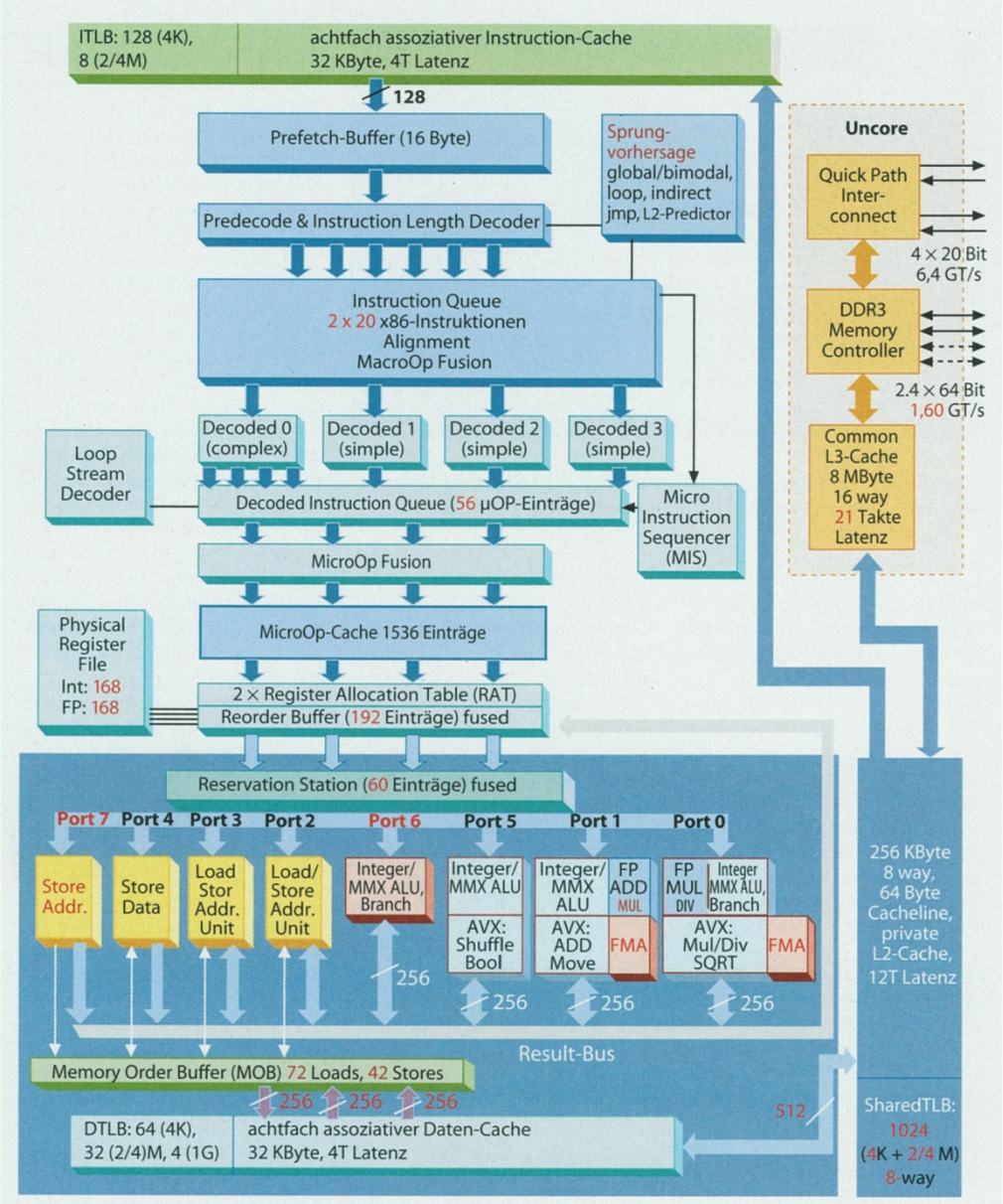
Einführung (1)

Erster funktionsfähiger programmgesteuerter Rechenautomat Z3, fertiggestellt 1941
Bild: Nachbau im Deutschen Museum München

Einführung (2)



Architektur des Haswell-Prozessors (aus c‘t)

Einführung (3)

Was ist ein Rechner?

