4.3 PC-Hardware/Bussystem

4.3.1 Einfluss des Bussystems auf die Eigenschaften des Computers

Wenn man die Hardware von Computersystemen in der Praxis miteinander vergleicht, fällt einem auf, dass es unterschiedliche Bussysteme gibt. Manche *embedded systems* haben nur acht Leitungen für ihren Datenbus, Desktop-PCs dagegen meist 64 Leitungen. Genauso ist es mit dem Adressbus. Die ersten PCs, die kompatibel zu heutigen PCs sind, hatten 20 Leitungen für den Adressbus. Heute sind 32 bis 64 Leitungen üblich. Auf der anderen Seite werben Mainboard-Hersteller mit Angaben wie „geeignet für DDR4-4600“. Wie ist der Zusammenhang zwischen diesen Tatsachen?

4.3.2 Datenmenge

Der kleinste in einem Computer existierende Speicherplatz besteht aus einem Element, das wahlweise den Zustand null oder eins annehmen kann. In der Anfangszeit der Computertechnik hat man in Computern vorwiegend Zahlen für wissenschaftliche Berechnungen gespeichert und verarbeitet; in einem solchen kleinsten Element ließ sich dann genau eine Ziffer (im Dualsystem) unterbringen. Aus diesem Grund heißt dieses Element bis heute 1 Bit = 1 b (für *binary digit* = Binärziffer).

Die Größe, die man damit charakterisiert, heißt Informationsmenge bzw. Datenmenge (hier einfach abgekürzt mit DM).

In der Informatik gibt es eine weitere Definition:

$$DM = -\log_2(Wk)\,[\text{Bit}] \tag{1}$$

Die Informations- oder Datenmenge eines Ereignisses ist dort der negative Zweierlogarithmus der Wahrscheinlichkeit eines Ereignisses. Hat ein Ereignis (Kopf oder Zahl bei einem Münzwurf) die Wahrscheinlichkeit $Wk = 0,5$, dann ist der Zweierlogarithmus gleich -1. Die Information dieses Ereignisses beträgt dann $DM = 1\,\text{Bit}$

In der Praxis benutzt man dagegen meist die obere Definition: Ein Bit ist, was zu einem Zeitpunkt in eine binäre Speicherzelle oder auf eine binäre Leitung passt.

Mit einem Bit allein kommt man meist nicht weit. Daher sind Vielfache eingeführt worden, zunächst das Nibble, in dem man eine Dezimalziffer speichern konnte, später das Byte, das sich zur Speicherung eines Buchstaben (richtiger: eines alphanumerischen Zeichens) eignete.

Zur Angabe der Datenmengen auf Festplatten und Hauptspeichern brauchte man noch wesentlich größere Einheiten. Die Hersteller von Festplatten konnten sich auf die aus der Physik bekannten SI-Vorsätze (kilo=1000, Mega=10^6) stützen.

Hauptspeicher dagegen hat aus technischen Gründen meistens Größen, die direkten Zweierpotenzen entsprechen, so etwa 1024, 2048 usw. Wegen der mathematischen Nähe von 1024 zu 1000 haben die RAM-Hersteller einfach definiert: 1 KiloBit = 1 Kb = 1024 Bit. Dabei wurde das K von Kilo groß geschrieben, um es vom gleichnamigen SI-Vorsatz abzugrenzen. Jetzt musste man also aufpassen, ob man 1 kb, 1 kB, 1 Kb oder 1 KB vor sich hatte (entsprechend 1000, 8000, 1024 oder 8192 Bit)! Bei den nächstgrößeren Vorsätzen wie Mega und Giga klappte das schon nicht mehr, so dass von der IEEE neue Einheitenvorsätze für die Informatik vorgestellt wurden, die jeweils in 1024-er-Schritten abgestuft sind: 1 kibiBit = 1 kib = 1024 Bit. Tabelle 1 zeigt eine Übersicht; im mittleren Teil sind die neuen Vorsätze, im unteren Teil die alten SI-Vorsätze aus der Physik.

