

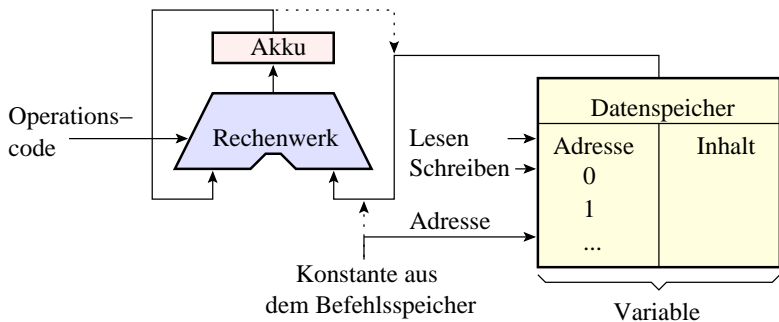


# Praktikum Mikrorechner 3 (Adressierungsarten)

G. Kemnitz

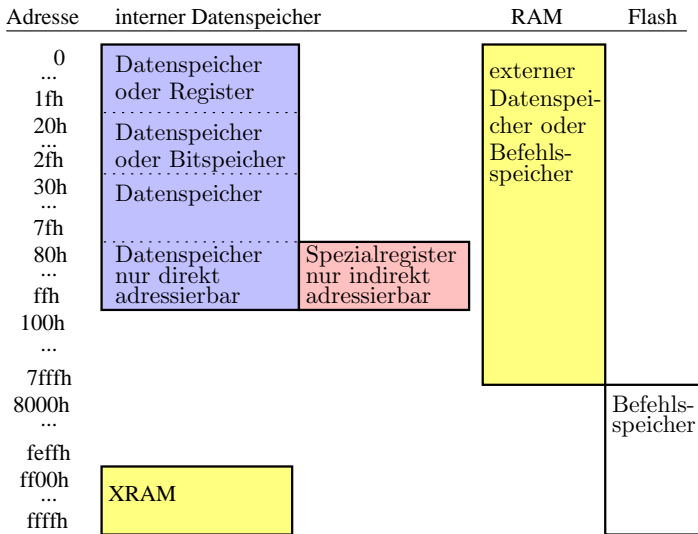
Institut für Informatik, Technische Universität Clausthal  
November 5, 2014

## Wo steht der 2. Operand



- im internen Datenspeicher
- in einem Register
- im externen Datenspeicher
- im Befehlsspeicher (Konstante)

## Die Speicherbereiche und Adressräume im Überblick



## Interner Datenspeicher

- direkte Adressierung: Angabe der Operandenadresse im Befehl

```
mov a, 60h
```

- indirekte Adressierung: Angabe der Adresse, wo die Adresse des Operanden steht

```
mov r0, #60h
```

```
mov a, @r0
```

0	Datenspeicher oder Register
1fh	
20h	Datenspeicher oder Bitspeicher
2fh	
30h	Datenspeicher
...	
7fh	
80h	Datenspeicher nur direkt adressierbar
...	
ffh	

### Besonderheiten und besetzte Adressen

- DS(0) bis DS(1fh) auch als Register ansprechbar
- DS(20h) bis DS(2fh) auch bitadressierbar
- DS(80h) bis DS(ffh) nur indirekt adressierbar
- DS(0) bis DS(ffh) auch als Stack benutzbar
- Variablen des Monitors: 20h, 21h, 59h, 80h-92h, ...

## Register

- Die ersten 32 Byte des internen Datenspeichers
- Speicherplätze für häufig benutzte Variablen.
- 4 Registerbänke mit je 8 Registern
- Die Register r0 und r1 jeder Bank können als Indexregister (Zeiger) verwendet werden
- Datenspeicheradressen der Register:

$$dadr = n + 8 \cdot RS$$

( $0 \leq n \leq 7$  – Registernummer;

$0 \leq RS \leq 3$  – Banknummer, bestehend aus den Bits RS1 und RS0 im Spezialregister PSW (Programmstatuswort)

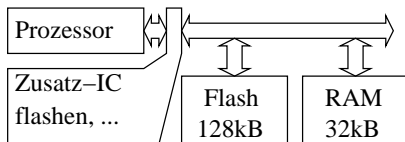
0h	r0	Bank 0
1h	r1	
⋮	⋮	
7h	r7	
8h	r0	Bank 1
9h	r1	
⋮	⋮	
0fh	r7	
10h	r0	Bank 2
11h	r1	
⋮	⋮	
17h	r7	
18h	r0	Bank 3
19h	r1	
⋮	⋮	
1fh	r7	



