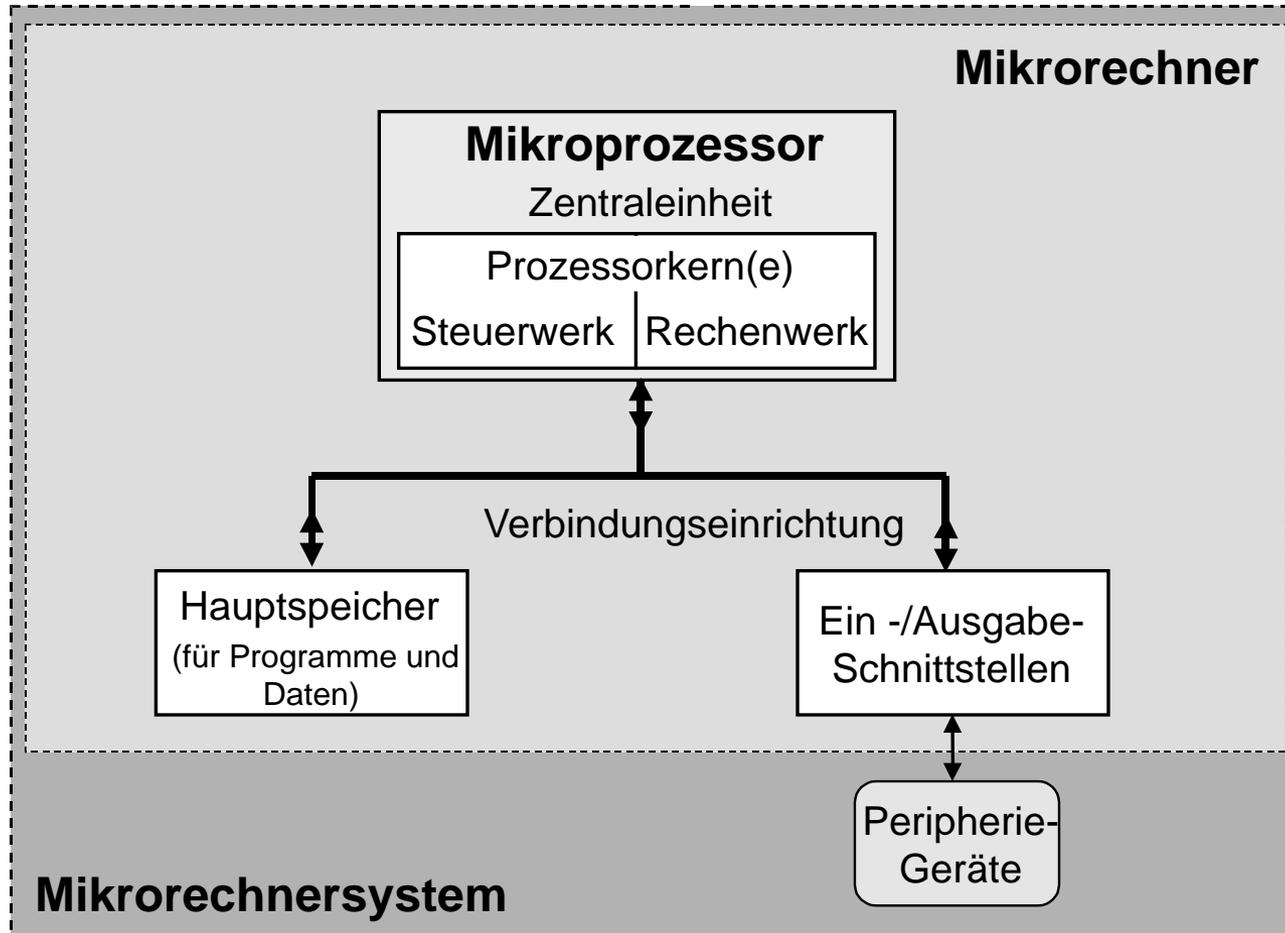


Kapitel 3

Rechnerarchitekturen für Echtzeitsysteme

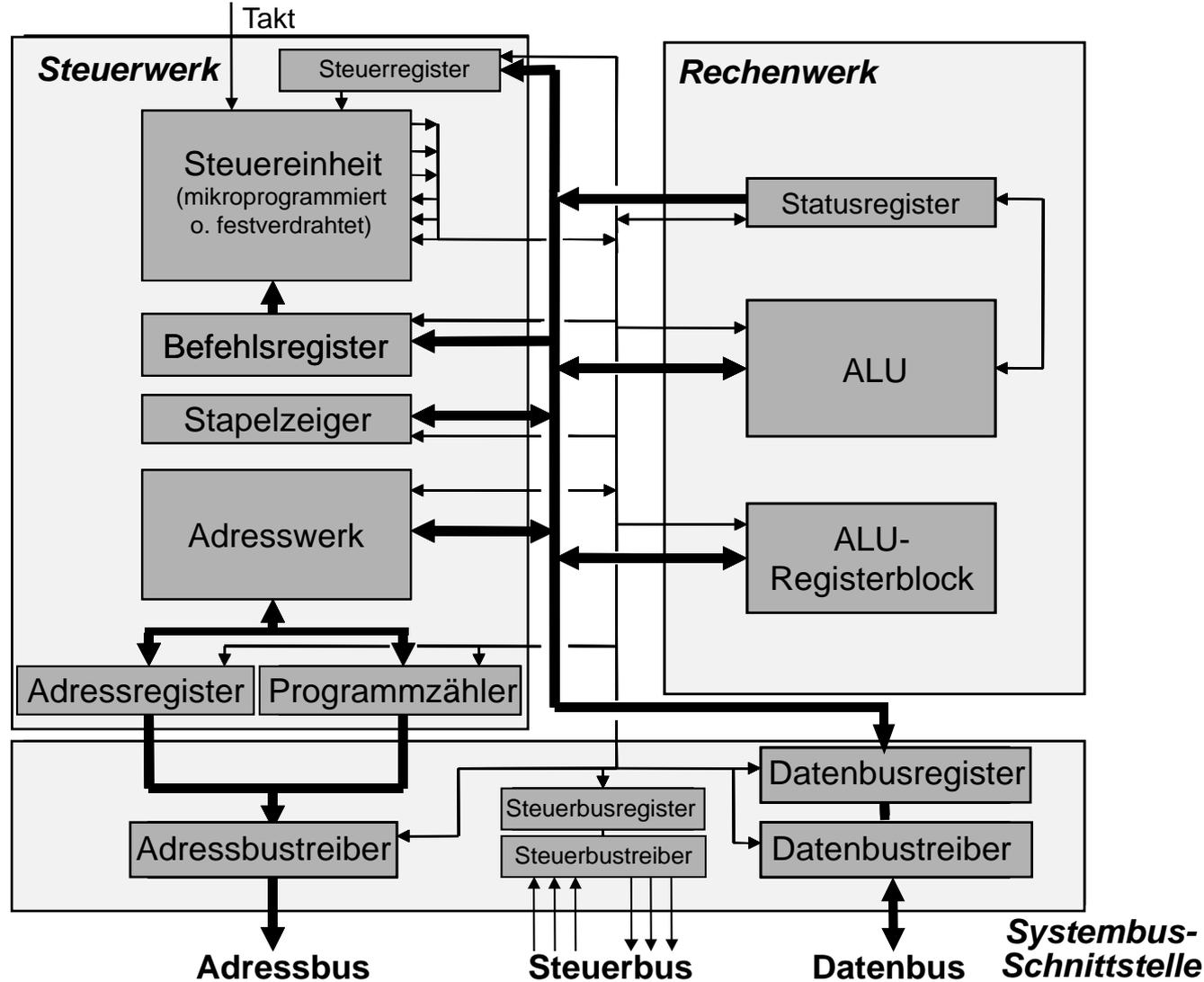
Abgrenzung der Begriffe Mikroprozessor, Mikrorechner und Mikrorechnersystem



