



Berner Fachhochschule  
Haute école spécialisée bernoise  
Bern University of Applied Sciences

# Hardwarenahe Softwareentwicklung

## Computer-Systeme und Recheneinheiten

V5.1, ©2023 roger.weber@bfh.ch

# Lernziele

Sie sind in der Lage:

- ▶ Architekturen von Computer-Systemen zu erklären.
- ▶ Die Funktionalität einer CPU zu verstehen.
- ▶ Eine Applikation für eine FPU zu programmieren.
- ▶ Einen Microcontroller für ein konkretes Projekt auszuwählen.



# Inhaltsverzeichnis

## 1. Computer-Systeme

- Komponenten und Kategorien
- Klassifizierung nach Flynn
- von-Neumann und Harvard Architekturen

## 2. Recheneinheiten

- CPU
- FPU
- Prozessor-Typen und Familien

### Bemerkung

Dieses Kapitel ist sehr allgemein gehalten und geht mit wenigen Ausnahmen nicht auf einzelne Hersteller und CPU-Familien ein.

# Computer-Systeme

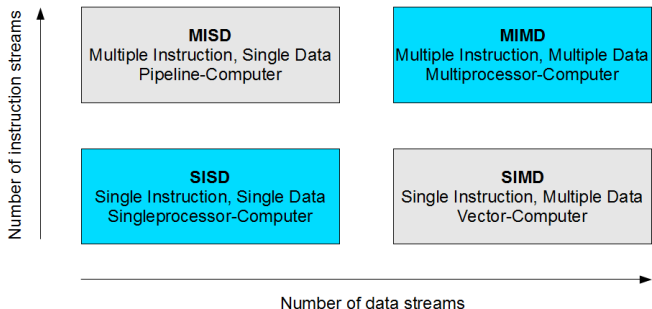
# Komponenten eines Computer-Systems



- ▶ Welches sind die wichtigsten Komponenten eines Computer-Systems?

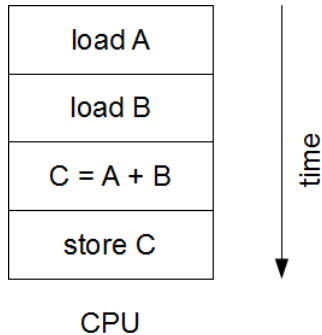
# Die vier Computer-Kategorien nach Flynn

- ▶ Michael J. Flynn teilt Computer in vier Kategorien ein.



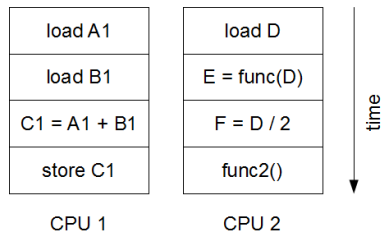
# SISD

- ▶ Sequenziell
- ▶ **Single Instruction:** Pro Taktzyklus eine Programminstruktion.
- ▶ **Single Data:** Pro Taktzyklus nur ein Datensatz.



# MIMD

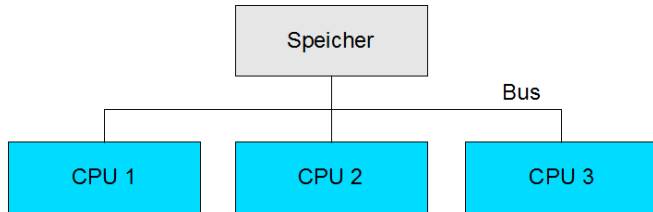
- ▶ **Multiple Instruction:** Pro CPU eine Programminstruktion pro Taktzyklus. Programme auf mehrere CPUs aufgeteilt (Prozesse, Threads, Tasks).
- ▶ **Multiple Data:** Jede CPU bearbeitet einen anderen Datensatz.





