

Neuere Entwicklungen in der Informationstechnologie

Prof. Dr. Hans-Jürgen Buhl

1993 / 1995 / 1998 / 2000

Fachbereich Mathematik (7)
Institut für Angewandte Informatik
Bergische Universität – Gesamthochschule Wuppertal

Interner Bericht der Integrierten Arbeitsgruppe
Mathematische Probleme aus dem Ingenieurbereich
IAGMPI – 9305
Oktober 1993

4. Auflage, 2000

Praktische Informatik 01

Inhaltsverzeichnis

1	Rechnerarchitekturen	1
1.1	Das von-Neumann-Konzept	1
1.2	CISC/RISC	3
1.3	Field replaceable units (FRU's)	25
1.4	SIMD/MIMD und „data parallel programming“ (SPMD)	28
1.4.1	Cache-Speicher	29
1.4.2	Vektorrechner	30
1.4.3	SIMD- und MIMD-Parallelrechner	36
1.4.4	Mehrprozessorsysteme	41
1.4.5	Multiprozessor-Server	47
1.4.6	Massiv parallele Computersysteme	49
1.4.7	Anwendung: Die massiv parallele Simulation	53
1.4.8	Klassifikation von parallelen Rechnern	54
1.5	Speichermodule (SIMM's, . . .)	55
1.6	Assoziative Speicher	56
1.7	Neuronale Netze	60
1.8	ANHANG: Benchmarks	66
2	Peripherie	71
2.1	Scanner, Strichcodeleser, OCR	71
2.1.1	Historische Entwicklung der Informations-Input-Codes	71
2.1.2	Strichcodeleser	73
2.1.3	Scanner	74
2.2	Spracheingabe	92
2.3	Tintenstrahldrucker, Laserdrucker, Farbdrucker	98

Abbildungsverzeichnis

1.1	Universalrechner	1
1.2	Dekomposition von Prozessen in Sequenzen von Primitiva	3
1.3	CISC-CPU-Aufbau	4
1.4	Programmausführung	5
1.5	Microprogrammierte CISC-CPU	6
1.6	Nanoprogrammierte CISC-CPU	7
1.7	Registerstack am Beispiel SPARC	12
1.8	FRU's der SPARC-ELC	25
1.9	Entfernen des NVRAM-Moduls	26
1.10	Einbau von SIMM's	27
1.11	Der Flaschenhals zwischen CPU und Memory	28
1.12	Der Cache als Daten-Vorratsbehälter	29
1.13	Verarbeitungseinheit: Vektor von Worten	30
1.14	Vektorregister und Vektoralu	31
1.15	Rechnerstruktur eines Cray-1-Rechners	33
1.16	Die Register und Funktionseinheiten einer Cray-1	34
1.17	CRAY-2 (Juni 1985)	36
1.18	NEC-SX2 SIMD, mehrere Pipelines	38
1.19	Systemarchitektur und Datendurchsatz	40
1.20	„shared memory“ und „message passing“	42
1.21	Zeitdiagramm „circuit switched“ Datentransfer	44
1.22	Schaltschema der CM2	50
2.1	Verteilung UNICODE	82
2.2	Bidirectional Ordering	83
2.3	General Scripts	84

Tabellenverzeichnis

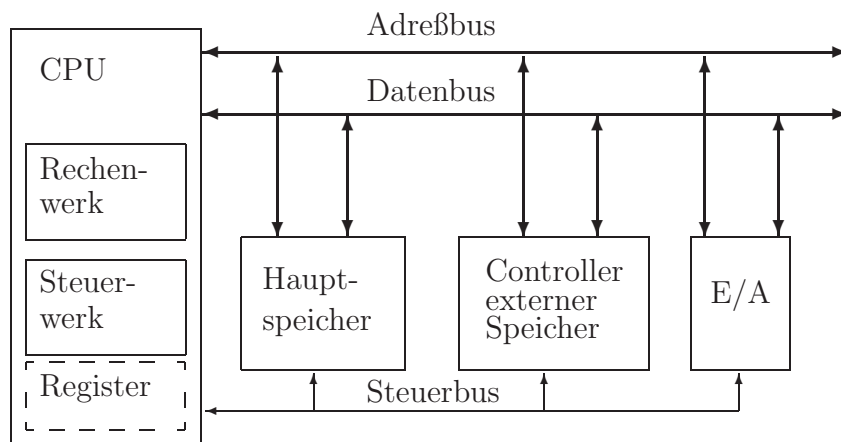
1.1	Analyse typischer Computeranwendungen auf ihre Instruktionstypen hin	8
1.2	Von CISC zu RISC	9
1.3	Unterscheidungsmerkmale CISC/RISC	10
1.4	RISC-CPU's im Vergleich zu Intel 80x86-CPU's	13
1.5	64 Bit RISC-CPU's	14
1.6	PC-Bussysteme und Durchsatzraten	14
1.7	RISC PC-CPU's: PowerPC (Apple und IBM)	14
1.8	CISC PC-CPU's: i80x86	15
1.9	CISC PC-CPU's: i80x86 (Forts.)	16
1.10	Marktallianzen im RISC-Bereich	19
1.11	Klassifikation von Mehrprozessorsystemen	41
1.12	Speicher bei Mehrprozessorsystemen	42
2.1	UNICODE Version 1.0, Character Blocks 0000-00FF	85
2.2	Weitere Zeichenbereiche	86
2.3	UNICODE to Adobe Standard Mappings	87
2.4	The UNICODE to SGML (ISO DIS 6862.2) Mappings	88
2.5	UNICODE to Macintosh Mappings	89

Kapitel 1

Rechnerarchitekturen

1.1 Das von-Neumann-Konzept

Abbildung 1.1: Universalrechner

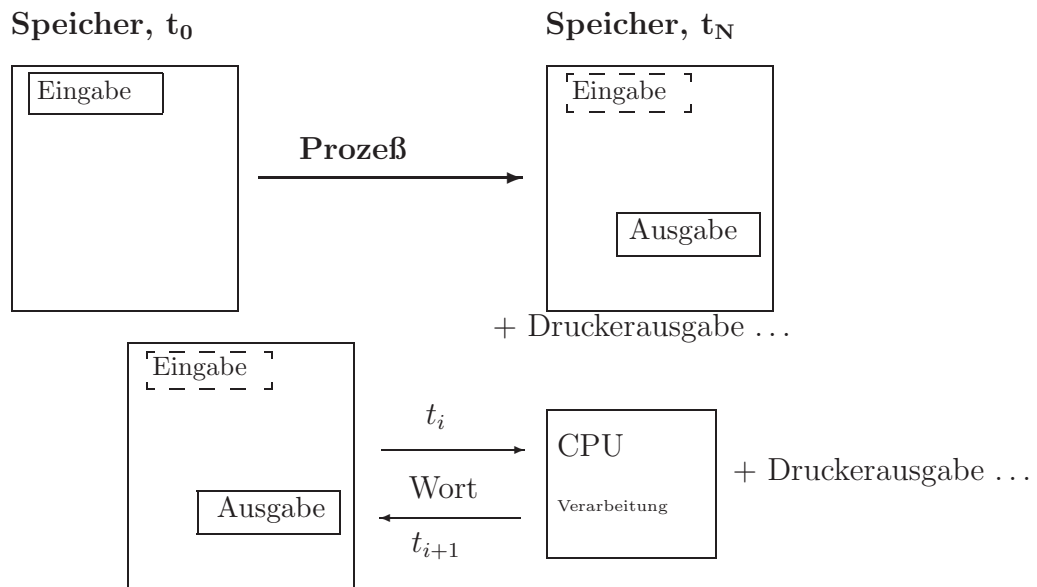


Prinzipien:

1. **Computerbestandteile:** CPU (= Rechenwerk und Steuerwerk), Speicher, Ein- und Ausgabewerk.
2. **Universalität:** Spezialisierung nur durch von außen eingegebenes Programm.
3. **Programme als Daten:** Eingabedaten, Zwischen- und Endergebnisse sowie Programme werden im selben Speicher abgelegt.

4. **linearer Speicher:** Der Speicher ist in gleichgroße Zellen, die fortlaufend numeriert sind, eingeteilt.
5. **sequentielle Ausführung:** Aufeinanderfolgende Befehle eines Programms werden in aufeinanderfolgenden Speicherzellen abgelegt. Beim Programmablauf wird im allgemeinen der Befehlszähler fortlaufend inkrementiert.
6. **Sprünge:** Sprungbefehle ermöglichen ein Durchbrechen der linearen Ausführungsreihenfolge.
7. **zusätzliche Primitiva:** Datentransferbefehle, arithmetische Operationen, logische und Vergleichsoperationen, *heute auch — jedoch nicht in klassischen CPUs* : Multimedia-Befehle (insbesondere Echtzeit-Komprimierung/Dekomprimierung; vgl. MPEG = Standard der Moving Picture Experts Group und JPEG = Standard der Joint Photographic Experts Group); indizierte, relative, ... Adressierung.
8. **binäre Codierung:** Daten (Befehle, Adressen, Operanden) werden binär codiert. Geeignete Schaltwerke (Decodierer) sorgen für die richtige Entschlüsselung.

Abbildung 1.2: Dekomposition von Prozessen in Sequenzen von Primitiva



1.2 CISC/RISC

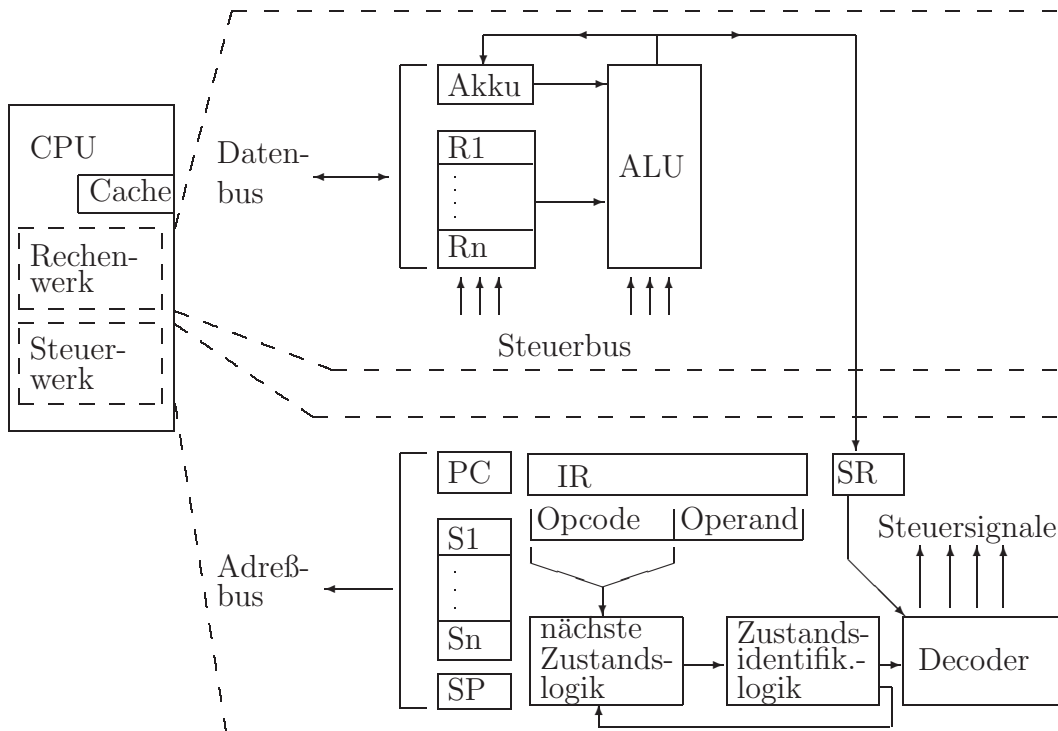
Traditionelle CISC's (= complex instruction set computers) sind:

- IBM360
- DEC VAX
- Intel 80386 (80x86)
- Motorola 68030 (680x0)
- ...

In Zeiten, als der Zeitraum zur Übertragung eines Datenwortes vom Speicher in ein Register oder umgekehrt Größenordnungen länger dauerte als ein einfacher logischer oder arithmetischer Primitiv-Befehl, mußte zunächst die Programmabarbeitung durch *wait-states* künstlich verlangsamt werden.

Man versuchte dann aus Gründen der Ökonomie, eine CPU immer komplexere Primitiv-Befehle (deren Ausführungszeit lediglich anfangs immer noch

Abbildung 1.3: CISC-CPU-Aufbau

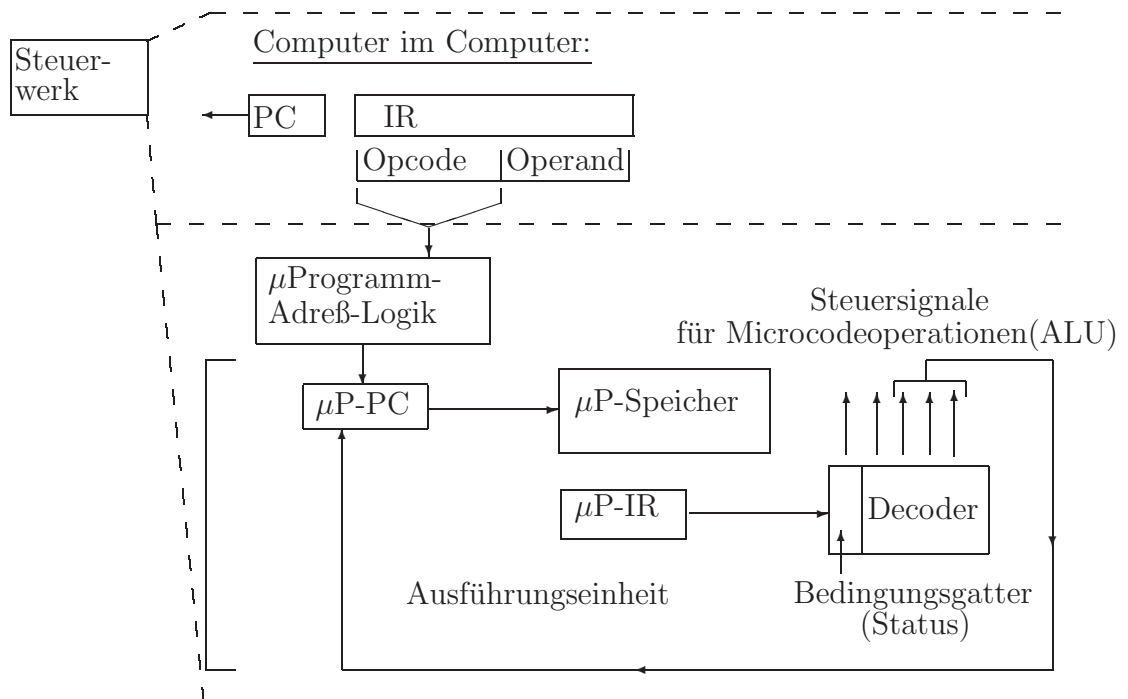


höchstens eine „Datentransferzeit Register in Speicher“ lang war) ausführen zu lassen. In der damaligen Zeit waren dafür jedoch nicht genügend Schal-
toperationen auf einem Chip realisierbar, weshalb man das Rechenwerk der
CPU wiederum als Computer aufbaute:

Abbildung 1.4: Programmausführung

```
PC := StartingAddress;
RunBit := 1;
while RunBit = 1 do begin
  IR := memory(PC);
  PC := PC + Befehlslänge(IR);
             häufig=1, fest
  InstrType := DetermineInstrType(IR);
  DataAddress := LocateData(IR, InstrType, PC);
  if DataAddress <> nil then
    data := memory(DataAddress)
  else
    data := non_valid;
  Execute(IR, InstrType, data, memory, registers, PC, RunBit);
end;
```

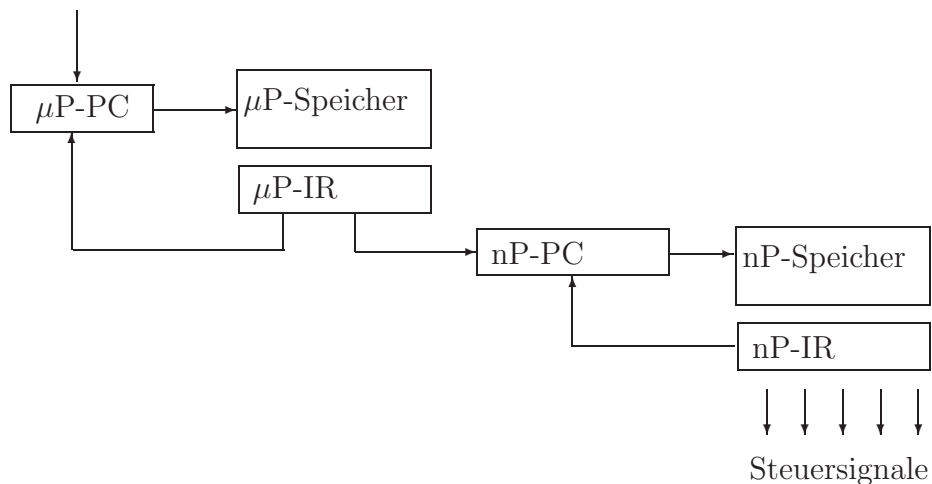
Abbildung 1.5: Microprogrammierte CISC-CPU



Software („SW“) reduziert den Schaltungsaufwand (= Anzahl der Gatter, Anzahl der Leitungen, ...) und „interpretiert“ den Maschinencode der CPU (Computer im Computer).

Eventuell werden sogar „noch“ Nanoprogramme benutzt (68000er, ...):

Abbildung 1.6: Nanoprogrammierte CISC-CPU



Das CISC-Designprinzip:

Reduziere die „semantic gap“ zwischen Maschinsprache und Hochsprache durch

- viele komplexe Maschinenbefehle ($\gtrsim 200$), etwa „case“, „while“, ...
- viele Adressierungsmodi, etwa für „Felder“, „Verbunde“, ...
- Unterprogramm-Management im Maschinencode

unter besonderer Beachtung der mindestens um den Faktor 10 langsameren Transferbefehle zum/vom Speicher.

„Überalterung“ von CISC:

Ungefähr 1970 wird

- die Komplexität des Microprogramms immer schlechter zu managen
- der Speicherzugriff schneller
- lohnt sich ein Redesign aufgrund von Messungen „typischer“ Anwendungen:

Tabelle 1.1: Analyse typischer Computeranwendungen auf ihre Instruktionstypen hin

%-Verteilung	SAL Operating System Tanenbaum (1978)	XPL Systempgm. Wortman (1972)	FORTRAN num. Pgm. Knuth (1971)	C Systempgm. Patterson (1982)	Pascal Systempgm. Patterson (1982)
:=	47 ¹⁾	55	51	38	45
if	17	17	10	43	29
call	25	17	5	12	15
loop	6	5	9	3	5
goto	0	1	9	3	0
sonstiges	5	5	16	1	6

Dabei sind in ¹⁾ 80% der Wertzuweisungen von dem einfachen Typ „Variable := Wert“.

Konsequenz: Der Overhead durch das komplexe Microprogramm und die vielen Adressierungsmodi wird nur sehr selten auch ausgenutzt.

RISC (= reduced instruction set computer)-Designprinzip:

- Analysiere Anwendungen, um die Schlüsseloperationen zu finden.
- Reduziere unter allen Umständen die „Datenpfadzykluszeit“ (Register → ALU → Register) für diese Operationen, also: *keine* Microcode Interpretation (Maschinencode = Microcode)
- Jedes „neue“ Feature ist nur zuzulassen, wenn es häufig benutzt wird und die Mehrzahl der existierenden nicht verlangsamt.

(Wiederhole diesen Design-Zyklus für die anderen CPU-Ressourcen: Cache, Speichermanagement, Gleitkomma-Koprozessoren, . . .)

Tabelle 1.2: Von CISC zu RISC

	CISC			RISC		
	IBM 370/168	VAX 11/780	Xerox Dorado	IBM 801	Berkeley RISC I	Stanford MIPS
Jahr	1973	1978	1978	1980	1981	1983
# Operationen	208	303	270	120	39	55
Microcode-Größe [byte]	54K	61K	17K	—	—	—
Befehlslänge [byte]	2–6	2–57	1–3	4	4	4
Ausführungsmodell	reg-reg reg-men men-men	reg-reg reg-men men-men	stack	reg-reg	reg-reg	reg-reg

CISC- und RISC-Architekturen lassen sich durch einige charakteristische Merkmale voneinander unterscheiden:

Tabelle 1.3: Unterscheidungsmerkmale CISC/RISC

Bei RISC üblich	Bei CISC üblich
1. einfache Befehle, 1 Befehl/Zyklus	komplexe Befehle, 1 Befehl/viele Zyklen
2. LOAD/STORE-Architektur (nur LOAD/STORE greift auf Speicher zu)	mem-reg, mem-mem Befehle
3. pipelined	kaum pipelined
4. Hardware, kein Microcode	Microcode
5. feste Befehlslänge	variable Befehlslänge (1 . . . 17 byte beim 80386)
6. Wenige Befehle, wenige Adressierungsmodi	viele Befehle und Adressierungsmodi
7. Komplexität im Compiler	Komplexität im Microprogramm
8. Registerstack	feste Registermenge

Bemerkungen:

zu 1. Deswegen evtl. sogar keine Multiplikation/Division, die dann in Runtime-Bibliotheken angeboten werden muß.

zu 2. Vorhandene RISC-Varianten: (load|store) (signed|unsigned) (byte|halfword|word)

zu 3. – delayed load/store: Codereorganisation bzw. NOP's

– Pipeline:

Zyklus	1	2	3	4	5	6	7 ...
instr1	fetch	decode	exec	write			
instr2		fetch	decode	exec	write		
instr3			fetch	decode	exec	write	
⋮							

zu 4. in RISC viel Chip-Platz frei: mehr Register realisierbar

zu 5. Die variable Länge benötigte Microcode (oder aufwendige Decoder).

zu 6. – Befehlsdecoder-Länge wächst exponentiell mit der Anzahl der Befehle.

– Geschwindigkeit/Komplexität: wenige Adressierungsmodi

zu 7. In RISC-Systemen sind gute Compiler unumgänglich.

zu 8. Registerstack:

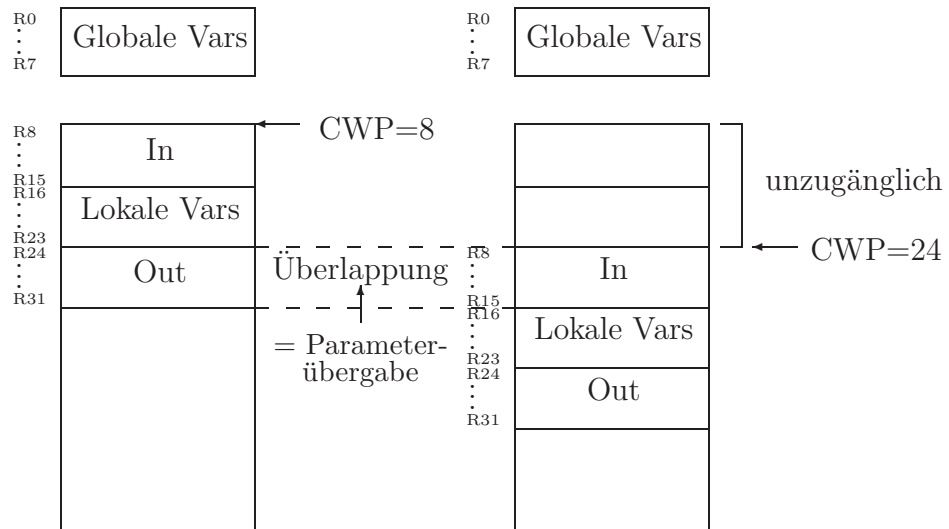
– ~ 500 Register, von denen 32 gleichzeitig verfügbar sind

– überlappende Register-Fenster (s.u.)

