

RA1 29.6.2012

weiter zu 3.4. Befehlsübersicht  
weiter zu Programmtransferbefehle

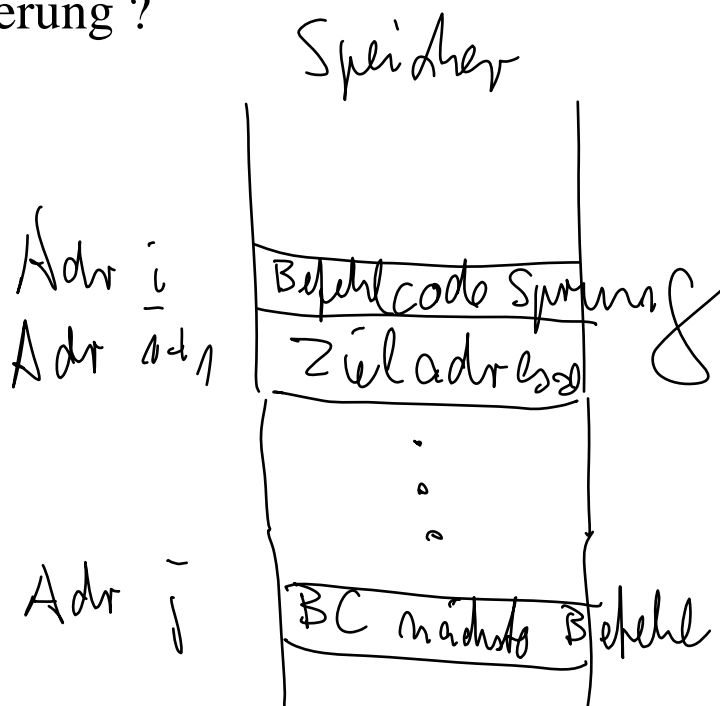
## Sprungbefehle

- dienen zur evtl. bedingten Verzeigung von Maschinenprogrammen bzw. zur Fortsetzung an einer Adresse, die nicht unmittelbar auf den aktuellen Befehl folgt (Zusammenführung (fest)) -> F3\_180 links

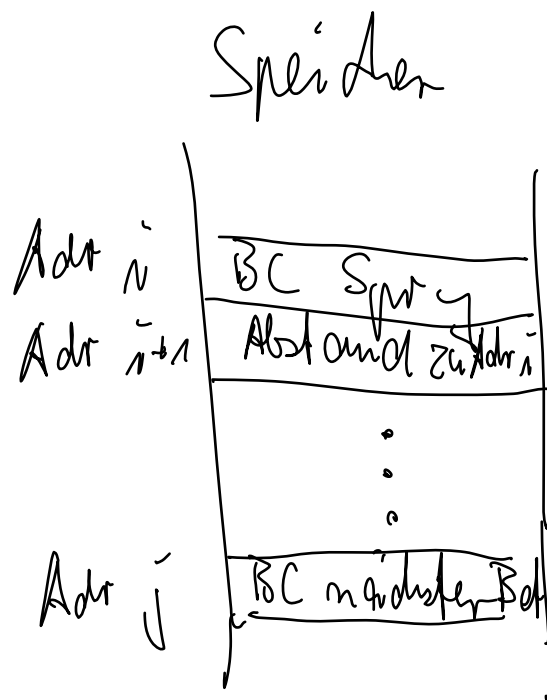
Unbedingter Sprung (benutzt für Zusammenführung)

Ziel: nach dem Sprungbefehl Weiterarbeit an vorgegebener Adresse

- Adresse wird im Befehl angegeben (Festwert, absolut)  
Kodierung ?



- Adresse wird im Befehl angegeben (Festwert Differenz (Abstand im Adressbereich) zur Adresse des aktuellen Befehls, relativ)



