

# Tutorat 2

# Hardware

# Korrektur

# Korrektur

## Häufige Fehler und Interessantes

- `last -s today` oder `last -s 0:00`
- `find -iname "*.pdf"` for insensitive
- `pwd` "Pfad" (I-Node System)
- "Datein und Ordner" bei `ls`
  - im I-Node System sind Verzeichnisse quasi Dateien
- `cd /usr/share/doc` statt `cd /; cd usr; cd share; cd doc`
- versteckte Dateien und Verzeichnisse ( `.file` und `.folder` )
- `cp -r` und `mv -r?`, `mkdir -r`, `rm -r`, alias
- `date +"Datum: %F, Zeit: %T nicht %X"`
- `cd` zu `/home/<user>` bzw. `~`
- `/` gehört Root, und `~` gehört User

# Korrektur

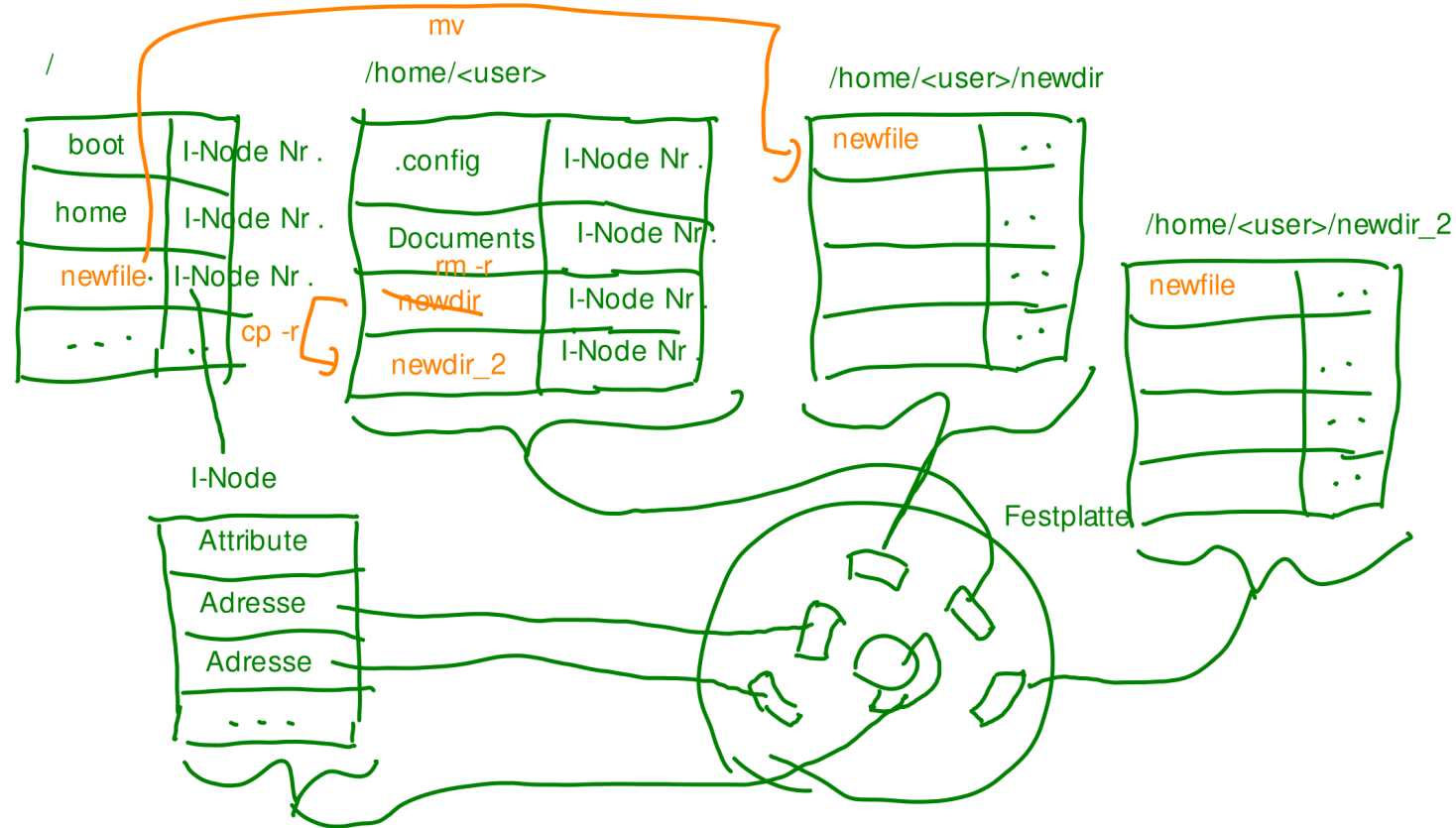
## Häufige Fehler und Interessantes

- `cp /pfad/datei_1 ./pfad/datei_2 /pfad_2`: kopieren
  - `cp /pfad/datei ./pfad/datei_2`: kopieren und umbenennen
  - `-r`: copy directories recursively (Inhalte von nichtleeren Ordnern werden mitkopiert)
- `mv /pfad/datei_1 ./pfad/datei_2 ./pfad_2`: verschieben
  - `mv /pfad/datei /pfad/datei_2`: umbenennen
- `rm /pfad/datei`: remove
  - `-r`: remove directories recursively (nichtleere Ordner löschen)

# Korrektur

## Aufgabe c) - 1

<del>pwd</del>
<del>cd ..</del>
<del>ls -l</del>
<del>cd</del>
<del>pwd</del>
<del>mkdir newdir</del>
<del>cd /</del>
<del>pwd</del>
<del>cd newdir</del>
<del>cd ../newdir</del>
<del>pwd</del>
<del>cd ..</del>
<del>touch newfile</del>
<del>ls</del>
<del>mv newfile newdir</del>
<del>ls newdir</del>
<del>cp -r newdir newdir2</del>
<del>rm -r newdir</del>
<del>ls</del>