– Register-Mehrfachbenutzung („Lebenszeit“ von Variablen)

Der Registerstack ist auch bei CISC möglich, wenn genügend Chip-Platz verfügbar ist.

Abbildung 1.7: Registerstack am Beispiel SPARC



Aktuelle Übersicht: (4/97)

Tabelle 1.4: RISC-CPU's im Vergleich zu Intel 80x86-CPU's

	MIPS 4400	SPARC SS20	DEC- Alpha	Intel Pentium P5	HP PA7100	IBM Power 6264	MIPS R4000SC	SPARC SS10	Intel 486DX2
Taktrate (MHz)	200	75	nom.200	66	100	62,5	100	40	66
# Pipelinestufen		4	7		5	6	8	4	5
max # Instr./Zyklus		3	2		2	4	1	3	1
Cache Instr.	16 KB	20 KB	8 KB	16 KB	—	32 KB	8 KB	20 KB	8 KB
Cache Data	16 KB	16 KB	8 KB	gemeinsam	—	—	8 KB	16 KB	gemeins.
max Durchsatz (MB/s)			3200		k.A.	1000	1200	960	1067
sek. Cache Instr.	1 MB	1 MB	32 MB	256 KB	1 MB	—	4 MB	2 MB	512 KB
sek. Cache Data	insges.	insges.	insges.	gemeinsam	2 MB	64 KB	insges.	gemeins.	gemeins.
max Durchsatz (MB/s)			1067	k.A.	1600	500	800	320	133
Bus-Taktrate (MHz)			67		67	62,5	50	40	33
Bus-Durchsatz (MB/s)	1200		1067		267	1000	400	320	133
SPECint92	140	125,8	69,6	64,5	80,0	59,2	61,7	53,2	32,2
SPECfp92	131	121,2	182,1	56,9	150,6	124,8	63,4	63,4	16,0
SPECint95		3,11			3,13			1,0	
SPECfp95		3,10			4,0			1,0	

Zum Vergleich:

Der MPC620 ist ein 64 Bit-Prozessor, der MPC604e hat zwar einen 64 Bit-Datenbus, aber nur einen 32 Bit-Adreßbus.

Tabelle 1.5: 64 Bit RISC-CPU's

	UltraSPARC II	Ultra-SPARC I	MIPS R10000	DEC-alpha 21164	HP PA8200	IBM PowerPC 604e
Taktrate (MHz)	336	200	250	533	240	332
Cache Instr.	16 KB	16 KB	32 KB	8 KB	2 MB	32 KB
Cache Data	16 KB	16 KB	32 KB	8 KB	2 MB	32 KB
sek. Cache	4 MB	1 MB	4 MB	96 KB + 4 MB	—	256 KB
SPECint95	14,9	7,81	14,7	16,9	17,4	14,4
SPECfp95	37,6	10,4	62,5	56,7	28,5	12,6

Tabelle 1.6: PC-Bussysteme und Durchsatzraten

	ISA	MCA 1.0	EISA	EISA EMB-133	VL-Bus	VL-Bus 64-bit	PCI 32	PCI 64
Datenbus- breite (bit)	16	32	32	64	32	64	32	64
Taktrate (MHz)	8	4	8	8	33	50	33	33
Bus-Durchsatz (MB/s)	8	16	33	125	75	160	117	234

Tabelle 1.7: RISC PC-CPU's: PowerPC (Apple und IBM)

	MPC620	PowerPC „G3“	MPC604e	MPC604	MPC603e	MPC603	MPC601
Taktrate (MHz)	~133	250	332	166	200 . . . 240	66/80	50/60/66/80
Cache Instr.	32 KB	32 KB	32 KB		16 KB	8 KB	32 KB
Cache Data	32 KB	32 KB	32 KB		16 KB	8 KB	gemeins.
sek. Cache	1 . . . 128 MB	256 . . . 1024 KB	256 KB		1 MB	1 MB	1 MB
SPECint92	~225					~60/75	~62 . . . 85
SPECfp92	~300					~70/85	~72 . . . 105
SPECint95	~5,6	~10	14,4	5,2	7,1		
SPECfp95	~5,6		12,6	4,3	4,2		

Tabelle 1.8: CISC PC-CPU's: i80x86

	i486DX2	i486DX4	Pentium	Pentium	Pentium	Pentium MMX	PentiumPro
Taktrate (MHz)	66	100	100	133	200	200	200
Cache Instr.	8 KB	16 KB	8 KB	8 KB	8 KB	8 KB	8 KB
Cache Data	gemeins.	gemeins.	8 KB	8 KB	8 KB	8 KB	8 KB
sek. Cache	256 KB	256 KB	512 KB	512 KB	512 KB	512 KB	512 KB
SPECint92	32,2	51,4	100,0				
SPECfp92	16,0	26,6	80,6				
SPECint95			3,05	4,1	5,5	6,41	8,58
SPECfp95			2,07	2,5	3,92	4,66	6,48

Tabelle 1.9: CISC PC-CPU: i80x86 (Forts.)

	Pentium II	„Deschutes“
	MMX, AGP, SDRAM	MMX, AGP
Taktrate (MHz)	333	300 . . .
Cache Instr.	16 KB	
Cache Data	16 KB	
sek. Cache	512 KB	
SPECint95	12,8	~12
SPECfp95	9,25	~10

MMX („matrix manipulation extensions“) stellt eine Erweiterung des Befehlssatzes für Bild/Audio/Video zu Lasten der FloatingPoint-Befehle dar. Der PentiumPro (P6) ist intern ein RISC-Prozessor mit eingebautem „CISC-RISC translator“ (Kompatibilität!). Für den Pentium II wird ein schnelleres Motherboard mit SDRAM („synchr. dyn. RAM“) benötigt; die Graphikkarte wird aus Geschwindigkeitsgründen nicht mehr über den PCI-Bus sondern über den dedizierten AGP („accelerated graphics ports“) angeschlossen.

Aktuelle Daten entnehme man dem WWW:

<http://www.specbench.org>, <http://www.ibm.com>,
<http://www.intel.com>, . . .

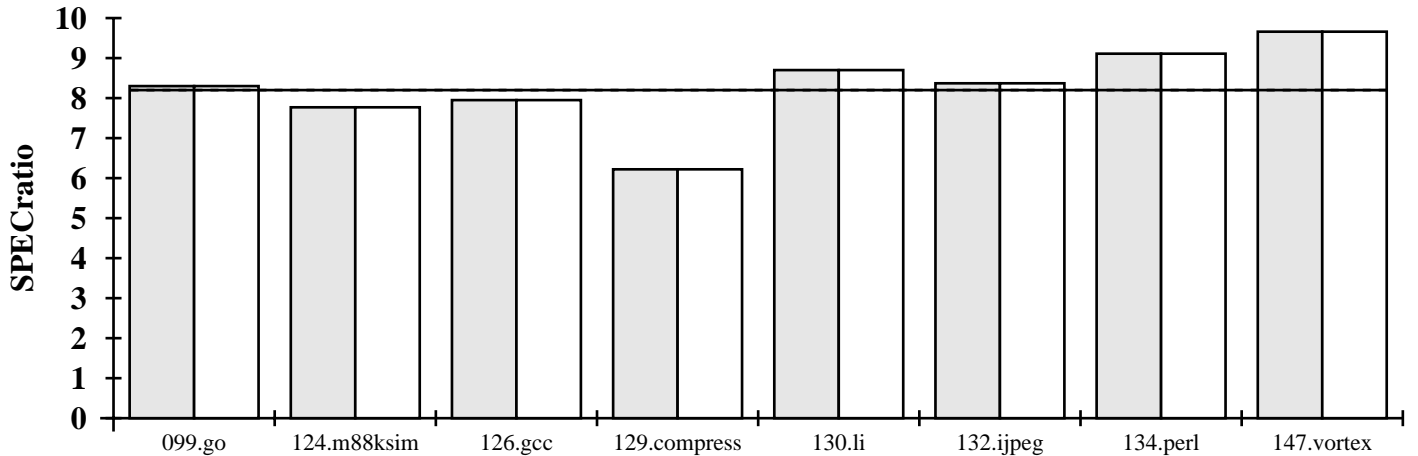
SPEC CINT95 Results

©Copyright 1995, Standard Performance Evaluation Corporation

Dell Computer Corporation
Dell Dimension XPS Pro200n

SPECint95 = 8.20
 SPECint_base95 = 8.20

SPEC license # 14 | Tested By: Intel | Test Date: May-96 | Hardware Avail: Jun-96 | Software Avail: Mar-96



Hardware/Software Configuration for: Dell Dimension XPS Pro200n		Benchmark # and Name	Reference Time	Base Run Time	Base SPEC Ratio	Run Time	SPEC Ratio	
Hardware		099.go	4600	554	8.30	554	8.30	
Model Name:	XPS	124.m88ksim	1900	244	7.77	244	7.77	
CPU:	200MHz Pentium Pro processor	126.gcc	1700	214	7.95	214	7.95	
FPU:	Integrated	129.compress	1800	289	6.22	289	6.22	
Number of CPU(s):	1	130.li	1900	218	8.70	218	8.70	
Primary Cache:	8KBI+8KBD	132.jpeg	2400	287	8.37	287	8.37	
Secondary Cache:	256KB(I+D)	134.perl	1900	209	9.11	209	9.11	
Other Cache:	None	147.vortex	2700	280	9.66	280	9.66	
Memory:	64MB(EDO)	SPECint_base95 (G. Mean)					8.20	
Disk Subsystem:	1MB IDE Quantum Fireball	SPECint95 (G. Mean)					8.20	
Other Hardware:	Integrated EIDE disk controller							
Software								
Operating System:	UnixWare 2.0, UnixWare SDK							
Compiler:	Intel C Reference Compiler V2.3							
File System:	UFS							
System State:	Single User							

Notes/Tuning Information

Base and peak flags are the same and use Feedback Directed Optimization
 Pass1: -tp p6 -ipo -prof_gen -ircdb_dir /proj/tmp/IRCDB
 Pass2: -tp p6 -ipo -prof_use -ircdb_dir /proj/tmp/IRCDB
 -ircdb_dir is a location flag and not an optimization flag
 Portability: 124: -DSYSV -DLEHOST 130, 134, 147: -lm 132: -DSYSV 126: -lm -lc -L/usr/ucblib -lucb -lmalloc

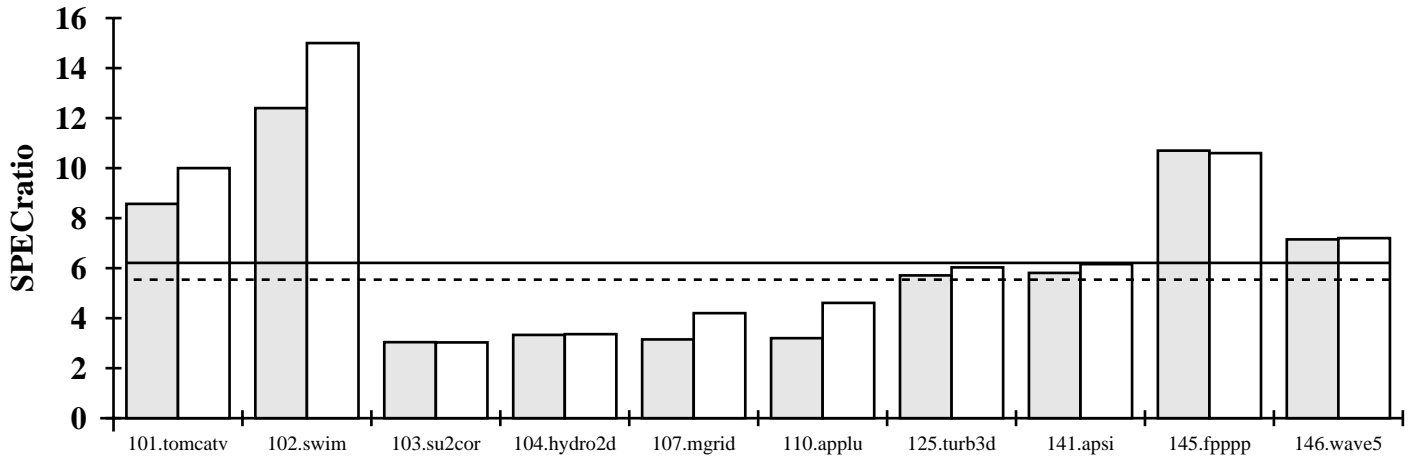
SPEC CFP95 Results

©Copyright 1995, Standard Performance Evaluation Corporation

Dell Computer Corporation
Dell Dimension XPS Pro200n

SPECfp95 = 6.21
 SPECfp_base95 = 5.54

SPEC license # 14 | Tested By: Intel | Test Date: May-96 | Hardware Avail: Jun-96 | Software Avail: Mar-96



Hardware/Software Configuration for: Dell Dimension XPS Pro200n		Benchmark # and Name	Reference Time	Base Run Time	Base SPEC Ratio	Run Time	SPEC Ratio
Hardware		101.tomcatv	3700	432	8.57	369	10.0
Model Name:	XPS	102.swim	8600	695	12.4	572	15.0
CPU:	200MHz Pentium Pro processor	103.su2cor	1400	461	3.04	462	3.03
FPU:	Integrated	104.hydro2d	2400	720	3.33	715	3.36
Number of CPU(s):	1	107.mgrid	2500	793	3.15	595	4.20
Primary Cache:	8KBI+8KBD	110.applu	2200	687	3.20	478	4.61
Secondary Cache:	256KB(I+D)	125.turb3d	4100	718	5.71	679	6.03
Other Cache:	None	141.apsi	2100	361	5.81	341	6.16
Memory:	64MB(EDO)	145.fpppp	9600	901	10.7	902	10.6
Disk Subsystem:	1MB IDE Quantum Fireball	146.wave5	3000	420	7.15	417	7.20
Other Hardware:	Integrated EIDE disk controller	SPECfp_base95 (G. Mean)		5.54			
Software				SPECfp95 (G. Mean)		6.21	
Operating System:	UnixWare 2.0, UnixWare SDK						
Compiler:	Intel C/FORTRAN Reference Compiler V2.3						
File System:	UFS						
System State:	Single User						

Notes/Tuning Information

```

Flags: Base: -tp p6 -ipo -pc 64 -mem
Peak:
101: -tp p6 -ipo -mem -pad -distype -fcon -pc 64
102: -tp p6 -ipo -pad -fcon -pc 64
103: Pass1: -tp p6 -prof_gen -ircdb_dir /proj/tmp/IRCDB
Pass2: -tp p6 -ip -mem -mill -pad -fcon -pc 64 -prof_use -ircdb_dir /proj/tmp/IRCDB
104: Pass1: -tp p6 -prof_gen -ircdb_dir /proj/tmp/IRCDB
Pass2: -tp p6 -ip -mem -mill -pc 64 -prof_use -ircdb_dir /proj/tmp/IRCDB
107: -tp p6 -pad -fcon -pc 64
110: -tp p6 -ipo -mem -mP2OPT_opt_bblock_stat_limit=10000 -pc 64
125: -tp p6 -ip -mem -mill -pad -fcon -pc 64
141: Pass1: -tp p6 -w -prof_gen -ircdb_dir /proj/tmp/IRCDB
Pass2: -tp p6 -pc 64 -prof_use -ircdb_dir /proj/tmp/IRCDB
145: -tp p6 -ipo -pc 64
146: Pass1: -tp p6 -prof_gen -ircdb_dir /proj/tmp/IRCDB
Pass2: -tp p6 -ipo -mem -pad -distype -pc 64 -prof_use -ircdb_dir /proj/tmp/IRCDB
    
```

18

For More
Information
Contact:

SPEC
10754 Ambassador Drive, Suite 201
Manassas, VA 22110

(703) 331-0180
info@specbench.org
http://www.specbench.org

--

Prepared By: --

Tabelle 1.10: Marktallianzen im RISC-Bereich

Das Sparc-Lager (unter anderem): SPARC

- Sun Microsystems
- Amdahl
- Cray
- Fujitsu
- ICL
- Solbourne
- Toshiba
- Xerox
- Matsushita
- Tadpole Technology
- Goldstar
- Hyundai
- Tatung
- Force Computers
- LSI Logic
- Tulip Computers
- Meiko

Das HP-PRO-Lager: HP PA (Power Architecture)

- HP
- Hitachi
- Mitsubishi
- Hughes
- Oki
- Yokogawa
- Prime
- Convex
- Sequoia

Das Motorola-Lager (88-Open unter anderem): 88000 (siehe auch Power-PC)

- Motorola
- Data General
- Omron
- Encore Computer
- Dolphin Server Technology
- Sanyo/Icon International Inc.
- Ceta (Thomson CSF)
- Harris Computer Systems Division
- McDonnell Douglas Information Systems
- Micro Focus
- Philips Information Systems
- SAS Institute
- Tadpole Technology
- Unify
- Wordperfect
- Oracle
- Unix International

Das Power-Open-Lager: POWER (siehe auch Power-PC)

- IBM
- Apple
- Bull HN Information Systems Inc.
- Harris Corp.
- Motorola
- Tadpole Technology
- Thomson CSF

Das Digital-Lager: Alpha

- DEC
- Kubota
- Olivetti
- Raytheon
- Encore Computer
- Advanced Computer Research International
- Carrera Computers
- Cray Research

Das Mips-Lager (unter anderem): MIPS

- Silicon Graphics (Mips)
- DEC
- Pyramid
- SNI
- Siemens
- Olivetti
- Tandem
- NEC
- Sony
- AT&T
- Sumitomo
- Control Data
- Microsoft
- LSI Logic
- Integrated Device Technology
- Toshiba
- Acer
- Carrera Computers

Intel: (i860)

- PentiumPro (nur im Kern ?!)

Das Power-PC-Lager: POWER-PC

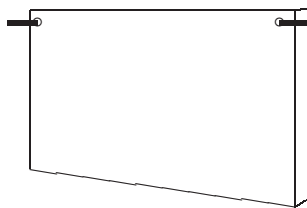
- IBM
- Apple
- ASI
- Bull HN Information Systems Inc.
- Canon
- Ford (für den Einsatz in Kraftfahrzeugen)
- Harris Corp.
- Motorola
- Peacock
- Tadpole Technology
- Taiwan New PC Consortium (24 Hersteller)
- Thomson CSF
- Vobis

1.3 Field replaceable units (FRU's)

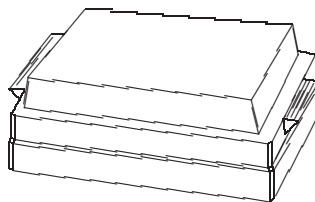
Am Beispiel des kleinsten RISC-Rechners von Sun Microsystems, der ELC:

- Systemplatine, darauf:
 - NVRAM, d.h. non volatile (= nicht flüchtiger) Speicher
 - Speichermodule SIMM's (= single inline memory module)
 - gesockelte Chip's (FPU, sekundärer Cache, ...)
- Bildschirm/Monitor

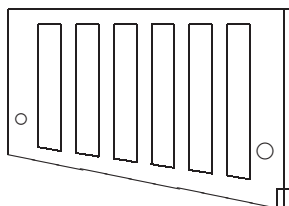
Abbildung 1.8: FRU's der SPARC-ELC



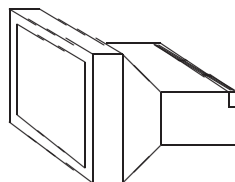
Systemplatine



NVRAM-Modul



Speichermodul



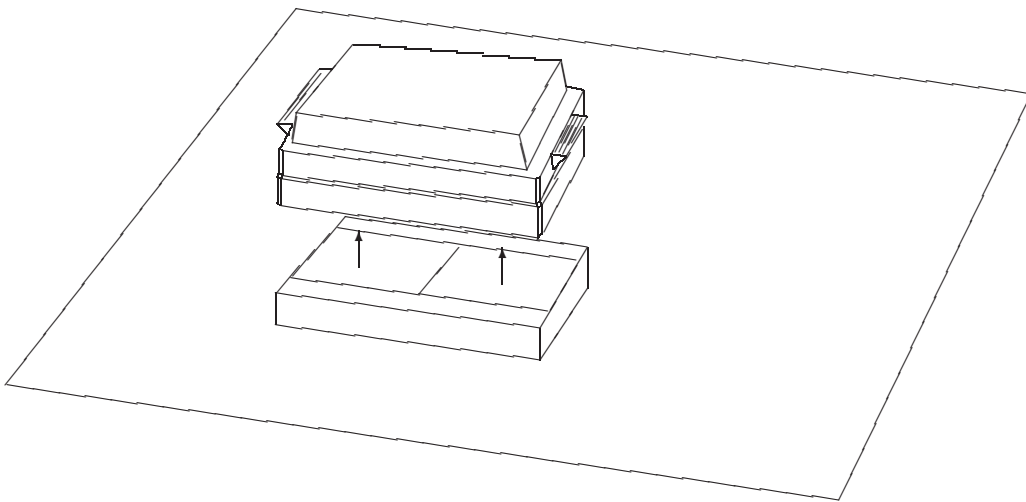
Bildschirm

Auszüge aus Reparaturanleitungen für Feld-Techniker:

- **Entfernen des NVRAM-Moduls:**

Fassen Sie das Gehäuse an den Griffen und ziehen Sie das NVRAM kräftig nach oben ab (siehe Abbildung):

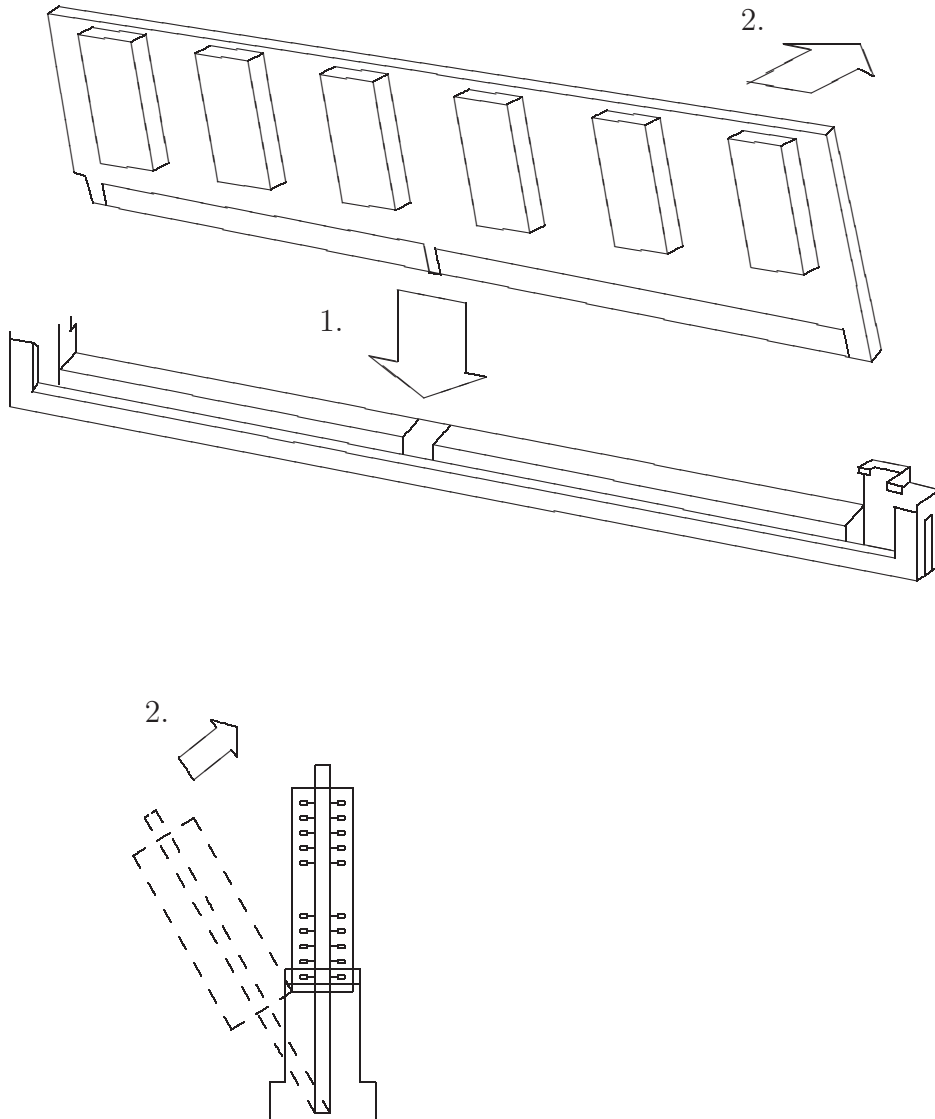
Abbildung 1.9: Entfernen des NVRAM-Moduls



Vorsicht: Die Systemplatine und das NVRAM-Modul bestehen aus hochempfindlichen elektronischen Bauteilen, die gegen statische Aufladung empfindlich sind. Vergewissern Sie sich, daß Sie beim Austausch der Komponenten alle Sicherheitsbestimmungen einhalten.