$$\text{Adr. } j = \text{Adr } i + \text{Abstand}$$

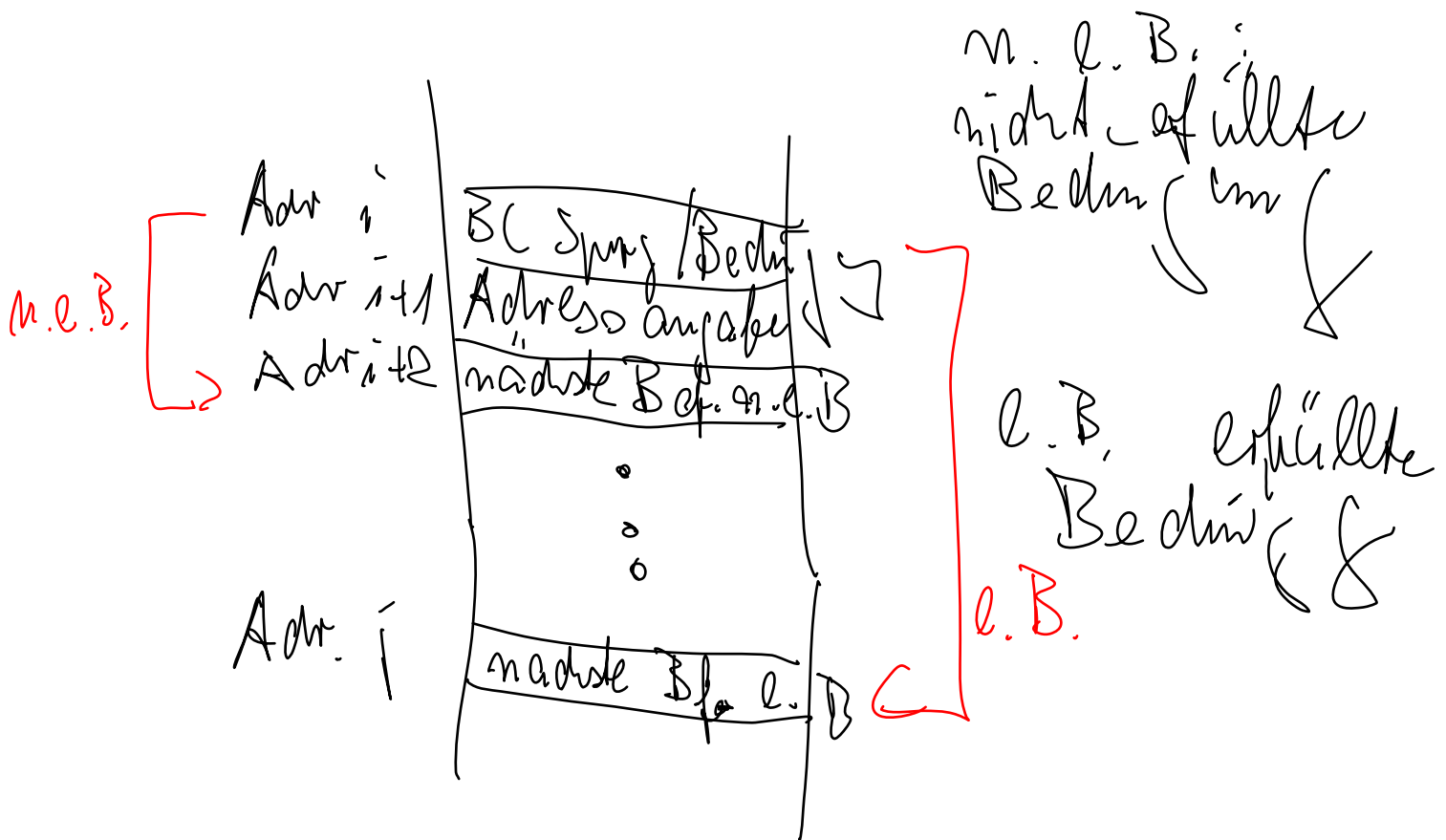
Basisadressierung im Programmbereich ist ähnlich:

Unterschied: Abstand (offset) ist hier (Basisadr.) immer auf den ersten Befehl des Teilprogramms bezogen

- Adresse steht in einem Register und kann evtl. Ergebnis einer Berechnung sein (indirekt)

Bedingte Sprünge:

Bedingung erfüllt (Bedingung ist i.a. bit im PSR)



