

ETI Fragenkatalog – Antworten

Hardware

Prozessor (116): Ein Prozessor ist ein Schaltwerk zur Verarbeitung von Daten, das grob gesehen aus Rechenwerk, Steuerwerk und internem Speicher besteht. Über ein Bus-Interface greift der Prozessor auf den Systembus zu. Moderne Prozessoren haben einen zusätzlichen Speicher, einen integrierten Speichercontroller und Coprozessor-ähnliche Verarbeitungseinheiten, die in den Prozessor-Chip integriert sind.¹

Rechenwerk (117): Funktionseinheit innerhalb eines digitalen Rechensystems, die arithmetischen Operationen ausführt, und zwar durch Rückführung auf die Addition. Arithmetic logical Unit: ALU

Leitwerk (118): (auch Steuerwerk) die Funktionseinheit eines digitalen Rechensystems, die

- die Reihenfolge steuert, in der die Befehle eines Programms ausgeführt werden,
- diese Befehle entschlüsselt und modifiziert und
- die für die Ausführung benötigten digitalen Signale abgibt

Die Befehle werden interpretiert und deren Ausführung veranlasst, gesteuert und überwacht.²

Dazu existiert ein Befehlszähler, der die Adresse des nächsten zu verarbeitenden Befehls enthält, und ein Befehlsregister, das den gerade zu verarbeitenden Befehl enthält.

Der Befehlszähler wird nach jedem abgearbeiteten Befehl um die Länge eines Befehls erhöht.

Das Steuerwerk entschlüsselt (decodiert) den zu verarbeitenden Befehl unter der Fragestellung "Was soll womit getan werden?" über den Decodierer.

Außerdem gibt es für die Befehlsausführung nötigen Signale z. B. an das Rechenwerk.

Zur Steuerung der Ablaufgeschwindigkeit gibt es einen Taktgeber.³

1 <http://www.elektronik-kompendium.de/sites/com/1310171.htm>

2 <http://www.netzmafia.de/skripten/dvs/dvs2.html>

3 <http://www.petra-budde.de/fachinformatiker.php?id=2&content=Fachinformatiker&thema=Computer&tid=13>

2. Logische Schaltungen

2.1 Grundbegriffe

Übertragungskennline eines Inverters. Wo liegt die verbotene Zone? (61)

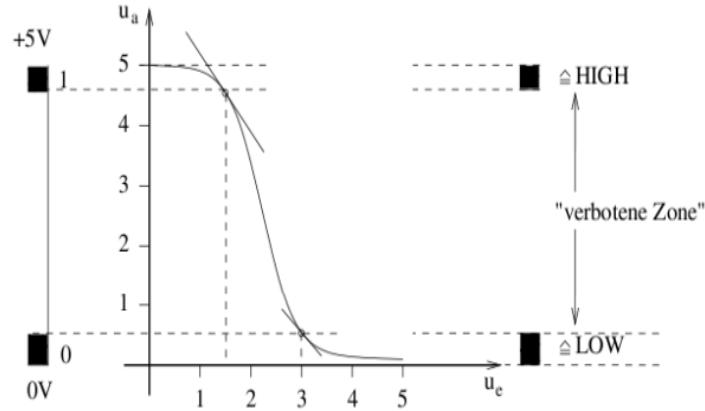


Abbildung 2.2: Übertragungskennlinie $u_a = f(u_e)$ eines Inverters

Die verbotene Zone liegt in jenem Bereich, in dem die Kennliniensteigung dem Betrag nach größer gleich 1 ist.

Bauen Sie die Grundoperationen AND, OR & NOT aus NAND Gattern auf! (138)

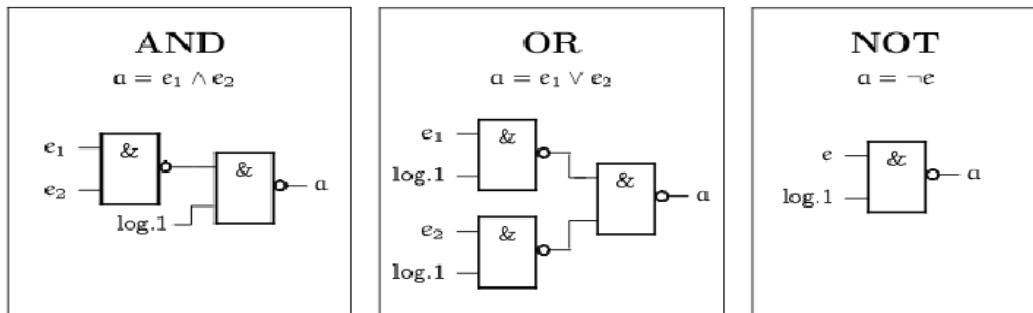


Abbildung 2.7: Boolesche Operatoren aus NAND-Gattern aufgebaut

Bauen Sie die Grundoperationen AND, OR & NOT aus NOR Gattern auf! (139)

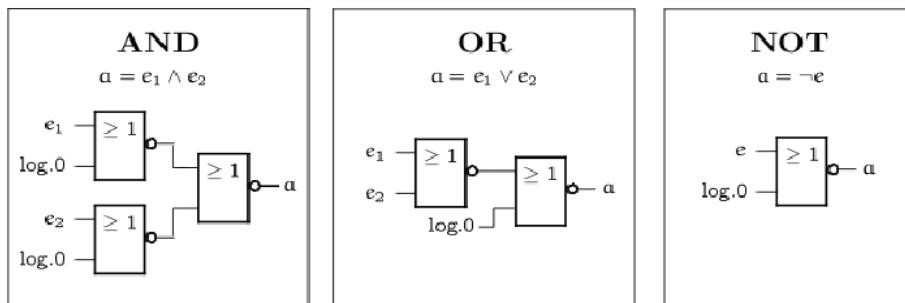


Abbildung 2.6: Boolesche Operatoren aus NOR-Gattern aufgebaut

Was sind „0-aktive Eingänge“ bei ICs (integrierten Schaltungen)? (59)

Ein 0-aktiver Eingang hat den Ruhezustand logisch 1 und wird bei logisch 0 aktiv.

Was bedeutet bei negativer Flanke triggern? Flankentriggern stellt sicher, dass der Zustandswechsel wirklich nur zu diskreten Zeitpunkten erfolgt (z.B. D-Latch). Trigger bei negativer Flanke: der Zustandswechsel erfolgt beim Wechsel von log. 1 auf log. 0 (positiv beim Wechsel von 0 auf 1).

Was versteht man unter Fan-In? (66)

Der FanIn gibt die maximale Anzahl an Gatterausgängen an, die an den jeweiligen Gattereingang angeschlossen werden dürfen.

Was versteht man unter Fan-Out? (67)

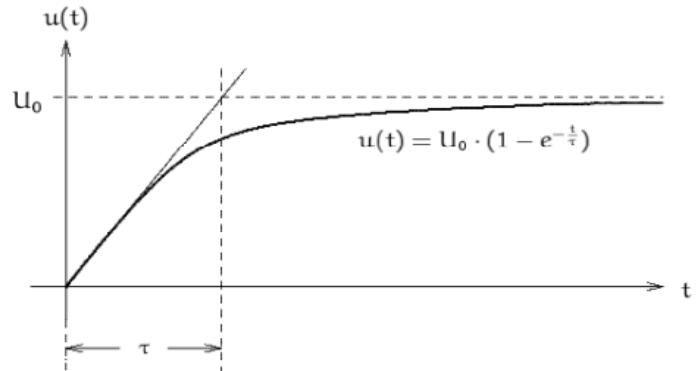
Der Fan-Out gibt die maximale Anzahl an Gattereingängen an, die an den jeweiligen Gatterausgang angeschlossen werden dürfen.

Geben Sie die Formel für die Kondensatorladung / Entladung an! (57)

Kondensatoren sind Bauelemente, die elektrische Ladungen bzw. elektrische Energie speichern können. Die einfachste Form eines Kondensators besteht aus zwei gegenüberliegenden Metallplatten. Dazwischen befindet sich ein Dielektrikum, welches keine elektrische Verbindung zwischen den Metallplatten zulässt. Das Dielektrikum ist als Isolator zu verstehen.⁴

$$\text{Aufladen: } U(t) = U_0 * (1 - e^{-\frac{t}{T}})$$

$$\text{Entladen: } U(t) = U_0 * e^{-\frac{t}{T}}$$



2.2 Realisierung von Funktionen

Anhand welcher Kriterien können Schaltkreisfamilien unterschieden werden?

- Power Dissipation: Verlustleistung eines Gatters
- Propagation Delay: Zeit zwischen Anlegen des Eingangssignals und Erreichen des eingeschwungenen Zustandes des Ausgangssignals
- Fan Out: Anzahl der Eingänge, die ohne Verlust der Funktionalität angeschlossen werden können.

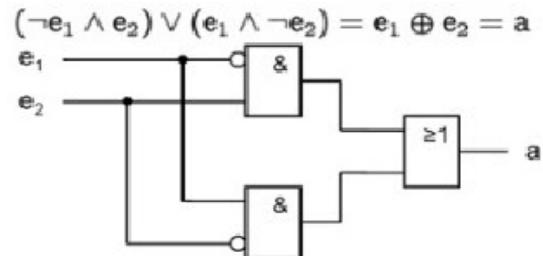
4 <http://www.elektronik-kompendium.de/sites/bau/0205141.htm>

Welche Schaltkreisfamilien kennen Sie? (127)

- TTL – Transistor-Transistor-Logik: großer Fan-Out / wegen hoher Gatterlaufzeiten (propagation delay) heute kaum noch verwendet, die Weiterentwicklung jedoch schon
- ECL – Emitter Coupled Logic: geringe Gatterlaufzeit / große Verlustleistung & hohe Kosten
- MOS – Metal Oxid Semiconductor: bei einfachen Gattern wird diese Technik selten eingesetzt / allerdings bei hoch integrierten Schaltungen
- CMOS – Complementary MOS geringste Leistungsaufnahme / großer Betriebsspannungsbereich / Standard im PC Bereich

Zeichnen Sie eine Gatterschaltung für die XOR Funktion (68)

$$\begin{aligned}
 0 \wedge 0 &= 0 \\
 0 \wedge 1 &= 1 \\
 1 \wedge 0 &= 1 \\
 1 \wedge 1 &= 0
 \end{aligned}$$



Halbaddierer (98): Addiert 2 einstellige Binärzahlen. Mögliche Überträge aus der Vorstelle werden nicht berücksichtigt. 2 Ausgänge, einer für den Übertrag.

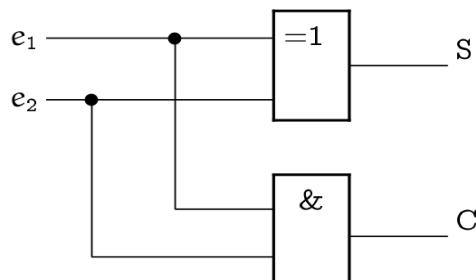


Abbildung 2.22: Halbaddierer

Volladdierer (99): Addiert Binärzahlen mit n Stellen unter Berücksichtigung des Übertrags von Vorstellen.

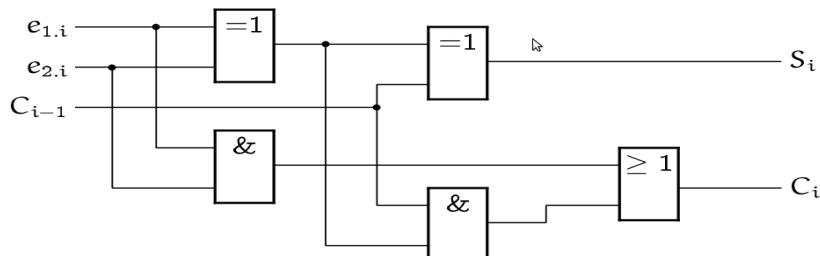


Abbildung 2.23: Volladdierer

Codierer (101): Die Schaltung eines Codierers besitzt bei n Eingängen genau $m=2^n$ Ausgänge. Es kann immer nur ein Eingang aktiv sein. Der Codierer wandelt die Bitfolge an den Eingängen in eine Binärzahl an den Ausgängen um. Das heißt: e_7 aktiv am Eingang ergibt 111 am Ausgang.

Welche Funktion hat der ENABLE Eingang bei einem Decoder? (60)

legt fest, wann das Eingangssignal decodiert werden soll und wann nicht. Hat dieser Eingang den Wert logisch 1, dann wird das Eingangssignal decodiert, ansonsten (logisch 0) liegt allen Ausgängen logisch 0 an.

Multiplexer (102)

Der Multiplexer ist eine Erweiterung des Codierers. Er wählt aus m binären Eingängen jenen aus, dessen Nummer mit der Zahl übereinstimmt, die an den Steuervariablen anliegt und schaltet dessen Information unverändert an den Ausgang durch.

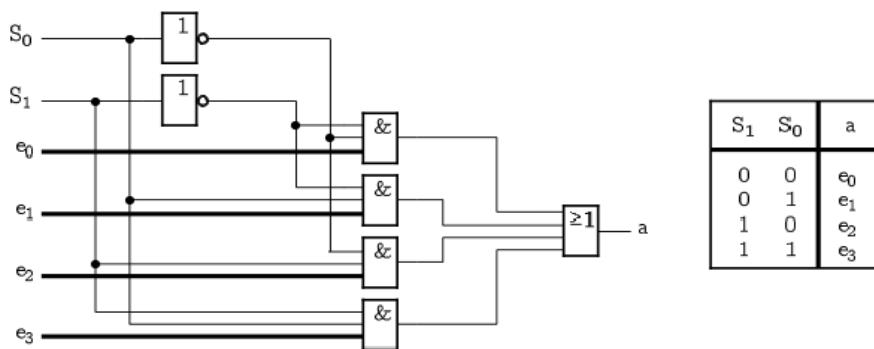


Abbildung 2.30: (4 zu 1)-Multiplexer

Welche Funktion hat der Inverter bei einem Multiplexer? (Wiki) rein aus logischer Sicht eigentlich nicht notwendig, ist vorgesehen, um zu der vorgesetzten, treibenden Stufe eine Entkopplung herzustellen – muss laufzeitmäßig entsprechend berücksichtigt werden.