- Einbau von Speichermodulen:

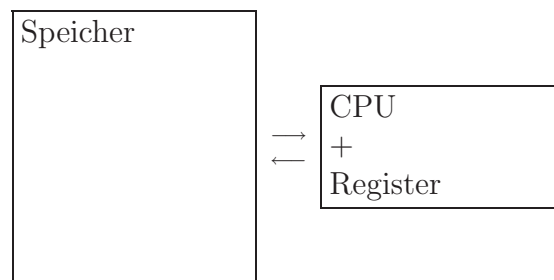
Abbildung 1.10: Einbau von SIMM's



Hinweis: Es spielt keine Rolle, welches Modul Sie in welchen Steckplatz installieren. Sie können jedoch nicht ein Modul zwischen zwei anderen bereits installierten Modulen montieren, da die Sockel dafür zu eng nebeneinander liegen.

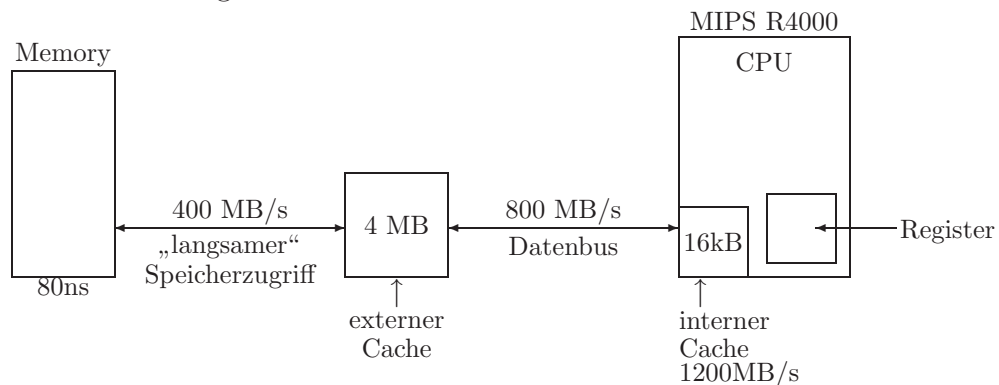
1.4 SIMD/MIMD und „data parallel programming“ (SPMD)

Abbildung 1.11: Der Flaschenhals zwischen CPU und Memory



1.4.1 Cache-Speicher

Abbildung 1.12: Der Cache als Daten-Vorratsbehälter



Bemerkungen:

- $80\text{ns} \longleftrightarrow \frac{1}{80 \cdot 10^{-9}\text{s}} = 12.5 \cdot 10^6 \text{ Zugriffe/s}$
- Bus: 50 MHz und 8 Byte Busbreite führt zu 400 MB/s
- im SIMM müssen also 32 Bit-Zugriffe „parallel“ erfolgen
- externer und interner Cache bilden einen Puffer, der mit „look ahead“ bei einer Trefferquote von $\sim 90 - 95\%$ betrieben wird.
- Es handelt sich um eine „schnelle CPU“ mit 100 MHz; also ergeben sich bei einem Durchsatz von 1200 MB/s durchschnittlich $\leq 12\text{B}$ pro Taktperiode ≤ 3 Worte/Eintakt-Befehl (32 Bit Befehl und 64 Bit Daten).

Hoher Durchsatz wird erreicht bei:

- sequentieller Ausführung von Befehlen,
- Zugriff auf ganze Vektoren (Daten),
- Zugriff auf 64/128 Bit-Gleitkommazahlen bei 32 Bit Datenbusbreite,
- durchschnittlich geringem Bedarf an Daten (aus dem Speicher) pro Befehl

Geringer Durchsatz wird erzielt bei:

- wilden Sprüngen, ...
- Benutzung von verketteten Listen mit „großen“ Knoten (CAD-Programme), ...

Bemerkung: Es wurden SparcStation10's ohne SuperCache ausgeliefert, um bei speziellen Anwenderprogrammen das „verlangsamende“ Cache-Füllen zu umgehen.

1.4.2 Vektorrechner

Statt der Verarbeitung einzelner Worte pro Taktperiode versucht man ganze Vektoren/Takt zu verarbeiten (Vervielfältigung der ALU):

Abbildung 1.13: Verarbeitungseinheit: Vektor von Worten

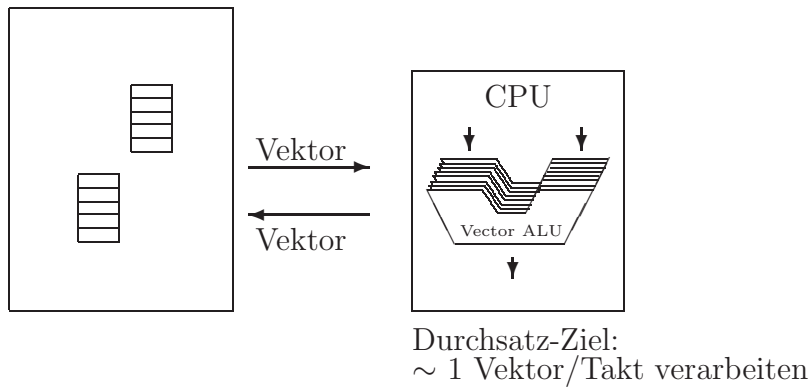
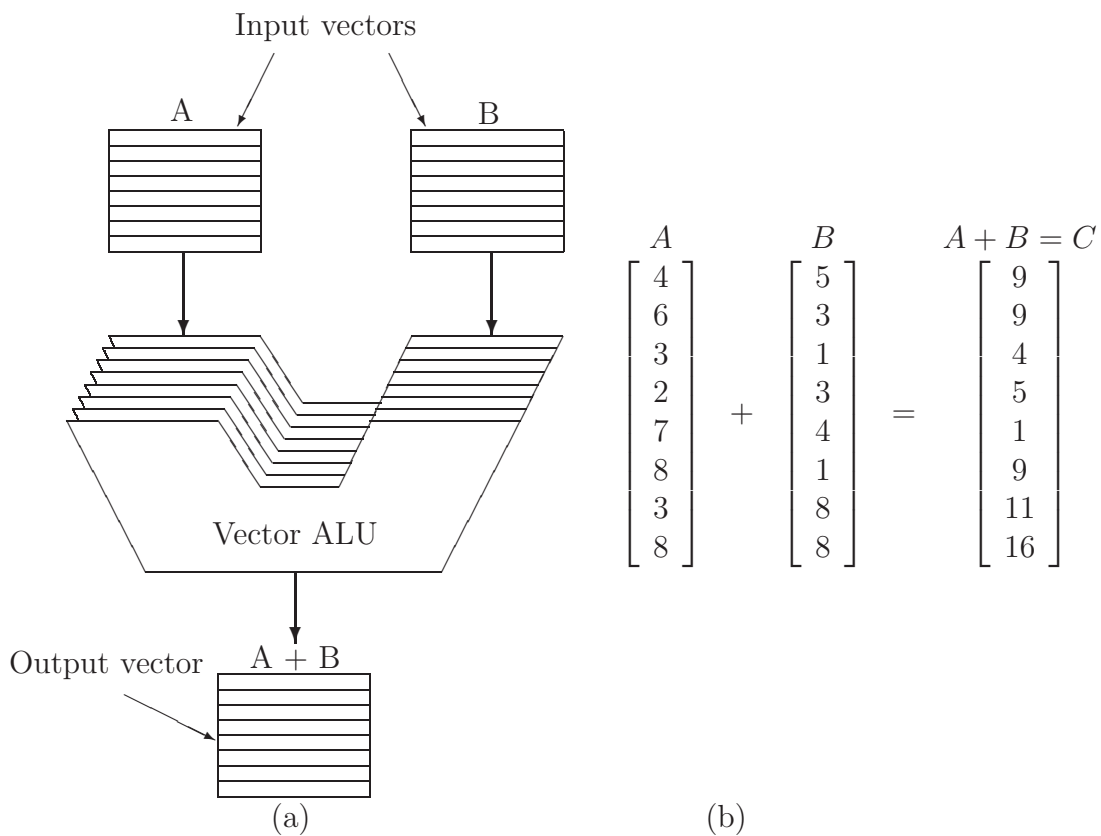


Abbildung 1.14: Vektorregister und Vektoralu



(a) Vektor-ALU.

(b) Beispiel für die Vektor-Addition.

Andere Beispiele für Vektoroperationen:

- $C_i = f_1(A_i)$ mit $f_1 \in \{\cos, \sqrt{}, \dots\}$ komponentenweise ausgewertet
- $s = \sum_{i=1}^N A_i$ Summe, Minimum, Produkt
- Minimum
- Produkt
- Maximum
- \vdots

N ALU's sind i.a.:

- zu teuer, da jede einzelne ALU sehr schnell sein soll
- unflexibel in der Vektor-Länge

Kompromiß: „eine ALU im Pipelinebetrieb statt eines Vektors von ALU's“

FPU-Pipeline (baue ALU aus „unabhängigen“ Einheiten auf):

Beispiel: Subtraktion

1) fetch operands	$1.082 \cdot 10^{12}$, $9.212 \cdot 10^{11}$
2) adjust exponent	$1.082 \cdot 10^{12}$, $0.9212 \cdot 10^{12}$
3) execute „-“	$0.1608 \cdot 10^{12}$
4) normalize result	$1.608 \cdot 10^{11}$

$$C_i = A_i - B_i \quad \forall i = 1 \dots N$$

Zyklus	1	2	3	4	5	...
fetch	B_1, C_1	B_2, C_2	B_3, C_3	B_4, C_4	B_5, C_5	
adjust	—	B_1, C_1	B_2, C_2	B_3, C_3	B_4, C_4	
execute	—	—	$B_1 - C_1$	$B_2 - C_2$	$B_3 - C_3$	
normalize	—	—	—	$B_1 - C_1$	$B_2 - C_2$	

\swarrow \searrow
 $B_1 - C_1$ $B_2 - C_2$

Resümee:

- Ein Einzelbefehl (z.B. Subtraktion) braucht 4 Zyklen Zeit.
- Nach einer Startup-Zeit von 4 Zyklen (Füllen der Pipe) wird dann jedoch pro Zyklus ein Endergebnis geliefert. (Das ist allerdings nur eine Komponente des Ergebnis-Vektors.)
- Alte Vektorrechner benötigen eine Vektorlänge ≥ 100 , damit sie schneller als ein Skalarrechner arbeiten; neuere Vektorrechner kommen jedoch schon mit einer Vektorlänge von $\geq 2 \dots 4$ aus.
- **Problem:** „Datenabhängigkeit“

Ein Wert ist noch in der Pipe, obwohl er schon als neues Argument benötigt wird.

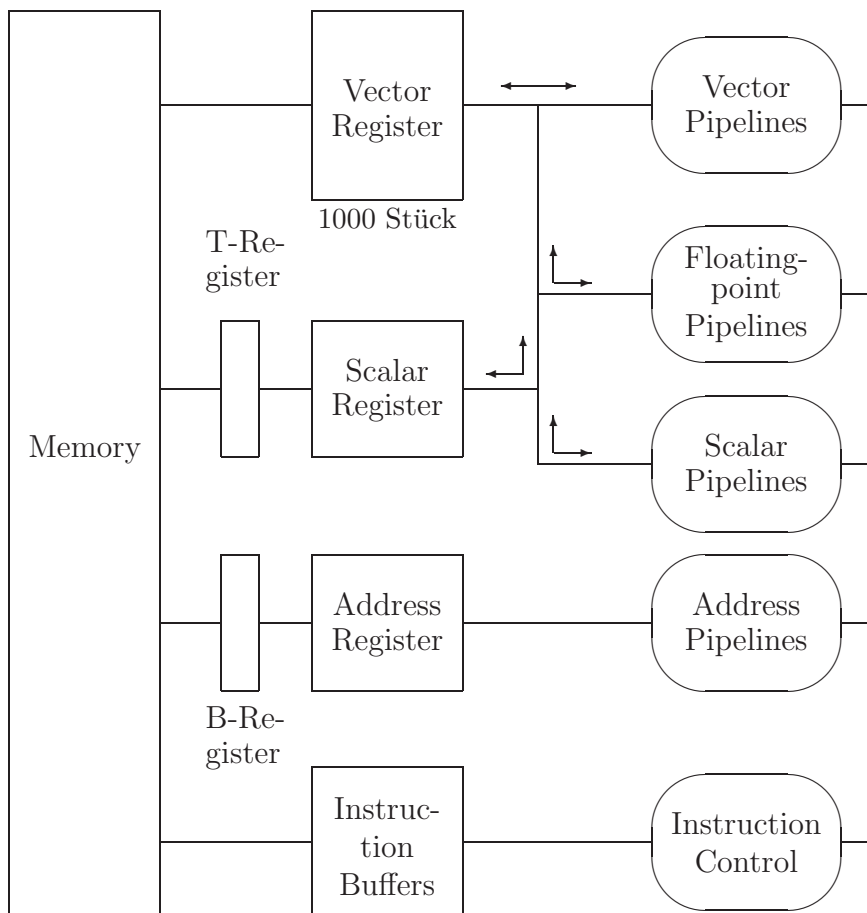
- Jede ALU arbeitet denselben Befehl ab (SIMD = „single instruction multiple data“)
- COMPILER vektorisieren z.T. automatisch.

Beispiel: Cray-1 (1976)

mehrere parallel arbeitende Pipelines,
viele Register,
Taktzeit ~ 12.5 ns

} peak_Leistung: 250 MFlops

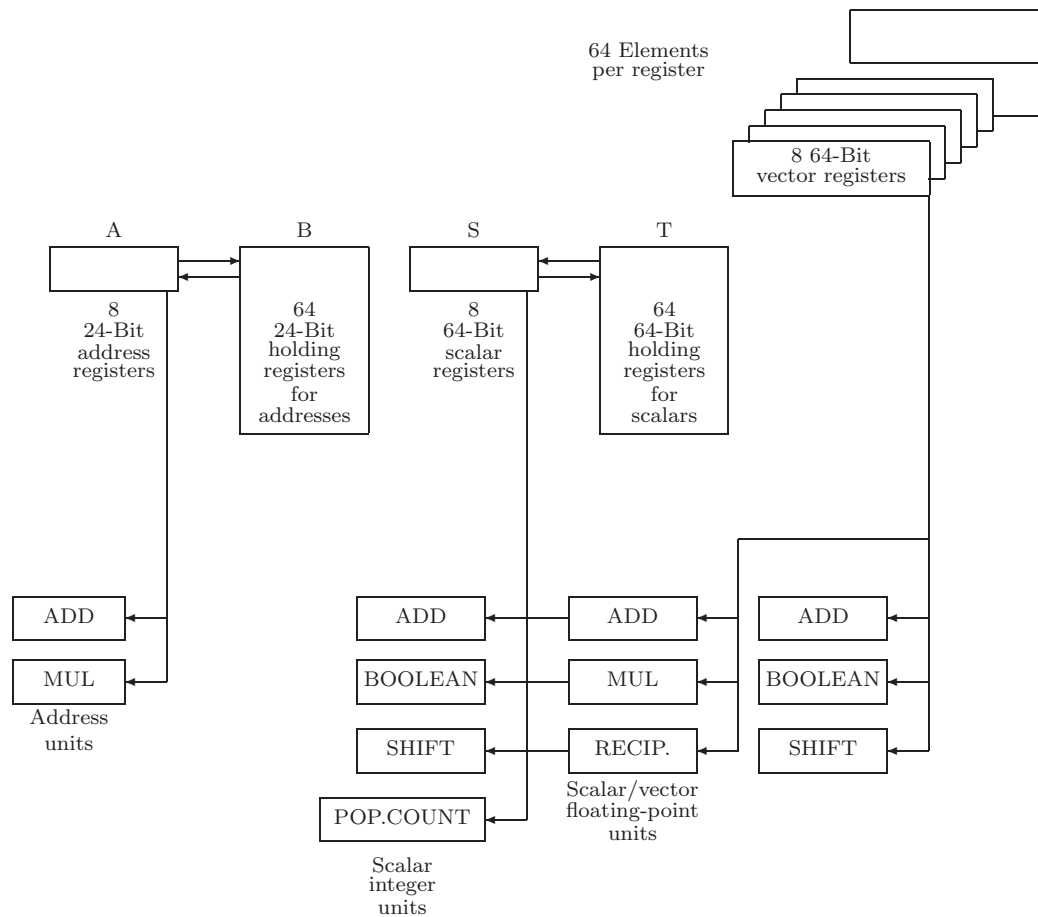
Abbildung 1.15: Rechnerstruktur eines Cray-1-Rechners



Chaining von Pipelines

- | | | |
|-------------------------|--------------------------|------------------------------|
| Vector Pipelines | – Additionen | } von Ganzzahlvektoren |
| | – Verschieben | |
| | – logische Operationen | |
| Floatingpoint Pipelines | – Additionen | } von Gleitkommazahlvektoren |
| | – Multiplikationen | |
| | – Divisionen | |
| Address Pipelines | – Adreß-Multiplikationen | |
| | – Adreß-Additionen | |

Abbildung 1.16: Die Register und Funktionseinheiten einer Cray-1



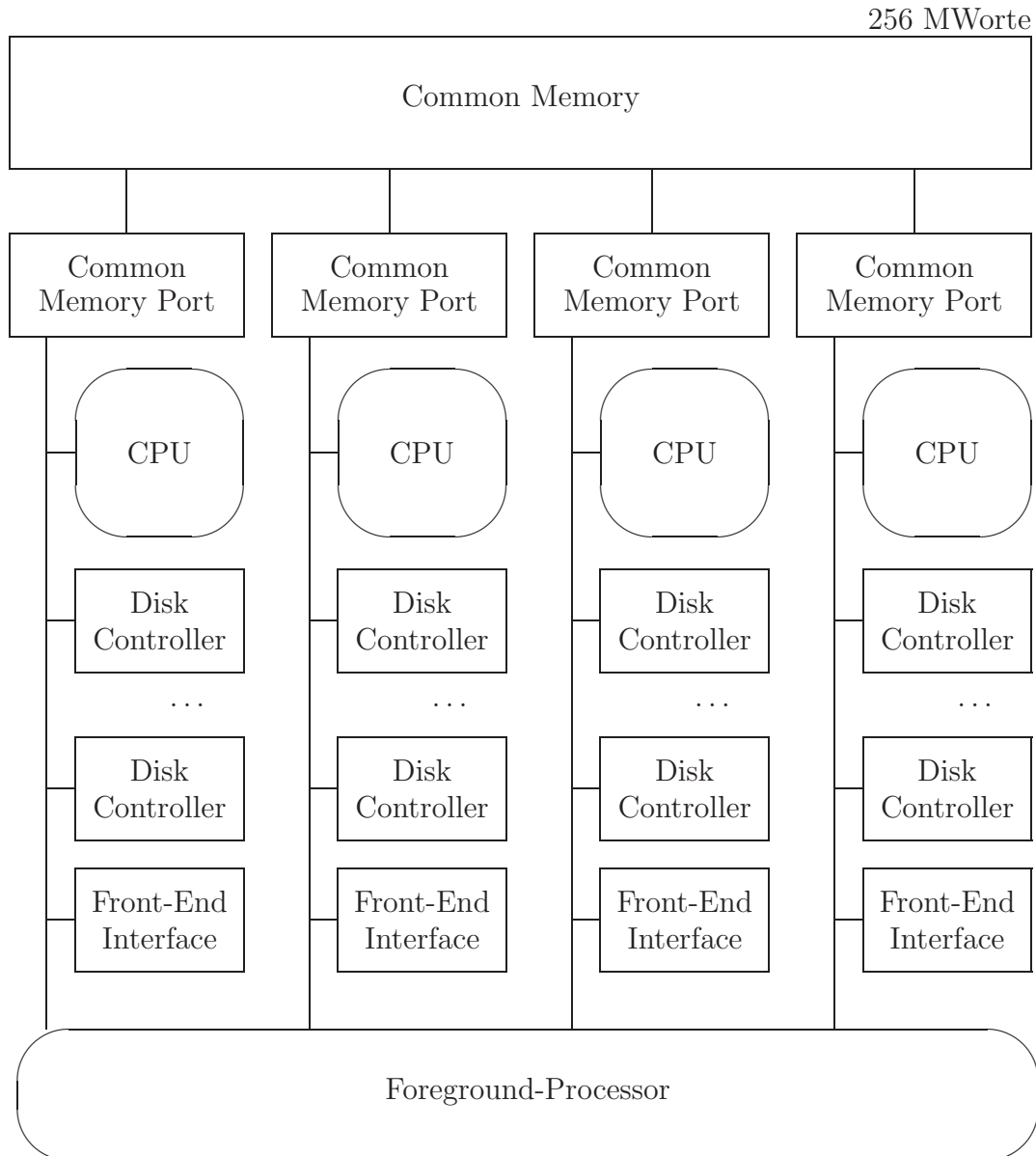
Weitere Geschwindigkeitssteigerungen¹ sind erreichbar durch

ein speichergekoppeltes **Multiprozessorsystem**, das mehrere Zentraleinheiten (MIMD = „multiple instruction multiple data“) oder nur mehrere ALU's (SIMD = „single instruction multiple data“) enthalten kann.

¹ Selbst heutige PC-CPU's enthalten parallel benutzbare Funktionseinheiten (ein oder mehrere Integer-Units, ein oder mehrere FP-Units, . . .), die häufig im Sinne einer Pipeline arbeiten (z.B. beim Pentium: zwei 32 Bit-Integer-Pipelines und eine FP-Pipeline). Naturgemäß sind Programme aus wenigen einfachen Befehlen viel besser parallel zu „dispatchen“ (zur parallelen Abarbeitung vorzubereiten und „in Marsch zu setzen“), weshalb das CISC-RISC-Translatorprinzip des PentiumPro zwar genaugenommen einer Microcode-CPU entspricht, aber das Design des „Microcodes“ als RISC-Code bessere Parallelisierbarkeit und deshalb schnellere CPU's ermöglicht (PentiumPro: dreiweg Superskalarausführung, „speculative execution“, „branch prediction“, 14-stufige Superpipeline).