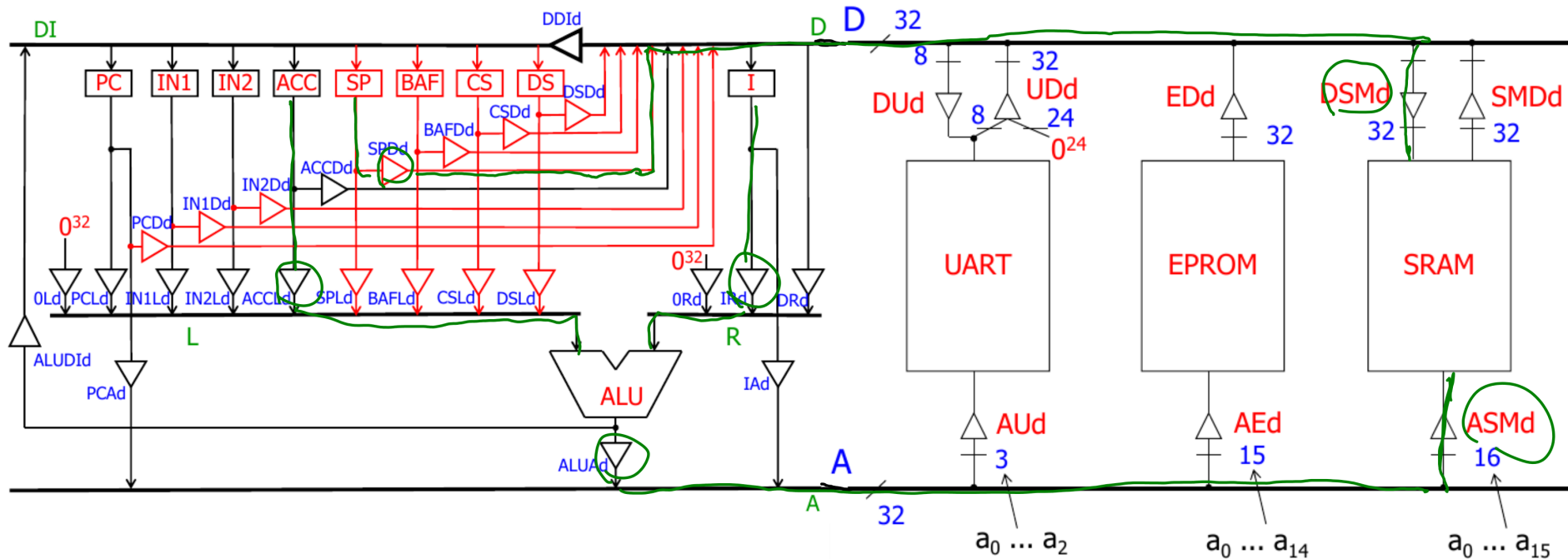


# Übungsblatt

# Übungsblatt

## Aufgabe 1

- STOREIN ACC SP i :  $M(<ACC>+[i]) := SP$



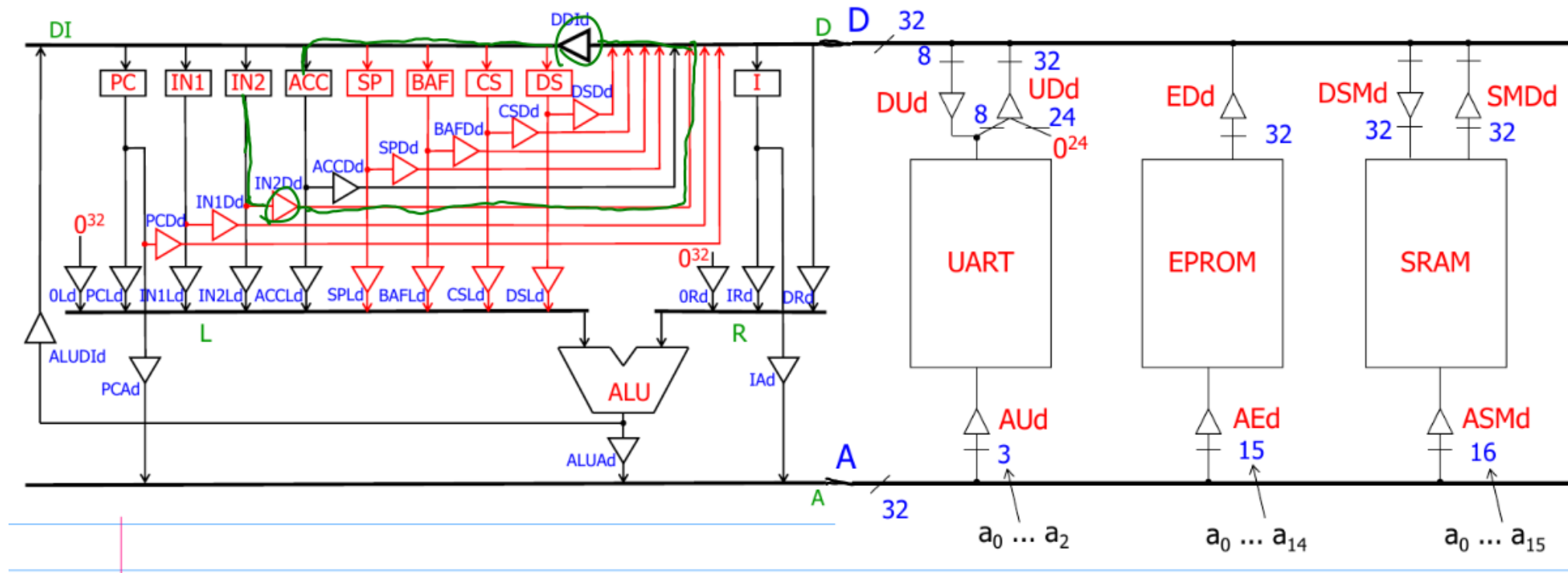
Die erweiterte RETI



