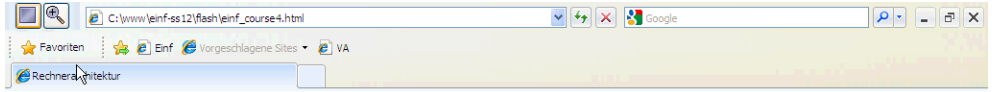
Script **generated by TTT**

Title: Einf_HF (14.05.2012)

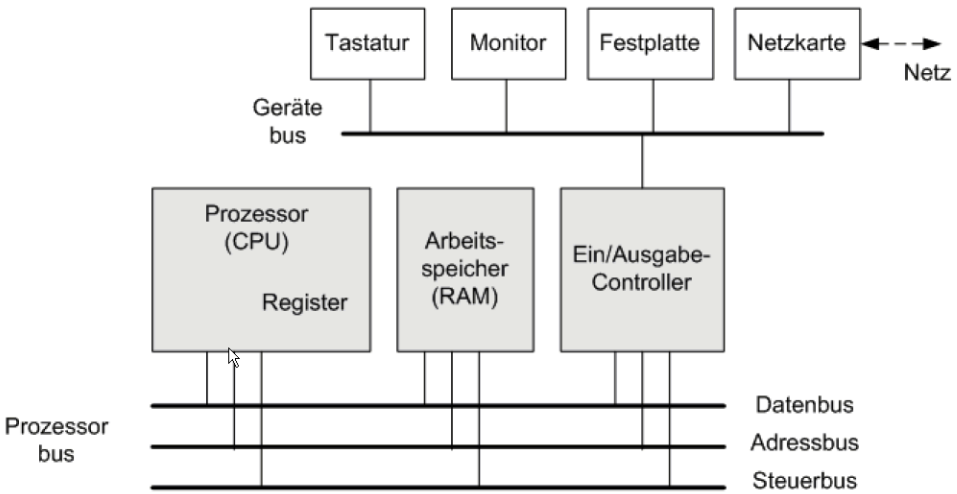
Date: Mon May 14 14:16:59 CEST 2012

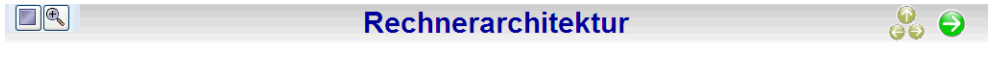
Duration: 89:11 min

Pages: 29



- Fragestellungen des Abschnitts:
  - Aus welchen (Hardware-)Elementen setzt sich ein Rechner zusammen?
  - Wie kommunizieren die einzelnen Komponenten eines Rechners?
  - Wie sieht die Schnittstelle zwischen Hardware und Software aus (d.h. Maschinenbefehle)?
  - Wie werden Zahlen, Text, Bilder, und Töne intern dargestellt?





- Fragestellungen des Abschnitts:
  - Aus welchen (Hardware-)Elementen setzt sich ein Rechner zusammen?
  - Wie kommunizieren die einzelnen Komponenten eines Rechners?
  - Wie sieht die Schnittstelle zwischen Hardware und Software aus (d.h. Maschinenbefehle)?
  - Wie werden Zahlen, Text, Bilder, und Töne intern dargestellt?

Aufbau eines Rechners

Maschinenbefehle

Befehlszyklus

Interndarstellung von Information

Generated by Targeteam

Befehlsvorrat

Transportbefehle

z.B. LOAD, STORE. LOAD: Transportieren von Daten vom Arbeitsspeicher in ein Register; STORE spezifiziert den umgekehrten Weg.

Arithmetische und logische Befehle

z.B. ADD, SUB, AND, OR, CMP

Schiebebefehle

z.B. SH (Shift links, rechts), ROT (Schieben im Kreis)

Sprungbefehle

z.B. JMP (Jump), JGT (Jump Greater Than) - (bedingte) Änderung der Ablaufreihenfolge

Sonderbefehle

Behandlung von Unterbrechungen (z.B. Alarm bei Division durch 0), Änderungen des Maschinenstatus, Rückmeldungen von E/A Geräten, Laden von Prozessbeschreibungen, Synchronisationsbefehle bei Speicherzugriff etc.