- Unterprogrammbefehle

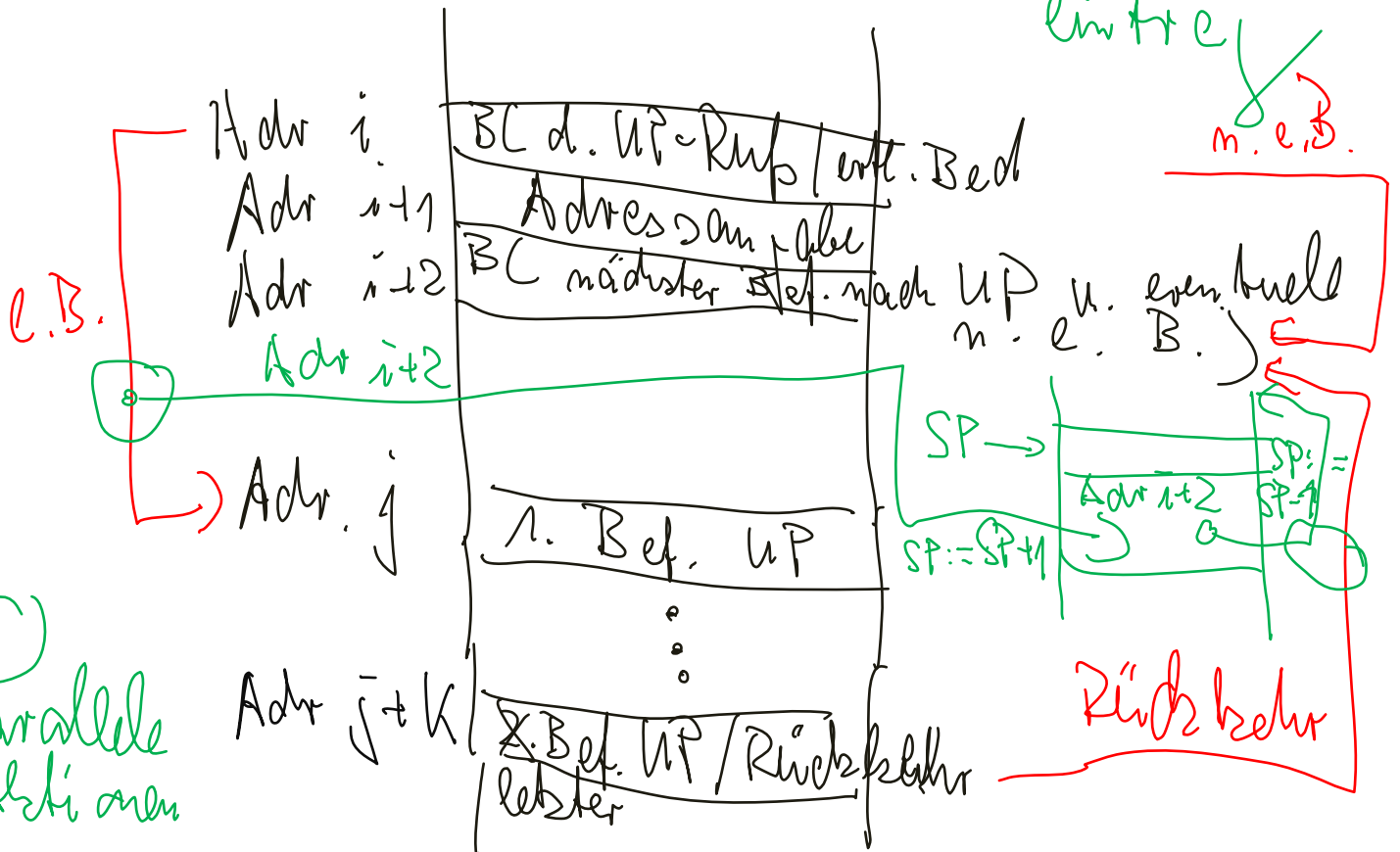
Unterprogramm ? (F3\_180 rechts)

- mehrfach nutzbares Teilprogramm
- wird von verschiedenen Stellen im Befehlscodebereich aufgerufen
- nach Abarbeitung Zusammenführung auf den Befehl nach dem aufrufenden Befehl (variabel)
- Grundlage der Prozedur- bzw. Funktionsaufrufe einer höheren Programmiersprache

notwendig: Abspeichern der Rückkehradresse im Stack (LIFO-Prinzip)

SP: Adresse abt. letzter Stack-Eintrag

m. e. B.