4.3.3 Datenmenge und Datenrate

Häufig spricht man im Alltag von der Geschwindigkeit einer Übertragung. Die physikalische Größe Geschwindigkeit (Symbol v) ist dabei allerdings nicht gemeint, denn v ist definiert als:

$$v = \frac{\Delta s}{\Delta t} \tag{2}$$

In IT-Systemen soll dagegen keine Strecke zurückgelegt, sondern eine Datenmenge übertragen

Anzahl	Anzahl	Bezeichnung	weitere Bezeichnung
2^3 Bit	8 Bit	1 Byte	1B, 1 Oktett
2^{10} Bit	1024 Bit	1 KibiBit	1 Kib, früher: 1 Kb
2^{20} Bit	1048576 Bit	1 MebiBit	1 Mib
2^{30} Bit	1073741824 Bit	1 GibiBit	1 Gib
2^{40} Bit	1099511627776 Bit	1 TebiBit	1 Tib
$1e3$ Bit	1000 Bit	1 KiloBit	1 kb
$1e6$ Bit	1000000 Bit	1 MegaBit	1 Mb
$1e9$ Bit	1000000000 Bit	1 GigaBit	1 Gb
$1e12$ Bit	1000000000000 Bit	1 TeraBit	1 Tb

Tabelle 1: Vielfache der Maßeinheit Bit

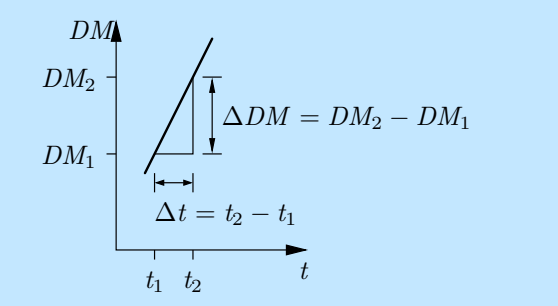

Abbildung 1: Definition der Datenrate

werden. Deshalb spricht man besser von der Datenrate DR (oder auch Übertragungsrate). Sie ist definiert als (Abbildung 1):

$$DR = \frac{\Delta DM}{\Delta t} \tag{3}$$

Die Einheit der Datenrate ist $[DR] = \frac{Bit}{s}$.

In der Praxis sieht es allerdings so aus: Am Beginn einer Übertragung erhält der Empfänger einige Zeit nichts. Dieses Zeit nennt man *Totzeit*. Die Totzeit t_0 (oder Latenz) ist definiert als die Zeit, die gebraucht wird, bis das erste Bit einer Übertragung am Empfänger angekommen ist.

Nach dieser Zeit ist die Datenrate oft annähernd konstant (Abbildung 2). Es ergibt sich dann

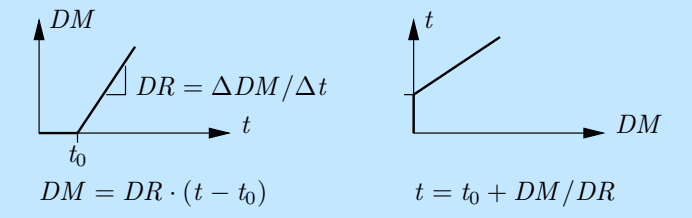

Abbildung 2: Zusammenhang Datenmenge, Datenrate und Totzeit

der Zusammenhang:

$$DM = \begin{cases} 0 & : \quad t \leq t_0 \\ DM \cdot (t - t_0) & : \quad t > t_0 \end{cases} \tag{4}$$

Wenn man nach t umformt, erhält man

$$t = t_0 + \frac{DM}{DR} \tag{5}$$

Beispiel: Das Paket `FreeBSD-9.2-RELEASE-amd64-dvd1.iso` hat die Größe $DM = 2{,}4\,\text{GB}$.