Maschinenbefehle

Einfache Kommandos, die die CPU ausführen kann; Setzen sich zusammen aus Operationsteil und Operandenteil (Adressteil).

Befehlsvorrat

Beispielprogramm in Maschinensprache (Assembler)

Einfache Kommandos, die die CPU ausführen kann; Setzen sich zusammen aus Operationsteil und Operandenteil (Adressteil).

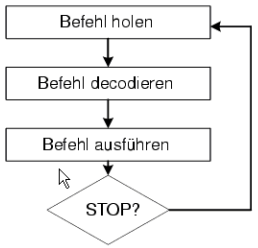
Befehlsvorrat

Beispielprogramm in Maschinensprache (Assembler)

Generated by Targeteam

Ausführung eines Maschinenbefehls: festes Schema (Bereitstellung, Bereitstellen der Operanden, Befehl entschlüsseln, Ausführung). Dieses Schema heißt Befehlszyklus.

Sequentielle Bearbeitung

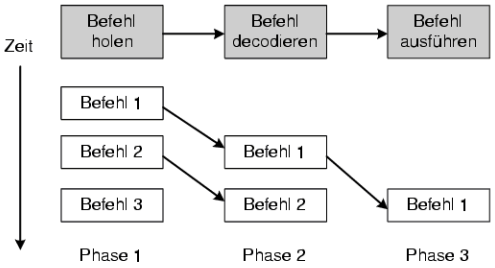


kein ausführbares Programm ⇒ Ausführung von NOP ("No Operation").

Fließband Bearbeitung (Pipelining)

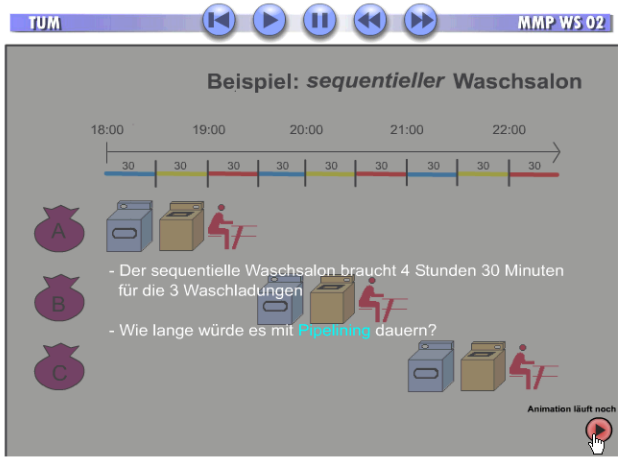
Fließband Bearbeitung (Pipelining)

Bearbeitung jeden Befehls in mehreren Phasen. Überlappende Verarbeitung. Quasi-parallele Ausführung mehrerer Maschinenbefehle.

