

TINF PRÜFUNGSVORBEREITUNG

SS08

11.05.08

by KingBerT

Dokumente die u.a. aufgenommen wurden:

- ti_tu_schildt-po_2007-10-16.pdf
- eti_tu_schildt-po-26-06-07.pdf
- ETI Fragenkatalog.pdf
- ... diverse andere Fragen aus verschiedensten POs

Inhaltsverzeichnis

TINF PRÜFUNGSVORBEREITUNG.....	1
1 Definieren Sie die Hysterese beim Schmitt Trigger.....	6
2 Nennen Sie die Bitfolgen der undefinierten Zustände beim Asynchronzähler, wenn er vom Zählerstand 7 auf den Zählerstand 8 übergeht.....	7
3 Geben Sie die Schaltung eines Synchronzählers an!.....	8
4 Die ALU generiert 2 Signale N & Z. Geben Sie kurz an, was die Signale N & Z bedeuten.....	8
5 Erläutern Sie die Funktion der anti-parallelen Diodenschaltung im Gegenkopplungsweig des OPVs eines Zero-Crossing-Detectors.....	8
6 Wie ist mit dem Vorzeichen-Bit bei einem arithmetischen Rechtsshift zu verfahren?.....	8
7 Was passiert mit dem Vorzeichenbit bei einem arithmetischen Linksshift?.....	9
8 Was versteht man unter Context-Switch-Time?.....	9
9 Wodurch unterscheidet sich bei USB die Connect Erkennung von Full- und Low Speed Geräten?.....	9
10 Was versteht man unter Interlocking beim Pipelining Konzept?.....	9
11 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.....	9
12 Skizzieren Sie die Zustandsübergänge eines laufenden Prozesses beim Round-Robin-Scheduling (RRS)!.....	10
13 Modell eine Betriebssystem. Fehlende Zustände sowie Zustandsübergänge als gerichtete Kanten eintragen.....	10
14 Wie ist bei einem Betriebssystem ein Zustandsübergang vom Zustand BLOCKED in den Zustand RUNNING zu bewerten?.....	11
15 Was versteht man unter dem Cycle-Stealing Verfahren?.....	11
16 In einem Betriebssystem ist der folgende Scheduling Algorithmus implementiert: Shortest Job First. Folgende Ausführungszeiten sind angegeben:.....	11
17 Skizzieren Sie als einfaches Blockdiagramm die Harvard Architektur eines Rechners und beschriften Sie die Komponenten!.....	12
18 Nennen Sie die Adressierungsarten die Sie kennen!.....	12

19 Erläutern Sie die direkte, indirekte, register-indirekte, programm-counter-relative Adressierung!.....	12
20 Was ist Interleaved Memory und wozu wird er eingesetzt?.....	12
21 Nennen Sie die 3 Replacement Strategien (Cache)!.....	13
22 Erklären Sie die Buffered-Write-Methode im Zusammenhang mit Caches! Welche Probleme können dabei auftreten?.....	13
23 DMA: Was es ist und warum es verwendet wird?.....	13
24 Vorgang des DMA erklären.....	13
25 Arten von DMA (2).....	13
26 Controller & Co-Prozessoren.....	13
27 Wozu dient ein virtueller Speicher.....	14
28 Welchen Vorteil besitzt die „von Neumann“ und die „Harvard-Architektur“!.....	14
29 Welchen Vorteil besitzt die „Harvard-Architektur“ gegenüber „von Neumannschen“, und wo wird dieser genutzt?.....	14
30 Geben Sie einige Quellen für Probleme an, die beim Instruction Pipelining auftreten.....	14
31 Was sind Race Conditions?.....	14
32 Was versteht man unter Trashing?.....	15
33 Nennen Sie alle Layer des ISO OSI Reference Models!.....	15
34 Welches Problem tritt beim Linken im Zusammenhang mit direkter Adressierung auf und wie kann es behoben werden?.....	15
35 Kann es zwischen Threads zu einem Deadlock kommen?.....	15
36 Welche Weiterentwicklung des RISC Architektur sind im Vorlesungsbuch beschrieben?.....	16
37 Welche löschbaren optischen Platten gibt es und auf welchen physikalischen Prinzipien beruhen sie?	16
38 Was ist ein endlicher deterministischer Automat?.....	16
39 Wann tritt ein Page-Fault auf?.....	16
40 Im Falle eines Page-Faults sind welche Heuristiken (Page Replacement Methoden) anwendbar?	16
41 Was versteht man unter atomic actions?	17
42 Was ist der Unterschied zwischen den Zuständen BLOCKED & SUSPENDED?.....	17
43 Was versteht man unter einem Semaphor?.....	17
44 Aus welchen Datenbereichen besteht die Datenstruktur des Semaphors?.....	17
45 Wie arbeiten Semaphoren?.....	17
46 Welche potentiellen Gefahren bestehen in einem Rechnersystem, wenn ein direkter Zustandsübergang von BLOCKED nach RUNNING zulässig ist?.....	17
47 Was muss ein OS machen, wenn zwischen 2 Threads, die zum gleichen Prozess gehören, umgeschaltet wird?.....	17
48 Was versteht man unter Flow Control?.....	18
49 Was versteht man unter bedingten bzw. unbedingten Sprüngen?.....	18
50 Was versteht man unter dem Program Status Word?.....	18
51 Was sind Subroutine Calls?.....	18
52 Was sind Interrupts?.....	18
53 Was versteht man unter einem Deadlock?.....	18

54 Wann entstehen Deadlocks?.....	19
55 Erklären Sie die RISC Architecture!.....	19
56 Welche Refresh Verfahren gibt es bei Speicherzellen?.....	19
57 Geben Sie die Formel für die Kondensatorladung / Entladung an!.....	19
58 Welche Arten von Speicherbausteinen kennen Sie?.....	20
59 Was sind „0-aktive Eingänge“ bei ICs (integrierten Schaltungen)?.....	20
60 Welche Funktion hat der ENABLE Eingang bei einem Decoder?.....	20
61 Geben Sie die Übertragungskennlinie eines Inverters an. Wo liegt die verbotene Zone?.....	20
62 Welche 3 Sichtweisen gibt es beim Entwurf elektronischer Schaltungen laut dem Y-Modell?21	
63 Welche 3 Arbeitsschritte sind zur Erstellung eines Moduls bzw. einer Komponente in VHDL erforderlich?.....	21
64 Von welchen zwei Faktoren hängen Zustandsänderungen bei Schaltwerken ab?.....	22
65 Wozu dient die Ziehkapazität bei einem Quarzoszillator?.....	22
66 Was versteht man unter Fan-In?.....	22
67 Was versteht man unter Fan-Out?.....	22
68 Zeichnen Sie eine Gatterschaltung für die XOR Funktion.....	22
69 Erklären Sie die Abkürzung ASIC!.....	22
70 Zeichnen Sie die Übertragungskennlinie eines invertierenden Schmitt Triggers!.....	22
71 Was versteht man unter dem Tastgrad g einer Rechteckimpulsfolge des Taktgenerators eines Rechners?.....	22
72 Was versteht man unter dem Scratchpad eines Prozessors?.....	22
73 Welchen Vorteil hat die Erhöhung der Baudichte bei Mikroprozessoren?.....	22
74 Nennen Sie einen Vorteil, sowie einen Nachteil von Instruktionen mit variabler Länge im Vergleich zu Instruktionen mit fixer Länge.....	23
75 Welche 2 Möglichkeiten gibt es, die Adressen der Ports für Input/Output Operationen zu vergeben?.....	23
76 Was ist ein Trap? Geben Sie ein Beispiel?.....	23
77 Wo werden die Return Adressen bei Unterprozeduraufrufen gespeichert?.....	23
78 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.....	23
79 Was versteht man unter logischer Parallelität von Prozessoren?.....	23
80 Nennen Sie einen Nachteil von asynchronen Methoden zur Interprozesskommunikation IPC im Vergleich zu synchronen Methoden.....	23
81 Nennen Sie zumindest 3 Bestandteile des Prozessdeskriptors!.....	23
82 Durch welche 3 im Buch erläuterten Arten der Parallelverarbeitung kann die Performance eines Prozessors gesteigert werden?.....	24
83 Was ist die Aufgabe eines Linkers?.....	24
84 Was versteht man unter dem Program Counter bei einem Mikroprozessor?.....	24
85 Welche 2 Möglichkeiten gibt es, die Adresse einer Interrupt Service Routine (ISR) zu bestimmen?.....	24
86 Welche 2 Komponenten benötigt jeder Thread für sich alleine?.....	24
87 Nennen Sie mindestens 2 Probleme bei der Verwendung von Threads!.....	24
88 Was versteht man unter einem System Call?.....	24