Demultiplexer (103): Beim Demultiplexer wird durch die Steuereingänge bestimmt, auf welchen Ausgang die Information des Eingangs durchgeschaltet werden soll.

2.3 Sequenzielle Logik

RS Latch (100)

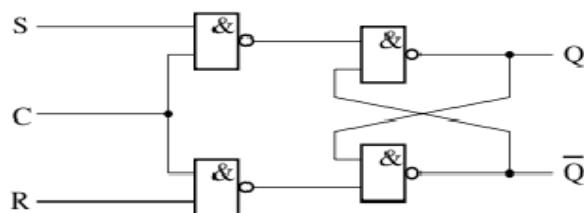


Abbildung 2.36: RS-Latch mit Kontrolleingang C

Nennen Sie die Bitfolgen der undefinierten Zustände beim Asynchronzähler, wenn er vom Zählerstand 7 auf den Zählerstand 8 übergeht. (2)

0111 => 1000

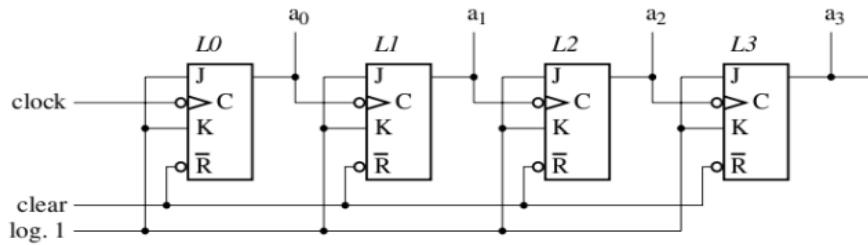


Abbildung 2.48: 4-Bit-Asynchronzähler (TTL-Realisierung)

Hierbei treten folgende Zwischenzustände auf:

0110 (L0 von 1 auf 0)
0100 (L1 von 1 auf 0)
0000 (L2 von 1 auf 0)

Alle Latches (bis aufs erste – L0) werden indirekt angesteuert (durch das davor). Deshalb ergibt sich Laufzeitbedingt eine kleine Verzögerung und es kommt zu Zwischenzuständen beim asynchronen Zähler. Wird ein falscher Wert abgegriffen, so bezeichnet man das als Hazard (=Gefahr).

Nennen Sie einen Nachteil eines asynchronen Zählers! (140)

Außer dem ersten Latch werden die nachfolgenden nur indirekt angesteuert. Dadurch nimmt der Zähler für kurze Zeit ungültige Zwischenwerte an.

In welchem Zahlensystem zählt der synchrone Zähler (141)? Im binären Zahlensystem.

Hazard (96): Fehler bei digitalen Schaltungen, die aufgrund eines undefinierten Zustands auftreten und zu falschen Werten führen. (zB Übergang von 0111 zu 1000 bei asynchronem Zähler)

Geben Sie die Schaltung eines Synchronzählers an! (3)

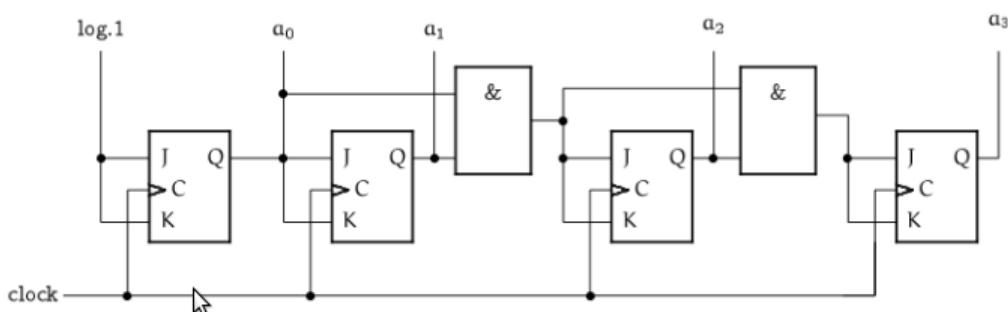
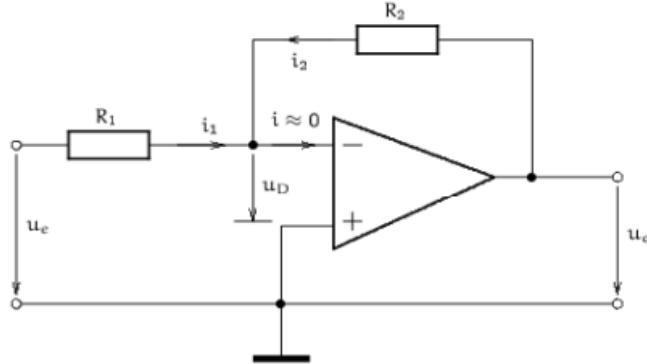


Abbildung 2.50: Vier-Bit-Synchronzähler

2.4 Signalverarbeitende elektronische Schaltungen

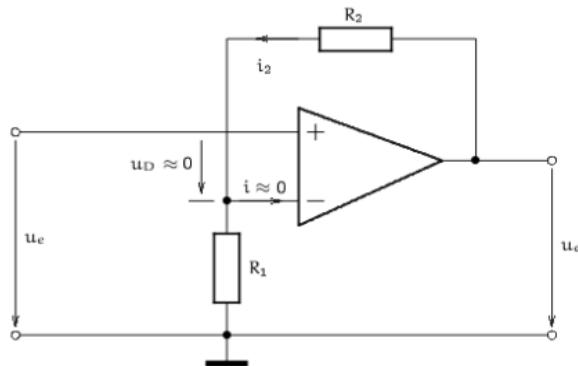
Operationsverstärker (104)

Häufig an der Schnittstelle zwischen analogen & digitalen Schaltkreisen. Charakteristisch ist eine sehr hohe Leerlaufspannungsverstärkung.



$$u_D \approx 0; \quad i_1 \approx -i_2; \quad \frac{u_e}{R_1} \approx -\frac{u_a}{R_2}; \quad v_u = \frac{u_a}{u_e} \approx -\frac{R_2}{R_1}$$

Abbildung 2.51: Invertierender Operationsverstärker (engl. *closed loop amplifier*)



$$i \approx 0; \quad u_D \approx 0; \quad \frac{u_a}{R_1 + R_2} \approx \frac{u_e}{R_1}; \quad v_u = \frac{u_a}{u_e} \approx \frac{R_1 + R_2}{R_1}$$

Abbildung 2.52: Nicht-invertierender Operationsverstärker

Komparatoren für analoge Signale (105)

Komparatoren für analoge Signale sind Schaltungen, mit deren Hilfe festgestellt werden kann, ob und zu welchem Zeitpunkt die momentane Amplitude eines Signals gleich einem konstanten oder zeitlich veränderbaren Referenzsignal ist. Das Ausgangssignal kennt nur zwei diskrete Signalzustände abhängig davon, ob die Eingangsspannung größer oder kleiner als die Referenzspannung ist.

Komparatoren für digitale Signale (106)

In der Digitaltechnik werden häufig Vergleicher für digitale Signale benötigt, um zu entscheiden, ob zwei binäre Variablen x und y gleich oder ungleich sind. Die Arbeitsweise des Komparators für digitale Signale kann durch die Äquivalenzfunktion beschrieben werden.

Torschaltungen (107)

Unter Torschaltungen versteht man Netzwerke, die aus einem Signal für eine vorgegebene Zeit einen zeitlichen Anteil des Signals „herausschneiden“. Man bezeichnet solche Netzwerke auch als Zeitfilter. Die Arbeitsweise solcher Schaltungen hängt allein von der Zeitbedingung ab, die über einen Steuereingang (control input) binär vorgegeben wird.

Was ist der Schmitt-Trigger? (Fugo)

Ein Schmitt-Trigger ist ein Komparator, bei dem Ein- und Ausschaltpegel um die Schalthysteres U_{HST} differieren. Er wird mit einem analogen Signal beliebiger Kurvenform angesteuert. Überschreitet die Eingangsspannung eine vorgegebene obere Schaltschwelle, so nimmt der Ausgang einen binären Wert a. Unterschreitet die Eingangsspannung eine untere Schaltschwelle, nimmt der Ausgang den anderen binären Wert an.

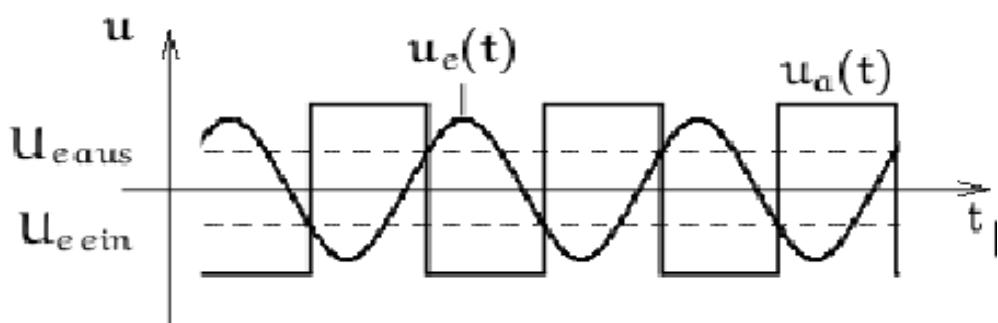
Definieren Sie die Hysteres beim Schmitt-Trigger (1)

Der Schmitt-Trigger ist ein analoger Komparator mit Mitkopplung. Er arbeitet als Vergleicher für zwei analoge Spannungen und funktioniert als Schwellwertschalter: Bei Überschreiten einer bestimmten im Schmitt-Trigger eingestellten Schwellspannung nimmt der Ausgang den Maximalwert (logisch-1) an, im anderen Fall den Minimalwert (logisch-0). Durch die Mitkopplung besitzt er im Gegensatz zum reinen Komparator jedoch geringfügig unterschiedliche Ein- und Ausschaltschwellen, die um den Hysteresenwert auseinanderliegen.

$$U_{HST} = U_{SO} - U_{SU} \rightarrow \text{Hysteresespannung} = \text{Obere Schaltschwelle} - \text{Untere Schaltschwelle}$$

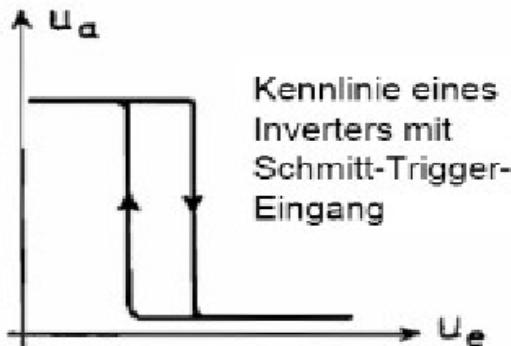
Die Hysterese beim Schmitt-Trigger ist die Differenz zwischen Ein- und Ausschaltpegel. Durch die Hysterese kann ein Schmitt-Trigger auch bei langsamem Umschaltzeiten nicht schwingen.

Geben Sie zum angegebenen Spannungsverlauf am Eingang eines invertierenden Schmitt-Triggers den Verlauf des Ausgangspegels an! (147)



Beim invertierenden Schmitt Trigger ist die „Abschaltspannung“ größer als die „Einschaltspannung“.

Zeichnen Sie die Übertragungskennlinie eines invertierenden Schmitt Triggers! (70)



Was versteht man unter dem Tastgrad g einer Rechteckimpulsfolge des Taktgenerators eines Rechners? (71)

Der Tastgrad ist das Verhältnis von Impulslänge zu Impulsperiodendauer:
$$g = \frac{T_i}{T}$$

Fensterdiskriminator (108): Mit Hilfe eines Fensterdiskriminators kann festgestellt werden, ob der Wert der Eingangsspannung U_e unterhalb, innerhalb oder oberhalb des durch die Spannungsgrenzwerte U_{GO} & U_{GU} festgelegten Fensters liegt.

Was ist ein Zero-Crossing-Detectors? (Fugo) Nullspannungsschalter, der ausgangsseitig binär umschalten, wenn die Eingangsspannung den Wert Null durchläuft. Von besonderem Interesse ist dabei, dass das binäre Ausgangssignal mit minimaler zeitlicher Verzögerung den Nulldurchgang der Eingangsspannung anzeigt, um damit Schaltvorgänge auszulösen, die an die Bedingung des Nulldurchgangs der Eingangsspannung geknüpft sind.