Wie lange dauert der Download bei einer Datenrate $DR = 6000\,\text{kb/s}$ und $t_0 = 5\,\text{s}$?

$$t = t_0 + \frac{DM}{DR} = 5\,\text{s} + \frac{2{,}4 \cdot 10^9 \cdot 8\,\text{b}}{6000 \cdot 10^3\,\text{b/s}} = 5\,\text{s} + 3200\,\text{s} = 53{,}4\,\text{min}$$

4.3.4 Systemgeschwindigkeit und Busdatenrate

Die Arbeitsgeschwindigkeit von PCs hängt von vielen Faktoren ab. Zuerst ist der Benutzer zu nennen, der nur langsam seine Daten eingibt und nur langsam die Ergebnisse lesen und verstehen kann. Dann ist die Software zu nennen, die im Sinne hoher Geschwindigkeit günstig oder ungünstig programmiert sein kann. Bei der Hardware hat die Geschwindigkeit der Massenspeicher einen großen Einfluss, denn ein Zugriff auf ein einzelnes Bit auf einer Festplatte dauert etwa 1000000 mal so lange wie ein Zugriff auf ein Bit im RAM[1].

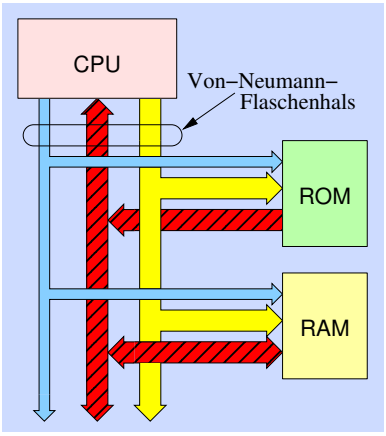

Abbildung 3: Von-Neumann-Flaschenhals

Der nächste wichtige Faktor ist die Übertragung von Daten über das Bussystem zwischen CPU auf der einen Seite und RAM (und ROM) auf der anderen Seite. Hier müssen sehr viele Daten in kurzer Zeit übertragen werden. Bei Von-Neumann-Rechnern spricht man sogar vom *Von-Neumann-Flaschenhals*, weil über diese Strecke sowohl Befehle als auch Daten übertragen werden (Abbildung 3). Aber auch in der Harvard-Architektur ist die Datenrate zwischen CPU und RAM (und ROM) wichtig. Deshalb ist es sinnvoll, sie zu berechnen.

4.3.5 Einfluss der Bustaktfrequenz

Wie sieht der Ablauf eines Bus-Zyklus aus? Zuerst wird auf den Adressbus zugegriffen (eine Adresse aufgelegt), dann auf den Steuerbus und schließlich auf den Datenbus.

Es gibt synchronen und asynchronen Zugriff. Beim asynchronen Zugriff orientiert man sich an Steuersignalen: Wenn das RAM fertig ist und ein Datenwort auf den Datenbus legen kann, dann meldet es das über eine Steuerleitung. Daraufhin kann die CPU das Datenwort einlesen. Man nennt diesen Mechanismus *handshake*.

Aktuelle PCs benutzen dagegen den synchronen Zugriff. Hier wird der Ablauf durch das Bustaktsignal getaktet. Das Bustaktsignal ist eine einfache rechteckförmige Wechselspannung. Jeweils an einem festen Punkt dieser Wechselspannung (beim Wechsel von 0 nach 1, genannt *positive Taktflanke*) kann ein Buszugriff begonnen werden (Abbildung 4). Die Zeitdauer zwischen zwei benachbarten Flanken heißt Bustaktdauer T_B und wird meist in Nanosekunden ($1\,\text{ns} = 1 \cdot 10^{-9}\,\text{s}$) angegeben.

Statt der Bustaktdauer T_B wird jedoch häufig die Bustaktfrequenz f_B angegeben. Eine Bustaktfrequenz $f_B = 10\,\text{MHz}$ bedeutet: Maximal 10 Millionen Ereignisse sind pro Sekunde auf dem

[1] Wenn man sehr viele aufeinanderfolgende Daten von der Festplatte liest, holt die Festplatte etwas auf.

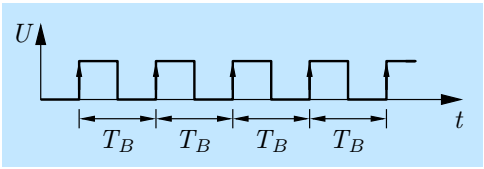

Abbildung 4: Bustaktsignal, Bustaktdauer und Bustaktfrequenz

Bussystem möglich. Damit kann man bei optimaler Ausnutzung maximal 10 Millionen Speicherzugriffe pro Sekunde auf diesem Bussystem erreichen.

