht im Hauptspeicher untergebracht werden kann.
- Ein Prozess, seinen Hauptspeicherplatz erweitern möchte, jedoch kein zusätzlicher Speicher verfügbar ist.
- Ein *wichtigerer* Prozess aus dem Swap-Space eingelagert werden muss und für ihn kein Hauptspeicherbereich verfügbar ist.

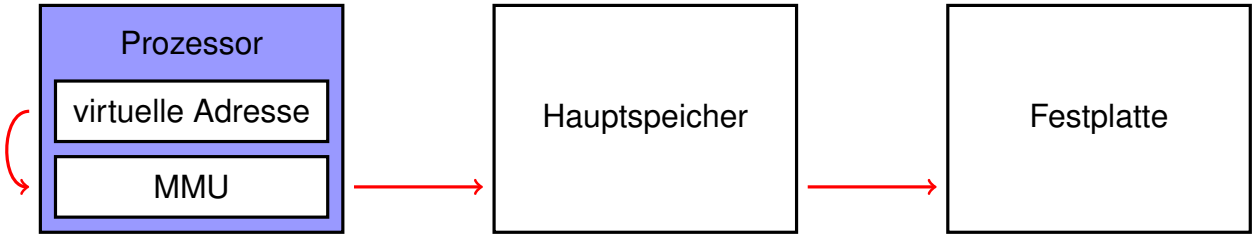
7.3 virtueller Speicher

Festplattenspeicher und Hauptspeicher werden zu einem Speicherblock vereint. Dadurch stehen mehr Speicheradressen zur Verfügung, wie tatsächlich im Hauptspeicher vorhanden sind = virtueller Speicher.

Im Gegensatz zum Swapping können beim virtuellen Speicher Prozesse auch nur teilweise im Hauptspeicher stehen, während der Rest meist auf dem Festplattenspeicher liegt.

Jedem Prozess wird ein virtueller Speicher geboten, in den alles außer arbeitenden Code und Daten (bleiben im Hauptspeicher) eingelagert wird. Wird von einem Prozess auf den Speicher zugegriffen, verweist die virtuelle Adresse auf eine reelle Adresse des Hauptspeichers. Dieser Verweis geschieht durch die *MMU (Memory Management Unit)*. Befinden sich die Daten in einem ausgelagerten Festplattenblock, wird dieser in den Hauptspeicher eingelagert.
Durch den Einsatz eines Virtuellen Speichers ergeben sich folgende Vorteile:

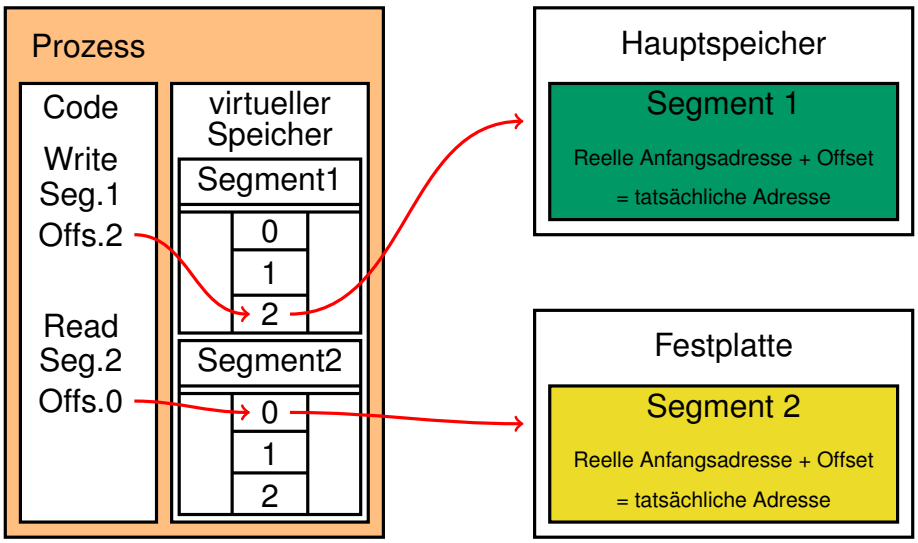




- Programmbereiche und Datenbereiche sind in ihrer Länge nicht durch die reelle Größe des Hauptspeichers begrenzt.
- Es können gleichzeitig mehrere Programme ausgeführt werden, deren Gesamtlänge die Hauptspeichergröße überschreitet.

7.4 segmentorientierter Speicher

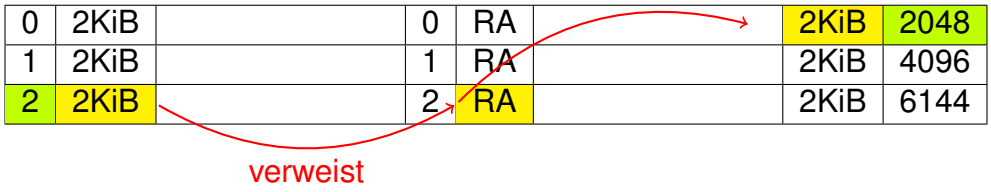
Der Adressraum wird in Segmente variabler Größe entsprechend der der logischen Einheiten des Programms unterteilt. Jedem Segment ist ein Speicherbereich im Haupt oder Festplattenspeicher zugeteilt. Die Virtuelle Adresse besteht aus zwei Teilen, der Segmentnummer und dem Offset. Dieser gibt die Position innerhalb des Segments an.
Der Prozess gibt mit der virtuellen Adresse an, auf welches seiner Segmente und auf welche Position innerhalb dieses Segmentes er zugreifen möchte. Die Umrechnung von virtuellen in reelle Adressen geschieht mit der Segmenttabelle. Jeder Prozess besitzt eine eigene Tabelle. Die Segmenttabelle eines Prozesses enthält für jedes Segment einen Eintrag mit folgenden Informationen:



- Reelle Anfangsadresse des Segments
- Valid-Bit (gibt an, ob sich das Segment auf dem Hauptspeicher oder auf der Festplatte befindet)
- Länge des Segments
- Zugriffsrechte

7.5 seitenorientierter Speicher - *Paging*

virtuelle Adresse Seitentabelle reelle Adresse



Die Seitentabelle stellt die *Schnittstelle (Interface)* zwischen virtueller und reeller Adresse dar. Die virtuelle Adresse, welche hauptsächlich zum leichteren Programmieren eingeführt wurde, verweist auf die zugehörige Seite der Seitentabelle. Die Seitentabelle stellt demnach ein 1:1 Abbild der virtuellen Adresstabelle dar. Diese wiederum adressiert die reelle Startadresse im *Hauptspeichers (RAM)*. Demnach muss für jede reelle Adresse eine Seitenspalte existieren.
Wichtig: die Größe der Segmente der virtuellen und reellen Adresse sind *immer* gleichgroß! (Hier: 2kiB)

7.5.1 Beispielaufgaben

7.5.2 Buch Aufgabe 1, S.243

Die Größe einer Seite beträgt 4096 Byte. Es wird nun geprüft, in welche Seite die virtuelle Anfangsadresse "reinpasst". Z.b.: die virtuelle Anfangsadresse 9000 passt nur in die Seite 2. Diese geht von 8182-12288. Der Offset wird berechnet indem man die gesamt Seitengröße (0; 4096; 8192; 12288; . . .) von der virtuellen Anfangsadresse abzieht. (Bsp.: Offset für

Seitengröße	0	4096	8182	12288	16384	20480
Seite	0	1	2	3	4	5
Startadresse	12288	P3	32678	0	8192	P12

Seiten Nummer	virtuelle Anfangs-adresse	reelle Anfangs-adresse	Offset	reelle Adresse
0	2000	12288	2000	14288
1	5000	P3	904	P3 + 904
2	9000	32768	808	33576
3	14000	0	1712	1712
4	18000	8192	1616	9808
5	21000	P12	520	P12 + 520