Maschine, die Probleme für Menschen lösen kann, in dem sie Befehle (Instruktionen) ausführt, die ihr gegeben werden.

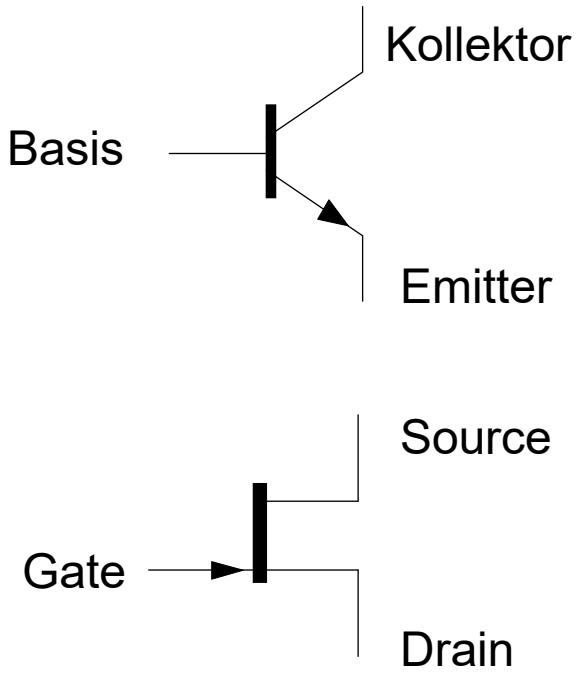
Wer führt die Befehle aus?

- elektronische Schaltkreise
- Alle Komponenten eines Rechners, die sich aus solchen Schaltkreisen zusammensetzen, bilden zusammen die **Hardware** eines Rechners

Transistor

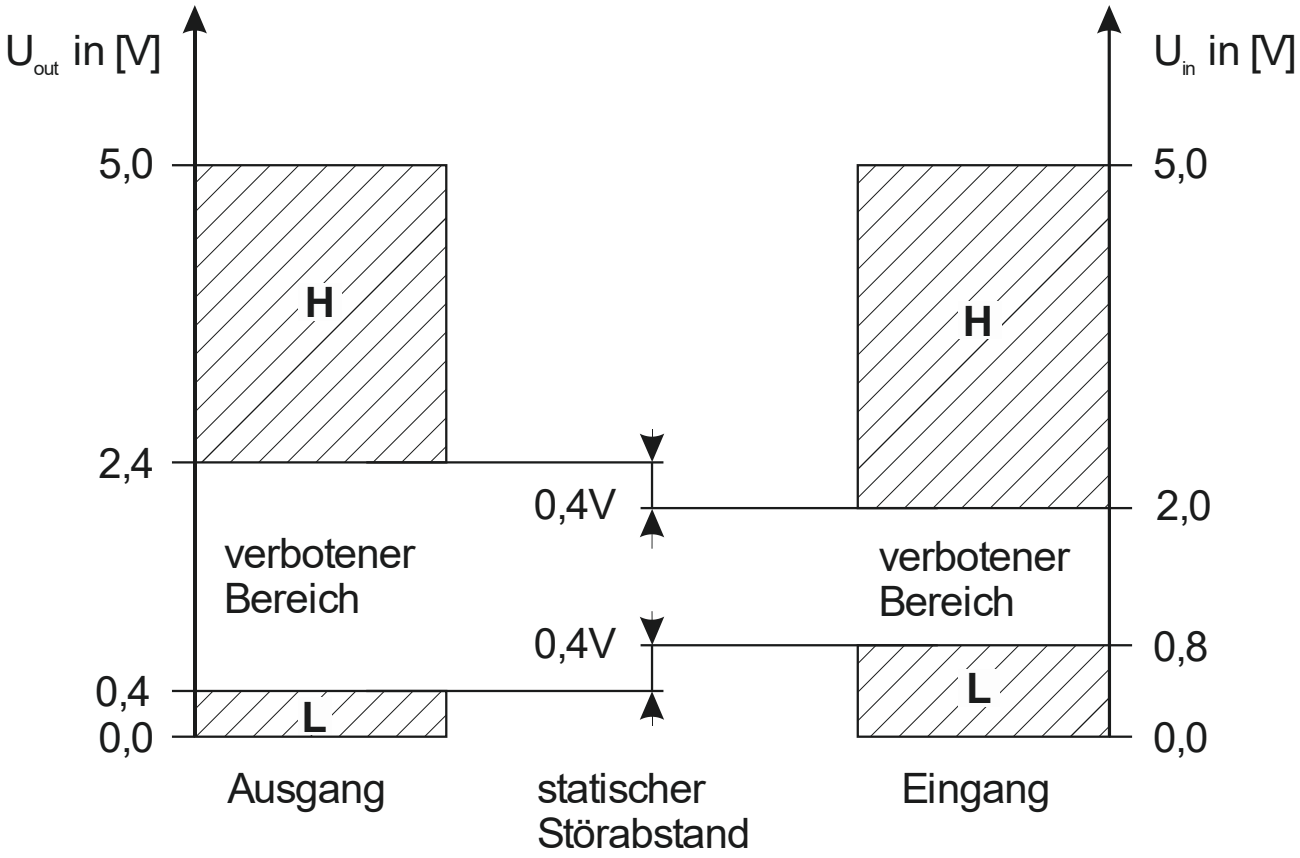
- Grundeinheit einer integrierten Schaltung
- analoges System: Variable haben kontinuierlichen Wertebereich
- können im Schaltbetrieb betrieben werden: 2 Zustände
- modellierbar als digitales System: Anzahl der Elemente (*Symbole*) eines Wertebereiches (*Alphabet*) ist endlich, in der Regel durch 2 Zustände repräsentiert

