

Betriebssysteme (BS)

VL 4.1 – Unterbrechungen, Software – Grundlagen

Volkmar Sieh / Daniel Lohmann

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme

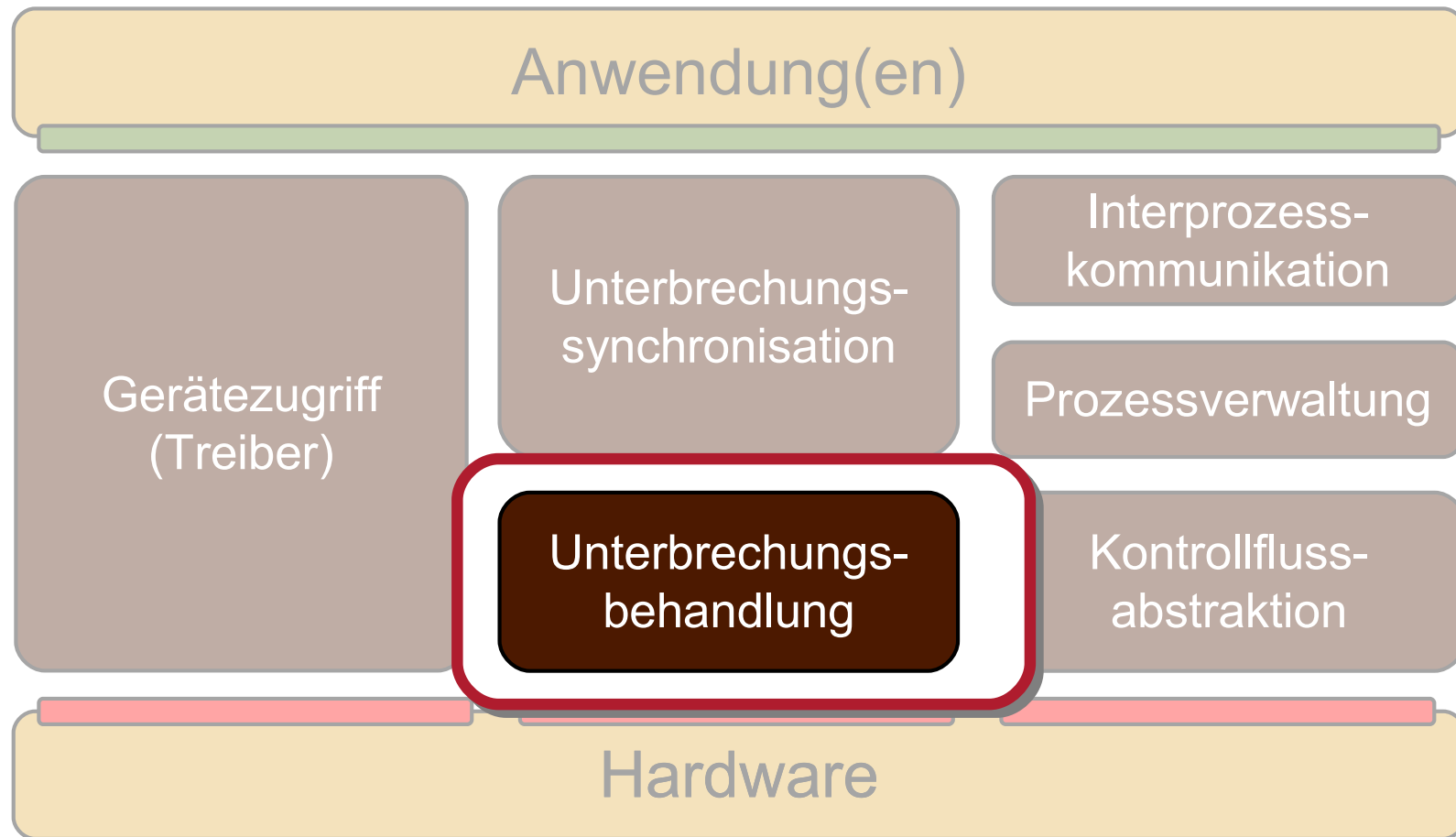
Friedrich-Alexander-Universität
Erlangen Nürnberg

WS 20 – 16. November 2020



https://www4.cs.fau.de/Lehre/WS20/V_BS

Überblick: Einordnung dieser VL



Betriebssystementwicklung



Agenda

Einordnung
Begriffe und Grundannahmen
Zustandssicherung
Zustandsänderungen
Zusammenfassung