Die Bustaktfrequenz f_B ist der Kehrwert der Bustaktdauer T_B:

$$f_B = \frac{1}{T_B} \tag{6}$$

Beispiel: $f_B = 233\,\text{MHz}$; dann ist

$$T_B = \frac{1}{f_B} = \frac{1}{233 \cdot 10^6\,\text{Hz}} = 4{,}29\,\text{ns}$$

Das ist eine sehr kurze Zeit: In 1 ns legt das Licht gerade 30 cm zurück; man sagt, eine Nanosekunde sei ein „Licht-Fuß“. Entsprechend kürzer müssten die Wege auf dem Mainboard eigentlich sein!

4.3.6 Einfluss der Datenbusbreite

Die Breite des Datenbusses (hier einfach mit *DBB* abgekürzt) gibt an, wie viele Bit bei jedem Buszugriff zugleich geschrieben oder ausgelesen werden. Je höher diese Zahl, desto mehr Bit können pro Sekunde über den Bus verschoben werden:

- [Datenbusbreite] $= \frac{\text{Bit}}{\text{Buszugriff}}$
- [Bustaktfrequenz] $= \frac{\text{Buszugriffe}}{\text{Sekunde}}$
- [Busdatenrate] $= \text{Bit/s}$
- [Datenbusbreite] $*$ [Bustaktfrequenz] $= \text{Bit/Buszugriff} * \text{Buszugriffe/s} = \text{Bit/s}$

Damit kommt man auf eine Formel für die Bus-Datenrate DR_B:

$$DR_B = f_B \cdot DBB \tag{7}$$

Also sind sowohl Datenbusbreite als auch Bustaktfrequenz entscheidend für die Geschwindigkeit eines Bussystems verantwortlich.

Besonders beim Bus zwischen CPU und RAM (dem so genannten *front side bus*) beeinflusst die Busdatenrate entscheidend die Geschwindigkeit des gesamten PCs.

Daher hat man vereinbart, dass die Datenbusbreite am Ausgang der CPU aussagt, ob es sich um ein 16-, 32-, 64-Bit-Computersystem handelt: Hat die CPU 32 Datenbusleitungen, ist es eine 32-Bit-CPU und damit ein 32-Bit-Computer (die Etikettierung der CPU-Hersteller sollte man kritisch beobachten).

Keinen Einfluss hat die Datenbusbreite darauf, ob in dem System auch größere Zahlen verarbeitet werden können. Schon seit Anfang an können Mikroprozessoren größere Zahlen (und sonstige Daten) stückweise nacheinander aus dem Speicher holen, verarbeiten und ablegen. Es dauert nur bei schmalem Datenbus (wie die Formel dies zeigt) entsprechend länger.

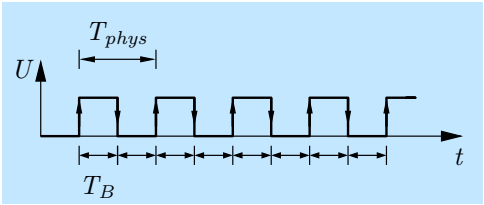

Abbildung 5: Bustaktsignal und Bustaktdauer bei DDR

4.3.7 DDR-Mainboards und DDR-RAM

Beim Zugriff auf DDR-RAM wird jede Bustaktdauer doppelt ausgenutzt: Auch bei einer fallenden Taktflanke kann ein Bustakt begonnen werden (Abbildung 5). Damit halbiert sich die Bustaktdauer, verglichen mit der physikalischen Periodendauer (die sich dadurch ja nicht ändert):

$$T_B = \frac{T_{phys}}{2} \tag{8}$$

Die Bustaktfrequenz verdoppelt sich entsprechend im Vergleich zur physikalischen Frequenz:

$$f_B = f_{phys} * \cdot 2 \tag{9}$$

Mit der doppelten Bustaktfrequenz erhält man auch die doppelte Busdatenrate. Daher kommt auch der Name: DDR ist eine Abkürzung für *double data rate*. Auf diese Weise können vergleichsweise „langsame“ Mainboards schnelle Speicherbausteine ansteuern.

