

Lösungen zu den Aufgaben und Testfragen

Zu Kapitel 1: Einführung

1. Mechanische Rechner ab dem 17. Jahrhundert, Röhrenrechner ab Mitte der 40er Jahre, Transistorrechner ab 1961 und Mikroprozessoren ab 1969.
2. Ansteuerung der übrigen Funktionseinheiten und Verarbeitung der Daten.
3. Signalprozessoren, Mikrocontroller, Arithmetik-Prozessoren, Kryptographieprozessoren.
4. Ein Harvardrechner kann gleichzeitig auf Daten und Befehle zugreifen, ist aber aufwändiger im Aufbau.
5. Nach dem Mooreschen Gesetz ergibt sich in drei Jahren eine Vervierfachung der Transistorzahl und somit eine Halbierung der Strukturbreite; in zwölf Jahren wäre demnach die Strukturbreite also auf $1/16$ also ca. 6 nm abgesunken. Der Abstand der Atome im Festkörper liegt aber in der Größenordnung von 0.5 nm und bei Strukturen aus wenigen Atomlagen treten quantenphysikalische Effekte auf; die Prognose für 2027 kann nicht mit dem Mooreschen Gesetz gemacht werden!

Zu Kapitel 2: Informationseinheiten und Informationsdarstellung

1. 8 Bit, 64 Bit
2. MSB = Bit 31
3. 64 KByte, 16 MByte, 4 Gbyte
4. a) 46E7h, 18151d b) 011111010010b, 2002d c) 5BA0h, 0101101110100000b
5. Bei a) und c) das Übertragsbit (Carryflag), bei b) und d) das Überlaufbit (Overflow)
6. Invertieren und Inkrementieren ergibt 1111101 und 10100111; -160 liegt bei 8-Bit-Zahlen außerhalb des darstellbaren Bereiches (-128 ... +127).

7. Die Invertierung einer Binärziffer kann durch $\bar{a}_i = (1 - a_i)$ beschrieben werden. Aus Gl. 2.3 ergibt sich damit für \bar{Z}

$$\begin{aligned}\bar{Z} &= \sum_{i=0}^{n-1} \bar{a}_i \cdot 2^i = -(1 - a_{n-1}) \cdot 2^{n-1} + \sum_{i=0}^{n-2} (1 - a_i) \cdot 2^i \\ &= -2^{n-1} + a_{n-1} \cdot 2^{n-1} + (2^{n-1} - 1) - \sum_{i=0}^{n-2} a_i \cdot 2^i \\ &= a_{n-1} \cdot 2^{n-1} - \sum_{i=0}^{n-2} a_i \cdot 2^i - 1 = -Z - 1\end{aligned}$$

Das entspricht Gl. 2.4.

8. 0 ... 01001011b
9. 1.125

Zu Kapitel 3: Halbleiterbauelemente

1. Von den beiden Dioden ist die obere in Durchlassrichtung gepolt, die untere in Sperrrichtung. Zwischen den Dioden liegt also +5V. Dadurch wird T1 gesperrt und T2 leitend. Am Ausgang ergibt sich also eine Spannung von annähernd 0V.
2. Drain, Source und Gate, Ansteuerung erfolgt über das Gate.
3. Der Kollektorstrom beträgt ein Vielfaches des Basisstromes, z.B. 100 mal so viel.
4. Aus dem Vergleich von Abb. 3.3 mit Abb. 3.5 ergibt sich: CMOS-Pegel können ohne Probleme in TTL-Eingänge gespeist werden; TTL-Pegel können nicht ohne weiteres in CMOS-Eingänge gespeist werden, weil bei TTL-Ausgängen der HIGH-Pegel nur 2.4V garantiert, der CMOS-Eingang aber mindestens 3.5V braucht.

Zu Kapitel 4: Speicherbausteine

1. a) Masken-ROM, b) 4 Wort- und 8 Bitleitungen, c) 5 Adressbits, davon 2 für den Zeilendekoder und 3 für den Spaltendekoder, d) 32 Bit, e) 0,0,1,0
2. a) MROM, b) SRAM, c) PROM, d) EPROM oder EEPROM.
3. Größer, die Ausgänge Q und \bar{Q} würden sonst nie LOW-Pegel erreichen.
4. Mit Dielektrika, durch dreidimensionale Gestaltung und durch Faltung.
5. Jede Zelle enthält 4 oder 6 Transistoren, das ergibt einen viel größeren Stromverbrauch als bei DRAMs.
6. Ohne Adressmultiplexing 22 Anschlüsse, mit symmetrischer 11/11-Adressierung nur noch 11.

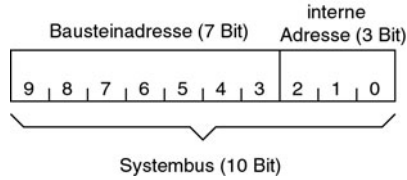
7. 2048×1024 ergibt 2M Adressen zu je 4 Bit, also 8 MBit = 1 MByte. Die Adressierung ist 11/10. Nach 64 ns muss jede Zelle wieder einen Refresh erhalten, bei 2048 Zeilen muss in Abständen von 31 μ s ein Zeilenrefresh stattfinden.
8. Ein 9/11-DRAM hat 512 Zeilen und 2048 Spalten. Bei jedem Schreib- und Lesevorgang werden also 2048 Schreib-/Leseverstärker aktiviert, innerhalb der Retention-Zeit müssen 512 Zeilen einen Refresh erhalten. Ein 11/9-DRAM hat 2048 Zeilen und 512 Spalten. Es gibt nur 512 Schreib-/Leseverstärker, der Baustein braucht deshalb weniger Strom; wegen der 2048 Zeilen muss aber vier mal so oft ein Zeilenrefresh stattfinden.
9. Das Auslesen eines ganzen Blockes von Daten aus dem Hauptspeicher nach Übermittlung der Startadresse.
10. Beim Auslesen aus einer Bank bleiben die anderen Bänke betriebsbereit. Dadurch kommt man beim Wechsel auf eine neue Speicheradresse mit einer gewissen Wahrscheinlichkeit auf eine betriebsbereite Bank, die durchschnittliche Zugriffszeit wird verringert.
11. Bei 133 MHz dauert jeder Bustakt 7.5 ns. DDR-SDRAM überträgt mit jedem Takt zwei Worte, hier also 128 Bit d.h. 16 Byte. Für 32 Byte werden 2 Takte, d.h. 15 ns, gebraucht.
12. Der Hauptvorteil ist die Nicht-Flüchtigkeit, das Funktionsprinzip ist in Abschn. 4.4 erklärt.

Zu Kapitel 5: Ein- und Ausgabe

1. Es wird ein Ausgabebaustein gebraucht, dieser enthält ein Flipflop für jede Ausgangsleitung.
2. Steuerwort 99h.
3. Alle Ports arbeiten im Modus 0, Port A: Ausgabe, Ports B und C: Eingabe.