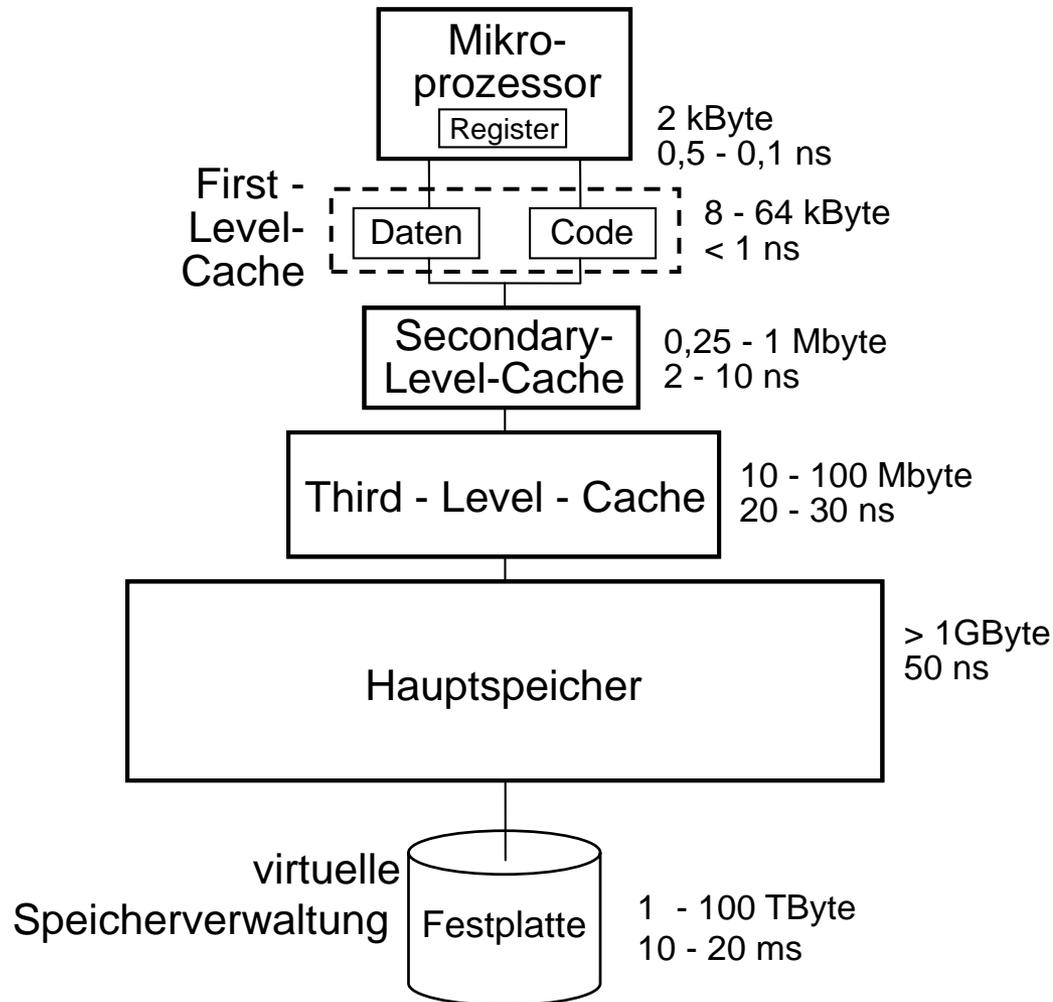
Kosten spielen eine wichtige Rolle

⇒ Eingesetzt werden nicht die schnellsten und neuesten Prozessoren, sondern diejenigen, die die **gestellte Aufgabe mit minimalen Kosten erledigen**