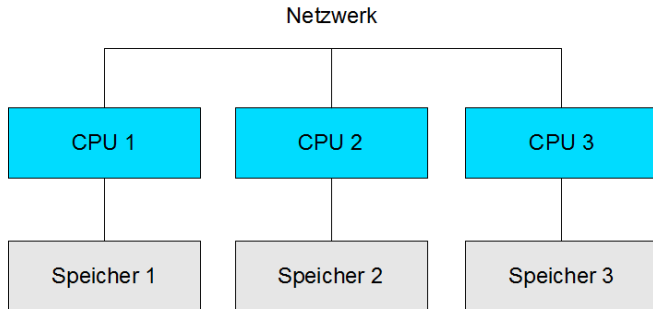
## MIMD: Shared Memory (eng gekoppelte Systeme)

- ▶ Mehrere CPU teilen sich einen gemeinsamen Speicher (Shared Memory).
- ▶ Vorteil: Ändert eine CPU einen Speicherbereich, so ist dies für die anderen CPUs sofort ersichtlich.
- ▶ Nachteil: Der Zugriff auf den Speicher muss synchronisiert werden (kein gleichzeitiger Zugriff).



## MIMD: Distributed Memory (lose gekoppelte Systeme)

- ▶ **Mehrere CPU kommunizieren über ein Netzwerk miteinander.**
- ▶ Jede CPU hat ihren eigenen Speicher. Der Datenaustausch erfolgt über das Netzwerk.
- ▶ Dieses System kann aus mehreren SISD-Systemen mit Netzwerkanbindung aufgebaut werden.



# Vergleich Shared / Distributed Memory



- ▶ Welche Architektur ist besser skalierbar?
- ▶ Bei welcher Architektur ist der Speicherzugriff schneller?



In Embedded Systems werden häufig 8-Bit oder 32-Bit Microcontroller eingesetzt.

- ▶ Welcher Kategorie entsprechen diese?

Für die Kommunikation mit anderen Komponenten werden diese häufig über Bussysteme wie CAN verbunden.

- ▶ Sind dies eng oder lose gekoppelte Systeme?



Heute werden im PC-Bereich oft  
Multicore-Prozessoren eingesetzt, beispielsweise der  
Intel® Core™ i7 Prozessor (4 bis 14 Cores). Link:  
<https://www.intel.de/content/www/de/de/products/details/processors/core/i7.html>

- ▶ Welcher Kategorie entsprechen diese?
- ▶ Sind diese Cores eng oder lose gekoppelt?

# Übersicht Architekturen

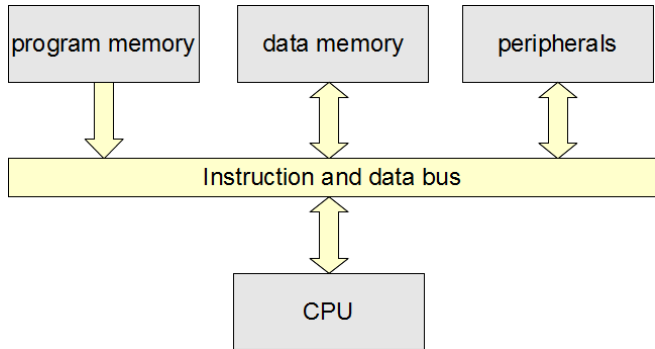
Für den Zugriff von der CPU auf den Speicher existieren bei Computern zwei gängige Architekturen:

- ▶ **von-Neumann**
- ▶ **Harvard**

Beide Architekturen gehören nach dem Klassifizierungsverfahren von Michael J. Flynn zur Kategorie der **SISD-Systeme**.

# von-Neumann Architektur

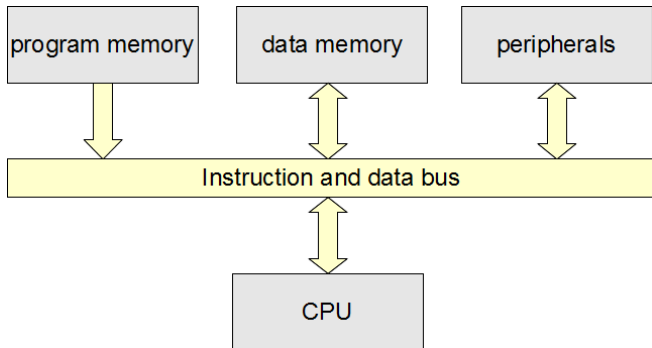
- ▶ Sequenzieller Speicherzugriff
- ▶ Programmspeicher:
  - ▶ Programminstruktionen
  - ▶ nicht flüchtig
- ▶ Datenspeicher:
  - ▶ Variablen und Anwenderdaten
  - ▶ flüchtig
- ▶ Peripherie:
  - ▶ Sensoren, Einlesen von Daten
  - ▶ Aktoren, Ausgabe von Daten