89 Was versteht man unter Scheduling?.....	24
90 Job Scheduling.....	25
91 Prozess Scheduling.....	25
92 Forderungen an Scheduling Algorithmen.....	25
93 Scheduling Methoden.....	25
94 Thread Scheduling.....	26
95 Message Passing.....	26
96 Was versteht man unter einem Hazard?.....	26
97 Welche Arten von Pipeline Hazards gibt es?.....	26
98 Halbaddierer.....	26
99 Volladdierer.....	26
100 RS Latch.....	26
101 Codierer.....	27
102 Multiplexer.....	27
103 Demultiplexer.....	27
104 Operationsverstärker.....	27
105 Komparatoren für analoge Signale.....	28
106 Komparatoren für digitale Signale.....	28
107 Torschaltungen.....	28
108 Fensterdiskriminator.....	28
109 Univibrator.....	28
110 Quarzoszillator.....	29
111 Sägezahngenerator.....	29
112 Tristate Output.....	30
113 Open-Collector Schaltungen.....	30
114 Moore Schaltwerk.....	31
115 Mealy Schaltwerk.....	31
116 Prozessor.....	31
117 Rechenwerk.....	31
118 Leitwerk.....	31
119 Woraus ergibt sich die Performance eines Caches.....	31
120 Was versteht man unter Swapping?.....	31
121 Was versteht man unter Paging?.....	31
122 Was versteht man unter Segmentierung?.....	32
123 Was versteht man unter einem Cache Speicher? Wie ist er aufgebaut?.....	32
124 Welche Arten von Caches kennen Sie?.....	32
125 Was versteht man unter VHDL?.....	32
126 Welche Entwurfsansichten gibt es in VHDL?.....	32
127 Welche Schaltkreisfamilien kennen Sie?.....	32
128 Erklären Sie die Abkürzung VLSI!.....	33
129 Nennen Sie die Hauptaufgabe des Layer 3 (Network Layer) im OSI-Referenzmodell. Geben Sie das vorherrschende Network Layer Protokoll im Internet an.	33

130 Geben Sie ein Beispiel für asynchrone IPC-Methoden, die unter Unix und auch in MS Windows realisiert sind.	33
131 Parent & Child Prozesse – wait() & exit() System Calls.....	33
132 Mikroprozessor 16 : immediate addressing mode.....	33
133 Nennen Sie 2 Aufgaben und 1 weitverbreitetes Protokoll des Transport Layers.....	33
134 EEPROM: Nachteil im Vergleich zum RAM.....	34
135 Geben Sie 4 Pipeline Sprungstrategien an (bei den Control Hazards).....	34
136 Wohin werden beim Context-Switch die Register des alten Prozesses gespeichert?.....	34
137 Bestimmen Sie die Funktion der folgenden Schaltungen.....	34
138 Bauen Sie die Grundoperationen AND, OR & NOT aus NAND Gattern auf!.....	35
139 Bauen Sie die Grundoperationen AND, OR & NOT aus NOR Gattern auf!.....	35
140 Nennen Sie einen Nachteil eines asynchronen Zählers!.....	35
141 In welchem Zahlensystem zählt der synchrone Zähler?.....	36
142 Geben Sie 3 Merkmale für die Leistung eines Betriebssystems an!.....	36
143 Nennen Sie zumindest 4 Design Prinzipien, deren Befolgung zu robusteren und damit besseren Endprodukten führen!.....	36
144 Was versteht man unter Trusted Computing?.....	36
145 Zählen Sie grundlegende Sicherheitsanforderungen auf!.....	36
146 Beschreiben Sie die 4 Bedrohungsklassen!.....	36
147 Geben Sie zum angegebenen Spannungsverlauf am Eingang eines invertierenden Schmitt-Triggers den Verlauf des Ausgangspegels an!.....	37
148 Handelt es sich beim Round Robin Scheduling um ein preemptives Verfahren?.....	37
149 Was ist ein SSID?.....	37
150 Nennen Sie 2 Vorteile von IPv6 gegenüber IPv4!.....	37
151 Welche Aufgaben besitzt ein Betriebssystem?.....	37
152 Welche OSI Layer benötigt ein Hub, Funktionalität eines Hubs?.....	37
153 Wozu dient die Architektur in VHDL?.....	37

1 Definieren Sie die Hysterese beim Schmitt Trigger

Wenn die Eingangsspannung eine vorgegebene obere Schaltschwelle erreicht, so nimmt der Ausgang einen binären Wert an. Unterschreitet die Eingangsspannung die untere Schaltschwelle, so nimmt der Ausgang den anderen binären Wert an.

Damit werden unterschiedliche Wege beim Schalten durchlaufen.

NICHT INVERTIEREND

INVERTIEREND

Diesen Sachverhalt nennt man Hysterese:

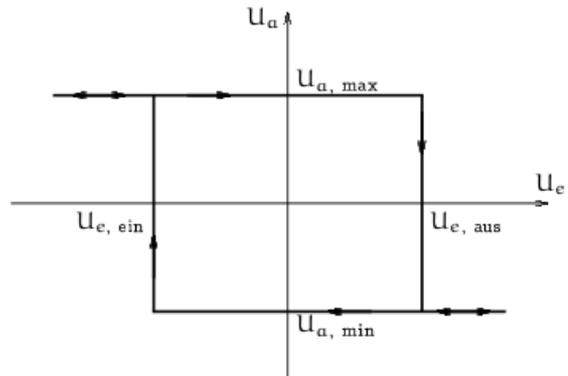
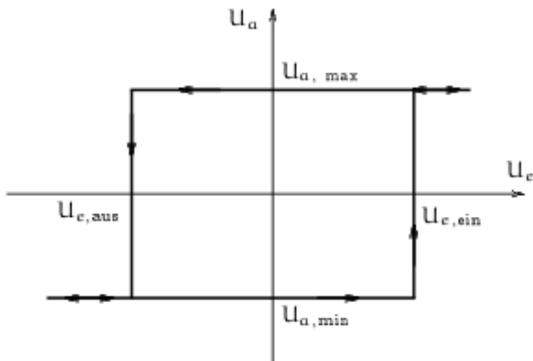


Abbildung 2.62: Übertragungskennlinie

Abbildung 2.59: Übertragungskennlinie

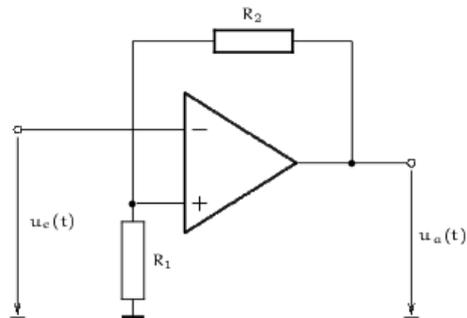
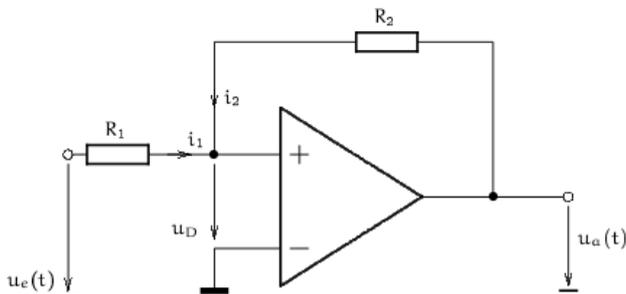


Abbildung 2.61: Nicht-invertierender Schmitt-Trigger

Abbildung 2.58: Invertierender Schmitt-Trigger

Einschaltpegel: $U_{e\text{ ein}} = -\frac{R_1}{R_2} \cdot U_{a\text{ min}}$

Einschaltpegel: $U_{e\text{ ein}} = \frac{R_1}{R_1 + R_2} \cdot U_{a\text{ min}}$

Ausschaltpegel: $U_{e\text{ aus}} = -\frac{R_1}{R_2} \cdot U_{a\text{ max}}$

Ausschaltpegel: $U_{e\text{ aus}} = \frac{R_1}{R_1 + R_2} \cdot U_{a\text{ max}}$

Hysterese: $U_{\text{HST}} = \left(\frac{R_1}{R_2}\right) \cdot (U_{a\text{ max}} - U_{a\text{ min}})$

Hysterese: $U_{\text{HST}} = \frac{R_1}{R_1 + R_2} \cdot (U_{a\text{ max}} - U_{a\text{ min}})$

Durch die Hysterese kann ein Schmitt Trigger auch bei langsamen Umschalten nicht schwingen.