1.4.3 SIMD- und MIMD-Parallelrechner

Abbildung 1.17: CRAY-2 (Juni 1985)



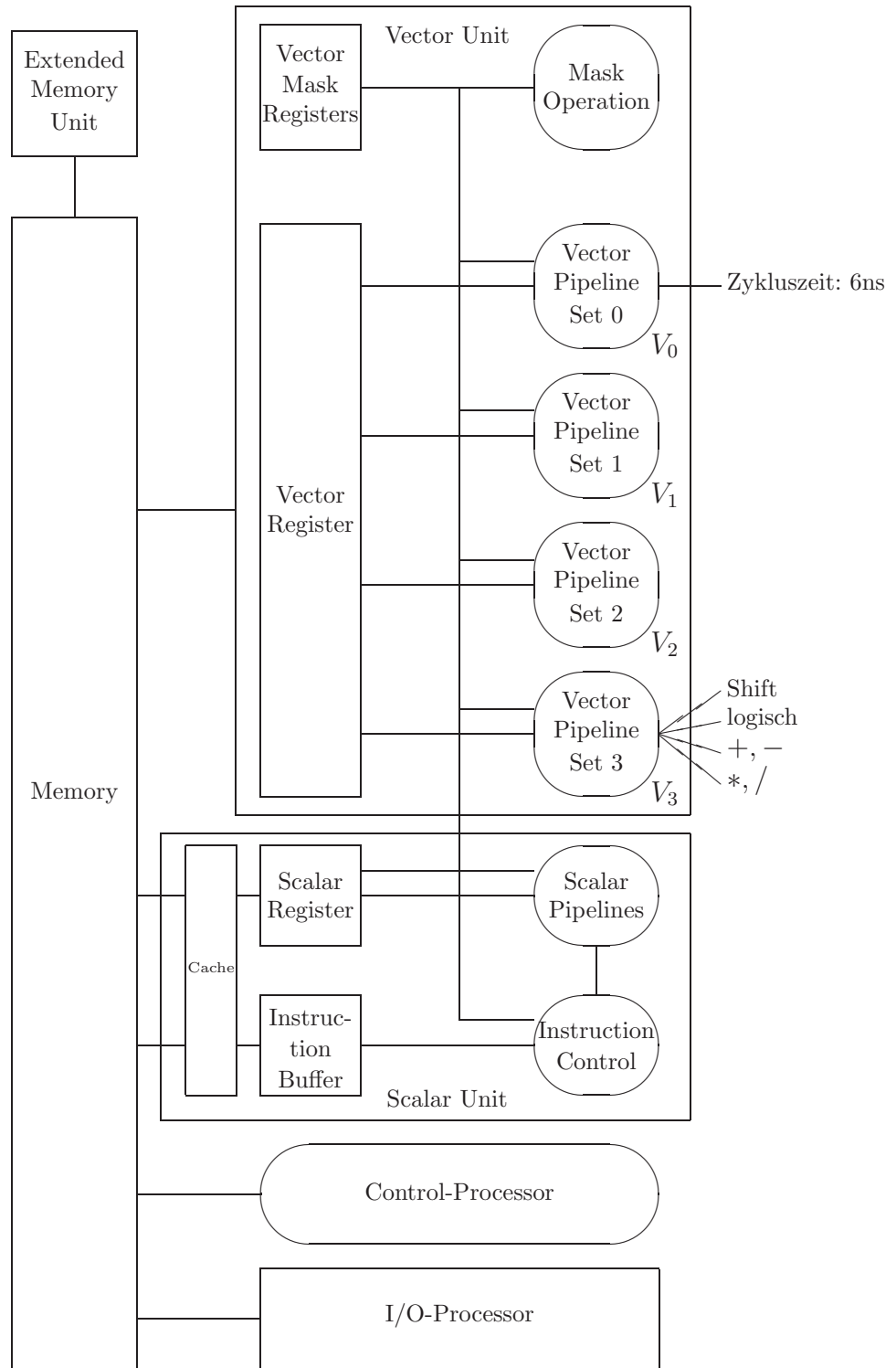
Rechnerstruktur eines Cray-2-Rechners 6 ... 12× schneller als Cray-1

Cray-3	16 CPU's mit 2ns Taktzeit	
Cray-4	64 CPU's mit 1ns Taktzeit	~ 128 GFLOPS (~ 1000× Cray-1)
Cray T90 ²	32 CPU's mit je 2 Vektorpipelines	~60 GFLOPS
Cray Origin 2000	2048 CPU's	~2.5 TFLOPS
Cray T3E-1200 ³	2048 CPU's	~2.5 TFLOPS

²Vgl. auch <http://www.cray.com/hpc/>

³Vgl. auch <http://www.cray.com/products/systems/crayt3e/1200/>

Abbildung 1.18: NEC-SX2 SIMD, mehrere Pipelines



Überlappende Abarbeitung der Vector-Pipeline-Sets:

V_0 1,5,9, ...
 V_1 2,6,10, ...
 V_2 3,7,11, ...
 V_3 4,8,12, ...

nichtsichtbar, automatisch
Start-Up-Zeit: 60ns

NEC SX-3/11 mit:

- 256MB Hauptspeicher
- 1GB Erweiterungsspeicher (max. 2GB)
- 20GB Platte
- IBM 3084Q als „front end“ Rechner
- 2.9 ns Zykluszeit
- skalar: 170 MFlops peak, vektoriell: $4 \times 345 \text{MFlops} = 1.386 \text{GFlops}$ pro Vektorpipeline

SX-3/44 mit:

4 Pipelines pro CPU \rightarrow 5.5 GFlops/CPU
4 CPU's in Modell 44 \rightarrow 22 GFlops

NEC SX-4 mit: ⁴

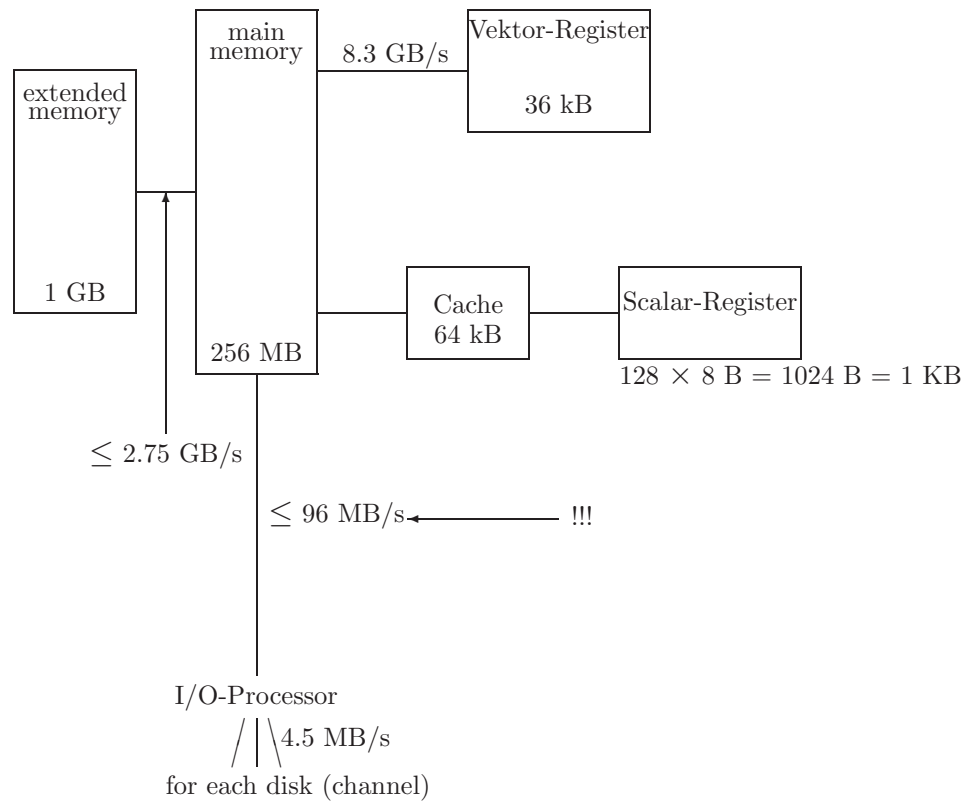
- 512 CPUs mit je 8 Pipelines \rightarrow 1 TFlops⁵

⁴Vgl. <http://www.hpc.comp.nec.co.jp/sx-e/sx-world/no18/report.html>

⁵Vgl. <http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi?prefix>

Kritisch bei High-Performance Rechnern ist auch heute noch, das I/O-Subsystem mit einer geeignet hohen Durchsatzrate an die CPUs zu koppeln:

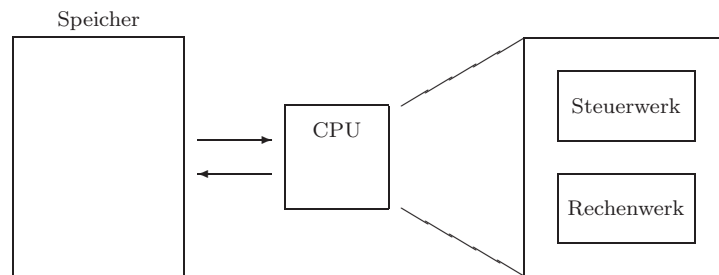
Abbildung 1.19: Systemarchitektur und Datendurchsatz



Einen Überblick zu den jeweils aktuell leistungsstärksten Supercomputern bietet:

<http://www.top500.org/>

1.4.4 Mehrprozessorsysteme



Welche der Systemeinheiten wird vervielfältigt?

- ALU
- ALU und Steuerwerk
- ALU, Steuerwerk und Speicher

Tabelle 1.11: Klassifikation von Mehrprozessorsystemen

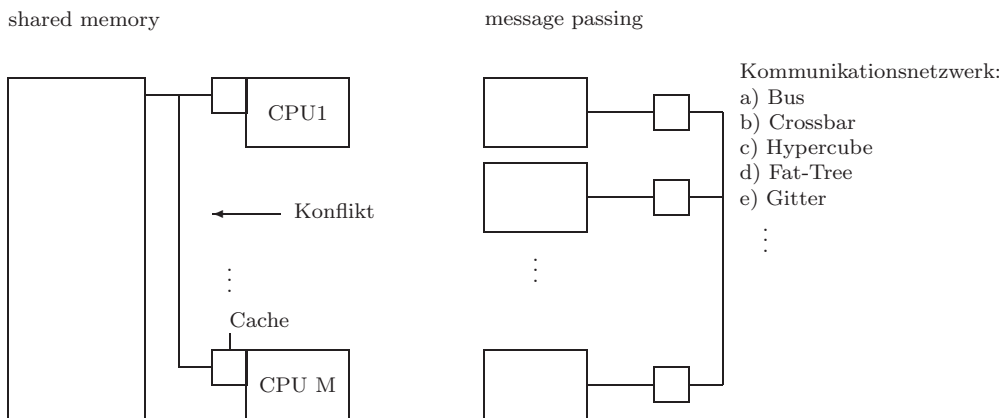
	CPU's		Memory	Beispiel
	Steuerwerke	Rechenwerke		
von Neumann	1		1	„alte“ PCs
Vektorrechner	1	N	1	Cray-1
Multiprozessor-Vektorrechner	M (mit je N Rechenwerken)		1	Cray-2

Bemerkung: Rechenwerke oder Pipelines

Tabelle 1.12: Speicher bei Mehrprozessorsystemen

Multi- prozessor mit	CPU's		Memory	Beispiel
	Steuerwerke	Rechenwerke		
shared memory	M		1	AlphaServer
message passing	M		M	Transputersysteme
„Mischung“	M		M+1	...

Abbildung 1.20: „shared memory“ und „message passing“



Unter M+1 soll verstanden werden: „Ein gemeinsamer und je ein lokaler Speicher.“

Zuweilen klassifiziert man Multiprozessorsysteme durch folgendes zweidimensionale Schema:

SW \ HW	homogen	inhomogen
symmetrisch		
unsymmetrisch		

Häufig: 1 Steuerprozessor und N Arbeitsprozessoren

Aktuell wird daran geforscht, zwischen „shared memory“ und „distributed memory“ (=message passing systems) im sogenannten „virtuellen distributed memory“ den Effizienzvorteil des „distributed memory“ mit der einfacheren Bedienbarkeit des „shared memory“ zu kombinieren.

Kopplung der Komponenten: schnelle Busse

Zum Beispiel bei Sun-Workstations:

- **SBus:** für I/O, zum Beispiel: SCSI
 - offener Chip-Bus, direkt mit internem CPU-Bus verbunden
 - keine Treiberlogik auf Karte nötig (je ≤ 10 Watt)
 - keine Jumper nötig (Selbstidentifikation)
 - 96-poliger Stecker, 32 Bit Daten
 - ≤ 100 MB/s (aktuell 20 ... 25 MB/s)

} circuit switched
(Wort-Transfer ist blockierend)

- **MBus:** (Design für ≤ 4 Prozessoren – SPARCStation 10)
 - offener, schneller Memory Bus
 - 40 MHz Takt
 - 64 Bit Daten
 - ≤ 320 MB/s (effektiv 105 MB/s)

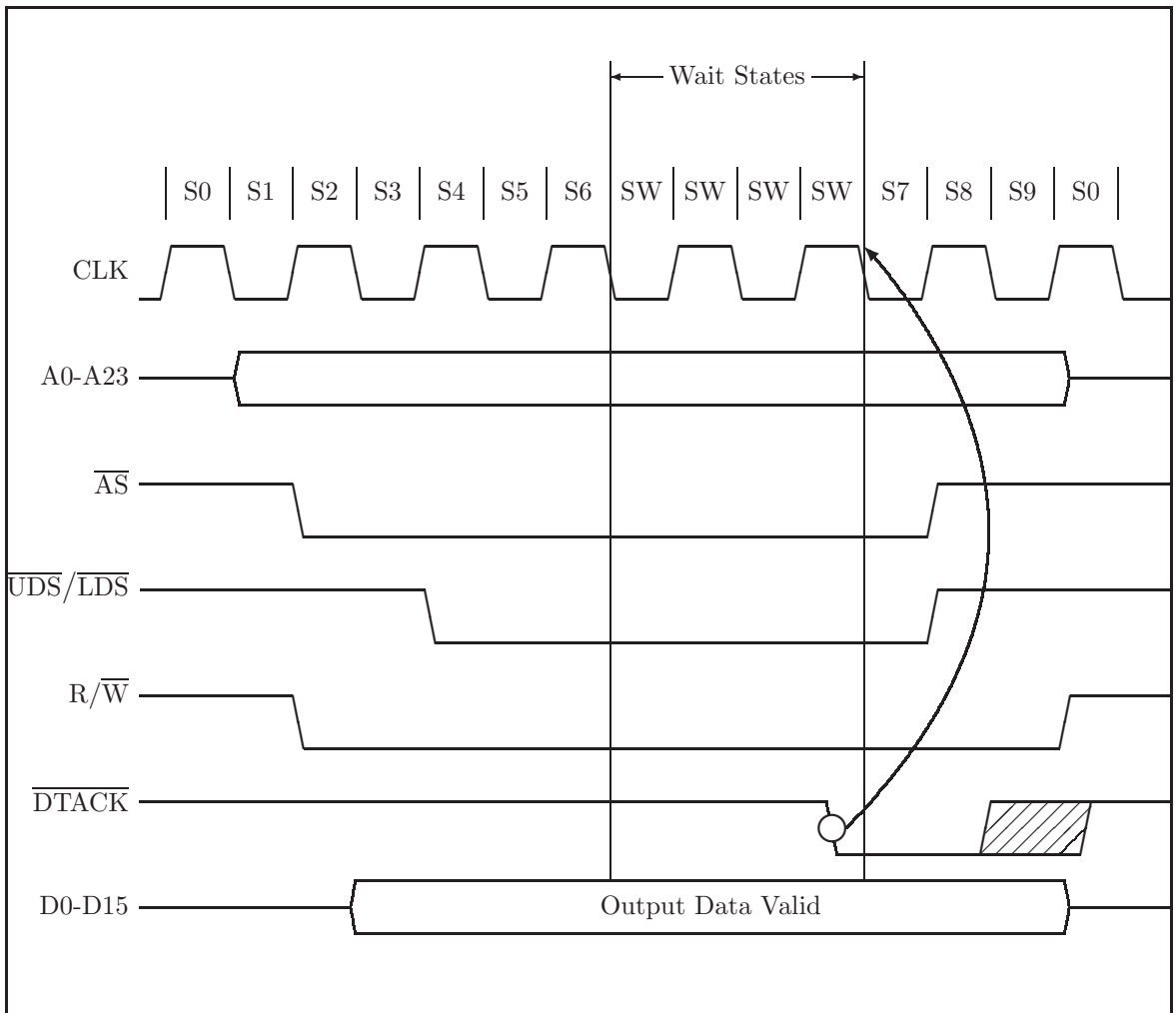
} circuit switched (Wort-Transfer ist blockierend)

- **XDBus:** (für SPARCServer 2000/1000, Design für ≤ 8 Prozessoren)
 - offener, schneller Memory/CPU-Bus
 - 40 MHz
 - 64 Bit Daten
 - 320 MB/s (effektiv 250 MB/s)

packet switched, deshalb: überlappende Arbeitsweise möglich

„circuit switched“ Datentransfer: Wait States

Abbildung 1.21: Zeitdiagramm „circuit switched“ Datentransfer



Wait States während einer Schreib-Operation

Bemerkungen:

- Signalnamen:

CLK	=	clock
A0–A24	=	address bus
AS	=	address strobe
R/ \overline{W}	=	read/write
UDS,LDS	=	upper/lower data strobe
D0–D15	=	data bus
DTACK	=	data transfer acknowledge

- strobe = Gültigkeitssignal für z.B. die Daten auf dem Adreßbus

Zusammenfassend:

- circuit switched: Der Bus blockiert, bis die anstehende Operation ausgeführt ist: „wait states“
- packet switched⁶: Jeder Datentransfer wird in kleine Pakete zerlegt. Diese werden nacheinander nichtblockierend auf den Bus geschickt. Dadurch kann eine weitere CPU (ein weiterer Thread) schon wieder Daten transferieren, obwohl die erste CPU ihren Transfer noch nicht abgeschlossen hat.

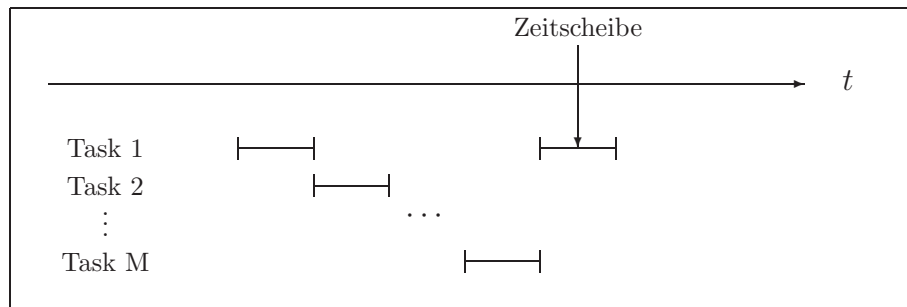
Problem: Steuerung der CPU nichtsequentiell (mehrere Prozesse lösen sich beim Warten ab, Threads, ...)

⁶Der USB-Anschluß (*Universal Serial Bus*) an neueren PCs bietet eine paketorientierte Datenübertragung mit bis zu 12 Mbps zur Kommunikation von CPU und Peripherie. Insbesondere ermöglicht er die Übertragung von Audiodaten mit zuvor fest ausgehandelter Bandbreite.

Multiprozes- und Multiprozessorsysteme

Prozesse (Tasks)

- Programmeinheiten, die unterbrech- und wiederfortsetzbar sind
- Multitasking:
1 Rechner erscheint M Benutzern als M jeweils dedizierte einzelne Rechner. Ein Rechner kann (scheinbar) zugleich mehrere Prozesse bearbeiten.



Beispiele:

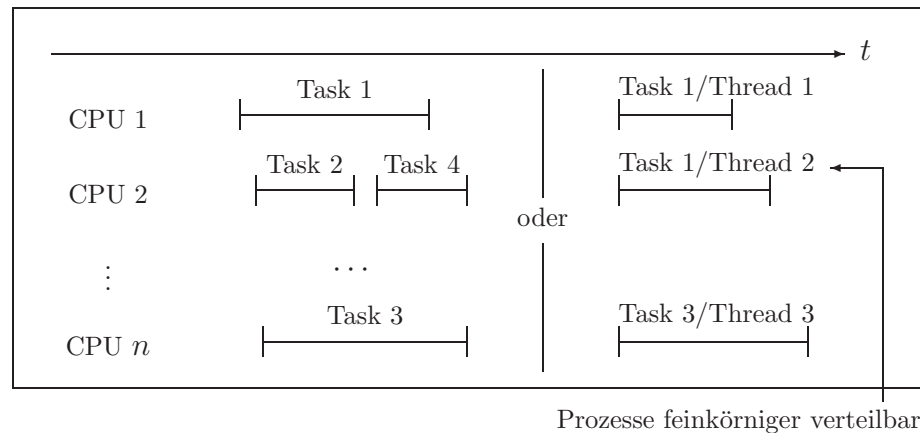
Windows 3.x: Multitasking, nicht preemptiv.

OS/2, Windows 95, Windows NT, UNIX: preemptives Multitasking.

Preemptiv (= „wahlweise vergebbar“) heißt ein Multitasking-Betriebssystem dann, wenn der Betriebssystemkern allein für die Vergabe der Zeitscheiben verantwortlich ist, also nicht auf die kooperative Mitarbeit der Tasks angewiesen ist. Die Tasks werden von „außen“ unterbrochen und später wieder fortgesetzt.

Threads

- parallele, unabhängig durchführbare Teile eines Prozesses, die deshalb auf Multiprozessorsystemen (\neq Multiprozesssystemen) durch verschiedene CPU's durchgeführt werden können: An der Aufgabe eines Prozesses arbeiten mehrere Prozessoren zugleich.



1.4.5 Multiprozessor-Server

Die Programmierung von Anwendungen, die mehrere Prozessoren im MIMD-Modus nutzen sollen, ist i.a. sehr arbeitsaufwendig. Deshalb gibt es auch nicht viele solche Programme, die auf jedem Knoten wirklich unterschiedliche Algorithmen ablaufen lassen.

Einfacher ist es zum Beispiel, ein Betriebssystem so zu konzipieren, daß bei einem Mehrbenutzerbetrieb die einzelnen Prozesse und Threads jeweils einzelnen Prozessoren zur Abarbeitung übergeben werden: Multiprozessor-Server mit hohem Prozeß-/Thread-Durchsatz.

Für solche DV-Anlagen ist dann jedoch eine hohe Verfügbarkeit unbedingte Voraussetzung: unterbrechungsfreie Stromversorgung, Hotswap, redundante Komponenten, ...

Der hohe Preis von solchen Servern ist jedoch nicht immer akzeptabel, so daß zur Zeit an *verteilten Betriebssystemen* gearbeitet wird, die mehrere billigere PCs oder Workstations zu einem virtuellen Mehrprozessorserver vereinigt erscheinen lassen: Man meldet sich dann an einem „virtuellen“ Rechner an und arbeitet auf einem gerade freien Rechner des ganzen Clusters. Beim nächsten Anmelden arbeitet man eventuell auf einem anderen Rechner, was man jedoch überhaupt nicht merkt, da das Betriebssystem die ganze Arbeitsumgebung unabhängig von dem gerade wirklich benutzten Rechner einheitlich erscheinen läßt.