Zu Kapitel 6: Systembus und Adressverwaltung

1. Der Zustand der Busleitung (HIGH/LOW) wird nur durch einen Baustein bestimmt, alle anderen Bausteine dürfen den Leitungszustand nicht beeinflussen. Die Ausgänge müssen daher über einen hochohmigen Zustand verfügen. Man muss außerdem die Lastfaktoren der Busteilnehmer beachten.
2. LOW, HIGH, hochohmig
3. Ein Open-Collector-Ausgang kann eine Leitung in den LOW-Zustand bringen, aber nicht in den HIGH-Zustand. Dies muss durch den externen Pull-Up-Widerstand übernommen werden.
4. Ein bidirektionaler Bustreiber ist in Abb. 6.7 dargestellt.
5. Der Baustein belegt die acht Systemadressen von 240h bis 247h, dabei ist 240h die Basisadresse. Das Adress-Aufteilungswort ist wie folgt:



6. a) 4 MByte, b) 32 Chips, c) 4 Bänke, d) 128 MByte, e) 512 MByte, f) 8000000h – FFFFFFFh.
7. Chip 3 auf Modul 1.
8. a) Ein FSB1066 überträgt 1066 Megabit/s pro Leitung; bei QDR werden mit jedem Takt 4 Bit übertragen, daher $f=1066/4=266$ MHz.
b) Bei 64 Leitungen ergeben sich $64 \text{ mal } 1066 = 68224$ Megabit/s oder $68224/8 = 8528$ Megabyte/s.
c) CPU-Takt/ Bustakt = $2660 \text{ MHz} / 266 \text{ MHz} = 10$ (Taktmultiplikator)
9. $2964d = 0B94h$; nach Vertauschung der Bytes erhält man $940Bh = 37899d$.

Zu Kapitel 7: Einfache Prozessoren

1. 13 Byte
2. Die Aktivierung der Systembus-Schnittstelle und der Adressdekoder entfällt; oft ist auch der Prozessor höher getaktet als der Speicherbus.
3. Die Abfolge der Steuersignale ist ungefähr:
 1. Programmzähler auf den Adressbus legen.
 2. Aktivierung der externen Steuerleitungen für Lesezugriff im Speicher.
 3. Opcode von Datenbus entnehmen und im Befehlsregister einspeichern.
 4. Dekodierung des Opcodes.
 5. Programmzähler inkrementieren.
 6. Programmzähler auf den Adressbus legen.
 7. Aktivierung der externen Steuerleitungen für Lesezugriff im Speicher.
 8. Adress-Operand von Datenbus entnehmen und in Adresspuffer-Register legen.
 9. Adresspuffer-Register auf Adressbus legen.
 10. Aktivierung der externen Steuerleitungen für Lesezugriff im Speicher.
 11. Datenwort von Datenbus entnehmen und auf internen Bus legen.
 12. Einspeicherimpuls für Register 1 erzeugen, Datenwort wird vom internen Bus übernommen.
 13. Programmzähler inkrementieren.
 Dieser Befehlszyklus enthält insgesamt drei Buszyklen.
4. $15 \cdot 0.5 \text{ ns} = 7.5 \text{ ns}$.
5. a) $M=L, \overline{C_n}=H, S_3=H, S_2=L, S_1=L, S_0=H$,
b) $M=H, S_3=H, S_2=H, S_1=H, S_0=L$,
c) $M=H, S_3=L, S_2=H, S_1=L, S_0=H$.

6. 90h
7. 72h
8. Siehe S. 101.
9. Siehe S. 102.
10. Dadurch beginnt der Prozessor erst zu arbeiten, wenn alle anderen Bausteine stabil laufen.

Zu Kapitel 8: Besondere Betriebsarten

1. Wenn externe Geräte Daten liefern, muss Datenverlust durch Pufferüberlauf vermieden werden.
2. Maskierung, Registrierung, Vektorisierung und Priorisierung der Interrupt-Anforderungen, Abwicklung des Interrupt-Anforderungs- und Bestätigungszyklus mit dem Prozessor und Verwaltung des Systemzustands.
3. Es erfolgt keine Zwischenspeicherung im DMA-Controller. Der DMA-Controller sorgt nur dafür, dass eine Komponente sendet und eine andere Komponente die Daten direkt vom Bus abnimmt.

Zu Kapitel 9: Beispielarchitekturen

1. Es ist A=10h
2.


```

      CLRA      ; A=0, Z=1
      ADD #80  ; A=50h, kein Flag
      ADD #80  ; A=A0h, V=1, N=1 da Überlauf und negativ
      ADD #80  ; A=F0h, V=0, N=1 da negativ aber kein neuer Überlauf
      ADD #32  ; A=10h, rechnerisches Ergebnis 272 nicht mehr darstellbar,
                ; da größer als 255, deshalb Übertrag:
                ; A=16=10h, C=1
      SUB #16  ; A=0, Z=1
                ; nicht genannte Flags = 0.
      
```
3. Der Stackaufbau nach PSHH ist


```

      Adresse 14F: 07h
      Adresse 14E: 08h
      Adresse 14D: 09h
      Die Registerinhalte nach letztem Befehl sind: A=09, H=09, X=07
      
```
4. Die Schleife wird 8 mal durchlaufen und der Inhalt des Akkumulators ist A=18h (24d).
5. Der Inhalt der Speicherzellen zwischen 40h und 50h ist in der folgenden Tabelle aufgeführt; alle anderen Speicherzellen bleiben unverändert.

Adresse	Inhalt	Adresse	Inhalt
42h	0Ah	46h	0Eh
43h	0Bh	47h	0Fh
44h	0Ch	48h	10h
45h	0Dh	49h	11h

6. Der Grund liegt im Stackaufbau: Der Stackpointer verweist bei der CPU08 immer auf den ersten freien Platz unterhalb des Stacks. Eine unveränderte Übertragung des Stackpointers in das Zeigerregister H:X ergäbe also einen ungültigen Zeiger. Umgekehrt würde eine unveränderte Übertragung des H:X-Registers den Stackpointer auf eine schon belegte Stackadresse verweisen lassen mit der Gefahr des Datenverlustes.
7. Durch INCX wird nur X verändert, ein eventueller Übertrag wird nicht an H weitergegeben, H:X arbeitet also nicht als 16-Bit Zeiger.
8. Set Carry Bit
9. Weil POP ein Autoinkrement braucht, das im Adresswerk ohnehin verfügbar ist; PUSH braucht aber ein Autodekrement, das in den elementaren Transportbefehlen nicht verfügbar ist.