Präzisions-Schmitt-Trigger

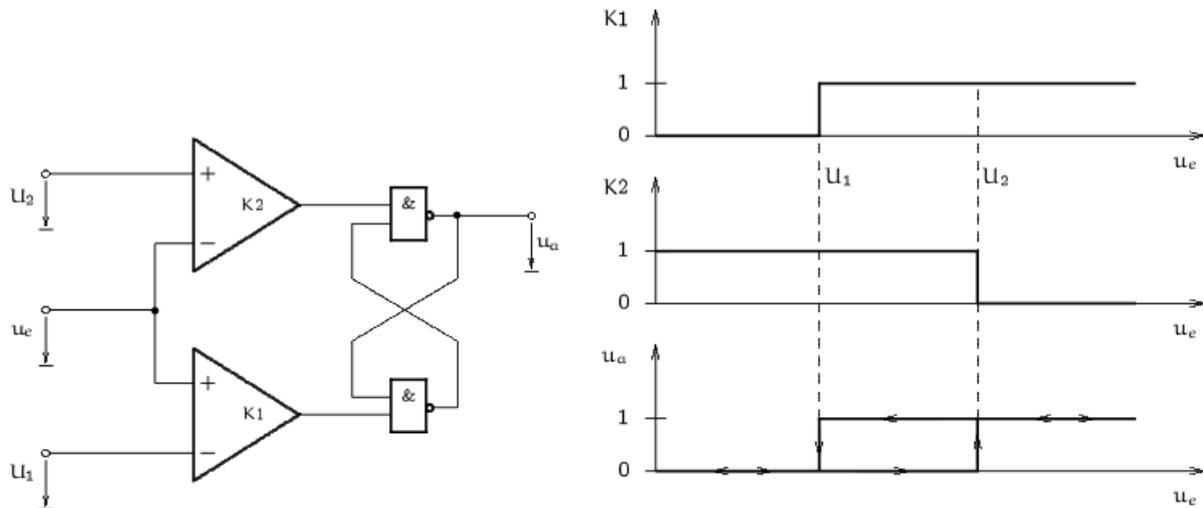


Abbildung 2.64: Präzisions-Schmitt-Trigger und dessen Arbeitsweise

Einschaltpegel $u_{e \text{ ein}} = U_2$ mit $U_2 > U_1$
 Ausschaltpegel $u_{e \text{ aus}} = U_1$

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

0111 => 1000

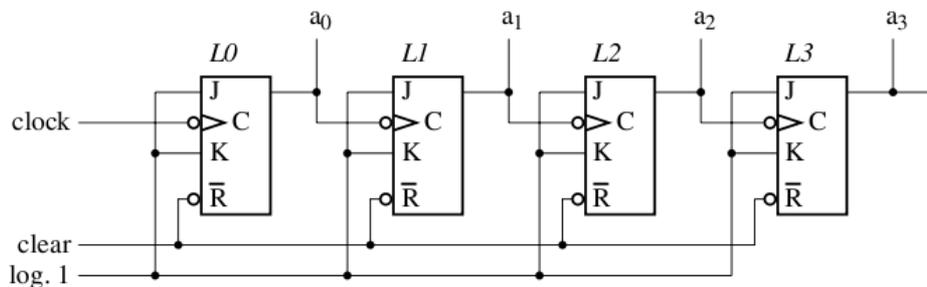


Abbildung 2.48: 4-Bit-Asynchronzähler (TTL-Realisierung)

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).

3 Geben Sie die Schaltung eines Synchronzählers an!

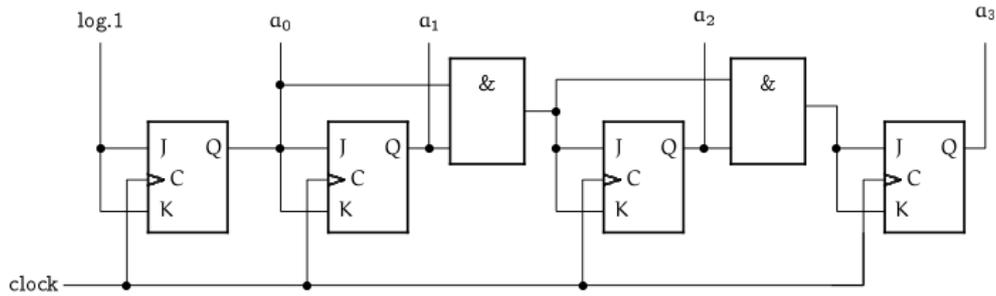


Abbildung 2.50: Vier-Bit-Synchronzähler

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

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)

4 Punkte

5 Erläutern Sie die Funktion der anti-parallelen Diodenschaltung im Gegenkopplungs-zweig des OPVs eines Zero-Crossing-Detectors

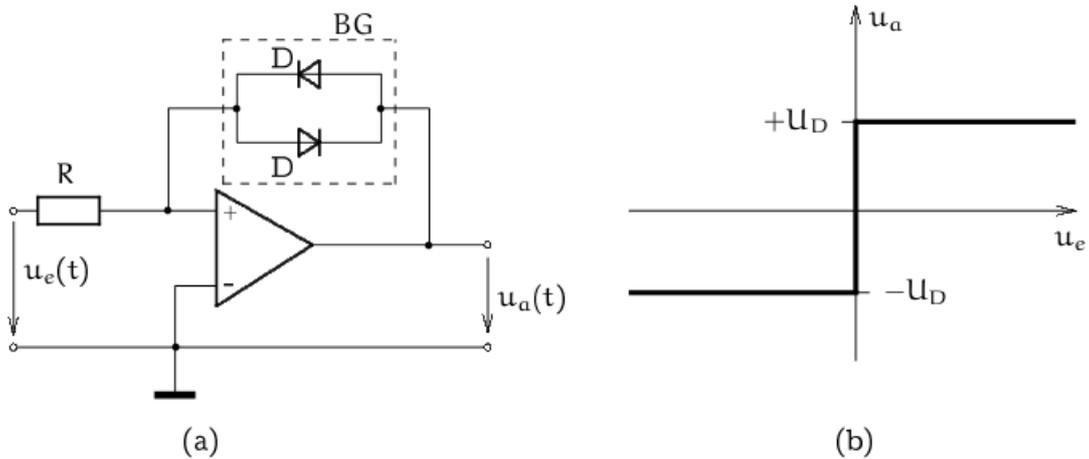


Abbildung 2.67: Nullspannungsschalter mit Operationsverstärker (BG = Begrenzerschaltung und Übertragungskennlinie $u_a = f(u_e)$)

Zur Vermeidung des gesättigten Betriebes des OPV (+ dadurch bedingte Schaltverzögerung) nutzt man eine zweiseitige Begrenzerschaltung zwischen Ein & Ausgang. (Begrenzung der negativen, sowie positiven Ausgangsspannung)

5 Punkte

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

Laut Wikipedia (http://de.wikipedia.org/wiki/Bitweiser_Operator) wird bei einem Rechts-shift das Vorzeichenbit kopiert.

1100 RECHTS-SHIFT

oder

0100 RECHTS-SHIFT

= 1110

= 0010

4 Punkte

7 Was passiert mit dem Vorzeichenbit bei einem arithmetischen Linksshift?

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

8 Was versteht man unter Context-Switch-Time?

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

4 Punkte

9 Wodurch unterscheidet sich bei USB die Connect Erkennung von Full- und Low Speed Geräten?

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.

4 Punkte

10 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 merken

4 Punkte (S164)

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

4 Punkte

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

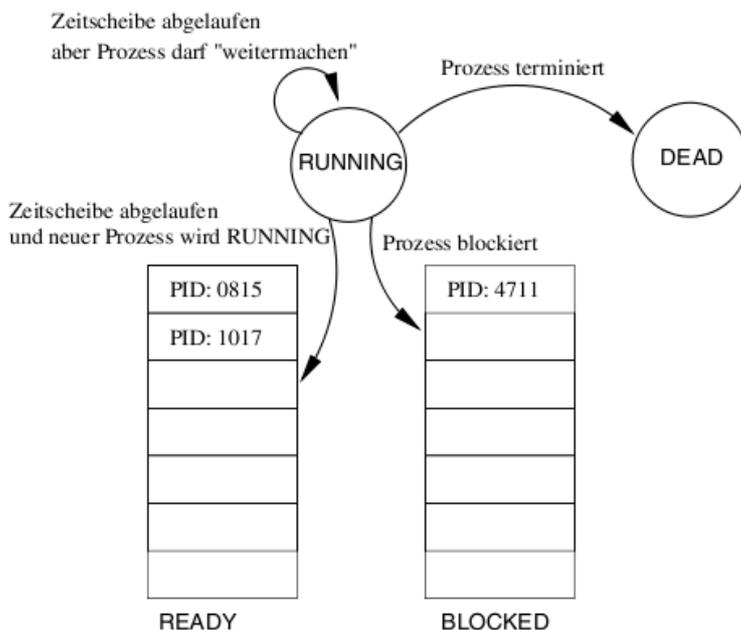


Abbildung 10.13: Zustandsübergänge eines laufenden Prozesses bei Round Robin Scheduling

8 Punkte

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

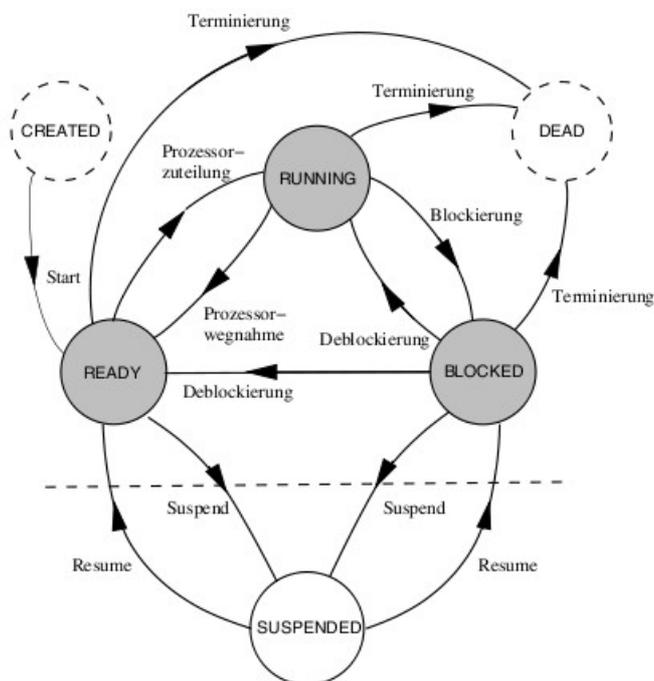


Abbildung 10.5: Prozess-Zustände

8 Punkte

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

Deblockierung: Das externe Ereignis des blockierten Programmes, auf das gewartet wurde (zB Tastatur-Eingabe) 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.