Offset $9000 = 9000 - 8192 = 808$)
Die reelle Adresse wird berechnet indem man die reelle Anfangsadresse und den Offset addiert. (Bsp.: reelle Adresse für $18000 : 8192 + 1616 = 9808$)

Nun wird die reelle Anfangsadresse der Seite 2 in den Festplattenblock *P23* verschoben, dadurch wird ein Hauptspeichersegment frei. In dieses wird nun der Festplattenblock *P12* geladen. Die Tabelle ändert sich nun wie folgt:

Seiten Nummer	virtuelle Anfangs-adresse	reelle Anfangs-adresse	Offset	reelle Adresse
0	2000	12288	2000	14288
1	5000	P3	904	P3 + 904
2	9000	P23	808	P23 + 808
3	14000	0	1712	1712
4	18000	8192	1616	9808
5	21000	32768	520	33288

7.5.3 Buch Aufgabe 3, S.243

Ein Prozessor besitzt virtuelle Adressen mit *32 Bit*, physische Adressen (reell) mit *28 Bit* und Seiten mit *2 KiB*.
Wie viele Bit werden für die virtuellen und physischen (reell) Seitennummern benötigt?

ausführliche Lösung:
Um *2KiB* darstellen zu können, werden *11Bit* benötigt:

$$2KiB = 2048Byte = 2^{11}$$

Die gesamte Länge der virtuellen Adressen beträgt *32Bit*, wovon *11Bit* dazu verwendet werden, jede einzelne Adresse in einem Adressblock von *2KiB* zu adressieren. Die *11Bit* stellen demnach den Offset dar. Daraus ergibt sich:
Es bleiben noch *21Bit*, die dazu verwendet werden können, die Anzahl von Adressblöcken anzugeben.

$$32Bit - 11Bit = 21Bit$$

$$28Bit - 11Bit = 17Bit$$

Es ergeben sich demnach 2^{21} adressierbare virtuelle Speicheradressen mit je *2KiB*, und 2^{17} adressierbare reelle Adressen mit je *2KiB*.

7.5.4 Buch Aufgabe 4, S.243

virtuelle Adressen = *48Bit* reelle Adressen = *36Bit*
Hauptspeichergröße = *128MiB* Seitengröße = *4KiB*

a) $4KiB = 4096 = 2^{12}$ $12Bit = Offset$
$48Bit - 12Bit = 36Bit \longrightarrow$ virtuelle Speicheradressen = 2^{36}
$36Bit - 12Bit = 24Bit \longrightarrow$ reelle Speicheradressen = 2^{24}

b) $\frac{128MiB}{4KiB} = 32786 = 2^{15}$

1 Prozess = $2^{24} \longrightarrow$ große Festplattenauslagerung
verfügbarer RAM = 2^{15} weil RAM kleiner als Prozess

7.5.5 Buch Aufgabe 5, S.243

virtuelle Adresse = *32Bit* reelle Adresse = *24Bit* Seitengröße = *2KiB*

a) Wie viele Seitentabelleneinträge werden in diesem System benötigt?
Lösung:

Seitentabelleneinträge = virtuelle Adressen

$$32Bit - 2^{11} = 21Bit$$

$$21Bit = 2^{21} = \textit{Seitentabelleneinträge}$$

b) Wie groß ist jeder Seitentabelleneintrag auf das nächste volle Byte aufgerundet?
Lösung:

$$24Bit - 11Bit = 13Bit$$

13Bit aufgerundet = *16Bit*

$$16Bit = 2Byte$$

c) Wie viel Speicherplatz ist für die Seitentabelle erforderlich?
Lösung:

$$\frac{2^{21} \cdot 2Byte}{1024} = 4MiB$$

7.5.6 Buch Aufgabe 6, S.243

Bestimmen Sie die Seitentabelle mit Seitennummer und Seitenrahmen für:

virtueller Speicher = *16MiB* reeller Speicher = *4MiB* 4 Seitenrahmen

Seitenrahmen = reelle Speichergröße

$$\frac{4MiB}{4Seitenrahmen} = \textit{eine Seitenrahmengröße}$$

Seitennummer	0	1	2	3	bis 15
Seitenrahmen	1MiB	1MiB	1MiB	1MiB	P0-P11

virtueller Speicher = *2GiB* reeller Speicher = *265MiB* 16 Seitenrahmen

Seitennummer	0	1	[...]	15	bis 127
Seitenrahmen	16MiB	16MiB	16MiB	16MiB	P0-P111

virtueller Speicher = *2GiB* reeller Speicher = *8MiB* 16 Seitenrahmen

Seitennummer	0	1	[...]	15	bis 4095
Seitenrahmen	0.5MiB	0.5MiB	0.5MiB	0.5MiB	P0-P4079

7.5.7 Buch Aufgabe 7, S.243

virtueller Speicher = *16Bit* reeller Speicher = *8Bit* Offset = *4Bit*

Die angeforderte virtuelle Adresse lautet: 1000 0011 0000 1111

Die Seitentabelle verweist für die Seite 1000 0011 0000 auf den Seitenrahmen 0101

a)

Gegebenheiten:

virtueller Adress Rahmen = Seitenrahmen (SR) = reeller Adress Rahmen

virtuelle Adressen = Seiteneinträge

virtueller Speicher = VS virtuelle Adresse = VA	ges. Seitengröße = SG Seitenadressen = SA	reeller Speicher = RS reelle Adresse = RA
16Bit-4Bit Offset = *12Bit* VA Offset $2^4 = 16Byte = VARahmen$ $\frac{16 \cdot 2^{12}}{1024} = 64KByte = VS$	Offset $2^4 = 16Byte = SR$ $\frac{16 \cdot 2^{12}}{1024} = 64KByte = SG$	*8Bit-4Bit* Offset = *4Bit* RA Offset $2^4 = 16Byte = RARahmen$ $2^4 \cdot 16Byte = 265Byte = RS$

b)

Frame Number = 0101	Offset der Adresse = 1111

Die reelle Adresse lautet demnach: 0101 1111

7.6 Speicherverwaltungsmethodenchronologie

direkte Speicherverwaltung > Swapping > seitenorientierte Speicherverwaltung > segmentorientierte Speicherverwaltung

8 Datensicherung

Ein Backup dient als Schutz vor:

- Schadsoftware z.B. Viren
- versehentliches Datenüberschreiben oder Löschen
- Hardware Schäden
- logische Fehler in einer Datei
- Diebstahl

8.1 Sicherungsmedien

Bekannte Sicherungsmedien sind:

- DVD's
- USB - Sticks
- DLT (Digital Line Tape)
- DAT Laufwerk

Wichtig bei Sicherungsmedien ist, dass die Backup Zeit so klein wie möglich bleibt! Eine Formel zur Berechnung der Backup Zeit wäre:

$$t_b = \frac{m}{r_b}$$

t_b = Backup Zeit m = Datenmenge r_b = Schreibrate in Bit/s

8.2 Datenverlust

Durch falsche Lagerung, äußere Gewalteinwirkung, aussetzen von bestimmten Strahlungen (Magnetfeld), kann es zu Schäden der Dateien auf dem Datenträger kommen. Auch bei versehentlichem Löschen einer Datei kann diese noch wiederhergestellt werden. Beim Löschen oder Formatieren werden Dateien nicht wirklich *vernichtet*, generell werden sie nur ausgeblendet bzw. Verzeichniseinträge werden gelöscht. Man unterscheidet zwischen drei *Recovery Verfahren:*

1. Einfache Rettung:
Eine Datei wurde versehentlich gelöscht, der Eintrag über die gelöschte Datei im Inhaltsverzeichnis des Datenträgers ist noch vorhanden. Die Datei kann mit Hilfe einer *Recovery Software* wiederhergestellt werden.

2. Aufwendige Rettung:
Der Eintrag im Verzeichnis ist nicht mehr vorhanden. Die Datei muss mithilfe spezieller Dateisignaturen, die den Dateityp, den Dateianfang und das Dateiende enthalten, erkannt und wiederhergestellt werden.