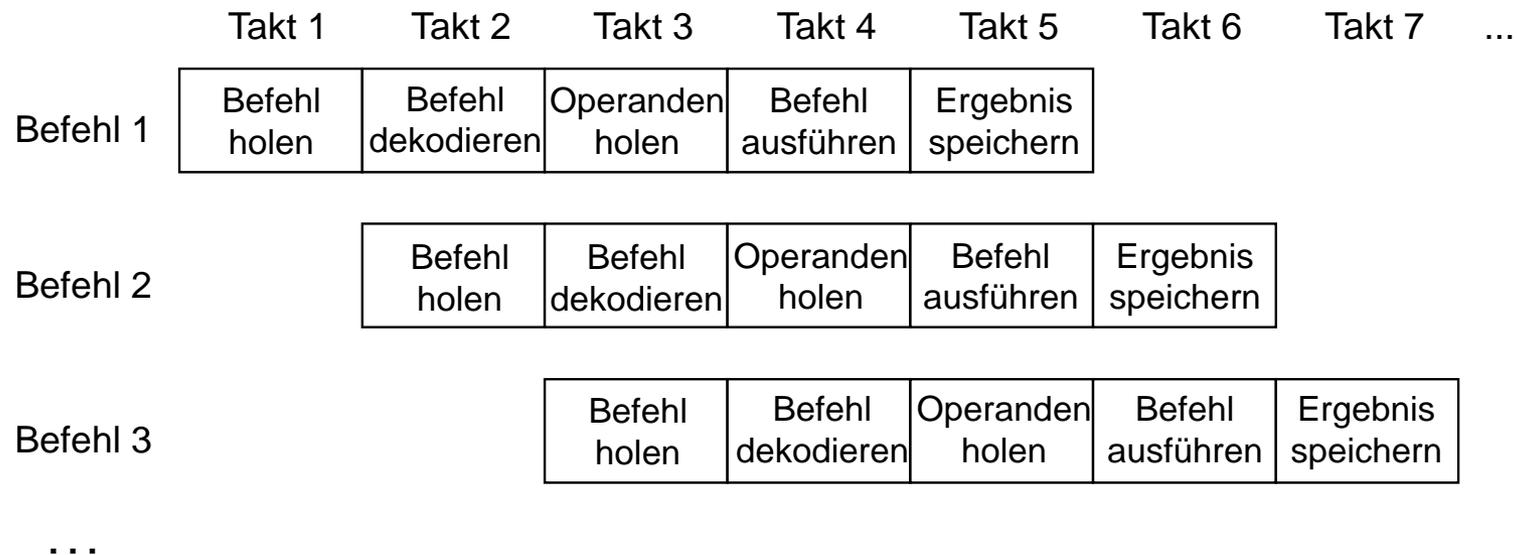
Ein einfacher Mikroprozessor



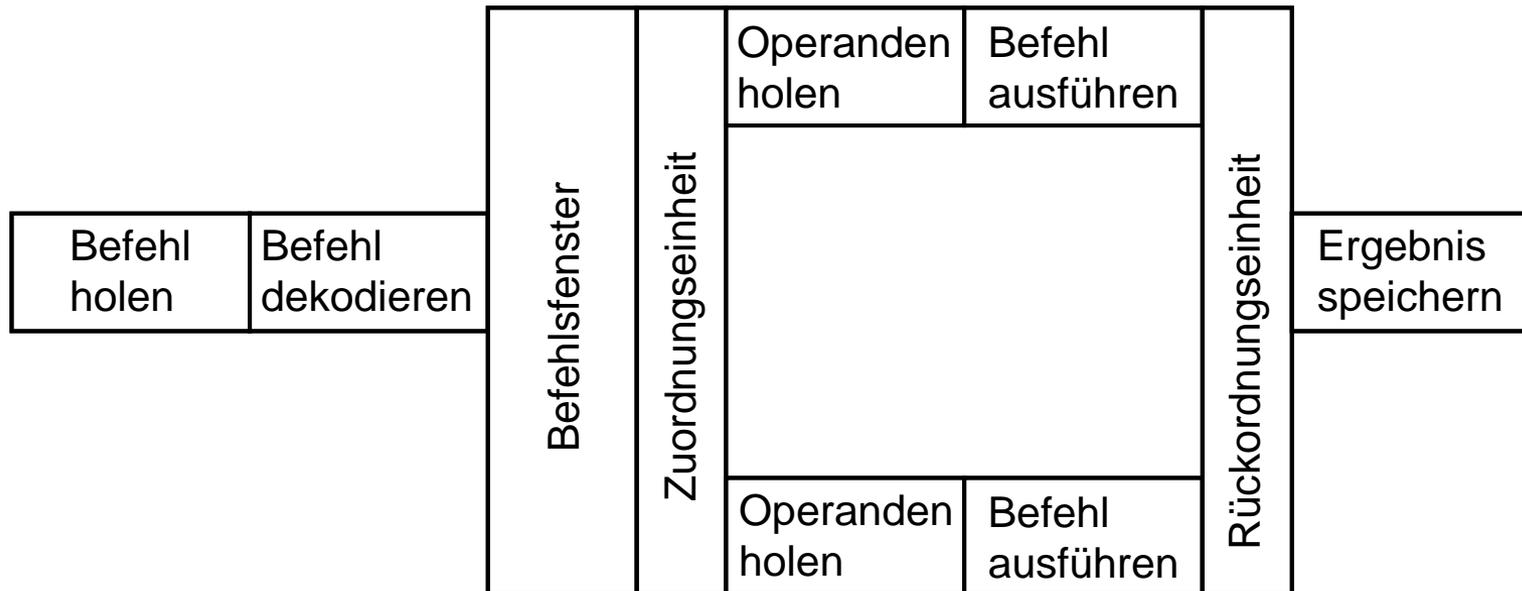
Mehrstufige Speicherhierarchie bei heutigen Mikroprozessoren