4 Punkte

15 Was versteht man unter dem Cycle-Stealing Verfahren?

Dynamische RAMs müssen zyklisch refreshed werden – 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 DMAs

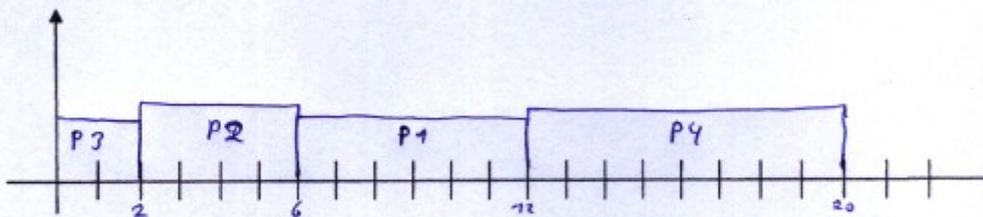
Bei einem DMA stiehlt der Controller dem Prozessor den Maschinenzyklus, indem er den Bus anfordert, sobald das Gerät seine Bereitschaft zum Datentransfer meldet. Der Prozessor kann erst wieder auf den Speicher zugreifen, wenn die Aktion beendet ist.

4 Punkte

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

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)

$$t_{MWZ} = \frac{0 + 2 + 6 + 12}{4} = \frac{20}{4} = 5$$

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

- Betriebssystem kann nicht hell sehend und Laufzeiten vorhersagen
- Letztes Programm wartet eventuell lange auf Ausführung.

10 Punkte (4+3+3)

17 Skizzieren Sie als einfaches Blockdiagramm die Harvard Architektur eines Rechners und beschriften Sie die Komponenten!

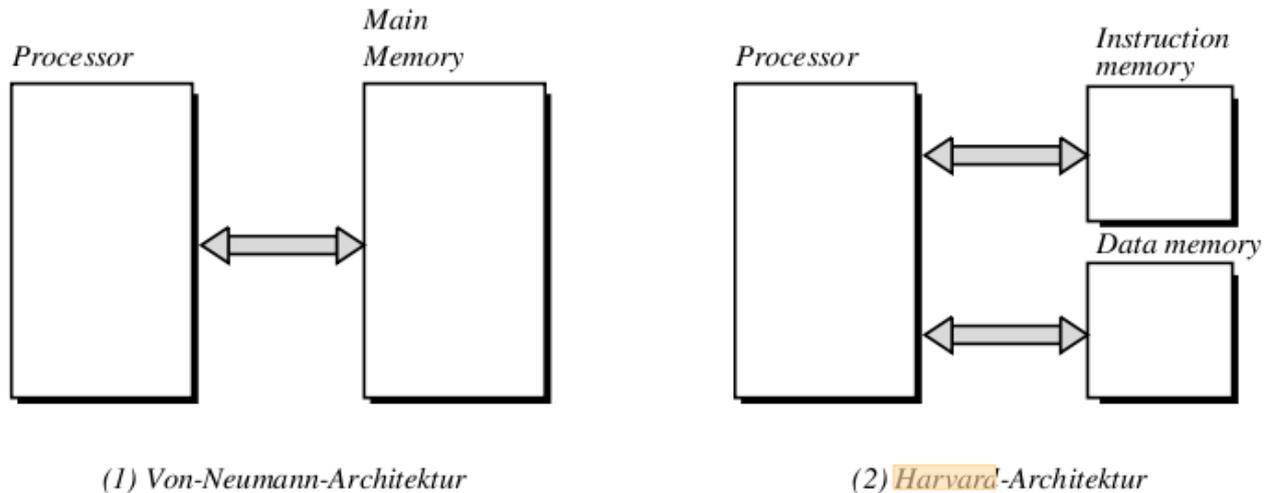


Abbildung 5.6: Harvard-Architektur

6 Punkte

18 Nennen Sie die Adressierungsarten die Sie kennen!

- Implied Mode – implizite Adressierung (zB enable interrupts)
- Register Mode – ($R3 \leftarrow R4 + R5$)
- Immediate Mode – ($R \leftarrow 0$)
- Direct Addressing Mode – ($R6 \leftarrow \text{memory}[(500)_{16}]$)
- Register Indirect Mode – ($R6 \leftarrow \text{memory}[R5]$)
- Program Counter Relative Adressing Mode
- Indirect Adressing Mode ($R6 \leftarrow \text{memory}[\text{memory}[(500)_{16}]]$)

S154

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

a) direkt: Der Wert im Operandenfeld entspricht der Speicheradresse des gewünschten Operanden.

b) indirekt: Der Zellenwert entspricht der effektiven Adresse des gesuchten Operanden.

$\text{memory}[\text{memory}[\text{adress}]]$

c) register-indirekt: Das Register, das im Befehl angegeben wird, enthält die Adresse des gewünschten Datenwortes. Register fungiert als Pointer

d) program-counter-relative: Die effektive Adresse erhält man durch Addition des im Befehl angegebenen Offsets mit dem Programmzählerstand. So ein Programm an beliebigen Stellen im Arbeitsspeicher arbeiten.

20 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. Beispielsweise alle geraden Adressen in der ersten Bank und die ungeraden in der zweiten Speicherbank. (=2-fach interleaved – geht auch vierfach)

Es wird hier das Pipelining-Prinzip 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).

21 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.

S171

22 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 geladen und auch in einen zweiten schnellen Zwischenspeicher (Pufferspeicher). Während er Prozessor weiterarbeitet, 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.

S172

23 DMA: Was es ist und warum es verwendet wird?

Direct Memory Access ermöglicht es, die Kommunikation zwischen dem Prozessor und den meist sehr viel langsameren peripheren Geräten zu beschleunigen. Er dient zur direkten Übertragung großer Datenmengen vom bzw. zum Speicher, ohne die CPU dabei dabei 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. Auch ist eine zusätzliche Kontrolleinheit nötig – der DMA Controller.

24 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 Interruptsignal. Die Daten werden also direkt zwischen I/O Device & Speicher ausgetauscht.

25 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)

26 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. (Mathematik-Co-, Graphik-Co-Prozessor)

27 Wozu dient ein virtueller Speicher

Bei einer großen Anzahl gleichzeitig speicherresistenter 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 der Externspeicher 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 Programmes speicherresistent.

28 Welchen Vorteil besitzt die „von Neumann“ und die „Harvard-Architektur“!

Von Neumannscher Flaschenhals: Daten, als auch die Programme befinden sich im Hauptspeicher.

Harvard	Von Neumann
getrennte Speicher / getrennte Busse: paralleler Instruction & Operand Fetch möglich	Programme & Daten werden in ein & demselben Speicher gehalten und über einen einzigen Datenbus übertragen.

29 Welchen Vorteil besitzt die „Harvard-Architektur“ gegenüber „von Neumannschen“, und wo wird dieser genutzt?

Instruction & Operand Fetch zur gleichen Zeit möglich – wird genutzt beim Instruction Pipelining.

30 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.

siehe auch Hazards beim Pipelining

31 Was sind Race Conditions?

Als Race Condition (Wettlaufsituation) werden Konstellationen bezeichnet, 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 äußerst schwierige und zeitraubende Angelegenheit.

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.

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 jedoch noch im READY Zustand ist, wird der Aufruf ignoriert und der P_SLEEP vollführt. Das eingetragene File wird erst gedruckt, wenn ein neuer Druckauftrag 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 andere Client setzt ebenfalls einen Druckauftrag 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

32 Was versteht man unter Trashing?

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

33 Nennen Sie alle Layer des ISO OSI Reference Models!

1. Physical Layer
2. Data-Link Layer
3. Network Layer
4. Transport Layer
5. Session Layer
6. Presentation Layer
7. Application Layer

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

Beim Linken werden Module zusammengefügt. Dadurch verändern sich die Adressen der Sprungziele. Bei direkter Adressierung muss der Linker eine Modifikation der Operanden durchführen, damit die Adresse ihr Sprungziel nicht verfehlt. Für die Code Segmente können Probleme dieser Art durch die Berücksichtigung der Konvention für Position Independent Code (PIC) gelöst werden. Dabei werden zB statt absoluter Sprungadressen nur die Distanzen (Displacements) von der aktuellen Adresse angegeben. Da sich durch die Verschiebung die Displacements nicht ändern, entfallen die Operanden-Modifikationen völlig.

35 Kann es zwischen Threads zu einem Deadlock kommen?

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 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 das Terminieren der beteiligten Prozesse aufgelöst. 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 vorrausschauend zu klären, ob die Zuteilung Deadlocks nach sich ziehen könnte. Es gibt Algorithmen dafür – diese liefern aber nur bei gewissen Zusatzinformationen sichere Zuteilungsentscheidungen

36 Welche Weiterentwicklung des RISC Architektur sind im Vorlesungsbuch beschrieben?

RISC – Reduced Instruction Set Computer

1. *SPARC Architektur (Scalable Prozessor Architecture)*

Konventioneller RISC Prozessor mit ausgefeilter Registertechnik. Offene Architektur, kann frei verwendet werden.