Erläutern Sie die Funktion der anti-parallel Diodenschaltung im Gegenkopplungszweig des OPVs eines Zero-Crossing-Detectors (5):

Zur Vermeidung des gesättigten Betriebes des Operationsverstärkers (+ dadurch bedingte Schaltverzögerung) nutzt man eine zweiseitige Begrenzerschaltung bestehend aus einer anti-parallel Diodenschaltung zwischen Ein- & Ausgang zur Begrenzung der negativen sowie positiven Ausgangsspannung

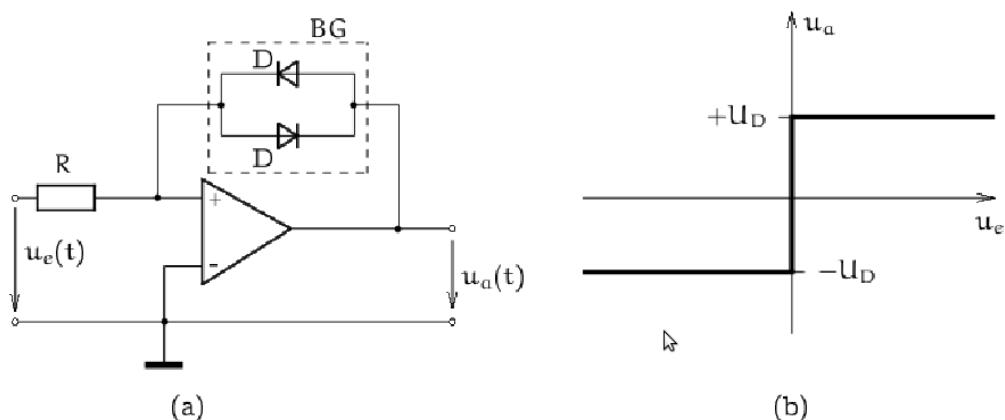


Abbildung 2.67: Nullspannungsschalter mit Operationsverstärker (BG = Begrenzerschaltung und Übertragungskennlinie $u_a = f(u_e)$)

Univibrator (109): Schaltung, die ausgangsseitig 2 Zustände annehmen kann. Von diesen beiden Zuständen ist nur ein Zustand stabil, der andere Zustand quasi-stabil und kann nur für eine vordefinierte Zeit, die Verweilzeit T_0 , nach einem eingangsseitigen Triggerimpuls angenommen werden. Es lassen sich Schaltzeiten von einigen Sekunden bis zu einigen Minuten realisieren.

Wozu dient die Ziehkapazität bei einem Quarzoszillator? (65) Abgleich der Resonanzfrequenz bei Serienresonanz und wird mit dem Quarz in Serie geschalten.

Quarzoszillator (110): eine elektronische Schaltung zum Erzeugen von Schwingungen mit einem Schwingquarz als frequenzbestimmendes Bauelement. Im engeren Sinne ist ein Quarzoszillator eine fertig aufgebaute Oszillatorschaltung, die zusammen mit dem frequenzbestimmenden Quarz in einem Gehäuse eingebaut ist und als Standardbauteil erhältlich ist. Quarzoszillatoren sind in ihrer Frequenz sehr genau.

Sägezahngenerator (111): Von den gegengekoppelten Schaltungen mit OPVs eignet sich besonders der Miller-Integrator zur Erzeugung sägezahnförmiger Spannungsverläufe. Durch periodisches Schließen des Schalters T und die dadurch entstehende Entladung der Kapazität C entsteht ein sägezahnförmiger Verlauf.

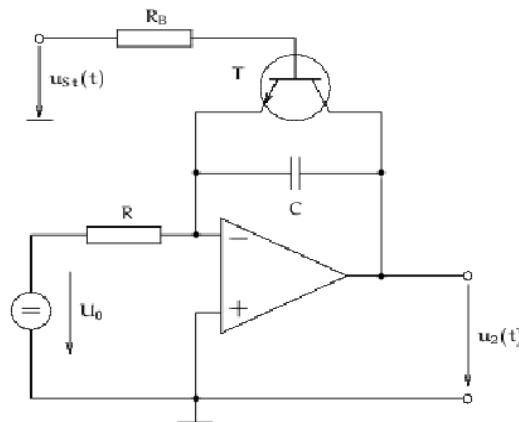


Abbildung 2.75: Sägezahngenerator mit Miller-Integrator und Schalttransistor T

2.5 Halbleiterspeicher

Tristate Output (112)

Tristate Outputs stellen abschaltbare Ausgänge dar. Das Ein- und Abschalten des Ausganges erfolgt dabei über einen eigenen Steuereingang. Somit gibt es zu den normalen Ausgangszuständen logisch 0 und logisch 1 noch den dritten Zustand mit abgeschaltetem, hochohmigen Ausgang.

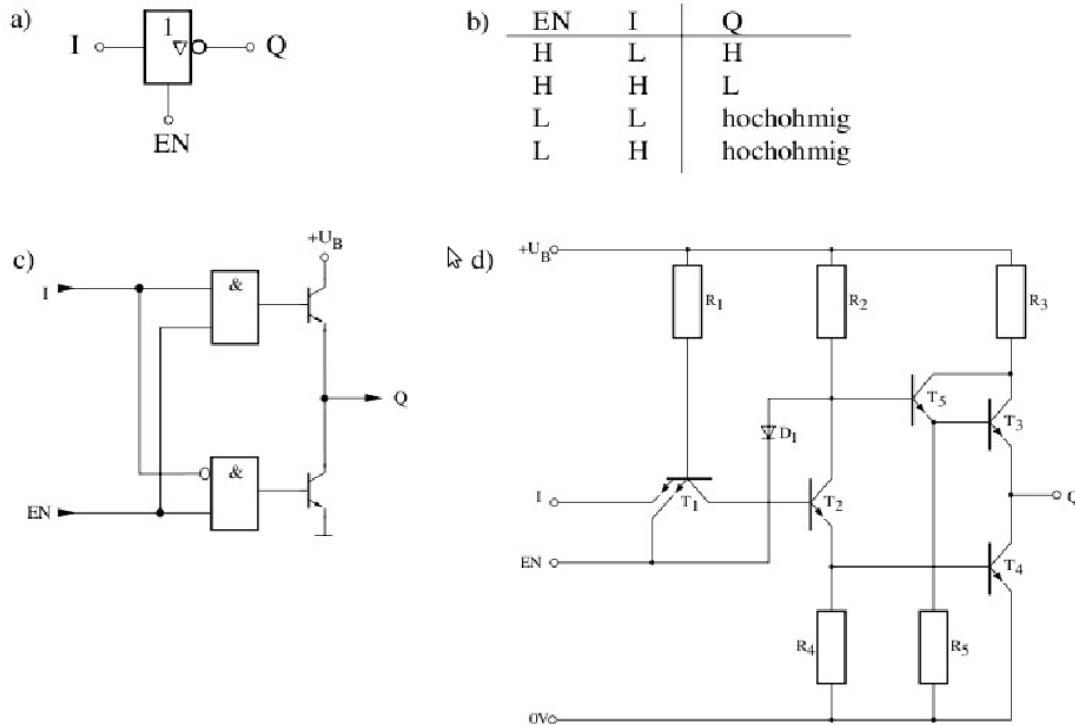


Abbildung 2.84: Tristate Outputs

Open-Collector Schaltungen (113)

Im Gegensatz zu Tristate Outputs können hier mehrere Ausgänge aktiv sein. Es werden Ausgänge mit offenem Kollektor-Ausgang eingesetzt. Diese besitzen einen NPN-Transistor am Ausgang, wobei der Emitter an Masse liegt und der Kollektorausgang unbeschaltet an den Ausgang des Gatterbausteins herausgeführt wird. Solche Ausgänge kann man nun parallel schalten und mit einem gemeinsamen Kollektowiderstand beschalten.

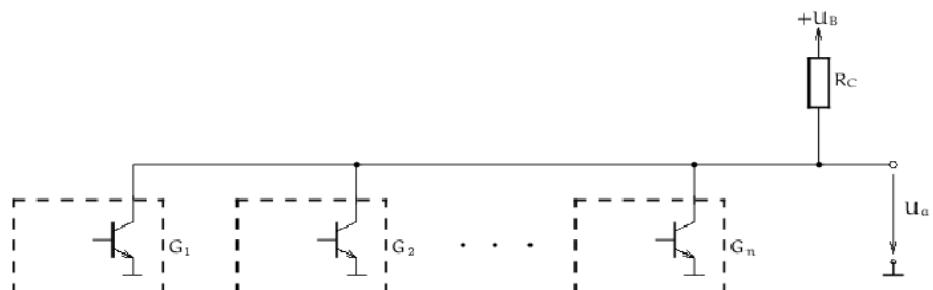


Abbildung 2.85: Zusammenschaltung von Gatterausgängen mit offenem Kollektor

Was versteht man unter dem Cycle-Stealing Verfahren? (15)

Ein mögliches Refresh-Verfahren für DRAMs. Dynamische RAMs müssen zyklisch refreshed werden, weil sie nicht mit Latches, sondern aus Kondensatoren aufgebaut sind und daher mit einer zusätzlichen Schaltung die Daten aufrecht erhalten werden müssen. Damit diese nicht zu lange blockiert werden, führt man beim Cycle Stealing Verfahren die Refreshvorgänge für einzelne Teile des Speichers getrennt durch.

oder: Buszuteilung bei DMA's Bei einem DMA stiehlt der Controller dem Prozessor den Maschinencyklus, indem er den Bus anfordert, sobald das Gerät seine Bereitschaft zum Datentransfer meldet. Der Prozessor kann erst wieder auf den Speicher zugreifen, wenn die Aktion beendet ist.

Welche Refresh Verfahren gibt es bei Speicherzellen? (56)

- **Burst Refresh** : Der Normalbetrieb wird unterbrochen um bei allen Speicherzellen ein Refresh durchzuführen. Während dieser Zeit ist kein Zugriff auf den Speicher möglich.
- **Transparent Refresh** : Refreshcontroller mit dem Prozessor synchronisiert um die laufenden Prozesse nicht anhalten zu müssen.
- **Cycle Stealing** : Um lange Blockierungen zu vermeiden, kann man die Refreshvorgänge für die einzelnen Teile des Speichers getrennt durchführen.

Welche Arten von Speicherbausteinen kennen Sie? (58)

- **Statische RAM** : Informationen wird in Latches gespeichert. Sehr kurze Zugriffszeiten. Hoher Preis. Kaum hoch integrierbar.
- **Dynamische RAM** : Informationen wird in Kondensatoren gespeichert. Refresh Cycle notwendig.

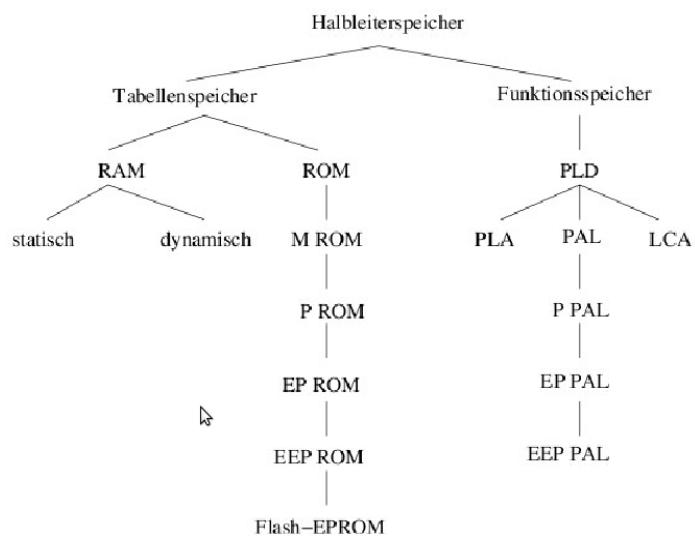
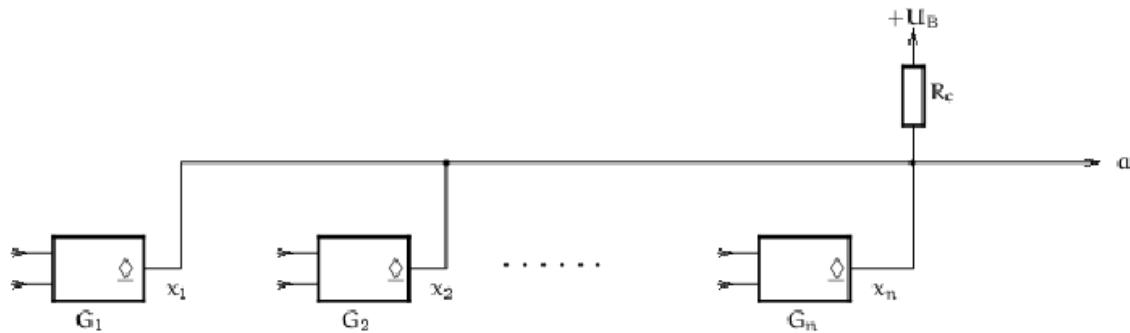


Abbildung 2.88: Übersicht über Halbleiterspeicher

Was ist Interleaved Memory und wozu wird er eingesetzt? Sequentielle direkt hintereinander liegende Befehle können schneller erreicht werden, da Adressen, die in verschiedenen Speicherblöcken liegen, schneller auf den Adressbus gelegt werden können, als es sonst – bedingt durch Zugriffszeiten der Bausteine – möglich wäre. Die Daten kommen dadurch ebenfalls schneller beim Prozessor an.

