

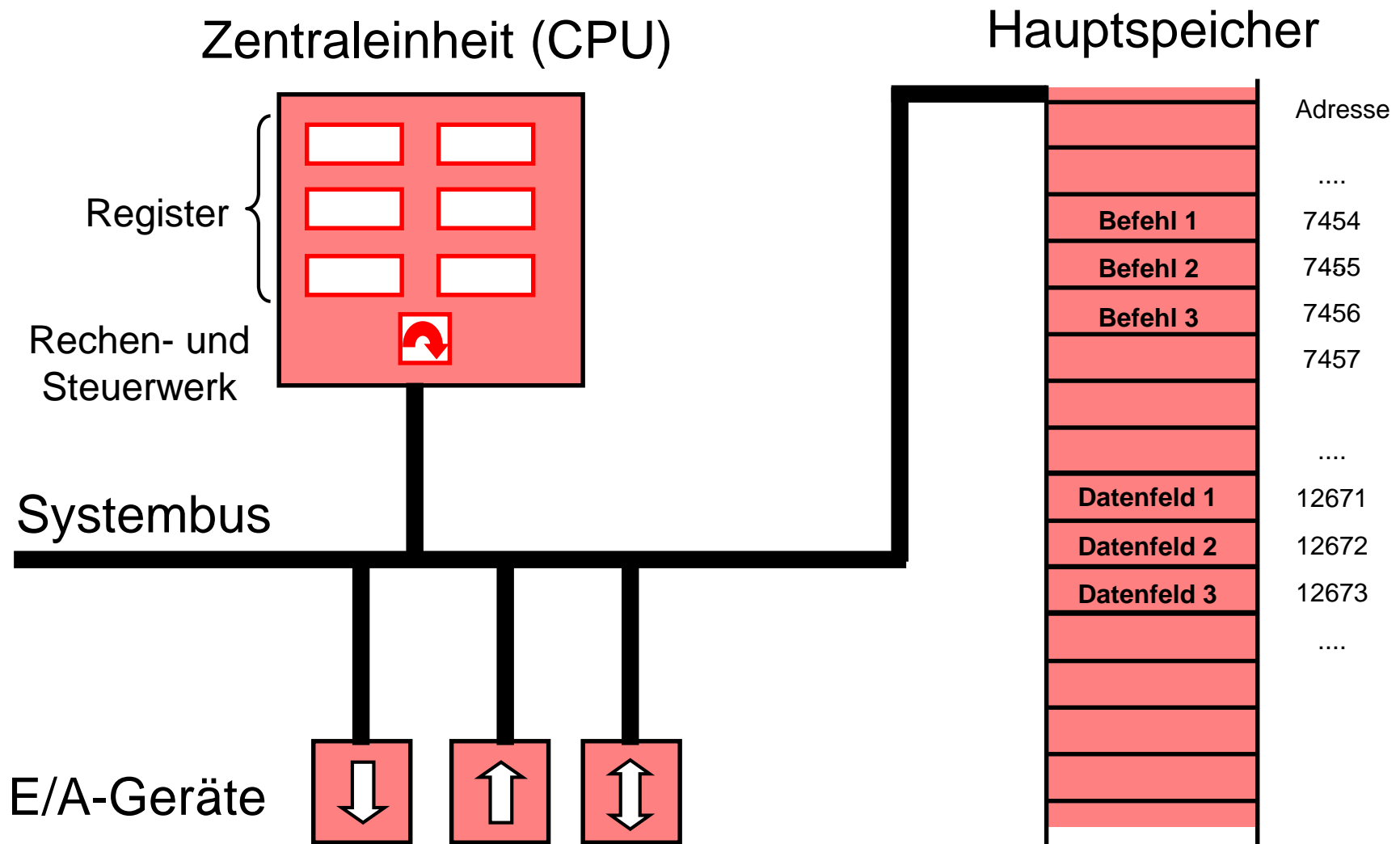


Betriebssysteme

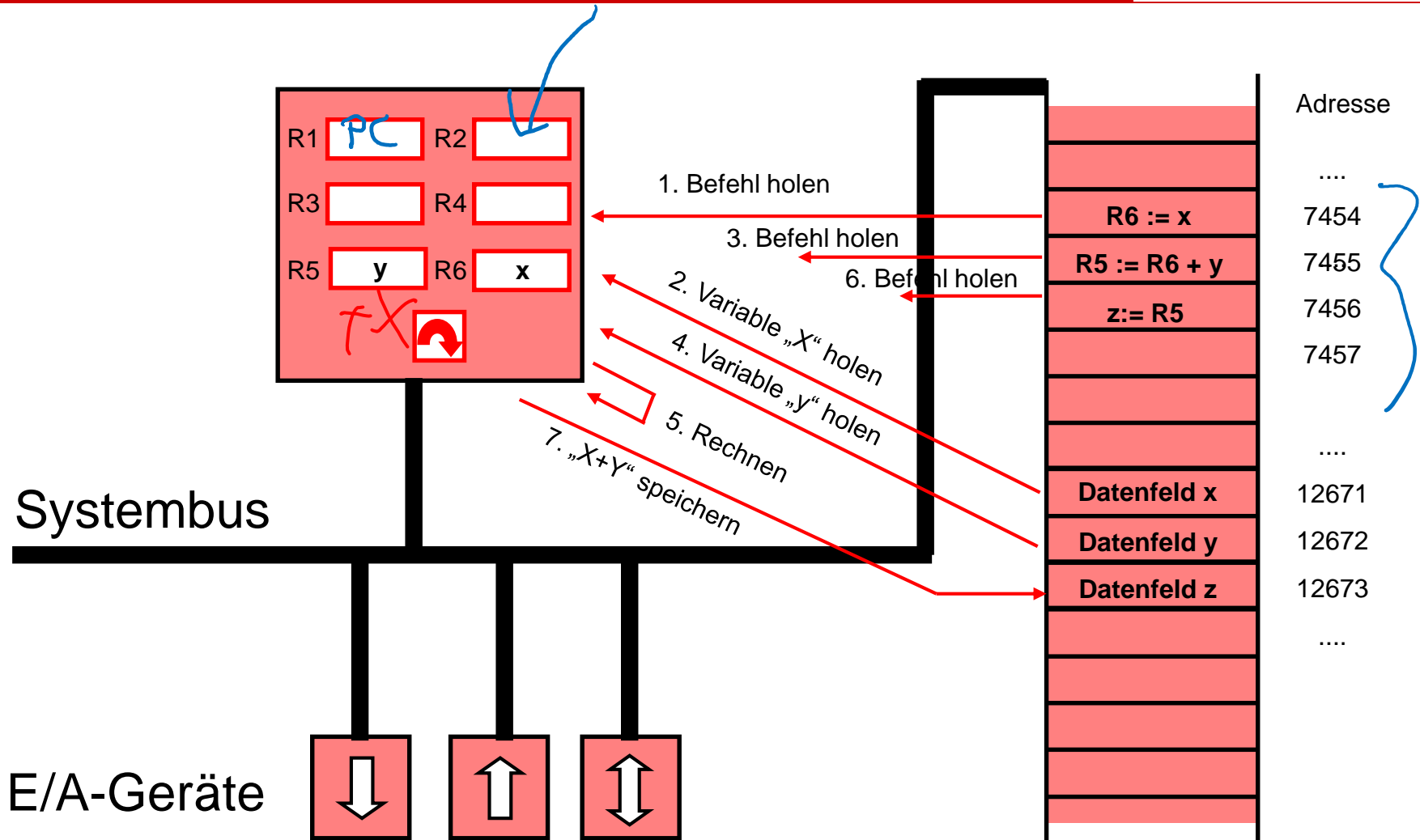
2. Grundlagen

Tobias Lauer

Rechnerarchitektur („Von-Neumann-Architektur“)