Wirkliche Multiprozessor-Server sind dann nur noch in den Fällen wirtschaftlich, wenn in Abteilungsservern spezielle Dienste (Datenbanken, WWW-Intranets, Fileexport (NFS), ...) bereitgestellt werden müssen.

Aktuelle Daten zu Multiprozessor-Servern findet man unter:

<http://www.sun.de/Produkte/Hardware/index.html>
<http://www.sgi.de/produkte/index.html>
<http://www.compaq.com/products/servers/index.html>
<http://www.hewlett-packard.de/enterprise/>
<http://www.ibm.com/products/>
...

1.4.6 Massiv parallele Computersysteme

Am Beispiel der CM2:

Technische Daten:

SIMD

≤ 64 K Prozessoren (*Bit-slice*)

≤ 10 GFlops

≤ 256 Kbit/Prozessor (d.h. ≤ 2 GB) Hauptspeicher

pro 32 Prozessoren je 1 Gleitkommaprozessor,

d.h. ≤ 2 K Gleitkommaprozessoren à 32 Bit / 64 Bit

≤ 60 GB Data Vault

Hier in Wuppertal (zur Zeit: 4 K Prozessoren, kein Data Vault):

8 K Prozessoren, d.h. 256 Gleitkommaprozessoren à 64 Bit

8 K · 256 KBit = 256 MB Hauptspeicher

$\leq 1,25$ GFlops

10 GB Data Vault

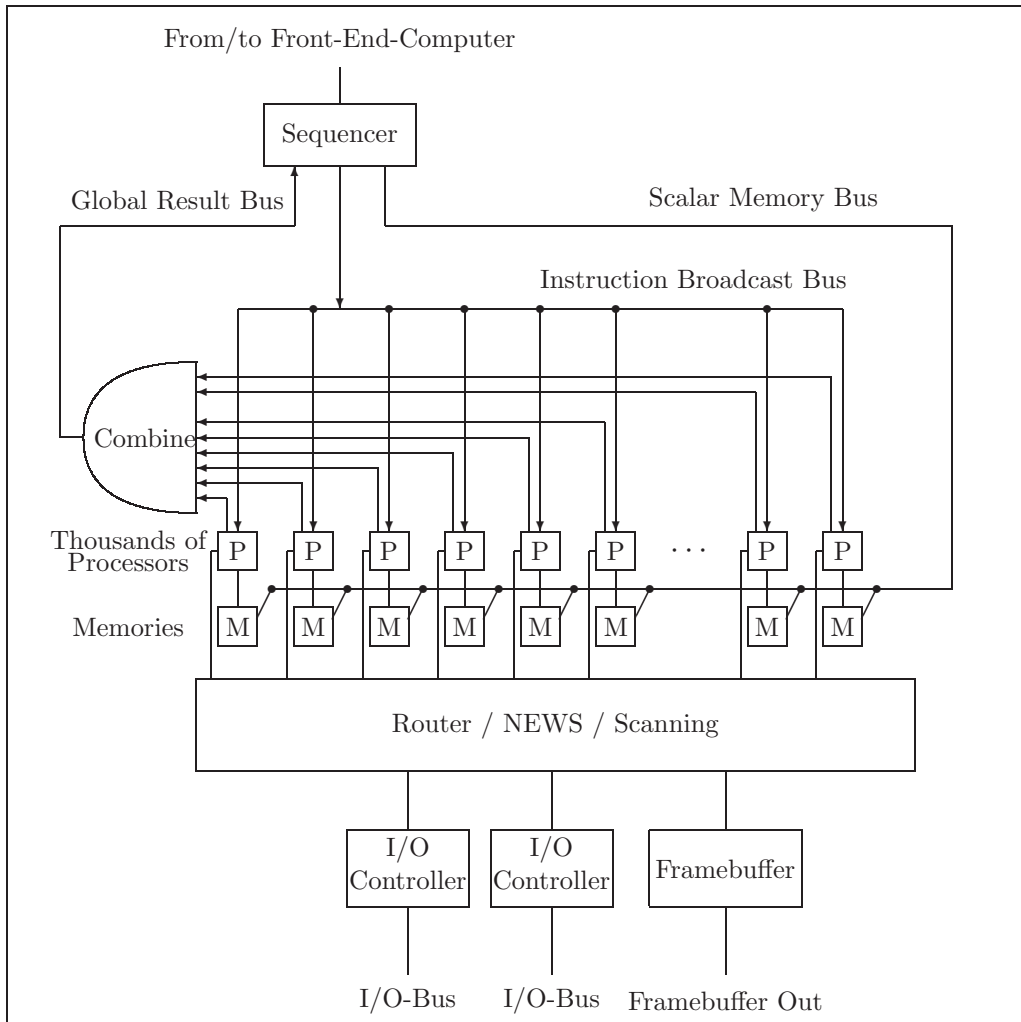
Bemerkungen:

- „message passing“ durch Nachrichten über NEWS
- 12-dim Hypercube

- Datavault: $\underbrace{32}_{\text{Bits/Wort}} + \underbrace{7}_{\text{error correction}} + \underbrace{3}_{\text{Reserve}} = 42 \text{ HD's}$
„fault tolerant disc system“

- Framebuffer: für je 32 Prozessoren existiert 1 Leitung zum Display = 256 Leitungen bei 8 K Prozessoren

Abbildung 1.22: Schaltschema der CM2



Am Beispiel der CM5 (CM5E-LD):

- Zur CM2 kompatible SIMD-Maschine:
„Vektor“-Operationen:

```
shape [16384]t;  
int:t p1,p2,p3;  
with (t)  
    p3 = p2 + p1
```

- MIMD-Modus: 32-Prozessor-System (hier in Wuppertal)

32 Prozessoren mit je 4 VU's (Vektorunits à 64 Bit) und einem SuperSPARC-Knoten (40 MHz) für Skalaroperationen, ≤ 5 GFlops peak-Leistung (realistisch: $\sim 40\%$),
1 GB Hauptspeicher (32 MB pro PN – processing node –)
24GB scalable disk array (SDA)

```
message passing (broadcast, ... ) und verschiedene In-  
struktionsströme für verschiedene Prozessoren
```

Probleme bei SIMD und MIMD:

Engpaß bei SIMD:

```
with(t)  
    where (B(p1))  
        ...  
    elsewhere  
        ...
```

← Gesamtausführungszeit
= Zeit_für_where + Zeit_für_elsewhere

Problem bei MIMD:

Synchronisation!

Als Ausweg ein Kompromiß:

SPMD (= „same program multiple data“) — „data parallel programming“

- außer in where/elsewhere-Zweigen reines SIMD
- where- und elsewhere-Zweige laufen parallel zur selben Zeit
- where/elsewhere-Zweige werden gleichzeitig gestartet
- automatische Synchronisation am Ende von where/elsewhere
- zum Teil wird auch dasselbe Programm auf alle Knoten verteilt, wo es dann jeweils zu einem Zeitpunkt an verschiedenen Stellen der Befehlsausführung angekommen sein kann (unterschiedliche Fallunterscheidungswege ...)

Aktuelle massiv parallele Computersysteme sind Z.B. die Cray-T3E und die NEC SX-4 (vergleiche Abschnitt 1.4.3).

Aktuelle Entwicklungen:

<http://www.theorie.physik.uni-wuppertal.de/Computerlabor/ALiCE.phtml>
- Cluster-Computing mit handelsüblichen Rechnern als Knoten

<http://parastation.ira.uka.de/> - Paralleles verteiltes Rechnen

1.4.7 Anwendung: Die massiv parallele Simulation

Die massiv parallele Simulation stellt ein leistungsfähiges Tool zu
Analyse, Verifikation, Optimierung und Erforschung

komplexer Systeme in wirtschaftlichen/sozialwirtschaftlichen/
... Disziplinen dar; sie ermöglicht Experimente. Beispiele:

- Prognosen für komplexe ökonomische Systeme,
- Optimierung von Fertigungsanlagen, Kraftwerken, Produktionsstätten, ...
- nichtlineare dynamische Systeme als Modelle
(Hier sind analytische Methoden häufig zu aufwendig oder nicht existent.)
 - ↓ chaotische dynamische Systeme

Lit.: H.-W. Lorenz

Simulation komplexen Systemverhaltens in der nichtlinearen dynamischen Wirtschaftstheorie,
Simulationstechnik — 7. Symposium in Hagen, 9/91,
Bd.4, S.104

- Lager- und Montagesysteme
- Werkzeugsteuerung
- Prozeßleitsteuerung
- Betriebsleitebene von Kraftwerken (Simulation von Störfällen, ...), auch als Lehrinstrument ohne materielle Gefahren

Komplexe Systeme sind meist aus Teilsystemen zusammensetzbar, die einzelnen Prozessoren zugeordnet werden können und deshalb nicht schwierig zu parallelisieren sind.

1.4.8 Klassifikation von parallelen Rechnern

Datenstrom \ Befehlsstrom	ein	mehrere	
ein	SISD (=von Neumann)	MISD	← ?
mehrere	SIMD	MIMD	← CM5

↑
CM2

# Computer \ Memory	unshared	shared	
ein Computer	Nachrichtenaustausch (message passing)	shared memory	
n Computer	Message Passing in distributed computing	distributed virtual memory	← vgl. Kapitel 3

↑
LINDA, ...

Speicher \ Kommunikation	gemeinsame Variablen	Nachrichtentausch	
globaler Speicher	GM \underbrace{SV} shared variables	GM \underbrace{MP} message passing	→ global memory
verteilter Speicher	DMSV	DMMP	→ distributed memory

↑
CM5

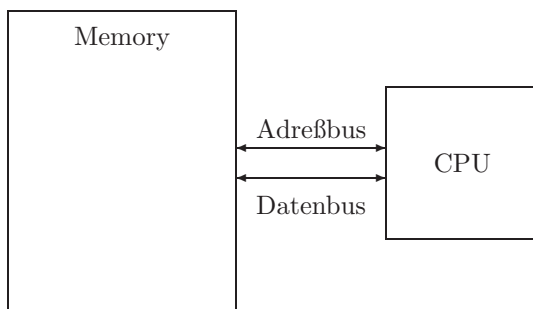
1.5 Speichermodule (SIMM's, ...)

S = single
I = inline
M = memory
M = module

Zum Aussehen und zum Einbau von SIMM's vergleichen Sie bitte den Abschnitt über SIMM's in Kapitel 1.3 „*Field replaceable units*“.
DIMM's, Rambus-RAM, ...

1.6 Assoziative Speicher

Die sequentielle Adressierung als *Flaschenhals*:



Probleme bei der Suche nach assoziierten Daten mit Hilfe von „von-Neumann Rechnern“:

- Suche nach der Telefonnummer von „Meyer, A.“:

a) sequentielle Suche:

Adresse	Name	Nummer	
⋮	⋮	⋮	...
1012	Meyer, A.	237704	
1112	Adler, K.	1244	
1212	Schulte, B.	124466	
⋮	⋮	⋮	...

```

⋮
ResetListe;
Found := false;
Repeat
  GetNextRecord(R);
  if R.Name = 'Meyer,A.' then
    Found := true;
until (Found = true) or EndofListe;
⋮

```

Zeitaufwendig, da im schlimmsten Falle alle Listenelemente ausprobiert werden müssen!

- b) Im Falle eines nach dem Schlüsselfeld alphabetisch geordneten Datenbestandes: binäre Suche

55 > 8:

1	...
3	...
7	...
8	...
12	...
55	...
101	...

55 > 12:

8	...
12	...
55	...
101	...

55 = 55:

12	...
55	...
101	...

```

:
ResetGesamteListe;
Found := false;
Repeat
  GetMiddleListElement(R);
  if R.Name = 'Meyer,A.' then
    Found := true;
  else if 'Meyer,A.' < R.Name then
    Liste := erste Halbliste
  else
    Liste := zweite Halbliste
until (Found = true) or (Liste = leer);

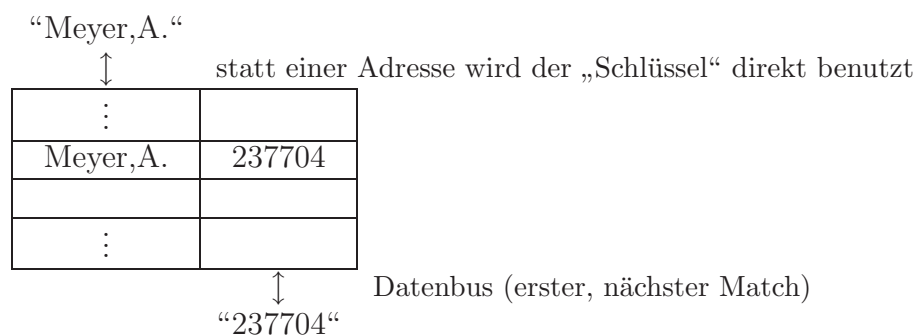
```

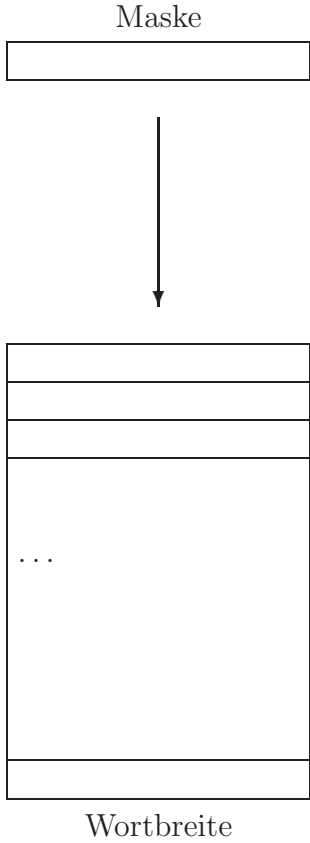
In einem sequentiell adressierten Computer ist die Anweisung

Liste := zweite Halbliste

nur mit zusätzlichem Verwaltungsaufwand effektiv realisierbar (*Suchbaum*).

Alternative: „Finde **per Hardware** alle *Listenelemente* mit bestimmten Eigenschaften“ in nur „einer“ Taktperiode — die assoziative Adressierung:





Wortseriell und bitparallel organisierter Assoziativspeicher

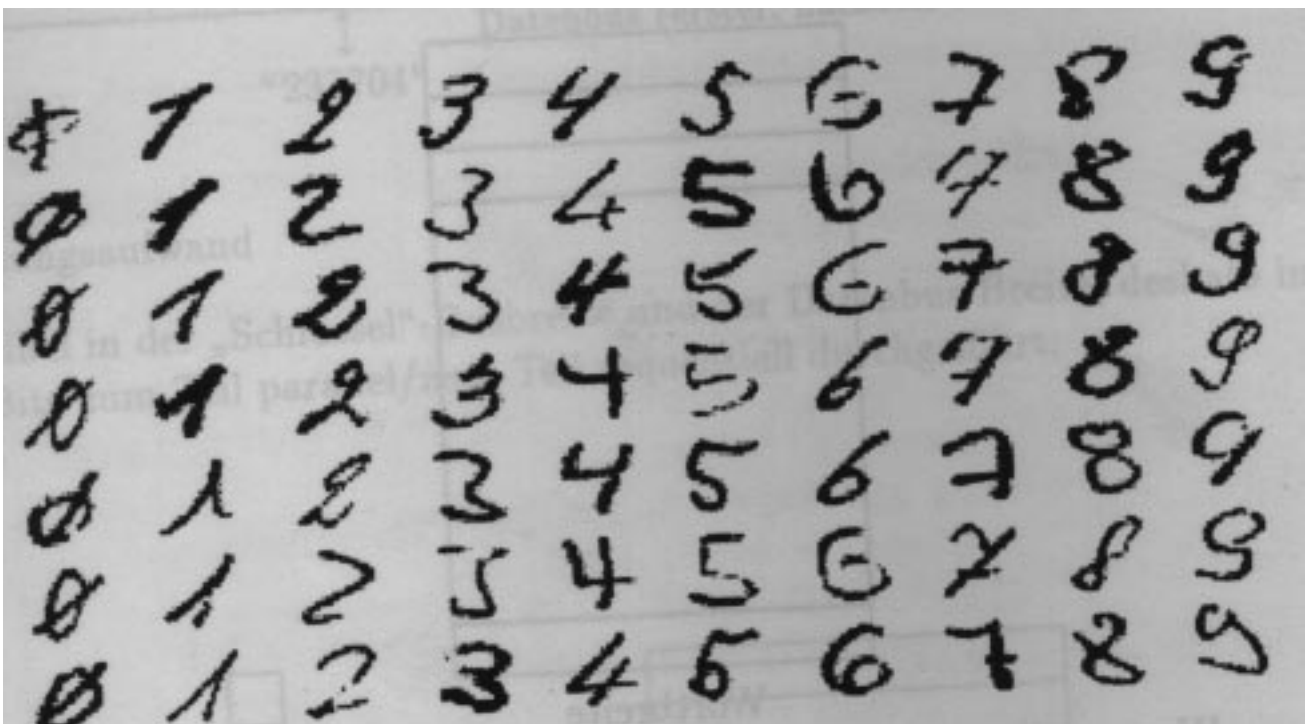
In einer Taktperiode wird in einer Serie von Worten genau ein Wort auf (exakte) Übereinstimmung mit der Suchmaske getestet.

Assoziative Speicher werden im Daten- und Instruktions-Cache benutzt. Daneben finden sie Anwendung in Spezialrechnern („PEPE“ von Burroughs beziehungsweise „LUCAS“). Sollten die Probleme zur preisgünstigen Fertigung von großen flexiblen Assoziativspeichern einmal gelöst sein, so ist mit einer Anwendung im Datenbankbereich unbedingt zu rechnen.

1.7 Neuronale Netze

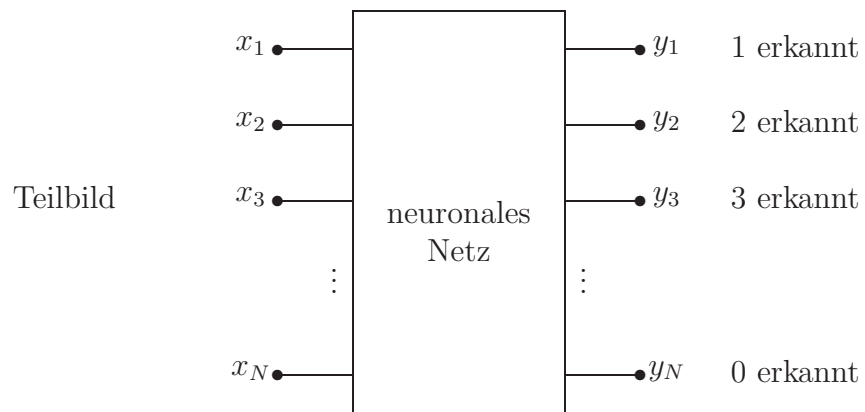
Komplexe Aufgaben sind zum Teil nicht mehr analytisch (d.h. durch Algorithmen) lösbar bzw. müssen für jede Eingabedatenkonfiguration aufwendig wieder neu gelöst werden.

Beispiel:



Handgeschriebene Ziffern
(größennormierte Darstellung mit einer 16×16 Grauwertmatrix)

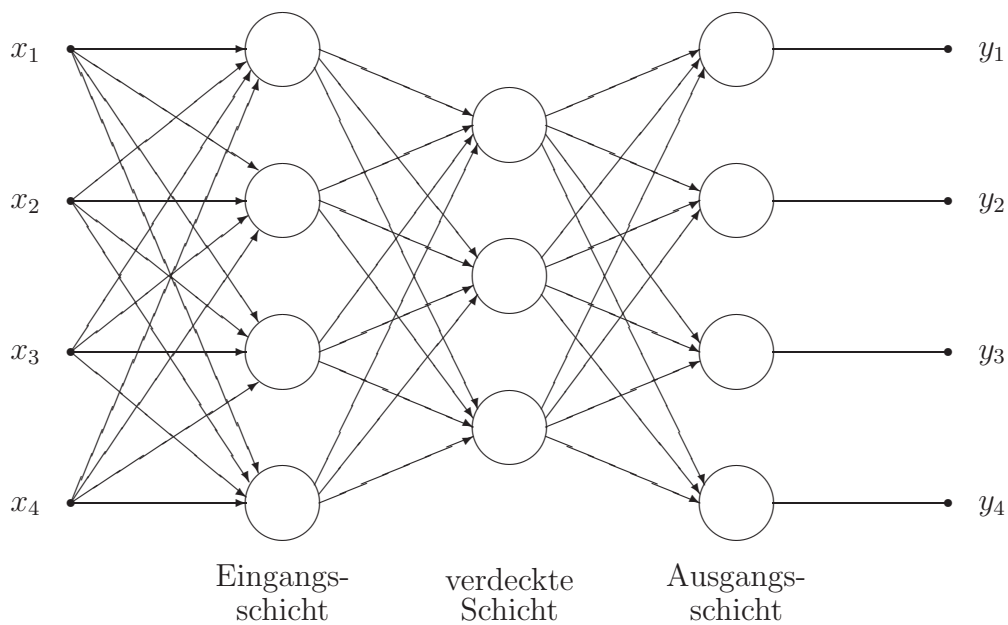
Die Erkennung handgeschriebener Ziffern (etwa für die Postleitzahlen-Erkennung) ist analytisch nicht sehr gut beschreibbar. Deshalb wird versucht, ein dem menschlichen Gehirn ähnlich komplexes, „lernendes“ Schaltwerk — ein neuronales Netz — zur Problemlösung einzusetzen.



Neuronales Netz mit Ein- und Ausgangsvektoren

Merkmale („künstlicher“) neuronaler Netze:

- *vielen* Neuronen
- feste problemspezifische Verschachtelung der Neuronen
- Verbindungen gewichtet (hemmend, fördernd)
- jedes Neuron agiert durch
 - ständige gewichtete Aufsummierung aller Eingänge und „Schwellwert“-Ausgabe auf andere Neuronen
- häufig modularer Aufbau: Schichten



Bei einigermaßen komplexen Netzen sind die benötigten Gewichte für die gewünschte Aufgabe nicht mehr berechenbar; deshalb:

Training / Lernen der Gewichte

durch „Rückkopplung“ oder „Backpropagation“.