2. *MIPS Architektur (Microprocessor without Interlocking Stages)*

Performancesteigerung durch feinstufige Befehlspipeline und realisierte Speicherhierarchie

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

1. rein optisch

schreiben: ein dünner Film einer besonderen Metalllegierung wird mittels leistungsstarkem Laser zum Schmelzen gebracht (Übergang vom kristallinen zum amorphen Zustand)

lesen: beim Abtasten mit einem schwächeren Laser misst man das reflektierte Licht.

löschen: einheitlich erhitzen – Metalllegierung wird wieder kristallin

2. magneto-optisch

schreiben: Metalllegierung wird erhitzt – erleichtert Umpolung – Metall kühl rasch ab – Info bleibt

lesen: Laser wird je nach Magnetisierungsrichtung nach links oder rechts gedreht. (log. 0 oder 1)

löschen: allgemeine Erhitzung der gesamten Oberfläche – magnetische Bezirke werden gleich ausgerichtet.

38 Was ist ein endlicher deterministischer Automat?

endlich: Die Anzahl der Zustände des Automaten sind begrenzt.

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

39 Wann tritt ein Page-Fault auf?

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.

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

- *First In First Out (FIFO):* Jede Page bekommt zum Zeitpunkt Ihres Ladens einen Zeitstempel. Wenn ein Page-Fault 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.

41 Was versteht man unter atomic actions?

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.

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

BLOCKED - Zustand 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 anderen Prozess verursacht.

43 Was versteht man unter einem Semaphore?

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

44 Aus welchen Datenbereichen besteht die Datenstruktur des Semaphors?

Ein Semaphore besteht aus einem Zähler und einer Warteschlange.

45 Wie arbeiten Semaphore?

Semaphore 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.

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

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.

47 Was muss ein OS machen, wenn zwischen 2 Threads, die zum gleichen Prozess gehören, umgeschaltet wird?

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

zurückschreiben)

48 Was versteht man unter Flow Control?

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

49 Was versteht man unter bedingten bzw. unbedingten Sprüngen?

Sprünge können sowohl auf Register Transfer Ebene, als auch auf Maschinen Code Ebene erfolgen. Bei dieser Operation wird der MIC (Micro Instruction Counter) mit einem neuen Wert geladen.

Bei bedingten Sprüngen (Branch-Operation) kann zusätzlich eine Bedingung angegeben werden, die erfüllt sein muss, damit ein Sprung erfolgen kann.

50 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.

51 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. 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

52 Was sind Interrupts?

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.

53 Was versteht man unter einem Deadlock?

Darunter versteht man einen Zustand, in dem ein oder mehrere Prozesse auf Betriebsmittel warten, die dem Prozess selbst oder einem anderen beteiligten Prozess zugeteilt sind.

Ein Beispiel für einen Deadlock aus dem realen Leben ist eine Straßenkreuzung, an der von allen vier Seiten Autos gekommen sind und nun (die Regel *rechts vor links* vorausgesetzt) darauf warten, dass das jeweils rechts wartende Auto zuerst fährt.

Beispiel: Dining Philosophers Problem

54 Wann entstehen Deadlocks?

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.

55 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.

56 Welche Refresh Verfahren gibt es bei Speicherzellen?

● *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*

Bei diesem Verfahren synchronisiert man Refreshcontroller mit dem Prozessor 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.

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

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

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

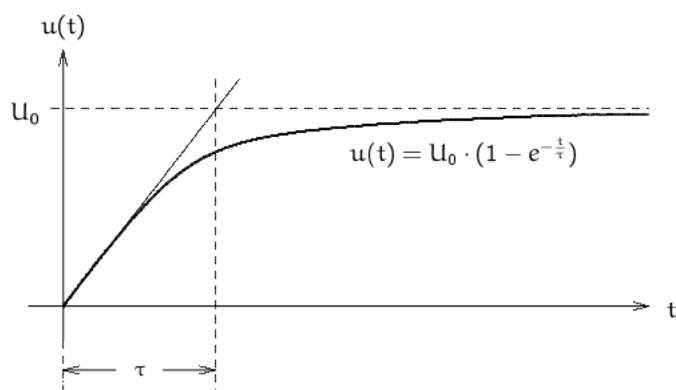


Abbildung 2.13: Spannungsverläufe bei kapazitiver Belastung (Aufladevorgang)

58 Welche Arten von Speicherbausteinen kennen Sie?

- *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.

oder

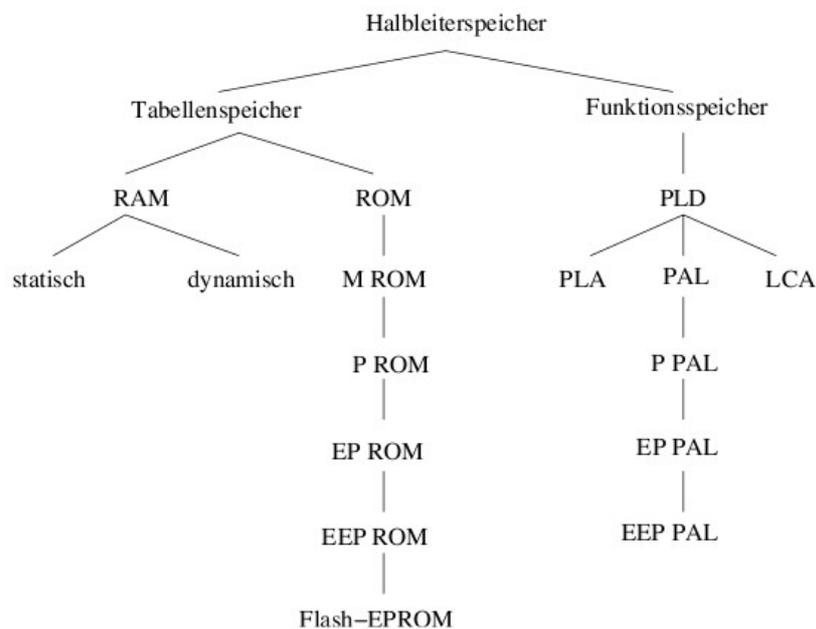


Abbildung 2.88: Übersicht über Halbleiterspeicher

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

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

60 Welche Funktion hat der ENABLE Eingang bei einem Decoder?

Der Enable Eingang 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.

61 Geben Sie die Übertragungskennlinie $U_a = f(U_e)$ eines Inverters an. Wo liegt die verbotene Zone?

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

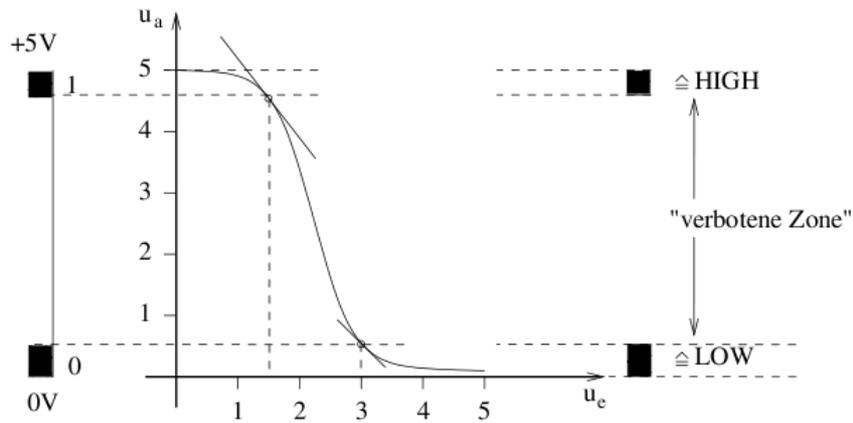


Abbildung 2.2: Übertragungskennlinie $u_a = f(u_e)$ eines Inverters

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

- Verhalten
- Struktur
- Geometrie

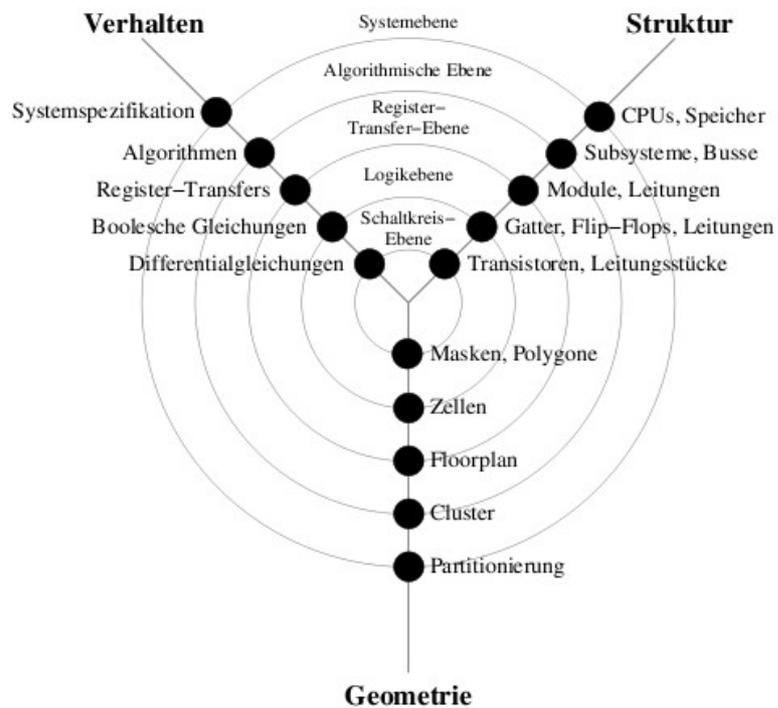


Abbildung 3.1: Y-Diagramm

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

- Schnittstellenbeschreibung
- Architektur
- Konfiguration

64 Von welchen zwei Faktoren hängen Zustandsänderungen bei Schaltwerken ab?

Von den *Eingangswerten* und dem *aktuellen Zustand* des Schaltwerks.