# Übungsblatt

## Aufgabe 1

- **MOVE IN2 ACC** :  $ACC := IN2$



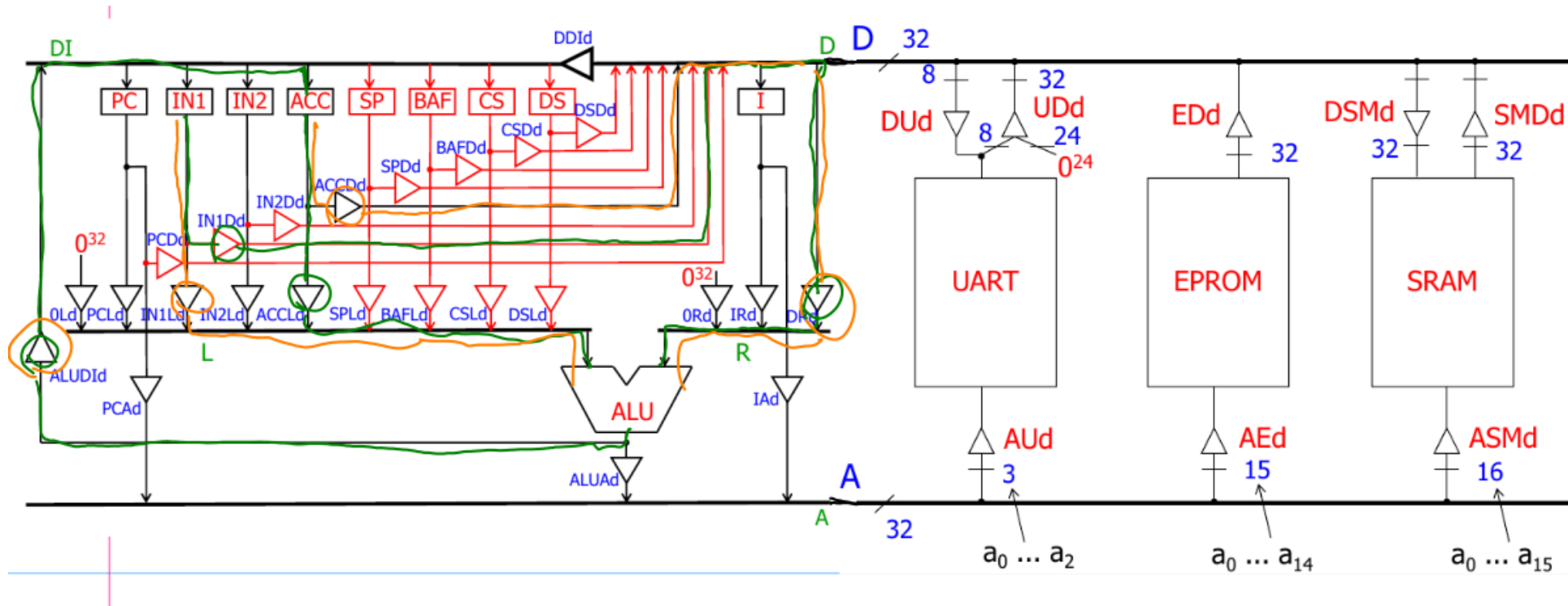
Die erweiterte RETI



# Übungsblatt

## Aufgabe 1

- ADD ACC IN1 :  $ACC := ACC + IN1$

