

# **Kap.2**

## **Befehlsschnittstelle**

### **Prozessoren, externe Sicht**

- **2.1** elementare Datentypen, Operationen
- **2.2** logische Speicherorganisation
- **2.3** Maschinenbefehlssatz
- **2.4** Klassifikation von Befehlssätzen
- **2.5** Unterbrechungen
- **2.6** Prozesse

- Speicher „spiegeln“ den Zustand eines Rechensystems, bzw. eines Programms

„Konfiguration“

- Wir unterscheiden Hauptspeicher und Registerspeicher.

BB - RA - SS00

Kap. 2.2

2.2/3

## Hauptspeicher

- „großes“ adressierbares Array
- i.d.R. Byte-Adressierung, d.h. jedes Byte besitzt eine eigene, eindeutige Adresse
- größere Datentypen (z.B. int) sind mehreren Bytes zugeordnet (little endian, big endian: siehe unten)

BB - RA - SS00

Kap. 2.2

2.2/4

- Aus Performance Gründen ist Schreiben einzelner Bytes/Bits oft gar nicht möglich

- Ausrichtung an Wortgrenzen:

## Alignment

BB - RA - SS00

Kap. 2.2

2.2/5

## Alignment

- zusätzliches Problem: ausgerichtet (aligned) und nicht ausgerichtet (misaligned) Zugriff auf Speicher:

Objektadresse	Ausgerichtete Byteabstände	Nicht ausgerichtete Byteabstände
Byte	0, 1, 2, 3, 4, 5, 6, 7	ausgeschlossen
Halbwort	0, 2, 4, 6	1, 3, 5, 7
Wort	0, 4	1, 2, 3, 5, 6, 7
Doppelwort	0	1, 2, 3, 4, 5, 6, 7

- "misaligned" Zugriffe (die in der Regel erlaubt sind) sparen Speicherplatz aber kosten Zeit bzw. erfordern zusätzlichen Hardwareaufwand

BB - RA - SS00

Kap. 2.2

- maximale Größe:  
m Bits für Adresse
  - $2^m$  adressierbare Speicherzellen
  
- 16-bit Adressen bei DEC PDP-11
- 24-bit Adressen bei IBM Großrechnern
- 64-bit bei DEC Alpha, SuperSPARC

BB - RA - SS00

Kap. 2.2

2.2/7

### **Wie sind die Bytes numeriert?**

- little endian  
am wenigsten signifikanter Teil enthält die  
niedrigste Byte Adresse
  
- big endian  
der signifikanteste Teil enthält die  
niedrigste Byte Adresse

BB - RA - SS00

Kap. 2.2

2.2/8

## Little Endian - Big Endian

- Little Endian: z.B. 80x86, DEC PDP11/VAX, DEC ALPHA

Wortadresse	Byteadresse				Wertigkeit des Bytes
	03	02	01	00	
0	3	2	1	0	
4	7	6	5	4	

- Big Endian: z.B. MIPS, MC680x0, IBM 360/370

Wortadresse	Byteadresse				Wertigkeit des Bytes
	03	02	01	00	
0	0	1	2	3	
4	4	5	6	7	

BB - RA - SS00

Kap. 2.2

- von Rechner zu Rechner verschieden
- manche unterstützen beide Formate
- i.d.R. merkt der Benutzer nichts davon, es sei denn man möchte auf einzelne Bytes explizit zugreifen
- bei Strings sind big endians natürlicher, bei Zahlen little endians

BB - RA - SS00

Kap. 2.2

2.2/10

- Soviel zur logischen (Haupt-)Speichersicht  
strukturelle Eigenschaften später
- jetzt:

BB - RA - SS00

Kap. 2.2

2.2/11

## **Registerspeicher**

- Viel kleiner als Hauptspeicher, aber auch viel schneller
- kann genutzt werden, um lokale Informationen in Registern zu halten und so die Ausführung von Programmen zu beschleunigen

BB - RA - SS00

Kap. 2.2

2.2/12

## Designparameter

### ■ Größe

je nach Architektur unterschiedlich große Registerspeicher

- kleine Registerbank: wenige Bit zur Adressierung, aber für schnelle Ausführung ist dann schnelle Speicherarchitektur notwendig
- große Zahl: Befehlsgröße wächst, Ausführungszeit klein

BB - RA - SS00

Kap. 2.2

2.2/13

### ■ Funktionalität

- homogen: alle Register haben gleiche Funktionalität
  - einfach zu nutzen, man muß eventuell angeben wie Inhalt zu interpretieren ist
- inhomogen: „special purpose register“
  - erleichtert schnelle Ausführung von speziellen Operationen