65 Wozu dient die Ziehkapazität bei einem Quarzoszillator?

Die Ziehkapazität eines Quarzoszillators dient dem Abgleich der Resonanzfrequenz bei Serienresonanz und wird mit dem Quarz in Serie geschaltet.

66 Was versteht man unter Fan-In?

Der Fan-In gibt die maximale Anzahl an Gatterausgängen an, die an den jeweiligen Gattereingang angeschlossen werden dürfen.

67 Was versteht man unter Fan-Out?

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

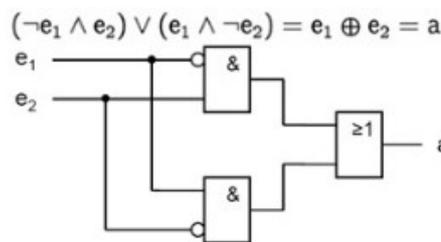
68 Zeichnen Sie eine Gatterschaltung für die XOR Funktion

$$0 \wedge 0 = 0$$

$$0 \wedge 1 = 1$$

$$1 \wedge 0 = 1$$

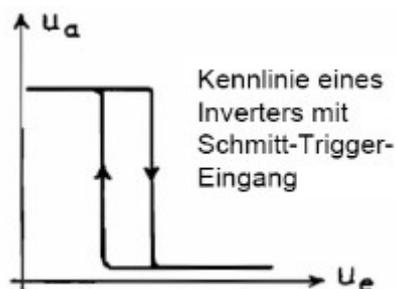
$$1 \wedge 1 = 0$$



69 Erklären Sie die Abkürzung ASIC!

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

70 Zeichnen Sie die Übertragungskennlinie eines invertierenden Schmitt Triggers!



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

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

72 Was versteht man unter dem Scratchpad eines Prozessors?

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.

73 Welchen Vorteil hat die Erhöhung der Baudichte bei Mikroprozessoren?

Reduzierung der Wegstrecken \rightarrow Erhöhung der Prozessorgeschwindigkeit, da sich elektrischer Strom mit

etwa 0,7-facher Lichtgeschwindigkeit ausbreitet.

74 Nennen Sie einen Vorteil, sowie einen Nachteil von Instruktionen mit variabler Länge im Vergleich zu Instruktionen mit fixer Länge.

- *Vorteil von variabler Länge*
Oft verwendete Befehle können mit kürzeren Bitmustern codiert werden, wodurch die Länge von Programmen reduziert wird.
- *Nachteil von variabler Länge*
Interpreter arbeitet langsamer, wenn er Maschineninstruktionen variabler Länge verarbeiten muss.

75 Welche 2 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.

76 Was ist ein Trap? Geben Sie ein Beispiel?

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, Memory-Management-Ereignisse, etc – sogenannte Traps aus.

77 Wo werden die Return Adressen bei Unterprozeduraufrufen gespeichert?

Die Rücksprungadressen werden im Stack gespeichert.

78 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 seiner Sprungzieladresse, damit gleich die Daten der Zieladresse dem Sprungbefehl in die Pipeline folgen können.

79 Was versteht man unter logischer Parallelität von Prozessoren?

Logische Parallelität von Prozessoren liegt dann vor, wenn mehrere Prozesse scheinbar gleichzeitig auf einem einzelnen Prozessor ausgeführt werden.

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

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)*

81 Nennen Sie zumindest 3 Bestandteile des Prozessdeskriptors!

1. Process ID

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

82 Durch welche 3 im Buch erläuterten Arten der Parallelverarbeitung kann die Performance eines Prozessors gesteigert werden?

1. Vektorverarbeitung
Zielt darauf ab, dass gleichartige Operationen nebeneinander berechnet werden können.
2. Superskalare Verarbeitung
bestehen aus mehreren skalaren Verarbeitungseinheiten, die mit einem ausgeklügelten Pipelining arbeiten.
3. Instruction Pipelining

83 Was ist die Aufgabe eines Linkers?

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

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

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

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

1. Fixe Zuordnung
2. Interruptvektor

86 Welche 2 Komponenten benötigt jeder Thread für sich alleine?

1. Registersatz des Prozessors
2. Thread-Spezifische Daten (lokale Variablen, Rücksprungadressen, etc)

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

1. Mehrere Threads in einem Prozess verwenden denselben Adressraum. Ein Prozess kann daher durch Speicheroperationen 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.

88 Was versteht man unter einem System Call?

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

Beispiel: Erzeugung eines Prozesses

89 Was versteht man unter Scheduling?

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

90 Job Scheduling

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.

91 Prozess Scheduling

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.

92 Forderungen an Scheduling Algorithmen

- 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.

93 Scheduling Methoden

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 – wird 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 READY-LIST 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

Eine adaptive Scheduling Strategie. Durch dynamische Änderung von Prioritäten wird sichergestellt, dass kein Prozess auf der Strecke bleibt. Dies kann zB durch Zuteilung eines Soll-Services passieren.

Weitere: *Shortest Job First*

94 Thread Scheduling

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

95 Message Passing

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)

96 Was versteht man unter einem Hazard?

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

97 Welche Arten von Pipeline Hazards gibt es?

- *Strukturelle Hazards*: mehrerer Pipeline Stufen benötigen dieselbe Ressource
=> geeignete Struktur
- *Control Hazard*: nachfolgende Befehle hängen vom Ausgang des Sprungs ab
=> Prediction Branch, Delayed Branch
- *Data Hazard*: Berechnung erfordert Ergebnis des Vorgängerbefehls
=> Forwarding (direkte Übergabe an das Rechenregister), Code-Optimierung

Jeder fünfte bis zehnte Befehl ist eine Sprunganweisung.

98 Halbaddierer

Zählt 2 einstellige Binärzahlen zusammen. Mögliche Überträge aus der Vorstelle werden nicht berücksichtigt.

99 Volladdierer

Addiert Binärzahlen mit n Stellen unter Berücksichtigung des Übertrags.

100 RS Latch

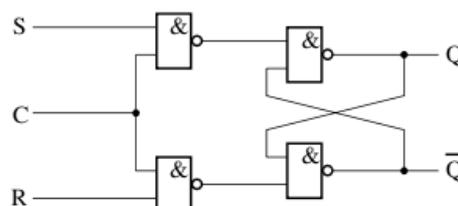


Abbildung 2.36: RS-Latch mit Kontrolleingang C

101 Codierer

Die Schaltung eines Codierers besitzt bei n Eingängen genau $m = \lg(n)$ 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: e7 aktiv am Eingang ergibt 111 am Ausgang.

102 Multiplexer

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.

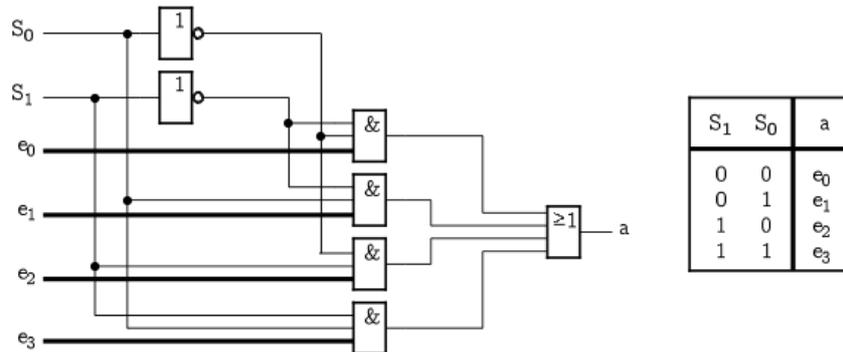


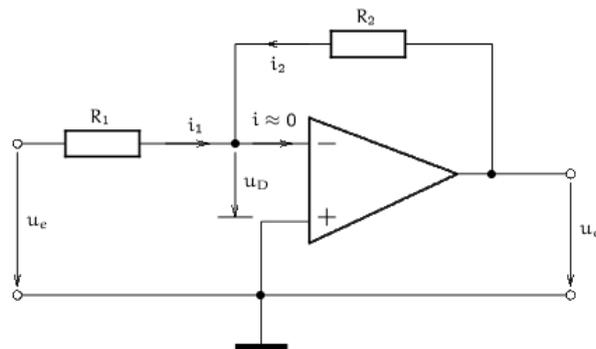
Abbildung 2.30: (4 zu 1)-Multiplexer

103 Demultiplexer

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

104 Operationsverstärker

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*)

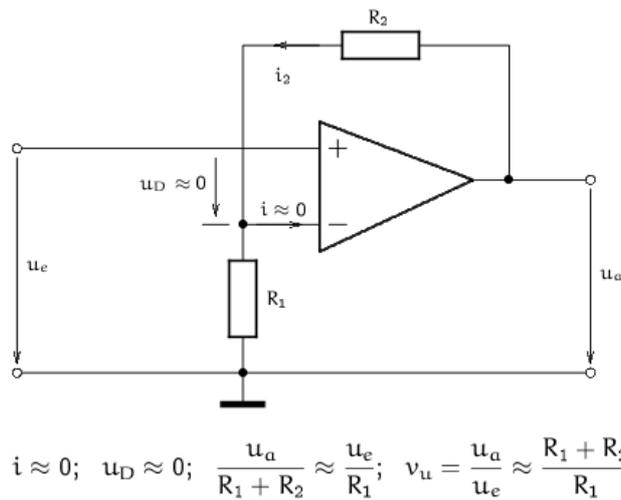


Abbildung 2.52: Nicht-invertierender Operationsverstärker

105 Komparatoren für analoge Signale

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.