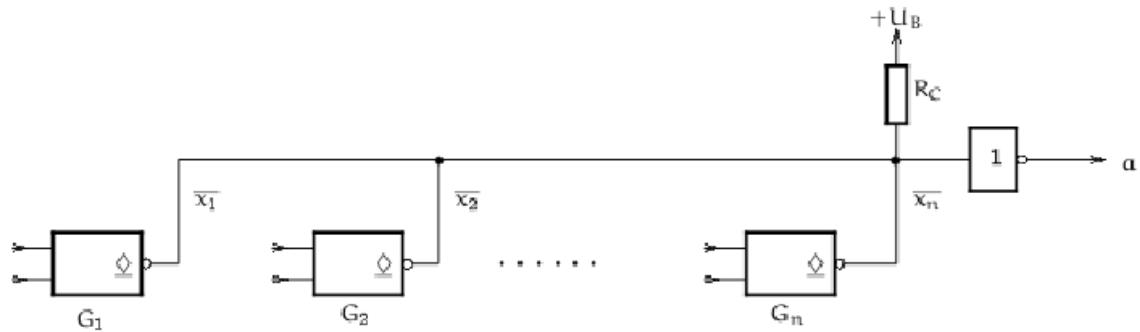
Bestimmen Sie die Funktion der folgenden Schaltungen. (137)



$$a = x_1 \wedge x_2 \wedge \dots \wedge x_n$$

Abbildung 2.86: Wired-AND-Verknüpfung (\diamond =Symbol für Open-Collector-Ausgang)

→ WIRED AND



$$a = x_1 \vee x_2 \vee \dots \vee x_n$$

Abbildung 2.87: Wired-OR-Verknüpfung (\diamond =Symbol für Open-Collector-Ausgang)

→ WIRED OR

Erklären Sie die Abkürzung ASIC! (69)

ASIC steht für Application Specific Integrated Circuit. Es handelt sich dabei um einen Funktionsspeicher, also einen integrierten Baustein, der zur Speicherung einer Funktion dient.

EEPROM: Nachteil im Vergleich zum RAM (134)

Beim EEPROM dauern die Zugriffszeiten länger und die Speicherstellen sind nur begrenzt beschreibbar (maximal 10^6 mal). RAMs sind permanent lösbar & wiederbeschreibbar. Die Nichtflüchtigkeit des EEPROMs stellt wiederum einen Vorteil gegenüber den RAMs dar.

3. VHDL

Was versteht man unter VHDL? (125)

Die Very (High Speed Integrated Circuit) Hardware Description Language ist eine Sprache für Hardware-design, besonders für komplexe Schaltwerke.

Welche Entwurfsebenen gibt es in VHDL? (126)

Dokumentation des Gesamtsystems:

- Systemebene
- Algorithmische Ebene

Simulation:

- Register-Transfer-Ebene
- Logik-Ebene

Modellierung des Systemverhaltens mit Differenzialgleichungen:

- Schaltkreis-Ebene

Welche 3 Sichtweisen gibt es beim Entwurf elektronischer Schaltungen laut dem Y--Modell? (62)

- Geometrie
- Struktur
- Verhalten

Welche 3 Arbeitsschritte sind zur Erstellung eines Moduls bzw. einer Komponente in VHDL erforderlich? (63)

- Schnittstellenbeschreibung
- Architektur
- Konfiguration

Wozu dient die Architektur in VHDL? (153)

Die Architektur beschreibt die Funktionalität des Moduls entweder als Verhaltensbeschreibung oder als Netzliste.

4 Mikroprozessoren

Was ist ein endlicher deterministischer Automat? (38)

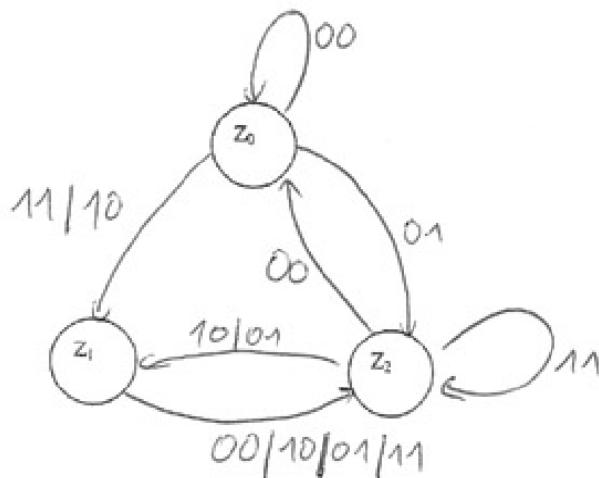
endlich: Die Anzahl der Zustände des Automaten sind begrenzt.

deterministisch: aus der Eingangsinformation und dem Vorzustand lässt sich stets eindeutig vorherbestimmen, in welchen Zustand der Automat wechseln wird.

Zeichnen Sie ein Zustandsdiagramm das folgender Tabelle entspricht.

A	0	1	0	1
B	0	0	1	1
Z_0	Z_0	Z_1	Z_2	Z_1
Z_1	Z_2	Z_2	Z_2	Z_2
Z_2	Z_0	Z_1	Z_1	Z_2

↳



Schaltwerk vs. Schaltnetz

Schaltnetz: logische Funktionen ohne Speicherwirkung. Die Ausgänge hängen immer nur von den momentan anliegenden Eingangssignalen ab.

Schaltwerk: enthält einen Zustandsspeicher, der aktuellen Zustand festhält. Ausgangssignale sind nicht nur vom Eingang, sondern auch vom momentanen Zustand abhängig, wobei der Übergang von einem Zustand zum nächsten vom Eingangssignal gesteuert wird.

- **Asynchrone Schaltwerke:** ändert seinen Zustand praktisch sofort bei Eingangsänderungen – allerdings oft nicht praktikabel
- **Synchrone Schaltwerke:** arbeiten mit Takt signal, das die Speicherelemente des Schaltwerks so steuert, dass sich der in den Latches gespeicherte Zustand nur zu gewissen Taktzeitpunkten ändert kann.

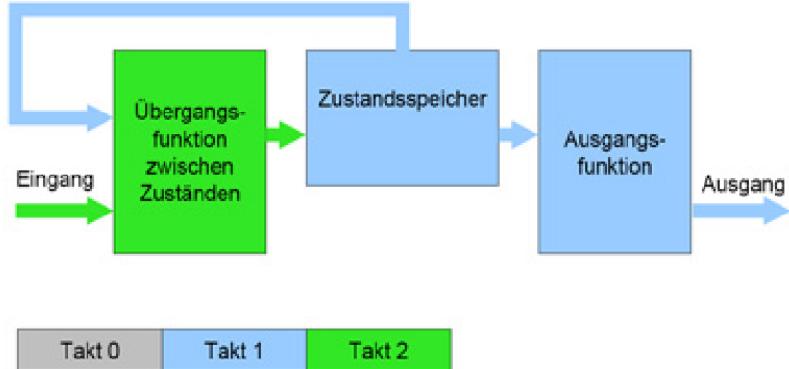
Von welchen beiden Faktoren hängen Zustandsänderungen bei Schaltwerken ab? (64): von den Eingangswerten und dem aktuellen Zustand des Schaltwerkes

Moore Schaltwerk (114)

Beim Moore Schaltwerk können die Zustände in beliebiger Reihenfolge aufeinanderfolgen. Ein Zustand kann mehrere Folgezustände haben. Die Eingangssignale entscheiden, welcher dieser Folgezustände tatsächlich eingenommen wird. Beim Moore Schaltwerk sind die Ausgänge im einfachsten Fall mit den Latch Ausgängen identisch. Meist bildet aber eine Ausgangsfunktion die eigentlichen Ausgänge aus den Latch Ausgängen, um damit Latches einzusparen.

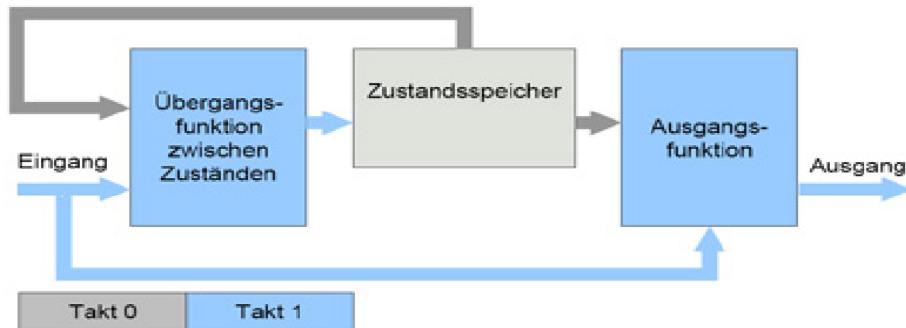
Beim Moore Schaltwerk hängen die Ausgänge nur vom momentanen Zustand ab. Damit können die Eingänge die Ausgänge nicht direkt beeinflussen. Für Ausgangsänderungen ist ein Zustandswechsel notwendig, der immer erst bei der nächsten Taktflanke erfolgen kann.

Diese Einschränkung kann auch vorteilhaft sein, weil sie einen zeitlich sauberen Verlauf mit sich bringt. Die Ausgänge sind mit dem Takt synchronisiert und ändern sich exakt zum Zeitpunkt der schaltenden Taktflanke und behalten ihren Wert zwischen zwei Taktflanken verlässlich bei.



Mealy Schaltwerk (115)

Das Mealy Schaltwerk unterscheidet sich vom Moore Schaltwerk nur durch eine andere Ausgangsfunktion. Damit lassen sich Ausgangsreaktionen erreichen, die mit einem Moore Schaltwerk nicht möglich sind. Beim Mealy Schaltwerk hängen die Ausgänge nicht nur vom momentanen Zustand (Ausgänge der D-Latches) ab, sondern auch von den Eingängen. Für jeden Zustand sind – gesteuert von den Eingängen – verschiedene Ausgänge möglich.



Auch direkter Durchgriff vom Eingang auf Ausgang mit sofortiger Reaktion möglich. Typisch für Mealy im Vergleich zu Moore: weniger Zustände und Taktzyklen nötig.

Bei synchronisierten Eingangssignalen ändern sich auch bei Mealy die Ausgänge nur synchron mit dem Takt. Da hier aber die Ausgangsfunktion meist wesentlich komplexer ist, kann es beim Mealy-Schaltwerk eher zu Hazards und damit zu kurzzeitigen Störungen an den Ausgängen kommen.

4.4 Prozessoren

Die ALU generiert 2 Signale N & Z. Geben Sie an, was die Signale N & Z bedeuten. (4)

2 Steuerleitungen, die auch für die Durchführung von Sprungbefehlen herangezogen werden.

- N=1 wenn das ALU Ergebnis < 0 (Negativbit – Ergebnis muss als negative Zahl interpretiert werden)
- Z=1 wenn das ALU Ergebnis = 0 (Nullbit-Zero)

Was versteht man unter dem Scratchpad eines Prozessors? (72)

Das Scratchpad (oder Register File) eines Prozessors ist eine Konfiguration, bei der die ALU (Arithmetic Logic Unit) die Register A & B nur mehr als Zwischenspeicher verwendet.

Welchen Vorteil hat die Erhöhung der Baudichte bei Mikroprozessoren? (73)

Reduzierung der Wegstrecken → Erhöhung der Prozessorgeschwindigkeit, da sich elektrischer Strom mit etwa 0,7-facher Lichtgeschwindigkeit ausbreitet.

Erklären Sie die Abkürzung VLSI! (128)

Bei der Very Large Scale Integration werden nach dem Chipdesign alle Bauteile und Verbindungsleitungen gleichzeitig auf einem Siliziumplättchen hergestellt.

Was versteht man unter dem Program Counter bei einem Mikroprozessor? (84)

Der PC ist ein Prozessorregister, welches die Speicheradresse speichert, an welcher der auszuführende Maschinencode gespeichert ist.

Mikroprozessor 16 : immediate addressing mode . Der im Buch beschriebene Mikroprozessor Mikro16 enthält leider keinen eigenen Befehl, um einen beliebigen numerischen Wert (das heißt, ein immediate value) in ein Register zu laden. Geben Sie eine Sequenz von Instruktionen an für den Mikro16 an, der den Wert 17 in das Register R11 lädt. Sie können dazu beliebige, andere Register überschreiben. Bedenken Sie jedoch, dass die Register R0, R1 und R2 nicht überschrieben werden können, weil sie die Konstanten 0, 1 bzw. -1 enthalten. (132)

```
1. R3 <- lsh(1+1);      # 1+1=(10)2; lsh((10)2)=(100)2=(4)10
2. R3 <- lsh(R3);      # lsh((100)2)=(1000)2=(8)10
3. R3 <- lsh(R3) ;    # lsh((1000)2)=(10000)2=(16)10
4. R4 <- 1;              # 16+1=17
5. R11 <- R3+R4;      # 16+1=17
```

5. Computersysteme

2 Möglichkeiten, Adressen der Ports für Input/Output Operationen zu vergeben?

- Independent I/O System: Hauptspeicher und Ports haben voneinander unabhängige Adressen
- Memory Mapped I/O System – Ports werden so behandelt, als wären sie gewöhnliche Speicherstellen.

Was versteht man unter dem Program Status Word?

Bei der Ausführung von arithmetischen Operationen können Fehler wie zB ein Überlauf bei der Addition zweier großer Zahlen auftreten. Um dies zu dokumentieren und im weiteren Programmablauf nutzen zu können, besitzen die meisten Prozessoren ein PSW. Die Bits des PSW zeigen beispielsweise einen Überlauf oder Übertrag an und werden nur durch das Ergebnis einer Operation gesetzt oder gelöscht.

Wie ist mit dem Vorzeichen-Bit bei einem arithmetischen Rechtsshift zu verfahren?

Das Vorzeichenbit (MSB) wird einfach wieder eingefügt und so das Vorzeichen erhalten.

1100 RECHTS-SHIFT = 1110 oder
0100 RECHTS-SHIFT = 0010

Was passiert mit dem Vorzeichenbit bei einem arithmetischen Linksshift?

Das VZ Bit wird nicht verändert – Das Bit vor dem MSB geht verloren.

Was versteht man unter Flow Control?