Agenda

Einordnung

Begriffe und Grundannahmen

Interrupt, Exception, Trap

Grundannahmen

Zustandssicherung

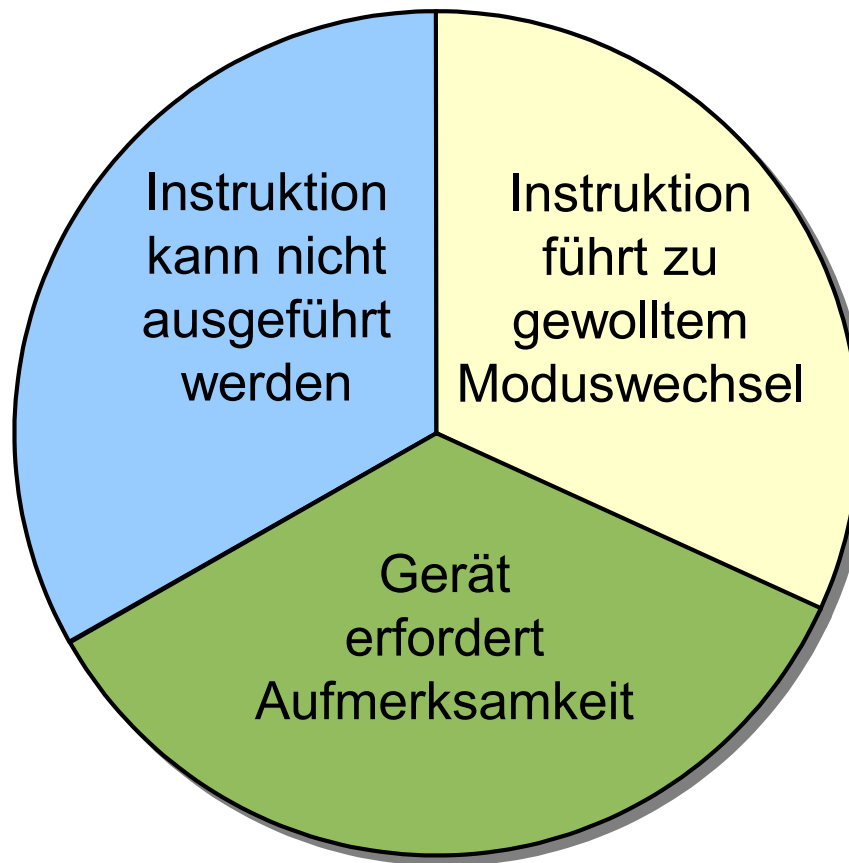
Zustandsänderungen

Zusammenfassung



- das Verständnis der Begriffe ist unterschiedlich ...
 - zwecks Klärung begeben wir uns auf die technische Ebene

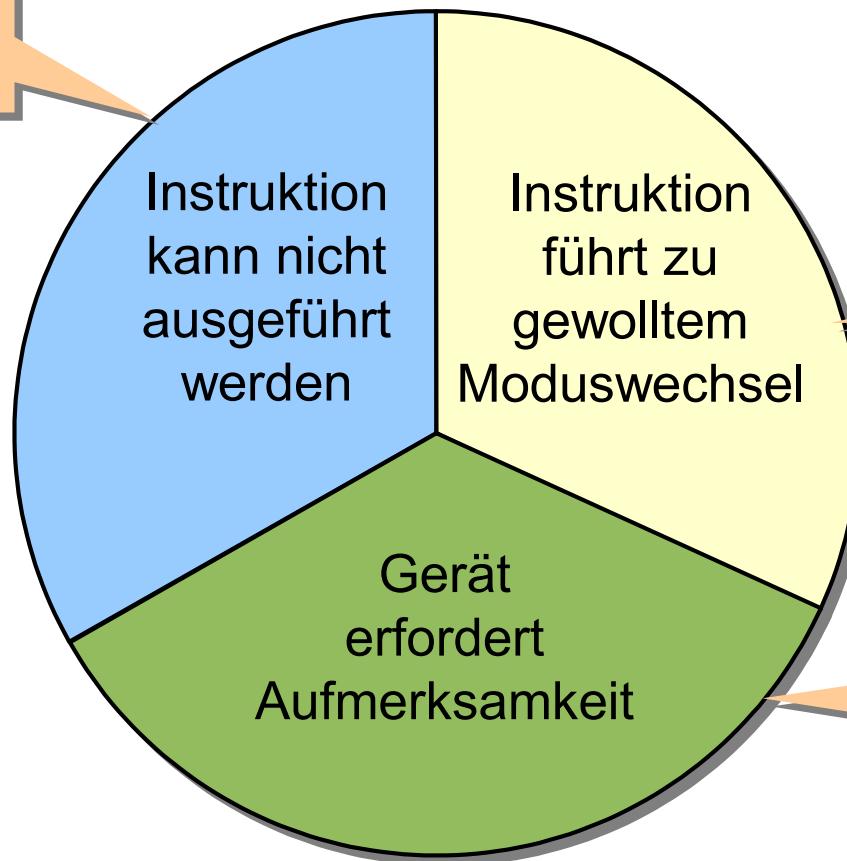
Grob können drei Fälle unterschieden werden



Begriffe

- das Verständnis der Begriffe ist unterschiedlich ...
 - zwecks Klärung begeben wir uns auf die technische Ebene

- Seitenfehler
- Schutzfehler
- Division durch 0



Grob können drei Fälle unterschieden werden

- Systemaufruf
- Haltepunktinstruktion

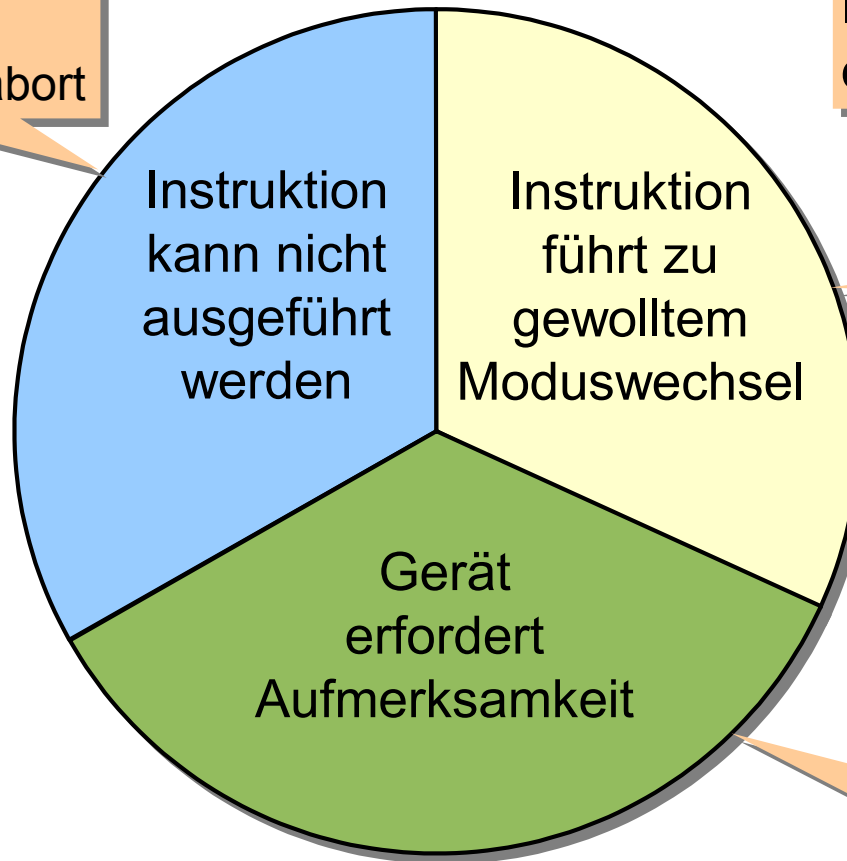
- Alarm durch Zeitgeber
- Taste gedrückt
- „NMI“