Vorteile beim Einsatz von neuronalen Netzen:

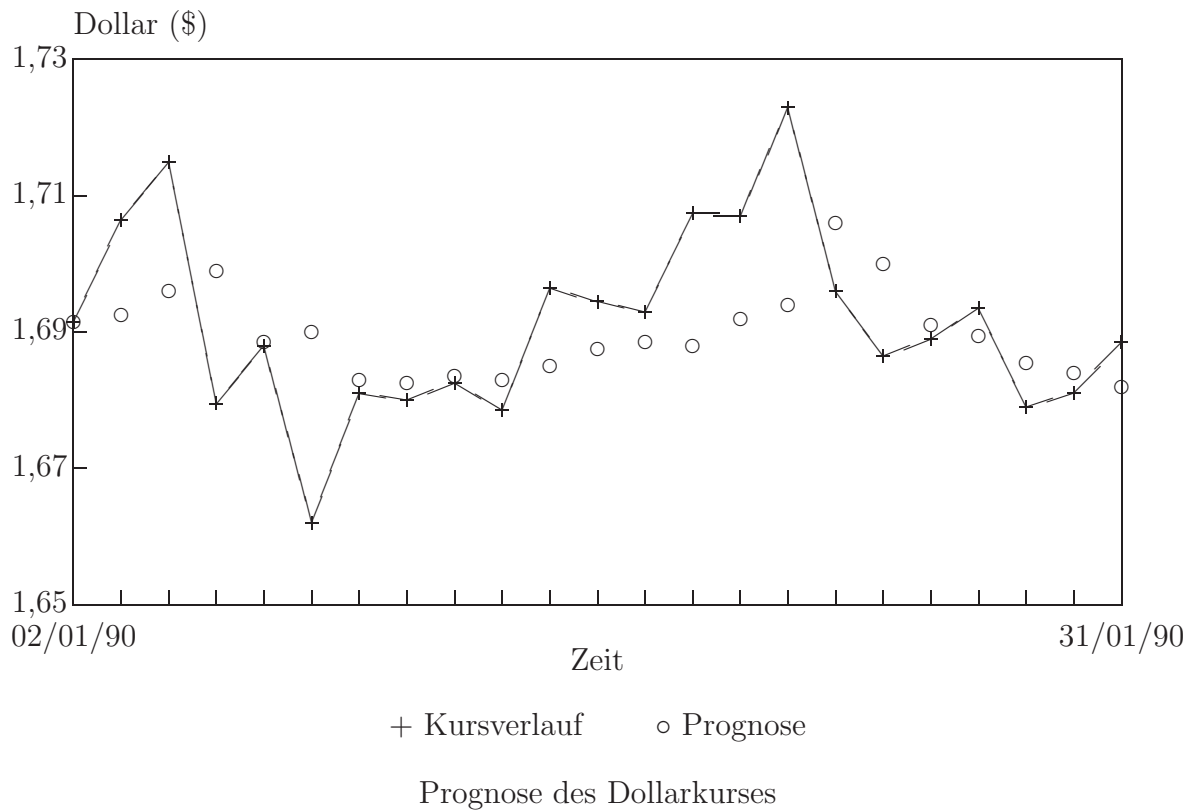
- Lernfähigkeit (jedoch lange Dauer)
- Generalisierung (erlernt auch das Verhalten für „ähnliche“ Situationen, d.h. auch für nicht gelernte Eingabesituationen werden „sinnvolle“ Ausgaben erzeugt)
- Fehlertoleranz (gegenüber dem Ausfall einzelner Neuronen oder Eingangsdaten)
- Unempfindlichkeit gegen Rauschen

Nachteile:

- Analyse der Klassifikationskriterien unmöglich
- Lernen (= Programmieren) nur durch Training möglich
- das Trainingsprogramm muß auf die zu lernende Aufgabe abgestimmt sein; das Erreichen dieser Abstimmung ist häufig nur unklar zu erkennen
- Lernen sehr zeitaufwendig (dadurch aber keine Analyse der Probleme nötig)
- sehr eingeschränkt einsetzbar, meist Klassifikation, Mustererkennung, Optimierung, Kontrollaufgaben (Robotersteuerung)
- kein universelles neuronales Netz, d.h für jede Aufgabe
 - spezielles Netz
 - spezielle Verbindungsstruktur
 - spezielle Trainingsdaten

In gewissem Rahmen verlangt der Einsatz von Expertensystemen ebenfalls keine analytische Durchdringung des zu lösenden Problems (sondern „nur“ eine Sammlung von „Wissen“ in Form von Fakten und Regeln). Im Gegensatz zu neuronalen Netzen liefert aber der Einsatz von Expertensystemen eine nachvollziehbare Begründung der Problemlösung.

Anwendungen:



- Bei Kursprognosen wurde eine Trefferquote von $\sim 56\%$ erreicht.
- Mustererkennung beim Lesen von Postleitzahlen.
- Klassifizierung von Gefahrensituationen
- ...

Vergleich: Elektronik – Biologie, künstliche — natürliche Neuronen

	Elektronik	Biologie
Clock-Frequenz	$10^8 s^{-1}$	$10^3 s^{-1}$ Frequenz der Neuronenimpulse
Speicherdichte	10^7 Bit/cm^{-3}	$10^{12} \text{ Synapsen/cm}^{-3}$
Verbindungen	10 fan-in, fan-out von digitalen Gattern	10^4 Synapsen pro Neuronen
Anzahl Elemente	10^9 Transistoren	10^{11} Neuronen, 10^{15} Synapsen
Leistungsverbrauch	1 Kilowatt	50 Watt

Eine Auswahl von elektronischen neuronalen Netzen:

		Verbindungen pro Sekunde ($\times 10^9$)	Verbindungen Auflösung	Anzahl Neuronen	Anzahl Verbin- dungen	Anzahl Gewichte	Technologie	Lernen
1	M.Griffin et al. Adaptive Solutions	1,6	9 Bit x 16 Bit fixed point	64	64	16.384	0,8 μm CMOS 26,2 mm x 27,5 mm	auf einem Ch verschiedene Algorithmen
2	M.Asai et al. Hitachi	1,5	8 Bit x 8 Bit product	576	576	36.864	0,8 μm CMOS Gate Array, Wafer 5 Inch	nicht auf dem Wafer
3	M.A.Sivilotti et al. Neural Semiconductor	2	4 Bit x 4 Bit product	32	1.024	1.024	?	nicht auf dem Chip
4	H.Graf, D.Henderson AT&T Bell Labs.	80	1 Bit x 4 Bit product flexible	256	8.192	8.192	0,9 μm CMOS 4,5mm x 7 mm	nicht auf dem Chip
5	Y.Arima et al. Mitsubishi Electric	28	1 Bit x 6 Bit product	336	28.000	28.000	1 μm CMOS 14,5 mm x 14,5 mm	auf dem Chip Boltzmann
6	R.Allen, J.Alspector and A.Jayakumar, Bellcore	0,1	5 Bit x 5 Bit product	32	992	992	1,25 μm CMOS	auf dem Chip Boltzmann
7	R.Benson et al. Intel Corp.	2	6 Bit x 6 Bit product	64	10.240	10.240	1 μm CMOS EEProm	nicht auf dem Chip

Quelle:Weber

Die ersten drei Schaltungen sind digitale Prozessoren, während die folgenden vier zumindest teilweise analoge Schaltungen sind.

1.8 ANHANG: Benchmarks

Relative Rechnersystem-Leistungen sind abhängig von

- der CPU-Leistung,
- der Systemarchitektur (Cache, Bus-Leistung, Hauptspeichergröße und -geschwindigkeit, Swapping-Bereich, Plattengröße und -geschwindigkeit, ...)
- dem Betriebssystem (Art und Version),
- dem Compiler (Sprache, Implementierung und Version, mit/ohne Optimierung, Art der Optimierung, ...)

Deshalb sind Benchmarks nur bei vollständiger Notation der Betriebsumgebung (CPU, Systemarchitektur, Anwendungsprogramm-Typ, Betriebssystem und Compiler) und auch dann nur beschränkt für eine Vorauswahl sinnvoll einsetzbar.

Besser: Test der gewünschten Anwendersoftware unter Einsatzbedingungen.

(Manche Benchmarks – etwa SPEC92 – versuchen einen dedizierten Anwendungs-Mix zu mitteln.)

Beispiele:

- $MIPS = \frac{\text{million instructions per second}}{1} = \frac{1}{(\text{cycles per average instruction}) \cdot (\text{Zykluszeit in } \mu s)}$ Beachte: RISC-Mips und CISC-Mips sind unvergleichbar, da die relative Codelänge unterschiedlich ist; die Größe von „cycles per ...“ ist maschinenabhängig.
- *MFLOPS* ähnlich für „million floating point operations per second“.
- *Whetstone* testet relative Laufzeiten für ein Programm mit 106 „Whetstone-Instructions“, die als statistisches Mittel aus 949 Anwenderprogrammen ermittelt wurden (FORTRAN-Anwendungen, Schwerpunkt: numerische Berechnung und Gleitkommaoperationen).
- *Dhrystone* testet ein Programm aus 100 Anweisungen mit 53 Zuweisungstypen, 32 Steueranweisungen, 15 Prozedur- und Funktionsaufrufen (ADA-, C-, Pascal-Anwendungen; Schwerpunkt: Systemprogrammierung).

- *LINPACK-MFLOPS* testen eine Unterprogramm-Sammlung zur „Linearen Algebra“, 100 oder 1000 Gleichungen mit 100 bzw. 1000 Unbekannten in doppelter Gleitkomma-Genauigkeit (FORTRAN-Anwendungen, Schwerpunkt: Vektorrechnen, Supercomputing, Workstations, ...)
- *OLTP* (= OnLine Transactions Processing), Schwerpunkt: System-/Datenbanksoftware, I/O, Kommunikation; TP's = transactions per second.
- *Level 1 BLAS MFLOPS* (= basic linear algebra subroutines, etwa bzgl. saxpy — entspricht $A \cdot x + y$ — , ...)
- *SPECint92* mißt das geometrische Mittel von sechs INTEGER-Applikationen (espresso, li, eqntott, compress, sc, gcc) relativ zur Leistung einer VAX 11/780.
- *SPECfp92* mißt das geometrische Mittel von vierzehn Floatingpoint-Applikationen (spice2g6, doduc, mdljdp2, wave5, tomcatv, ora, alvinn, ear, mdljsp2, swm256, su2cor, hydro2d, nasa7, fpppp) relativ zur Leistung einer VAX 11/780.
- *SPECrate int92*, *SPECrate fp92* mißt ähnlich den relativen Multiprozeß-/Multiprozessordurchsatz (vergangene Zeit, bis mehrere gleichzeitig gestartete Kopien der Benchmark-Programme zu Ende gekommen sind).
- *SDET Peak Throughput*, *KENBUS1 Peak Throughput*, *Concurrent Workload* sind relative Größen zum Vergleich des Multiprozeß-/Multiprozessordurchsatzes unter zeitlich ansteigender Last.
- *SPECmark89*, *SPECint89*, *SPECfp89*, *SPECThruput89* sind alte Varianten mit weniger Benchmaks bzw. älteren Versionen dieser Benchmark-Programme.
- *SpecCPU2000*, *SpecMail2000*, *SpecWEB99*
vergleiche: <http://www.specbench.org/>

Auszug aus dem SPEC Newsletter:

Trademark Information:

Here is a list of trademarks mentioned in this newsletter. Other trademarked items that may also appear in this newsletter are trademarks of their respective companies.

UNIX is a registered trademark of Unix Systems Laboratories.

Ethernet is a trademark of Xerox Corporation.

...

SPEC, SPECratio, SPECmark89, SPECint89, SPECfp89, SPECint92, SPECfp92, SPECrate_int92 and SPEC_fp92 are trademarks of the Standard Performance Evaluation Corporation.

...

SPEC Member Companies:

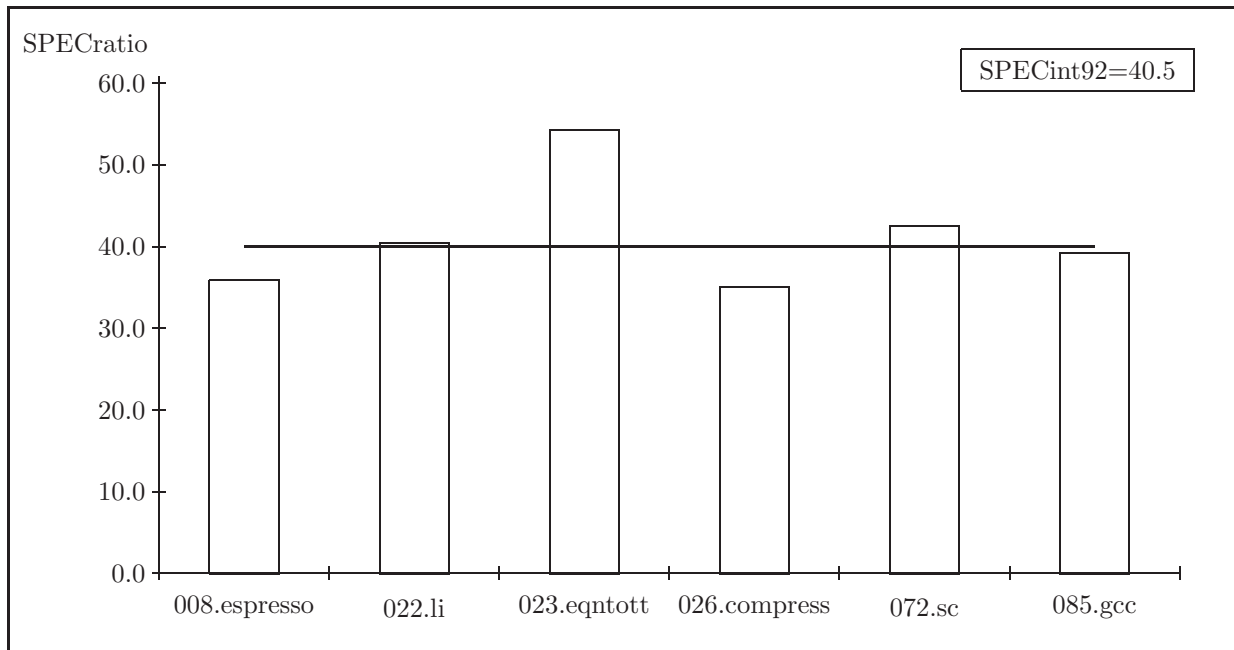
- AT&T/NCR
- Auspex Systems, Inc.
- Compaq Computer Corp.
- Control Data Systems Inc.
- Data General Corp.
- Digital Equipment Corp.
- Electronic Data Systems
- Fujitsu Microelectronics
- HaL Computer Systems Inc.
- Hewlett Packard Co.
- IBM Corp.
- Intel Corp.
- Kubota Pacific Computer Inc.
- Intergraph Corp.
- Motorola Inc.
- NeXT Computer Inc.
- Novell Inc.
- Olivetti
- Siemens Nixdorf Informationssysteme AG
- Silicon Graphics Inc.
- Solbourne Computer Inc.
- Sun Microsystems Inc.
- Unisys Corp.

SPEC Associates:

- Leibniz Rechenzentrum, Germany
- Science and Engineering Research Council (SERC), England
- National Taiwan University, Taiwan
- Center for Scientific Computing (CSC), Finland

SPEC Benchmark CINT92 Summary

RESULTS:	SPEC Reference Time (Sec)	Time (Sec)	SPEC Ratio	IBM Corporation RS/6000 POWERstation 335	
Benchmark Name					
008.espresso	2270	63.9	35.5	Hardware	
022.li	6210	151.4	41.0	Modell Number	POWERstation 355
023.eqntott	1100	20.8	52.9	CPU	41.67 MHz POWER 4132
026.compress	2770	81.1	34.2	FPU	Integrated
072.sc	4530	105.8	42.8	Number of CPU's	1
085.gcc	5460	138.7	39.4	Primary Cache	32KBI+32KBD on chip
				Secondary Cache	None
				Memory	32MB
				Disk Subsystem	1 x 1.0 GB
				Other Hardware	None
				Software	
				O/S & Version	AIX Version 3.2.3
				Compilers & Version	XL C/6000 Version 1.2.1
				Other Software	Kuck & Associates Beta KAP-C
Geometric Mean: SPECint92			40.5	File System Type	AIX
Hardware Availability:			Mar-93	System State	Multi-user
Software Availability:			Dec-92		
License #:	11	Tested:	Jan-93	By: IBM	Of: Austin, TX



Kapitel 2

Peripherie

2.1 Scanner, Strichcodeleser, OCR

2.1.1 Historische Entwicklung der Informations-Input-Codes

Historisch entwickelten sich verschiedene Informationscodierungen im Bereich der automatischen Datenverarbeitung:

- Lochkarten/-streifen
- TÜV-Formulare, . . . (Markierungen: 1 aus N)
- Strichcodes (Barcodes) auf Waren, Kassenerfassung (Markierungen):

EAN = europäische Artikel-Nummer, DIN 66236:

13-stelliger Dezimalcode			
1–2	3–7	8–12	13
Landnr.	Betriebsnr.	Artikelnr.	Prüfziffer

- Magnetcodes auf Kreditkarten, Telefonkarten, . . . (Codes):
„passiv, 3 Spuren, DIN 9785“, z.B. Kreditkarten
oder fälschungssicherer
„aktiv (Chips)“, z.B. Telefonkarten
- OCR = „optical character recognition“

$\left\{ \begin{array}{l} \text{OCR-A} \\ \text{OCR-B} \\ \text{Maschinenschrift} \end{array} \right. \quad \text{mit Drucker, TeX, \dots erzeugbar}$

- Klarschrift normiert
- Klarschrift handschriftlich (Blockschrift)
- Schreibschrift (Erkennung durch neuronale Netze, \dots)

2.1.2 Strichcodeleser

Strichcodeleser (z.B. in Bibliotheken zur Signatur-Erfassung genutzt) führen die optische Abtastung durch mechanische Führung (Bewegung durch eine Bedienungsperson) über die Vorlage aus. Dazu ist eine angemessene gleichmäßige Bewegung notwendig!

Es gibt (PC-)Tastaturen mit integriertem Anschluß für Strichcode-Lesestifte, Laser-Scanner-Pistolen, Schlitz-Leser und Durchzugs-Leser. Diese liefern die entsprechenden Zeichencodes an den Personal-Computer, als ob sie per Tastatur eingegeben worden wären (es ist also keine spezielle Eingabegeräte-Schnittstelle der benutzten Software nötig).

Typische Einsatzgebiete sind:

1. Produktidentifizierung bei Warenein- und -ausgang
2. Datenerfassung im Produktionsablauf
3. Betriebsdaten-Erfassung
4. Inventur-Eingabe
5. Kassenterminal im Handel
6. Arzneimittelverwaltung
7. ...

Bei der Auswahl von Strichcodeleser-Hardware ist insbesondere auf eine automatische Erkennung der gängigen Strichcodes zu achten: Code 3 of 9, Code 3 of 9 Extended, Code 128, Interleaved 2 of 5, Industrial 2 of 5, Codebar, Code 93, UPC A, UPC E, EAN/JAN 8, EAN/JAN 13, Plessey, ASCII, ...

2.1.3 Scanner

Bei Scannern geschieht die optische Abtastung

- entweder wie beim Strichcodeleser oder
- ganzseitig wie beim Kopierer (d.h. keine Bewegung nötig).

Als Ergebnis ist

- Pixelgraphik oder
- mit Hilfe von OCR-Software auch Schrift (ASCII, Textverarbeitungsformate mit Fettschrift, verschiedenen Schriftgrößen, verschiedenen Schriftarten, . . .)

möglich. Alte OCR-Systeme sind nur geeignet für

- Normschrift,
- feste Lage (Drehproblem) der Vorlage.

Neuere OCR-Systeme

- erkennen Texte auch lage- und verzerrungsunabhängig durch Nutzung geometrischer Invarianten statt durch bloßes Pixel-Matching,
- sind lernfähig (d.h. Hinzulernen besonderer Zeichen oder Symbole: Logos, griechische Buchstaben, mathematische Symbole, „verschmierte“ Zeichen, . . .).

Fax-Geräte können als Scanner „geringer“ Auflösung genutzt werden. Kopierer mit Computerausgang können als Scanner genutzt werden, solche mit Computereingang als Drucker.

0123456789
 JYH|
 ABCDEFGHIJKLM
 NOPQRSTUVWXYZ
 . : ; = + - - / * ■ —

1234567890
 ABCDEFGHIJKL.
 NOPQRSTUVWXYZ
 * + - = / . , < > |

Schriften OCR-A und OCR-B



Rasierwasser
(Deutschland)



Watte
(Österreich)



Tabak
(Schweden)

Strichcodes

OCR Prüfstand

Erkennungsgüte und Erkennungszeiten						
Differenz der Wortzahlen						
	c't-Text		Listing		Stern-Text	
Gesamtzahl	670		631		671	
	absolut	relativ	absolut	relativ	absolut	relativ
Optopus*	46	6,87%	136	21,55%	96	14,31%
Omnipage	47	7,01%	40	6,34%	38	5,66%
WordScan	5	0,75%	45	7,13%	35	5,22%
Perceive	48	7,16%	631	100%	53	7,90%
Catchword	21	3,13%	294	46,59%	117	17,44%
GO-CR	50	7,46%	169	26,78%	52	7,75%
Recognita	5	0,75%	371	58,80%	38	5,66%
Readiris	48	7,16%	411	65,13%	75	11,18%
Differenz verschiedener Wörter						
	c't-Text		Listing		Stern-Text	
Gesamtzahl	457		229		418	
	absolut	relativ	absolut ¹	relativ ¹	absolut	relativ
Optopus*	9	1,97%	83	36,24%	74	17,70%
Omnipage	65	14,22%	96	41,92%	50	11,96%
WordScan	31	6,78%	113	49,34%	61	14,59%
Perceive	93	20,35%	229	100%	77	18,42%
Catchword	23	5,03%	237	103,5%	85	20,33%
GO-CR	52	11,38%	237	103,5%	88	21,05%
Recognita	36	7,88%	289	126,2%	70	16,75%
Readiris	53	11,60%	27	11,79%	75	17,94%
* Höhere Werte als die Gesamtzahl ergeben sich durch zerrissene Wörter.						

Die *Differenz der Wortzahlen* ergibt sich als Differenz der Gesamtzahl der Worte in der Vorlage und Gesamtzahl der Worte im gescannten Text (Ursache der Differenz: Zeichenfehler werden i.a. durch

Benutzung eines Wörterbuches „berichtigt“; Leerräume in Proportionalschrift können fälschlich als Wortende interpretiert werden, ...). Beim heutigen Stand der Technik kann man davon ausgehen, durchschnittlich 10% der gescannten Worte nachbearbeiten zu müssen.

