

1 Von-Neumann Schicht

1.1 Von-Neumann Architekturkonzept

1.
Unabhängigkeit zwischen Soft- und Hardware

2.
Rechner besteht aus vier Werken:

- Hauptspeicher (speichert Programme und Daten)
- Leitwerk (interpretiert Programm)
- Rechenwerk (führt arithmetische Operationen durch)
- I/O incl. Sekundärspeicher (Kommuniziert mit Umgebung, Langzeitspeicher)

3.
Der Hauptspeicher besteht aus fortlaufend adressierter Zellen gleicher Größe. Dabei besteht eine Zelle aus Adresse (beginnend ab 0 normalerweise) und Wert. Die Adresslänge bestimmt die maximale Speichergröße. 4.
Programm und Daten existieren im selben Speicher und können manipuliert werden:

Heute oft getrennt verwaltet (Veränderbare/nicht veränderbare Speicherbreite aka. Princeton Architektur)

Alternativen:

1. Programme und Daten in verschiedenen Speichern aka. Harvard Architektur
2. Meist Mischform: gemeinsamer Hauptspeicher/getrennte Zwischenspeicher

5.
Die Maschine benutzt Binärcodes. Zahlen werden dual dargestellt

6.
Das Programm besteht aus einer Folge von aufsteigenden, sequentiellen Instruktionen

1.2 Der Speicher

Dezimal			Binär		
Wert	Abk.	Metrik	Wert	Abk.	Metrik
1	B	byte	1	B	byte
1000	KB	kilobyte	1024	KiB	kibibyte
1000^2	MB	megabyte	1024^2	MiB	mebibyte
1000^3	GB	gigabyte	1024^3	GiB	gibibyte
1000^4	TB	terabyte	1024^4	TiB	tebibyte

Ein Adressraum mit folgender Adresslänge hat folgende Größe:

16 bit: $2^{16} = 64 \text{ KiB}$
 32 bit: $2^{32} = 4 \text{ GiB}$
 64 bit: $2^{64} = 16 \text{ EiB}$

Daten laden/speichern:

1. Adresse an Speicher anlegen (Memory Access Register = MAR)
2. Daten auslesen/an Speicher anlegen (Memory Data Register = MDR)

Speicherzellen:

1. Speicherzellengrößen entsprechen der Verarbeitungsgröße (16/32/64bit)
2. Speicher sind byte-adressierbar (Adresse im Bezug auf Bytegrenzen)
3. Unterschiede in Behandlung großer Datenformate (Datenausrichtung)

Ausrichtung der Daten im Speicher (data alignment):

Ein Datum ist auf eine n-Byte Grenze ausgerichtet wenn seine Adresse ein Vielfaches von n ist ($A \bmod n = 0$)

Häufige Forderung:

1. Datenformate mit n-Bytes müssen auf n-Byte-Grenze ausgerichtet werden
2. Bei ausgerichteten Operanden *min.* # Speicherzugriffe

Nichtausgerichtete Daten (misaligned data):

- Vorteile:
- *Lückenlose Speichernutzung (kein Padding benötigt)*
 - *Bessere Kompatibilität zu älteren ISAs*
- Nachteile:
- *Ggf. zusätzliche Speicherzugriffe nötig*
 - *Höherer Hardwareaufwand nötig*

Byte-Reihenfolge:

24 16 8 0

76	54	32	10
----	----	----	----

 = 0x76543210 (Little Endian)

Little Endian: from lowest to highest (Adresse N zeigt auf lowest Byte)

Big Endian: from highest to lowest (Adresse N zeigt auf highest Byte)

1.3 Von-Neumann Speicher heutzutage

Neuerungen:

1. Segmentierung
2. Virtueller Speicher (Indirektion aller Adressen)

Kritik:

1. Datentypinformation sollte zur Laufzeit mitgeführt werden
2. Benutzung von Variablen führt zu schwierig beweisbaren Eigenschaften

1.4 Operationsprinzip der Von-Neumann Architektur

- zu jedem Zeitpunkt führt die CPU einen einzigen Befehl aus
- die Adresse des Befehls befindet sich im Befehlszähler (BZ)
- Laden des Befehls ins Instruktionsregister (IR)
- Anschließend Dekodierung/Ausführung

Nach Befehlsausführung:

- Inkrementieren des Befehlszähler (nächster Befehl)
- Setzen auf neuen Wert (Sprung)

Zwei-Phasen-Konzept der Befehlsverarbeitung:

1. Interpretationsphase:

- durch die Adresse im BZ wird der Inhalt einer Speicherzelle geholt
- Interpretation als Befehl

2. Ausführungsphase:

- die im Befehl enthaltene Adresse bewirkt das holen einer weiteren Zelle
- Verarbeitung des Inhalts als Datenwert

Rechenoperationen laufen im Rechenwerk (ALU), dabei verarbeitet ein Befehl höchstens ein Datenpaar. Operand und das Ergebnis stehen im Akkumulator (AC). Repräsentierung von weiteren Operanden und evtl. Ergebnis kann durch Speicheradressen erfolgen.

Das Ergebnis im AC beeinflusst ein Statusregister im Leitwerk (Flags).

1.5 Register

Speicherplätze im Prozessor, auf die sehr schnell mit kurzer Adresse (Registernummer) zugegriffen werden kann.

Unterscheidung zwischen Allgemeine Register, Gleitkomma-Register, Spezialregister, Systemregister (für Flags z.B.)

1.6 Funktionseinheiten heute

- Zusammenfassung von Leit- und Rechenwerk zur CPU
- Indirekter Anschluss der I/O-Geräte über Gerätesteuerung
- Aufspalten des Speichers in Speicherhierarchie
- komplexere, hierarchische Verbindungssysteme
- mehrere Rechenwerke im Zentralprozessor