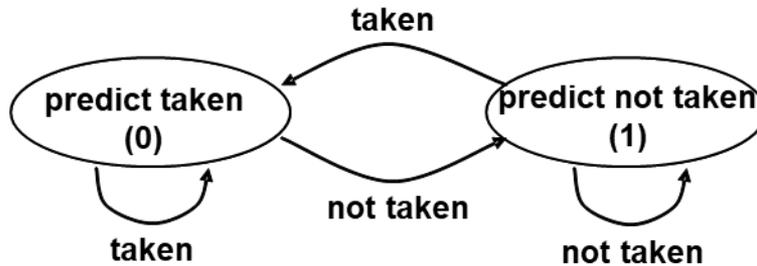
Eine fünfstufige Befehlspipeline



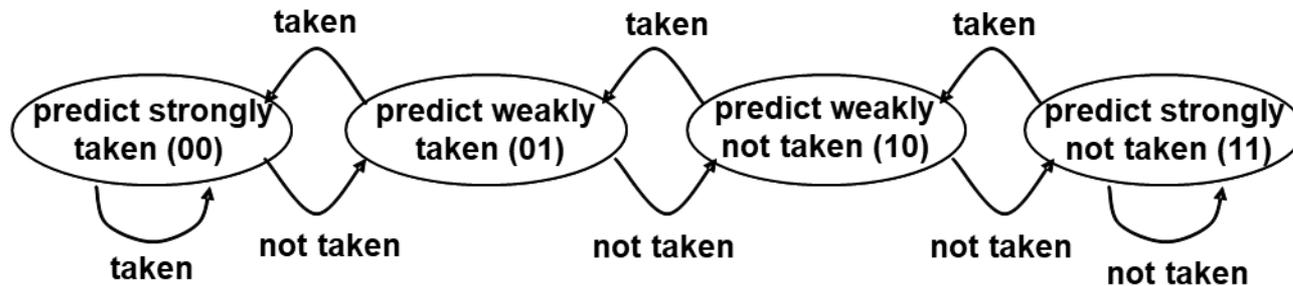
Eine superskalare Pipeline



Ein- und Zwei-Bit-Prädiktoren zur Sprungvorhersage



a) Ein – Bit - Präditkor



b) Zwei – Bit - Präditkor

Best- und Worst-Case Ausführungszeiten verschiedener Mikroarchitekturen

Programm	Benötigte Anzahl Taktzyklen							
	Keine Pipeline		Einfache Pipeline		Pipeline, Spekulation		Pipeline, Cache, Spekulation	
	Best=Worst	Best=Worst	Best	Worst	Best	Worst		
LD A,(2000)	5	1+4	1+4	1+4	1	1+4		
INC B	3	1	1	1	1	1		
SUB A,B	3	1+1	1+1	1+1	1+1	1+1		
JZ 10000	4	1+3	1	7	1	7		
<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>	<hr/>		
Gesamt	15	12	9	15	5	15		

Unterbrechung und Priorisierung

- Wie sehen Unterbrechungen aus?
- Welche Signale können Unterbrechungen auslösen?
- Wie können Unterbrechungen priorisiert werden?