Einführung (4)



- **Bipolar-Transistor** (hier: npn) verstärkt Strom, der in die Basis fließt, um einen Verstärkungsfaktor zum Kollektorstrom
- **Feldeffekt-Transistor** (hier: n-kanal): Der Stromfluss zwischen Source und Drain wird durch die Spannung am Gate gesteuert.
- Beide Transistorvarianten sind *analoge* Bauelemente.

Einführung (5)



Pegeldefinition für TTL
High (H) entspricht logisch 1, Low (L) entspricht logisch 0.

Einführung (6)

digitale Rechner (Computer)

- haben ein Alphabet mit 2 Symbolen, in Zahlen - und Sprachdarstellung 0 und 1, in der Logik auch F(alse) und T(rue) genannt.
- benutzen also das Binärsystem, ein Symbol in Zahlendarstellung ist also eine binäre Ziffer (**Bi**nary digi**t**), kurz **Bit** genannt.
- Ein Bit ist die kleinste Informationseinheit in einem Computer
- Ein Bit wird in einem Computer durch einen bestimmten Spannungsbereich dargestellt

Zusammenfassung:

- 2 wesentliche Ebenen im Bereich der Hardware:
  - Logische oder Gatterebene (Gatter als kleinste Komponente einer logischen Schaltung)
  - Physikalische oder Transistorebene (Transistor als kleinste Komponente von elektronischen, analogen Schaltungen)

Einführung (7)