Flow Control Operationen sind Maschinenbefehle, die es ermöglichen, den sequentiellen Ablauf eines Programmes zu unterbrechen. (Sprünge, Schleifen, etc)

Sprünge (bedingte und unbedingte): Sprünge können sowohl auf Register-Transfer-Ebene als auch auf Maschinen- Code-Ebene erfolgen. Bei dieser Operation wird der MIC (bzw. Programm- Counter) mit einem neuen Wert geladen. Bei bedingten Sprüngen kann zusätzlich eine Bedingung angegeben werden, die erfüllt sein muss, damit ein Sprung erfolgen kann.

Was sind Subroutine Calls?

Programme bestehen aus Unterprogrammen, auch Subroutinen genannt. Diese können von verschiedenen Stellen aufgerufen werden. Nach einem „return from subroutine“ wird im Programm fortgesetzt. Es sind auch geschachtelte Aufrufe möglich. Die beiden Subroutine Befehle führen stets folgende Kommandos aus:

CALL SUBROUTINE

1. Retten des PSW, der Registerinhalte und des Program Counters (PC)
2. 2. Laden des PC mit der Prozedur Startadresse
3. Abarbeiten der Prozedur im Interpreter

RETURN FROM SUBROUTINE

1. Wiederherstellen des PSW & der Registerinhalte
2. Laden des PC mit dem Wert, der beim CALL SUBROUTINE gesichert wurde.
3. Fortsetzen des Programmablauf

Was sind Interrupts?

Dieses Konzept ermöglicht die Bewältigung von diversen Problemen, die nicht im Programmablauf vorgesehen sind. Interrupts unterbrechen den herkömmlichen Programmablauf & springen in bestimmte – als ISR (Interrupt Service Routine) bezeichnete Service Routinen. Danach kann meist das unterbrochene Programm fortgesetzt werden. Eine ISR ist ähnlich einer Subroutine. Auslöser ist ein prozessorientierter Ausnahmefall (zB Division durch 0) oder ein externes Ereignis.

Wo werden die Return Adressen bei Unterprozeduraufrufen gespeichert?

Die Speicherung der Return-Adressen erfolgt auf einem Stack.

Welche 2 Möglichkeiten gibt es, die Adresse einer Interrupt Service Routine (ISR) zu bestimmen?

Fixe Zuordnung und Interruptvektor

Nennen Sie die Adressierungsarten die Sie kennen!

- Implied Mode – implizite Adressierung (zB enable interrupts). Keine Adressierungsart im eigentlichen Sinn. Die Gruppen von Befehlen, die dieses Verfahren verwendet, ist nur für einen bestimmten Operanden definiert, darum muss dieser nicht eigens in einem Adressfeld angegeben werden.
- Register Mode – ($R3 \leftarrow R4 + R5$). Die Register werden über die ihnen zugeordneten Adressen angesprochen, wobei diese auch in der Maschinen-Code-Schicht verwendet werden können. Am Mode-Field erkennt der Interpreter, dass es sich bei der Angabe nicht um eine Adresse im Hauptspeicher, sondern um die eines Registers handelt.
- Immediate Mode – ($R \leftarrow 0$): Keine Adressierung im eigentlichen Sinn, weil der Wert des Operanden direkt im Maschinenbefehl enthalten ist. Operandenfeld enthält Konstante statt einer Adresse. Verwendung zur Initialisierung von Registern oder Inkrementieren eines Schleifenzählers.
- Direct Addressing Mode – ($R6 \leftarrow \text{memory}[(500)16]$). Der Wert im Operandenfeld entspricht der Speicheradresse des gewünschten Operanden. Verwendung bei Transferoperationen und Sprungbefehlen.
- Register Indirect Mode ($R6 \leftarrow \text{memory}[R5]$). Das im Maschinen-Befehl als Operand angegebene Register, enthält die effektive Adresse des gewünschten Datenwortes.
- Program Counter Relative Addressing Mode. Eine Variante der Base- bzw. Indexed-Adressierung. Dabei berechnet sich die effektive Adresse durch die Addition eines im Befehl angegebenen Offsets zum aktuellen Programmzählerstand. Das gestattet das Erstellen von Programmen, die an einer beliebigen Stelle des Arbeitsspeichers lauffähig sind.
- Indirect Addressing Mode ($R6 \leftarrow \text{memory}[\text{memory}[(500)16]]$). 2-stufige Speicher-Adressierung. Erfordert mehrere sequenziell aufeinander folgende Adressberechnungen. Das Ergebnis der 1. Berechnung liefert die Adresse eines Speicherwortes, das entweder Adresse oder Offset der folgenden Berechnung enthält.

Erläutern Sie die direkte, indirekte, register-indirekte, program counter relative Adressierung!

- direkt: Der Wert im Operandenfeld entspricht der Speicheradresse des gewünschten Operanden.
- indirekt: Im Maschinenbefehl ist eine Speicheradresse enthalten, an der die Adresse des Datums steht. $\text{memory}[\text{memory}[\text{adress}]]$
- register-indirekt: Das Register, das im Befehl angegeben wird, enthält die Adresse des gewünschten Datenwortes. Register fungiert als Pointer
- program-counter-relative: Die effektive Adresse erhält man durch Addition des im Befehl angegebenen Offsets (Displacement, positiv oder negativ) mit dem Programmzählerstand. So ein Programm an beliebigen Stellen im Arbeitsspeicher arbeiten.

Erklären Sie die RISC Architecture!

Reduced Instruction Set Computer – das Arbeiten mit reduziertem Befehlssatz. Ein RISC Befehlssatz verzichtet – zugunsten eines niedrigeren Dekodierungsaufwandes auf Seiten der CPU – konsequent auf komplexe Befehle, wodurch die Ausführung einfacher als bei einem CISC (Complex Instruction Set Computer) Befehlssatz ist. Zusätzlich wird auf optimale Implementierung der wenigen Befehle geachtet, was zu Performancesteigerung führt.

Welche Weiterentwicklung des RISC Architektur sind im Vorlesungsbuch beschrieben?

1. SPARC Architektur (Scalable Processor Architecture) Konventioneller RISC Prozessor mit ausgefeilter Registerarchitektur. Offene Architektur, kann frei verwendet werden.
2. MIPS Architektur (Microprocessor without Interlocking Stages) Performancesteigerung durch feinstufige Befehlspipeline und realisierte Speicherhierarchie

Geben Sie einige Quellen für Probleme an, die beim Instruction Pipelining auftreten.

1. Sprünge
2. Subroutine Calls
3. Interfering instructions (Ergebnis der n-ten Operation wird als Operand für die (n+1)te benötigt.

Welche Arten von PipelineHazards gibt es?

- Strukturelle Hazards: Mehrere Pipeline Stufen benötigen die selbe Ressource. Maßnahmen: Geeignete Architektur
- Control Hazards: Nachfolgebefehl hängt vom Ausgang des Sprungs ab. Maßnahmen: prediction, delayed branch
- Data Hazards: Berechnung erfordert Ergebnis des Vorgängerbefehls. Maßnahmen: forwarding, Code-Optimierung

Geben Sie 4 Pipeline Sprungstrategien an (bei den Control Hazards)

- **Interlocking** : Pipeline Mechanismus wird gestoppt bis Zieladresse des Sprunges ermittelt wurde
- **Delayed Branch** : Sprungausführung erfolgt scheinbar verzögert, da die Anweisung nach dem Sprung noch ausgeführt wird.
- **Prediction Branch** : Hier wird versucht die Sprungadresse vorrauszusagen (zB bei Schleifen immer wieder die selben Adressen)
- **Branch History** : Hier wird zusätzlich noch ein Sprungziel Cache verwendet.

Was versteht man unter Interlocking beim Pipelining Konzept?

Der Pipeline Mechanismus wird gestoppt, sobald die Decoding Unit einen Sprungbefehl erkennt. Die Freigabe erfolgt wieder, wenn die Zieladresse ermittelt wurde oder der Program Counter erneuert wurde. Verbesserte Methoden bzw andere Methoden:

- Delayed Branch – Anweisung nach Sprungbefehl noch ausführen (nur wenn Sprungbedingung nicht beeinflusst)
- Predicted Branch – Ob ein Sprung erfolgt wird versucht zu prognostizieren.
- Branch History – Sprungziel Cache, Zieladressen merke

Warum ist eine hohe Trefferquote der branch prediction Einheit bei modernen Prozessoren so wichtig? Denken Sie daran, dass die Prozessoren eine sehr lange Pipeline verwenden.

Normalerweise werden der Pipeline die Daten sequentiell, wie sie kommen, zugeführt. Wird aber ein Sprung ausgeführt, sind alle Daten, die nach diesem Sprung in die Pipeline geraten sind, wertlos und müssen verworfen werden. Ziel der branch prediction ist ein möglichst frühes Erkennen eines Sprungbefehls und Erkennen seiner Sprungzieladresse, damit gleich die Daten der Zieladresse dem Sprungbefehl in die Pipeline folgen können.

Welchen Vorteil besitzt die „von Neumann“ und die „Harvard-Architektur“. Geben Sie auch eine schematische Darstellung dieser beiden Architekturen an.

Von Neumannscher Flaschenhals: Daten und Programme sind sich im Hauptspeicher

- **Harvard**: Programme und Daten in getrennten Speichern und mit getrennten Bussen übertragen: paralleler Instruction & Operand Fetch möglich
- **Von Neumann**: Programme & Daten werden in demselben Speicher gehalten und über einen einzigen Datenbus übertragen.

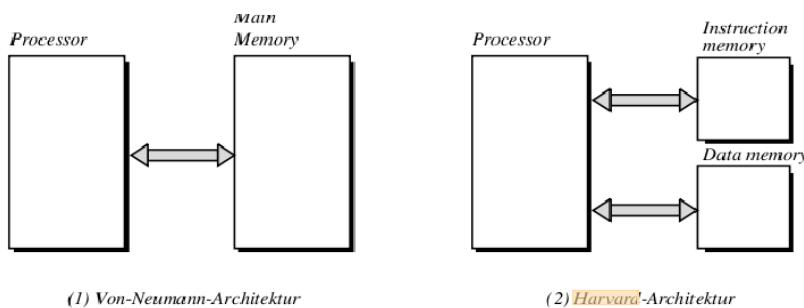


Abbildung 5.6: Harvard-Architektur

Was ist Interleaved Memory und wozu wird er eingesetzt?

Sequentielle Zugriffe auf den Speicher können schneller bedient werden, wenn die Adressen auf die einzelnen Speicherbänke aufgeteilt werden. Z.B. alle geraden Adressen in der ersten Bank und die ungeraden in der zweiten Speicherbank. (2-fach interleaved, auch vierfach)

Es wird hier das PipeliningPrinzip auf Speicherzugriffe übertragen – die Daten kommen schneller beim Prozessor an, bzw. können schneller gespeichert werden.

Blöd nur, wenn hintereinander mehrmals auf denselben Speicherbaustein zugegriffen werden muss (entspricht keinem sequentiellen Zugriff).

Woraus ergibt sich die Performance eines Caches?

Hit Time, Hit Rate, Miss Penalty

Was versteht man unter einem Cache Speicher? Wie ist er aufgebaut?

sehr schneller, nahe am Prozessor liegender Speicher, der sehr teuer und klein ist. Er dient als Zwischenspeicher für häufig benötigte Daten. Ein Cache ist transparent. Sind die angeforderten Daten im Cache (Cache Hit), so können sie sehr schnell zum Prozessor geliefert werden. Sind die Daten nicht im Cache (Cache Miss), werden sie aus dem Hauptspeicher nachgeladen. Die Cache Hit Rate gibt Auskunft über den Erfolg der Zugriffe. Der Cache besteht aus mehreren Speicherzellen (Cache lines)

Welche Arten von Caches kennen Sie?

- **Voll assoziativer Cache:** Daten müssen erst ersetzt werden, wenn der Cache voll ist. Suche nach vorhandenen Daten dauert länger.
- **Direct Mapping:** Jede Adresse wird auf eine bestimmte Cache Line abgebildet. Kommt ein neuer Datenwert, muss der alte Wert an dieser Adresse ersetzt werden. Datensuche sehr schnell

Geben Sie als Abschätzung in Form einer Gleichung die durchschnittliche Speicher-Zugriffszeit t_{eff} für ein Rechnersystem an, das sowohl einen Cache Speicher (Zugriffszeit t_{cache}) als auch einen Hauptspeicher (t_{main}) besitzt. Die Trefferrate für Daten im Cachespeicher soll mit h angegeben werden.

Formel: $t_{eff} = h * t_{cache} + (1-h) * t_{main}$

Nennen Sie die 3 Replacement Strategien (Cache)!

- LRU: Least Recently Used (Zellen, deren Aufruf am längsten zurückliegt)
- LFU: Least Frequently Used (Zellen, die am seltensten in letzter Zeit benutzt wurden)
- RANDOM: Es wird einer zufällig ausgewählt

Erfahrungen und Messungen zeigen, dass alle 3 Methoden ungefähr gleich gut sind, wenn die Kapazität des Caches nicht zu klein ist.

Erklären Sie die Buffered-Write-Methode im Zusammenhang mit Caches! Welche Probleme können dabei auftreten?

Verbindet die Vorteile des Copy-Back-Verfahrens (schnelle Schreiboperationen) und des Write-Through-Verfahrens (Datenkohärenz – Daten im Cache & im Hauptspeicher zu jedem Zeitpunkt identisch). Der neue Wert wird direkt in den Cache und in einen zweiten schnellen Zwischenspeicher (Pufferspeicher) geladen. Während der Prozessor mit der Abarbeitung des Programms fortsetzt, werden die gepufferten Daten in den Hauptspeicher übertragen (=> Datenkohärenz).

