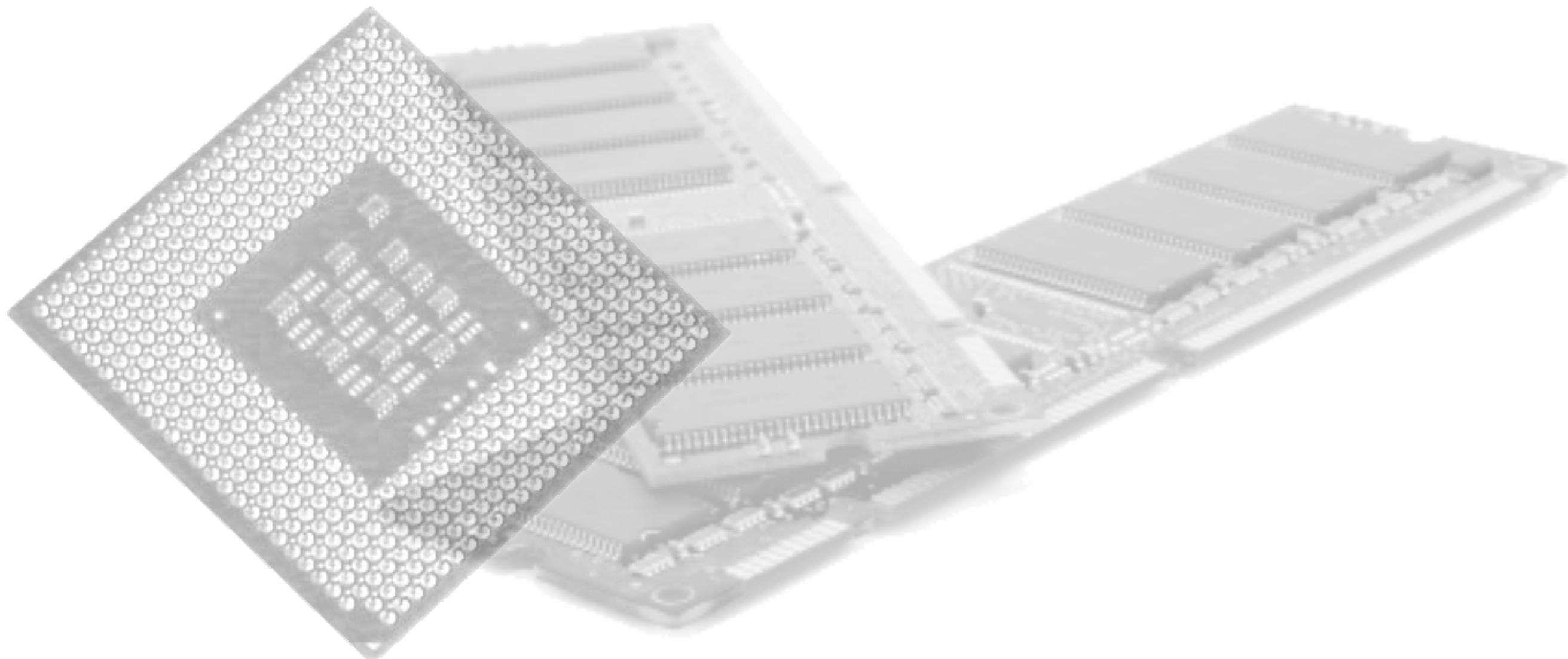
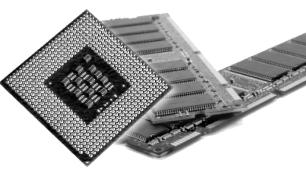


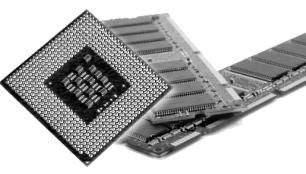
## Zusammenfassung Assembler





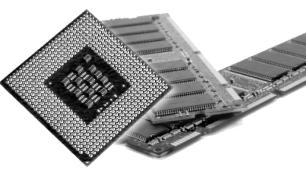
## Assembler Arithmetik

- **MOV** <Ziel>, <Quelle> (Quelle nach Ziel verschieben)
- **ADD** <Ziel>, <Quelle> (Quelle zu Ziel addieren)
- **SUB** <Ziel>, <Quelle> (Quelle von Ziel subtrahieren)
- **MUL** <Quelle> (2.Operand: EAX; Ergebnis: EDX + EAX)
- **DIV** <Quelle> (Dividend: EDX + EAX; Ergebnis: EAX, Rest: EDX)
- **INC** <Ziel> (Ziel um eins erhöhen)
- **DEC** <Ziel> (Ziel um eins verringern)



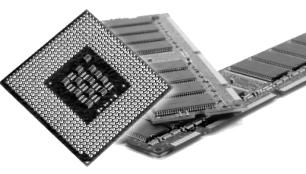
## Assembler Variablen

- in section .data
- **Hallo DD 12h**
- Typen:
  - DB - 1 Byte
  - DW - 2 Byte
  - DD - 4 Byte
  - DQ - 8 Byte



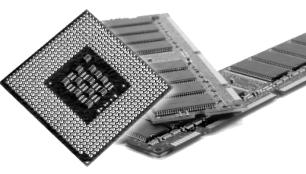
## Stack

- **PUSH** <Quelle> (Daten ablegen)
- **POP** <Ziel> (Daten auslesen)
- **PUSHF** (Flags auf den Stack)
- **POPF** (Flags vom Stack)