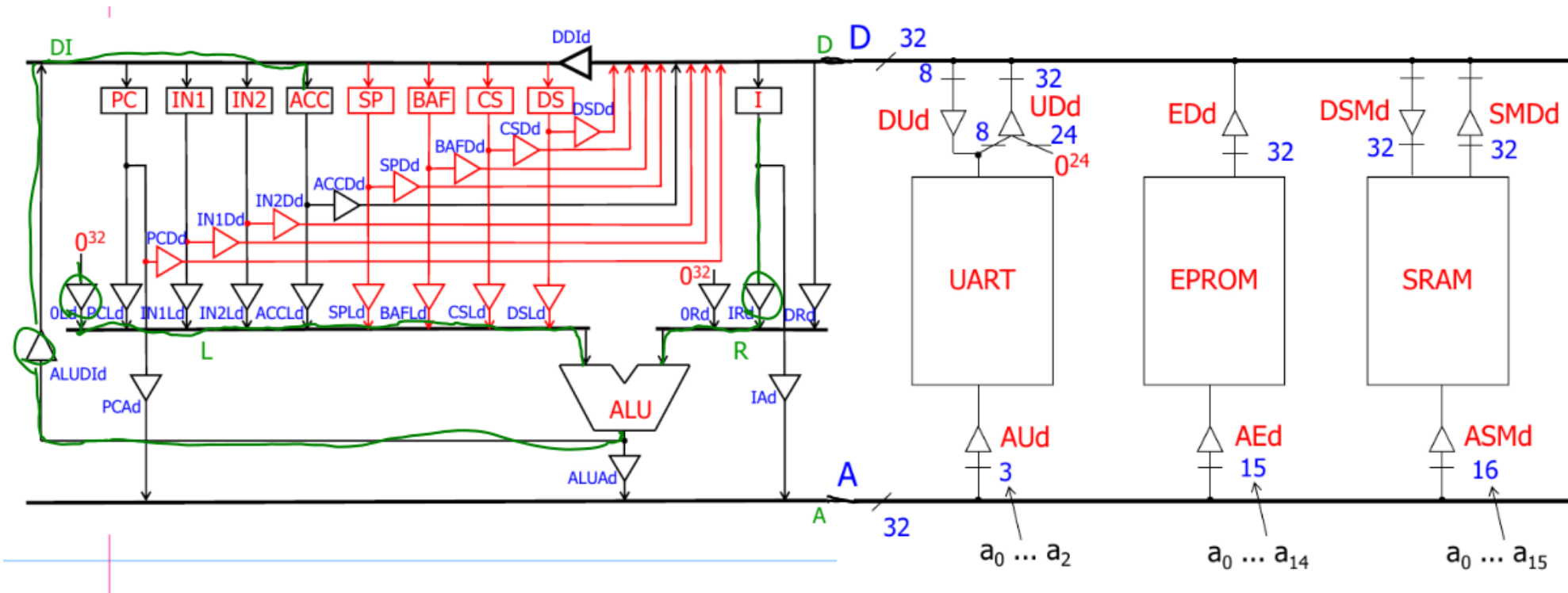


Die erweiterte RETI

# Übungsblatt

## Aufgabe 1

- 0 auf dem linken Operanden-Bus: `LOADI ACC i` ( $ACC := 0^{10}i$ )