Begriffe: Intel IA-32

- das Verständnis der Begriffe ist unterschiedlich ...
 - zwecks Klärung begeben wir uns auf die technische Ebene

(software-generated)
exceptions
• fault, trap oder abort



Intel **IA-32-Architektur** (x86):
exceptions und **interrupts**

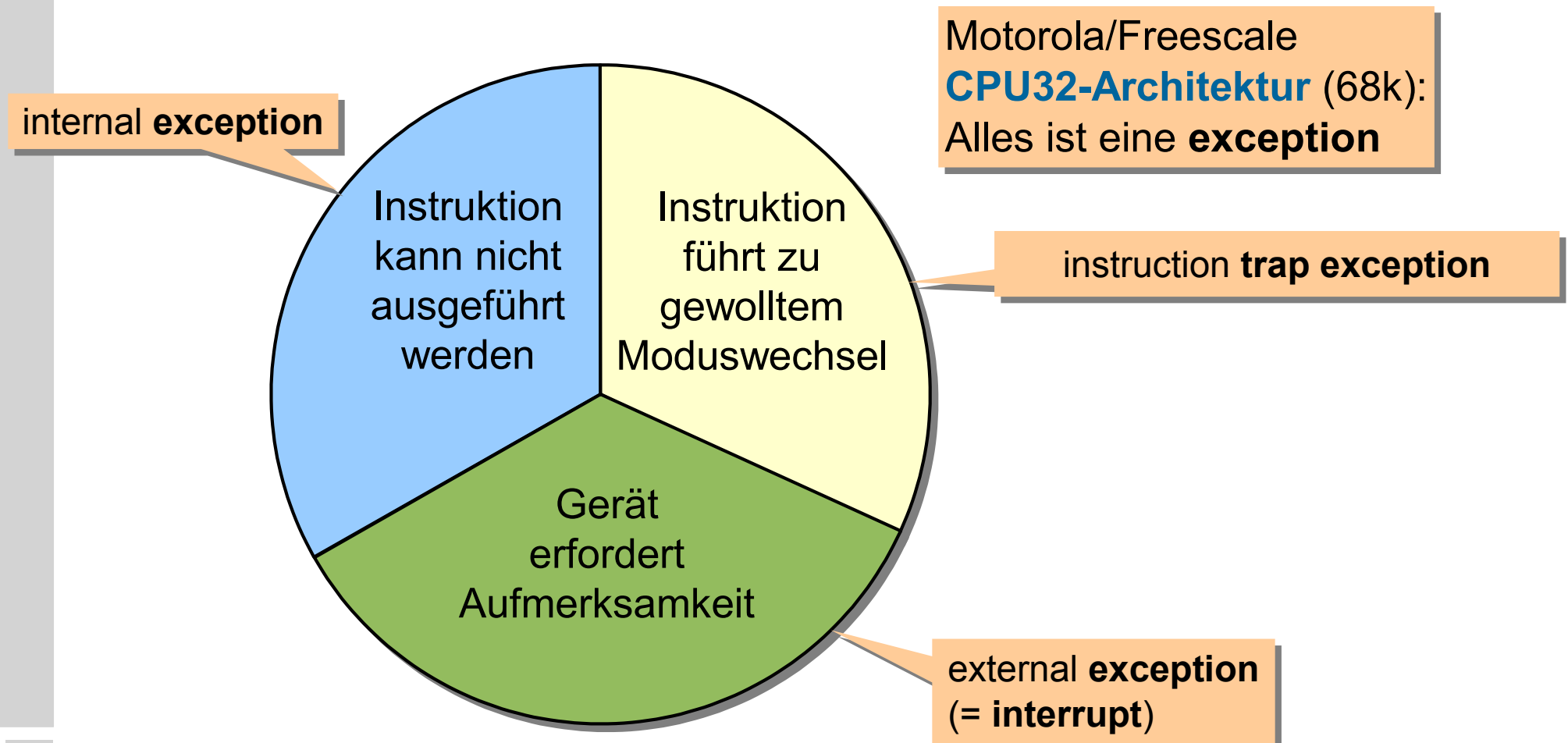
software(-generated) **interrupts**

external (hardware-generated) **interrupts**



