

**Unterlagen
zur
Vorlesung**

Rechnerstrukturen

**Martin Lehmann
Dietmar P. F. Möller**

Wintersemester 2008/2009

Inhalt Rechnerarchitektur:

Grundbegriffe der Informationsverarbeitung:

**Information,
Repräsentation von Information,
Codierung,
Zahlssysteme.**

Rechnerarithmetik:

**Operationen mit Zahlen,
Ganzzahl-Arithmetik,
Gleitkomma-Arithmetik.**

Lokale und globale Konzepte der Digitaltechnik:

**Boolesche Algebra,
Automatentheorie,
Schaltnetze,
synchrone und asynchrone Schaltwerke,
kooperierende Schaltwerke (Operationswerk,
Steuerwerk),
Datenpfade,
Datentore,
Bus-Systeme,
Zähler,
Register,
Registerfiles,
Speicher.**

von Neumann-Rechner:

**Architektur,
Befehlssatz,
Maschinen- und Assembler-Code,
Betriebssystemebene.**

Speicherhierarchie:

**Speichertypen,
virtueller Speicher,
Cache-Organisation.**

Pipelining:

**CISC und RISC,
superskalare Architekturen.**

Parallele Architekturen

Kommunikation, Schnittstellen und Peripheriegeräte

**Bemerkung: Die Veranstaltung folgt dieser
Inhaltsangabe nicht sequentiell.**

Literatur:

**R. E. Bryant, D. R. O'Hallaron: Computer Systems:
A Programmer's Perspective, Prentice Hall, 2003,
ISBN-10: 0-13-034074-X.**

**M. Dal Cin: Rechnerarchitektur, Teubner, 1996,
ISBN-10: 3-519-02941-3.**

**E. K. Giloi: Rechnerarchitektur, Springer, 1999,
ISBN-10: 3-540-56355-5.**

**J. L. Hennessy, D.A.Patterson, Computer Architecture:
A Quantitative Approach, 4th ed., Morgan
Kaufman / Elsevier, 2007,
ISBN-13: 978-0-12-370490-0.**

**V. P. Heuring, H.F. Jordan: Computer Systems Design
and Architecture, 2nd ed., Pearson Prentice Hall,
2004, ISBN-10: 0-13-191156-2.**

**D. A. Patterson, J. L. Hennessy: Computer Organization
and Design, 3rd ed., Morgan Kaufmann, 2005,
ISBN-10: 1-55860-604-1.**

**W. Schiffmann, R. Schmitz: Technische Informatik 1,
Springer, 2004, ISBN-10: 3-540-40418-X.**

**W. Schiffmann, R. Schmitz: Technische Informatik 2,
Springer, 2002, ISBN-10: 3-540-43854-8.**

G. H. Schildt, D. Kahn, C. Kregel, C. Moerz:
Einführung in die Technische Informatik, Springer,
2005, ISBN-10: 3-211-83853-8.

D. A. Patterson, J. L. Hennessy: Computer Organization
and Design, Third Edition, Morgan Kaufmann,
2005, ISBN-10: 1-55860-604-1.

W. Stallings: Computer Organization and Architecture:
Designing for Performance, Sixth Edition, Prentice-
Hall International, 2003, ISBN-10: 0-13-035119-9.

A. S. Tanenbaum: Computerarchitektur, Pearson
Studium, 2006, ISBN-13: 978-3-8273-7151-5.

A. S. Tanenbaum: Structured Computer Organization,
Fifth Edition, Pearson Prentice Hall, 2006,
ISBN-10: 0-13-148521-0.

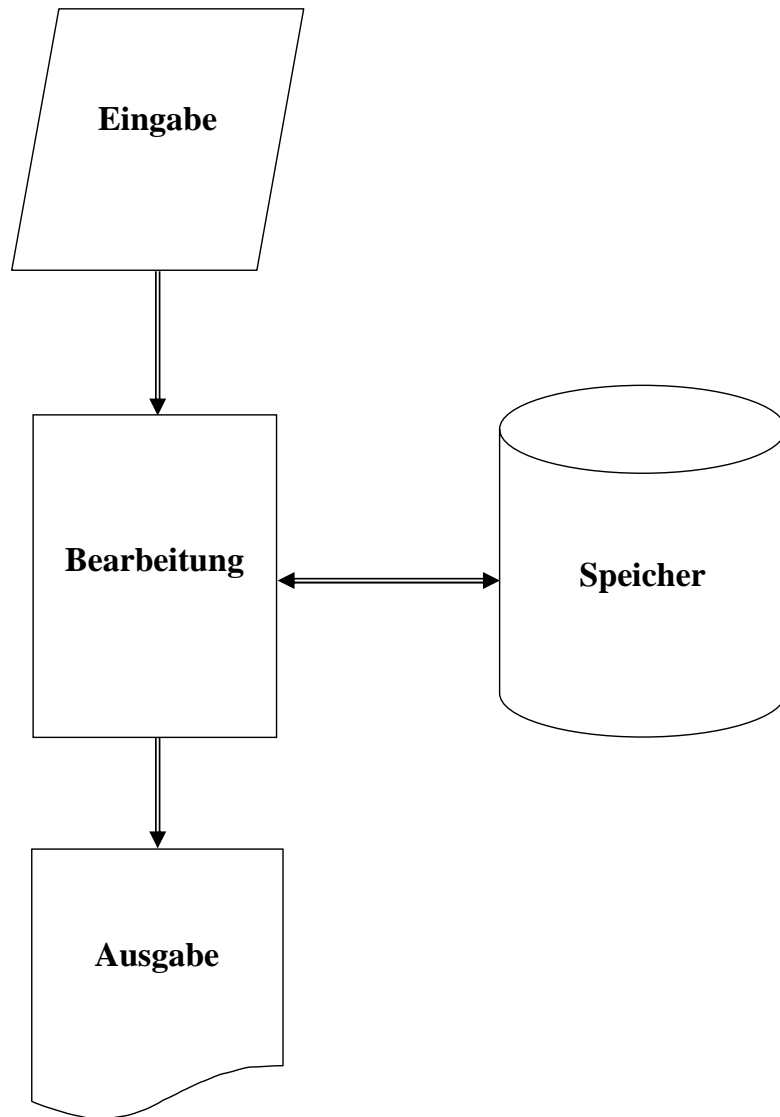
Literatur zur Sparc:

The SPARC architecture manual : version 8
SPARC International, Inc.
Englewood Cliffs, N.J. : Prentice Hall, 1992
ISBN-10: 0-13-825001-4

Das Sparc-Manual ist auch erhältlich über
<http://www.sparc.org/>
Rubrik: Standard Specs

- 1 Einleitung**
- 1.1 Nutzung einer Rechenanlage**
- 1.2 Leistungssteigerung von Rechnern**
- 1.3 "Moore's Law"**
- 1.4 Tabelle von Reddy und Newell**
- 1.5 Amdahls Gesetz**
- 1.6 Klassifikation nach Flynn**
- 1.7 ILLIAC IV**
- 1.8 Beschreibungsebenen**
- 1.9 Von Neumann Architektur**
- 1.10 Präfixe für Maßangaben**

Ein Modell der Datenverarbeitung:



Aufgaben einer Rechanlage:

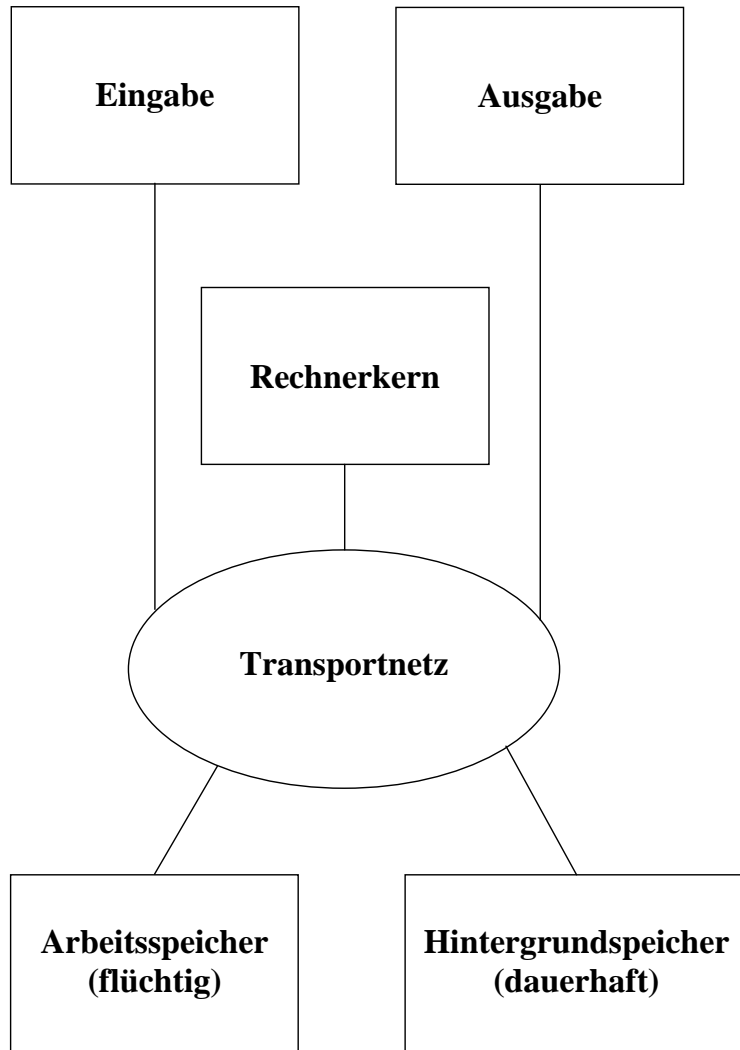
- **Verarbeitung von Daten,**
- **Speicherung von Daten,**
- **Transport von Daten zwischen Rechner und Außenwelt,**
- **Kontrolle obiger Operationen.**

Es lassen sich grob vier Ebenen der Bearbeitung von Daten unterscheiden.

- (i) **Datenverarbeitung**
- (ii) **Informationsverarbeitung**
- (iii) **Wissensverarbeitung**
- (iv) **Gedankenbearbeitung**

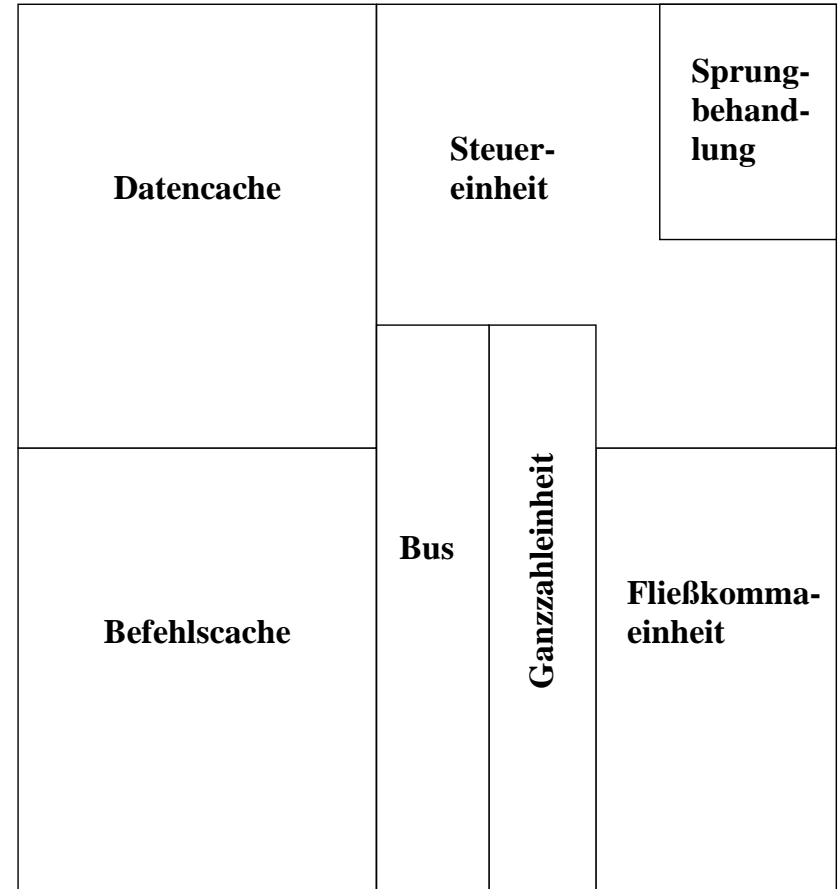
Bemerkung: Die Anforderung an die Rechenleistung nimmt mit der Verarbeitungsebene zu.

Ein Modell einer Rechenanlage:

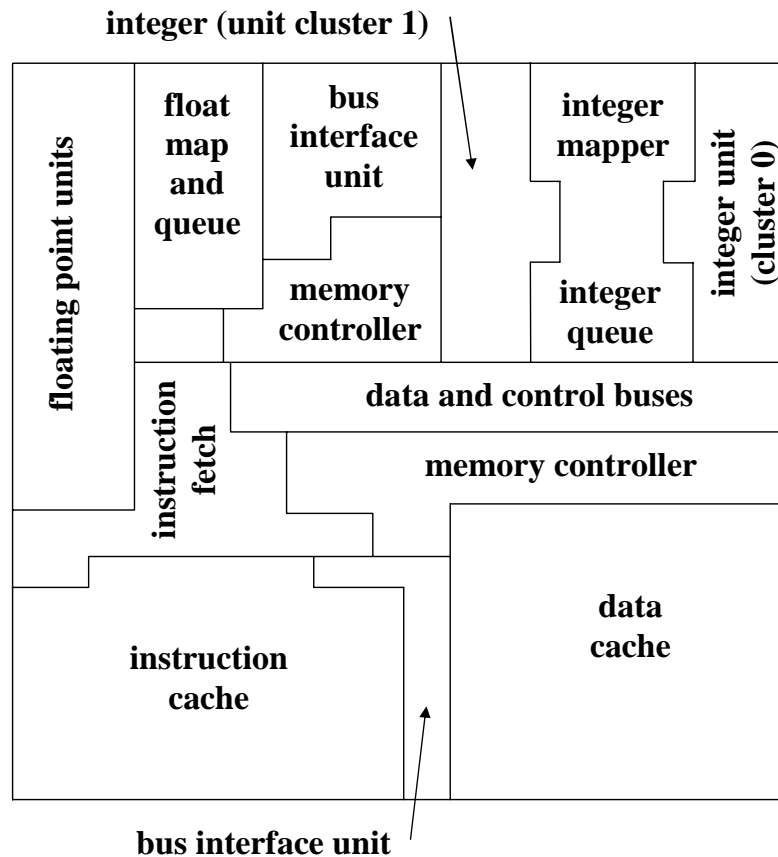


Bemerkung: In einer wohlausgewogenen Rechenanlage sind alle Einheiten vollbeschäftigt.

