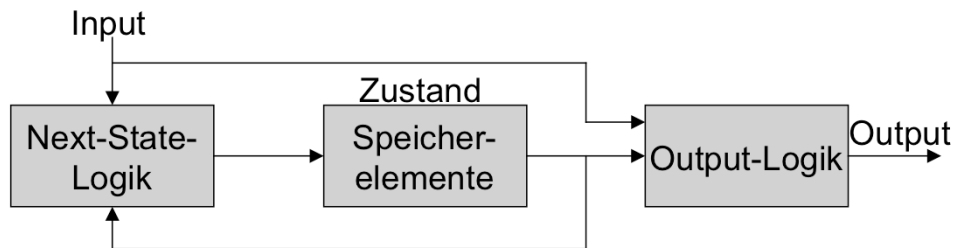


2. Von Neumann Architektur



Sequenzieller Rechner



- ❑ Beschreibung durch endlichen Automaten
- ❑ ungeeignet zur Beschreibung von realen Rechnern
 - Verarbeitung / Speicherung grosser Datenmengen und Transport dieser Datenmengen zwischen Rechnermodulen
 - Rechner kann sein Verhalten ändern.
Die Funktionsweise des Automaten ist aber vorgegeben.

Rechnerarchitektur

2

Oberschelp, S. 181, 203

Described by finite automata

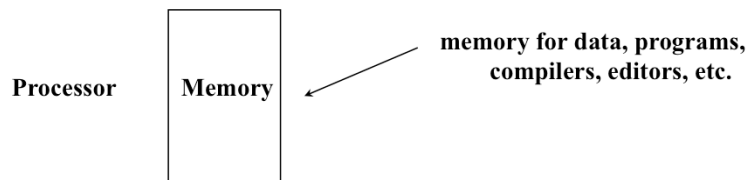
Ineligible to describe real calculators (for two reasons)

processing / memorization of larger datavalues and transportation of these datavalues between calculator modules

Calculator can change its behaviour. The functioning of automata is given (fixed)

Von Neumann Konzept

- ❑ Anweisungen sind Bitfolgen
- ❑ Programme werden im Speicher gehalten
— gelesen und geschrieben wie *normale* Daten



- ❑ Fetch & Execute Cycle
 - Anweisungen werden aus dem Speicher geholt (fetch) und in ein **Spezialregister** abgelegt
 - Bits im Register **kontrollieren** die folgenden Aktionen
 - Hole (fetch) die nächste Anweisung und fahre fort...

Rechnerarchitektur

3

Directives are bitstrings

A program is kept in memory
read and written like a regular datum

Instructions/Directives are fetched from the memory and a special register stored

Bits in a Register control the following actions

Fetch the next directive and so on

Modell eines Rechners

- ❑ Grundbestandteile eines Rechners
 - Zentraleinheit (Central Processing Unit, CPU)
 - Speicher
 - Ein-/Ausgabeeinheiten
- ❑ Problem-unabhängige Rechnerstruktur
 - Für jedes neue Problem wird ein eigenes Programm im Speicher abgelegt.
 - Programm-gesteuerter Universalrechner
- ❑ Speicher für Programme und Daten
 - besteht aus Plätzen fester Wortlänge
 - Ansprechen über Adressen

Rechnerarchitektur

4

Calculator Model

Components of a calculator/computer

central unit: CPU

memory

I/O devices

Problem-independent Computer structure

for every new problem its own program is stored in memory

program-controller universal computer

Memory for programs and data

consist of places fixed at word length

accessed via an address

Kennzeichen eines von-Neumann-Rechners

- ❑ Bearbeitung eines speziellen Problems erfolgt durch ein Programm (Befehlsfolge).
- ❑ Befehl: Binärzahl mit festem Format
- ❑ Daten und Programme werden nicht in getrennten Speichern untergebracht, ohne weitere Massnahmen besteht kein Schutz vor inkorrektem Zugriff.
- ❑ Alle Speicherworte können als Daten, Befehle oder Adressen verwendet werden.
- ❑ Zu jedem Zeitpunkt führt die CPU genau einen Befehl aus, welcher höchstens einen Datenwert bearbeiten kann (Single Instruction Single Data, SISD).