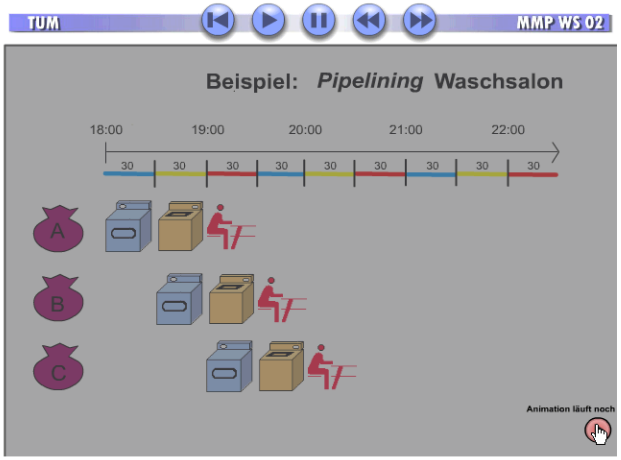


Pipelining Animation

Pipelining Animation

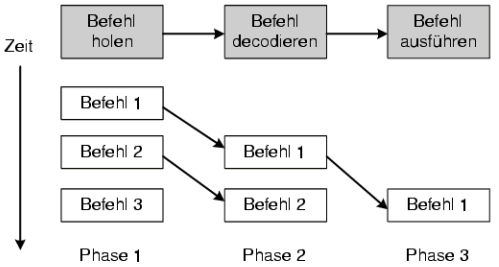


Pipelining Animation



Fließband Bearbeitung (Pipelining)

Bearbeitung jeden Befehls in mehreren Phasen. Überlappende Verarbeitung. Quasi-parallele Ausführung mehrerer Maschinenbefehle.



Pipelining Animation

Komprimierung

Datenkompression: reduzierte Speicher- und Übertragungskosten.

Verlustfreie Kompression

Ausnutzung von Mustern und Redundanzen in den Daten; Ausnutzung der Häufigkeit von Symbolen durch Änderung der Codierung.

Verlustbehaftete Kompression

Ausnutzung von Medien- und Wahrnehmungseigenschaften, z.B. bei MP3.

Codierung

Zuordnung (oder Abbildung) der Werte eines Zeichenvorrats auf Werte eines anderen Zeichenvorrats.

Zeichen: Ausprägung (Form, Wert) eines Signals; auch: Symbole.

Zeichenvorrat: Menge der Zeichen (d.h. Formen, Werte), die ein bestimmtes Signal annehmen kann.

Codierung erfolgt für bestimmten Zweck:

- Speicherung
- Übertragung
- Komprimierung, z.B. von Bildern oder Video
- Verschlüsselung
- Veranschaulichung

Codierung z.B. notwendig um für Menschen verständliche Information auf für Rechner verständliche oder speicherbare Darstellung abzubilden. (Symbole auf Bitfolgen.)

Abbildung berechenbar, eindeutig und (in der Regel) umkehrbar.

Codierung im Binärsystem. Zwei Ziffern 0,1 ("Bits") geben Anzahl von Zweierpotenzen an. Vgl. Dezimalsystem: Zehn Ziffern geben Anzahl von Zehnerpotenzen an.

Beispiel

Dezimalsystem: $148 = 1 * 10^2 + 4 * 10^1 + 8 * 10^0$

Binärsystem: $1010 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0$ (= 10 im Dezimalsystem)

Formel für Wert einer Binärsystem-Zahl

$W = \sum_{i=0}^{n-1} (b_i \times 2^{n-1-i})$

mit den Binärziffern $b_i \in \{0,1\}$ und n ist die Anzahl der verwendeten Bits (d.h. eine n-stellige Zahl). Beachte, es wird die Folge $b_0\, b_1 \ldots b_{n-1}$ betrachtet.

Beispiel

eine ganze Zahl sei als 8 bit lange Zahl zur Basis 2 dargestellt

$W(00001101_2) = 0 \times 2^7 + \ldots + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 1 = 13_{10}$

[Verfahren zur Umwandlung](#)

Feste Ziffernanzahl

Typisch: Feste Bitzahl, meist ebenfalls Zweierpotenz. Z.B. 4 Bit, 16 Bit, 32 Bit oder 64 Bit. Aktuell entweder 32 oder 64 Bit verwendet. Mit n Bit codierbar: Werte 0 bis 2^n -1.

[Negative Zahlen](#)

Umwandlung einer Dezimalzahl w in eine Dualzahl z

dividiere w durch 2: Ergebnis w_1 und Rest r_0

dividiere w_1 durch 2: Ergebnis w_2 und Rest r_1

fahre fort, bis das Ergebnis der Division 0 und Rest r_k ist.