3. Schwierige Rettung:
Ist die Datei darüber hinaus auch noch fragmentiere, erschwert das die Rettung erheblich. Die Recovery Software muss die Inhalte der Datenfragmente analysieren und sie dann zusammenfügen. Mit zusätzlicher Software kann der Datenverlust durch Erkennung von Verschleißerscheinungen vermieden werden.

8.3 Datenvernichtung

Es ist ratsam sensible Daten vor der Entsorgung eines Datenträgers durch verschiedene Maßnahmen annährend vollständig zu vernichten (es besteht keine 100% -ige Garantie, dass die Daten nicht wiederhergestellt werden können).
Angemessene Maßnahmen können:

- Säurebad

- zusätzliche Fragmentierung
- Überschreiben mit *Zufallsdaten oder Bitmustern*
- Hardwarezerstörung

sein.

8.4 Sicherungsverfahren

Abhängig vom Maschinentyp, Dateisensibilität und Anwendungsbereich können verschiedene Verfahren angewendet werden.

8.4.1 Vollständige Sicherung

Alle Daten werden auf einem Backup-Medium z.B. externer Festplatte gesichert.

Vorteile	Nachteile
Alle Daten liegen komplett vor. Keine lange Suche bei der Wiederherstellung. Zur Wiederherstellung verlorener Dateien wird nur die letzte Vollsicherung benötigt.	überflüssige Kopiervorgänge, Backups brauchen viel Zeit, hohe Datenmengen

$\longrightarrow$ Jeden Tag wird eine Vollsicherung durchgeführt.

8.4.2 Differenzielle Sicherung

Es werden die Daten gespeichert, die seit der letzten Vollsicherung erstellt oder verändert wurden. Praxisbezug: Es wird Täglich die Datenmenge gesichert, welche seit der letzten Vollsicherung erstellt oder verändert wurde.

Vorteile	Nachteile
Die Wiederherstellung ist unkomplizierter und schnell	Es wird mehr Zeit und Speicherplatz auf dem Sicherungsmedium im Vergleich zur *Inkrementellen Sicherung* benötigt.

$\longrightarrow$ Eine Vollsicherung wird einmal pro Woche durchgeführt. Die einzelnen Tage werden nicht unabhängig gesichert. Das Backup vom Mittwoch enthält die Sicherungen von Montag, Dienstag und Mittwoch.

8.4.3 Inkrementelle Sicherung

Zuwachssicherung. Nur die Daten, die seit dem letzten Backup erstellt oder geändert wurden werden gesichert. Praxisbezug: Es wird Täglich nur die Datenmenge gesichert, welche erstellt oder verändert wurde.

Vorteile	Nachteile
Schnelle Datensicherung und wenig Speicherplatzbedarf auf dem Sicherungsmedium	Hoher zeitlicher Aufwand bei der Wiederherstellung, da die letzte Vollsicherung und darauffolgende Zuwachssicherungen benötigt werden.

$\longrightarrow$ Eine Vollsicherung wird einmal pro Woche durchgeführt. Die einzelnen Tage werden unabhängig voneinander gesichert. Das Backup vom Mittwoch enthält nur den Tag Mittwoch.

8.4.4 Sicherungsstrategie: Generationen Konzept

Die einzelnen Backups können nach dem *Generationen Konzept* logisch geordnet werden:

Söhne
Die am einzelnen Tag ausgeführten Backups werden als Söhne bezeichnet.

Väter
Die Vollsicherungen (einmal pro Woche) werden als Väter bezeichnet.

Großväter
Die Vollsicherungen im Zeitraum eines Monates werden als Großväter zusammengefasst.

8.5 Vernetzte Sicherung

Vor allem in Firmen arbeiten die Mitarbeiter, jeder eigenständig, an Computern. Um die entstandenen Datenmengen zu sichern, werden Sicherungen meist über ein Netzwerk durchgeführt. Dafür können folgende Strategien angewendet werden:

8.5.1 verteilte Sicherung

Jeder Rechner ist mit einem eigenen Sicherungsmedium im Netzwerk (Backup Server) ausgestattet. Jeder Rechner wird einzeln gesichert, es kann aber über das Netzwerk von jedem Rechner aus auf die Backups jedes anderen zugegriffen werden. Alle Backups sind voneinander digital getrennt.

8.5.2 lokale Sicherung

Jeder Rechner ist mit einem eigenen Sicherungsmedium, unabhängig von einem Netzwerk ausgestattet. Es kann nicht über das Netzwerk auf Backups anderer Rechner zugegriffen werden. Alle Backups sind voneinander physisch getrennt.

8.5.3 Netzwerk Sicherung

Alle Rechner besitzen eine Anknüpfung zum Netzwerk, und erstellen ihre Backups zusammen auf einem Backup Server. Die Effizienz der Backup Medien steigt, da alle Rechner auf diese sichern können. Es kann zu Engpässen oder Netzwerküberlastung kommen, wenn z.B. alle Rechner gleichzeitig hohe Datenmengen sichern wollen. Alle Daten sind weder digital noch physisch getrennt, da sie alle auf das selbe Sicherungsmedium (Backup Server) gesichert werden. Der Backup Server ist jedoch physisch von den einzelnen Rechnern getrennt, was die Sicherheit erhöht.

8.5.4 Kapazität des Netzwerks

Die Transportkapazität des Netzwerkes stellt bei der Netzwerk und lokalen Sicherung einen limitierenden Faktor dar. Die Backup Zeit t lässt sich mit folgender Formel berechnen:

$$t = \frac{m}{G}$$

m= Datenmenge G= Übertragungsgeschwindigkeit $\frac{Mbit}{s}$

8.5.5 Logfiles

Alle bisher genannten Verfahren haben den Nachteil, dass bei z.B. Virusbefall auf das letzte vorhandene Backup zurückgegriffen werden muss, und somit Transaktionsdaten das laufenden Tages verloren gehen. Um dies zu verhindern ist es ratsam eine *Logfile* zu führen, in der jede Änderung an Dateien gespeichert wird. Die *Logfile* enthält alle Datenänderungen seit dem letzten Backup.

8.6 RAID Systeme

RAID = Redundant Array of Independent Disks
ist ein Sicherungsmedien Verband, mit dem folgende Ziele verfolgt werden können:

- Erhöhung der Ausfallsicherung durch Redundanz
- Steigerung der Transferraten
- Austausch von Sicherungsmedien während des Systembetriebs.

8.6.1 RAID 0

Eine zu sichernde Datei wird in verschiedene Segmente aufgeteilt. Jedes Segment wird auf einem seperaten Sicherungsmedium gesichert. Dadurch werden hohe Transferraten durch Aufteilung der Datenmenge auf verschiedene Sicherungsmedien erreicht. Es besteht keine Datensicherheit. Geht ein Sicherungsmedium kaputt, Sind alle Dateien im RAID verband verloren, da jede Datei in Segmente unterteilt wird, und somit ein Segment von jeder Datei auf jedem Sicherungsmedium liegt.

8.6.2 RAID 1

Bei diesem Verfahren werden alle Daten doppelt abgelegt. Geht ein Sicherungsmedium kaputt, sind alle Dateien auf einem zweiten im RAID System verknüpften Sicherungsmedium verfügbar. Kaputte platten können einfach ausgetauscht werden. Es kann nur 50% der Gesamtkapazität der im RAID Verband verknüpften Sicherungsmedien genutzt werden, da alles doppelt vorhanden ist. Es besteht kein Performance Gewinn.