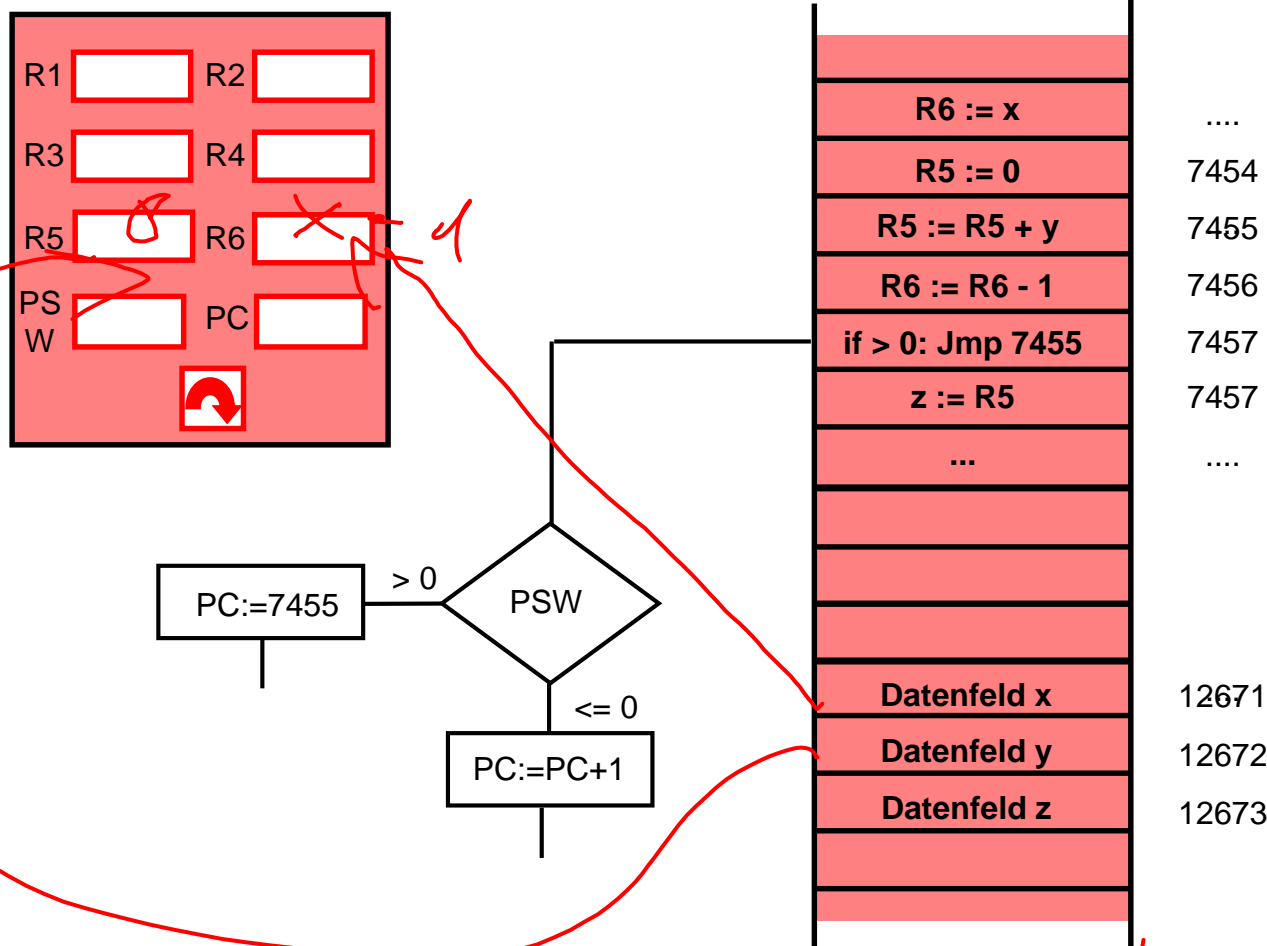
Beispiel: Berechnung von „ $z=x+y$ “



Die wichtigsten Registertypen

- Befehlsregister = aktueller Befehl
(z.B. Read/Write, +, - , AND, ..)
- Programmzähler, Program Counter (PC)
 - Adresse des nächsten Befehls
- Datenregister
 - geladene Werte, Zwischenergebnisse, etc.
 - herausgehoben: Akkumulator (= zentrales Rechenregister)
 - häufig dedizierte Register für Ganzzahl- und Gleitkommaoperationen (FPU)
- Adressregister (zeigen auf Hauptspeicher)
 - z.B. Stapelregister (stack pointer)
- Programmstatuswort (PSW): aktueller Rechenzustand
 - Unterbrechungszustand
 - Modus (User Mode, Kernel Mode)
 - Boole'sche Ergebnisse der letzten arithmetischen oder Vergleichsoperation

Beispiel für verzweigende Rechenausführung



Rechenzustand

- Der **Zustand** eines Rechners lässt sich (in erster Näherung) komplett beschreiben durch:
 - Inhalt des Hauptspeichers („was steht auf welcher Adresse“)
 - Inhalt aller Register („was steht in den Registern, inkl. Befehlsregister, Programmzähler, Akkumulator, PSW, etc.“)
 - (E/A Geräte unberücksichtigt)
- **Das heißt:** Sichert man Hauptspeicherzustand und speichert Inhalte aller Register irgendwo ab, so kann man eine Berechnung unterbrechen und genau an der gleichen Stelle wieder fortführen.