Zeichensätze:

ASCII (7 Bit):

00	NUL	01	SOH	02	STX	03	ETX	04	EOT	05	ENQ	06	ACK	07	BEL
08	BS	09	HT	0A	NL	0B	VT	0C	NP	0D	CR	0E	SO	0F	SI
10	DLE	11	DC1	12	DC2	13	DC3	14	DC4	15	NAK	16	SYN	17	ETB
18	CAN	19	EM	1A	SUB	1B	ESC	1C	FS	1D	GS	1E	RS	1F	US
20	SP	21	!	22	"	23	#	24	\$	25	%	26	&	27	'
28	(29)	2A	*	2B	+	2C	,	2D	-	2E	.	2F	/
30	0	31	1	32	2	33	3	34	4	35	5	36	6	37	7
38	8	39	9	3A	:	3B	;	3C	<	3D	=	3E	>	3F	?
40	@	41	A	42	B	43	C	44	D	45	E	46	F	47	G
48	H	49	I	4A	J	4B	K	4C	L	4D	M	4E	N	4F	O
50	P	51	Q	52	R	53	S	54	T	55	U	56	V	57	W
58	X	59	Y	5A	Z	5B	[5C	\	5D]	5E	^	5F	_
60	`	61	a	62	b	63	c	64	d	65	e	66	f	67	g
68	h	69	i	6A	j	6B	k	6C	l	6D	m	6E	n	6F	o
70	p	71	q	72	r	73	s	74	t	75	u	76	v	77	w
78	x	79	y	7A	z	7B	{	7C		7D	}	7E	~	7F	DEL

Landesspezifische Varianten durch Zeichenersatz (ISO-Austauschtabelle):

ISO Nr.	Zeichensatz	Dezimalform												
		ID	35	36	64	91	92	93	94	96	123	124	125	126
6	ANSI ASCII	0U	#	\$	@	[\]	^	`	{		}	~
11	Schweden: Namen	0S	#		É	Ä	Ö	Å	Ü	`	ä	ö	å	ü
10	Schweden	3S	#		@	Ä	Ö	Å	^	`	ä	ö	å	
17	Spanien	2S	£	\$	§	ı	Ñ	ı	^	`	°	ñ	ç	~
69	Frankreich	1F	£	\$	à	°	ç	§	^	μ	é	ù	è	
21	Deutschland	1G	#	\$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
4	Großbritannien	1E	£	\$	@	[\]	^	`	{		}	
16	Portugal	4S	#	\$	§	Ã	Ç	Õ	^	`	ã	ç	õ	°
60	Norwegen 1	0D	#	\$	@	Æ	Ø	Å	^	`	æ	ø	å	
61	Norwegen 2	1D	§	\$	@	Æ	Ø	Å	^	`	æ	ø	å	
2	IRV		#		@	[\]	^	`	{		}	
15	Italien	0I	£	\$	§	°	ç	é	^	`	à	ò	è	ì

(„Industriestandard“) PC-8 Zeichensatz:

0	16	32	0	@	P	‘	p	Ç	É	á				α	≡
			48	64	80	96	112	128	144	160	176	192	208	224	240
1	17	33	!	1	A	Q	a	q	ü	æ	í			β	±
			49	65	81	97	113	129	145	161	177	193	209	225	241
2	↑ ↓	”	2	B	R	b	r	é	Æ	ó		⊥		Γ	≥
			50	66	81	98	114	130	146	162	178	194	210	226	242
3	!!	#	3	C	S	c	s	â	ô	ú		⊢		π	≤
			51	67	82	99	115	131	147	163	179	195	211	227	243
4	¶	\$	4	D	T	d	t	ä	ö	ñ	⊣			Σ	
			52	68	83	100	116	132	148	164	180	196	212	228	244
5	♣	§	%	5	E	U	e	u	à	ò	Ñ			σ	
			53	69	84	101	117	133	149	165	181	197	213	229	245
6	♠		&	6	F	V	f	v	â	û	ª			μ	÷
			54	70	85	102	118	134	150	166	182	198	214	230	246
7		’	7	G	W	g	w	ç	ù	º				τ	≈
			55	71	86	103	119	135	151	167	183	199	215	231	247
8	↑	(8	H	X	h	x	ê	ÿ	¿				Φ	°
			56	72	87	104	120	136	152	168	184	200	216	232	248
9	○	↓)	9	I	Y	i	y	ë	Ö				Θ	·
			57	73	88	105	121	137	153	169	185	201	217	233	249
10	→	*	:	J	Z	j	z	è	Û	¬				Ω	·
			58	74	89	106	122	138	154	170	186	202	218	234	250
11	←	+	;	K	[k	{	ï	ç					δ	√
			59	75	90	107	123	139	155	171	187	203	219	235	251
12		,	<	L	\	l		î	£					∞	ⁿ
			60	76	91	108	124	140	156	172	188	204	220	236	252
13	↔	-	=	M]	m	}	ì		ì				φ	²
			61	77	92	109	125	141	157	173	189	205	221	237	253
14		.	>	N	^	n	~	Ä	Pt	«				€	
			62	78	93	110	126	142	158	174	190	206	222	238	254
15		/	?	O	-	o		Å	f	»				∩	
			63	79	94	111	127	143	159	175	191	207	223	239	255

Windows 3.0 Zeichensatz:

NUL			0	@	P	`	p				°	À		à	
0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
		!	1	A	Q	a	q		'	i	±	Á	Ñ	á	ñ
1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
		”	2	B	R	b	r		'	ç	²	Â	Ò	â	ò
2	18	34	50	66	81	98	114	130	146	162	178	194	210	226	242
		#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
3	19	35	51	67	82	99	115	131	147	163	179	195	211	227	243
		\$	4	D	T	d	t				'	Ä	Ô	ä	ô
4	20	36	52	68	83	100	116	132	148	164	180	196	212	228	244
		%	5	E	U	e	u				μ	Å	Õ	å	õ
5	21	37	53	69	84	101	117	133	149	165	181	197	213	229	245
		&	6	F	V	f	v				¶	Æ	Ö	æ	ö
6	22	38	54	70	85	102	118	134	150	166	182	198	214	230	246
BEL		'	7	G	W	g	w			§	·	Ç	×	ç	÷
7	23	39	55	71	86	103	119	135	151	167	183	199	215	231	247
BS		(8	H	X	h	x			¨		È	Ø	è	ø
8	24	40	56	72	87	104	120	136	152	168	184	200	216	232	248
HT)	9	I	Y	i	y			©	¹	É	Û	é	ù
9	25	41	57	73	88	105	121	137	153	169	185	201	217	233	249
LF		*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
10	26	42	58	74	89	106	122	138	154	170	186	202	218	234	250
VT	ESC	+	;	K	[k	{			«	»	Ë	Û	ë	û
11	27	43	59	75	90	107	123	139	155	171	187	203	219	235	251
FF		,	<	L	\	l				¬	¼	Ì	Ü	ì	ü
12	28	44	60	76	91	108	124	140	156	172	188	204	220	236	252
CR		-	=	M]	m	}			-	½	Í	Ý	í	ý
13	29	45	61	77	92	109	125	141	157	173	189	205	221	237	253
SO		.	>	N	^	n	~				¾	Î		î	
14	30	46	62	78	93	110	126	142	158	174	190	206	222	238	254
SI		/	?	O	_	o				-	¿	Ï	ß	ï	ÿ
15	31	47	63	79	94	111	127	143	159	175	191	207	223	239	255

ISO-8859 Latin 1 (ECMA-94 Latin 1) Zeichensatz:

0	16	32	48	0	@	P	'	p			°	À		à		
1	17	33	49	1	A	Q	a	q		i	±	Á	Ñ	á	ñ	
2	18	34	50	2	B	R	b	r			²	Â	Ò	â	ò	
3	19	35	51	3	C	S	c	s		£	³	Ã	Ó	ã	ó	
4	20	36	52	4	D	T	d	t			'	Ä	Ô	ä	ô	
5	21	37	53	5	E	U	e	u			μ	Å	Õ	å	õ	
6	22	38	54	6	F	V	f	v			¶	Æ	Ö	æ	ö	
7	23	39	55	7	G	W	g	w		§	·	Ç	×	ç	÷	
8	24	40	56	8	H	X	h	x		¨		È	Ø	è	ø	
9	25	41	57	9	I	Y	i	y		©	¹	É	Ù	é	ù	
10	26	42	58	*	:	J	Z	j	z		ª	º	Ê	Ú	ê	ú
11	27	43	59	+	;	K	[k	{		«	»	Ë	Û	ë	û
12	28	44	60	,	<	L	\	l			¬	¼	Ì	Ü	ì	ü
13	29	45	61	-	=	M]	m	}		-	½	Í	Ý	í	ý
14	30	46	62	.	>	N	^	n	~			¾	Î		î	
15	31	47	63	/	?	O	-	o			¯	Ï	ß	ï	ÿ	

Eingabe von Sonderzeichen / Tastaturen:

Alt- beziehungsweise Compose-Tastensequenzen:

Zeichen	Compose-Tastensequenz
Á	A ’
á	a ’
À	A `
à	a `
Â	A ^
â	a ^
Ã	A ~
ã	a ~
”A	A ”
”a	a ”
Ã	A *
ã	a *
Æ	A E
æ	a e
Ç	C ,
ç	c ,
	d -
	D -
È	E `
è	e `

⋮

Zeichen	Compose-Tastensequenz
Í	I ’
í	i ’
Î	I ^
î	i ^
”I	I ”
”i	i ”
Ñ	N ~
ñ	n ~
Ò	O `
ò	o `
Ó	O ’
ó	o ’
Ô	O ^
ô	o ^
Õ	O ~
õ	o ~
”O	O ”
”o	o ”
Ø	O /
ø	o /

Auf PC's existiert daneben noch die Möglichkeit durch gleichzeitiges Drücken der Modifikationstasten **AltGraph**, **Shift**, **AltGraph+Shift** eine Vierfachbelegung einiger

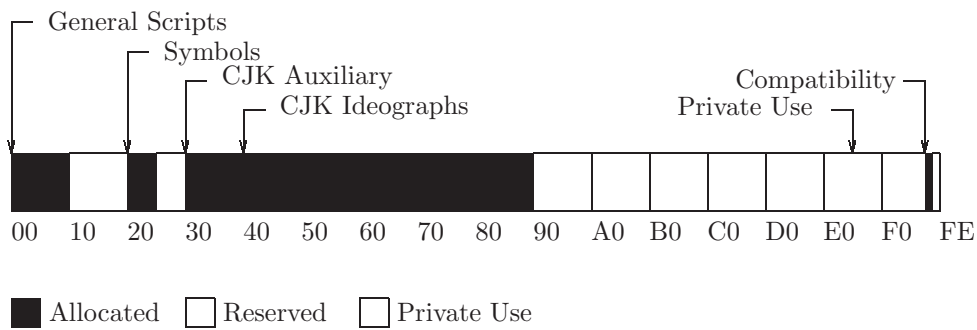
Tasten zu realisieren, etwa **AltGraph+Q** für @:

Q
@

 Damit immer noch nicht erreichbare Zeichen können durch ihre Dezimalcodes unter Benutzung der Alt-Taste erzeugt werden: **<Alt><2><3><6>** (die Alt-Taste muß dabei während der Zifferneingabe niedergedrückt bleiben und die Ziffern müssen im Zifferneingabeblock bei eingeschaltetem Num-Modus betätigt werden).

Unicode:

Eine weltweite Anwendbarkeit kann erreicht werden, falls 16Bit für die Codierung zur Verfügung stehen: *UNICODE*, der in neuen Programmiersprachen wie etwa *JAVA* schon benutzt wird.



CJK = Chinese, Japanese, Korean

Abbildung 2.1: Verteilung UNICODE

Statistik der Version 1.0 Die folgende Tabelle zeigt die Anteile des gesamten im UNICODE zur Verfügung stehenden Platzes, die verschiedenen Schrifttypen in der Version 1.0 bereits zugeteilt wurden:

	<i>Allocated</i>	<i>Unassigned</i>	<i>% Assigend</i>	
General	2336	5856	29%	
Symbols	1290	2806	31%	
CJK symbols	763	261	75%	
Hangul	2350	450	84%	
Han Compatibility	268	4	99%	(Volume 2)
Ideographic & other	20733	22275	48%	(Volume 2)
User Space	5632	N/A	N/A	
Compatibility Zone	362	133	73%	
Special	1	13		
FEFF	1	0		
FFFE, FFFF	N/A	2		
<i>Totals</i>	28706 (assigned)			
	+ 5632 (private use)			
	= 34338 (allocated) 52%			

Mit noch über 30000 unbenutzten Character Positionen besitzt der UNICODE auch für die Zukunft noch genug Raum für weitere Expansionen.

Die Zukunft des UNICODES

In Zukunft wird der UNICOD Standard um weniger verbreitete und veraltete Schrifttypen erweitert. Schrifttypen dieser Art werden jedoch nicht in ihrer ursprünglichen Form eingebunden, da sich ihr Nutzen schwer einschätzen läßt. So wird bei vielen dieser Schriften eine ausführliche Diskussion nötig sein, bis ein zufriedenstellendes Codierungsschema vorliegt. Die fünf Schriftarten *Ethiopian*, *Burmese*, *Khmer*, *Sinhala* und *Mongolian* werden zum Standard UNICOD hinzugefügt, sobald zuverlässige Informationen über sie vorliegen. Weitere Schriftarten, die für eine mögliche Aufnahme vorgesehen sind, sind

- *Inuktitut/Cree Syllabary*: Das Kommunikationsministerium von Kanada untersucht Standardisierungen von verschiedenen Dialektarten, die von Cree und/oder Inuktitut gesprochen werden und sucht Codierungsschemen.
- *Egyptian Hieroglyphics*: Ein einheitliches Codierungsschema existiert und wird vorangetrieben.
- *Korean Hangul Syllables*: Eventuell werden noch weitere Korean Hangul Dialekte hinzugefügt.

Der Unterschied zwischen der logischen Anordnung von Zeichen und der Anordnung auf dem Bildschirm zeigt die Abbildung 2.2

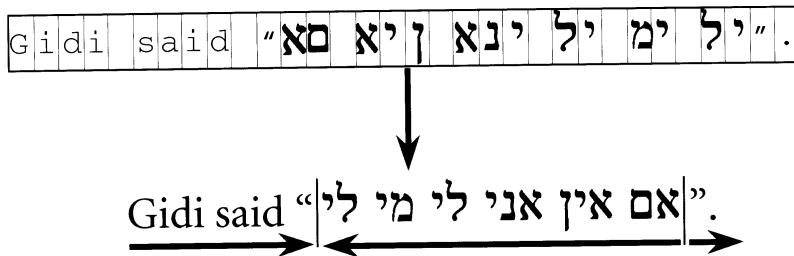


Abbildung 2.2: Bidirectional Ordering

General Scripts

Im *General Scripts*-Bereich des UNICODEs sind alle lateinischen und nicht-ideographischen Schriftzeichen codiert:

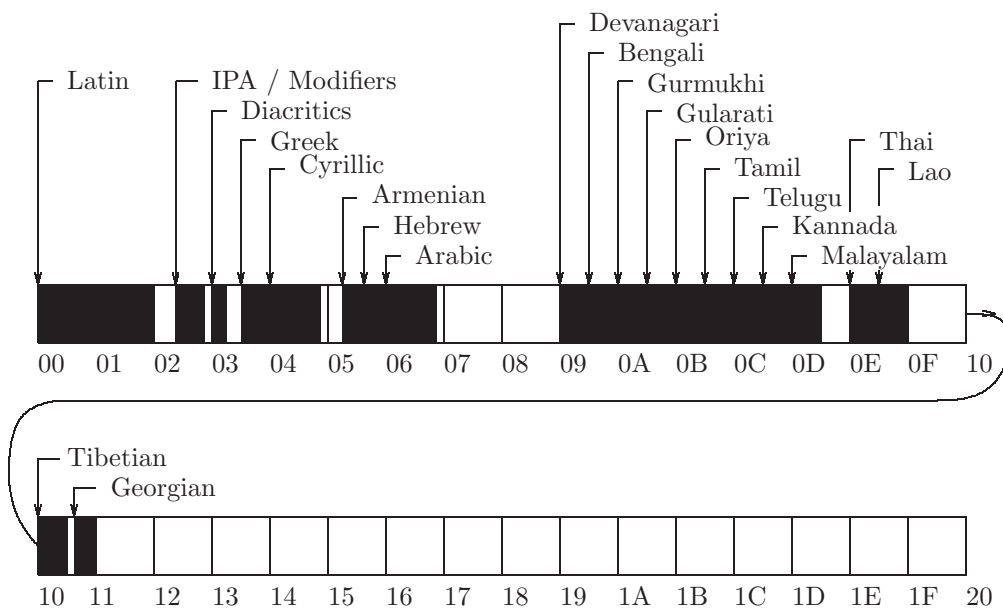


Abbildung 2.3: General Scripts

Control		ASCII						Control		Latin 1					
000	001	002	003	004	005	006	007	008	009	00A	00B	00C	00D	00E	00F
NUL	DLE		0	@	P	'	p	CTRL	CTRL		°	À		à	
SOH	DC1	!	1	A	Q	a	q	CTRL	CTRL	ı	±	Á	Ñ	á	ñ
STX	DC2	”	2	B	R	b	r	CTRL	CTRL		²	Â	Ò	â	ò
ETX	DC3	#	3	C	S	c	s	CTRL	CTRL	£	³	Ã	Ó	ã	ó
EOT	DC4	\$	4	D	T	d	t	CTRL	CTRL		,	Ä	Ô	ä	ô
ENQ	NAK	%	5	E	U	e	u	CTRL	CTRL		μ	Å	Õ	å	õ
ACK	SYN	&	6	F	V	f	v	CTRL	CTRL		¶	Æ	Ö	æ	ö
BEL	ETB	'	7	G	W	g	w	CTRL	CTRL	§	·	Ç	×	ç	÷
BS	CAN	(8	H	X	h	x	CTRL	CTRL	¨	˘	È	Ø	è	ø
HT	EM)	9	I	Y	i	y	CTRL	CTRL	©	ı	É	Ù	é	ù
LF	SUB	*	:	J	Z	j	z	CTRL	CTRL	ª	º	Ê	Ú	ê	ú
VT	ESC	+	;	K	[k	{	CTRL	CTRL	«	»	Ë	Û	ë	û
FF	FS	,	<	L	\	l		CTRL	CTRL	¬	¼	Ì	Ü	ì	ü
CR	GS	-	=	M]	m	}	CTRL	CTRL	-	½	Í	Ý	í	ý
SO	RS	.	>	N	^	n	~	CTRL	CTRL	-	¾	Î		î	
SI	US	/	?	O	_	o	DEL	CTRL	CTRL	-	¿	Ï	ß	ï	ÿ

Tabelle 2.1: UNICODE Version 1.0, Character Blocks 0000-00FF

2000..206F	Zeichen für die Zeichensetzung: ,,:;“„ ...
2070..209F	Subscripts und Superscripts: ^{2,3,4} , ...
20A0..20CF	Währungssymbole: £,\$, ...
20D0..20FF	diakretische Zeichen: ←, →, ...
2100..214F	buchstabenähnliche Zeichen: ℱ, °F, ...
2150..218F	Zahlen: $\frac{1}{3}$, I, VII, ...
2190..21FF	Pfeile: ↑, ↗, ...
2200..22FF	mathematische Sonderzeichen: ∀, ∃, ∈, ...
2300..23FF	verschiedene technische Sonderzeichen: #, ...
2400..243F	Symbole für Control-Zeichen: NUL, ESC, ...
2440..245F	OCR-Zeichen
2460..24FF	eingerahmte alphanumerische Zeichen: ©, ...
2500..257F	Formular- und Diagrammzeichen: †, ‡, , ...
2580..259F	Blockgraphik-Zeichen
25A0..25FF	graphische Symbole
2600..26FF	verschiedene Dingbats
2700..27BF	Zapf-Dingbats
...	
3000..303F	CJK-Symbole
3040..309F	Hiragana
30A0..30FF	Katakana
...	

Tabelle 2.2: Weitere Zeichenbereiche

Der UNICODE stellt weitgehende Kompatibilität zu bestehenden Codes durch (verschobenes) Einfügen oder Bereitstellen von Code-Umwandlungstabellen her: 0000..007F entspricht ASCII. Für andere Codes werden UNICODE-Übersetzungstabellen bereitgestellt, z.B. für UNICODE zu SGML (Tabelle 2.4), UNICODE zu Postscript (Tabelle 2.3) oder UNICODE zu MacIntosh (Tabelle 2.5). Analoge Tabellen gibt es zur Übersetzung von UNICODE zu Microsoft Windows, zu PC Code Page Mappings (Latin, Greek,...), zu EBCDIC Code Page Mappings und weiteren.

UNIC	ISO Latin1		ZapfDB	Adobe glyph name	Unicode character name
	StdEnc		Symbol		
0020	20	20	20	space	SPACE
0021	21	21	21	exclam	ECLAMATION MARK
0022	22	22		quotedbl	QUOTATION MARK
0023	23	23	23	numbersign	NUMBERSIGN
0024	24	24		dollar	DOLLAR SIGN
0025	25	25	25	percent	PERCENT SIGN
0026	26	26	26	ampersand	AMPERSAND
0027	A9			quotesingle	APOSTROPHE-QUOTE
0028	28	28	28	parenleft	OPENING PARENTHESIS
0029	29	29	29	parenright	CLOSING PARENTHESIS
002A	2A	2A		asterisk	ASTERISK
002B	2B	2B	2B	plus	PLUS SIGN
002C	2C	2C	2C	comma	COMMA
002D	2D	AD		hyphen	HYPHEN-MINUS
002D		2D		minus	HYPHEN-MINUS
002E	2E	2E	2E	period	PERIOD
002F	2F	2F	2F	slash	SLASH
0030	30	30	30	zero	DIGIT ZERO
0031	31	31	31	one	DIGIT ONE
0032	32	32	32	two	DIGIT TWO
0033	33	33	33	three	DIGIT THREE
0034	34	34	34	four	DIGIT FOUR
0035	35	35	35	five	DIGIT FIVE
0036	36	36	36	six	DIGIT SIX
0037	37	37	37	seven	DIGIT SEVEN
0038	38	38	38	eight	DIGIT EIGHT
0039	39	39	39	nine	DIGIT NINE
003A	3A	3A	3A	colon	COLON
003B	3B	3B	3B	semicolon	SEMIKOLON

Tabelle 2.3: UNICODE to Adobe Standard Mappings

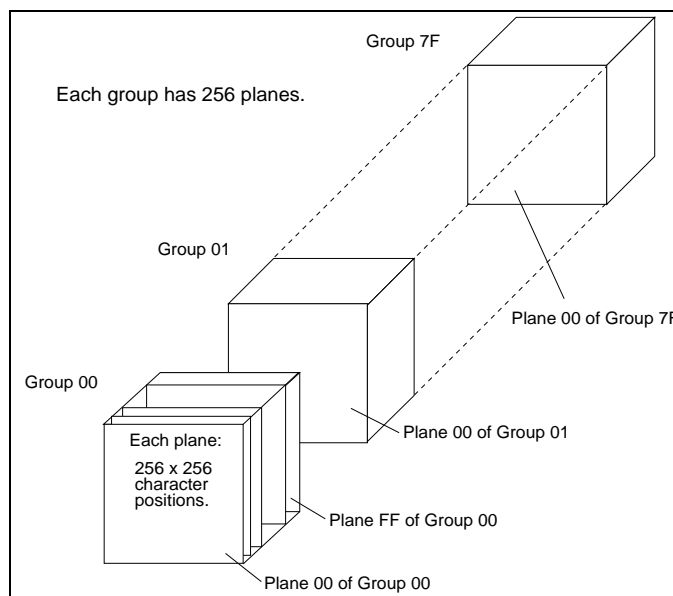
UNIC	6862.2	SGML	Unicode character name
0021		excl	EXCLAMATION MARK
0023		num	NUMBER SIGN
0024		dollar	DOLLAR SIGN
0025		percent	PERCENT SIGN
0026		amp	AMPERSAND
0027		quot	APOSTROPHE-QUOTE
0028		lpar	OPENING PARENTHESIS
0029		rpar	CLOSING PARENTHESIS
002A		ast	ASTERISK
002B	05.00	plus	PLUS SIGN
002C		comma	COMMA
002D		hyphen	HYPHEN-MINUS
002E		period	PERIOD
002F		sol	SLASH
003A		colon	COLON
003B		semi	SEMICOLON
003C		lt	LESS-THAN SIGN
003D		equals	EQUALS SIGN
003E		gt	GREATER-THAN SIGN
003F		quest	QUESTION MARK
0040		commat	COMMERCIAL AT
005B		lsqb	OPENING SQUARE BRACKET
005C		bsol	BACKSLASH
005D		rsqb	CLOSING SQUARE BRACKET
005E		circ	SPACING CIRCUMFLEX
005F		lowbar	SPACING UNDERSCORE
0060		grave	SPACING GRAVE
007B		lcub	OPENING CURLY BRACKET
007C		verbar	VERTICAL BAR
007D		rcub	CLOSING CURLY BAR
007E		tilde	TILDE
00A0		nbsp	NON-BREAKING SPACE
00A1		iexcl	INVERTED EXCLAMATION MARK
00A2		cent	CENT SIGN
00A3		pound	POUND SIGN