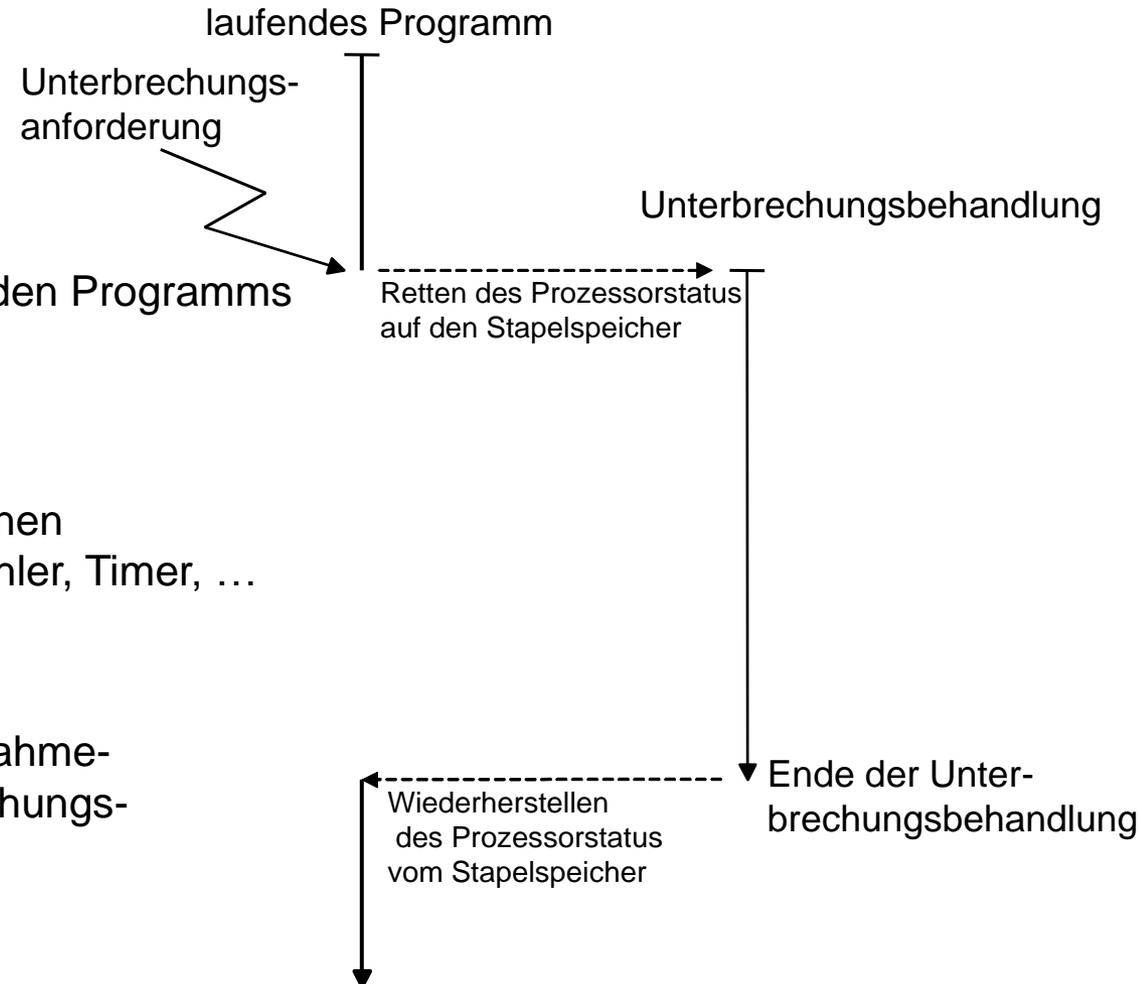
Unterbrechungsbehandlung

je nach Priorität des auslösenden Ereignisses erfolgt eine vorübergehende Unterbrechung des normalen Befehlsablaufs

Mögliche auslösende Ereignisse:

- Fehlersituationen \Rightarrow *Exception*
- Unterbrechungswunsch des laufenden Programms (z.B. wegen Fehlersituation oder Betriebssystemaufruf) \Rightarrow *Software Interrupt*