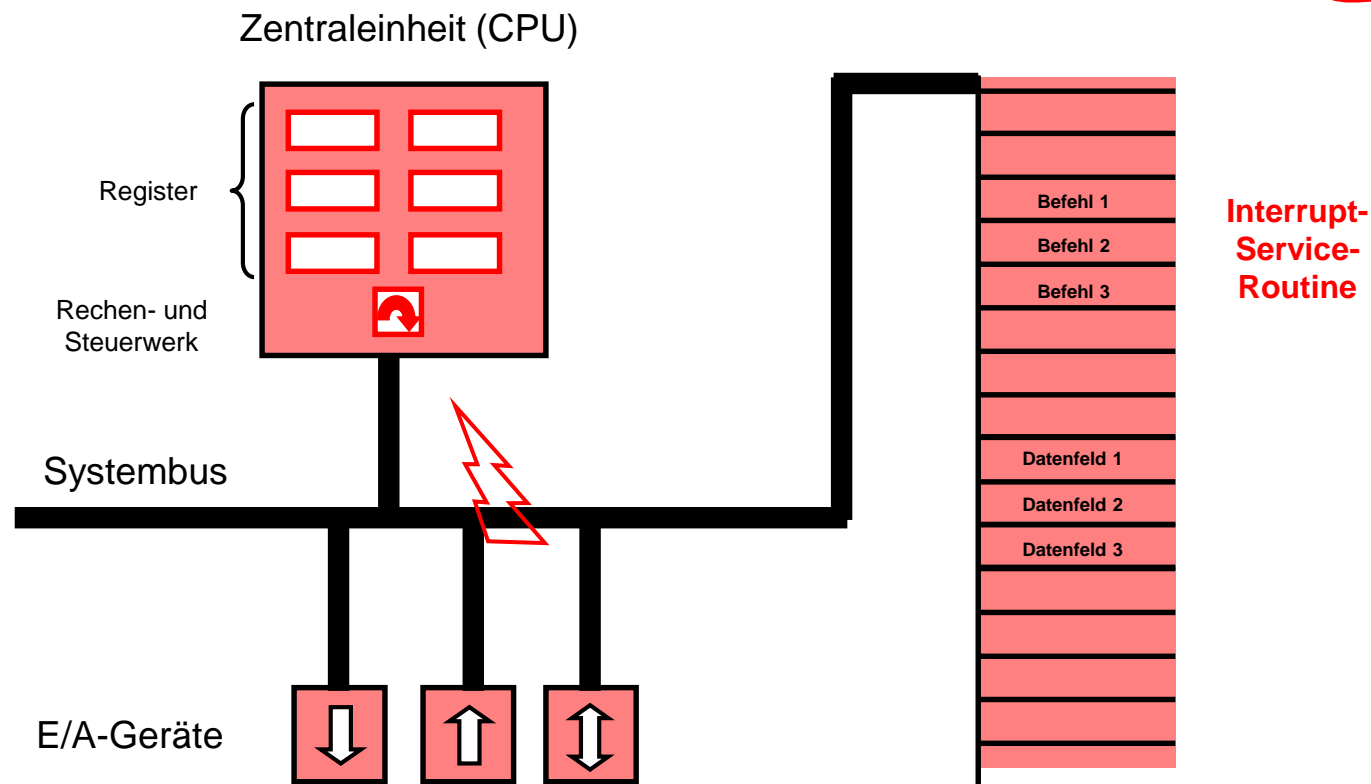
Wichtig für später!

Interrupts (Unterbrechungen)