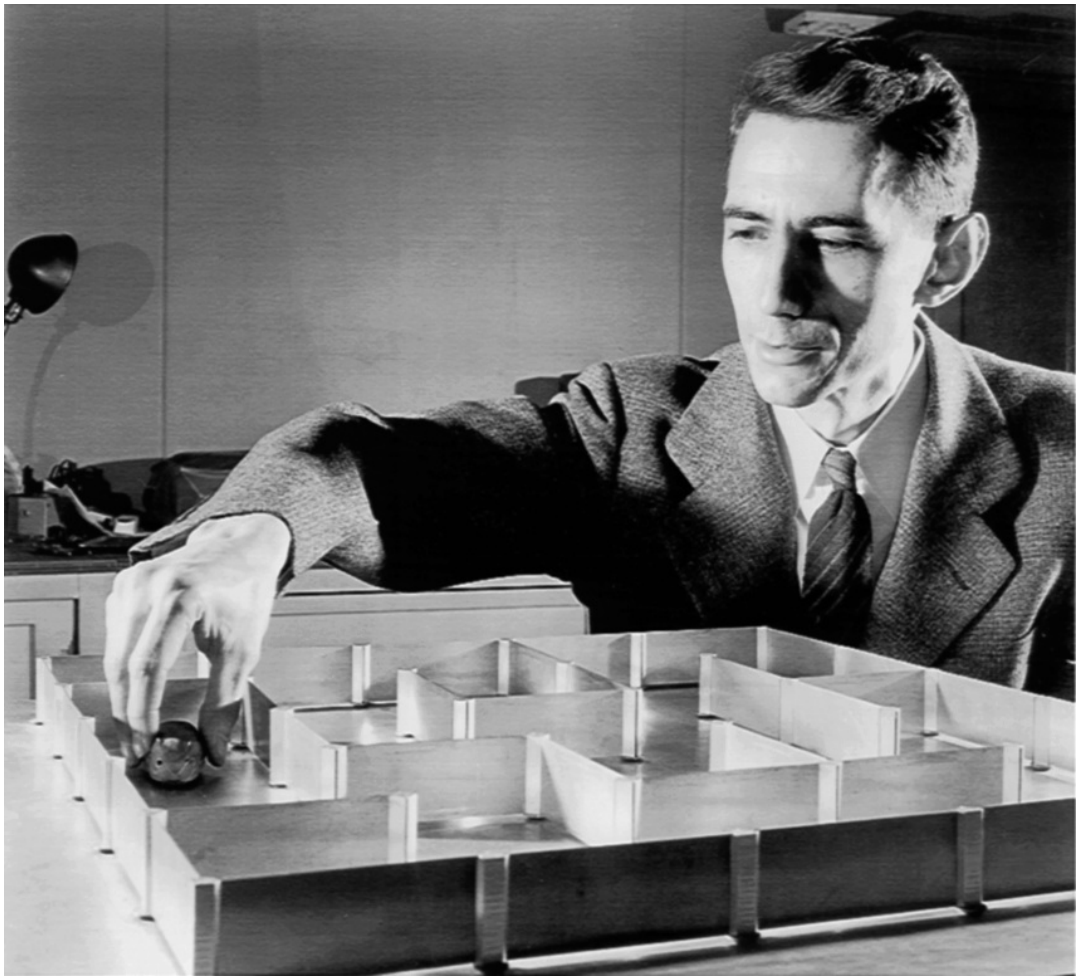


Foto: Bell Labs

Claude Elwood Shannon

Einführung (8)

Aufgabe der Software:

Festlegung, in welcher logischen Ordnung die Hardware welche Befehle zur Lösung der anstehenden Aufgabe ausführt

Algorithmus:

Formale Beschreibung der einzelnen Schritte eines solchen Verfahrens

Programm:

Formulierung des Algorithmus in einer dem Computer verständlichen Sprache. Diese Sprache ist definiert durch den Befehlssatz, d.h. die Menge der Befehle, welche die Hardware ausführen kann.

Problem:

Die Hardware versteht nur Befehle, die aus Bitfolgen (Folgen von Nullen und Einsen) bestehen.

Einführung (9)

Abstraktionsebenen

Ebene	Abstraktion	Beispiel
4	Metasprache, Hochsprache	graphische Programmierung
3	Hochsprache, problemorientiert	C/C++, Pascal, Fortran, Java
2	Maschinensprache	Assembler
1b	komplexe Funktionseinheiten	ALU, Steuerwerk, Speicher
1a	einfache Funktionseinheiten	Register
0	Gatter	AND, OR
-1	Transistoren	CMOS, TTL
-2	Atome	Elektronen-Loch-Modell
-3	Quanten	Bänder / Quantenmodell

Ebenen, die in dieser Lehrveranstaltung betrachtet werden

Einführung (10)

Lösung:

wiederholte Anwendung des Abstraktionsprinzips

Zusammenfassung:

- Sowohl HW - als auch SW-Designer benutzen das Abstraktionsprinzip, um die Komplexität eines Rechners besser zu beherrschen
- Die Details der Implementierung einer Ebene werden in der nächst-tieferen „versteckt“

→ Für die Implementierung einer Abstraktion gibt es prinzipiell verschiedene Möglichkeiten

Wir befassen in dieser LV hauptsächlich mit den Ebenen 0 bis 2

Einführung (11)

Warum Beschäftigung mit den niederen Ebenen?