Die Dualzahl ist $z = r_k r_{k-1} \ldots r_1 r_0$

Beispiel

Dezimalzahl w = 23

23 : 2 = 11 mit Rest 1

11 : 2 = 5 mit Rest 1

5 : 2 = 2 mit Rest 1

2 : 2 = 1 mit Rest 0

1 : 2 = 0 mit Rest 1

Die Dualzahl lautet: z = 00010111 (in 8-Bit Darstellung)

Codierung ganzer Zahlen

Codierung im Binärsystem. Zwei Ziffern 0,1 ("Bits") geben Anzahl von Zweierpotenzen an. Vgl. Dezimalsystem: Zehn Ziffern geben Anzahl von Zehnerpotenzen an.

Beispiel

Dezimalsystem: $148 = 1 * 10^2 + 4 * 10^1 + 8 * 10^0$

Binärsystem: $1010 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0$ (= 10 im Dezimalsystem)

Formel für Wert einer Binärsystem-Zahl

$$W = \sum_{i=0}^{n-1} (b_i \times 2^{n-1-i})$$

mit den Binärziffern $b_i \in \{0,1\}$ und n ist die Anzahl der verwendeten Bits (d.h. eine n-stellige Zahl). Beachte, es wird die Folge $b_0\ b_1 \ldots b_{n-1}$ betrachtet.

Beispiel

eine ganze Zahl sei als 8 bit lange Zahl zur Basis 2 dargestellt

$$W(00001101_2) = 0 \times 2^7 + \ldots + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 1 = 13_{10}$$

Verfahren zur Umwandlung

Feste Ziffernanzahl

Typisch: Feste Bitzahl, meist ebenfalls Zweierpotenz. Z.B. 4 Bit, 16 Bit, 32 Bit oder 64 Bit. Aktuell entweder 32 oder 64 Bit verwendet. Mit n Bit codierbar: Werte 0 bis 2^n -1.

Negative Zahlen

Codierung ganzer Zahlen

Zehn Ziffern geben Anzahl von Zehnerpotenzen an.

Beispiel

Dezimalsystem: $148 = 1 * 10^2 + 4 * 10^1 + 8 * 10^0$

Binärsystem: $1010 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0$ (= 10 im Dezimalsystem)

Formel für Wert einer Binärsystem-Zahl

$$W = \sum_{i=0}^{n-1} (b_i \times 2^{n-1-i})$$

mit den Binärziffern $b_i \in \{0,1\}$ und n ist die Anzahl der verwendeten Bits (d.h. eine n-stellige Zahl). Beachte, es wird die Folge $b_0\ b_1 \ldots b_{n-1}$ betrachtet.

Beispiel

eine ganze Zahl sei als 8 bit lange Zahl zur Basis 2 dargestellt

$$W(00001101_2) = 0 \times 2^7 + \ldots + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 1 = 13_{10}$$

Verfahren zur Umwandlung

Feste Ziffernanzahl

Typisch: Feste Bitzahl, meist ebenfalls Zweierpotenz. Z.B. 4 Bit, 16 Bit, 32 Bit oder 64 Bit. Aktuell entweder 32 oder 64 Bit verwendet. Mit n Bit codierbar: Werte 0 bis 2^n -1.

Negative Zahlen

Positive ganze Zahlen Darstellung im Binärsystem. Für negative ganze Zahlen mehrere Möglichkeiten.

Vorzeichen-Darstellung

Erstes Bit: Vorzeichen (0 = +, 1 = -), restliche Bits Absolutwert der Zahl im Binärsystem. Bei n Bits sind Zahlen von -2^{n-1} -1 bis 2^{n-1} -1 codierbar. Zwei Nullen: +0 (000...00), -0 (100...00).

Beispiel

Zweierkomplement-Darstellung

Beispiel

Für n = 4 sind die Zahlen von -7 bis +7 codierbar:

0 = 0000	-0 = 1000
1 = 0001	-1 = 1001
2 = 0010	-2 = 1010
3 = 0011	-3 = 1011
4 = 0100	-4 = 1100
5 = 0101	-5 = 1101
6 = 0110	-6 = 1110
7 = 0111	-7 = 1111

Negative Zahlen

Positive ganze Zahlen Darstellung im Binärsystem. Für negative ganze Zahlen mehrere Möglichkeiten.