8.6.3 RAID 5

Es werden mindestens drei Sicherungsmedien benötigt. Dieses Verfahren unterscheidet sich von RAID 0 in nur einem Punkt. Es werden zusätzlich Paritäten ermittelt, und auf einem

Sicherungsmedium gespeichert, um bei einem Ausfall Dateien vollständig zu rekonstruieren.
Die nutzbare Sicherungskapazität wird wie folgt berechnet:

$$K_5 = s \cdot (n-1)$$

K_5= nutzbare Kapazität s= kleinste Platte im Array n Anzahl der Platten

8.6.4 RAID 10

verbindet RAID 0 und 1. Im Prinzip funktioniert dieses System wie RAID 0, nur dass alle Dateien zusätzlich doppelt vorhanden sind. Durch dieses System erhält man eine hohe Datensicherheit, und gute Performance.

8.6.5 RAID 50

Kombination aus RAID 5 und 0. Die Performance im Vergleich zu RAID 10 steigt nicht. Es werden mindestens 6 Sicherungsmedien benötigt.
Die nutzbare Sicherungskapazität wird wie folgt berechnet:

$$K_{50} = s \cdot \left(\frac{n}{2} - 1\right)$$

K_{50}= nutzbare Kapazität s= kleinste Platte im Array n Anzahl der Platten

8.6.6 Gegenüberstellung der RAID Systeme

System	**Kenngrößen**
RAID 0	hoher Datendurchsatz, 100% Kapazität nutzbar, Keine Sicherheit
RAID 1	Hohe Sicherheit, kein Performance Gewinn, 50% der Kapazität nutzbar
RAID 5	Sicherheit, Performance Gewinn, >50% Kapazität nutzbar

9 Netze

9.1 Einführung

9.1.1 Beispiele für Netze

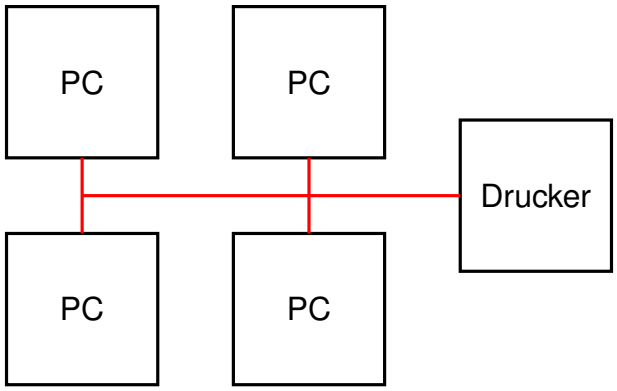
- Verkehrsnetz
- Smart Home
- Mobilfunknetz

- Stromnetz
- Soziale Netze (Familie, Freunde etc. . .)

9.1.2 Gemeinsamkeiten aller Netze

- Verbinden Dinge, Sachen, Knoten (Connection)
- Es bestehen immer Protokolle / Regeln (StVO)
- Dienen zum Informationsaustausch / Transport von Informationen

9.2 Einfaches Computernetz



Die Vernetzung verschiedener Systeme ermöglicht eine Verbesserung der Leistungsfähigkeit und reduziert Kosten. Computernetze erzielen diese Gegebenheiten hauptsächlich auf drei Arten:

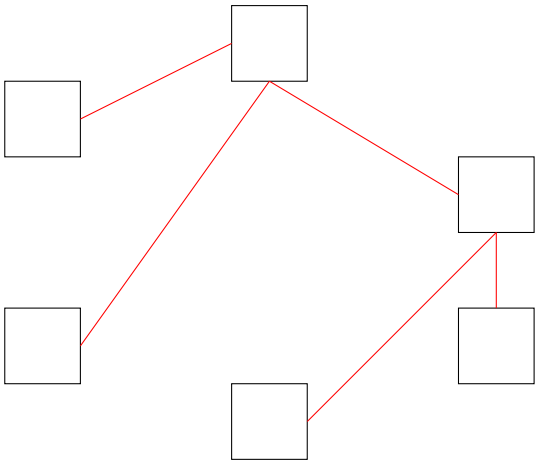
- Freigeben von Informationen / Daten
- Freigeben von Hard- und Software
- Zentralisieren von Verwaltung und Unterstützung

Spezifisch können Endgeräte eines Netzes folgende Komponenten freigeben:

- Betriebsmittel: Drucker, Maus, Monitor
- Sicherungsmedien: Festplatte, Laufwerke
- Sich auf diesen Sicherungsmedien befindende Daten
- Software

9.2.1 Netz Topologien

Baum Topologie



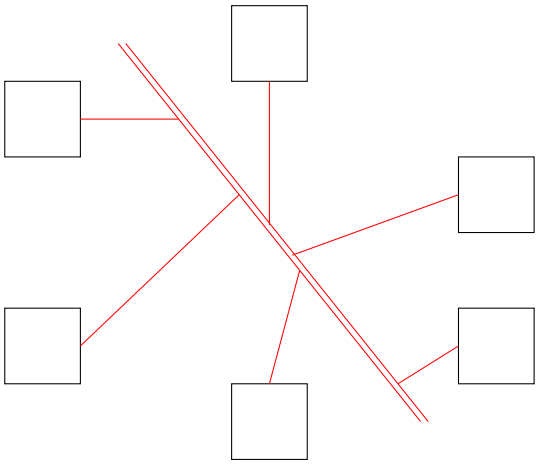
Vorteile:

- Bei Ausfall einer Komponente bricht nur ein Teil des Netzes zusammen
- leichte Skalierbarkeit

Nachteile:

- Durchsatzproblem an der Wurzel / jeder Netzkomponente $\longrightarrow$ höhere Laufzeiten

Bus-Topologie



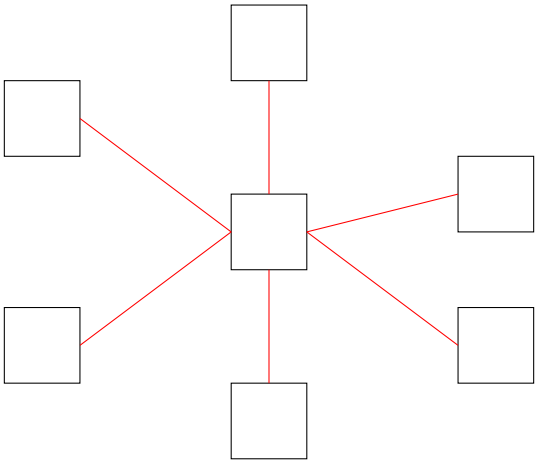
Vorteile:

- hohe Ausfallsicherheit
- leichte Skalierbarkeit

Nachteile:

- Ausfall der Hauptleitung $\longrightarrow$ Totalausfall
- Hauptleitung benötigt hohe Bandbreite

Stern Topologie



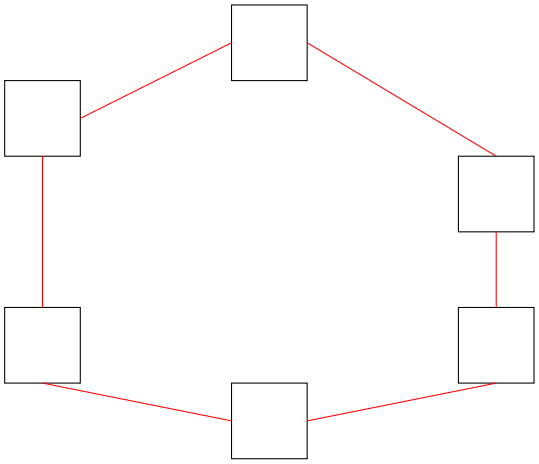
Vorteile:

- Leicht umsetzbar / skalierbar
- sehr schnell
- Beim Ausfall einer Komponente ist *nur* diese betroffen

Nachteile:

- Beim Ausfall der zentralen Wurzel $\longrightarrow$ Totalausfall
- Clients müssen immer über zentrale Wurzel kommunizieren.

