

Betriebssysteme (BS)

VL 3.3 – Unterbrechungen, Hardware – x86

Volkmar Sieh / Daniel Lohmann

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme

Friedrich-Alexander-Universität
Erlangen Nürnberg

WS 21 – 8. November 2021



https://www4.cs.fau.de/Lehre/WS21/V_BS

Unterbrechungen bei x86 CPUs



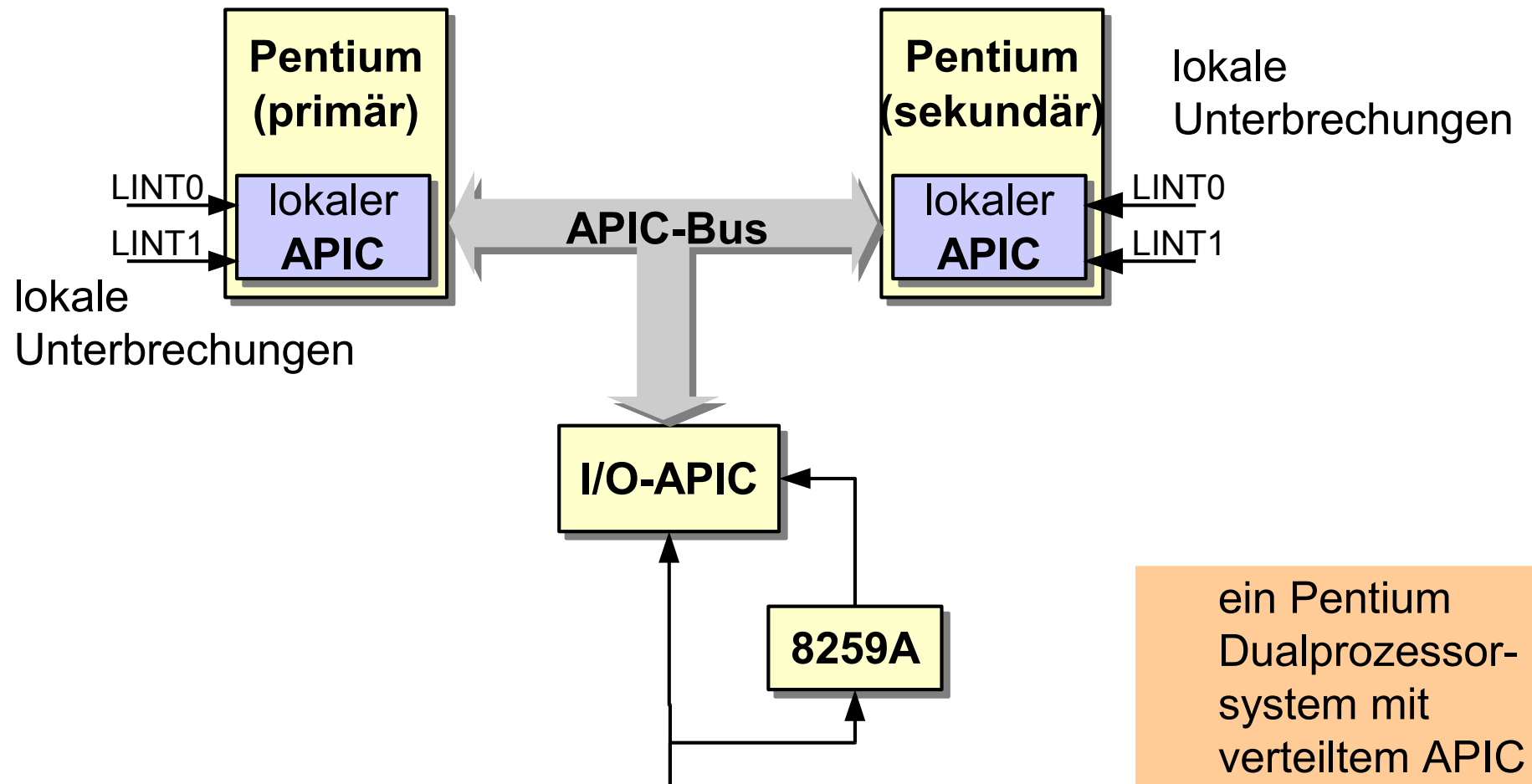
Unterbrechungen bei x86 CPUs

- bis einschließlich i486 hatten x86 CPUs nur einen IRQ und einen NMI Eingang
- externe Hardware sorgte für die Priorisierung und Vektornummerngenerierung
 - durch einen Chip namens **PIC 8259A**
 - 8 Interrupt-Eingänge
 - 15 Eingänge bei Kaskadierung von zwei PICs
 - keine Multiprozessorunterstützung
- heutige x86 CPUs enthalten den weit leistungsfähigeren **„Advanced Programmable Interrupt Controller“ (APIC)**
 - notwendig für **Multiprozessorsysteme**
 - inzwischen aber auch in allen Einprozessorsystemen aktiv
 - natürlich gibt es den PIC 8259A noch immer 😊