# von-Neumann Architektur

1. CPU liest  
  Programminstruktionen
2. CPU manipuliert Daten

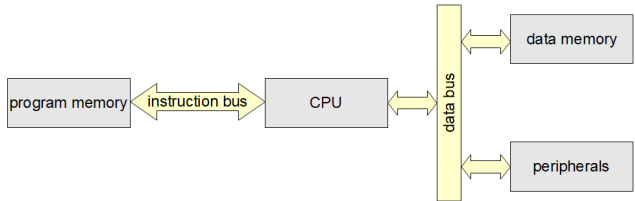
→ “von-Neumann-Flaschenhals“



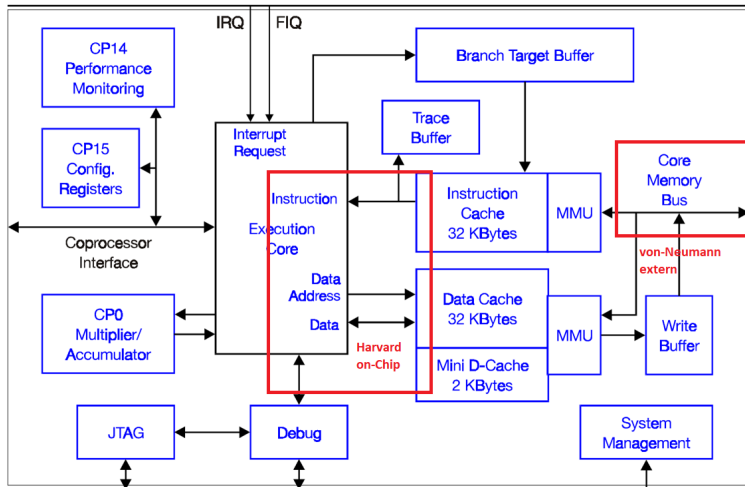


# Harvard-Architektur

- ▶ Getrennte Adressräume für Programmspeicher und Datenspeicher.
- ▶ Instruktionsbus und Datenbus, parallel.
- ▶ Bei DSPs (Digitale Signal-Prozessoren).
- ▶ Bei modernen Microcontrollern on-Chip.



# Hybrid von-Neumann / Harvard



Quelle: PXA 270 Data Sheet

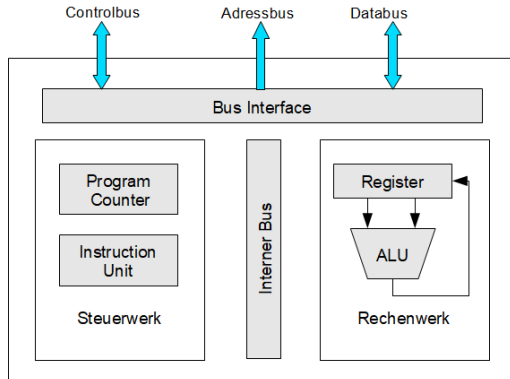


- ▶ Diskutieren Sie Vor- und Nachteile der von-Neumann und der Harvard-Architektur.

## Recheneinheiten

# CPU (Central Processing Unit)

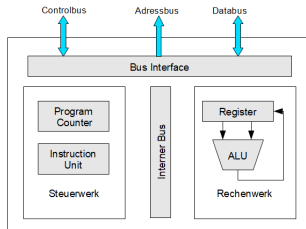
- ▶ Klassischer Aufbau einer CPU (je nach Hersteller und Typ sind Abweichungen möglich):



# Steuerwerk

Das Steuerwerk ist für den Programmablauf verantwortlich und beinhaltet:

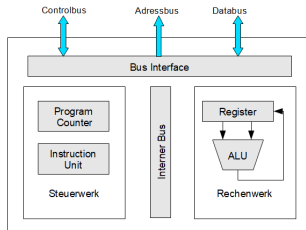
- ▶ **Instruction Unit** (Schaltwerk).
  - ▶ Interpretation der Instruktionen
- ▶ **Program-Counter** (PC).
  - ▶ Pointer auf nächste Programminstruktion



# Steuerwerk

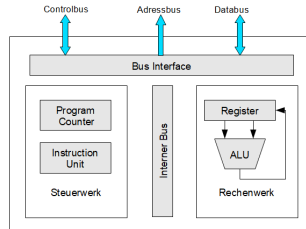
## Aufgaben des Steuerwerks:

- ▶ Laden der Programminstruktion aus dem Programmspeicher (**fetch**).
- ▶ Entschlüsseln der Programminstruktion (**decode**) nach Operation und Operanden.
- ▶ Befehlsausführung (**execute**) in folgenden Schritten:
  - ▶ Steuersignale für ALU oder andere Funktionseinheiten generieren.
  - ▶ Operanden adressieren und laden.
  - ▶ Ergebnis speichern.
  - ▶ Program-Counter anpassen.



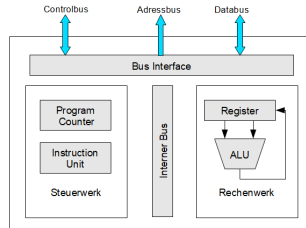
# Rechenwerk

- ▶ Verarbeitung der Daten.
- ▶ **ALU** (Arithmetic Logical Unit) führt Rechenoperationen durch (nur Integer-Operationen).
- ▶ **Register** sind schnelle Zwischenspeicher für Operanden und Resultate.



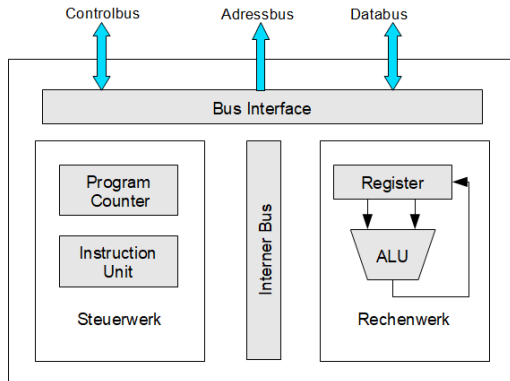


- ▶ Typische **ALU-Operationen** sind:
  - ▶ Boolesche Operationen (logisch AND, OR, EXOR, NOT)
  - ▶ Arithmetische Operationen (Addition, Subtraktion, Multiplikation)
  - ▶ Vergleiche und logische Entscheidungen (Vergleiche und Verzweige falls gleich)
  - ▶ Schiebe-Operationen (shift left, shift right)
  - ▶ Transfer Operationen (laden und speichern)
- ▶ Für Floating-Point Instruktionen oder komplexe mathematische Funktionen wird oft auch eine **FPU** (Floating Point Unit) verwendet.



# Interner Bus, Businterface

- ▶ Interner Bus: Verbindet Steuerwerk und Rechenwerk.
- ▶ Businterface: Kommunikation mit Speicher und Peripherie.

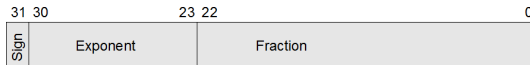


# FPU (Floating-Point-Unit)

- ▶ Vorteile des Floating Point Formats:
  - ▶ Grosser Wertebereich, keine Überläufe wie bei Integer.
  - ▶ Darstellung von sehr kleinen Zahlen.
- ▶ Realisierung
  - ▶ In Software mit Hilfe von Libraries
  - ▶ In Hardware mit Hilfe einer Floating Point Unit (FPU)

# Floating Point Darstellung

- ▶ IEEE-754 Format
- ▶ Single Precision (32 Bit) oder Double Precision (64 Bit)
- ▶ Single Precision: Sign (1 Bit), Exponent (8 Bit) und Fraction (23 Bit):



$$\text{Value} = (-1)^{\text{Sign}} * 2^{(\text{Exponent} - 127)} * (1 + (\frac{1}{2} * \text{Fraction}[22]) + (\frac{1}{4} * \text{Fraction}[21]) + \dots + (\frac{1}{2^{23}} * \text{Fraction}[0]))$$



- ▶ Wie wird der Wert 1.25 dargestellt? Ermitteln Sie die Hex-Zahl für das Single Precision Format.