BB - RA - SS00

Kap. 2.2

2.2/14

## Registerfunktionen

### ■ Adressregister

### ■ Datenregister

### ■ Spezialregister

BB - RA - SS00

Kap. 2.2

2.2/15

## Spezialregister

### ■ Akkumulator-Register (ACCU)

- speichert einen Operanden
- unterstützt Ausführung von 'Schieben'

### ■ Programmzähler (Befehlszähler)

- erzeugt durch Inkrementieren aufeinanderfolgende Programmadressen
- bei Sprungbefehlen neuen Wert laden

BB - RA - SS00

Kap. 2.2

2.2/16

## ■ Instruktionsregister

- nimmt den aus dem Speicher gelesenen Befehl auf

## ■ Statusregister

- Informationen, die aus Operationen resultieren (Überlauf, Übertrag, Vorzeichen)

BB - RA - SS00

Kap. 2.2

2.2/17

## ■ Stack-Register (Stapel-Register)

- aktueller Stand des Stackpointers
- Verwaltung von Rücksprungadressen und lokalen Variablen bei Unterprogrammen

BB - RA - SS00

Kap. 2.2

2.2/18

## Beispiele

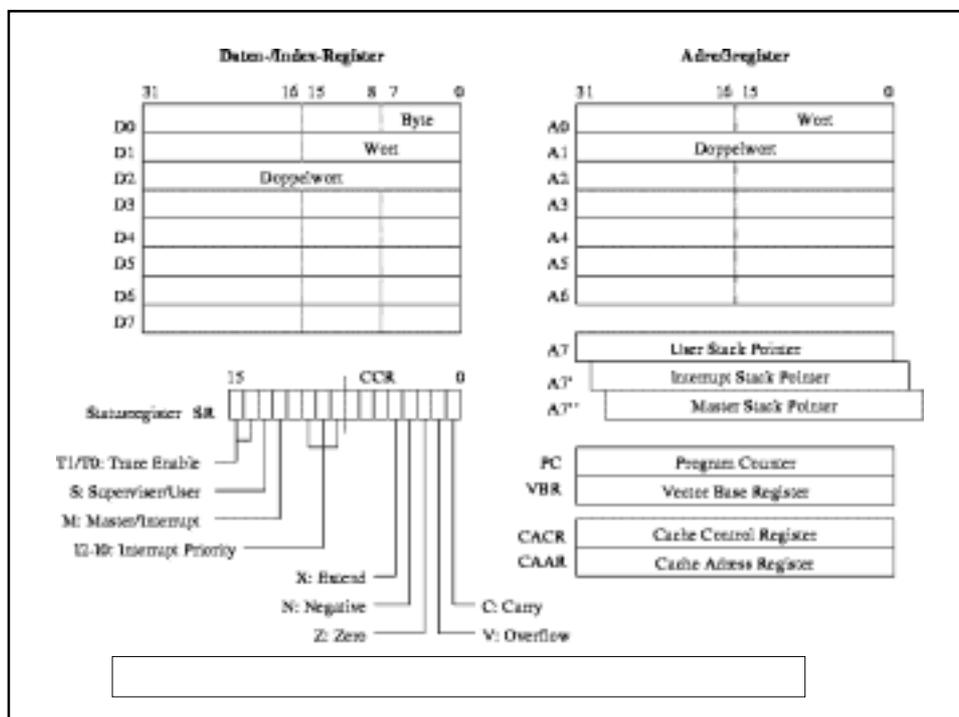
### ■ Motorola MC 68000 (1980-- )

- 16-bit Prozessor, interne Struktur 32 bit
- 16 weitgehend homogene 32-bit Register (8 Daten-, 8 Adress-)
- 5 Spezialregister
- optional 8 80-bit floating point Register

BB - RA - SS00

Kap. 2.2

2.2/19



## ■ MIPS I/II

- 32-bit RISC Prozessor
- 32 weitgehend homogene 32-bit Register (bis auf Register mit Nr. 0 und 1)
- 32 32-bit floating-point-Register (einzeln für single precision, paarweise für double precision)

BB - RA - SS00

Kap. 2.2

2.2/21

## ■ Intel 80386/486 (1985-90)

- 32-bit Prozessor
- 16 Register, 10 32-bit, 6 16-bit  
jedes Register hat (mehr oder weniger)  
spezielle Aufgaben
- 8 80-bit floating-point-Register

BB - RA - SS00

Kap. 2.2

2.2/22