Die APIC Architektur

- ein APIC *Interrupt*-System besteht aus lokalen APICs auf jeder CPU und einem I/O APIC



Unterbrechungsanforderungen



Der I/O APIC

- heute typischerweise in der *Southbridge* von PC Chipsätzen integriert
- normalerweise 24 *Interrupt*-Eingänge
 - zyklische Abfrage (Round-Robin Priorisierung)
- für jeden Eingang gibt es einen 64 Bit Eintrag in der ***Interrupt Redirection Table***
 - beschreibt das Unterbrechungssignal
 - dient der Generierung der APIC-Bus Nachricht



Aufbau (Bits) eines Eintrags in der *Interrupt Redirection Table*

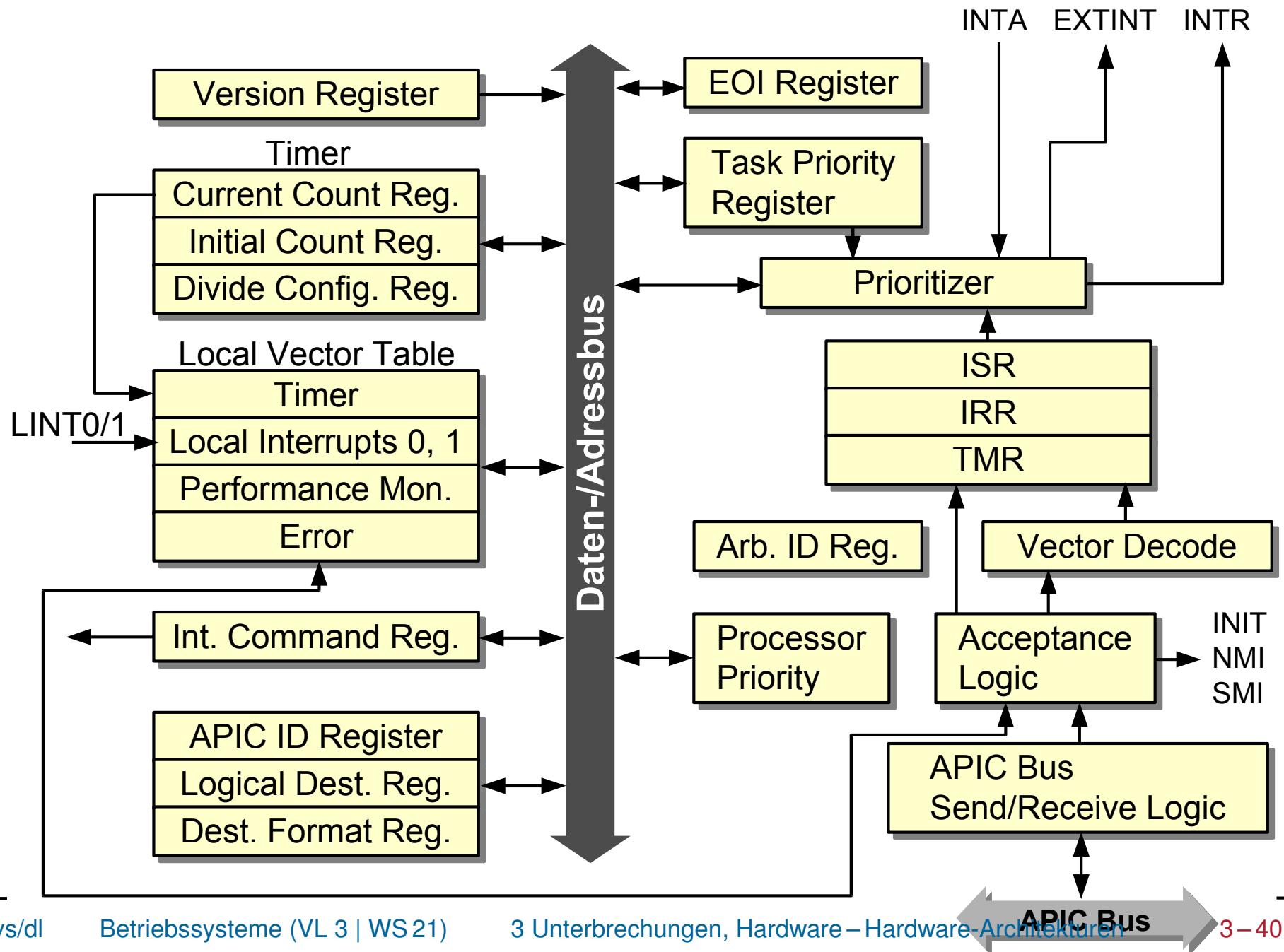
63:56	Destination Field je nach Bit 11:	– R/W. 8 Bit Zieladresse. APIC ID der CPU (<i>Physical Mode</i>) oder CPU Gruppe (<i>Logical Mode</i>)
55:17	<reserviert>	
16	Interrupt-Mask	– R/W. Unterbrechungssperre.
15	Trigger Mode	– R/W. <i>Edge-</i> oder <i>Level-Triggered</i>
14	Remote IRR	– RO. Art der erhaltenen Bestätigung
13	Interrupt Pin Polarity	– R/W. Signalpolarität
12	Delivery Status	– RO. Interrupt-Nachricht unterwegs?
11	Destination Mode	– R/W. <i>Logical Mode</i> oder <i>Physical Mode</i>
10:8	Delivery Mode	– R/W. Wirkung bei Ziel-APIC
	000 – <i>Fixed</i> :	Signal an alle Zielprozessoren ausliefern
	001 – <i>Lowest Priority</i> :	Liefern an CPU mit aktuell niedrigster Prio.
	010 – SMI:	<i>System Management Interrupt</i>
	100 – NMI:	<i>Non-Maskable Interrupt</i>
	101 – INIT:	Ziel-CPU's initialisieren (Reset)
	111 – ExtINT:	Antwort an PIC 8259A
7:0	Interrupt Vector	– R/W. 8 Bit Vektornummer (16 – 254)