Bild eines Prozessors auf einem Chip:



Haupteinheiten eines Alpha 21264 Mikroprozessors:



Bemerkung: In einem Alpha 21264 können sich gleichzeitig bis zu 80 Befehle in verschiedenen Stadien der Bearbeitung befinden. Dies erschwert die Leistungsbestimmung für 21264-Prozessoren.

Quelle: R. E. Kessler: The Alpha 21264 Microprocessor, IEEE Micro, Vol. 19, No. 2, pp. 24 - 36

In ihrem Buch "Parallel Computers 2" geben Hockney und Jesshope einen Faktor von 10 in 5 Jahren für die Steigerung der Rechengeschwindigkeit von Computern an. Diesen Wert leiten sie ab aus der Veränderung der Ausführungsgeschwindigkeit der Multiplikationsoperation im Zeitraum 1950 bis 1975.

Jahr	Steigerung
0	1.00
1	1.58
2	2.51
3	3.98
4	6.31
5	10
6	15.85
7	25.12
8	39.81
9	63.10
10	100
11	158.49
12	251.19
13	398.11
14	630.96
15	1000
20	10000
30	1000000
40	100000000

Bemerkung: Jeder exponentielle Verlauf stößt irgendwann an natürliche Grenzen. Nehmen wir die Multiplikationszeit einer UNIVAC I in 1951, sie beträgt 2150 µs. Im Jahre 2001 hätte eine normale Rechenanlage eine Multiplikationszeit von $2,15 * 10^{-3-10}$ s haben sollen, ein Schätzfehler etwa um den Faktor 1000.

In ihrem Buch "Computer Architecture, 4th ed." geben Hennessy und Patterson jährliche Steigerungsraten für die Leistung von Prozessoren an.

Zeitraum	Faktor
1978 – 1986	1,25 (1,225?)
1986 – 2002	1,52
2002 – 2006	1,20

Zitat aus H & P: "The increase in growth since then (1986) is attributable to more advanced architectural and organizational ideas."

Steigerungen seit 1978:

1978:	1
1979:	1,225
1980:	1,5
1986:	5,1
1990:	27
1995:	219
2000:	1781
2002:	4117
2003:	4940
2004:	5928
2005:	7114

Leistungssteigerung der Spitzenrechner seit 1993:

Jahr	Rechner	Linpackleistung in Gflop/s	Zahl der Kerne
1993	CM-5, TMC	60	1.024
1994	Intel Paragon XP/S MP	143	3.680
1995	Fujitsu NWT	170	140
1996	Hitachi SR2201/1024	220	1.014
1997	Intel ASCI Red	1.068	7.264
1998	Intel ASCI Red	1.338	9.152
1999	Intel ASCI Red	2.121	9.472
2000	Intel ASCI Red	2.379	9.632
2001	ASCI White, IBM	7.226	8.192
2002	Earth Simulator, NEC	35.860	5.120
2003	Earth Simulator, NEC	35.860	5.120
2004	Earth Simulator, NEC	35.860	5.120
2005	Blue Gene/L, IBM	136.800	65.536
2006	Blue Gene/L, IBM	280.600	131.072
2007	Blue Gene/L, IBM	280.600	131.072
2008	Roadrunner, IBM	1.026.000	122.400

Daten aus den TOP500-Listen, jeweils vom Juni, Jahresfaktor etwa 1,92.

"Moore's Law":

Im Jahre 1965 formulierte Gordon E. Moore in dem Artikel: "Cramming More Components Onto Integrated Circuits" im Magazin Electronics (Vol. 38, No. 8, April 19, 1965) ein Wachstumsgesetz für integrierte Schaltkreise.

Zitat:

"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000."

Benutzte Datenbasis:

1962	8
1963	16
1964	32
1965	64

Für 1975 erwartet: 65.536

Diese optimistische Vorhersage traf eigentlich nicht ein.

Später korrigierte Moore seine Vorhersage zu:

$$(\text{circuits per chip}) = C_{1975} * 2^{(\text{year} - 1975) / 1,5}$$

Allgemeine Interpretation:

Die zu einem festen Preis kaufbare Rechenleistung verdoppelt sich alle 18 Monate.

Umgangssprachliche Formulierung von Moores Vermutung:

Bei konstanten Kosten verdoppelt sich die Rechenleistung eines Mikroprozessors alle 18 Monate.

Bemerkungen:

- (i) Moores Vermutung bestätigt sich weitgehend seit 1975.
- (ii) Moores ursprüngliche Aussage betrifft die Steigerungsrate für die Anzahl der Schaltkreise auf einem Chip.
- (iii) Moores Vermutung wird allgemein als "Moore's Law" bezeichnet.

Formel:

$$L(t) = L(0) * 2^{t/18}$$

Die Zeit t wird in Monaten gemessen.

L(t) = Leistung zum Zeitpunkt t.

L(0) = Leistung zum Zeitpunkt 0.

Einige Formelwerte:

Jahr 0:	1
Jahr 1:	1,59
Jahr 2:	2,52
Jahr 3:	4
Jahr 5:	10,08
Jahr 6:	16
Jahr 7:	25,4
Jahr 8:	40,32
Jahr 9:	64
Jahr 10:	101,59
Jahr 11:	161,27
Jahr 12:	256
Jahr 13:	406,38
Jahr 14:	645
Jahr 15:	1024
Jahr 18:	4096
Jahr 20:	10321
Jahr 25:	104032
Jahr 30:	1048576
Jahr 40:	106528681
Jahr 50:	10822639410
Jahr 60:	1099511627776
Jahr 70:	111703418533305

Illustration zu Moores Beobachtung:

**Beispiel: Entwicklung der Intelprozessoren
(Daten von Intels Homepage)**

Jahr	Rechner	Transistorenzahl
1971	4004	2.300
1972	8008	2.500
1974	8080	4.500
1978	8086	29.000
1982	286	134.000
1985	i386	275.000
1989	i486	1.200.000
1993	Pentium	3.100.000
1997	Pentium II	7.500.000
1999	Pentium III	9.500.000
2000	Pentium 4	42.000.000
2001	Itanium	25.000.000
2002	Itanium 2	220.000.000
2004	Itanium 2	592.000.000

Eine Lösung der Gleichung $x^{33} = 592000000 / 2300$
ergibt $x = 1,45867$.

Nutzung von Moores Vermutung für schöpferische Pause:

Beispiel: Die Lösung einer Rechenaufgabe dauert bei gegenwärtiger Rechenleistung 4 Jahre. Ein mögliches Vorgehen ist das folgende. Man wartet 3 Jahre, kauft dann einen neuen Rechner und erledigt die Rechenaufgabe in einem Jahr.

Rechnung: Rechenaufwand = A
 Gegenwärtige Leistung = L
 Zeitdauer = t = A / L
 Bei Rechenstart nach s Monate beträgt die Rechenleistung $L * 2^{s/18}$.

