



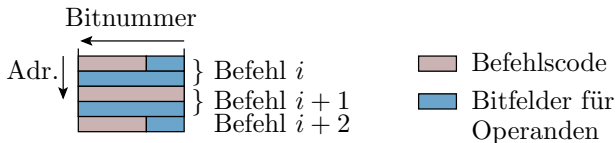
Praktikum Mikrorechner (Wiederholung Adressierungsarten)

Prof. Kemnitz

Institut für Informatik, Technische Universität Clausthal
November 5, 2014

Befehlsaufbau und Assembler

Befehlsspeicher



Assemblernotation für einen Befehl

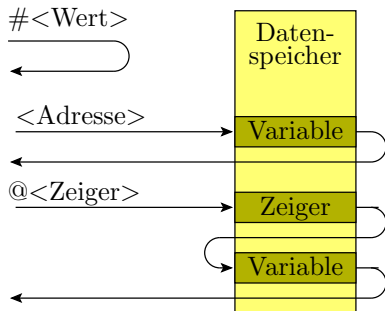
<Befehl> <Operand 1>, <Operand 2>

<Befehl> = {add, addc, subb, ...}

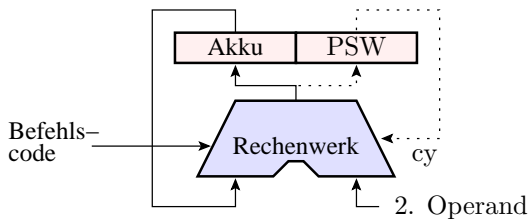
<Operand>

- ↗ implizit: Operand im Befehlscode festgelegt
- unmittelbar: Bitfeld ist eine Konstante
- direkt: Bitfeld ist eine Adresse
- ↘ indirekt: Bitfeld ist eine Adresse für eine Adresse

Konstanten, Variablen und Zeiger



Verarbeitungsbefehle



<Befehlscode> a, <2. Operand>

add	#<Wert>
addc	<Adresse>
subb	@<Zeiger>
...	...



Transportbefehle

Kopieren eines Bytes von einem Speicherplatz auf einen anderen.

- Zwischen Speicherplätzen im internen Speicher

```
mov <Ziel>, <Quelle>
```

<Ziel> = {Akku (implizit), Adresse (dirket),
Register, Adresse der Adresse}

<Quelle> wie <Ziel> und zusätzlich eine Konstante

- Zwischen dem externen Speicher und dem Akku

```
movx a, @dptr
```

```
movx @dptr, a
```

- Aus dem Befehlsspeicher in den Akku

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



1.