Local APICs

- empfangen Unterbrechungsanforderungen vom APIC Bus
- führen die Auswahl und Priorisierung durch
- können zwei lokale Unterbrechungen direkt verarbeiten
- enthalten weitere Funktionseinheiten
 - Eingebauten *Timer, Performance Counter*
 - *Command-Register*
 - um selber APIC-Nachrichten zu verschicken
 - insbesondere Inter-Prozessor-Interrupt (IPI)
- programmierbar über 32 Bit Register ab 0xfee00000
 - memory mapped (ohne externe Buszyklen)
 - jede CPU programmiert „ihren“ *Local APIC*



Local APICs - Register



APIC Architektur - Zusammenfassung

- flexible Verteilung an CPUs im x86 Multiprozessorsystem
 - fest, Gruppen, an die CPU mit der geringsten Priorität
 - Liegen mehrere IRQs an, so wird nach Vektornummer priorisiert
- Vektornummer 16-254 können frei zugeordnet werden
 - sollte (an sich) reichen, um „sharing“ zu vermeiden
- *Local* APIC erwartet explizites EOI
 - dafür muss die Software sorgen
- Mit APIC unterstützt x86 prinzipiell auch Prioritätsebenen
 - Systemsoftware muss jedoch entsprechend agieren
(Unterbrechungen freigeben, evtl. *Task-Priority-Register* verwenden)



Agenda

Einordnung
Grundlagen
Hardware-Architekturen
Zusammenfassung



Zusammenfassung und Ausblick

- Unterbrechungsbehandlungshardware befasst sich mit ...
 - Priorisierung
 - Zuordnung und Ausführung einer Behandlungsroutine
 - Zustandssicherung und geschachtelter Ausführung
- moderne Unterbrechungsbehandlungshardware kann ...
 - Unterbrechungsvektoren frei zuordnen
 - „*sharing*“ von Unterbrechungsvektoren vermeiden
 - Unterbrechungen im Multiprozessorsystem flexibel zuordnen
- das Betriebssystem muss ...
 - Probleme wie „*spurious interrupts*“ und „*interrupt storms*“ einkalkulieren.
 - das eingetretene Ereignis aus der Behandlungsroutine an die höheren Ebenen und letztendlich zum Anwendungsprozess weiterleiten.