- Unterbrechungswunsch einer externen Systemkomponenten: E/A, D/A, Zähler, Timer, ... \Rightarrow *Hardware Interrupt*

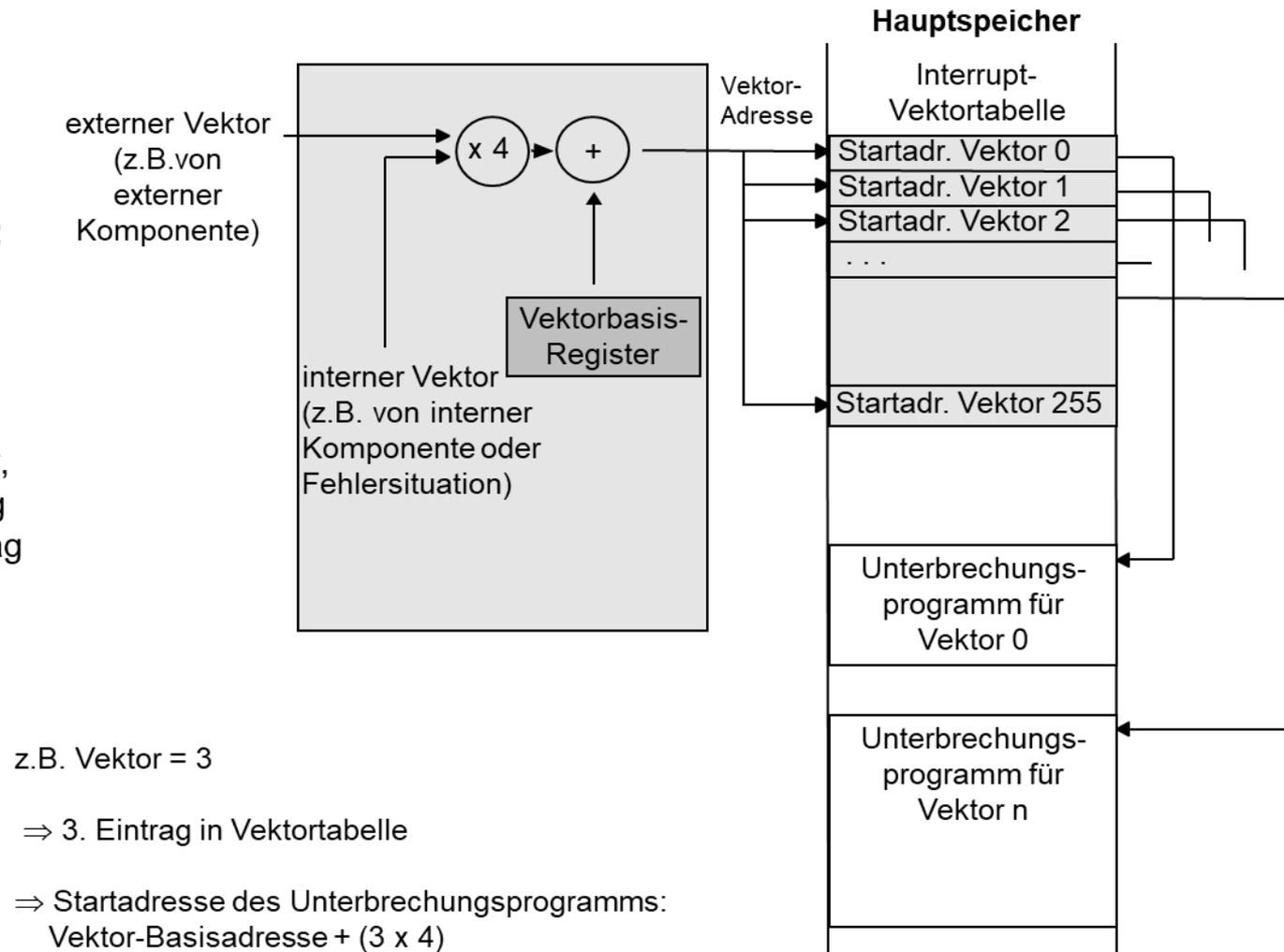
Die Behandlung einer solchen Ausnahme-Situation erfolgt durch ein Unterbrechungs-Programm \Rightarrow *Interrupt Service Routine*