Zu Kapitel 10: Speicherverwaltung

1. a) 000B05h wird unterteilt in: Virtuelle Seite 1, Offset 305h. Virtuelle Seite 1 wird laut Seitentabelle abgebildet auf physikalische Seite 2, diese ist präsent, es wird kein Seitenfehler ausgelöst. Physikalische Adresse ist 1305h. Umsetzung stimmt mit Abb. 10.2 überein.
b) 001440h wird unterteilt in: Virtuelle Seite 2, Offset 440h. Virtuelle Seite 2 wird abgebildet auf physikalische Seite 14, diese ist nicht präsent. Es wird ein Seitenfehler ausgelöst, die Behandlungsroutine lädt die Seite und trägt die neue physikalische Seitenadresse in der Seitentabelle ein. Daraus ergibt sich die neue physikalische Adresse.
2. $0.7 \cdot 1 + 0.3 \cdot 8 = 3.1$ Wartetakte; bei 5 ns Taktzykluszeit entspricht das 15.3 ns
3. Die Wortadresse in der Cache-Line ist in allen Fällen 4, d.h. die Bytes 4–7 werden adressiert.
 - a) Zeilennummer unvorhersehbar, da nur durch Ersetzungsstrategie bestimmt; Tag ist 0000 0000 0010 1010 0010 0101 0110b.
 - b) Zeile 86 da Index=010110b, Tag ist 0 0000 0000 0101 0100 0100b.
 - c) Satz 22 da Index=010110b, je nach Vorgeschichte auf Weg 0 oder Weg 1; Tag ist 00 0000 0000 1010 1000 1001b.
4. Die Kapazität des Caches ist $128 \cdot 16 = 2048$ Byte. Ein Datenbereich von 1 KByte = 1024 Byte umfasst 64 Blöcke zu je 16 Byte. Es werden beim ersten Durchgang 64 aufeinanderfolgende Cache-Zeilen gefüllt, weil sich nach jedem 16-Byte-Block der Index um eins erhöht. Ab dem zweiten Durchgang hat man also 100% Trefferquote. Für einen Bereich von 2 KByte gilt das gleiche, hier werden alle 128 Cache-Zeilen geladen. Werden

4 KByte sequenziell durchadressiert, so werden die ersten 2048 Adressen sequenziell im Cache abgelegt und füllen sequenziell alle 128 Zeilen. Der 129-te Block überschreibt den ersten Block, weil der 7-Bit-Index wieder den Anfangswert hat. Der 130-te Block überschreibt den zweiten Block im Cache usw. Die zweite Hälfte des 4 KByte-Bereiches überschreibt vollständig die erste Hälfte. Beim zweiten und allen folgenden Durchgängen hat man bei jedem Wechsel auf eine andere Cache-Line einen Fehltreffer.

5. Vorüberlegung: Die Kapazität des Caches ist $128 \cdot 16 = 2048$ Byte, die aufzunehmende Datenmenge ist 2560 Byte.
 - a) und c) LRU- und FIFO-Ersetzung: Treffer nur innerhalb neu geladener Cache-Zeilen, weil die Cache-Zeilen vor einem Folgezugriff überschrieben werden.
 - b) Bei LFU-Ersetzung kommt es auf die Vorgeschichte und den Zeitraum der Erfassung an, hier ist keine Aussage möglich.
 - c) Bei Random-Ersetzung hält der Cache immer 80% der relevanten Cache-Zeilen mit rein zufälliger Verteilung, die Trefferquote ist daher deutlich über 80%.

6. a) Der vollassoziative Cache bringt bis zu 64 Blöcke unabhängig von der Adresse im Cache unter. Da in der Programmsequenz nur 4 Blöcke angesprochen werden, hat er ab dem zweiten Durchgang 100% Trefferquote.
 - b-d) Für die anderen Fälle muss der Index bestimmt werden, dieser liegt wie folgt: b) Bits 9 - 4, c) Bits 8 - 4, d) Bits 7 - 4. Daraus ergibt sich: Die angegebenen Adressen werden immer auf Index 9 abgebildet. Die Ergebnisse für die beiden ersten Durchgänge sind in der folgenden Tabelle gezeigt. Es bedeutet T: Treffer, F: Fehltreffer; 0,1,2,3: Wege; T0: Treffer in Weg 0 usw.

Adresse	b) Direkt abbildend		c) Zweifach assoziativ		d) Vierfach assoziativ	
	1.Durchg.	2.Durchg.	1.Durchg.	2.Durchg.	1.Durchg.	2.Durchg.
00002091h	F	F	F0	F0	F0	T0
00005492h	F	F	F1	F1	F1	T1
00002093h	F	F	T0	T0	T0	T0
00005494h	F	F	T1	T1	T0	T1
0000109Ah	F	F	F0	F0	F2	T2
0000F49Bh	F	F	F1	F1	F3	T3
0000109Ch	F	F	T0	T0	T2	T2
0000F49Dh	F	F	T1	T1	T3	T3

Aus dieser Tabelle liest man ab: Trefferrate ab dem zweiten Durchgang bei b) 0%, bei c) 50% und bei d) 100%. Die Kapazität der Caches ist in allen Fällen 1024 Byte.

7. a) Selektor in DS ist 0017h, d.h. Index=2, TI=1, RPL=3. Da RPL=CPL, ist die Benutzung des Selektors erlaubt und EPL=3. Es erfolgt ein Zugriff auf Deskriptor 2 (Index) der LDT (TI=1). Der Deskriptor enthält folgende Informationen: Basis=028000h, Limit=200h, P=1 (Segment präsent), DPL=3, S=1 (Anwendungssegment), E=0 (Datensegment), E/C=0 kein expand down, W=1 (Segment beschreibbar), A=1 (Segment wurde schon benutzt). Der Offset 1A0h ist kleiner als das Limit 200h und es ist DPL=EPL. Der Zugriff ist erlaubt, die physikalische Adresse ist $00028000h + 000001A0h = 000281A0h$.

- b) Selektor in SS ist 0007h, d.h. Index=0, TI=1, RPL=3. Da RPL=CPL, ist die Benutzung des Selektors erlaubt und EPL=3. Es erfolgt Zugriff auf Deskriptor 0 der LDT. Dieser enthält die Informationen Basis=A100h und Limit=10800h. Die Zugriffsrechte des Selektors stehen dem Zugriff nicht entgegen, aber der Offset 1F000h ist größer als das Limit, es tritt eine Schutzverletzung auf, Abbruch des Zugriffs.
- c) Selektor in ES ist 000Fh, d.h. Index=1, TI=1, RPL=3. Da RPL=CPL, ist die Benutzung des Selektors erlaubt und EPL=3. Es erfolgt Zugriff auf Deskriptor 0 der LDT. Dessen DPL ist 0 (höchste Privilegstufe), daher reicht EPL=3 nicht für einen Zugriff aus, es tritt eine Schutzverletzung auf.
8. a) 40 (inc eax) b) 46 (inc esi)
 c) FF 03 (inc dword ptr [ebx]) d) FF 86 00 02 00 00 (inc dword ptr [esi+200h])
 e) 8B 01 (mov eax,dword ptr [ecx])
 f) 8B B7 00 04 00 00 (mov esi,dword ptr [edi+400h])
 g) 26 8B 50 10 (mov edx,dword ptr es:[eax+10h])
 h) 64 8B 45 50 (mov eax,dword ptr fs:[ebp+50h])