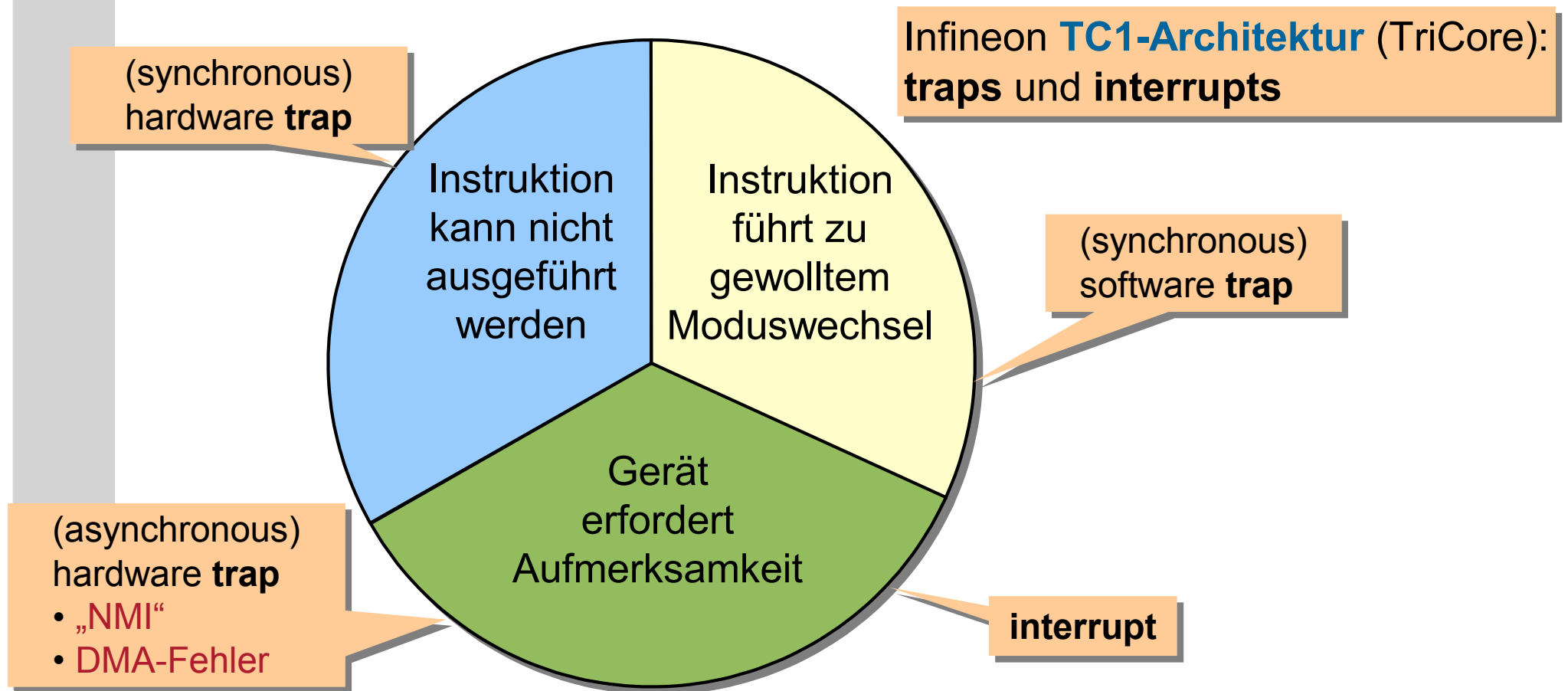
Begriffe: Motorola/Freescale CPU32

- das Verständnis der Begriffe ist unterschiedlich ...
 - zwecks Klärung begeben wir uns auf die technische Ebene



Begriffe: Infineon TC1

- das Verständnis der Begriffe ist unterschiedlich ...
 - zwecks Klärung begeben wir uns auf die technische Ebene

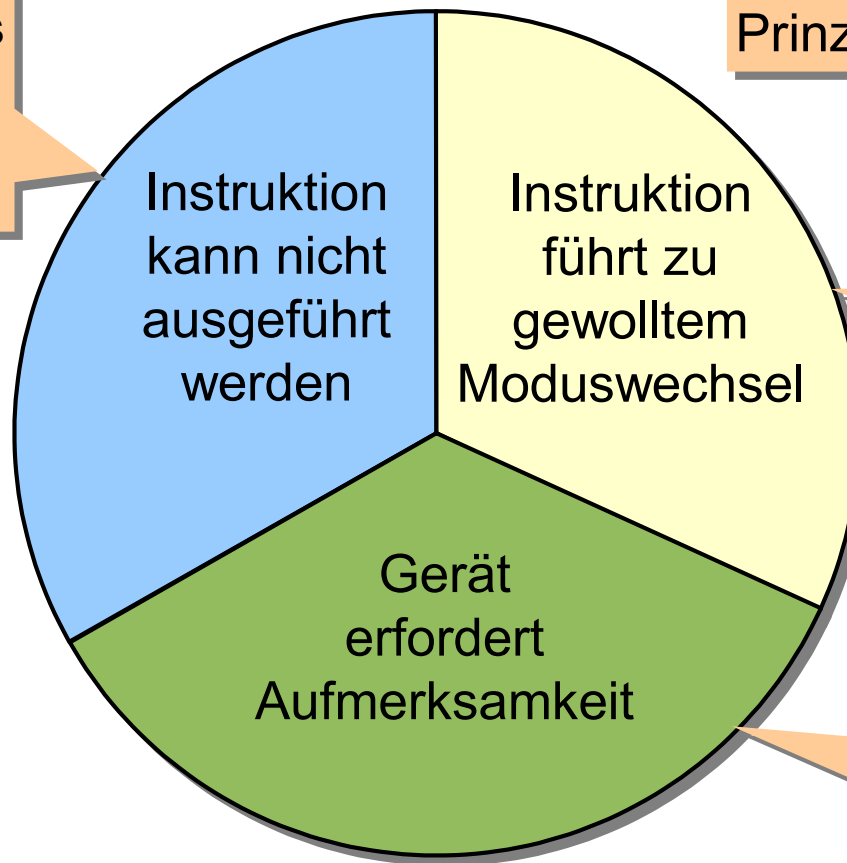


Begriffe: Literatur (Stallings)