4.3.8 Einfluss der Adressbussbreite

Beim Vergleich Bank – Computer wurde Folgendes gegenübergestellt:

- Adresse - Kontonummer
- Daten - Kontostände

Damit kann man vergleichen:

- Vergleich: Adressbusbreite - Stellenzahl/Breite der Kontonummer

Mit der Stellenzahl der Kontonummer wird die maximale Anzahl der Konten (und damit der Kontostände) begrenzt. Wie wirkt sich entsprechend die Stellenzahl der Adresse aus? **Mit der Adressbusbreite wird die maximale Anzahl der Datenworte festgelegt.**

4.3.9 Maximale Anzahl der Datenworte

Beispiel Kontostände: Dezimalsystem, Basis 10

- AKN:Maximale Anzahl der Kontonummern
- BKN:Breite der Kontonummern

BKN	AKN
1	$10=10^1$
2	$100=10^2$
3	$1000=10^3$
4	$10000=10^4$

Man erhält: $AKN = 10^{BKN}$

Adressen: Binärsystem, Basis 2

- ASW: Maximale Anzahl der Datenworte (Speicherworte)
- ABB: Breite des Adressbusses

ABB	ASW
1	$2{=}2^1$
2	$4{=}2^2$
3	$8{=}2^3$

Man erhält also als Ergebnis: $ASW = 2^{ABB}$

Bei einem Bussystem, das Geräte adressiert, kann man also für die maximale Systemgröße angeben:

$$MSG = 2^{ABB}\,[Ger.] \tag{10}$$

Beispiel: Der GPIB (*general purpose interface bus*) hat 8 Adressbits und kann damit $MSG = 2^{ABB} = 2^8 = 256$ Geräte adressieren.

Bei einem Bussystem, das Datenworte im RAM (Speicherworte) adressiert, erhält man für jeden Zugriff ein Datenwort:

$$MSG = 2^{ABB}[\mathrm{DW}] \tag{11}$$

Jedes Datenwort hat so viele Bits, wie der Datenbus breit ist:

$$1\,\mathrm{DW} = \mathrm{DBB} \cdot 1\,\mathrm{b} \tag{12}$$

Damit bekommt man für die zweite Formel [2]:

$$MSG = 2^{ABB} \cdot \mathrm{DBB} \cdot 1\,\mathrm{b} \tag{13}$$

Beispiel: Der C64-Computer hatte wie viele andere Systeme seiner Zeit $ABB = 16$ und $DBB = 8$. Damit war $MSG = 2^{ABB} \cdot DBB \cdot 1\,\mathrm{b} = 2^{16} \cdot 8\,\mathrm{b} = 524288b = 64\,\mathrm{kiB}$. Daher rührte auch sein Name.

Wichtig für diese Größe ist die Tatsache, dass es sich um ein absolutes Maximum handelt. Wenn man an eine solche Grenze stößt, hilft meistens nur ein völliges Neu-Design. Und das kann auch für eine große Firma oder Einrichtung fatal sein. Regel für den Adressbus sollte also immer sein, sehr großzügig mit seiner Breite umzugehen. Weitere Beispiele für zu knapp bemessene Adressen sind IP, Version 4 oder auch das längst vergangene Jahr-2000-Problem in vielen Programmen und Geräten – basierend auf dem Gedanken, dass zwei Dezimalziffern zur Adressierung einer Jahreszahl doch *ewig* reichen müssten...

[2] Bei PCs wird aus Gründen der *Kompatibilität* mit jeder Adresse ein Byte adressiert. Bei einem 32-Bit-Zugriff auf die Adresse 20 erhält man die Inhalte der Adressen 20, 21, 22 und 23. Bei einem Zugriff auf die Adresse 21 bekommt man die Inhalte von 21, 22, 23 und 24. Natürlich ist das nicht schlimm: Wenn man alle Inhalte braucht, greift man einfach nur auf die Adressen 20, 24, 28 usw. zu. Aber es hat Einfluss auf die maximale Systemgröße (man verschwendet ja drei Viertel aller Adressen). Bei PCs muss man bei der obigen Formel daher mit $DBB = 8\,\mathrm{Bit}$ rechnen, ganz egal, ob es sich um ein 16-, 32- oder 64-Bit-System handelt.