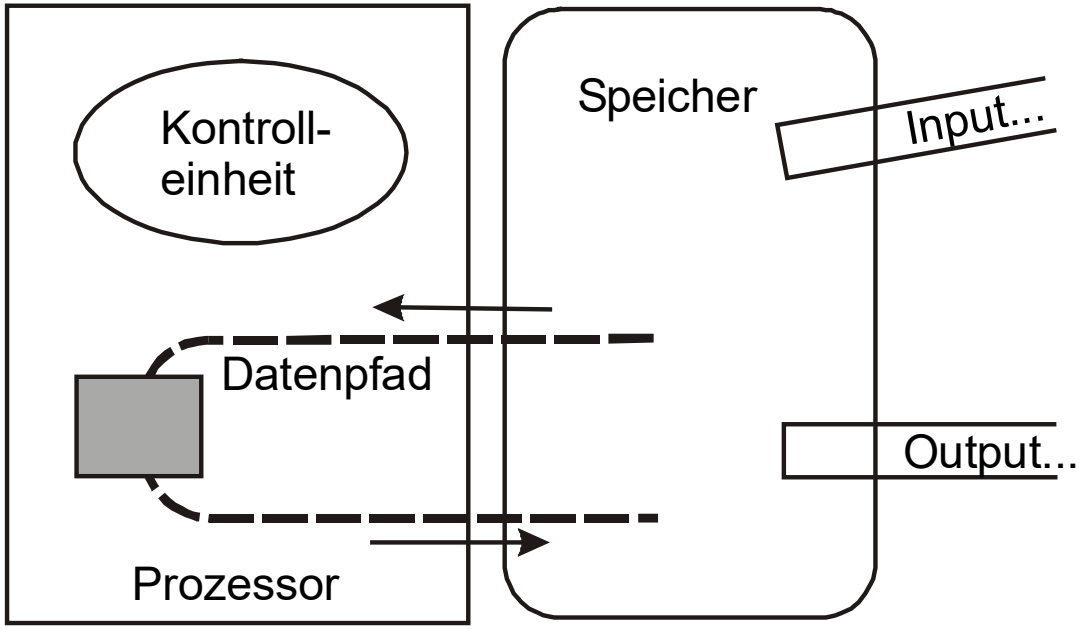
- Hardware und Software sind *logisch äquivalent*
  → Eine der wichtigsten Entscheidungen beim Entwurf eines Rechners ist, was wird in HW (Ebene 1 und tiefer) und was wird in Software (Ebene 2 und höher) realisiert

    Kriterien:
    - Kosten
    - Geschwindigkeit
    - Zuverlässigkeit
    - Häufigkeit der unveränderten Nutzung

- Hardware definiert die Grenzen eines Rechners (Geschwindigkeit)
- Rechner eingesetzt als „eingebettetes System“
- Wichtig zum Verstehen einiger Varianten von Sicherheitslücken
- Wichtig für Rechnerdesigner und -anwender (Informatiker), nicht Rechnernutzer

Einführung (12)