- das Verständnis der Begriffe ist unterschiedlich ...
 - zwecks Klärung begeben wir uns auf die technische Ebene

Trap

Behandlung eines Fehlers oder eines Ausnahmezustandes



William Stallings: „Betriebssysteme: Prinzipien und Umsetzung“

Supervisor-Aufruf

Aufruf einer Betriebssystemfunktion

Interrupt

Reaktion auf ein externes asynchrones Ereignis

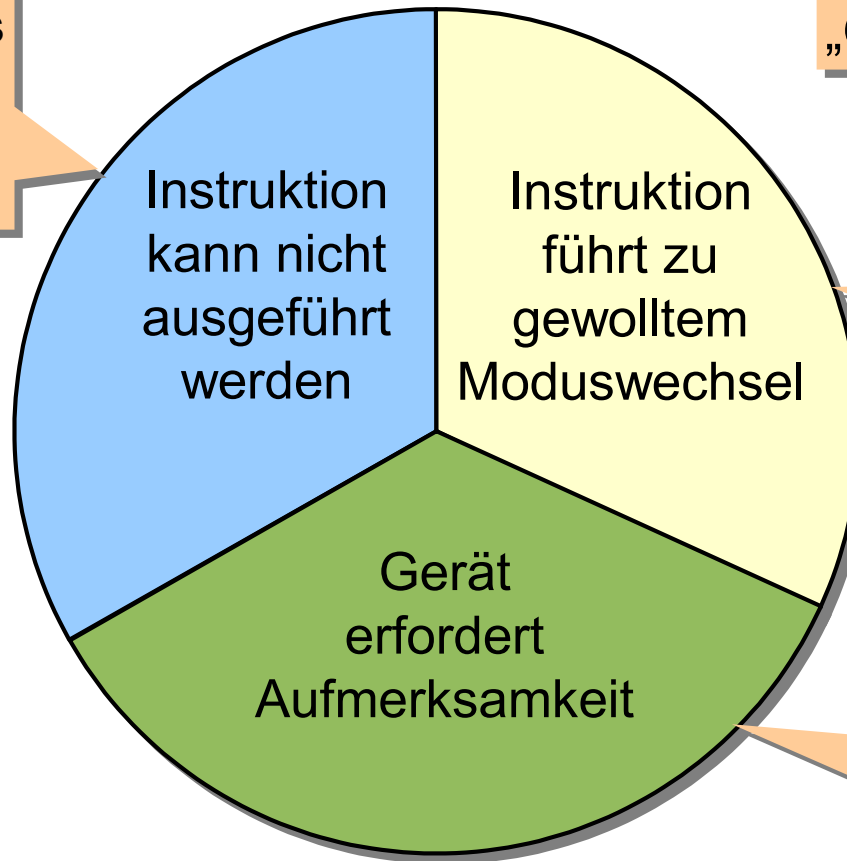


Begriffe: Literatur (Silberschatz)

- das Verständnis der Begriffe ist unterschiedlich ...
 - zwecks Klärung begeben wir uns auf die technische Ebene

exception

Behandlung eines Fehlers oder eines Ausnahmezustandes



Silberschatz et al :

„Operating System Concepts“

software interrupt / trap

Aufruf einer Betriebssystemfunktion

interrupt

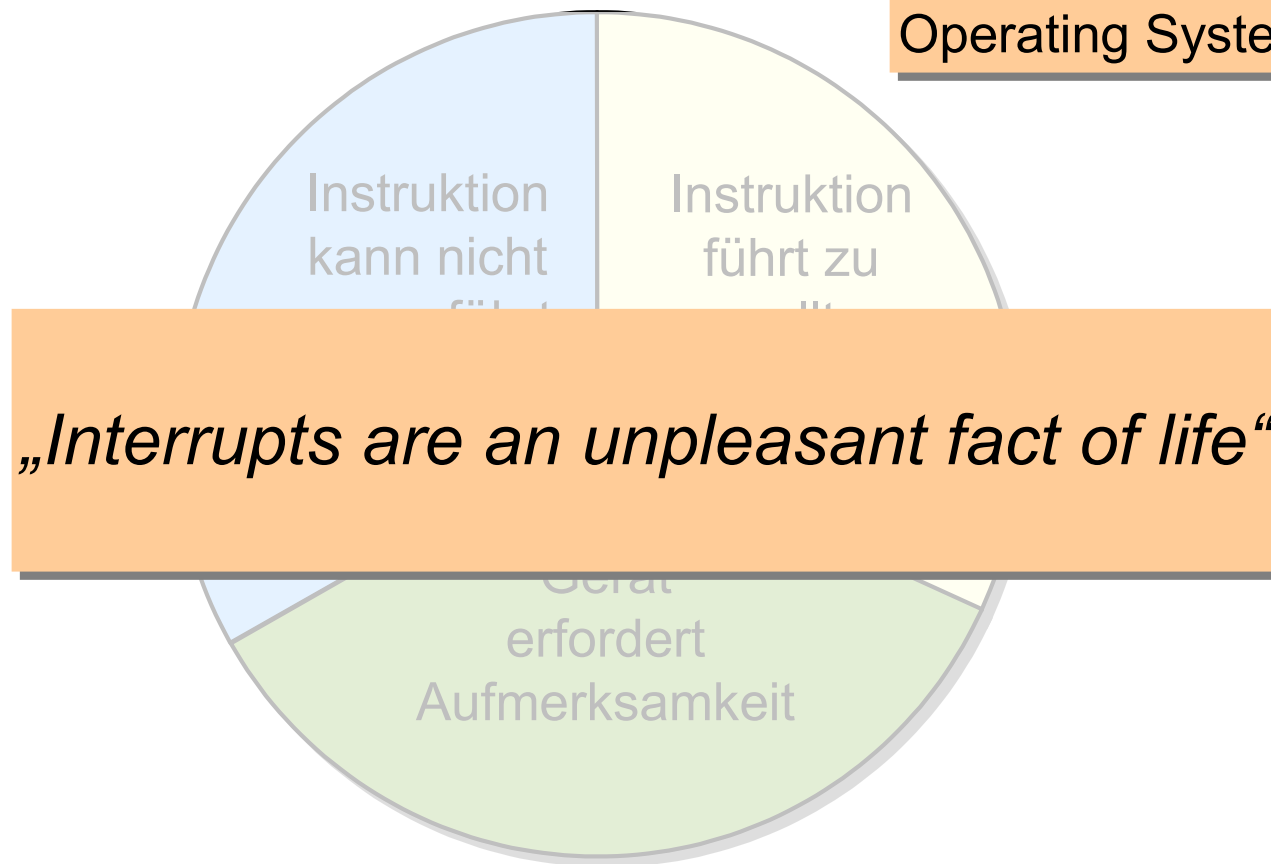
Reaktion auf ein externes asynchrones Ereignis