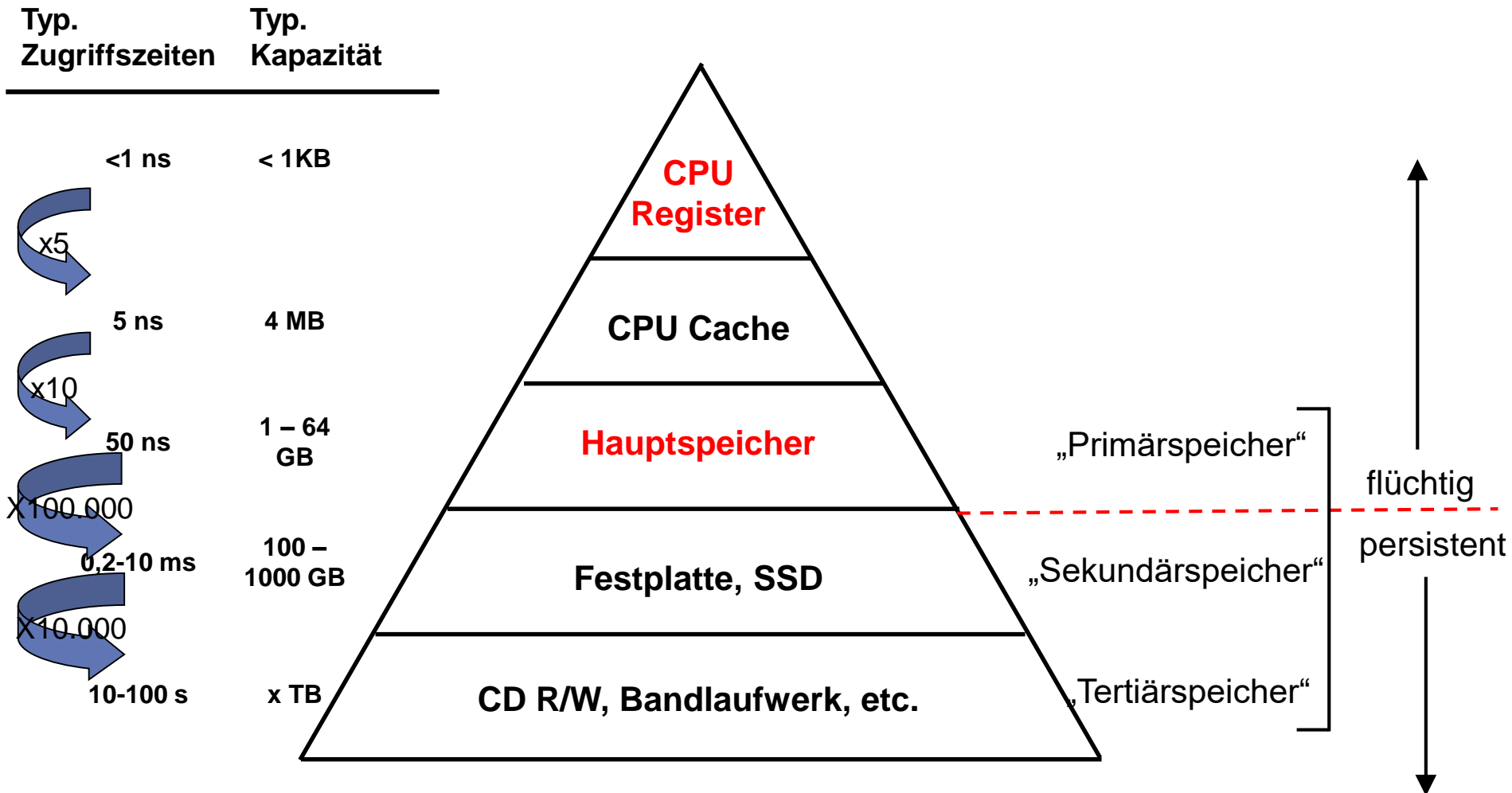
- E/A-Geräte "strecken die Finger" wenn externe Ereignisse zu behandeln sind.

Kennen Sie noch andere Typen von Interrupts?