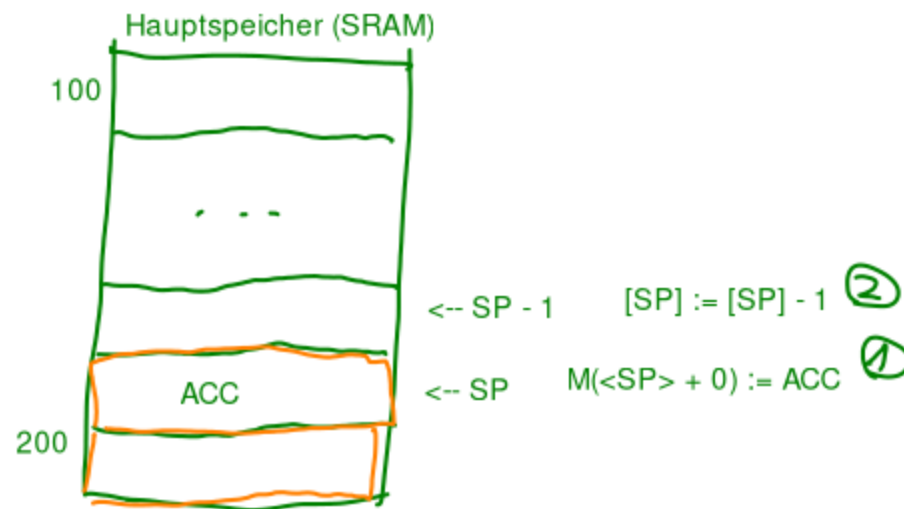
Die erweiterte RETI

# Übungsblatt

## Aufgabe 2

- **subtrahieren**  $\hat{=}$  Speicher allokieren, **addieren**  $\hat{=}$  Speicher freigeben
- **push:**

```
STOREIN SP ACC 0  
SUBI SP 1
```

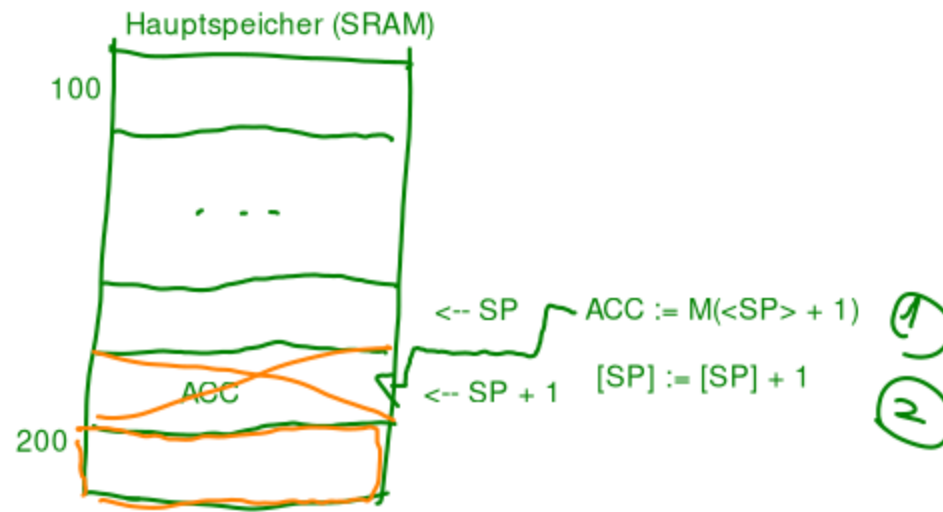


# Übungsblatt

## Aufgabe 2

- **pop:**

LOADIN SP ACC 1	ODER	ADDI SP 1
ADDI SP 1		LOADIN SP ACC 0



# Übungsblatt

## Aufgabe 2

“ Wozu INT i und RTI einführen? Warum nicht mit restlichen Befehlssatz das gleiche umsetzbar? ”

- **Software-Interrupt:**

- mit Nummer **i** aus **Interruptvektortabelle IVT** Anfangsadresse von Routine auslesen
- Stand des PCs auf dem Stack zwischenspeichern
- Sprung an Adresse  $IVT[i]$  ( $PC := IVT[i]$ )
- Wechsel in **Systemmodus**
- Rückkehr mit **RTI** als letzter Befehl der Betriebssystemroutine

**Vielen Dank für  
eure  
Aufmerksamkeit!**