Ein Problem tritt auf, wenn mehrere Schreiboperationen direkt aufeinander folgen und der Puffer daher nicht schnell genug in den Hauptspeicher übertragen werden kann. In diesem Fall muss der Prozessor warten. Datenkohärenz muss besonders dann erfüllt sein, wenn mehrere Prozessoren mit lokalen Caches einen gemeinsamen Hauptspeicher benutzen.

DMA: Was es ist und warum es verwendet wird?

Direct Memory Access ermöglicht, die Kommunikation zwischen Prozessor und meist sehr viel langsameren peripheren Geräten zu beschleunigen. Er dient zur direkten Übertragung großer Datenmengen vom/zum Speicher, ohne die CPU in Anspruch zu nehmen. Um Konflikte zu vermeiden, darf die CPU während des DMA nicht auf den Bus zugreifen. Somit ist nicht sichergestellt, dass der Prozessor in der Zeitspanne, die zum Transfer nötig ist, Aufgaben durchführt, die ohne externen Buszugriff möglich sind. Erforder eine zusätzliche Kontrolleinheit – der DMA Controller.

Vorgang des DMA erklären

- 1) Der Prozessor teilt dem DMAC
 - die Adresse der Quelle (resource pointer),
 - die des Ziels (destination pointer) und
 - die Größe der zu übertragenden Daten (block length) mit.Hierauf kann die CPU mit der Abarbeitung des Programms fortfahren.
- 2) Der DMAC fordert nun vom entsprechenden Gerät die Daten an und wartet, bis es zum Transfer bereit ist.
- 3) Nach dem Ende der Übertragung meldet der DMAC dem Prozessor den erfolgreichen Abschluss der Aktion meistens per Interrupt-Signal. Die Daten werden also direkt zwischen I/O Device & Speicher ausgetauscht.

Durch welche drei im Buch erläuterten Arten der Parallelverarbeitung kann die Performance eines Prozessors gesteigert werden?

- Vektorverarbeitung
- Superskalare Verarbeitung
- Instruction-Pipelining

Arten von DMA (2)

- Transparente DMA (beobachten Busse und transferieren während der Prozessor den Bus nicht benötigt)
- Cycle Stealing DMA (Bus wird dem Prozessor vor der Nase weggestohlen)

Controller & Co-Prozessoren

Controller – sind Prozessoren, die spezielle Aufgaben erfüllen, um den Prozessor zu entlasten. (zB DMA Controller. Co-Prozessoren – ebenfalls um die CPU zu entlasten – verschiedene Spezialaufgaben – meist auf dem Prozessorchip direkt integriert. (MathematikCo-, Graphik-Co-Prozessor)

Wie unterscheidet sich bei USB die Connect Erkennung von Full und Low Speed Geräten?

Es gibt 2 Datenleitungen D+ und D- an den Downstream Ports, die an den Ports der Hubs mit Widerständen von jeweils 15 kOhm mit der Masse verbunden sind. Dagegen ist beim USB-Gerät auf der Upstream-Seite eine Leitung über einen 15 kOhm Widerstand mit +3,3V verbunden. Bei Full-Speed Geräten ist das die D+ Leitung, bei Low-Speed Geräten die D- Leitung. Der Hub überwacht am Downstream Port beiden Datenleitungen D+ und D-.

- Kein Widerstand angeschlossen: die beiden 15 kOhm Pull-down-Widerstände ziehen beide Leitungen auf 0 V
- Bei Anschluss eines Full-Speed Geräts, bewirkt der 15kOhm Pull Up Widerstand auf der D+ Leitung einen High Pegel.
- Bei Low-Speed Geräten hat man an High Pegel an der D-Leitung.

Welche löschenbaren optischen Platten gibt es und auf welchen physikalischen Prinzipien beruhen sie?

- **rein optische Platten:** Beim Schreiben wird ein dünner Film einer besonderen Metallegierung mittels eines leistungsstarken Lasers zum Schmelzen gebracht, was einen Übergang vom kristallinen zum amorphen Zustand zur Folge hat, der weniger Licht reflektiert. Dies wird dann beim Abtasten mit einem schwächeren Laser ausgenutzt, indem man die Stärke des reflektierten Lichtes mißt. Beim Löschen wird die gesamte Plattenoberfläche einheitlich erhitzt und alle Teilchen der Metallegierung in den kristallinen zurückversetzt.
- **magneto-optisch Platten:** Beim Schreiben wird ein dünner Film einer besonderen Metallegierung mittels eines Lasers erhitzt, was die Umpolung von einzelnen magnetischen Bezirken mittels eines Magnets, der auf dem Schreib-Lesekopf angebracht ist, erleichtert. Das Metall kühl schnell ab und die Information bleibt erhalten. Beim Lesen nützt man die Eigenschaft aus, daß die Magnetisierungsrichtung der magnetischen Bezirke die Schwingungsebene des polarisierten Lichtstrahles des Lasers an der Plattenoberfläche entweder nach links oder rechts dreht (log. 1 oder 0). Das Löschen erfolgt wiederum durch allgemeine Erhitzung der gesamten Plattenoberfläche, wobei alle magnetischen Bezirke gleichsinnig ausgerichtet werden.

Nennen Sie einen Vorteil und einen Nachteil von Instruktionen mit variabler Länge im Vergleich zu Instruktionen mit fixer Länge.

Häufig verwendete Maschineninstruktionen können mit kürzeren Bitmustern codiert werden als weniger oft verwendete. Dadurch kann die Länge von Programmen reduziert werden. Dem gegenüber steht jedoch, dass ein Interpreter, der Maschineninstruktionen variabler Länge verarbeiten muss, naturgemäß langsamer arbeitet als ein Interpreter, der lediglich mit Instruktionen fester Länge zu tun hat.

Welche zwei Möglichkeiten gibt es, die Adressen der Ports für Input / Output-Operationen zu vergeben?

- Independent I/O-System: Hauptspeicher und Ports haben völlig voneinander unabhängige Adressen
- Memory-mapped-I/O-System: Ports werden so behandelt, als wären sie gewöhnliche Speicherstellen.

Netzwerke

Nennen Sie alle Layer des ISO OSI Reference Models!

1. Physical Layer: Übertragung einzelner Bits
2. Data-Link Layer: Sicherungsschicht – Physikalische Adressierung mit Fehlererkennung, Fehlerreparatur und Datenflusskontrolle. Portionierung, Header + Trailer → Frames
3. Network Layer: Vermittlungsschicht – Logische Adressierung der Endgeräte, Routing, Traffic Shaping. Herstellung der End-to-End-Verbindungen
4. Transport Layer: Abschirmung der höheren vor der Network Layer, Sicherstellung des Datenflusses zw. Quell- und Zielhost. Segmentierung von Datenpaketen und Stauvermeidung.
5. Session Layer: Prozessorkommunikation zw. Systemen, Aufbau von Sitzungen, Synchronisation. Kommunikation zw. Prozessen auf verschiedenen Hosts
6. Presentation Layer: Syntax & Semantik der übertragenen Information, Konversion in passende Formate, Verschlüsselung
7. Application Layer: enthält die eigentliche Applikation, stellt den Anwendungen Funktionalität zur Verfügung

Nennen Sie die Hauptaufgabe des Layer 3 (Network Layer) im OSI-Referenzmodell. Geben Sie das vorherrschende Network Layer Protokoll im Internet an.

Der Network Layer stellt End-to-End-Verbindungen zwischen einzelnen Hosts her. Vorherrschend im Internet ist das IP-Protokoll.

Nennen Sie 2 Aufgaben und 1 weitverbreitetes Protokoll des Transport Layers

- Sicherstellung des Datenfluss zwischen Quell- & Zielhost
- Datagramme werden beim Empfänger in der richtigen Reihenfolge zusammengefügt

Ein bekannter Vertreter ist das TCP Protokoll.

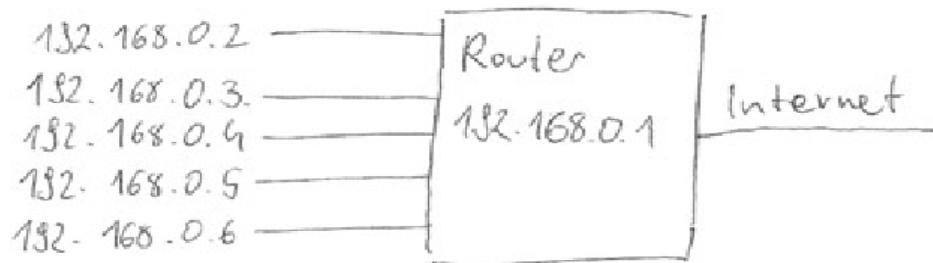
Was ist ein SSID?

SSID steht für Service Set Identifier und legt für ein WiFi Netzwerk einen eindeutigen Netzwerknamen fest. Sie besteht aus maximal 32 alphanumerischen Zeichen und muss bei beiden Kommunikationspartnern gleich sein, damit diese miteinander kommunizieren können.

Nennen Sie 2 Vorteile von IPv6 gegenüber IPv4!

- IPv6 besitzt einen wesentlich größeren Adressraum
- Die Header sind deutlich kleiner

Sie wollen von zu Hause 5 Rechner an das Internet anschließen. Ihr Internet Provider stellt aber nur eine Class B Adresse zur Verfügung. Geben Sie eine Netzwerktopologie an, sodass jeder Rechner mit einem Server im Internet kommunizieren kann.



Welche OSI Layer benötigt ein Hub, Funktionalität eines Hubs?

Ein Hub arbeitet auf dem OSI Layer 1 – Physical Layer.

Das Signal eines Netzteilnehmers wird in keinem Fall analysiert, sondern nur elektronisch aufgebessert (entrauscht und verstärkt) und im Gegensatz zum Switch der sich die richtigen Ports des Empfängers sucht an alle anderen Netzteilnehmer weitergeleitet. Ein Hub ist sozusagen ein „dummer“ Switch.

Betriebssysteme und Systemsoftware

Welche Aufgaben besitzt ein Betriebssystem? (151)

- Prozessmanagement
- Speicherverwaltung
- Interprozesskommunikation
- Zugriff auf I/O Geräte
- Zugriff auf Dateien
- Fehlerbehandlung
- Accounting

Was versteht man unter einem System Call? (88)

Betriebssystemaufrufe, mit Hilfe derer ein Prozess in der Lage ist, die verschiedensten Funktionen des Betriebssystems in Anspruch zu nehmen. Beispiel: Erzeugung eines Prozesses

Was versteht man unter logischer Parallelität von Prozessoren? (79)

Logische Parallelität von Prozessoren liegt dann vor, wenn mehrere Prozesse scheinbar gleichzeitig auf einem einzelnen Prozessor ausgeführt werden. Die logische Parallelität erlaubt es, belieb viele Prozesse zu erzeugen, ihren Ablauf zu beeinflussen und ihre Terminierung zu bewirken.

Wie ist bei einem Betriebssystem ein Zustandsübergang vom Zustand BLOCKED in den Zustand RUNNING zu bewerten? (14)

Deblockierung: Das externe Ereignis des blockierten Programmes, auf das gewartet wurde (zB TastaturEingabe) trat ein. Der abrupte Wechsel zu BLOCKED beim zur Zeit laufenden Prozess (weil da andere ja dran kommt) kann zu Datenverlusten führen => Entwickler muss das zuvor durchdenken.

Der Prozess wird asynchron beendet, was zu invaliden Daten führen kann. Bei den meisten Betriebssystemen ist dies nicht möglich. Bei Echtzeitsystemen ist diese Funktion beispielsweise für Notabschaltungen möglich.

Was ist der Unterschied zwischen den Zuständen BLOCKED & SUSPENDED? (42)

BLOCKED wird eingenommen, wenn der Prozess an einem Punkt angekommen ist, an dem er auf den Eintritt eines externen Ereignisses warten muss (zB Tastendruck auf Tastatur) SUSPENDED – Entspricht einer Blockierung von außen, wird also in der Regel vom OS oder einem anderem Prozess verursacht.

Welche potentiellen Gefahren bestehen in einem Rechnersystem, wenn ein direkter Zustandsübergang von BLOCKED nach RUNNING zulässig ist? (46)

Dadurch würde der aktuelle Prozess asynchron beendet, was zu invaliden Daten führen kann. Bei den meisten OS ist dies nicht möglich. Bei Echtzeitsystemen ist diese Funktion beispielsweise für Notabschaltungen möglich.

Modell eines Betriebssystems. Fehlende Zustände sowie Zustandsübergänge als gerichtete Kanten eintragen. (13)

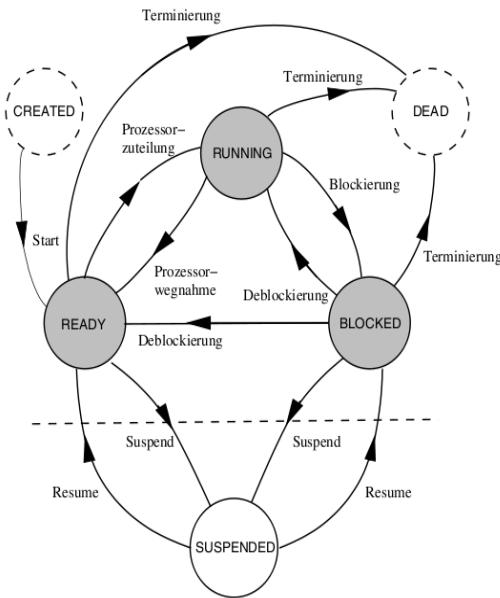


Abbildung 10.5: Prozess-Zustände

Nennen Sie alle Prozesszustände in einem Betriebssystem und zu welchem Zustand sie jeweils führen können.