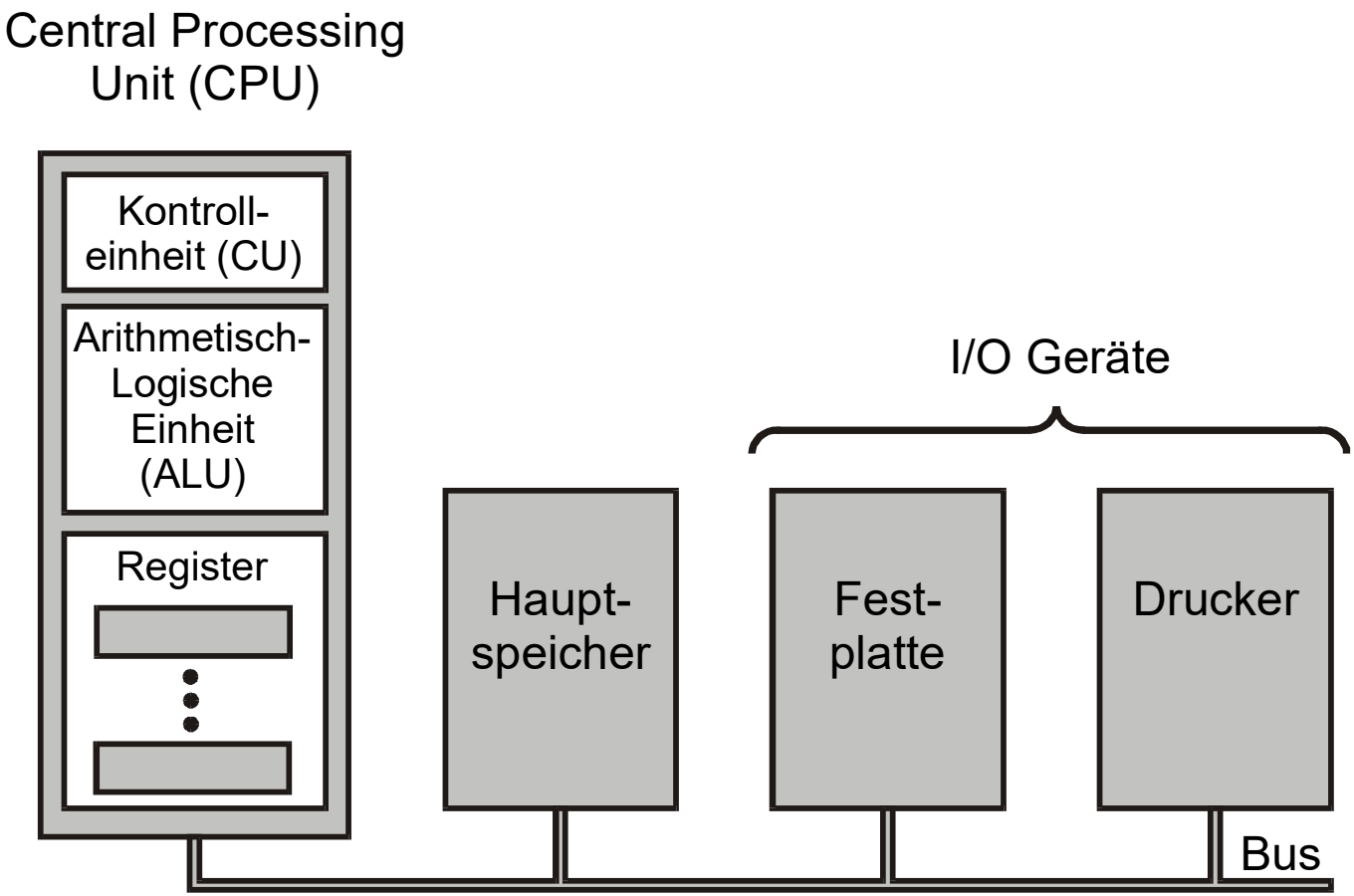
Grobstruktur der Organisation eines Rechners:



Eingabe → Verarbeitung → Ausgabe (EVA)

Einführung (13)