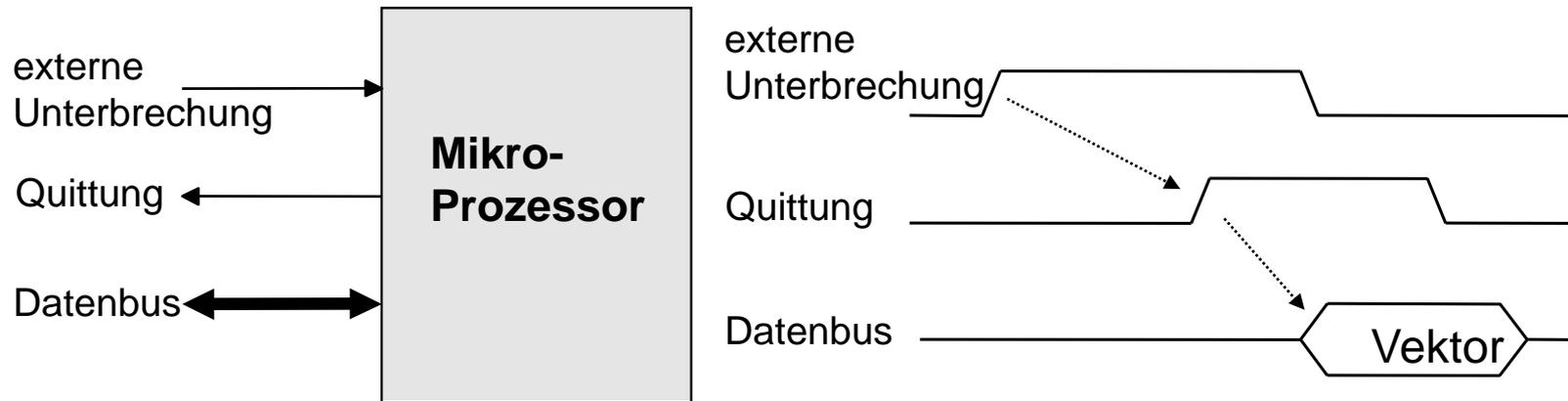
Ermittlung der Startadresse eines Unterbrechungsprogramms aus dem Interrupt-Vektor

Interrupt-Vektor-Tabelle:
enthält die Startadressen
der Behandlungsroutinen

Interrupt-Quelle liefert bei
Unterbrechung eine
Interrupt-Vektornummer,
welche die Unterbrechung
identifiziert und den Eintrag
in der Interrupt-Vektor-
Tabelle charakterisiert.



Auslösen einer externen Unterbrechung



Einführung von *Prioritäts-Ebenen*:

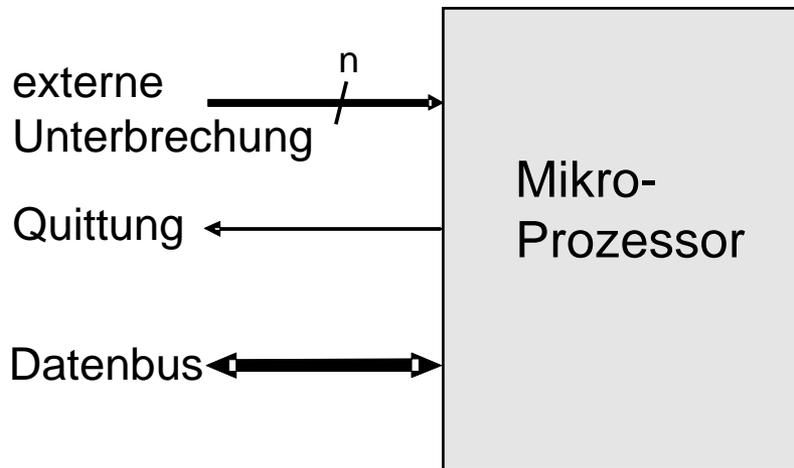
- **direkt im Mikroprozessor**

Jedes laufende Programm erhält eine Prioritätsebene, die im Interrupt-Maskenregister (Bestandteil des Steuerregisters) gespeichert wird.