CREATED → READY

READY → SUSPENDED, RUNNING, DEAD

RUNNING → READY, BLOCKED, DEAD

BLOCKED → RUNNING, DEAD; SUSPENDED, READY

SUSPENDED → READY, BLOCKED

DEAD

Was versteht man unter Race Conditions? (31)

Problem durch nicht genau bestimmmbare Reihenfolge paralleler Prozesse. Race Conditions entstehen nur dann, wenn unter bestimmten Umständen ein Prozess zufällig vor einem anderen „fertig“ wird. Dabei ist nicht das Scheduling das Problem, sondern die Parallelität.

Race Condition (Wettschlacht) sind Konstellationen, in denen das Ergebnis einer Operation vom zeitlichen Verhalten bestimmter Einzeloperationen abhängt. Das Eingrenzen dieses Problems, das durch die parallele Abarbeitung entsteht, ist durch das relativ seltene und indeterministische Auftreten eine sehr schwierige und zeitraubende Angelegenheit.

Beispiel 1: Ein Server Prozess ist dabei die SPOOL_QUEUE abzuarbeiten. Wenn diese leer ist bricht er ab und möchte per P_SLEEP in den Zustand BLOCKED. Bevor der Aufruf von P_SLEEP passiert, schreibt ein Client ein File in die SPOOL_QUEUE (Druckerauftrag) und sendet ein P_SIGNAL an den Serverprozess. Da dieser je doch noch im READY Zustand ist, wird der Aufruf ignoriert und der P_SLEEP vollführt. Das eingetragene File wird erst gedruckt, wenn ein neuer Druckerauftrag daherkommt. => CRITICAL SECTION, gegenseitiger Ausschluss (MUTUAL EXCLUSION)

Beispiel 2: Ein Client möchte einen Druckerauftrag absetzen und schreibt das zu druckende

File in die SPOOL_QUEUE. Dazu öffnet er die SPOOL_QUEUE und sucht sich einen leeren Eintrag mit Index k. In dem Moment wird ihm die Prozessorzuteilung entzogen und ein anderer Client setzt ebenfalls einen Druck auftrag ab, indem er sich an einer leeren Stelle mit Index k in der SPOOL_QUEUE einträgt. Nun kommt wieder der erste Client zum Zug und schreibt seinen Druckauftrag an die Stelle k, wo nun aber der andere Auftrag überschrieben wird und daher nie zur Ausführung kommt. => P_SIGNAL im Ready Zustand nicht ignorieren

Wohin werden beim Context Switch die Register des alten Prozesses gespeichert?

In den Prozessdeskriptor.

Nennen Sie zumindest 3 Bestandteile des Prozessdeskriptors! (81)

1. Process ID
2. Process State Information – Prozesszustand, Priorität, Registerinhalte
3. Process Control Information – Besitzer, Zugriffsrechte

Welche 2 Komponenten benötigt jeder Thread für sich alleine? (86)

1. Registersatz des Prozessors
2. ThreadSpezifische Daten (lokale Variablen, Rücksprungadressen, etc)

Parent & Child Prozesse – wait() & exit() System Calls (131)

Wenn der Parent Prozess in einem Unix Betriebssystem auf das Terminieren seines Child Prozesses wartet (mittels des wait() system calls), dann wird der Prozess so lange blockiert, bis der Child Prozess einen exit() system call macht. Dabei wird das Ergebnis des Child Prozesses (sein Rückgabewert) an den wartenden Parent Prozess übermittelt. Was passiert nun mit einem Child Prozess, der terminiert und sein Ergebnis an den Parent Prozess übermitteln will, wenn der Parent Prozess nicht vorher ein wait() aufgerufen hat?
=> Der Child Prozess terminiert, sein Rückgabewert geht verloren.

Nennen Sie mindestens 2 Probleme bei der Verwendung von Threads! (87)

1. Mehrere Threads in einem Prozess verwenden denselben Adressraum. Ein Prozess kann daher durch SpeicherManipulation andere Threads in demselben Prozess oder auch den Prozess selbst zum Absturz bringen.
2. Um das Prozessverhalten mit den Threads in Einklang zu bringen ist meistens eine gesonderte bzw. zusätzliche Behandlung notwendig.
3. Bei der Verwendung von Libraries, die nicht für den Einsatz in einem mehrfach genutzten Adressraum ausgelegt sind, können Probleme auftreten.

Was versteht man unter Scheduling? (89)

Als Scheduling bezeichnet man die Verteilung der Aufgaben eines OS auf die tatsächlich vorhandenen Ressourcen. Scheduling wird in mehreren Ebenen durchgeführt:

- Job Scheduling
- Prozess Scheduling
- Thread Scheduling

Prozess Scheduling (91)

Aufgabe des Prozess Scheduling ist das Verteilen der Prozesse (Context Switch) auf die Prozessoren. Es muss dafür gesorgt werden, dass jeder Prozess, der sich im Zustand READY befindet, in bestimmten Abständen und für eine bestimmte Zeit einen Prozessor zugeteilt bekommt, also in den Zustand RUNNING wechselt.

Prozesse in anderen Zuständen können ignoriert werden, da sie keinen Bedarf nach einem Prozessor haben. Wenn die Zuteilung sehr schnell geschieht, entsteht für jeden Prozess die Illusion einer kontinuierlichen Exekution.

Thread Scheduling (94)

Das Thread Scheduling ergänzt das Process Scheduling um eine untergeordnete Ebene.

Job Scheduling (90)

In Hochlastfällen ist es oft günstig, die angespannte Lage zu entschärfen und einige der konkurrierenden Prozesse zur Gänze auf Eis zu legen. Dadurch wird zB der System Overhead beim Prozess Scheduling verringert und damit die Nutzkapazität erhöht. Ziel des Job Scheduling ist es also, dem Prozess Scheduler gut bewältigbare Aufgaben zuzuteilen.

Forderungen an Scheduling Algorithmen (92)

- Fairness – Verteilung der Prozessorkapazität soll gerecht sein
- Effizienz – Prozessoren sollen möglichst optimal ausgelastet werden
- Durchsatz – Anzahl der verarbeiteten Jobs sollte maximal sein
- Antwortzeiten – Prozesse sollten raschen Response liefern
- Context Switch Time – Die Prozessorwechselzeit ist die für die Berechnung einer Scheduling Entscheidung notwendige Zeit und soll minimal sein.

Was muss ein Betriebssystem machen, wenn zwischen zwei threads, die zum gleichen Prozess gehören, umgeschaltet wird (dh Scheduler führt einen Context Switch zwischen den theade aus)? (47, p 260, 269)

CONTEXT SAVE (Threadspezifische Daten (lokale Variablen, Stack) und Register retten)
CONTEXT RESTORE (Threadspezifische Daten vom neuen Thread & Register vom neuen Thread zurückschreiben)

Handelt es sich beim Round Robin Scheduling um ein preemptives Verfahren? (148)
Nein, das RRS Verfahren ist nicht preemptiv. Ein laufender Prozess bleibt immer für einen ganzen Zeitzyklus im Zustand RUNNING.

Erklären Sie die Funktionsweise vom Round Robin Scheduling

Das Rundlauf-Verfahren gewährt allen Prozessen nacheinander für jeweils einen kurzen Zeitraum Zugang zu den benötigten Ressourcen. Man nennt dies auch Arbitrierung.

Skizzieren Sie die Zustandsübergänge eines laufenden Prozesses beim Round Robin Scheduling (RRS)! (12)

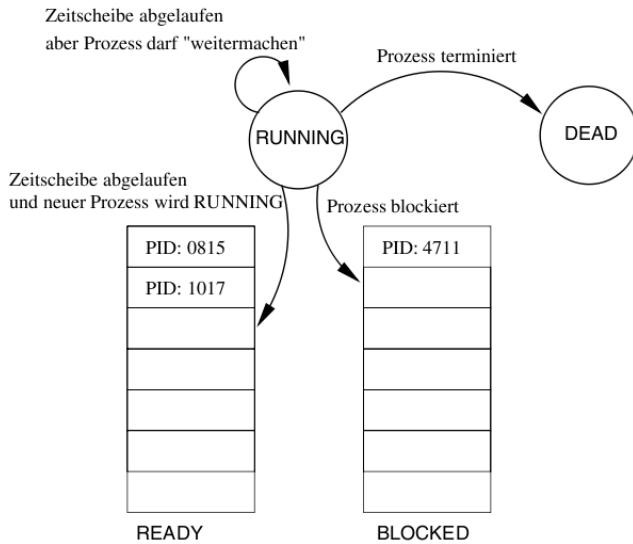


Abbildung 10.13: Zustandsübergänge eines laufenden Prozesses bei Round Robin Scheduling

Scheduling Methoden (93)

Das Hauptproblem des Scheduler ist, dass er keine Vorhersagen über das Verhalten der Prozesse zur Verfügung hat. Daher muss er seine Entscheidungen heuristisch treffen. Wegen des geringeren Overheads werden primitive Methoden bevorzugt.

- **FCFS – First Come First Served:** non preemptive Technik – heute nicht mehr angewendet
- **RRS – Round Robin Scheduling :** hier bekommt jeder READY Prozess für ein bestimmtes Zeitintervall einen Prozessor zugeteilt. Läuft das Quantum des Prozessors ab, wird er an das Ende der READYLIST wieder eingereiht. Kritisch für das RRS ist die Länge des Time Slices. Ist sie zu klein, reduziert der Scheduling Overhead die nutzbare Prozessorleistung unzulässig. Ist sie zu groß sind Antwortzeiten unbefriedigend.
- **SPS – Static Priority Scheduling :** Prozessen wird eine Priorität zugeordnet und die Prozessorkapazität wird entsprechend aufgeteilt. RUNNING ist immer jener Prozess mit höchster Priorität. Wechselt ein höherer Prozess nach READY, nimmt der Scheduler dem aktuellen Prozess den Prozessor weg und bringt den neuen Prozess zur Ausführung. Diese preemptive Strategie ist einfach zu implementieren. Es kann aber passieren, dass ein niedrigpriorer Prozess nie einen Prozessor zugeteilt bekommt (=Starvation)
- **DPS – Dynamic Priority Scheduling :** adaptive Scheduling Strategie. Durch dynamische Änderung von Prioritäten wird sichergestellt, dass kein Prozess auf der Strecke bleibt. Dies kann zB durch Zuteilung eines SollServices passieren.
- Weitere: Shortest Job First, Shortest Remaining Time First

Was versteht man unter Context Switch Time? (8, p 260, 269)

Wenn ein laufender Prozess von einem anderen unterbrochen wird, muss dafür gesorgt werden, dass die Inhalte der Register (der Context) gesichert wird. Bei einem Context Switch wird einerseits der aktuelle Context gespeichert, sowie ein anderer restored.

CST: Context Save + Context Restore + Zeit für die Findung einer Scheduling Entscheidung

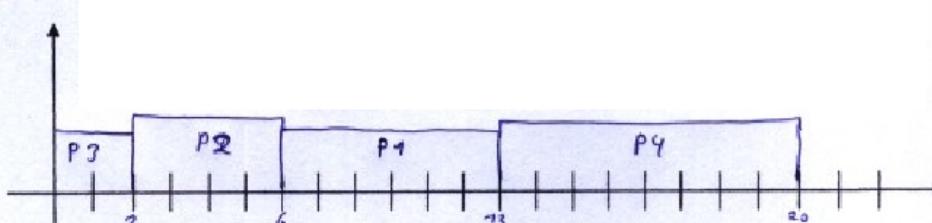
Was ist ein Trap? Geben Sie ein Beispiel!

Ein Trap (auch Interrupt) genannt ist eine Unterbrechung der momentan vom Prozessor ausgeführten Programmsequenz, um auf einen im normalen Programmablauf nicht vorgesehenen Zustand zu reagieren.

In einem Betriebssystem ist der folgende Scheduling Algorithmus implementiert: Shortest Job First. Folgende Ausführungszeiten sind angegeben: (16)

Progr.-Prozess #1	6 ZE
Progr.-Prozess #2	4 ZE
Progr.-Prozess #3	2 ZE
Progr.-Prozess #4	8 ZE

5.1 Geben Sie die Reihenfolge der zu bearbeitenden Programm-Prozesse an und zeichnen Sie auf einer vorgegebenen Zeitskala die Folge der Programmprozesse maßstabsgerecht ein ! (4)



5.2 Berechnen Sie die durchschnittliche Wartezeit t_{MWZ} für alle Programmprozesse. (3)

$$t_{MWZ} = \frac{0 + 2 + 6 * 12}{4} = \frac{80}{4} = 5$$

5.3 Welche Kritik kann man an diesem Scheduling-Algorithmus anbringen? (3)

- Betriebssystem kann keine Laufzeiten vorhersagen
- Letztes Programm wartet eventuell lange auf Ausführung.

Betriebssysteme – Interprozesskommunikation

Nennen Sie einen Nachteil von asynchronen Methoden zur Interprozesskommunikation IPC im Vergleich zu synchronen Methoden. (80)

Der Prozess hat zu jeder Zeit damit zu rechnen, aufgrund eines eingehenden Signals in der Ausführung unterbrochen zu werden. (=> abrupte Unterbrechung kann zu Problemen führen)
=> synchrone Methoden (Semaphoren, Message Passing)

Was versteht man unter einem Semaphor? (43)

Semaphore sind ein geeignetes Konzept zur Vermeidung von RaceConditions. Es handelt es sich hierbei um eine synchrone Methode hinsichtlich der IPC (Interprozesskommunikation). Ein Semaphor ist ein Objekt, das aus einem auf 0 initialisierten Counter und einer zunächst leeren Liste von ProzessIDs besteht. Eine Reihe von System Calls erlaubt nun die Synchronisation paralleler Prozesse.