Rechnerarchitektur

5

Features of a von neumann's machine

Processing of a specific problem succeeds through a program (a set of instructions)

Instruction: Binary values with a fixed format

Data and programs are not placed in separate memories

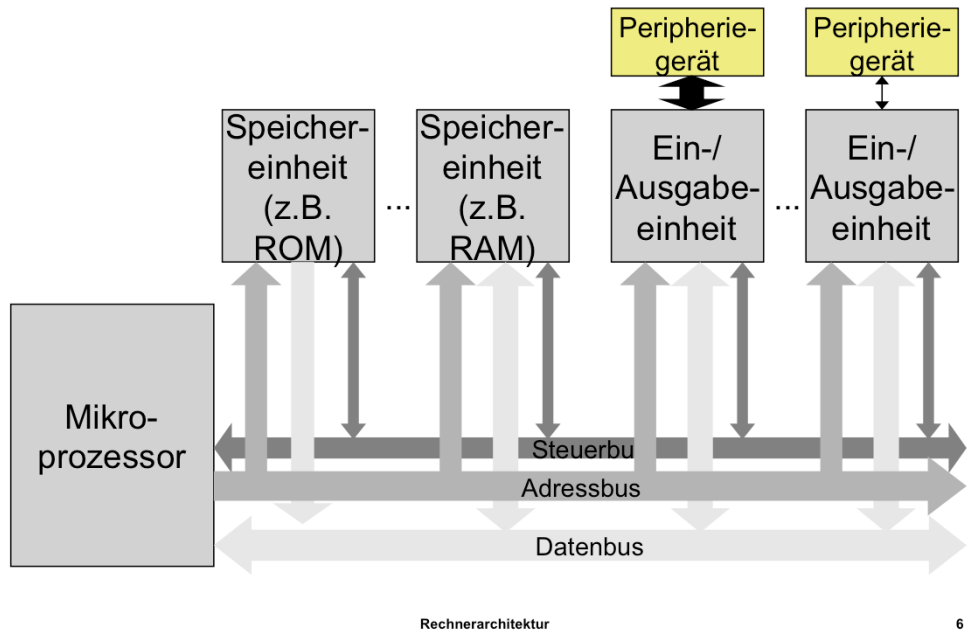
without further action it does not obtain any protection from incorrect access

All memory words could be equally used as Data, Instructions or addresses

At every time instant the CPU runs exactly one instruction, which can process at best one datavalue

Oberschelp, S. 204

Struktur eines von-Neumann-Rechners



Structure of a von Neumann machine

Microprocessor, memory unit, ... Peripheral device (i/o unit)

Control bus, address bus, data bus

Flik, S. 23

Rechnerkomponenten

□ CPU (Mikroprozessor)

- Verarbeiten von Daten durch ein Programm
- Steuerwerk
 - Lesen / Interpretieren von Befehlen und Operanden
 - Ablaufsteuerung
 - Ausführung von Befehlen
 - Ansteuerung der Ein-/Ausgabeeinheiten und des Hauptspeichers
- Operationswerk
 - Zwischenspeicherung
 - logische u. arithmetische Operationen

□ (Haupt-/Arbeits-)Speicher

- Speichern von Daten und Programmen

□ Ein-/Ausgabeeinheiten

- Schnittstelle zwischen Mikroprozessor und Peripheriegeräten
- Einlesen/Ausgabe von Daten von/an Peripheriegeräte
- Anpassung der Formate und Geschwindigkeiten bei Datenübertragung
- passiver Interface-Baustein/ Prozessor, Register

Computer components

CPU (microprocessor)

processing of data through a program

controller (computer control unit)

reading/interpretation of instructions and operands

workflow control

execution of instructions

control of I/O units and of the main memory

operational unit

caching/buffer storage

logical and arithmetic operations

main / working memory

memorization of data and programs

I/O units

Interface between microprocessor and peripheral devices

I/O of data from/to peripheral devices

matching of formats and speed of data transfer

Busse

- Bus = Verbindungsweg zwischen Systemkomponenten
- Bündel von funktional zusammengehörenden Signalleitungen, auf denen fest formatierte Bitfolgen transportiert werden müssen
 - seriell (1-Bit-Leitung, billige Lösung)
 - parallel (parallele Übertragung mehrerer Bits)
- Informationsarten
 - Daten
 - Adressen
 - Steuersignale
- Zusammenfassung von Daten-, Adress- und Steuerbus zu Systembus
- Busse können mehrere Rechnerkomponenten verbinden → Synchronisation erforderlich
- Mikroprozessor (CPU) als aktive Komponente (Master) steuert Bussystem.
- Speicher und E/A-Einheiten sind in der Regel passiv (Slaves).
- Zwei Systemkomponenten werden gleichzeitig auf den Bus geschaltet (je ein Sender und Empfänger).
- synchrone oder asynchrone Arbeitsweise