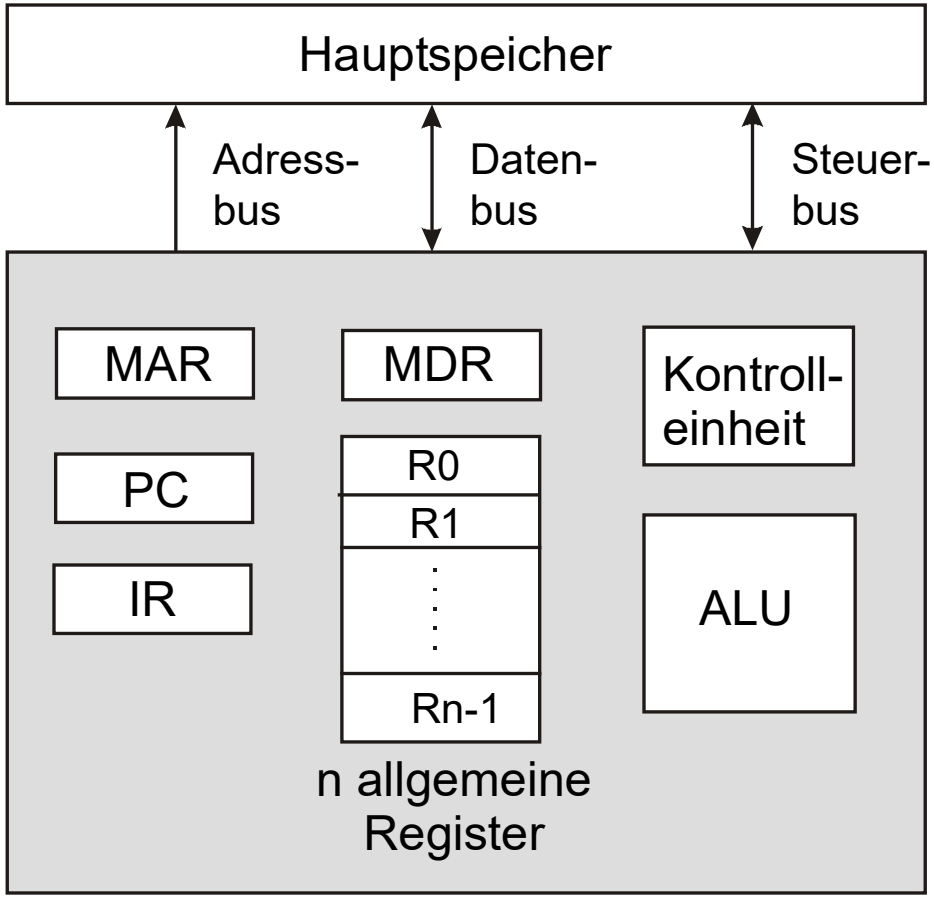
Kommunikationsstruktur zwischen den Komponenten:



Bus: Datenbus, Adressbus, Steuerbus

Einführung (14)