Vorzeichen-Darstellung

Erstes Bit: Vorzeichen (0 = +, 1 = -), restliche Bits Absolutwert der Zahl im Binärsystem. Bei n Bits sind Zahlen von -2^{n-1} -1 bis 2^{n-1} -1 codierbar. Zwei Nullen: +0 (000...00), -0 (100...00).

Beispiel

Zweierkomplement-Darstellung

Zweierkomplement-Darstellung

Eine negative Zahl mehr als positive Zahlen. Einfache Umsetzung von Addition und Subtraktion.

Beispiel für 4 bit Darstellung

Formel für Wert einer Zweierkomplement-Zahl

$$W = -b_0 \times 2^{n-1} + \sum_{i=1}^{n-1} \left(b_i \times 2^{n-1-i}\right)$$

mit $b_i \in \{0,1\}$. n ist hier die Anzahl der Bitstellen.

Beispiel

Wert der Zahl W: -1

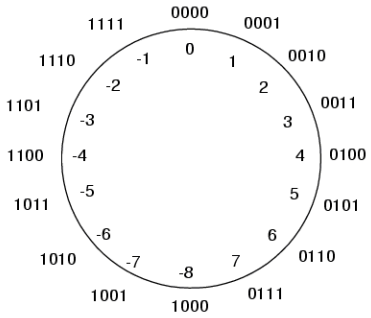
Binärdarstellung mit 4 Bit: 1111

$$W = -2^3 + 2^2 + 2^1 + 2^0 = -8 + 7 = -1$$

Rechnen mit Zweierkomplement-Zahlen

Beispiel für 4 bit Darstellung

Zweierkomplement-Darstellung mit 4 bit für eine ganze Zahl



$W_+ = \sum_{i=1}^{3} \left(b_i \times 2^{3-i}\right)$ (positive Zahl)

$W_- = -2^3 + \sum_{i=1}^{3} \left(b_i \times 2^{3-i}\right)$ (negative Zahl)

mit $b_i \in \{0,1\}$

positive Zahl: erstes Bit ist 0

negative Zahl: erstes Bit ist 1

Zweierkomplement-Darstellung

Eine negative Zahl mehr als positive Zahlen. Einfache Umsetzung von Addition und Subtraktion.

Beispiel für 4 bit Darstellung

Formel für Wert einer Zweierkomplement-Zahl

$$W = -b_0 \times 2^{n-1} + \sum_{i=1}^{n-1} \left(b_i \times 2^{n-1-i}\right)$$

mit $b_i \in \{0,1\}$. n ist hier die Anzahl der Bitstellen.

Beispiel

Wert der Zahl W: -1

Binärdarstellung mit 4 Bit: 1111

$$W = -2^3 + 2^2 + 2^1 + 2^0 = -8 + 7 = -1$$

Rechnen mit Zweierkomplement-Zahlen

Generated by Targeteam

Negativbildung und Grundrechenarten sind einfach durchführbar.

Negativbildung einer Zahl

Komplementbildung (Bits invertieren) und 1 addieren.

Beispiel

Zweierkomplement-Codierung mit 8 Bit für -14:

14 =	00001110
Komplement:	11110001
1 addiert:	11110010

Addition von zwei Zahlen

Stellenweise mit Übertrag, analog zum Dezimalsystem.

Differenzbildung von zwei Zahlen

Realisierbar durch Addition mit negativer Zahl.

Beispiel

Berechnung 17 - 14:

dezimal	dual
17	00010001
+(-14)	11110010
= 3	00000011

Rechnen mit Zweierkomplement-Zahlen

Negativbildung einer Zahl

Komplementbildung (Bits invertieren) und 1 addieren.