Man erhält die Gleichung

$$A / L = s + A / (L * 2^{s/18})$$

Setzen von L = 1 ergibt:

$$A = s + A / 2^{s/18}$$

Vereinfachen führt zu:

$$A = s / (1 - 2^{-s/18})$$

Einige Werte:	s Monate	A Monate
	6	29,839
	12	32,429
	18	36
	24	39,7911
	36	48
	48	56,9726
	60	66,6084
	120	121,193

Geschwindigkeitssteigerungen beim Problemlösen:

Tabelle aus
R. Reddy und Allen Newell: Multiplicative Speedup of Systems, Perspectives on Computer Science
 (Hrg. A. Jones), Academic Press, 1977.

Beobachtete Steigerungsfaktoren der Ausführungsgeschwindigkeit von Programmen auf Grund von Spartenfortschritten:

Technologie	20 bis 50
Architektur	2 bis 10
Systemsoftware	2 bis 10
Programmorganisation	2 bis 20
Algorithmen	2 bis 20
Reimplementation	2 bis 10
Problemwissen	10 bis 100
Heuristiken	10 bis 100

Bemerkung: Falls die einzelnen Steigerungen unabhängig voneinander sind, sind Geschwindigkeitssteigerungen um bis zu elf Zehnerpotenzen möglich.

Amdahls Gesetz:

Im April 1967 hielt G. Amdahl, der Architekt der Rechnerreihe IBM/360, einen Vortrag etwa folgenden Inhalts.

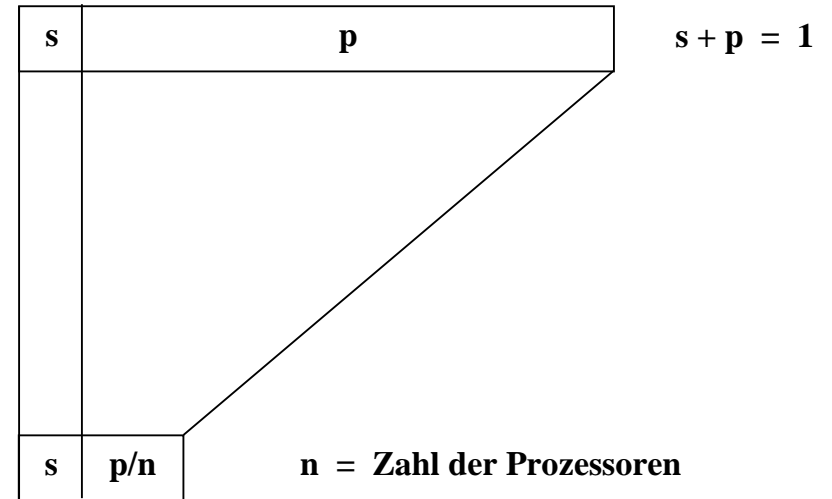
1. Seit 10 Jahren wird behauptet, daß der Monorechner seine "natürlichen" Grenzen erreicht hat, wirklicher Fortschritt ist nur im Bereich der Multirechner zu erwarten.
2. Es ist schwer einzusehen, daß man mit einem Parallelrechner im allgemeinen eine Geschwindigkeitssteigerung um mehr als einen Faktor 7 gegenüber einem Monorechner erreichen kann.

Ein Beispiel: Laufzeiten eines Programms auf verschiedenen Architekturen:

80386 mit 80387	1.181 sec.
80386 mit Weitek (2 MFLOPS)	515 sec.
80386 mit 4 T800 Transputern (4 * 1,5 MFLOPS)	
<direkte Übertragung>	1.652 sec.
<Überlappung E/A und Rechnung>	770 sec.
<Maschinencode für Kernschleife>	254 sec.

Daten aus Jon Flower & al.: Finite-Element Analysis on a PC, IEEE Software, Sept. 91, p.50-57

Jede Aufgabe besteht aus einem seriellen Teil s und einem parallelisierbaren Teil p .



Geschwindigkeitssteigerung:

$$b(s,n) = \frac{s + p}{s + \frac{p}{n}} = \frac{1}{s + \frac{p}{n}}$$

$$b^*(s) = \lim_{n \rightarrow \infty} b(s,n) = \frac{1}{s}$$

Beispiele:

$b^*(0,2)$	=	5
$b^*(0,1)$	=	10
$b^*(0,01)$	=	100
$b^*(0,001)$	=	1000

Amdahls Formel beantwortet Fragen nach dem Gesamtgewinn, falls man nur einen Teil eines Systems verbessern kann.

Beispiel:

Man habe einen Prozessor für Fließkommarechnung, 20% der ausgeführten Operationen sind Quadratwurzeloperationen. Man steht nun vor folgender Entscheidung:

- (i) Steigerung der Ausführungsgeschwindigkeit der Wurzeloperation um den Faktor 5,**
- (ii) Steigerung der Ausführungsgeschwindigkeit aller Operationen um 15%.**

Berechnung:

Fall (i):
$$\frac{1}{(1-0,2)+\frac{0,2}{5}} = \frac{1}{0,8+0,04} = 1,19$$

Fall (ii):
$$\frac{1}{0,85} = 1,18$$

Bemerkung: Die eigentliche Schwierigkeit bei der Anwendung von Amdahls Formel besteht in der Definition des zu berücksichtigenden Gütemaßes.

Klassifizierung nach Flynn:

		Datenstrom:	
		einfach	mehrfach
Befehlsstrom:	einfach	SISD	SIMD
	mehrfach	MISD	MIMD

Die Abkürzungen bedeuten:

SISD "Single instruction stream, single data stream"

SIMD "Single instruction stream, multiple data streams"

MIMD "Multiple instruction streams, multiple data streams"

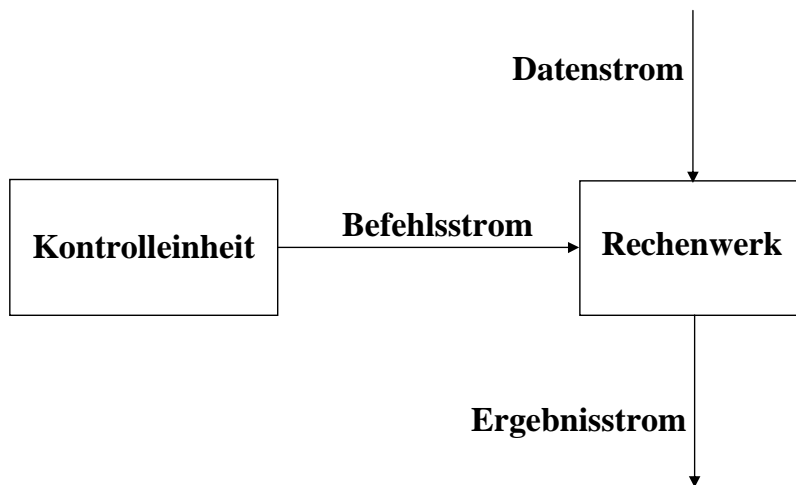
MISD "Multiple instruction streams, single data stream"

Bemerkungen:

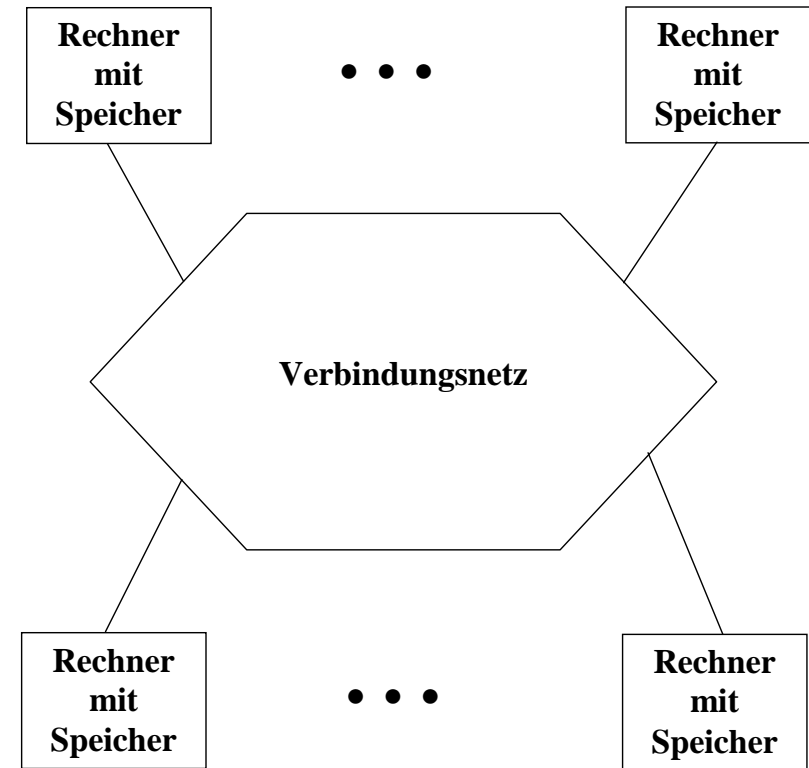
- (i) Die Abgrenzung von SISD gegen die anderen Klassen ist schwierig, z. B. wegen Eigenschaften wie Superskalarität u. ä.**

- (ii) **SISD** scheint bis heute (2008) die erfolgreichste Architektur zu sein.
- (iii) **SIMD** entspricht dem Matrizenkalkül der Algebra, es existieren vielfältige Anwendungen.
- (iv) Es fehlen allgemein anerkannte Modelle für **Mega-MIMD**.
- (v) **Realisierungen von MISD** sind als Exoten zu betrachten.

Bild eines SISD-Rechners:

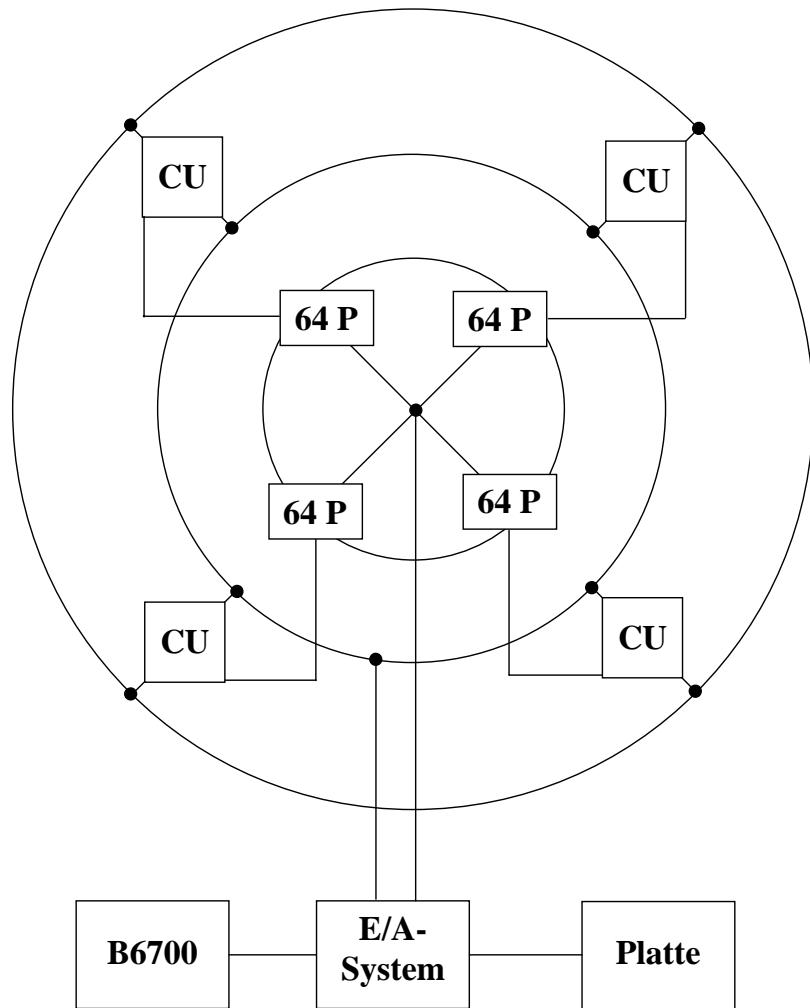


Die Klasse MIMD unterteilt man in zwei Subklassen, Shared Memory MIMD und Distributed Memory MIMD.



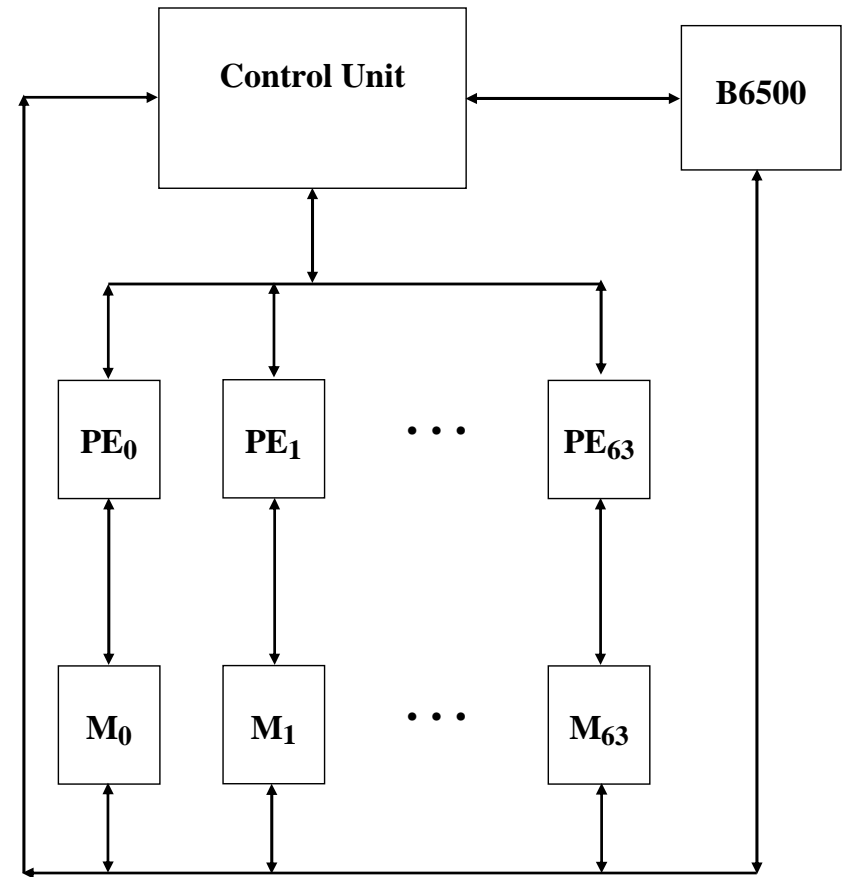
Bemerkung: 42% der Rechner der TOP500-Liste von Juni 2005 nutzen Gigabit Ethernet als ein Verbindungsnetz.