Begriffe: Literatur (Tanenbaum)

- das Verständnis der Begriffe ist unterschiedlich ...
 - zwecks Klärung begeben wir uns auf die technische Ebene

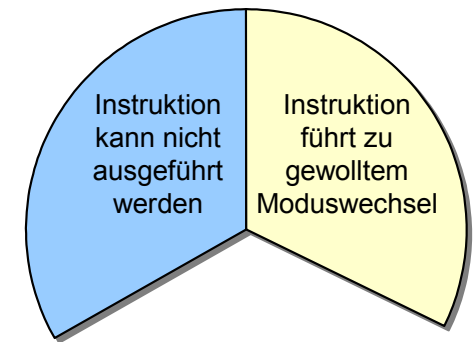
Andrew S. Tanenbaum: „Modern Operating Systems“ (ältere Ausgabe)



Begriffe: Unser Verständnis in BS

■ „Trap“

- durch **Instruktion** ausgelöst
 - auch die „trap“ oder „int“ Instruktion für Systemaufrufe
 - nicht definiertes Ergebnis (z.B. Division durch 0)
 - Hardware-Problem (z.B. Busfehler)
 - Betriebssystem muss eingreifen (z.B. Seitenfehler)
 - ungültige Instruktion (z.B. bei Programmfehler)
- Eigenschaften
 - oft vorhersagbar, oft reproduzierbar
 - Wiederaufnahme **oder Abbruch** der auslösenden Aktivität



■ „Unterbrechung“ (engl. *Interrupt*):

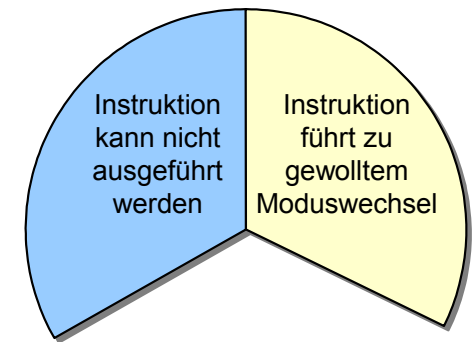
- durch **Hardware** ausgelöst
 - Hardware verlangt die Aufmerksamkeit der Software (Zeitgeber, Tastatursteuereinheit, Festplattensteuereinheit, ...)
- Eigenschaften:
 - nicht vorhersagbar, nicht reproduzierbar
 - in der Regel Wiederaufnahme der unterbrochenen Aktivität



Begriffe: Unser Verständnis in BS

■ „Trap“

- durch **Instruktion** ausgelöst
 - auch die „trap“ oder „int“ Instruktion für Systemaufrufe
 - nicht definiertes Ergebnis (z.B. Division durch 0)
 - Hardware-Problem (z.B. Busfehler)
 - Betriebssystem muss eingreifen (z.B. Seitenfehler)
 - ungültige Instruktion (z.B. bei Programmfehler)
- Eigenschaften
 - oft vorhersagbar, oft reproduzierbar
 - Wiederaufnahme **oder Abbruch** der auslösenden Aktivität



■ „Unterbrechung“ (engl. *Interrupt*):

- durch **Hardware** ausgelöst
 - Hardware verlangt die Aufmerksamkeit der Software (Zeitgeber, Tastatursteuereinheit, Festplattensteuereinheit, ...)
- Eigenschaften:
 - nicht vorhersagbar, nicht reproduzierbar
 - in der Regel Wiederaufnahme der unterbrochenen Aktivität



Wir betrachten die Behandlung von Unterbrechungen unter den folgenden Grundannahmen