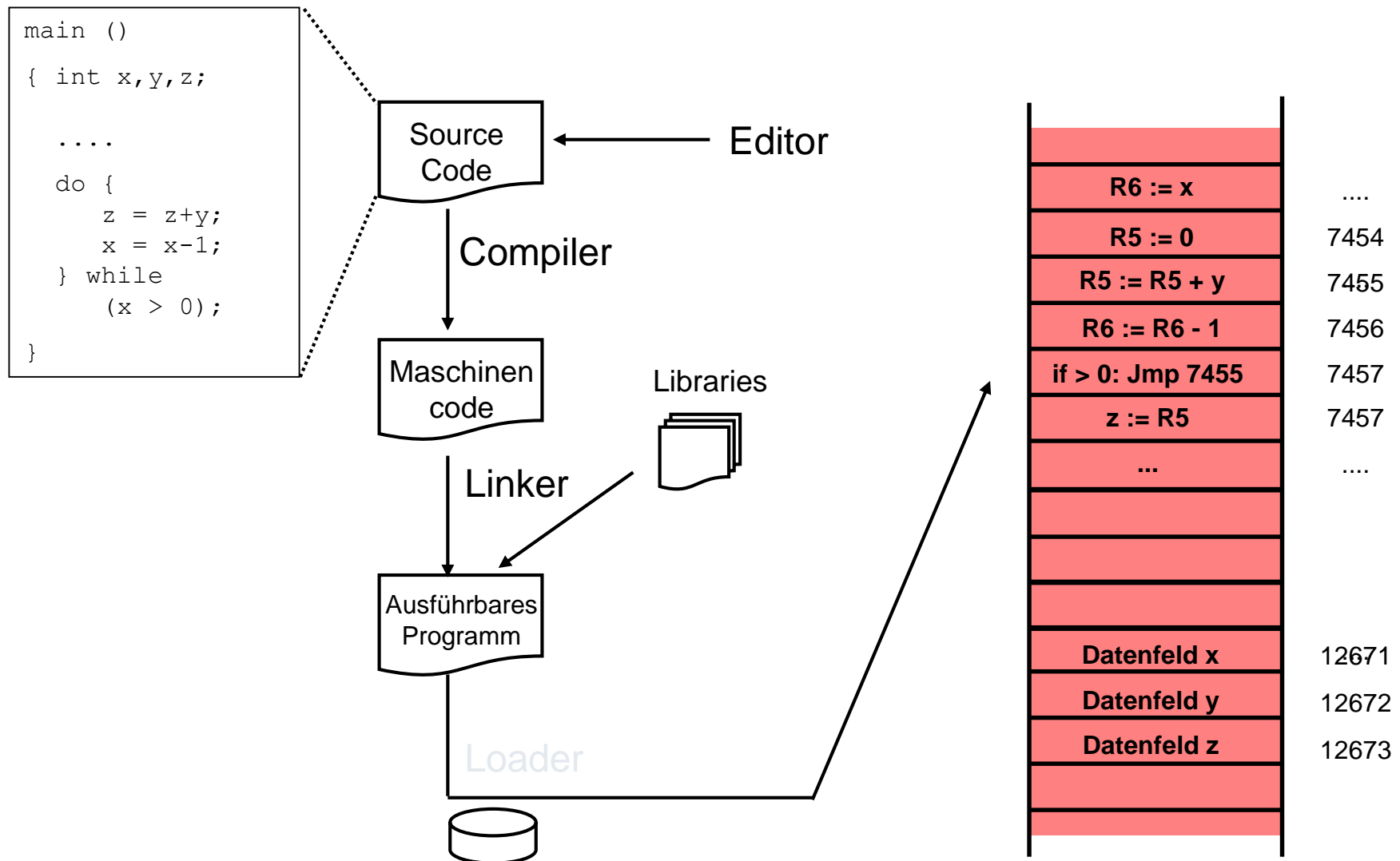
*Software
Times*



Speicherhierarchie eines Rechners



Vom Programm zur Maschine



Rolle von Compiler und Linker

- Compiler bildet höhere Programmiersprache auf Maschinencode ab, z.B. **for**-Konstrukt → Schleife mit bedingtem JMP-Statement
- Compiler definiert für jedes Programmobjekt einen Bereich im Hauptspeicher
- Compiler optimiert Anzahl der Befehle
- Compiler optimiert Registerausnutzung (möglichst wenig Hauptspeicher-Zugriffe)
- Compiler fügt Laufzeitprüfungen ein (z.B. Einhaltung von Arraygrenzen)
- Compiler fügt Aufrufe an Bibliotheksfunktionen ein, z.B. mathematische Funktionen oder Systemaufrufe („System Calls“)
- Linker fügt „ergänzende“ Programmteile hinzu (ausführender Code für Bibliotheksfunktionen)
 - a) statisch
 - b) dynamisch

Zusammenfassung

- Ein „Programm“ ist auf Maschinenebene eine Folge von Befehlen zum Lesen, Schreiben und Berechnen von Hauptspeicher- und Registerinhalten.
- Ein „Programmazustand“ ist (vereinfacht) die Kombination von Hauptspeicherinhalten, Registerinhalten und der Adresse des nächsten zu bearbeitenden Befehls
- Befehle verändern Dateninhalte („Werte“) und Kontrollfluss („nächster Befehl“) der Rechenausführung
- Dienstprogramme unterstützen den Schritt von der höheren Programmiersprache zur Maschine