Ursprüngliche Planung für ILLIAC IV:



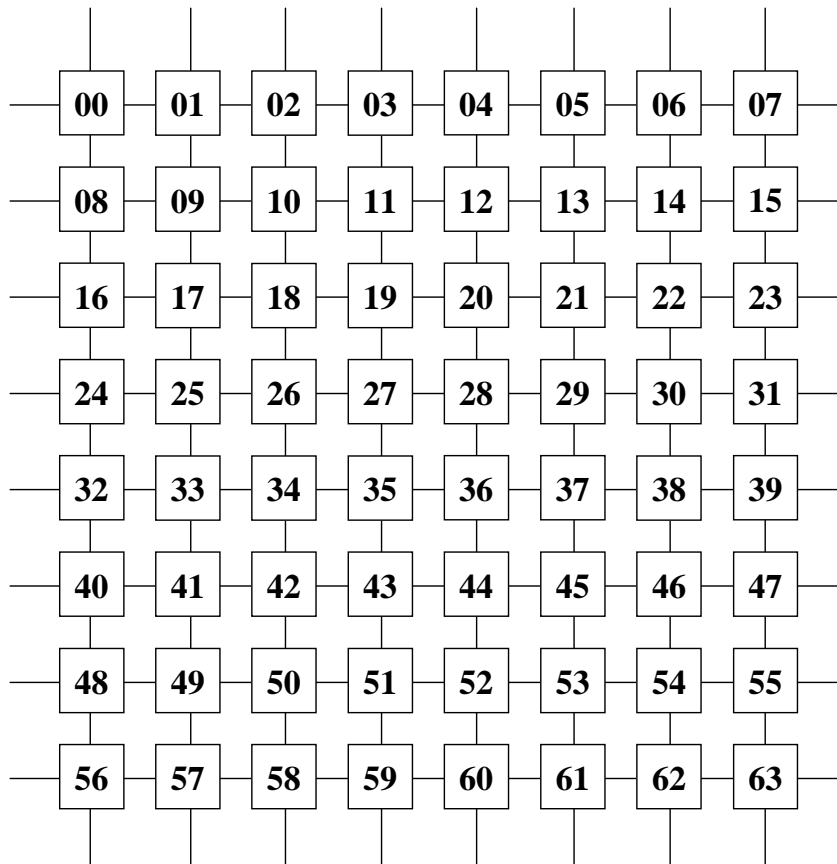
ILLIAC = Illinois Array Computer
CU = Control Unit
64 P = 64 Prozessoren

Blockbild eines gebauten Quadranten des ILLIAC IV:



Bemerkung: Die Fortran Schleife
DO 10 I = 1, 64
10 A(I) = B(I) + C(I)
wird in drei Assemblerbefehle übersetzt:
LDA ALPHA + 2
ADRN ALPHA + 1
STA ALPHA

Das quadratische Feld der "Processing Elements":



Routing Formeln:

- $P_x = (P_y + 1) \bmod 64$
- $P_x = (P_y - 1) \bmod 64$
- $P_x = (P_y + 8) \bmod 64$
- $P_x = (P_y - 8) \bmod 64$

Einige Leistungsdaten der ILLIAC IV:

Angaben in MFLOPS = Millions of Floating Point Operations per Second

Beispiel 1: Addition zweier Vektoren

$$E (*) = A (*) + B (*)$$

Vektorlängen:

5	10	50	100	500	1000
1,45	2,91	14,53	18,13	24,90	25,88

Beispiel 1: Addition dreier Vektoren

$$E (*) = A (*) + B (*) + C (*)$$

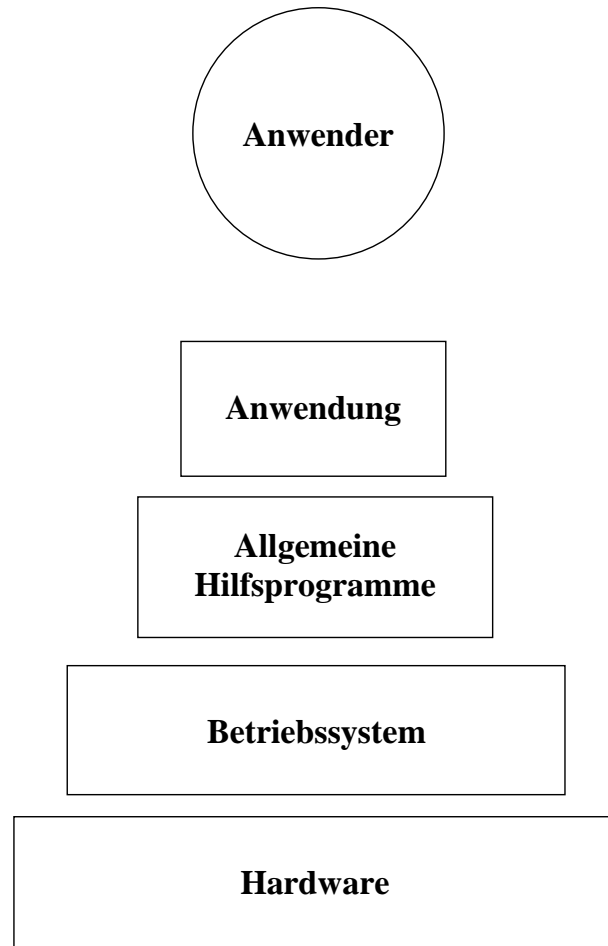
Vektorlängen:

5	10	50	100	500	1000
2,4	4,8	24,0	26,27	38,8	40,0

Frage: Wie sind diese Zahlen zu erklären, da die ILLIAC IV eine Präferenz für Vektorlängen-Vielfache von 64 besitzt?

Antwort: Innerhalb der Kontrolleinheit der ILLIAC IV existiert Simultanarbeit.

Beschreibungsebenen:



Bemerkung: Jede dieser Ebenen lässt sich in Unterebenen gliedern.