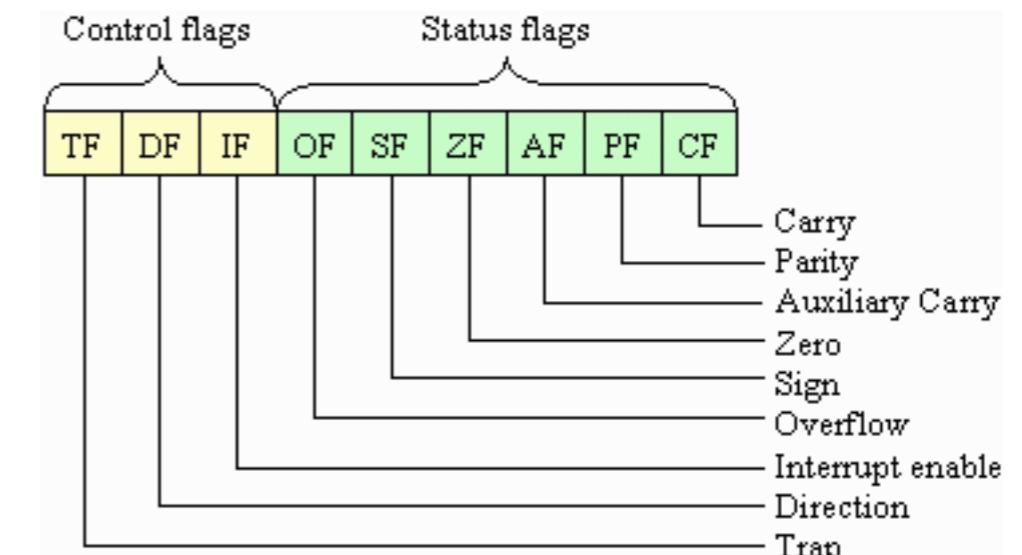
## Vergleichsbefehle

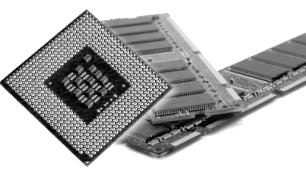
- **CMP** <Ziel>, <Quelle> (numerischer Vergleich von Ziel und Quelle)
- **CMPSW** (Compare String Word)
- **CMPSB** (Compare String Byte)
- Vergleicht ein Zeichen von der Speicherstelle die in DS:SI steht mit der von ES:DI



## Flags

- Control flags:
  - Trap
  - Direction
  - Interrupt enable
- Status flags
  - Overflow
  - Sign
  - Zero
  - Auxiliary Carry
  - Parity
  - Carry





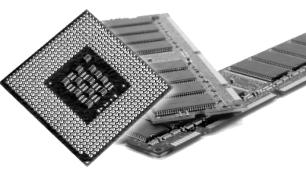
## Sprungbefehle

- **JMP <Ziel>** (Springt immer an das Ziel)
- **J... <Ziel>** (Springt je nach Bedingung)

Befehl	VZ	Flags
JO		OF = 1
JNO		OF = 0
JS		SF = 1
JNS		SF = 0
JE		ZF = 1
JZ		
JNE		ZF = 0
JNZ		

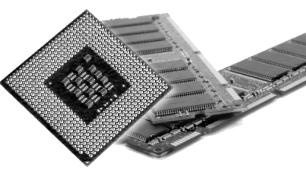
Befehl	VZ	Flags
JB		
JNAE	kein VZ	CF = 1
JC		
JNB		
JAE	kein VZ	CF = 0
JNC		
JBE		
JNA	kein VZ	CF = 1 oder ZF = 1
JA		
JNBE	kein VZ	CF = 0 und ZF = 0

Befehl	VZ	Flags
JL		
JNGE		VZ SF <> OF
JGE		
JNL		VZ SF = OF
JLE		
JNG		VZ ZF = 1 oder SF <> OF
JG		
JNLE		VZ ZF = 0 und SF = OF
JP		
JPE		PF = 1
JNP		
JPO		PF = 0
JCXZ		CX = 0
JECXZ		ECX = 0



## Schleifen

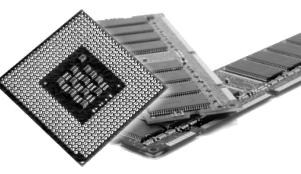
- **LOOP <Ziel>** (Solange ans Ziel springen, bis CX = 0)
- **LOOPE / LOOPZ <Ziel>** (solange Zero-Flag = 1)
- **LOOPNE / LOOPNZ <Ziel>** (solange Zero-Flag = 0)
- bei jedem Sprung wird CX um eins verringert



## Makros

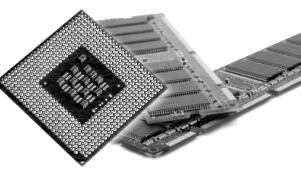
- **%macro** <Name> <Anzahl Parameter>
  - Code
  - Parameter mit %1, %2, %3, usw. verwendbar
- **%endmacro**

```
%macro AUSGEBEN 1
    NEWLINE
    PRINT_HEX 4, %1
%endmacro
```



## Unterprogramme

- <Name>:
  - Code
  - **RET**
- Aufruf: **CALL** <Name>



## Logische Operatoren

- **AND** <Ziel>, <Quelle>
- **OR** <Ziel>, <Quelle>
- **XOR** <Ziel>, <Quelle>
- **NOT** <Ziel>