Ring-Tropologie



Vorteile:

- simpler Aufbau
- leichte Skalierbarkeit

Nachteile:

- Unterbrechung des Rings $\longrightarrow$ Totalausfall

9.2.2 Netze in Unternehmen

Das Verwenden eines Netzes hat signifikante Vorteile für ein Unternehmen / Betrieb. Ressourcen und Betriebsmittel können gemeinsam genutzt werden (Zentral-Drucker). Dadurch können Kosten eingespart werden. Die Kommunikation innerhalb des Unternehmens wird durch verschiedene Netze (internes Telefon, Kommunikation zwischen den Systemen) um ein vielfaches erleichtert. Sollte das Unternehmen im Laufe der Zeit an Größe zulegen, ist die Skalierbarkeit des Systems im Allgemeinen unbegrenzt. Zusätzliche Systeme oder Server können problemlos hinzugefügt werden. Die Systemleistung eines einzelnen Computers kann außerdem gesteigert werden. Aufwendige Anwendungen können auf einem leistungsstarken Server ausgeführt werden, sind also nicht an die locale Leistung eines einzelnen Computers gebunden. Die Komponente der Teamarbeit im globalen Sinne wird durch ein globales Netzwerk zwischen Unternehmen außerdem vereinfacht.

9.2.3 Private Netze

Besitzt ein Haushalt einen Home Server mit Internetanbindung, so ist es Privatpersonen möglich, von nahezu überall auf der Welt auf ihre persönlichen Daten zugreifen zu können. Diese Option steht Privatpersonen außerdem durch Cloud Services zur Verfügung. Nachteil: Die Daten werden externen Firmen zur Verfügung gestellt, diese Option ist mit Vorsicht zu genießen. Privatpersonen können durch das Internet außerdem auf, generell betrachtet, Online Services zugreifen (Online Banking, Online Shopping). Die Kommunikation wird auch für Privatpersonen erheblich erleichtert. Durch Chat, Voice Chat oder sogar Video Chat Anwendungen ist die Kommunikation nahezu an jedem Ort der Erde möglich. Auch das Unterhaltungsprogramm profitiert vom Internet. Dienste wie YouTube oder Netflix wären ohne das Internet undenkbar.

9.3 Servermodelle

9.3.1 Ein-Server-Modell

Ein Computer / Server übernimmt alle zentralen Dienste (Standard)

9.3.2 Multi-Server-Modell

Mehrere Computer / Server teilen sich die Verwaltung (für große Netze)

9.4 Kommunikationsarten

9.4.1 Simplex

Einer spricht, der Rest hört zu

9.4.2 Halbduplex

Wechselseitiges Sprechen und Hören

9.4.3 Vollduplex

Jeder spricht und hört gleichzeitig

9.5 Netzwerkkomponenten

9.5.1 Client

9.5.2 Repeater

9.5.3 HUB

9.5.4 Bridge

9.5.5 Switch

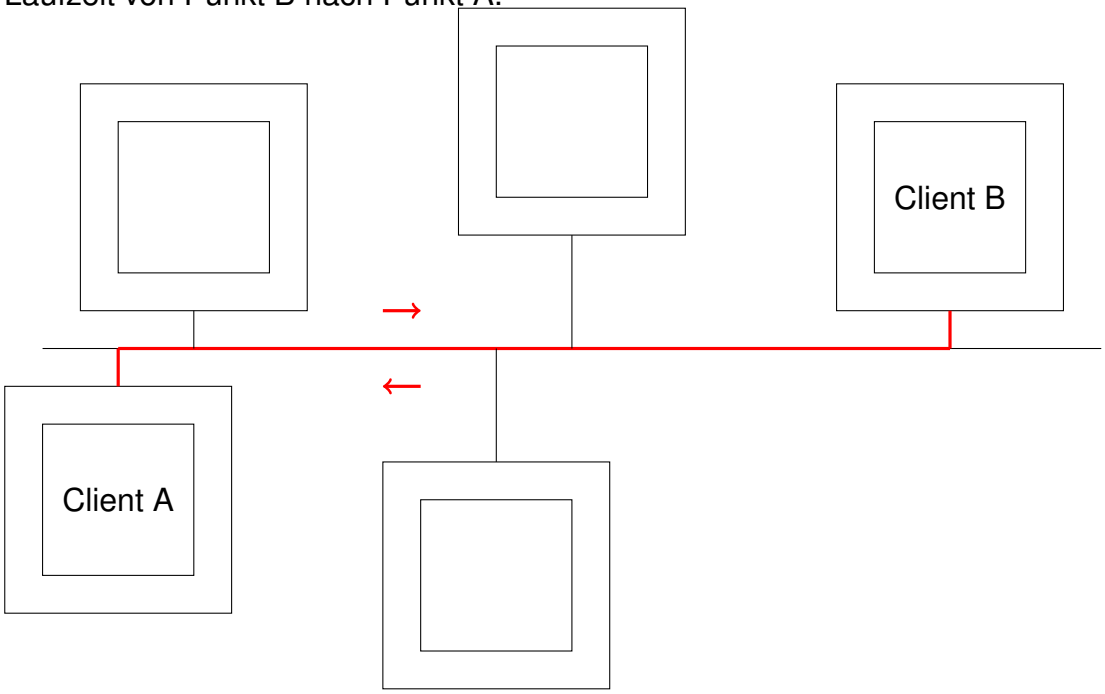
9.5.6 Port

9.5.7 Router

9.5.8 Einordnung der Komponenten ins OSI-Modell

9.6 Round Trip Delay Time - RTDT

Die Paketumlaufzeit bzw. Round Trip Time gibt die Zeit an, die ein Datenpaket in einem Rechnernetz benötigt, um von der weitesten entfernten Quelle zum Ziel und zurück zu reisen. Es handelt sich also um die Summe aus Laufzeit von Punkt A nach Punkt B und der Laufzeit von Punkt B nach Punkt A.

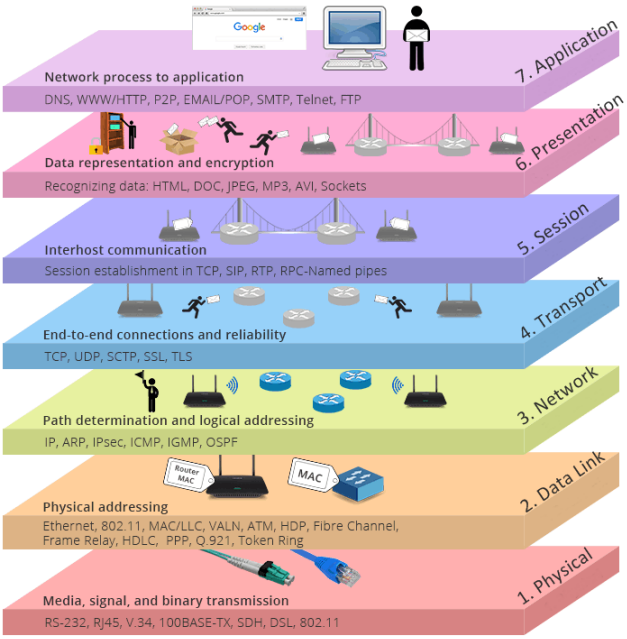


9.7 Bezeichnungen von Netzen

Abkürzung	Beschreibung	Kabellänge	Anwendungen
CAN	Control Area Network	5km	Steuerungstechnik
PAN	Personal Area Network	100m	Handynetz
LAN	Local Area Network	100m	Home Netz
MAN	Metropolitan Area Network	100km	Unternehmen mit mehreren Standorten
WAN	Worldwide Area Network	Weltweit	Internet, Telefon

9.8 OSI 7-Schichten Modell

Zweck des OSI-Modells ist, Kommunikation über unterschiedlichste technische Systeme hinweg zu beschreiben und die Weiterentwicklung zu begünstigen.