Zu Kapitel 11: Skalare und Superskalare Architekturen

1. Der LOAD-Befehl führt zu Verzögerungen, selbst bei einem L1-Cache-Treffer. Weiterhin: RAW-Abhängigkeit zwischen LOAD und SUB bei R1, Verbesserung durch Load-Forwarding. RAW-Abhängigkeit besteht zwischen SUB und ADD bei R2, Verbesserung durch Result-Forwarding.
2. a) Die Ausführung der Befehlsfolge in richtiger Reihenfolge und ohne Register-Umbenennung ist in der nachfolgenden Abbildung gezeigt:

Takt-zyklus	Dekoder1	Dekoder2	Ganzzahl-Einheit1	Ganzzahl-Einheit2	Gleit-komma-einheit	Load/Store
1	FADD	FSUB				
2	ADD				FADD	
3	SUB		ADD		FADD	
4			ADD		FADD	
5			ADD		FADD	
6	COPY	LOAD	SUB		FSUB	
7			SUB		FSUB	
8			SUB		FSUB	
9	...		COPY		FSUB	
10			COPY			
11			COPY			
12	...					LOAD
13						LOAD
14						LOAD
15						LOAD

- b) Es ändert sich nichts gegenüber a), weil die Datenabhängigkeiten parallele Bearbeitung verhindern.
- c) Die Datenabhängigkeit zwischen dem SUB und dem COPY-Befehl wird durch Registerumbenennung aufgehoben, die Ausführung der Befehlsfolge in veränderter Reihenfolge und mit Register-Umbenennung erfolgt entsprechend der nachfolgenden Abbildung:

Taktzyklus	Dekoder1	Dekoder2	Ganzzahl-Einheit1	Ganzzahl-Einheit2	Gleit-komma-einheit	Load/Store
1	FADD	FSUB				
2	ADD	SUB			FADD	
3	COPY	LOAD	ADD		FADD	
4	...		ADD	COPY	FADD	
5			ADD	COPY	FADD	
6	SUB	COPY	FSUB	
7	...		SUB		FSUB	LOAD
8			SUB		FSUB	LOAD
9					FSUB	LOAD
10						LOAD

d) Keine der aufgeführten Maßnahmen würde die Ausführungszeit der Befehlsfolge weiter verkürzen. Die Aufgabenstellung soll zeigen, dass nicht die Hardwareressourcen, sondern die inneren Abhängigkeiten die Parallelisierung begrenzen.

- Nach Amdahls Gesetz ist der Speedup maximal 1.666, also sind max. 66% zu erwarten.
- 2^{50} Byte = 1 Petabyte (ca. 1 Milliarde GigaByte), siehe auch Seite 8.

Zu Kapitel 12: Energieeffizienz von Mikroprozessoren

- Siehe Gleichung 12.2 auf Seite 239.
- Nach Gl.12.2 ist die Leistung bei 0.8 MHz nur noch ein Drittel von 42 W also 14 W. Die mittlere Leistung im Beobachtungszeitraum ist $P = 0.5 \cdot 0W + 0.4 \cdot 14W + 0.1 \cdot 42W = 9.8W$
- Gemäß Datenblatt und Abb.12.2 wird für 12 MHz Takt mindestens eine Versorgungsspannung von 2.7 V gebraucht, 2,5 V reicht nicht aus.
- Der Energieinhalt des Akkus ist $11V \cdot 4Ah = 44Wh$. Diese wären verbraucht in einer Zeit von $44Wh/250W = 0.176h = 10,5$ Minuten.

Zu Kapitel 13: Single Instruction Multiple Data (SIMD)

Da es sich um ganze Zahlen handelt, werden MMX-Befehle benutzt. Die Vektoren werden in MMX-Register geladen, addiert/subtrahiert, und das Ergebnis abgespeichert:

```

1. _asm{
    movq mm0, a
    movq mm1, b
    paddw mm0, mm1 // für a) ; psubw mm0, mm1 für b)
    movq a, mm0
}

```

Ergebnis: a) (5,7,9,0) und b) (3,3,3,0).

- Der Ablauf entspricht im Wesentlichen dem Beispiel auf S. 253:

```

_asm{
    movq mm0,a
    movq mm1,b
    pmaddwd mm0,mm1
    movq mm1,mm0
    psrlq mm0, 32
    padd mm0,mm1 // Ergebnis 38
    movd Skalarprodukt, mm0
}

```

Zu Kapitel 14: Mikrocontroller

1. In der ersten Entwicklungsphase wird man häufig ein neues Programm einspielen und Fehler suchen und beseitigen, günstig ist hier ein In-Circuit-Emulator. Falls nicht vorhanden, kann mit einer Flash-ROM- oder EEPROM-Variante des Controllers gearbeitet werden. Für einen Beta-Test mit beschränkter Stückzahl kann eine EPROM- oder OTP-Version benutzt werden, für die endgültige Version in großer Stückzahl ist Masken-ROM die billigste Lösung.
2. Man sollte einen zyklischen Interrupt mit einem Zeitgeber erzeugen und in der Interrupt-Service-Routine den Sensor auslesen. Schritte: Interrupt-Service-Routine schreiben – Interruptvektor/Sprungbefehl auf diese Routine eintragen – Wählen eines geeigneten Vorteilers für den Zähler; wenn innerhalb einer Sekunde ein Überlauf möglich ist: Kaskadierung eines zweiten Timers – Berechnung der Anzahl Zählerzyklen bei diesem Takt in einer Sekunde – Reload-Register mit diesem Wert laden – Zähler abwärts laufen lassen – Interrupt freigeben. Wegen höherer Betriebssicherheit: Watchdog-Timer aktivieren.
3. Der ADC10 des MSP430 hat einen 10-Bit-A/D-Umsetzer, bei einer Genauigkeit von ± 1 LSB ergibt sich eine Genauigkeit von $2/1024 = 0.002$ also 0.2 %; die geforderte Genauigkeit wird also mit dem ADC10 erreicht.
4. Weil Mikrocontroller oft in batteriebetriebenen Geräten eingesetzt werden; Möglichkeiten zur Verkleinerung der Stromaufnahme: Verkleinerung der Taktfrequenz, Abschaltung der CPU, Abschaltung von Peripheriebausteinen, Anhalten des Oszillators.
5. Ein Watchdog-Timer (Wachhund) ist ein freilaufender Zähler, der bei Überlauf einen Reset auslöst; dies soll genau dann geschehen, wenn das Programm nicht mehr ordnungsgemäß läuft und der WDT dadurch nicht mehr rechtzeitig zurückgesetzt wurde.
6. Mit einem PWM-Signal.

Zu Kapitel 15: Digitale Signalprozessoren

1. Bei der ALU: MAC-Einheit, Sättigungsarithmetik.
Beim Adresswerk: Modulo-Adressierung, bitreverse Adressierung.
2. Basisadresse=400, Modulowert=7, Schrittweite=2.