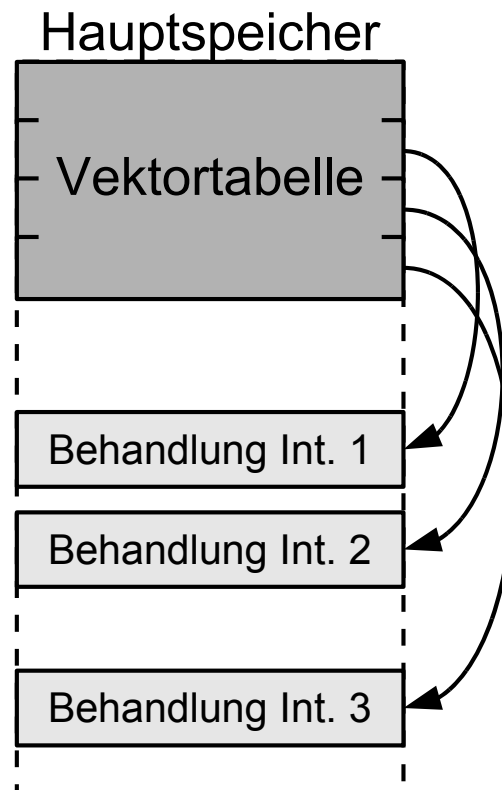
1. Die CPU startet die Behandlungsroutine automatisch.
2. Die Unterbrechnungsbehandlung erfolgt im Systemmodus.
3. Das unterbrochene Programm kann fortgesetzt werden.
4. Die Maschineninstruktionen verhalten sich atomar.
5. Die Unterbrechnungsbehandlung kann unterdrückt werden.



Grundannahmen: Behandlungsroutine

1. Die CPU startet die Behandlungsroutine automatisch.

- erfordert die Zuordnung einer Behandlungsroutine
- Ermittlung der Unterbrechungsursache nötig



Varianten:

- Register enthält Startadresse der Vektortabelle
- Tabelleneinträge enthalten Code
- Programmierbarer „Event Controller“ behandelt die Unterbrechung in Hardware
- Tabelle enthält Deskriptoren
- Behandlungsroutine hat eigenen Prozesskontext



2. Die Unterbrechungsbehandlung erfolgt im Systemmodus.

- Unterbrechungen sind der einzige Mechanismus, um nicht-kooperativen Anwendungen die CPU zu entziehen
- nur das BS darf uneingeschränkt auf Geräte zugreifen
- die CPU schaltet daher vor der Unterbrechungsbehandlung in den privilegierten Systemmodus

Varianten:

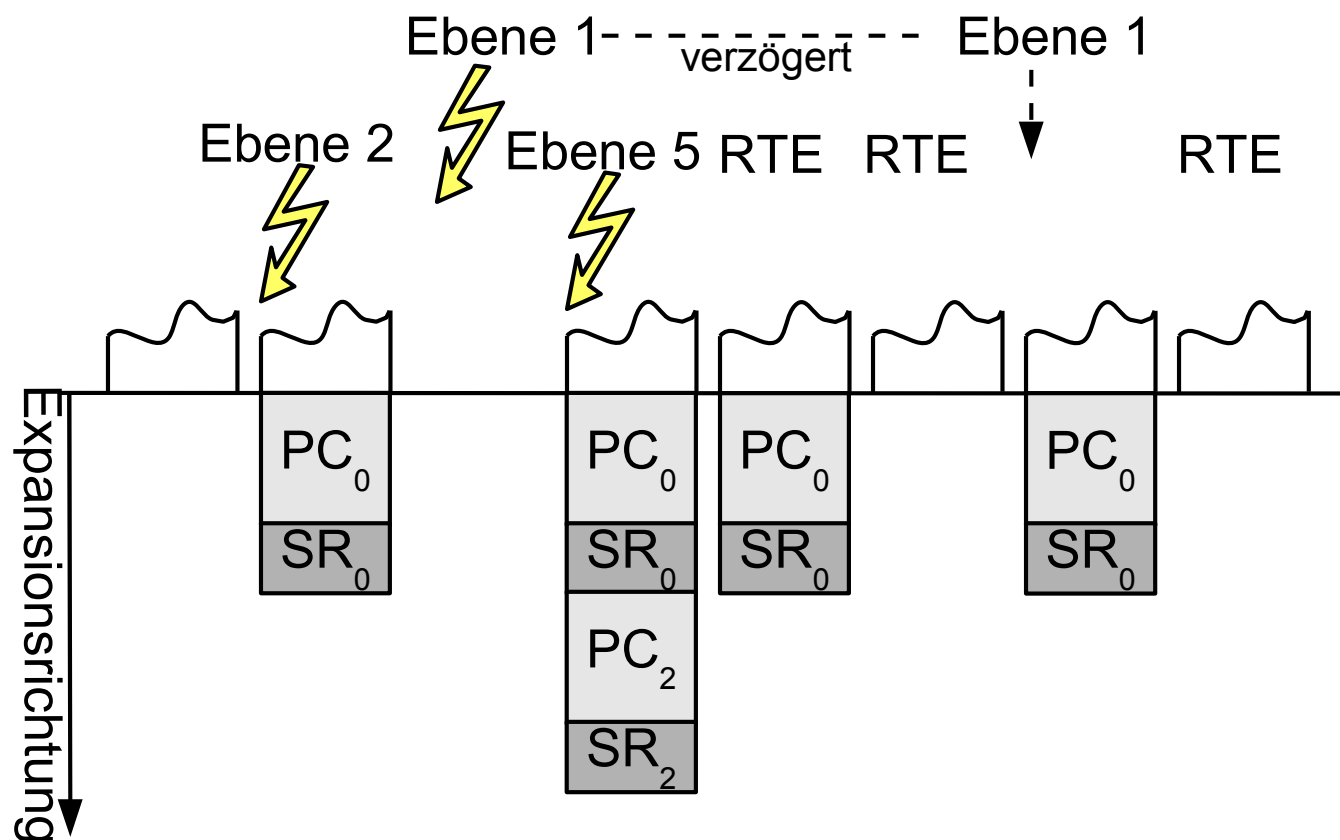
- bei **16-Bit-CPUs** ist eine Aufteilung in Benutzer-/Systemmodus eher die Ausnahme
- bei **8-Bit-CPUs** (oder kleiner) gibt es diese Aufteilung nicht