Tabelle 2.4: The UNICODE to SGML (ISO DIS 6862.2) Mappings

UNIC	ROM	SYM	GRK	GK2	HEB	ARB	NAME
0020	20	20	20	20	20/A0	21/A1	SPACE
0021	21	21	21	21	21/A1	21/A1	ECLAMATION MARK
0022	22		22	22	22/A2	22/A2	QUOTATION MARK
0023	23	23	23	23	23/A3	23/A3	NUMBERSIGN
0024	24		24	24	24/A4	24/A4	DOLLAR SIGN
0025	25	25	25	25	25/A5	25	PERCENT SIGN
0026	26	26	26	26	26	26/A6	AMPERSAND
0027	27		27	27	27/A7	27/A7	APOSTROPHE-QUOTE
0028	28	28	28	28	28/A8	28/A8	OPENING PARENTHESIS
0029	29	29	29	29	29/A9	29/A9	CLOSING PARENTHESIS
002A	2A		2A	2A	2A/AA	2A/AA	ASTERISK
002B	2B	2B	2B	2B	2B/AB	2B/AB	PLUS SIGN
002C	2C	2C	2C	2C	2C/AC	2C	COMMA
002D	2D		2D	2D	2D/AD	2D/AD	HYPHEN-MINUS
002E	2E	2E	2E	2E	2E/AE	2E/AE	PERIOD
002F	2F	2F	2F	2F	2F/AF	2F/AF	SLASH
0030	30	30	30	30	30/B0	30	DIGIT ZERO
0031	31	31	31	31	31/B1	31	DIGIT ONE
0032	32	32	32	32	32/B2	32	DIGIT TWO
0033	33	33	33	33	33/B3	33	DIGIT THREE
0034	34	34	34	34	34/B4	34	DIGIT FOUR
0035	35	35	35	35	35/B5	35	DIGIT FIVE
0036	36	36	36	36	36/B6	36	DIGIT SIX
0037	37	37	37	37	37/B7	37	DIGIT SEVEN
0038	38	38	38	38	38/B8	38	DIGIT EIGHT
0039	39	39	39	39	39/B9	39	DIGIT NINE
003A	3A	3A	3A	3A	3A/BA	3A/BA	COLON
003B	3B	3B	3B	3B	3B/BB	3B	SEMICOLON
003C	3C	3C	3C	3C	3C/BC	3C/BC	LESS-THAN SIGN
003D	3D	3D	3D	3D	3D/BD	3D/BD	EQUALS-SIGN
003E	3E	3E	3E	3E	3E/BE	3E/BE	GREATER-THAN SIGN
003F	3F	3F	3F	3F	3F/BF	3F	QUESTION MARK
0040	40		40	40	40	40	COMMERCIAL AT
0041	41		41	41	41	41	LATIN CAPITAL LETTER A
0042	42		42	42	42	42	LATIN CAPITAL LETTER B
0043	43		43	43	43	43	LATIN CAPITAL LETTER C
0044	44		44	44	44	44	LATIN CAPITAL LETTER D
0045	45		45	45	45	45	LATIN CAPITAL LETTER E
0046	46		46	46	46	46	LATIN CAPITAL LETTER F
0047	47		47	47	47	47	LATIN CAPITAL LETTER G
	:						

Tabelle 2.5: UNICODE to Macintosh Mappings

Die Notwendigkeit, Control-Codes anderer Codierungen auch verfügbar zu haben, und die Weigerung von Japan und Korea, die vereinheitlichte CJK-Codierung (die mit 19000 statt insgesamt über 31000 Codepositionen ausgekommen wäre) zu akzeptieren, führt zum 32Bit Zeichencode *ISO 10646*, der in seinen ersten 65536 Codes die Zeichen des Unicodes benutzt:



Näheres zum ISO- bzw. Unicode lese man bei

<http://dns.hti.umich.edu/htistaff/pubs/1997/janete.01/>,

<http://www.indigo.ie/egt/standards/iso10646/bmp-today-table.html>,

und

<http://www.unicode.org/>

nach. Die nun zur Verfügung stehenden 4294967296 Codes dürften voraussichtlich für eine Codierung auch der ausgefallensten (ausgestorbenen) Schriften ausreichen.

2.2 Spracheingabe

Probleme:

1 Sprache akzeptieren/erkennen:

- 1a) Sprache vorliegend als Schrift: zur Qualität siehe OCR in Abschnitt 2.1
- 1b) Sprache als Lautfolge: zur Umwandlung in Schrift ist ein gewisses Sprachverständnis nötig

2 Sprache verarbeiten:

- 2a) Sprache einlesen: vergleiche 1)
- 2b) Sprache übersetzen (MT = machine translation), verstehen, bearbeiten, . . .

Aktueller Stand:

Durch Spracherkennung erfaßte Texte

- sind als Rohtext zur Nachbearbeitung brauchbar,
- sind in trainierten Spracheingabesystemen als „Strichcodelese-Ersatz“ nutzbar (staubige Umgebungen . . .).

Zum Grad der allgemeinen Nutzbarkeit vergleiche man den folgenden Artikel:

Westdeutsche Zeitung vom 10.03.1993:

Sprachgenie unter Telefoncomputern

Tokio (cid). Ein wahres Sprachgenie unter den verschiedenen Telefoncomputern, die die Wünsche eines Anrufers erkennen sollen, hat die Mitsubishi Electronic Corporation in Tokio angekündigt. Ihr neuronales Telefonsystem soll auf zehn verschiedenen Identifikationsstufen jeweils 1000 gesprochene Worte erkennen können. Dies seien rund fünfzig mal mehr als alle bisherigen Prototypen leisten. Wie die Deutsche Industrie- und Handelskammer in Japan dazu mitteilt, soll die letzte Testphase gezeigt haben, daß die neue Anlage mit einer Trefferquote von 95 Prozent jede anrufende Stimme richtig „versteh“ und damit alle bekannten Systeme weit in den Schatten stellt. Damit rücke die Markteinführung des Computers, der auch für die automatische Reservierung von Hotelzimmern im 24-Stunden-Betrieb konzipiert worden sei, in greifbare Nähe.

Anwendungen: Voicemail, Voicefax, ...

Prognose beziffert den weltweiten Umsatz 1994 auf 3,3 Milliarden Dollar

Verlagerung von Voice-Mail auf PCs belebt das Geschäft

FREIBURG (CW) – Die Total Communication Gesellschaft für Kommunikationstechnologie hat eine Prognose über den weltweiten Markt von Voice-Mail-Systemen erstellt. Ausgehend von Recherchen in internationalen Informationsquellen, beziffert das Freiburger Unternehmen das Wachstum für 1994 auf 25 Prozent. Erstmals überschreitet das Volumen dabei in diesem Jahr die Drei-Milliarden-Dollar-Marke.

Total Communication gliedert den Voice-Mail-Markt, in dem 1994 rund 3,3 Milliarden Dollar umgesetzt werden sollen, in vier Segmente:

- Sprachnachrichtenübertragung (Voice Messaging),
- Telefonanruferverarbeitung (Voice Answering),
- Sprachausgabe (Voice Response) sowie
- interaktive Sprachinformation (Interactive Voice Response).

Sprachinformationen werden digitalisiert

In jedem dieser Bereiche werden die Sprachinformationen digitalisiert und von Rechnersystemen quasi wie Daten weiterverarbeitet. Als typische Anwendungen für die Voice-Mail-Lösungen nennt Total Communication die zeitversetzte Sprachkommunikation, Sprach-Mailboxen und Datenbankzugriffe per telefonischer Fernbedienung.

Für die Zukunft hat das Unternehmen zwei Trends ausgemacht. Der zunehmende Einsatz des PCs als Sprachverarbeitungsrechner belebt die Marktentwicklung, zudem prognostizieren die Breisgauer eine weitgehende Integration der Sprachdienste in Fax- und E-Mail-Systeme. Die Verlagerung der Anwendungen von Minicomputern der mittleren und oberen Leistungsklasse auf PCs bedingt zusätzlich einen Preisschwund, so daß auch kleinere Unternehmen diese Form

des Sprachdienste nutzen können. Besonders für Freiberufler bringen Voice-Mail-Systeme laut Total Communication den Vor-

teil der besseren Erreichbarkeit. Hohe Synergiepotentiale bestätigt die Prognose auch der Verknüpfung von Fax und Voice-Mail.

wie etwa bei Fax-on-demand-Anwendungen. Dabei können per Telefon Informationen abgerufen werden, die über Sprachausgabe via Fax an Interessenten geleitet werden.

Das Zusammenwachsen von Voice- und E-Mail-Systemen schätzen die Freiburger indes weniger positiv ein. Erst längerfristig könne eine Marktentwicklung ähnlich wie bei den Faxanwendungen stattfinden. Der Grund für

die schwächere Verbreitung der Sprachdienste im E-Mailbereich liegt laut Total Communications in der grundsätzlich unterschiedlichen Ausrichtung der Systeme: E-Mails werden typischerweise bei der Verteilung dupliziert, während Sprachinformationen zum Abrufen nur gespeichert werden. Die Vervielfältigung der gesprochenen Nachricht würde derzeit noch den Speicherbedarf und die Kosten ausufern lassen. ♦



Der europäische Imaging Kongreß.

Europas einzige wichtige Imaging-Messe 1994

Die Antworten finden Sie auf der IMC Document Imaging '94 in Berlin! Auf dieser Fachmesse bekommen Sie fundierte Informationen aus erster Hand und können die neuesten Anwendungen persönlich testen. Die IMC Document Imaging ist eine internationale Ausstellung, die Ihnen die ganze Palette des Angebots präsentiert: Produkte und Lösungen praktisch aller großen Hersteller.

Sind Sie bereit für den Erfolg? Bereit für die Herausforderungen, die Ihre Bild- und Dokumenten-Verarbeitung im kommenden

Telakom plant

Automatische Sprachübersetzungen: (ohne Nachbearbeitung durch einen Lektor)

Kapitel 5

BRAUCH DER TASTATUR

Diese sektion bittet den umriss des tastaturbrauchs an wenn die verwendeten funktionen ausgeführt werden.

Funktion	Kodes	Anmerkung
Heimschieber	Home	Schriftleiter; Wortverarbeiter
Zum aussersten menü zurück - kehren	Home	Getriebene anwendung des menüs
Schieber aufwärts fortbewegen	↑	Völlig schirmschriftleiter; Wortverarbeiter
Aufwärts paginieren, schnörkel rückwärts 25 linien und heim	PgUp	Schriftleiter; Wortverarbeiter
Schieber linkwärts fortbewegen	← Taste 75	Text. Kommandozugang
Schieber rechtwärts fortbewegen	→	Text, Kommandozugang
Schnörkel zum ende des textes . Am ende der linie schieber stellen	End	Schriftleiter; Wortverarbeiter
Schieber unterwärts fortbewegen	↓	Völlig schirmschriftleiter; Wortverarbeiter
Rückwärts paginieren, Schnörkel rückwärts 25 linien und heim	PgDn	Schriftleiter; Wortverarbeiter
Text im schieber einzufügen, text im puffer rechts zu bewegen beginnen/unterbrechen	Ins	Text, kommandozugang
Charakter im schieber austreichen	Del	Text, kommandozugang
Zerstörtischer rückplatz	← Taste 14	Text, kommandozugang
Streifen votwärts	→	Textszugang
Streifen reserve	←	Textszugang

Die grenzschnitt der STAFF tastatur ist begrenzt so dass das system software die maximalbiegbarkeit für die begrenzten gewissen tastaturoperationen hat. Dies ist durchgeführt mit die hin- und rück-fahrenden blickskodes der tastatur vielmehr als die Amerikanischen Standardkodes für die Kodes des Information-saustausches (ASCII). Ausserdem sind alle tasten typematic und erzeugen ein bearbeitungs- und anrisskode. Zum beispiel erzeugen die taste "I" blickskode hex 01 auf die bearbeitung und kode hex 81 auf den anriss. Die anrisskodes werden durch das hinzufügen hex 80 zur bearbeitungskodes gestaltet. Dieser führer der tastatur I/O kann die tasten wie shift tasten oder typematic as benötigt von der bewerbung begrenzer.

EINLEITUNG

Ihr neuer tragbarer Stereo-Rundfunkaufnahmegerät mit Doppelter Kassette wird anwerft besonders mit diesem Kassettendeck und einer Festen Stellung in AM, SW und FM-Stereo-Rundfunk. Die Zweien Kassettendecke versorgen Ihnen die Bequemlichkeit in verdoppelnder Bänder. Die Einheit wird anwerft zu erledigen den zwischenzeitlichen Norme.

WARNUNG

Lesen Sie dieses Unterweltauswahlmanual umständlich vor Irpenderer Wirkung zu sichern Sicherheit, Volle Nützlichkeit und Langes Leben für die Einheit.

Diese Einheit ist angemessen für die Wirkung auf 220-250 Volt, 50 Hz, die Ganzversorgung des Wechselstroms oder 7.5 Volt-Batterienversorgung nur. Diese Einheit muß nicht verbindet zur Ganz des Gleichstroms.

Ziehen Sie den Ganzstecker und Batterien zurück, ob diese Einheit wird nicht gebrauchen werden für einem bedeutenden langen Zeitraum. Batterien dürfen lecken ob sie werden lassen innerhalb der Einheit zurück und so sie dürfen verursachen den Schaden zur Einheit.

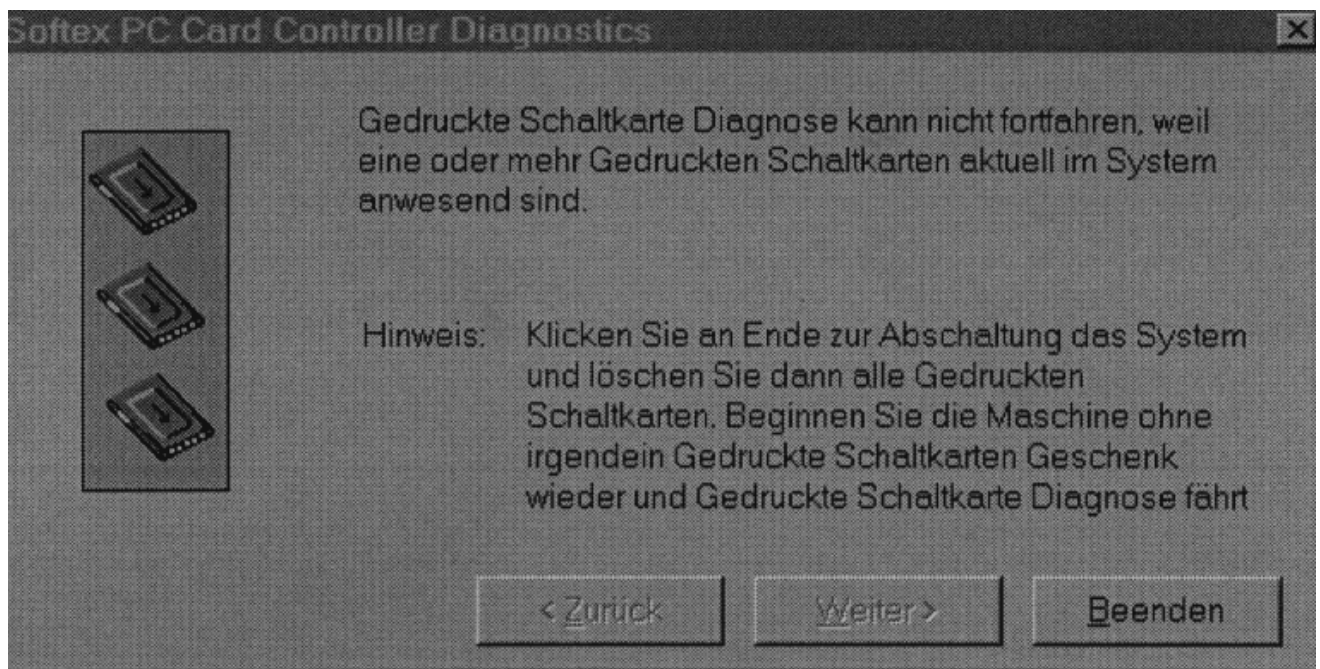
Nicht verbinden Sie einen Lautsprecher oder Kopfhörer von weniger als 4 Ohm Impedanz.

GANZENVERSORGUNGSWIRKUNG DES WECHSELSTROM

1. Setzen Sie den Verbinder des Ganzleitungsfadens in der Wechselstromsteckdose (20) der Einheit ein.
2. Sichern Sie die Ganzversorgung des Wechselstroms zu entsprechen die Starkstrom-Mischschaltung der Einheit bezeichnet an der Rückenseite der Einheit.
3. Stopfen den anderen Schluß des Ganzleitungsfadens in die Ganzversorgungsansteckdose des Wechselstroms zu.
4. Die Einheit ist nun bereit für die Wirkung.

BATTERIENWIRKUNG

1. Öffnen Sie die Batterienabteilungstür (28).
 2. Setzen Sie 6 Stück 1.5-Volt von UM-1, Batterien der D-Größe (nicht verpackt) oder gleichwertig in die Batterienabteilung (28), die Polarität als Anzeigen in der Batterienabteilungstür ein.
 3. Schließen Sie die Batterienabteilungstür und die Einheit ist nun bereit für die Wirkung.
- ANMERKUNG: Einsetzung des Verbinders des Ganzleitungsfadens in der Wechselstromsteckdose (20) wird entkuppeln Batterienversorgung selbsttätig.



Original und Fälschung

Eine Probeübersetzung – der Originaltext stammt aus dem „Wall Street Journal“ und lautet wie folgt:

„Western Europeans today endured a second night of howling winds after a day of picking through the destruction of a furious Sunday storm that killed upward of 70 people.“

Und das machten die Übersetzungsprogramme Babelfish und T-Mail daraus:


„Westeuropäer hielten heute eine zweite Nacht der Heulenwinde nach einem Tag der Ernte durch die Zerstörung eines furious Sonntag Sturmsaus, der aufwärts von 70 Leuten beendete.“

Derselbe Text, übersetzt bei Free-translation:

„Westliche Europäer haben eine zweite Nacht von Heulen Winde nach einem Tag von durch die Zerstörung eines wütenden Sonntags Sturm heute ausgedauert während hat, der mehr als 70 Leute getötet.“

In Kauderwelsch resultierte der Text bei E-Dolmetscher ITranslator:

„West Europäer heute ertrugen nach einem Tag vom Auswählen durch die Zerstörung von einem wütenden Sonntags Sturm, der aufwärts von 70 Leuten tötete, eine zweite Nacht heulender Winde.“

 Eine Übersicht von Übersetzungsdiensten:

http://kostenlos.de/@esdas_wuertabicher.htm

Übersetzungsprogramme:

Babylon:

<http://www.babylea.com>

Babelfish:

<http://babelfish.altavista.com>

Free-translation:

<http://www.free-translation.com>

ITranslator:

<http://usa.itranslator.com/publish>

Translate-Mail:

<http://www.t-mail.com>

Uni-Verse:

<http://www.universe.com>

2.3 Tintenstrahldrucker, Laserdrucker, Farbdrucker

Die Druckerauswahl sollte sich orientieren an:

Papierdurchsatz, Papiersorte	} Qualitätsmerkmale
Qualität des Drucks	
Einzelblätter/Endlospapier	
Doppelseitiger Druck	
Graphik	
Farbe	
Preis pro Blatt (inkl. Wartung)	

Sehr empfehlenswert ist die Seitenbeschreibungssprache Postscript/Level2.

Preisübersicht:

Schwarzweißdrucker:

<u>Matrix:</u>	24-Nadeln, 300 dpi oder 360 dpi, 200 . . . 500 cps (= characters per second)
<u>Tintenstrahl:</u>	300 dpi, <i>sehr</i> leise, langsamer als Laserdrucker
<u>Laserdrucker:</u>	300 dpi, 4 . . . 25 DIN A4-Seiten/min., viele Schriften, <i>extrem</i> leise, hoher pro-Seiten-Preis

Farbdrucker:

<u>(Matrix:</u>	Flächen ungleichmäßig)
<u>Tintenstrahl:</u>	nicht jedes Normalpapier, Schwarz- und Farbpatrone(!)	
<u>Thermotransfer:</u>	Spezialpapier, Wachsfarben, 16 Millionen Farben, Nebeneinander von verschiedenen Farbpunkten, Pixelstruktur wie bei Matrixdruckern	
<u>Sublimationsdrucker:</u>	thermische Erhitzung, gasförmig Sublimation in Spezialpapier, echte Farbauflösung von 300 x 300 dpi, <i>photorealistisch</i> , ~ 6 DM/Seite	
<u>Phase-Change:</u>	Hitze/Aufspritzen/Erkalten/Festwalzen, Normalpapier, Farbstifte	

Farbdrucker für digitale Fotografie:

- Fotorealistische Auflösung, Druck auf Photopapier/Folien...
- Farbdrucker mit Oberflächenversiegelung des Printmediums

Index

- 680x0, [3](#)
- 80x86, [3](#)

- 88000, [21](#)

- Adressierung, [2](#)

- Befehle, [2](#)
- Befehlszähler, [2](#)

- cache, [29](#)
- chip, [4](#)
- circuit switched, [43](#), [44](#)
- CISC, [3](#), [7](#), [10](#)
- CM2, [49](#)
- CM5, [51](#)
- Codierung, [2](#)
- CPU, [1](#)
- Cray Origin 2000, [37](#)
- Cray T3E-1200, [37](#)
- Cray T90, [37](#)
- Cray-1, [33](#)
- Cray-2, [36](#)
- Cray-3, [37](#)
- Cray-4, [37](#)

- data parallel programming, [28](#)
- Data Vault, [49](#)
- Daten, [1](#)
- Datendurchsatz, [29](#), [40](#)
- Datenpfad-Zykluszeit, [9](#)
- DEC-alpha, [13](#), [22](#)
- Decodierer, [2](#)
- Deschutes, [16](#)
- Durchsatz, [29](#)

- Ein- und Ausgabewerk, [1](#)
- EISA, [14](#)

- FRU, [25](#)

- General Scripts, [84](#)

- Hotswap, [47](#)

- IBM360, [3](#)
- ISA, [14](#)

- JPEG, [2](#)

- massiv, [49](#), [53](#)
- MBus, [43](#)
- MCA, [14](#)
- Mehrprozessorsysteme, [41](#)
- message passing, [42](#)
- Microprogramm, [6](#)
- MIMD, [28](#), [35](#)
- MIPS, [13](#), [23](#)
- MPEG, [2](#)
- Multimedia, [2](#)
- Multiprozessorsystem, [46](#)
- Multitasking, [46](#)

- Nanoprogramm, [7](#)
- NEC-SX2, [38](#)
- NEC-SX3/11, [39](#)
- NEC-SX3/44, [39](#)
- NEC-SX4, [39](#)
- NEWS, [49](#)
- NVRAM, [25](#), [26](#)

- P2SC, [14](#)

packet switched, 43, 45
parallel, 49, 53, 54
PCI, 14
Pentium, 13, 15
Pentium II, 16
Pentium MMX, 15
PentiumPro, 15, 17, 18
Pipeline, 11, 32
Power, 21, 22
Power Architecture, 13, 20
PowerPC, 14, 24
preemptiv, 46
Programm, 1, 5

Rechenwerk, 1
redundant, 47
Registerstack, 12
RISC, 9, 10
RISC-Designprinzip, 9

SBus, 43
Schaltwerk, 2
semantic gap, 7
Server, 47
shared memory, 42
SIMD, 28, 32, 35
SIMM, 25, 27
Simulation, 53
SPARC, 13, 19
SPECfp95, 18
SPECint95, 17
Speicher, 1
Speicherzelle, 2
SPMD, 28, 52
Sprungbefehl, 2
Steuerwerk, 1

tasks, 46
Threads, 46
Transputer, 42

Ultra-SPARC, 14

UNICODE, 82
Universalität, 1
USB, 45
USV, 47

VAX, 3
Vektoren, 29
Vektorrechner, 30
Vektorregister, 31
Vesa Local Bus, 14
virtuell, 42, 47
von-Neumann-Konzept, 1

wait-states, 3

XDBus, 43