Vier Beschreibungsebenen für Rechenanlagen:

Globalstruktur:

**Beschreibung der Systemstruktur durch
Prozessoren,
Speicher,
Steuerungen,
Verbindungsnetze.**

Prozessorebene:

**Festlegung von Architekturdetails wie
Schnittstellen,
Befehlssätze,
Datenrepräsentationen.**

Registertransferebene:

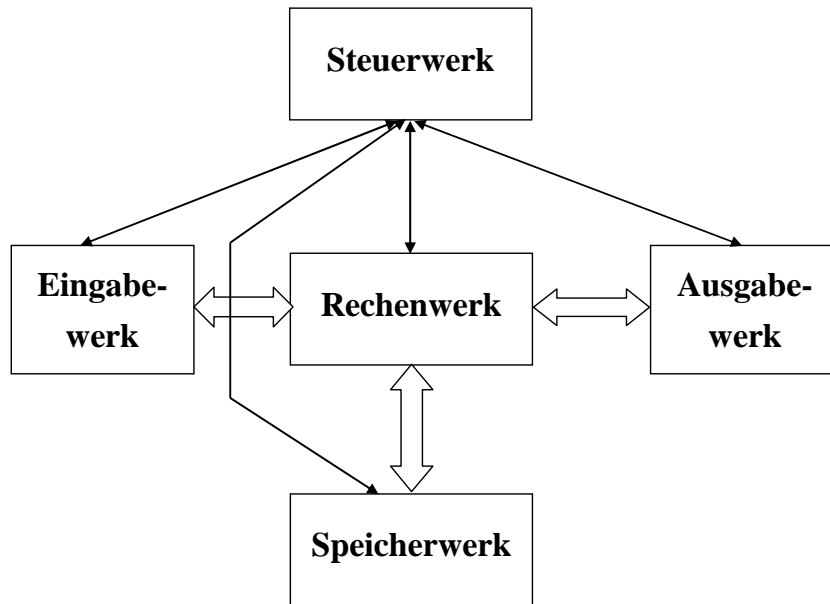
**Detailspezifikation von
Registern,
Speichern,
Zählern,
Rechenwerken,
Uhren, u. s. w.**

Gatterebene:

**Spezifikation der Operationen auf der Gatterebene
herunter bis zum Bit, Gatter werden als atomare
Bildungsblöcke betrachtet.**

von Neumann Architektur:

Das Modell des von Neumann Rechners kennt fünf Arbeitseinheiten: Steuerwerk, Rechenwerk, Speicherwerk, Eingabewerk und Ausgabewerk.



Die von Neumann Architektur wird erstmals beschrieben in:

Burks, A.W., H. H. Goldstine, J. von Neumann: Preliminary Discussion of the Logical Design of an Electronic Computing Instrument, Report to U.S. Army Ordnance Department (1946).

von Neumann Architektur:

Charakteristika:

- (i) Daten und Befehle liegen im gleichen Schreib-Lese-Speicher.
- (ii) Der Zugriff auf Speicherinhalte erfolgt über die Adresse der Speicherzelle, Speicherinhalte sind nicht typisiert.
- (iii) Die Programmausführung erfolgt sequentiell, Befehl für Befehl.

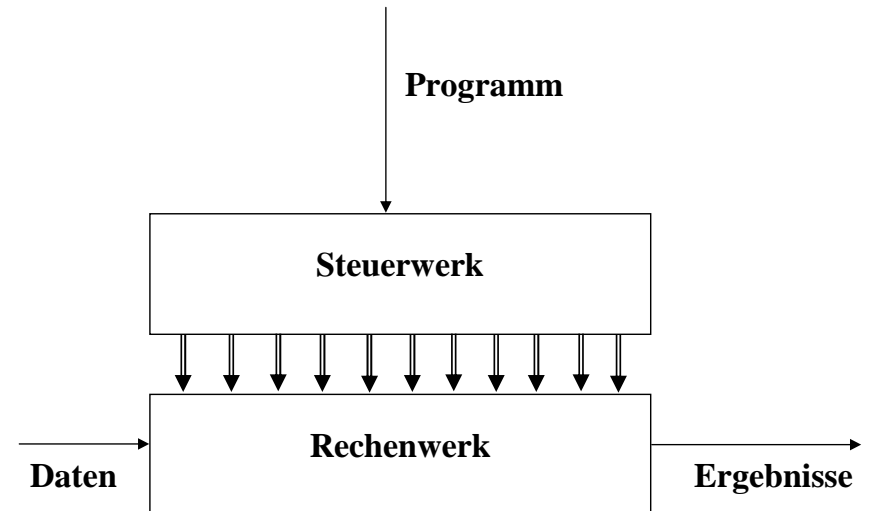
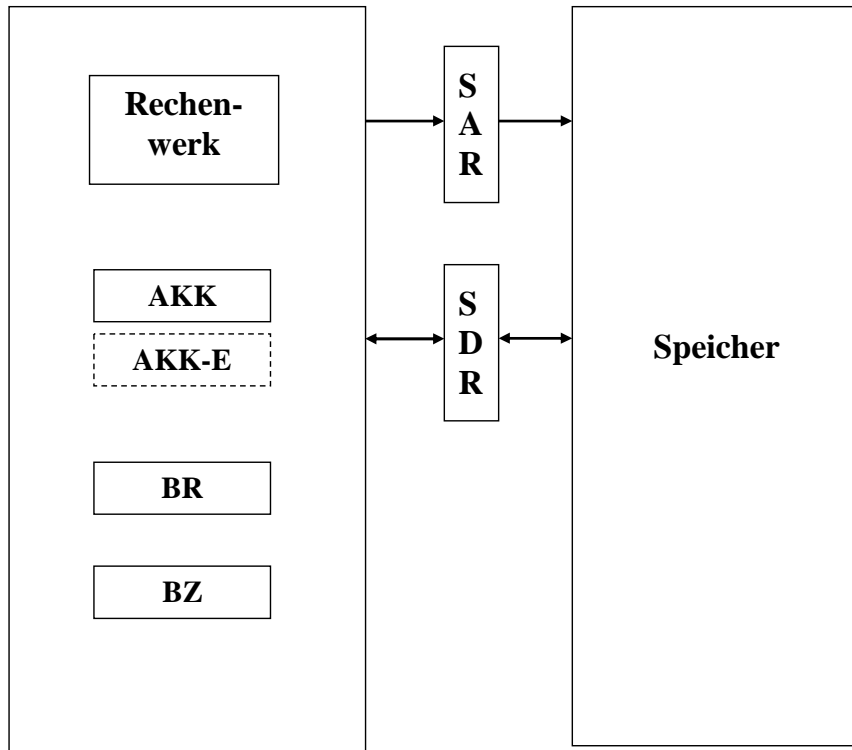


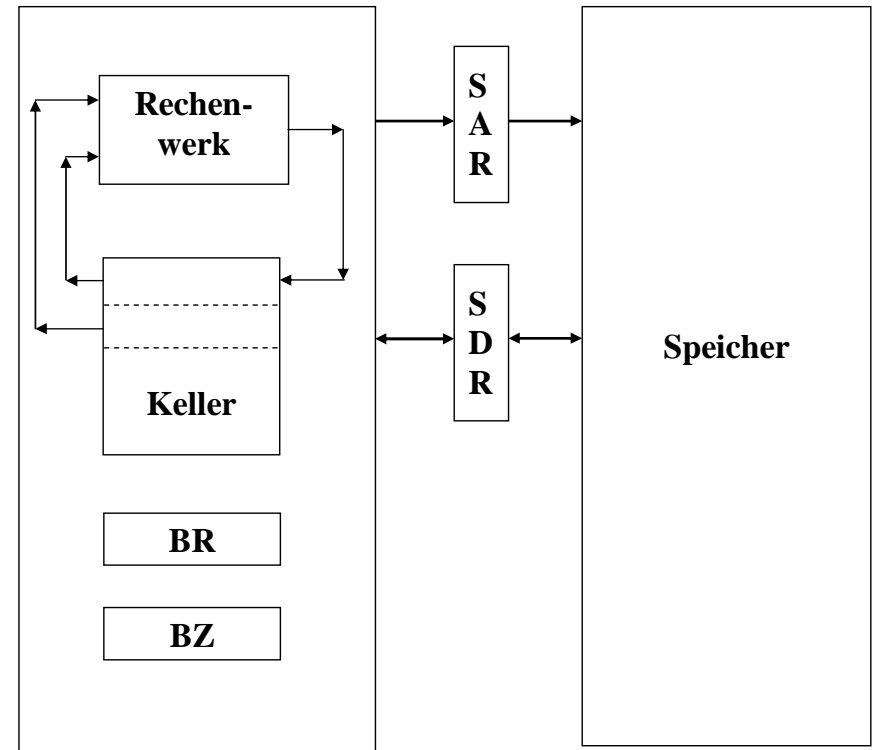
Schaubild einer Akkumulator-Maschine:



Legende:

- SAR** = Speicheradreßregister
- SDR** = Speicherdatenregister
- BZ** = Befehlszähler
- BR** = Befehlsregister
- AKK** = Akkumulator
- AKK-E** = Akkumulator-Erweiterung

Schaubild einer Keller-Maschine:



Legende:

- SAR** = Speicheradreßregister
- SDR** = Speicherdatenregister
- BZ** = Befehlszähler
- BR** = Befehlsregister

IAS-Rechner: (IAS = Institute for Advanced Study)

Bauzeit: 1946 – 1952

Hauptingenieur: Julian H. Bigelow

Speicher: erst 1024, später 4096 40 Bit-Wörter,
ein Wort nimmt zwei Befehle oder
eine Festpunktzahl auf.

Register: Ein Akkumulator AC,
ein Multiplizieren-Quotienten Register MQ.

Leistungskennzahlen:

Dauer einer Multiplikation: 600 μ sec,
Dauer eines Speicherzugriffs: 25 μ sec.

Zahldarstellung: Binäre Brüche zwischen -1 und $+1$,
die Größeninterpretation erfolgt über
virtuelle oder reale Skalierung.

Runden: Multiplikationen und Divisionen werden
gerundet. Divisionen werden immer so
gerundet, daß das niederwertigste Bit
des Quotienten auf 1 gesetzt wird, auch
im Fall $0 : x$.

Adressierung: Nur direkte Adressierung,
nur Ein-Adreß-Befehle.

Ein- und Ausgabe-Befehle:

Die Instruktion INPUT (X, N) transferiert N
Wörter von einem Eingabegerät über den Akku-
mulator in N aufeinanderfolgende Speicherzellen
ab Adresse X. Die Ausgabeinstruktion
OUTPUT (X, N) arbeitet analog zur Eingabe.

Befehlssatz der Princeton IAS Maschine (22 Befehle):

L	Load	; AC := M[x]
LMQ	Load MQ	; MQ := AC
AMQ	Load from MQ	; AC := MQ
ST	Store	; M[x] := AC
LN	Negate	; AC := - M[x]
LA	Load Absolute	; AC := M[x]
LNA	Negate Absolute	; AC := - M[x]
SAL	Store Address Left	; ersetze das linke ; Adreßfeld in M[x] durch Adreßbits des AC
SAR	Store Address Right	; ersetze das rechte ; Adreßfeld in M[x] durch Adreßbits des AC
A	Add	; AC := AC + M[x]
AA	Add Absolute	; AC := AC + M[x]
S	Subtract	; AC := AC - M[x]
SA	Subtract Absolute	; AC := AC - M[x]
MPY	Multiply	; AC.MQ := MQ * M[x]
DIV	Divide	; MQ.AC := AC : M[x] ; Quotient in MQ, Rest in AC
ASL	Shift Arithmetic Left	; AC := AC * 2
ASR	Shift Arithmetic Right	; AC := AC : 2
BL	Branch Left	; nächster Befehl ; steht in linker Hälfte von M[x]
BR	Branch Right	
BCL	Branch Conditional Left	; nächster Befehl ; steht in linker Hälfte von M[x], falls AC ≥ 0
BCR	Branch Conditional Right	; nächster Befehl ; steht in rechter Hälfte von M[x], falls AC ≥ 0
STP	Stop	

Bemerkung: Es existiert kein Befehl zum Auslesen des
Befehlszählers.

Nutzung des Befehlssatzes zur Addition von vier Zahlen:

$$\text{Erg} = Z1 + Z2 + Z3 + Z4$$

Speicherplan:

Adresse 0: Erg
Adresse 1: Z1
Adresse 2: Z2
Adresse 3: Z3
Adresse 4: Z4

Programm:

Annahme: Addenden stehen im Speicher

L	1
A	2
A	3
A	4
ST	0

Ergebnis im Speicherwort 0.

Frage: Wie läßt sich obiges Programm auf eine unbestimmte Zahl von Addenden erweitern?

Ein IAS-Programm zur Addition zweier Arrays:

$$c [1 .. 1000] := a [1 .. 1000] + b [1 .. 1000]$$

<u>Adresse</u>	<u>Datum/Befehl</u>	<u>Erläuterung</u>
0	999	; Schleifenzähler
1	1	
2	1000	
3L	L M [2000]	; Element von a
3R	A M [3000]	; Element von b
4L	ST M [4000]	; Element von c
4R	L M [0]	; Restzähler
5L	S M [1]	; berechnen
5R	BCR 6	; Test auf Schleifenende
6L	STP	; Programmende
6R	ST M [0]	; Speichern neuen Zählerwert
7L	A M [1]	;
7R	A M [2]	; neue Adresse für a-Element
8L	SAL 3	; Befehlsmodifikation !!
8R	A M [2]	; neue Adresse für b-Element
9L	SAR 3	; Befehlsmodifikation !!
9R	A M [2]	; neue Adresse für c-Element
10L	SAL 4	; Befehlsmodifikation !!
10R	BL 3	; zum Schleifenanfang
1001		; Speicherplatz für a
2001		; Speicherplatz für b
3001		; Speicherplatz für c

Präfixe für Maßangaben:

Präfixe für Einheiten im Dezimalsystem:

Faktor	Name	Symbol
10^{24}	yotta	Y
10^{21}	zetta	Z
10^{18}	exa	E
10^{15}	peta	P
10^{12}	tera	T
10^9	giga	G
10^6	mega	M
10^3	kilo	k
10^2	hecto	h
10^1	deka	da
10^{-1}	deci	d
10^{-2}	centi	c
10^{-3}	milli	m
10^{-6}	micro	μ
10^{-9}	nano	n
10^{-12}	pico	p
10^{-15}	femto	f
10^{-18}	atto	a
10^{-21}	zepto	z
10^{-24}	yocto	y

Präfixe im Dualsystem:

Standard: IEC 60027-2, Second edition, 2000-11, Letter symbols to be used in electrical technology – Part 2: Telecommunications and electronics.

Faktor	Kurzname	Symbol	Langnahme
2^{10}	kibi	Ki	kilobinary
2^{20}	mebi	Mi	megabinary
2^{30}	gibi	Gi	gigabinary
2^{40}	tebi	Ti	terabinary
2^{50}	pebi	Pi	petabinary
2^{60}	exbi	Ei	exabinary

Beispiele:

1 kibibit	=	1 024 bit
1 kilobit	=	1 000 bit
1 mebibit	=	1 048 576 bit
1 megabit	=	1 000 000 bit
1 gibibit	=	1 073 741 824 bit
1 gigabit	=	1 000 000 000 bit
1 tebibit	=	1 099 511 627 776 bit
1 terabit	=	1 000 000 000 000 bit