3. Es ist Basisadresse=320, Modulwert=7, Schrittweite=3. Daraus ergibt sich der Zugriff in der Adress-Reihenfolge 320, 323, 326, 322, 325, 321, 324, 320 usw.
4. Der Puffer umfasst 32 Adressen ab Adresse 64; Vorbereitungsbefehle:

```
MOVE #64, R0    ; Adresszeiger auf Pufferanfang
MOVE #31, M01   ; (Puffergroesse-1) in M01 Register
```

In der Interrupt-Service-Routine:

```
MOVEP X:($FFE0), X0 ; Peripherie-Gerät absolut adressieren
MOVE X0, X:(R0)+    ; Datenwort in Ringpuffer, Postinkrement R0
```

Literaturverzeichnis

- [1] ACPI *Advanced Configuration and Power Interface Specification* Version 4.0a, April 2010, www.acpi.com
- [2] Altenburg, J. , Bögeholz, Harald: *Mikrocontroller-Praxis* Teil 1–4, c't Magazin 24/2003, 25/2003, 5/2004, 6/2004
- [3] ARM Ltd: *Cortex-M3 Technical Reference Manual*, Revision: r1p1 www.arm.com
- [4] Bähring, H.: *Mikrorechner-Technik*, Bd.1. Mikroprozessoren und Digitale Signalprozessoren, Bd.2. Busse, Speicher, Peripherie und Mikrocontroller, Springer–Verlag, 3. Aufl., Berlin, 2002
- [5] Baetke, F.: *IA-64: Strickmuster für den Computer der Zukunft*, Spektrum der Wissenschaft, Rechnerarchitekturen, Dossier 4/2000, S.74
- [6] Barrett, S.F., Pack, D.J. *Embedded Systems*, Design and Applications with the 68HC12 and HCS12, Pearson Education, 2005
- [7] Bauer, F.L.: *Informatik*, Führer durch die Ausstellung, Deutsches Museum, München, 2004
- [8] Beierlein, Th. und Hagenbruch, O.: *Taschenbuch Mikroprozessortechnik*, Fachbuchverlag Leipzig, München, Wien, 1999
- [9] Benz, B.: *Spannungsfeld, Prozessoren: Sparsamkeit kontra Stabilität und Taktfrequenz*, c't Magazin 17/2010, S.166
- [10] Benz, B.: *Nachbrenner – Prozessor-Turbos von AMD und Intel*, c't Magazin 16/2010, S.170
- [11] Benz, B.: *Phenom inside – AMDs Vierkernprozessoren im Detail*, c't Magazin 2/2008, S.80
- [12] Beuth, K.: *Elektronik 2 – Bauelemente*, Vogel Buchverlag, 15.Aufl., Würzburg, 1997
- [13] Beuth, K.: *Elektronik 4 – Digitaltechnik*, Vogel Fachbuch Verlag, 9.Aufl., Würzburg, 1992
- [14] Bleul, A.: *Computer ad astra*, c't Magazin 5/1999, S.108

- [15] Brinkschulte, U., Ungerer, T.: *Mikrocontroller und Mikroprozessoren*, Springer, Berlin 2002
- [16] Elektronikpraxis: *Energieeffizienz und Eco-Design, Sonderheft*, Würzburg, April 2008
- [17] El-Sharkawy, M.: *Digital Signal Processing Applications with Motorola's DSP56002 Processor*, Prentice Hall, London, 1996
- [18] Flik, Th.: *Mikroprozessortechnik*, Springer-Verlag, 7. Aufl., Berlin, 2005
- [19] König, P.: *Sparprogramm - Am Rechner Geldbeutel und Umwelt schonen*, c't Magazin 4/2008, S.78
- [20] Freescale Inc.: *Technical Data DSP56F801 16-bit Digital Signal Processor, Rev. 15, 10/2005*
- [21] Freescale Inc.: *DSP56800 16-Bit Digital Signal Processor, Family Manual, Rev. 3.1, 11/2005*
- [22] Hennessy, J.L. und Patterson, D.A.: *Rechnerarchitektur* Vieweg-Verlag, Braunschweig/Wiesbaden 1994.
- [23] Herrmann, P.: *Rechnerarchitektur*, Vieweg-Verlag, 3. Aufl., Braunschweig/Wiesbaden 2002
- [24] Infineon Technologies AG,: *Halbleiter* , Publicis MCD Corporate Publishing, 2.Aufl., Erlangen und München, 2001
- [25] Infineon Technologies: *C167CR/SR Derivatives, 16-Bit Single-Chip Microcontroller Data Sheet V3.3, 2.2005, User's Manual V3.2, 5.2003, www.infineon.com*
- [26] Intel Corporation: *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 1: Basic Architecture, 2010, Volume 2,3 : Instruction Set Reference, 2010, Volume 4,5: System Programming Guide, 2010, alle: www.intel.com*
- [27] Intel Corporation: *Intel Itanium Architecture Software Developer's Manual Volume 1: Application Architecture, Rev. 2.2, 2006, Volume 2: System Architecture, Rev. 2.2, 2006, Volume 3: Instruction Set Reference, Rev. 2.2, 2006, alle: www.intel.com*
- [28] Johannis, R.: *Handbuch des 80C166*, Feger+Co. Hardware+Software Verlags OHG, Traunstein, 1993
- [29] Koopman, P.: *Microcoded Versus Hard-wired Control*, BYTE 2987, Jan. 1987, S.235
- [30] Kopp, C.: *Moore's law and its implication for information warfare*, 3rd International AOC EW Conference, Jan. 2002
- [31] Lindner, H.: Brauer, H. und Lehmann, C., *Taschenbuch der Elektrotechnik und Elektronik*, Fachbuchverlag Leipzig, 8. Aufl., München, Wien, 2004
- [32] Malone, S.M.: *Der Mikroprozessor, eine ungewöhnliche Biographie*, Springer-Verlag, Berlin, Heidelberg, 1996

- [33] Mengel, St., Henkel, J.: *Einer speichert alles, MRAM – der lange Weg zum Universal-speicher*, c't Magazin 18/2001, S.170
- [34] Messmer, H.P.: *Das PC-Hardwarebuch*, Addison-Wesley Deutschland, 6. Aufl., München, 2000
- [35] Mildenberger, O.: *System- und Signaltheorie* Vieweg-Verlag, Braunschweig/Wiesbaden 1995.
- [36] Müller, H. und Walz, L.: *Elektronik 5 – Mikroprozessortechnik*, Vogel Buchverlag, Würzburg, 2005
- [37] Nus, P.: *Praxis der digitalen Signalverarbeitung mit dem DSP-Prozessor 56002*, Elektor-Verlag, Aachen, 2000
- [38] Rohde, J.: *Assembler GE-PACKT*, mitp-Verlag, Bonn, 2001
- [39] Schmitt, G.: *Mikrocomputertechnik mit dem Controller C 167*, Oldenbourg-Verlag, 2000
- [40] Stiller, A.: *Architektur für echte Programmierer, IA-64, EPIC und Itanium* c't Magazin 13/2001, S.148
- [41] Stiller, A.: *Die Säulen des Nehalem – Die Core-Architektur des neuen Intel-Prozessors*, c't Magazin 25/2008, S.174
- [42] Stiller, A.: *Mikronesische Bauwerke – Architektur und Performance der Netbook-Prozessoren*, c't Magazin 18/2008, S.96
- [43] Sturm, M.: *Mikrocontrollertechnik am Beispiel der MSP430-Familie*, Hanser-Verlag 2006
- [44] Simon, D.E.: *An Embedded Software Primer* Pearson, 1999
- [45] Tanenbaum, A.S. und Goodman, J.: *Computerarchitektur*, 5. Aufl., Pearson Studium, München, 2006
- [46] Texas Instruments: *MSP430x22x2, MSP430x22x4 Mixed Signal Controller datasheet*: www.ti.com
- [47] Texas Instruments: *MSP430x2xx Family User's Guide* www.ti.com
- [48] Windeck, C.: *Spar-O-Matic, Stromsparfunktionen moderner x86-Prozessoren*, c't Magazin 15/2007, S.200
- [49] Werner, M.: *Signale und Systeme* Vieweg-Verlag, 2. Aufl., Braunschweig/Wiesbaden 2005.
- [50] Wittgruber, F.: *Digitale Schnittstellen und Bussysteme*, 2. Aufl., Vieweg-Verlag, Braunschweig/Wiesbaden, 2002
- [51] Yiu, J., *The definitive guide to the ARM Cortex-M3*, Newnes, 2. Ed.l., Amsterdam, 2010