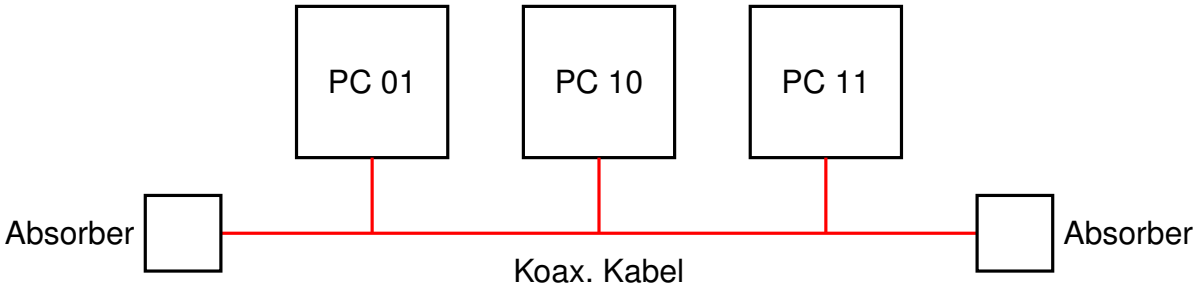


Hauptaufgaben der Schichten:

- Schicht 7: Anwendungen für Benutzer
- Schicht 6: Darstellung der Daten in verständliche Formate (jpg, ASCII)
- Schicht 5: Steuerung der Verbindung
- Schicht 4: Zuordnung der Datenpakete zu den Ports
- Schicht 3: Vermittelt Datenpakete
- Schicht 2: Fehlerfreie Übertragung
- Schicht 1: Bit-Übertragung

10 Ethernet

Mit der Erfindung des Ethernets wollte man individuellen Stationen / Systemen den Zugriff auf ein gemeinsames Medium zur Datenübertragung ermöglichen. Dieses gemeinsame Medium wurde durch ein Koaxial Kabel realisiert.



Soll ein Datenpaket verschickt werden (z.B. ein Zahlenwert), so muss der Sender das Ziel des Paketes kennen, um die Information zum korrekten Empfänger schicken zu können. Will *PC 01* die Zahl *5d* an *PC 10* senden, ergibt sich folgender Rahmen:

0010 *(Kennung des Ziels, 4Bit)*	0101 *(Daten, 4Bit)*

Damit der Empfänger den Erhalt der Daten quittieren kann, muss jedoch auch die Kennung des Senders im Rahmen enthalten sein:

0010 *(Kennung des Ziels)*	0001 *(Kennung der Quelle)*	0101 *(Daten)*

Der verschickte Rahmen lautet demnach:

0010 0001 0101

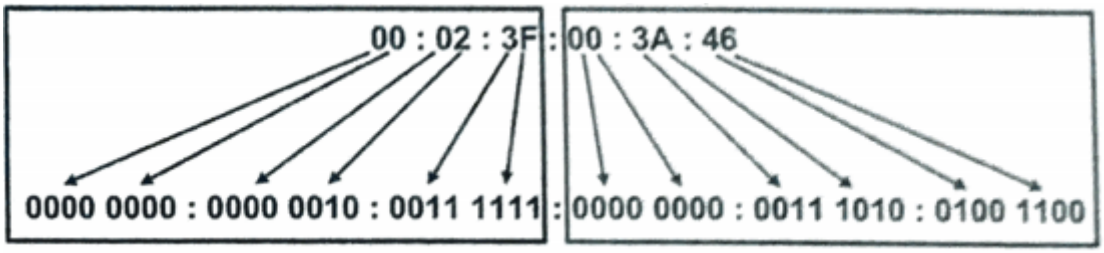
Um zu wissen, ob es sich bei den empfangenen Impulsen tatsächlich um eine gesendete Information handelt, wird ein Datenrahmen immer durch eine Präambel angekündigt. Es könnten sonst Missverständnisse durch Signalrauschen oder Störungen auftreten. Die endgültige Struktur eines Datenrahmens:

Präambel (7Byte)	SFD (1Byte)	Ziel (MAC)	Quelle (MAC)	Nutzdaten

10.1 MAC Adresse

Um Informationen im Ether / Internet zuverlässig und zielgenau verschicken zu können, muss jedes Endgerät im Netz seine eigene individuelle Kennung besitzen!
$\longrightarrow$ *MAC Adresse*
Die MAC Adresse ist in den ROM der Network Interface Card eines jeden Gerätes eingebrannt. Die MAC ist also keine virtuelle Softwarekennung, sondern eine durchaus physisch mit dem Gerät verbundene Kennnummer.

10.1.1 Aufbau MAC Adresse



3 Byte OUI (Herstellerkennung) 3Byte Gerät-Seriennummer

11 Netzwerkprotokolle

Hat die Netzwerkkarte einen Datenrahmen empfangen, muss ermittelt werden, welches Protokoll zur Weiterverarbeitung verwendet werden soll. Innerhalb des Datenrahmens muss also der Typ und das Ziel der Daten festgelegt sein. Diese Informationen stehen in einem 2Byte großem Typenfeld. Für jedes Protokoll existiert eine eigene Kennung:

ARP	0x0806
IPv4	0x0800
IPv6	0x86DD

Allerdings kommt dem Typenfeld eine weitere Bedeutung:

Wert kleiner als 0x0600	Länge des Datenrahmens
Wert größer als 0x0600	Protokollkennung

Alle Protokollkennungen müssen also größer als 1536d oder 0x0600h sein!

Achtung:
Das ICMP Netzwerkprotokoll ist ein Protokoll der IP-Familie und folgt somit dem IPv4-Protokoll!
Ebenso gehört das ICMPv6 Netzwerkprotokoll zur IPv6 Familie und folgt somit auch dem IPv6 Protokoll!

11.1 Datenübertragung

Um sicher zu gehen, dass alle Informationen fehlerfrei übertragen wurden, wird dem Datenrahmen ein 4Byte großes Prüffeld (CRC) angehängt. Dieses Prüffeld ergibt sich aus einer Polynomdivision.

Präambel	SFD	Ziel-Mac	Quell-MAC	Typ	Nutzdaten	CRC

Min: 64Byte - Max: 1518Byte

Unterschreitet der Anteil der Nutzdaten 46Byte, wird durch Padding aufgefüllt.