106 Komparatoren für digitale Signale

In der Digitaltechnik werden häufig Vergleiche 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.

107 Torschaltungen

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.

108 Fensterdiskriminator

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.

109 Univibrator

Ein Univibrator stellt eine Schaltung dar, 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.

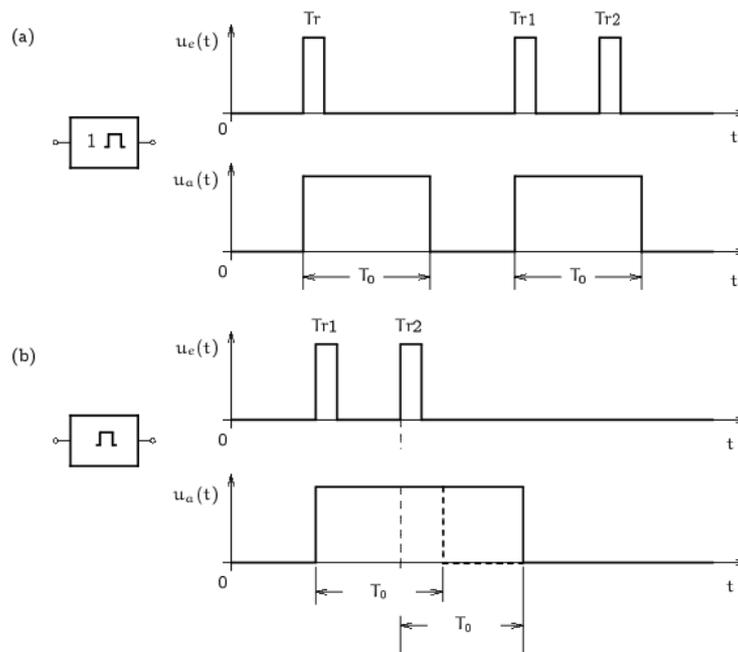


Abbildung 2.68: Schaltzeichen für Univibratoren und Impulsdiagramme nicht-nachtriggerbarer (a) und nachtriggerbarer (b) Univibrator; T_r =Triggerimpuls

110 Quarzoszillator

Ein Quarzoszillator ist eine elektronische Schaltung zum Erzeugen von Schwingungen, die sich eines Schwingquarzes als frequenzbestimmendes Bauelement bedient. 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.

111 Sägezahngenerator

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.

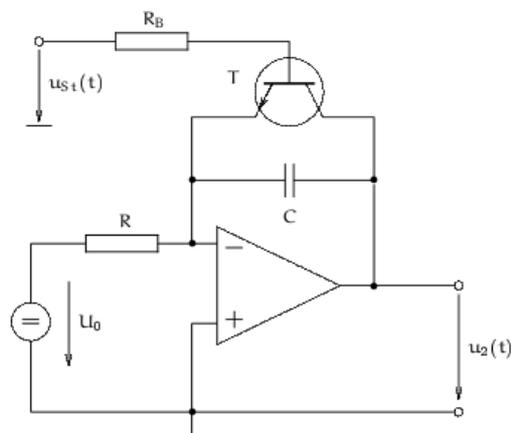


Abbildung 2.75: Sägezahngenerator mit Miller-Integrator und Schalttransistor T

112 Tristate Output

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.

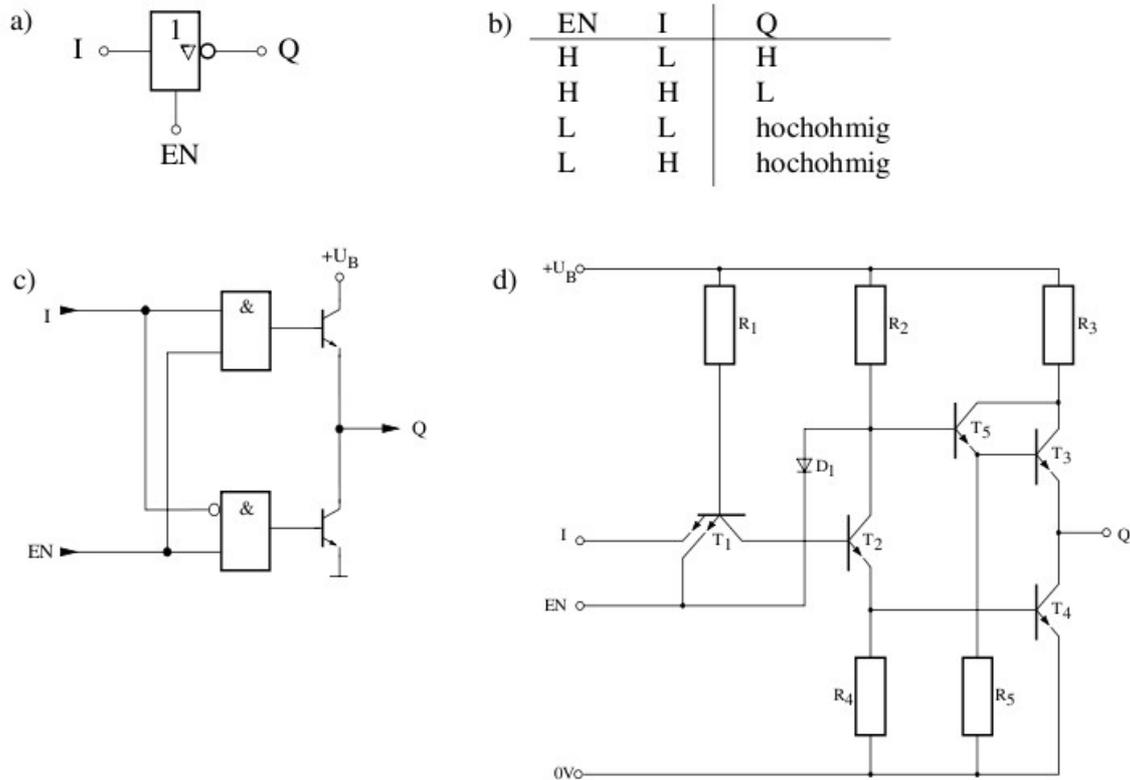


Abbildung 2.84: Tristate Outputs

S60

113 Open-Collector Schaltungen

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 Emittor 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 Kollektorwiderstand beschalten.

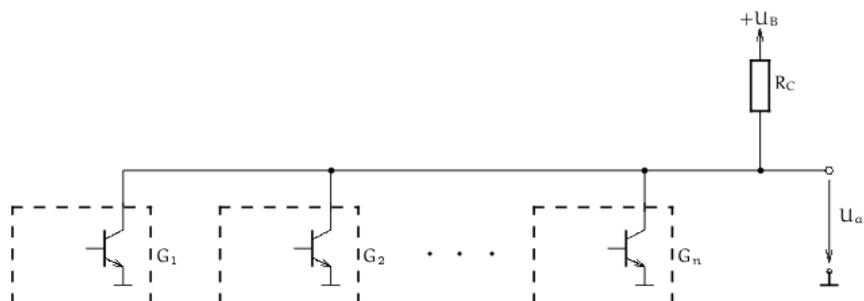


Abbildung 2.85: Zusammenschaltung von Gatterausgängen mit offenem Kollektor

Wired-AND, Wired-OR

S61

114 Moore Schaltwerk

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 Taktrate erfolgen kann.

115 Mealy Schaltwerk

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.

116 Prozessor

Ein Prozessor ist ein Schaltwerk zur Verarbeitung von Daten, das aus Rechenwerk und Leitwerk besteht.

117 Rechenwerk

Funktionseinheit innerhalb eines digitalen Rechensystems, die Rechenoperationen ausführt.

118 Leitwerk

Funktionseinheit eines digitalen Rechensystems, die

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

119 Woraus ergibt sich die Performance eines Caches

- Hit Time
- Hit Rate
- Miss Penalty

120 Was versteht man unter Swapping?

Beim Swapping (auch Roll-In/Roll-Out) werden Prozess-Images 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 Speicherverschwendung durch die Fragmentierung und die etwas unflexible Memory Protection.

121 Was versteht man unter Paging?

Aufteilung des virtuellen Adressraums in gleich große Teile, die vom Programm unabhängig sind. Die Aufteilung erfolgt transparent.