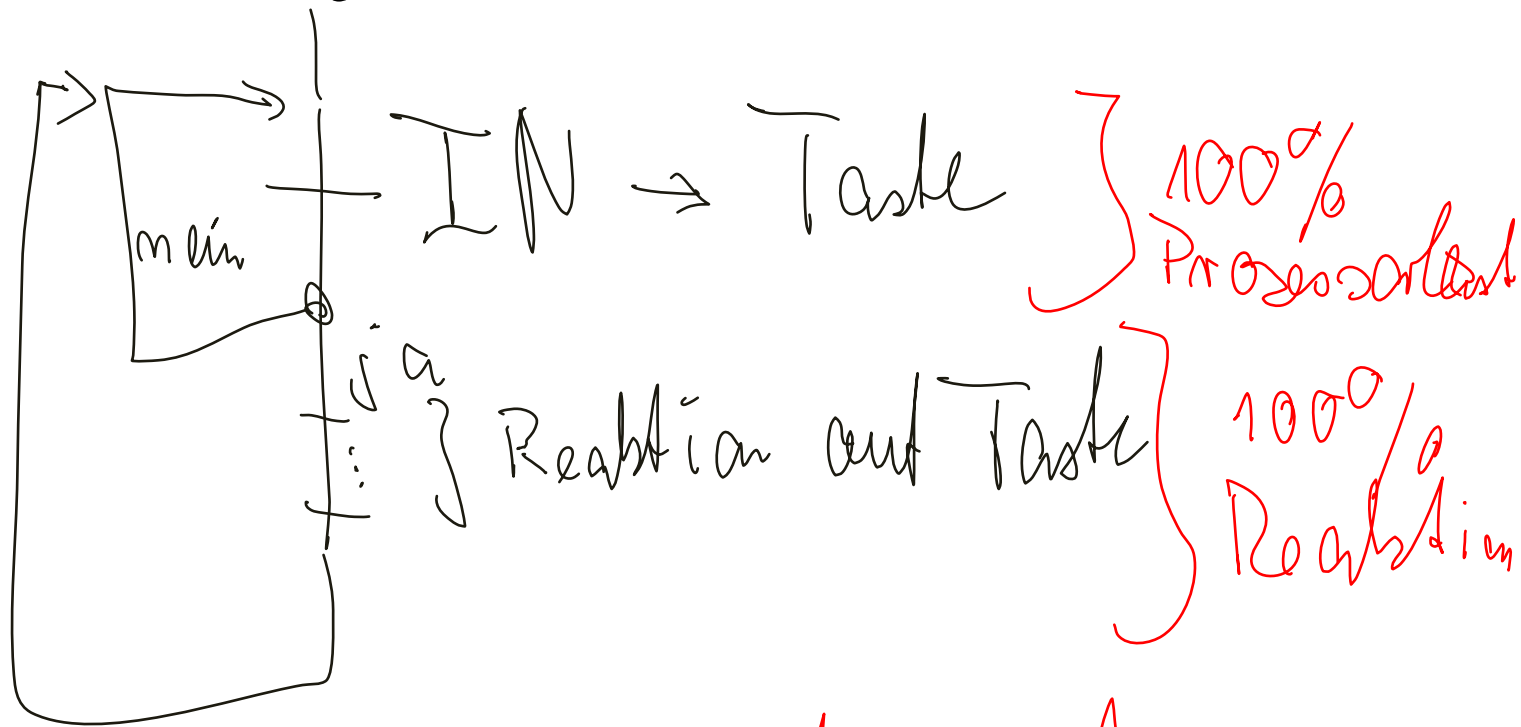


- Interrupt (Mischung zwischen Logik (Erweiterung der Prozessorgrundstruktur (Interruptprogrammaufruf) und Befehl (Interruptrückkehr))

Problem (anschaulich)

Bsp. Notwendig programmtechnische Reaktion auf Taste

mit den bisherigen Befehlen und Mechanismen:



Ausweg: Interrupt  
 (Programm mit Unterbrechung)

Funktionsweis Int. -> F3\_200

- i.a. arbeitet der Prozessor ein unabhängiges HP (von der Taste unabh.) mit 100% Prozessorleistung ab)

- zu einem dem HP nicht bekannten Zeitpunkt erfolgt ein Interruptereignis (hier Wechsel des logischen Werts der Taste (0->1))
- Prozessor ordnet dem Ereignis eine Startadresse zu und springt auf diese nach Ende des aktuellen Befehls (hier Adr. 1. Bef. IP)
- dabei wird die Rückkehradr. (hier die Adr. von Bef i+1) in dem Stack geschrieben (siehe UP)
- am Ende von IP steht der Befehl Int. rückkehr (hier IRET) -> Rückkehr aus Stack lesen und als Befehladresse für nächsten Bef. benutzen (hier Adr. von Bef +1)
- während IP 100% Prozessorleistung für IP
- meistens wird der aktuelle PSR-Wert auch im Stack gespeichert und mit IRET zurück geholt. (Grund 3\_210)

## 4. Speicher

### Aufgaben:

- physischer Träger der Programme und (Zwischen-)Operanden

### dafür notwendige Funktionen

- Schreiben in den Speicher
- Speichern der geschriebenen Werte bis zum nächsten Schreiben
- Lesen des letzten geschriebenen Wertes

### Hilfsfunktion:

- Auswahl des zu Lesenden bzw. zu Schreibenden Wertes im Speicher (Datenwort, DW) durch Adressierung (siehe F3\_20)

nächstes mal weiter mit interner Aufbau des Speichers und Einordnung in den Adressraum