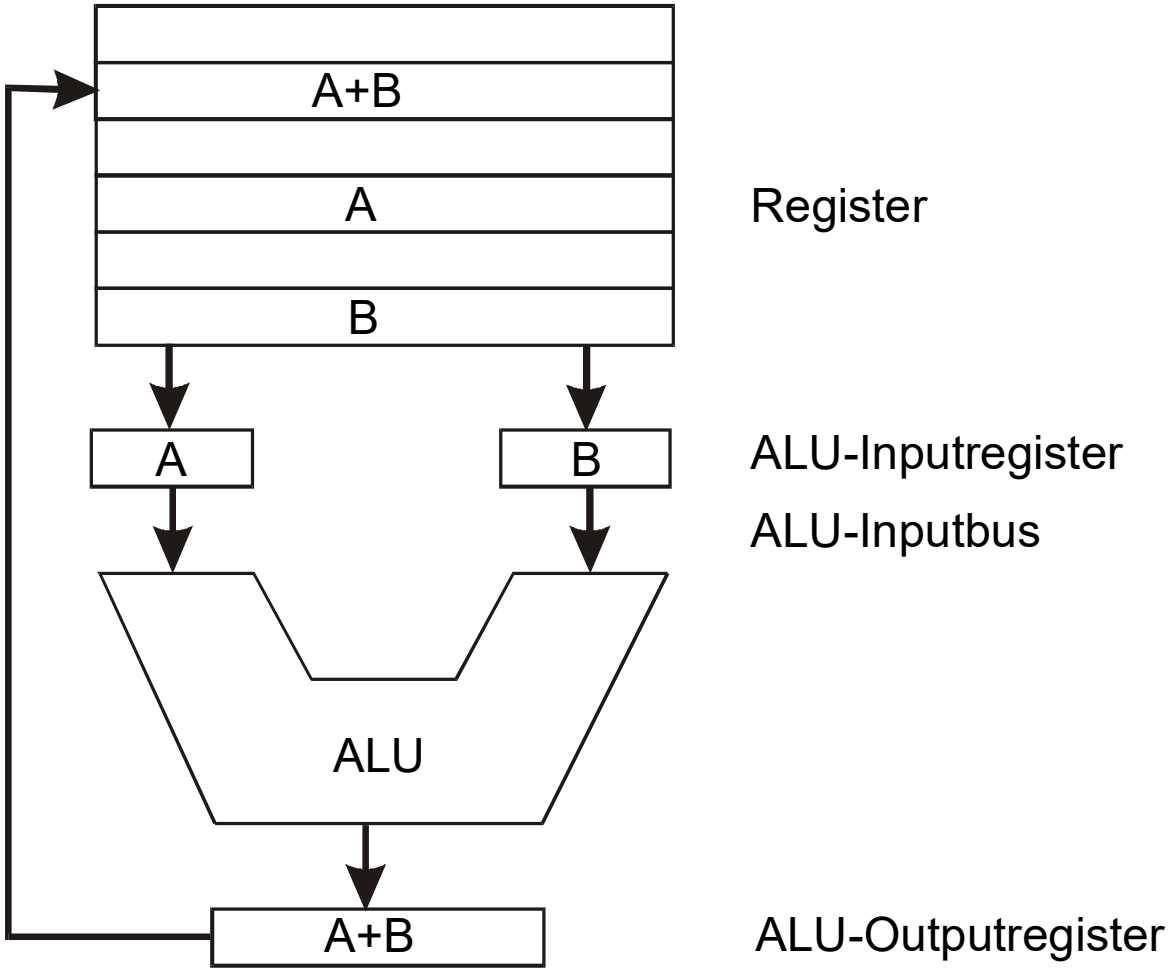
Verbindung Prozessor - Hauptspeicher:



MAR	Memory Adress Regisster
MDR	Memiry Data Register
ALU	Arithmetic Logic Unit
IR	Instruction Register
PC	Program Counter

Einführung (15)

Grobstruktur des Datenpfades :



Einführung (16)

- **Zusammenfassung**

  Wie sind diese Komponenten im Einzelnen entworfen (insbesondere Prozessor, Speicher und I/O) und wie funktionieren sie in sich und miteinander? Dies ist die Themenstellung dieser einsemestrigen Lehrveranstaltung

Einführung (17)

Gliederung

- Aufbau logischer Schaltnetze auf der Basis von Gattern
- Boolesche (Schalt-) Algebra: Vereinfachung und Synthese von logischen Schaltnetzen
- Aufbau von sequenziellen Schalzungen als Register, Zähler, Automaten mittels Flip-Flops
- Computerarithmetik
  - Zahlendarstellungen (z.B. Zweierkomplement, Gleitkomma)
  - Binäre Arithmetik (Grundrechenarten)
- Codes
  - Zeichendarstellungen
  - EDC, ECC, optimale Codes

Einführung (18)

Gliederung

- Programme auf Maschinen/Assemblerebene (Ebene 2)
  - Darstellung und Ausführung von Befehlsfolgen im Hauptspeicher
  - Adressierungsarten für Hauptspeicherzellen und CPU-Register
    - Überblick über verschiedene Varianten
    - Beispiele auf Basis M68k
- Struktur und Funktionsweise der CPU
  - Adresspfad, Datenpfad
  - Kontrolleinheit (Befehlsdekodierung)
  - RISC-Prozessoren, Weitere moderne Rechnerkonzepte
- Struktur und Funktionsweise des Speichers
  - Aufbau einer Speicherhierarchie
    - Cache
    - virtuelles Speicherkonzept für Haupt- und Plattenspeicher
- Parallelrechner
  - Konzepte,Topologien und Metriken