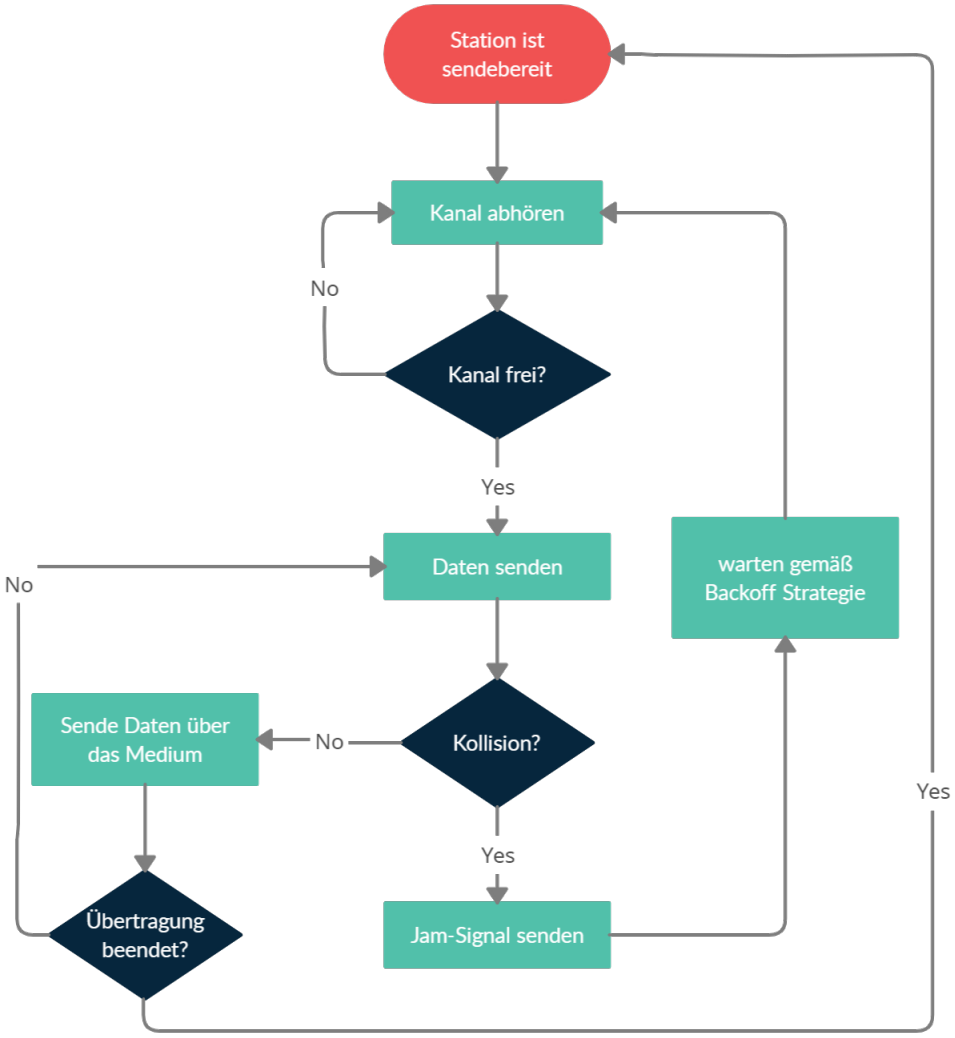
12 Vollständiges Ethernet-Paket

Präambel 7	SFD 1	Ziel-Mac 6	Quell-MAC 6	Typ 2	Nutzdaten 46-1500	CRC 4

Zahlenwert im Feld = Bytegröße des Feldes.

13 CSMA/CD - Verfahren

Programmablaufplan des CSMA/CD Verfahrens:

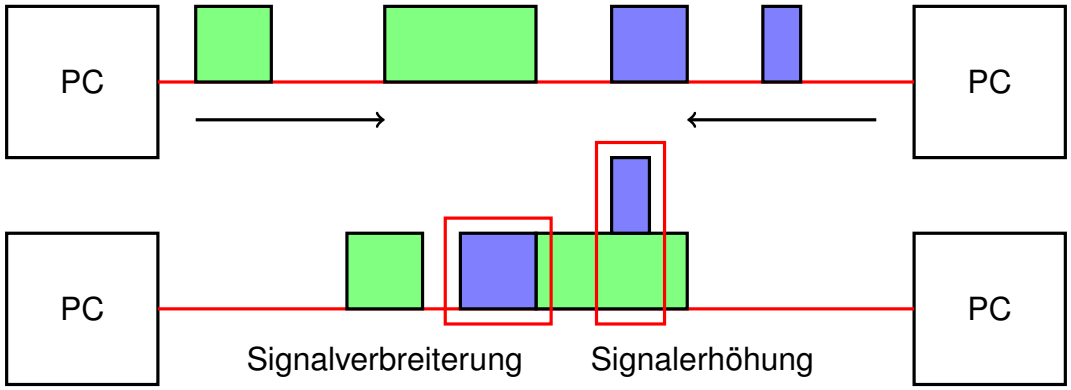


13.1 Kollision

Problem:
Da Alle Systeme mit nur einem Übertragungsmedium vernetzt sind, um Materialkosten zu sparen, und Skalierbarkeit zu erhalten, muss der Leiter bidirektional verwendet werden. Dadurch kann es zu Kollisionen von Impulsen kommen.
$\longrightarrow$ Es kommt zu einer Signalerhöhung / Signalverbreiterung und damit zu einer fehlerhaften Übertragung.

Lösung:
CSMA/CD Verfahren - Jam Signal

Situation:



Das sendende System vergleicht ständig das gesendete Signal mit dem Signal auf der Leitung. Kommt es zu Abweichungen bricht das System die Übertragung ab und sendet ein Jam-Signal welches andere Systeme im Netz über die Kollision informiert. Alle Systeme unterbrechen die Übertragung.
Ab wann dürfen die einzelnen Systeme wieder anfangen Daten zu senden?

Jedes Signal auf der Leitung braucht eine bestimmte Zeit, um zwischen den beiden entferntesten Systemen im Netz einmal hin, und wieder zurück zu laufen. Diese Zeit wird mit RTT bezeichnet. Ist die RTT abgelaufen, befinden sich keine Signale mehr im Netz, es kann neu gesendet werden.

Nach dem Ethernet-Standard 802.3 ist die RTT auf 51,2 Mikrosekunden festgelegt.

Kommt es nach dem Warten der RTT dennoch zu einer weiteren Kollision, variieren die Systeme ihre Wartezeit indem sie ein vielfaches der RTT warten. Als Vielfaches können die Faktoren 0,1,2 und 3 gewählt werden.
$\longrightarrow$ Es wird unterschiedlich lang gewartet.

Kommt es erneut zu einer Kollision, wird der Bereich der Vorfaktoren von 0 bis 7 erweitert.

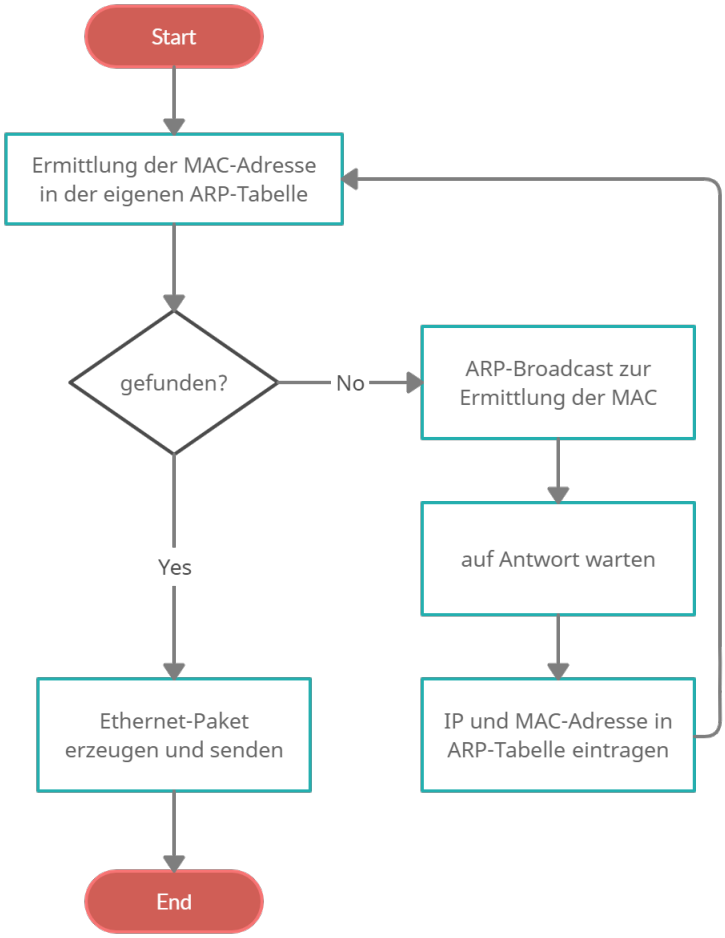
$$\text{Wartezeit} = k \cdot \text{RTD}$$

k = 0 bis 2^i-1 i = Anzahl Versuche

Nach 10 Versuchen wird i nicht mehr erhöht, nach 16 erfolglosen Versuchen wird der Sendeversuch abgebrochen.

14 ARP-Protokoll

Programmablaufplan des ARP Verfahrens:



15 Subnetting

Subnetting ermöglicht es Netzwerkadministratoren beispielsweise, das eigene Firmennetzwerk in Subnetze aufzuteilen, ohne dies im Internet bekannt zu machen. Das heißt, der Router, der schließlich das Netzwerk mit dem Internet verbindet, wird weiterhin als einfache Adresse angegeben.

Alle Subnetze eines Netzes funktionieren unabhängig voneinander und die Datenvermittlung läuft schneller. Warum ist das so? Subnetting macht das Netzwerk überschaubarer. Ein Broadcast, bei dem ein Teilnehmer Daten an das gesamte Netz sendet, verläuft ohne Ordnung durch Subnetze relativ unkontrolliert.