Beispiel

Zweierkomplement-Codierung mit 8 Bit für -14:

14 =	00001110
Komplement:	11110001
1 addiert:	11110010

11110001
00000001
+ 1
11110010

invertieren jedes einzelnen Bits
≙ -14

00001101
+ 00001110
11
00011011

Addition von zwei Zahlen

Stellenweise mit Übertrag, analog zum Dezimalsystem.

Differenzbildung von zwei Zahlen

Realisierbar durch Addition mit negativer Zahl.

Beispiel

Berechnung 17 - 14:

dezimal	dual
17	00010001
+(-14)	11110010
= 3	00000011

Codierung von Text

Alphanumerische Daten - ISO-ASCII 8-bit-Code

Darstellung von Buchstaben und Ziffern in einer 8-Bit Folge, d.h. wie Zahl zwischen 0 und 255.

ISO = International Standards Organisation

ASCII = American Standard Code for Information Interchange

Kleinbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (97 - 122)

Großbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (65 - 90)

Ziffern 0 bis 9 sind in aufsteigender Reihenfolge dargestellt (48 - 57)

Darstellung von Sonderzeichen, z.B. CR (Carriage Return = Absatzende), LF (Linefeed = Neuzeile)

Zu den entsprechenden Zeichen des ASCII Codes wird der jeweilige Zahlenwert zur Basis 10 angegeben.

Zeichen	Dezimal	Binärdarstellung
a	97	01100001
A	65	01000001
b	98	01100010
B	66	01000010
0	48	00110000
?	63	00111111
CR	13	00001101

Bei Netzübertragung häufig noch 7-bit ASCII Code. Spezielle Zeichen wie ü, ä oder ö nicht enthalten, daher in 7-bit Darstellung konvertieren. (Mittels "Escape-Zeichen").

Codierung von Text

Darstellung von Buchstaben und Ziffern in einer 8-Bit Folge, d.h. wie Zahl zwischen 0 und 255.

ISO = International Standards Organisation

ASCII = American Standard Code for Information Interchange

Kleinbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (97 - 122)

Großbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (65 - 90)

Ziffern 0 bis 9 sind in aufsteigender Reihenfolge dargestellt (48 - 57)

Darstellung von Sonderzeichen, z.B. CR (Carriage Return = Absatzende), LF (Linefeed = Neuzeile)

Zu den entsprechenden Zeichen des ASCII Codes wird der jeweilige Zahlenwert zur Basis 10 angegeben.

Zeichen	Dezimal	Binärdarstellung
a	97	01100001
A	65	01000001
b	98	01100010
B	66	01000010
0	48	00110000
?	63	00111111
CR	13	00001101

Bei Netzübertragung häufig noch 7-bit ASCII Code. Spezielle Zeichen wie ü, ä oder ö nicht enthalten, daher in 7-bit Darstellung konvertieren. (Mittels "Escape-Zeichen").

Unicode

Codierung von Text

ASCII = American Standard Code for Information Interchange

Kleinbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (97 - 122)

Großbuchstaben sind in alphabetischer Reihenfolge durchnummeriert (65 - 90)

Ziffern 0 bis 9 sind in aufsteigender Reihenfolge dargestellt (48 - 57)

Darstellung von Sonderzeichen, z.B. CR (Carriage Return = Absatzende), LF (Linefeed = Neuzeile)

Zu den entsprechenden Zeichen des ASCII Codes wird der jeweilige Zahlenwert zur Basis 10 angegeben.

Zeichen	Dezimal	Binärdarstellung
a	97	01100001
A	65	01000001
b	98	01100010
B	66	01000010
0	48	00110000
?	63	00111111
CR	13	00001101

Ziffer 0 und nicht die Zahl 0

Bei Netzübertragung häufig noch 7-bit ASCII Code. Spezielle Zeichen wie ü, ä oder ö nicht enthalten, daher in 7-bit Darstellung konvertieren. (Mittels "Escape-Zeichen").

gelegentlich

Unicode

Unicode codiert Zeichen mit zwei Bytes. 65536 Zeichen.