Umsetzung virtueller in physikalische Adressen:

$(001DA472)_{16}$ virtuelle Adresse
8KB Pagegröße entspricht $(2000)_{16}$

$$\text{Page Nummer} = \frac{\text{virtuelle Adresse}}{\text{Pagegröße}} = \frac{1DA472}{2000} = (ED)_{16}$$

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

$$\text{Startadresse} = \text{Page Nummer} * \text{Pagegröße} = ED * 2000 = (1DA000)_{16}$$

$$\text{Offset} = \text{virtuelle Adresse} - \text{Startadresse} = 1DA472 - 1DA000 = (472)_{16}$$

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

122 Was versteht man unter Segmentierung?

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

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

Ein Cache ist ein 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)

124 Welche Arten von Caches kennen Sie?

- *Voll Assoziativ*
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.

125 Was versteht man unter VHDL?

Die Very (High Speed Integrated Circuit) Hardware Description Language ist eine Sprache für Hardware-Design.

126 Welche Entwurfsansichten gibt es in VHDL?

- Algorithmische Ebene
- Register-Transfer Ebene
- Logikebene

127 Welche Schaltkreisfamilien kennen Sie?

- *TTL – Transistor-Transistor-Logik*
großer Fan-Out / wegen hoher Gatterlaufzeiten (propagation delay) heute kaum noch verwendet.
- *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 Schal-

tungen

- *CMOS – Complementary MOS*
geringste Leistungsaufnahme / großer Betriebsspannungsbereich / Standard im PC Bereich

128 Erklären Sie die Abkürzung VLSI!

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

129 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.

130 Geben Sie ein Beispiel für asynchrone IPC-Methoden, die unter Unix und auch in MS Windows realisiert sind.

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

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

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.

132 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.

```
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;
5. R11 <- R3+R4;        # 16+1=17
```

133 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.

134 EEPROM: Nachteil im Vergleich zum RAM

Beim EEPROM dauern die Zugriffszeiten länger und die Speicherstellen sind nur begrenzt beschreibbar (maximal 10^6 mal) RAMs sind permanent löscher & wiederbeschreibbar.

Die Nichtflüchtigkeit des EEPROMs stellt wiederum einen Vorteil gegenüber den RAMs dar.

135 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.

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

In den Prozessdeskriptor.

137 Bestimmen Sie die Funktion der folgenden Schaltungen.

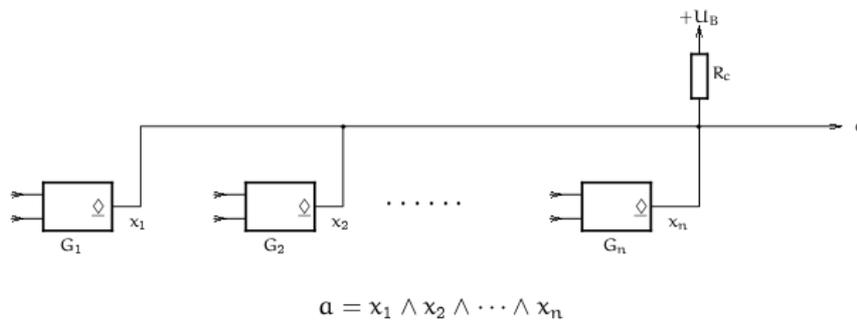


Abbildung 2.86: Wired-AND-Verknüpfung (\diamond =Symbol für Open-Collector-Ausgang)

=> WIRED AND

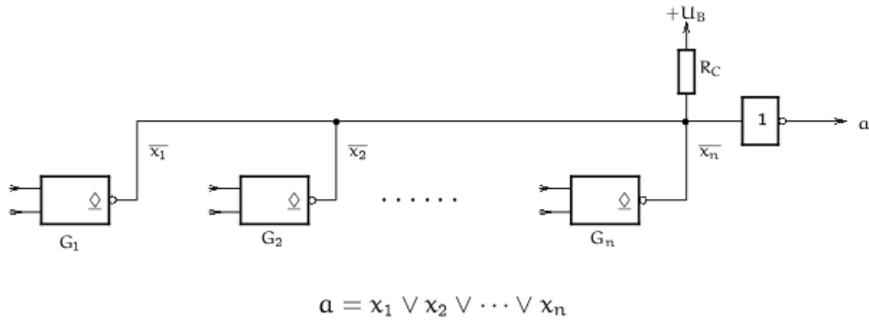


Abbildung 2.87: Wired-OR-Verknüpfung (\diamond =Symbol für Open-Collector-Ausgang)

=> WIRED OR

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

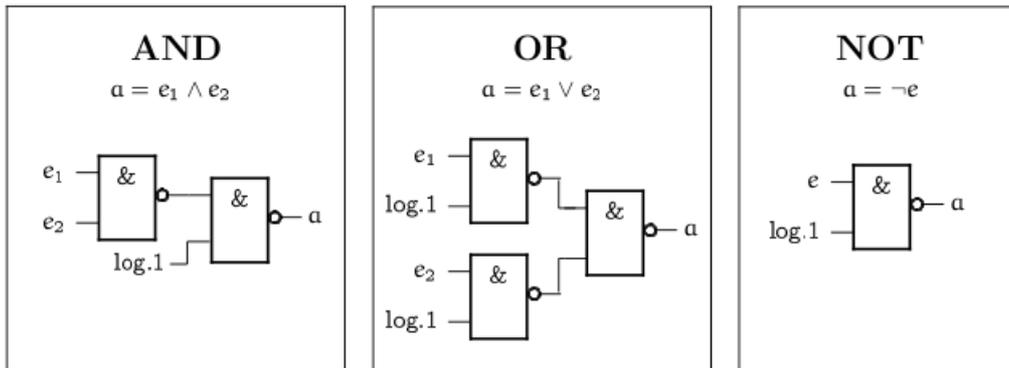


Abbildung 2.7: Boolesche Operatoren aus NAND-Gattern aufgebaut

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

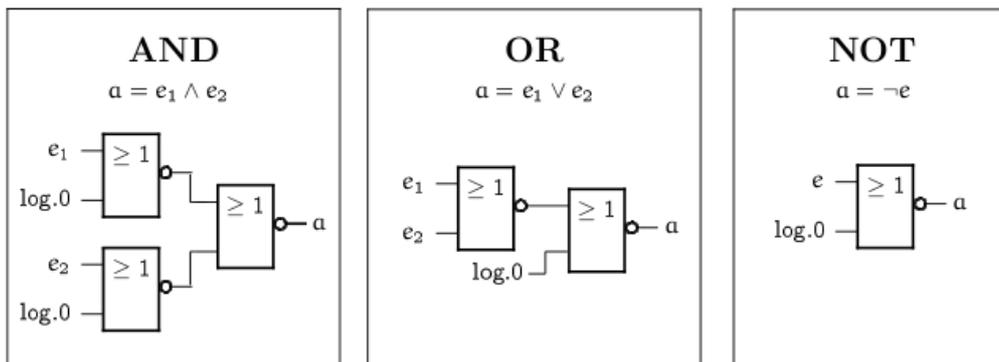


Abbildung 2.6: Boolesche Operatoren aus NOR-Gattern aufgebaut

140 Nennen Sie einen Nachteil eines asynchronen Zählers!

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

141 In welchem Zahlensystem zählt der synchrone Zähler?

Im binären Zahlensystem.

142 Geben Sie 3 Merkmale für die Leistung eines Betriebssystems an!

- Effizienz
- Mensch-Maschine Schnittstelle
- Aspekte zur Weiterentwicklung des Betriebssystems

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

- 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.

144 Was versteht man unter Trusted Computing?

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.

145 Zählen Sie grundlegende Sicherheitsanforderungen auf!

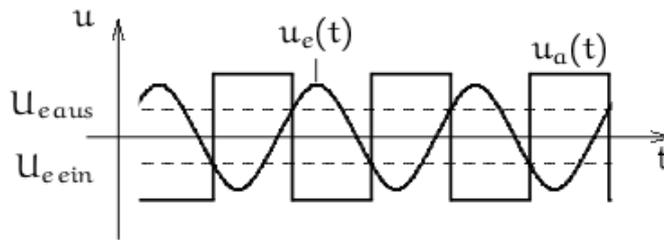
- *Secrecy (Geheimhaltung)* – Forderung nach Geheimhaltung von Daten. (nur autorisierte Personen)
- *Integrity (Integrität)* – Daten dürfen nur von autorisierten Benutzern verändert werden.
- *Verfügbarkeit (Availability)* – alle Teile eines Computersystems können von autorisierten Benutzern verwendet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

146 Beschreiben Sie die 4 Bedrohungsklassen!

- 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 Dateinhalten, etc
- Fabrication (Fälschung) – gefälschte Daten gelangen in den Informationsfluss, zB Einfügen von zusätzlichen Einträgen in Dateien (Passwortdateien)

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

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



148 Handelt es sich beim Round Robin Scheduling um ein preemptives Verfahren?

Nein, das RRS Verfahren ist nicht preemptiv. Ein laufender Prozess bleibt immer für einen ganzen Zeitzyklus im Zustand RUNNING.

149 Was ist ein SSID?

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

150 Nennen Sie 2 Vorteile von IPv6 gegenüber IPv4!

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

151 Welche Aufgaben besitzt ein Betriebssystem?

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

152 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 (ent-rauscht 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.

153 Wozu dient die Architektur in VHDL?

Die Architektur beschreibt die Funktionalität des Moduls entweder als Verhaltensbeschreibung oder als Netz-liste.