Aus welchen Datenbereichen besteht die Datenstruktur des Semaphors? (44)

Ein Semaphor besteht aus einem Zähler und einer Warteschlange.

Wie arbeiten Semaphoren? (45)

Semaphoren besitzen 2 Operationen: S_P(sID) & S_V(sID)

- S_P : Beim Aufruf wird zunächst der Counter dekrementiert. Ist zusätzlich der Counter beim Aufruf kleiner oder gleich 0, dann wird der aufrufende Prozess mit seiner PID in die Warteliste eingetragen und in den Zustand BLOCKED versetzt. Durch S_P wird der Prozess an seiner Ausführung gehindert und bleibt „stecken“.
- S_V : Hierbei wird der Counter inkrementiert. Wenn der Counter kleiner 0 ist, wird zusätzlich der erste Prozess der Warteschlange entfernt und in den Zustand READY versetzt.

Das Problem der RACE CONDITIONS wird hierbei auf die Betriebssystemebene verlagert, wo sie beispielsweise durch Serialized Actions hinsichtlich der SYSTEM CALLS gelöst werden können. Die Prozess Queue kann nach dem FIFO Prinzip arbeiten, muss es aber nicht.

Message Passing (95)

Im Gegensatz zu Semaphoren handelt es sich beim Message Passing um einen Mechanismus, der eine Kommunikation im Sinne eines Datenaustausches zwischen Prozessen erlaubt. Üblicherweise deponiert dabei ein Senderprozess Nachrichten an einem Message Exchange punkt, von wo sie ein Empfängerprozess abholen kann. Praktisch wird dies wieder durch System Calls ermöglicht. (synchrone Methode)

Geben Sie ein Beispiel für asynchrone IPCMethoden, die unter Unix und auch in MS Windows realisiert sind. (130)

Sockets werden unter Windows & unter Unix verwendet. Sind die aber asynchron? {TODO: Überprüfen}

Was versteht man unter atomic actions? (41, p217, 289, 323)

Werden auch als unteilbare Operationen bezeichnet. Eine Folge von Instruktionen, die entweder ganz oder gar nicht durchgeführt werden müssen. Bei asynchronem Wechsel von Blocked auf Running dienen sie dazu, invaliden Daten vorzubeugen.

Kann es zwischen Threads zu einem Deadlock kommen? (35, Seite 290)

Die meisten Programme werden für sequentielle Abarbeitung geschrieben, weshalb es zu Deadlocks bei paralleler Ausführung kommen kann. Je mehr Programme parallel ausgeführt werden müssen, desto größer die Wahrscheinlichkeit eines Fehlers, weil die Verwaltung komplizierter wird.

Möglichkeiten zur Behandlung ?

1. **Deadlock Detection & Recovery** : Der Resource Allocation Graph wird ständig aktualisiert. Finden sich darin Zyklen, so wird dieser Zyklus durch Terminieren der beteiligten Prozesse gelöst. Diese brutale Vorgehensweise kann zu Problemen führen. Die Analyse des Graphen verursacht zudem einen nicht unerheblichen System Overhead.
2. **Deadlock Prevention** : Dabei werden durch die Beachtung gewisser Kriterien beim Design eines Betriebssystems viele notwendige Bedingungen für einen Deadlock einfach verhindert. zB Vermeidung von Mutual Exclusion: beliebig viele Prozesse können gleichzeitig auf den Drucker schreiben. Das eigentliche Device selbst wird durch den Printer Server verwaltet.
3. **Deadlock Avoidance** : Hierbei werden die Objekt Anforderungen sorgfältig beobachtet und versucht vorausschauend zu klären, ob die Zuteilung Deadlocks nach sich ziehen könnte. Es gibt Algorithmen dafür – diese liefern aber nur bei gewissen Zusatzinformationen sichere Zuteilungsentscheidungen

Was versteht man unter einem Deadlock? (53, p. 290, 253)

Zustand, in dem ein oder mehrere Prozesse auf Betriebsmittel warten, die dem Prozess selbst oder einem anderen beteiligten Prozess zugeteilt sind. ZB eine Straßenkreuzung, an der von allen vier Seiten Autos kommen und nun (die Regel rechts vor links vorausgesetzt) darauf warten, dass das jeweils rechts wartende Auto zuerst fährt.

Wann entstehen Deadlocks? (54)

Vier Bedingungen sind für die Entstehung eines Deadlocks notwendig:

1. **Mutual Exclusion**: Ein bestimmtes Objekt kann zu jedem Zeitpunkt von höchstens einem Prozess okkupiert sein. Der Zugriff auf die Betriebsmittel ist damit exklusiv.
2. **Resource Waiting**: Wenn ein beantragtes Objekt gerade besetzt ist, geht der anfordernde Prozess in den Zustand BLOCKED über und wartet auf dessen Freigabe. Die Prozesse fordern Betriebsmittel an, behalten aber zugleich den Zugriff auf andere.
3. **Partial Allocation**: Prozesse, die bereits im Besitz von Objekten sind, können die Zuteilung weiter beantragen.
4. **Nonpreemption**: Die Betriebsmittel werden ausschließlich durch die Prozesse freigegeben. Ein einmal zugeteiltes Objekt muss explizit durch den die Ressource haltenden Prozess wieder freigegeben werden, kann ihm also nicht zwangsweise entzogen werden.

Betriebssysteme – Speicherverwaltung

Wozu dient ein virtueller Speicher ?

Bei einer großen Anzahl gleichzeitig speicherresidenter Programme waren vor der Erfindung dieses Konzeptes sehr große Speicher notwendig. In der Zeit der Kernspeicher waren jedoch der Kapazität Grenzen gesetzt. Diesem Problem wurde durch die Erfindung des virtuellen Speichers abgeholfen: Im Prinzip dient dabei ein Externspeicher (eine Disk) als virtueller Hauptspeicher. Da ein Prozessor nur auf den physikalischen Hauptspeicher direkt zugreifen kann, werden die für die Exekution benötigten „Abschnitte“ einfach vom Externspeicher geladen. Zu jedem Zeitpunkt ist also nur der gerade benötigte Teil eines Programms speicherresident.

Was ist die Aufgabe eines Linkers? (83, p. 296)

Der Linker fügt die als „Relocatable Object Code“ vorliegenden Programmmodulen, welche aus entsprechen dem Sourcecode kompiliert worden sind, zu einem ausführbaren Programm zusammen.

Welches Problem tritt beim Linken im Zusammenhang mit direkter Adressierung auf und wie kann es behoben werden?

Beim Linken entstehen durch das Zusammenfügen von Modulen Veränderungen in den Adressen der Sprungziele. Bei direkter Adressierung muß der Linker eine Modifikation der Operanden durchführen, da sonst die Adressen der Sprungziele verfehlt werden. Für die Code-Segmente können Probleme dieser Art durch die Berücksichtigung der Konvention für Position Independent Code (PIC) gelöst werden. Dabei werden z.B. statt absoluter Sprungadressen nur die Distanzen (Displacements) von der aktuellen Adresse angegeben. Da sich durch die Verschiebung die Displacements nicht ändern, entfallen die Operandenmodifikationen völlig.

Was versteht man unter Swapping? (120)

Beim Swapping (auch RollIn/RollOut) werden ProzessImages als Ganzes zwischen dem physikalischen und dem virtuellen Speicher hin und her bewegt. Diese einfache Strategie benötigt sehr wenig Hardware-Unterstützung, allerdings dauert das Kopieren des gesamten Images sehr lange. Dazu kommt noch die Speicherverschwendungen durch die Fragmentierung und die etwas unflexible Memory Protection.

Was versteht man unter Paging? (121, p303)

Aufteilung des virtuellen Adressraums in gleich große Teile, die vom Programm unabhängig sind. Die Aufteilung erfolgt transparent. Umsetzung virtueller in physikalische Adressen:

(001DA472)16
8KB
virtuelle Adresse
Pagegröße entspricht (2000)16

$$\text{Page Nummer} = \frac{\text{virtuelle Adresse}}{\text{Pagegröße}} = \frac{1\text{DA}472}{2000} = (ED)_{16}$$

nachsehen in Page Table => Frame Nummer= $(1B)_{16}$

$$\text{Startadresse} = \text{Page Nummer} * \text{Pagegröße} = ED * 2000 = (1\text{DA}000)_{16}$$

$$\text{Offset} = \text{virtuelle Adresse} - \text{Startadresse} = 1\text{DA}472 - 1\text{DA}000 = (472)_{16}$$

$$\text{Physikalische Adresse} = \text{Frame Nummer} * \text{Pagegröße} + \text{Offset} = 1B * 2000 + 472 = (36472)_{16}$$

Wann tritt ein Page Fault auf? (39, p 305)

Wenn eine Adresse angefordert wird, die in einer Page ist, die sich zur Zeit nicht im Hauptspeicher, sondern auf der Platte (Externspeicher) befindet, dann tritt ein Page Fault auf. Dabei wird das Programm unterbrochen und die benötigte Page geladen.

Im Falle eines Page Faults sind welche Heuristiken (Page Replacement Methoden) anwendbar? (40, 305)

- First In First Out (FIFO): Jede Page bekommt zum Zeitpunkt Ihres Ladens einen Zeitstempel. Wenn ein PageFault aufzulösen ist, wird die „älteste“ Page ersetzt.
- Least Recently Used (LRU): Jene Page wird ersetzt, deren letzte Referenz am weitesten zurück liegt
- Least Frequently Used (LFU): Jene Page ersetzen, die am wenigsten benutzt wurde
- Not Used Recently (NUR): Pages erhalten eine Kennung Referenced/Not Referenced, welche in gewissen Abständen zurückgesetzt wird.

Was versteht man unter Segmentierung? (122, p. 307)

Aufteilung des virtuellen Adressraums in unterschiedlich große Teile, die direkt auf Teile des Programms abgebildet werden können. Die Aufteilung erfolgt nicht transparent.

Was versteht man unter Trashing? (32, p. 334)

Trashing ist ein Phänomen, das beim Paging auftritt, wenn der Speicher zu klein ist, um alle Working Sets aufzunehmen: Die Maschine ist praktisch ausschließlich damit beschäftigt, die referenzierten Pages vom Externspeicher zu laden.

Betriebssysteme – Sicherheit

Zählen Sie grundlegende Sicherheitsanforderungen auf! (145, p. 329)

- Secrecy (Geheimhaltung) – Forderung nach Geheimhaltung von Daten. (nur autorisierte Personen)
- Integrity (Integrität) – Daten dürfen nur von autorisierten Benutzern verändert werden.
- Verfügbarkeit (Availability) – alle Teile eines Computersystems können von autorisierten Benutzern verwendet werden

Beschreiben Sie die 4 Bedrohungsklassen! (146, p. 329)

- Interruption (Unterbrechung) – zB Zerstören von Hardware, Durchtrennen von Kommunikationsleitungen, DoS Attacken
- Interception (Abfangen) – zB Sniffen des Netzwerkverkehrs, Kopieren von unautorisierten Daten
- Modification (Modifikation) – zB verändern von übertragenen Daten, Angriffe gegen die Integrität von Informationen, Verändern von Dateiinhalten, etc
- Fabrication (Fälschung) – gefälschte Daten gelangen in den Informationsfluss, zB Einfügen von zusätzlichen Einträgen in Dateien (Passworddateien)

Was ist ein Trap? Geben Sie ein Beispiel! (76, p. 332)

Ein Trap ist ein Interrupt auf Softwareebene. Genauer gesagt handelt es sich um eine Unterbrechung der momentan vom Prozessor ausgeführten Programmsequenz, um auf einen im normalen Programmverlauf nicht vorgesehenen Zustand zu reagieren.

Die CPU löst bei Fehlern geschützte Zugriffe, verbotene Instruktionen (zB Division durch Null), Singlestep Debugging, MemoryManagementEreignisse, etc – sogenannte Traps aus.

Nennen Sie zumindest 4 Design Prinzipien, deren Befolgung zu robusteren und damit besseren Endprodukten führen! (143, 336)

- Least Privilege : Benutzer & Prozesse sollen über minimale Privilegien verfügen, die für die korrekte Abarbeitung der Aufgabe notwendig sind.
- Complete Mediation : Jeder Zugriff auf eine Ressource muss kontrolliert werden.
- Acceptability : Sicherheitsmaßnahmen dürfen den Benutzer nur minimal beeinträchtigen (ansonsten Deaktivierung)
- Economy of Mechanism : Sicherheitsmaßnahmen sollen so einfach und kompakt wie möglich sein. Sicherheit muss schon während der Designphase bedacht werden.
- Open Design : Die Sicherheit darf nicht darauf beruhen, dass die dahinterliegenden Mechanismen geheim gehalten werden. Nur wenn das Design offen gelegt und von Experten analysiert wurde, kann der Benutzer Vertrauen in das System haben.

Was versteht man unter Trusted Computing? (143, p. 336)

Die Idee des Trusted Computing ist es, ein System vor Attacken durch bösartige Software zu schützen. Zu diesem Zweck soll sichergestellt werden, dass auf Daten nur von vertrauenswürdigen Applikationen zugegriffen werden kann. Realisierung sehr kompliziert und mit viel Kryptografie verbunden.