Durch Subnets werden Datenpakete durch den Router viel gezielter an die Empfänger geleitet. Befinden sich Sender und Empfänger im gleichen Subnetz, können die Informationen direkt zugestellt und müssen nicht umgeleitet werden.

15.1 Die IP Adresse

Bei dem Netzwerkprotokoll IPv4 (heute aktuell: IPv6) besteht eine IP-Adresse aus 32 Bit. Diese sind in vier Abschnitte mit je einem Oktett aufgeteilt.
Beispiel IP Adresse:

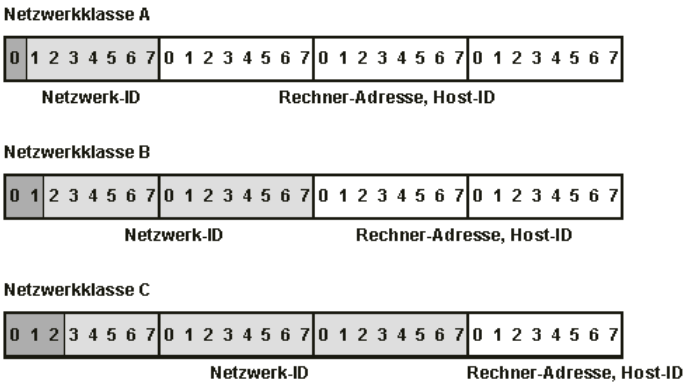
dezimal	192.168.0.1
binär	11000000.10101000.0000000.00000001

Zu beachten ist dabei, dass jedes Oktett als eigenständig bei der Berechnung der Wertigkeit angesehen wird.
Der höchste darstellbare Wert eines Oktettes beläuft sich demnach auf 255.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	46	32	16	8	4	2	1

15.2 Netzwerkklassen

Eine IP Adresse ist immer einem bestimmten System, welches sich in einem bestimmten Netz befindet zuzuordnen. Demnach besitzt eine IP Adresse einen Netzwerk und einen Hostbereich. Die IP Adresse muss dazu nicht in der Mitte geteilt sein (2 Oktette Netzwerk, 2 Oktette Hostbereich), sondern kann dazu beliebig zwischen jedem Bit geteilt werden. Netzen, mit 1 Oktett, 2 Oktetten und 3 Oktetten Netzwerkbereich wurden besondere Bezeichnungen zugewiesen:



15.3 Subnetzmaske

Um ein bestehendes Computernetz zur besseren Übersicht und Verwaltung in mehrere kleine Netze aufteilen zu können, bedarf es einer Subnetzmaske. Diese gibt den Netzwerkbereich einer IP Adresse an. Mit Hilfe der Subnetzmaske kann außerdem überprüft werden, ob sich zwei Geräte im gleichen Subnetz befinden oder nicht. Verknüpft man IP Adresse und Subnetzmaske durch eine AND Verknüpfung, erhält man die Netzwerkkennung des Subnetzes, indem sich die IP Adresse befindet. Stimmen die Netzwerkkennungen zweier Systeme überein, befinden sie sich im selben Subnetz.

15.4 CIDAR

Die CIDAR ist eine vereinfachte Schreibweise für die Subnetzmaske. Da eine Subnetzmaske dadurch definiert wird, dass sie in binär Schreibweise eine fortlaufende Kette von gesetzten Bits haben muss, die nicht durch ein nicht gesetztes Bit unterbrochen werden darf, kann man die Schreibweise dahingehend vereinfachen, dass einfach die gesetzten Bits gezählt werden:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In diesem Falle wäre die CIDAR /17

15.5 VLSM

16 Schaltnetze

16.1 Einfache Bauteile

16.1.1 NOT

16.1.2 AND

16.1.3 OR

16.1.4 NAND

16.1.5 NOR

16.1.6 XOR

16.2 Schaltnetze

Schaltnetze sind Zusammensetzungen von logischen Gattern ohne Speicherverhalten.

16.3 Wahrheitstabelle

16.4 KV-Diagramm

16.4.1 Don’t-Care-Positions

16.5 Disjunktive Normalform

16.6 Impulsdiagramm

16.7 Zustandsfolgetabelle

17 Schaltwerke

17.1 Schaltwerke

Schaltwerke besitzen Speicherverhalten, d.h. der Ausgabewert hängt nicht nur von der Eingabe, sondern auch vom Zustand des Schaltwerks ab.

17.2 Taktgeber

17.3 Flip-Flops

17.3.1 RS Flip Flop

17.3.2 T Flip Flop

17.3.3 JK Flip Flop

17.3.4 D Flip Flop

17.4 Zähler

17.5 Addierschaltungen

17.6 Subtrahierschaltungen

18 Mirkocontroller

19 Assembler

Eine Assemblersprache, kurz auch Assembler genannt, ist eine Programmiersprache, die auf den Befehlsvorrat eines bestimmten Computertyps ausgerichtet ist.

20 Objektorientierte Programmierung

20.1 Grundverständnis

Die objektorientierte Programmierung ist ein auf dem Konzept der Objektorientierung basierendes Programmierparadigma. Die Grundidee besteht darin, die Architektur einer Software an den Grundstrukturen desjenigen Bereichs der Wirklichkeit auszurichten, der die gegebene Anwendung betrifft.

20.2 Klassen

20.3 Objekte

20.4 Datentypen

20.5 Notationen

20.6 UML

20.6.1 Struktogramm

20.6.2 Objektdiagramm

20.6.3 Assoziationen

20.6.4 Polymorphie

20.6.5 Vererbungen

20.6.6 Sequenzdiagramm

20.6.7 Zustandsdiagramm

21 Datenbanken

21.1 Grundverständnis Datenbanken

21.2 Verwendung

21.3 Entity-Relationship-Diagramm

21.4 Realtionen

21.4.1 Primärschlüssel

21.4.2 Sekundärschlüssel

21.5 SQL

21.5.1 SQL Abfragen

21.5.2 SQL Selektion

22 Fachsprache

22.1 Fachwörter

Fachbegriff	Erklärung
Algorithmus	Eine genau definierte Handlungsvorschrift zur Lösung von Problemen in endlich vielen Schritten
Attribut	Eigenschaft einer Klasse
ASCII	American Standard Code for Information Interchange
ARP	Adress Resolution Protocol

22.2 Typfeld Kennungen

0x0800	IPv4 Internet Protocol, Version 4
0x0806	ARP Adress Resolution Protocol
0x0842	WoL Wake on Lan
0x8035	RARP Reverse Adress Resolution Protocol
0x809B	EtherTalk Appletalk
0x80F3	AARP Appletalk Adress Resolution Protocol
0x8100	VLAN Tag
0x8137	Novell IPX
0x8138	Novell
0x86DD	IPv6 Internet Protocol, Version 6
0x8863	PPPoE Discovery
0x8864	PPPoE Session
0x8870	Jumbo Frames
0x8892	Echtzeit-Ethernet PROFINET
0x88A2	ATA over Ethernet Coraid AoE
0x88A4	EtherCAT Echtzeit-Ethernet
0x88A8	Provider Bridging
0x88AB	Echtzeit-Ethernet POWERLINK
0x88CD	Echtzeit-Ethernet SERCOS III
0x8906	Fibre Channel over Ethernet
0x8914	FCoE Initialization Protocol FIP

23 Epilog

Diese Zusammenfassung wurde erstellt und geteilt von *Jannis Müller*. Für Falschaussagen oder Fehlinformationen übernimmt der Autor keinerlei Gewähr. Es ist empfehlenswert, sich

selbst mit den Inhalten vertraut zu machen und die hier dargestellten Inhalte gegebenenfalls zu korrigieren. Bei Fragen oder inhaltlichen Fehlern kontaktieren sie bitte umgehend den Autor und weisen auf die von Ihnen festgestellten Mängel hin.