Nur Interrupts höherer Priorität können dieses Programm unterbrechen (Interrupt höchster Priorität immer).

Eine Interrupt-Service-Routine erhält die Priorität des zugehörigen Interrupts

Interne Realisierung von Prioritäten durch einen Unterbrechungsbus



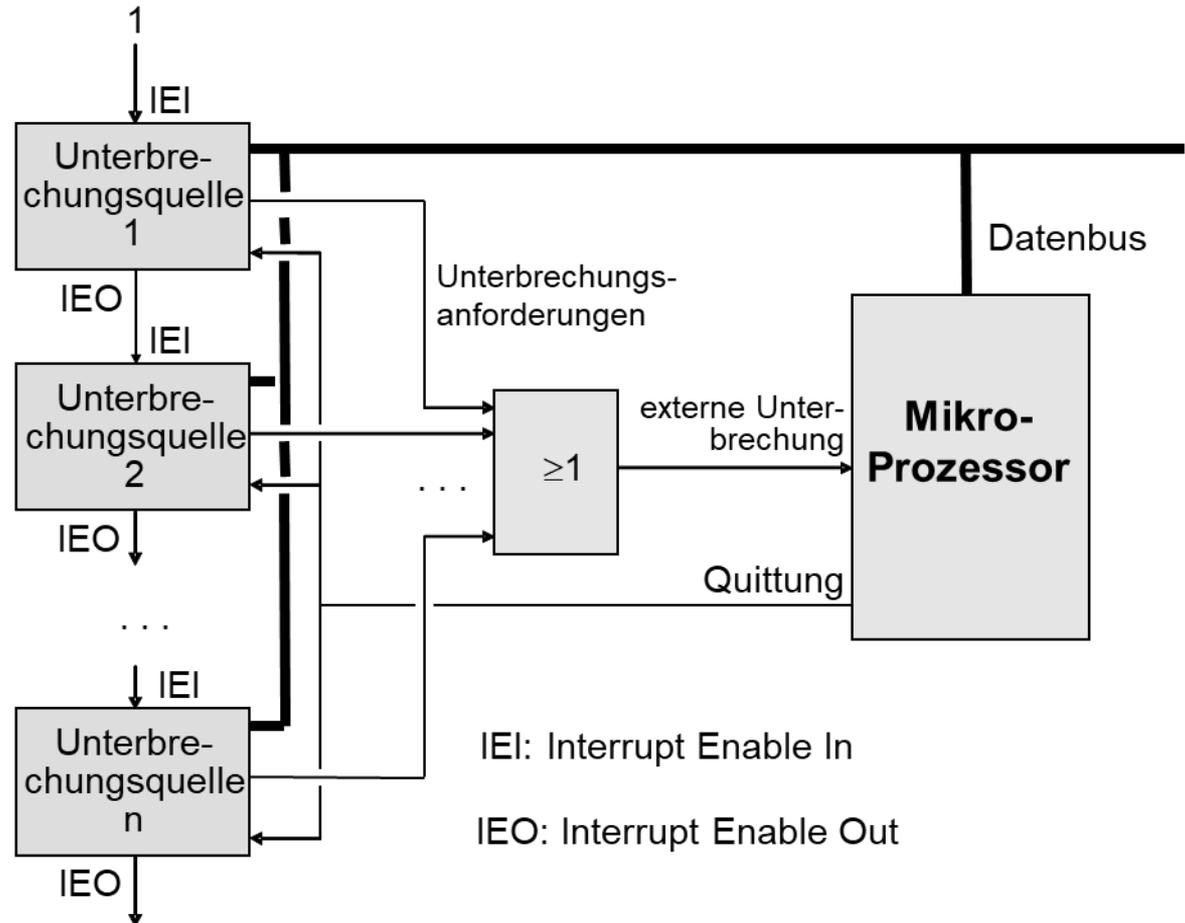
n Unterbrechungs-Leitungen
z.B. n = 3

Code	Bedeutung
0 0 0	keine Unterbrechung
0 0 1	Unterbrechung Priorität 1
0 1 0	Unterbrechung Priorität 2
0 1 1	Unterbrechung Priorität 3
	...
1 1 1	Unterbrechung Priorität 7

Dezentrale externe Prioritätensteuerung mit Daisy Chain

Werden mehr
Prioritätsebenen
benötigt, so muss dies
durch externe Kompo-
nenten realisiert werden:

a) dezentral mittels Daisy – Chain



b) zentral mittels Interrupt - Controller