## Kontrollaufgabe: Was steht am Ende des Programmfragments im Akku?

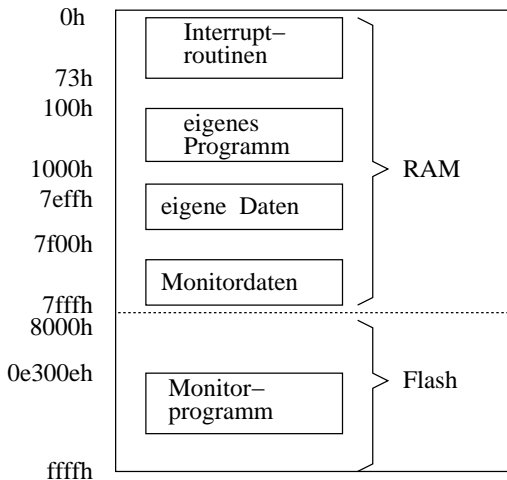
```
mov r0, #35h
mov r1, #62h
mov 62h, #11h
mov a, @r1
add a, r0
```

## Externer Speicher



- Extern angeschlossene Speicherschaltkreise
  - 128 kByte Flash-Speicher
  - 32 kByte RAM
- Die oberen Adressbits des Flash- und des RAM-Speichers sind über den Zusatz-IC umprogrammierbar.
- In der Standardkonfiguration belegt der RAM die unteren 32 k und die Flash-Seite mit dem Monitor die oberen 32 k des externen Adressraums.
- Die Schnittstelle zum externen Speicher benötigt 18 der 32 IO-Leitungen des Prozessors.

## Standardkonfiguration







## Adressierung

- der externe Speicher ist nur indirekt adressierbar
- mögliche Indexregister (Zeiger): r0, r1, dptr  
(16-Bit-Spezialregister)

ext. Speicher  $\Rightarrow$  Akku

```
movx a, @dptr
```

```
movx a, @r0
```

```
movx a, @r1
```

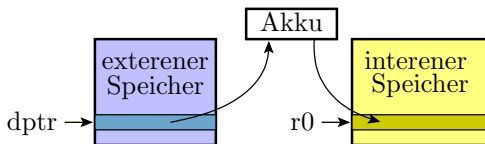
Akku  $\Rightarrow$  ext. Speicher

```
movx @dptr, a
```

```
movx @r0, a
```

```
movx @r1, a
```

## Macro zum Lesen des externen Speichers



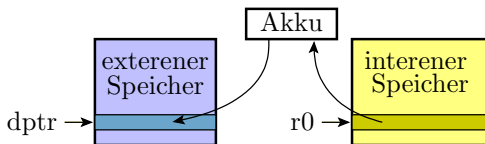
```

rdex_mac MACRO dadr, xadr
; DS(dadr) := EXTS(xadr)
; V: acc, dptr, r0
mov dptr, #xadr
mov r0, #dadr
movx a, @dptr
mov @r0, a
ENDM

```

**V:** veränderte Speicherinhalte

## Macro zum Schreiben des externen Speichers



```
wrex_mac MACRO xadr, dadr
; EXTS(xadr):=DS(dadr)
; V: acc, dptr, r0
mov dptr, #xadr
mov r0, #dadr
mov a, @r0
movx @dptr, a
ENDM
```



## Befehlsspeicher

- nur lesbar, Speicherplatz für Konstanten
- Lesen von der Adresse  $\text{dptr} + a$

```
movc a, @a+dptr
```

- Programmieren von Konstanten im Befehlsspeicher

```
org 200h  
db 7fh, 43h
```

- Benutzung:

```
mov dptr, #200h  
mov a, #0  
movc a, @a+dptr
```



## Aufgabe 3.1

Ergänzen Sie im nachfolgenden Programmskelett die Verarbeitungsbefehle entsprechend der als Kommentar angegebenen Funktion:

```
Wert      equ    4h
m_extRAM  equ    2a28h
n_intrAM  equ    62h
p_extRAM  equ    9fh
...
org 100h
; extRAM(m_extRam) = Wert+1
; intrAM(n_intrAM) = extRAM(m_extRam)+1
; Register r0 = intrAM(n_intrAM) + 1
ljmp 0e300h
```

Testen Sie das Programm im Schrittbetrieb.



## Aufgabe 3.2

Entwickeln Sie ein Programm, dass

- auf der Adresse 300h im Befehlsspeicher die Konstante 56h enthält
- den internen Speicherplatz mit der Adresse 80h mit dieser Konstanten beschreibt (Achtung indirekte Adressierung)
- in den externen Speicher Adresse 2000h die Konstante 03h schreibt (Adressierung mittels Data-Pointer)
- den Wert auf der Adresse 2000h des externen Speichers in den Akku kopieren
- den Wert auf Adresse 80h des internen Speichers dazu addiert
- das Ergebnis in Register r0 ablegt
- abschließend zum Monitor springt.

Testen Sie das Programm im Schrittbetrieb.



## Aufgabe 3.3

Entwickeln Sie ein Makro, das zwei 8-Bit-Zahlen aus dem externen Speicher addiert und das Ergebnis in den externen Speicher zurückschreibt.

```
add8ext MACRO sum, s1, s2
; sum, s1, s2: Adressen im externen Speicher
; extRAM(sum) = extRAM(s1) + extRAM(s2)
```