Sachwortverzeichnis

- 386, 189
- 8255, 59
- 56800, 308
- 74181, 94

- Abtastfrequenz, 268
- Abtastung, 302
- Abwärtskompatibilität, 108
- ACPI, 242, 246
- AD-Umsetzer, 267
- ADC, 267, 297
- Address-Multiplexing, 43
- Adress-Aufteilungswort, 71
- Adressbereich, 74
- Adressbus, 5, 62, 71
- Adressbus-Puffer, 99
- Adresse, 70
- Adressierung
 - Basis-, 96
 - Basis-indizierte, 97
 - direkte, 96
 - Index-, 97
 - indizierte, 97
 - nachindizierte, 98
 - registerindirekte, 96, 199
 - speicherindirekt, 97
 - unmittelbare, 95
 - vorindizierte, 98
- Adressierungsarten, 96
- Adressleitungen, 5, 99
- Adressrechner, 96
- Adresswerk, 309
- Adresswort, 71
- ADU, 267
- Advanced RISC Machines Ltd, 156
- AIM, 35
- Akkumulator, 94, 121
- Akzeptoren, 19

- Aliasing, 303
- alignment, 79
- allgemeine Schutzverletzung, 192, 193
- ALU, 93, 307, 309
- Amdahls Gesetz, 218
- Analog-Digital-Umsetzer, 267, 302, 308, 313
- Analogausgang, 267
- Analoge Signale, 267
- Analogeingang, 267
- arithmetisch/logische Einheit, 93, 307, 309
- Arm Cortex-M3, 156
- ARM-Prozessoren, 156
- ASCII-Zeichensatz, 9
- Assembler, 106
- Assemblersprache, 106
- Auflösung, 267
- Ausgabe, 56
- Ausgabebausteine, 5
- Ausgangslastfaktor, 24
- Ausnahmen, 116
- Ausrichtung, 79
- Auto-Reload-Betrieb, 263
- Avalanche Induced Migration, 35

- Bank, 50
- Barrel-Shifter, 307
- Basisregister, 96
- Baustein-Freigabe, 65
- Befehls-Pipelining, 203
- Befehlslesezyklus, 87
- Befehlssatz, 103
- Befehlszyklus, 88, 92
- Biased-Exponent, 15
- bidirektionale Bustreiber, 66
- big-endian, 80
- Bit, 7
- Bit Banding, 164
- Bitleitungen, 32

- bitreverse Adressierung, 307
- Bootloader, 285
- Bootvorgang, 108
- Boundary Scan Path, 275
- Boundary-Scan-Cells, 275
- Branch Prediction, 208
- Branch-History-Tabelle, 208
- Brown-Out, 274
- BSC, 275
- Bulk, 20
- Burst, 47
- Burst EDO-DRAM, 47
- Bus, 4, 62
 - asynchroner, 68
 - synchroner, 67
- Bus Receiver, 66
- Bus Snooping, 189
- Bus snooping, 219
- Bus Transceiver, 66
- Bus-Skew, 69
- Busarbitration, 69
- Busleitungen, 4
- Busmaster, 69
- Busprotokoll, 62
- Busspezifikation, 67
- Bustreiber, 65
- Busy-Waiting, 112
- Buszyklus, 92
- Byte, 7
- Cache
 - Kohärenz, 187
 - Konsistenz, 187
 - direkt abbildender, 184
 - mehrfach assoziativer, 185
 - n-fach-assoziativer, 185
 - n-Wege-, 185
 - teillasoziativer, 185
 - vollassoziativer, 184
- Cache Clear, 189
- Cache Flush, 189
- Cache Lines, 183
- Cache-Fehltreffer, 181
- Cache-Line, 181
- Cache-Speicher, 181
- Cache-Treffer, 181
- Cache-Zeilen, 183
- CAN-Bus, 272
- Capture, 263, 296
- Carry Flag, 11, 95
- CAS, 43
- CAS-Before-RAS-Refresh, 46
- Central Processing Unit, 4
- Chip, 21
- Chip Enable, 65
- Chip-Select, 65
- Chipsätze, 81
- CISC, 101
- CMOS, 25
- CMSIS, 168
- Compare, 263, 295
- Complex Instruction Set Computer, 101
- Control Speculation, 234
- Control Unit, 91
- Copy-Back-Strategie, 188
- Core-Architektur, 227
- CoreSight, 169
- Cortex-M3, 156
- Cortex-Prozessoren, 156
- counter, 262
- CPU, 4
- DAC, 268
- Daisy-Chaining, 69, 113
- Data Speculation, 233
- Daten-Hazards, 205
- Datenabhängigkeit, 205, 311
- Datenbus, 5, 62
- Datenbus-Puffer, 99
- Datenleitungen, 5, 99
- DAU, 268
- DDR-SDRAM, 48
- Debugging, 169, 286
- Dekodierung, 87
- Delay-Branch-Technik, 207
- Demand-Paging, 175
- Derivate, 259
- Deskriptoren, 191
- Dezimalsystem, 9
- Die, 21
- Digital-Analog-Umsetzer, 267, 268, 308
- digitale Signalprozessoren, 304

- Digitale Signalverarbeitung, 303
- Dioden-Transistor-Logik, 27
- Direct Memory Access, 117
- direkte Adressierung, 124
- Dirty-Bit, 175
- Displacement, 96
- DMA, 117
- DMA-Controller, 117
- Donatoren, 19
- Double Data Rate-SDRAM, 48
- Drain, 20
- DRAM, 40
 - Fertigung, 51
 - EDO-, 47
 - synchrones, 47
- DSP, 304
- DTL, 27
- Dual Inline Memory Modules, 46