Grundannahmen: Kontextsicherung

3. Das unterbrochene Programm kann fortgesetzt werden.

- notwendiger Zustand wird automatisch gesichert
- ggf. auch geschachtelt, erfordert Stapel



Varianten:

- weitere Informationen über die Ursache auf dem Stapel
- keine Prioritäten
- spezieller „Interrupt-Stack“
- Kontextsicherung in Registern



4. Die Maschineninstruktionen verhalten sich atomar.

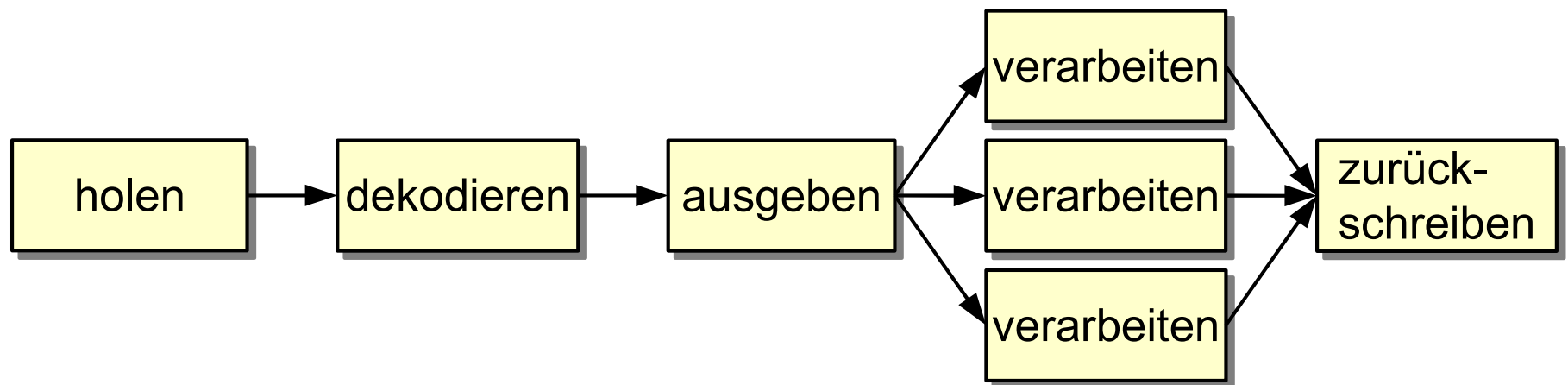
- definierter CPU-Zustand zu Beginn der Behandlungsroutine
- Wiederherstellbarkeit des Zustands
- trivial bei CPUs mit klassischem von-Neumann-Zyklus
- nicht-trivial bei modernen CPUs:
 - Fließbandverarbeitung: Befehle müssen annulliert werden
 - Superskalare CPUs: zusätzlich Befehlsreihenfolge merken



Grundannahmen: Atomares Verhalten

4. Die Maschineninstruktionen verhalten sich atomar.

Befehlsverarbeitung bei superskalaren Prozessoren:
(stark vereinfacht!)



Im Idealfall werden alle Stufen immer benutzt, d.h. mehrere Befehle werden parallel ausgeführt. Wann soll geprüft werden, ob eine Unterbrechungsanforderung anliegt?



4. Die Maschineninstruktionen verhalten sich atomar.

Trotz der Schwierigkeiten liefern die meisten CPUs
„**präzise Unterbrechungen**“:

- *„All instructions preceding the instruction indicated by the saved program counter have been executed and have modified the process state correctly.“*
- *„All instructions following the instruction indicated by the saved program counter are unexecuted and have not modified the process state.“*
- *„If the interrupt is caused by an exception condition raised by an instruction in the program, the saved program counter points to the interrupted instruction. The interrupted instruction may or may not have been executed, depending on the definition of the architecture and the cause of the interrupt. Whichever is the case, the interrupted instruction has either completed, or has not started execution.“*

J. E. Smith and A. R. Pleszkun,
„Implementing Precise Interrupts in Pipelined Processors“,
IEEE Transactions on Computers, Vol. 37, No. 5, 1988



5. Die Unterbrechungsbehandlung kann unterdrückt werden.

- Beispiele:
 - Motorola 680x0: entsprechend der Priorität
 - Intel x86: global mit `sti, cli`
 - *Interrupt Controller*: jede Quelle einzeln
- automatische Unterdrückung erfolgt auch durch die CPU vor Betreten der Behandlungsroutine



5. Die Unterbrechungsbehandlung kann unterdrückt werden.

- automatische Unterdrückung erfolgt auch durch die CPU vor Betreten der Behandlungsroutine
 - Unterbrechungen nicht vorhersagbar, theoretisch beliebig häufig
 - Stapelüberlauf könnte nicht ausgeschlossen werden
- unterdrückt wird (durch die Hardware) die Behandlung...
 - pauschal aller Unterbrechungen (sehr restriktiv)
 - Unterbrechungen niedriger oder gleicher Priorität (weniger restriktiv)
 - bestimmte Geräte werden bevorzugt
- bessere Modelle mit Hilfe von Software (z.B. in Linux):
 - Unterbrechungen, die bereits behandelt werden, werden unterdrückt
 - hohe Reaktivität ohne Bevorzugung einzelner Geräte