# Floating Point Programmierung in C

## Übliche Operationen:

- ▶ Addition,  $z = a + b$
- ▶ Subtraktion,  $z = a - b$
- ▶ Multiplikation,  $z = a * b$
- ▶ Division,  $z = a / b$
- ▶ Quadratwurzel,  $z = \text{sqrtf}(a)$
- ▶ Negation,  $z = -a$
- ▶ Betrag,  $z = \text{absf}(a)$

```
float  pi = 3.141592F;           // Single Precision , 32 Bit
double pi2 = 3.14159265358979323846264338; // Double Precision , 64 Bit

float  a, b, c;
c = sinf(a) + cosf(b) + 1.0F;    // Single Precision

double x, y, z;
z = sin(x) + cos(y) + 1.0;       // Double Precision
```

# Prozessor-Typen

## ▶ **Microprocessor**

- ▶ “Micro” → keine MMU
- ▶ CPU mit Rechenwerk und Steuerwerk.
- ▶ Extern: Speicher und Peripherie
- ▶ Kommunikation über den Adress- / Datenbus.

## ▶ **Microcontroller**

- ▶ Microprocessor mit integrierter Peripherie / Speicher.
- ▶ Platz- und kostenoptimiert.

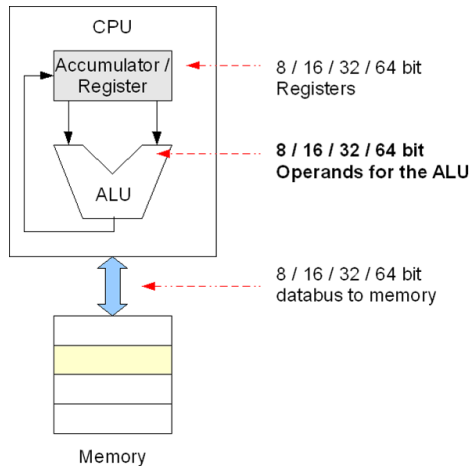
## ▶ **Digital Signal Processor (DSP)**

- ▶ Für rechenintensive Aufgaben (Filterberechnung, Fourier-Transformationen, usw.).
- ▶ Spezialbefehle und Hardware für numerische Algorithmen, z. B. MAC-Einheit, führt eine Multiplikation / Addition in einem Taktzyklus ausführt.

## 8 / 16 / 32 und 64 bit Architekturen

8 / 16 / 32 und 64 bit Architektur bezieht sich auf:

- ▶ Die Datenbreite, welche durch die Arithmetic Logic Unit (ALU) verarbeitet wird.
- ▶ ev. die Breite der CPU-Register.
- ▶ ev. die Breite des Datenbusses.



# (Micro)-Controller-Familien

- ▶ Atmel AVR
- ▶ 8051-Familie
- ▶ PIC10 bis PIC18



8-Bit

- ▶ MSP430



16-Bit

- ▶ ARM / Cortex-Mx / Cortex-Ax
- ▶ MIPS, Power PC
- ▶ Intel x86

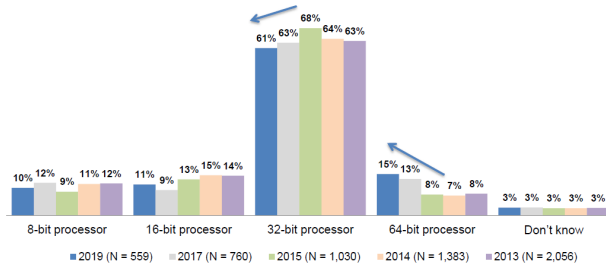


32-Bit



# Auswahlkriterien

My current embedded project's main processor is a:



Quelle: Embedded Market Study 2019



- Nach welchen Kriterien wählen Sie einen Microcontroller aus?