- E/A-Adressierung
 - isolierte, 75
 - speicherbezogene, 75
- E/A-Bausteine, 56, 70, 99
- E/A-Zugriff, 70, 75, 112, 113, 197, 308
- Echtzeit-Betriebssystem, 281
- Echtzeitfähigkeit, 269
- ECL, 27
- EEPROM, 37
- Effective Privilege Level, 193
- Einchip-Computer, 258
- Eingabe, 56
- Eingabebausteine, 5
- Eingangslastfaktor, 24
- eingebettete Systeme, 258
- Einkristall, 19
- Embedded Systems, 4, 258
- Emitter Coupled Logic, 27
- Enable, 65
- Energieeffizienz, 237
- EPIC, 231
- EPL, 193
- EPROM, 2, 36
- EPROM-Simulator, 285
- Erholungszeit, 45
- erweiterte Adressierung, 331
- Evaluation-Board, 287

- Exception, 165
- Exceptions, 116
- Execution-Trace-Cache, 226

- FAMOST, 35
- Fan-In, 24
- Fan-Out, 24
- Fast Page Mode, 47
- Feldeffekttransistor, 19
- Ferroelektrische RAM, 54
- Festwertspeicher, 34
- FET, 19, 238
 - selbstleitender, 21
 - selbstsperrender, 20
- First-Level-Cache, 182
- Flags, 95
- Flash-Speicher, 37
- Fließbandverarbeitung, 203
- Flipflop, 38
- Floating Gate, 35
- FPM, 47
- Fragmentierung
 - externe, 179
 - interne, 174
- Frontsidebus, 82
- Fusible Link, 35

- ganze Zahlen
 - vorzeichenbehaftete, 10
 - vorzeichenlose, 10
- Gate, 20
- Gated Timer, 262
- Gates, 190
- GDT, 191
- Gegentakt-Endstufe, 64
- General Protection Fault, 192
- gepackte Daten, 249
- Giant Magnetoresistive Effect, 53
- Giftbit, 215
- Gleitkommaformat
 - doppelt genaues, 15
 - einfach genaues, 15
- Globale Deskriptorentabelle, 191
- Grabenkondensatoren, 41

- Hardware-Reload, 263
- Harvard-Architektur, 5, 305

- Hidden-Refresh, 46
- Hint-Befehle, 164
- hochohmiger Zustand, 63
- Hyperthreading, 216
- I²C-Bus, 271, 300
- I/O-Adressing
 - Isolated, 75
 - Memory Mapped, 75
- I/O-Ports, 56
- IA-32, 189
- IC, 21
- ICE, 287
- IDE, 288
- IDT, 191
- IGFET, 20
- In-Circuit-Emulator, 287
- in-order-issue, 209
- In-System-Debugging, 286
- Interrupt-Behandlungsroutine, 112
- Interrupt-Handler, 112
- Indexregister, 96, 121
- inhärente Adressierung, 123
- Inklusion, 188
- Instruction Set Architecture, 107
- Integrated Circuits, 21
- Integrated Development Environment, 288
- Integrationsgrad, 22
- Integrierte Schaltkreise, 21
- Intel
 - 80386, 189, 223
 - Itanium, 231
 - Pentium 4, 189, 223, 226, 250
 - Pentium II, 189
 - Pentium III, 189, 224, 251
- Intel 64-Architektur, 223
- Inter IC-Bus, 271
- Interleaving, 47
- Interrupt, 112, 259, 263, 309
 - Auslösung, 113
 - Behandlungsroutine, 112
 - Behandlungsroutine, 270
 - Priorisierung, 113
 - System, 269
 - präziser, 213
 - zyklischer, 265
- Interrupt Enable Flag, 116
- Interrupt-Behandlungsroutinen, 112
- Interrupt-Controller, 115, 116
- Interrupt-Deskriptoren-Tabelle, 191
- Interrupt-Freigabe-Bit, 116
- Interrupt-Service-Routine, 112
- Interrupt-Vektor, 114
- Interrupt-Vektorisierung, 114
- Interrupteingang
 - maskierbarer, 116
 - nicht maskierbarer, 116
- Interruptquellen, 69, 269
- IO-Gatterblock, 44
- IO-Permission-Bitmap, 197
- IO-Privilege-Level, 197
- IrDA, 273, 300
- ISA, 107
- Joint Test Action Group, 275
- JTAG, 275
- Kernspannung, 246
- L1-Cache, 182
- L2-Cache, 182
- Langsame störsichere Logik, 27
- Lastfaktoren, 24
- Latenzzeit, 204
- LDT, 191
- Least Significant Bit, 8
- Lesezyklus, 33
- Limitation, 308
- lineare Adresse, 193
- little-endian, 81
- Load Forwarding, 206
- Load/Store-Architektur, 102
- logische Adresse, 173
- Lokale Deskriptorentabelle, 191
- Loop Stream Decoder, 227
- Loop-Unrolling, 216
- LOW-aktives Signal, 65
- LSB, 8
- LSL, 27
- MAC-Befehl, 251, 253, 308
- MakroOp-Fusion, 227
- Maschinenbefehle, 87

- Maschinencode, 87, 198
- Maschinenwort, 7
- Maschinenzyklen, 92
- Memory Management Unit, 177
- Memory Protection Unit, 167
- MESI-Kohärenz-Protokoll, 189
- MicroOp-Fusion, 227
- Mikroarchitekturebene, 107
- Mikrocode, 92, 100, 235
- Mikrocode-ROM, 91, 100
- Mikrocomputer, 2, 4
- Mikrocontroller, 258
- Mikrooperationen, 209
- Mikroprogrammierung, 91, 100
- Mikroprozessor, 2
- Mikrorechnersystem, 4
- MIPS, 101
- MMU, 177
- MMX-Einheit, 250
- Mnemonic, 106
- MOD-R/M-Byte, 198
- modifizierte Harvard-Architektur, 305
- Modulation, 300
- Modulo-Adressierung, 306
- Monitor-Programm, 287
- Mooresches Gesetz, 4
- MOSFET, 20
- Most Significant Bit, 8
- MRAM, 53
- MROM, 34
- MSB, 8
- Multiplexing, 69
- Multiplizieren-Akkumulieren, 251, 308
- Multitasking, 177
- Multithreading, 216
- MXCSR-Register, 255

- n-leitend, 19
- NaN, 17
- NAND-Schaltglied, 29
- Negation, 29
- Nehalem, 227
- Nested Vector Interrupt Controller, 165
- Netburst Micro-Architecture, 226
- Nibble, 7
- NMI, 116

- NMOS, 25
- No-Write-Allocate-Strategie, 188
- Non-cacheable-Area, 188
- Normalisierung, 15
- Northbridge, 84
- Nullbit, 95
- Nur-Lese-Speicher, 34
- NV-RAM, 37
- NVIC, 165
- Nyquist-Kriterium, 303

- ODER-Verknüpfung, 28
- Opcode, 87
- Opcode Fetch, 87
- Open-Drain-Ausgänge, 63
- orthogonaler Befehlssatz, 141
- Orthogonalität, 108
- Oscillator-Watchdog, 273
- OTPROM, 35
- out of order execution, 212
- Output Enable, 65
- Overflow Flag, 13, 95
- Overlays, 173
- Ovonic Unified Memory, 54

- p-leitend, 19
- P6, 227
- P6-Mikroarchitektur, 224
- packed data, 249
- Page, 51
- Paging, 173, 195, 231
- Parameter, 137
- Paritätsbit, 95
- Parity Flag, 95
- PC, 2, 46, 47, 59, 66–68, 74, 81, 112, 118, 173, 183, 195, 225, 249, 250, 271, 277
- PCI-Express, 82
 - Lane, 82
- PCIe, 82
- Performance Monitor, 225
- Personal Computer, 2
- Pipeline, 107, 204, 209, 309, 311
 - Hemmnisse, 204
- PLL, 84
- Plug and Play, 74
- PMOS, 25

- Poison Bit, 215
- Polling, 112
- Prädikation, 216, 233
- Präfix-Byte, 198
- precharge time, 45
- Privilege Levels, 189
- privilegierte Befehle, 190
- Program Counter, 122
- Programmierimpuls, 35
- Programmzähler, 5, 87, 104
- PROM, 35
- Protected Mode, 189
- Protected Virtual Address Mode, 189
- Prozessor, 4
- PSR, 159
- pulsweitenmoduliertes Signal, 266
- PWM-Signal, 266

- Quantisierungsfehler, 268, 303
- QuickPath Interconnect, 229

- räumliche Lokalität, 181
- RAM, 31
 - dynamisches, 40
 - ferroelektrisches, 52
 - magnetoresistives, 52, 53
 - nicht-flüchtiges, 37, 260
 - nonvolatile, 37, 260
 - NV-, 260
 - statisches, 38
- RAM-Simulator, 285
- Random Access Memory, 31
- RAS, 43
- RAS-Only-Refresh, 46
- RAW-Hazard, 205
- Read Only Memory, 31, 34
- Read-After-Write-Hazard, 205
- Reduced Instruction Set Computer, 101
- Refresh, 41, 46
- Register, 89, 261
- Register Renaming, 214
- Register-Bypass, 206
- Register-Rotation, 235
- Register-Umbenennung, 214
- Registeradressierung, 96
- Registerfenster, 103, 231
- Registersatz, 89, 259

- Reset-Vektor, 108
- Ressourcenkonflikte, 204
- Result Forwarding, 206
- RISC, 101
- ROM, 31, 34
- RTOS, 281

- Sättigungsarithmetik, 251, 308
- Sample-and-Hold-Baustein, 297
- Saturation, 308
- Schaltkreisfamilien, 23
- Schnittstellen, 58
- Schreib-/Lese-Speicher, 31
- Schreibzyklus, 33
- Scoreboard, 211
- Scrambling, 51
- SDRAM, 47, 50
- Second-Level-Cache, 182
- Segmenttabelle, 179
- Seitenauslagerung, 173
- Seitenersetzung, 176
- Seitenfehler, 175
- Seitentabelle, 174
- Selektor, 191
- Service-Anforderungen, 111
- Sign Flag, 95
- Signal-Laufzeit, 23
- Signale, analoge, 302
- Signalverarbeitung, digitale, 302
- signed binary numbers, 11
- SIMD, 249
- Simulator, 287
- Single Inline Memory Modules, 46
- Single Instruction Multiple Data, 249
- Single-Chip-Computer, 258
- Skalarität, 102
- Skalierung, 97
- Slave, 69
- Source, 20
- Southbridge, 84
- SPD-EEPROM, 47
- Special Function Register, 261
- Speedstep, 247
- Speicher-Latenzzeit, 181
- Speicherbandbreite, 181
- Speichermodell, flaches, 194

- Speichermodule, 46
- Speichersegmentierung, 177, 191
- Speichersteuerung, 43
- spekulative Ausführung, 208, 215
- spekulative Laden, 229
- spekulatives Laden, 215, 233
- Spezialregister, 90
- SPI, 272, 313
- Spiegeladressen, 71, 74
- Sprungvorhersage
 - dynamische, 208
 - statische, 208
- SRAM, 38
- SSE-Einheit, 251
- SSE2, 251
- SSE3, 256
- Stack, 98
- Stack Technology, 42
- Stackpointer, 98, 122
- Standard-TTL, 24
- Standardmikrocontroller, 258
- Stapeltechnik, 42
- Statusregister, 95
- Steuerbus, 5, 62
- Steuerleitungen, 5, 99
- Steuerwerk, 89, 91, 101, 259, 309, 311
- Streaming SIMD Extension, 251
- Strukturbreite, 22
- super-pipelined, 214
- Superskalarität, 102
- Swapping, 179
- Symmetrischen Multiprocessing, 219

- Tag, 183
- Taktgenerator, 6, 109
- Taktzyklus, 67, 92
- TAP, 275
- Task State Segment, 197
- TDP, 229, 245
- Test Access Port, 275
- Tetrade, 7
- Thermal Design Power, 229, 245
- Thrashing, 177, 185
- Thread, 216
- timer, 262
- Timing, 33, 67

- TLB, 197
- Transistor-Transistor-Logik, 24
- Translation Lookaside Buffer, 197
- Traps, 116
- Treiber, 65
- Trench-Technologie, 41
- Tristate-Ausgang, 65
- TTL, 24
- Tunnel Magneto-resistive Effect, 53
- Tunnelstrom, 53
- Turbo Boost, 229

- UART, 299
- Überlauf, 13
- Überlaufbit, 13, 95
- Überschuss-Exponent, 15
- Übertragsbit, 11, 95
- Uncore, 229
- UND-Gatter, 28
- UND-Verknüpfung, 28
- Universal Serial Bus, 272
- Universalregister, 90
- unmittelbare Adressierung, 123
- Unterabtastrfehler, 303
- Unterbrechung, 112
- Unterprogramm, 129
- Urladeprogramm, 108
- USB, 272
- USCI, 299

- Vektorprozessor, 249
- Vergleicher, 71
- Verlustleistung, 23
- very long instruction word, 216
- virtuelle Adressen, 173
- virtueller Speicher, 173
- VLIW, 216
- von-Neumann-Maschine, 1
- von Neumann-Architektur, 5
- Vorladeschaltkreise, 44
- Vorladezeit, 45
- vorzeichenbehaftete ganze Zahlen, 11
- Vorzeichenbit, 12, 95

- Wartetakt, 67
- Watchdog-Timer, 273
- WDT, 273, 290

Wort, 7
Wortleitungen, 32
Write-After-Read-Hazard, 207
Write-Allocate-Strategie, 188
Write-Through-Strategie, 188

Zähler/Zeitgeber, 262, 293
Zeichensatz, 9
zeitliche Lokalität, 181
Zeitmarkenzähler, 225
Zentraleinheit, 4
Zero Flag, 95
Zugriffszeit, 33
Zweierkomplement-Darstellung, 13
Zykluslauf, 118
Zykluszeit, 33