Rechnerarchitektur

8

Buses

Bus = connection between components of the system

Bundle of functional signal lines with similar functionality

over which a fixed format bit sequence is transferred

serial (1-Bit line, cheap solution)

parallel (parallel transfers more bits)

Types of information

data

addresses

control signals

Bundle together data, addresses, control and system buses

Buses could connect more computer components

synchronization required

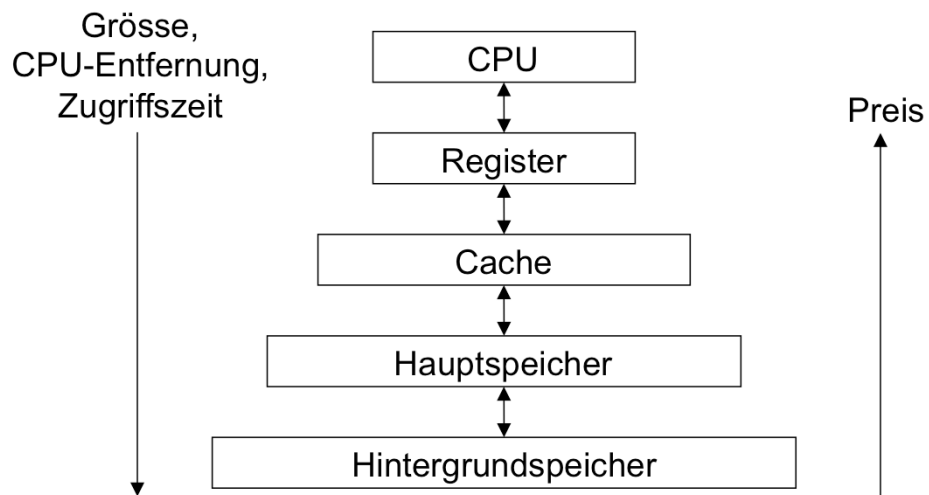
CPU as active component (master) controls the bus system

Memory and I/O units are typically passive (slaves)

Two components of the system become simultaneously active on the bus (a sender and a receiver)

Working synchronously or asynchronously

Speicherhierarchie



Rechnerarchitektur

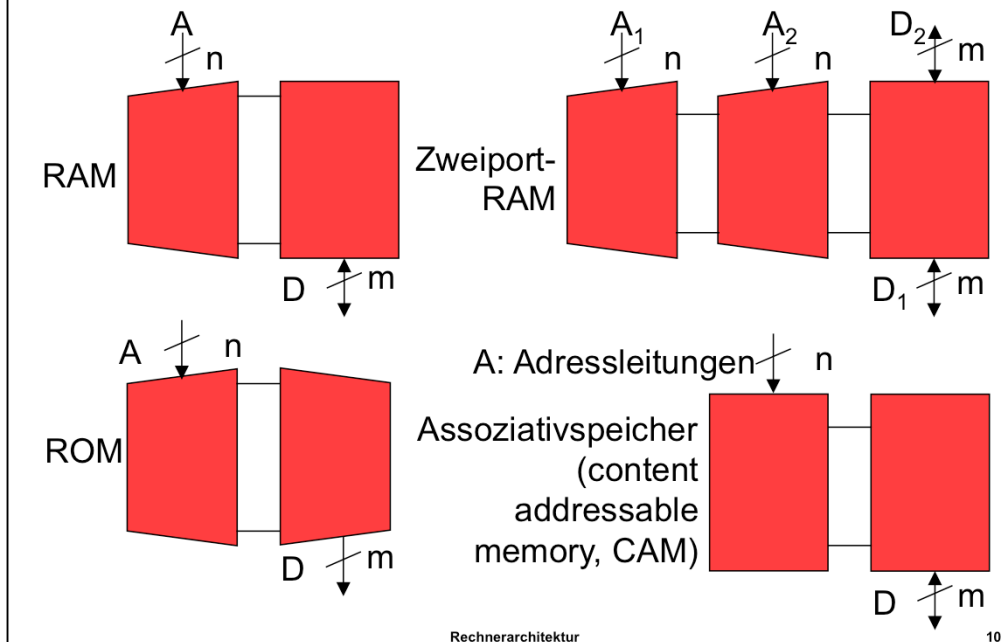
9

Great CPU distance, time of access
Price

Main memory
Background memory

Oberschelp, S. 212

Speichersymbole



Memory symbols

RAM: provide address, get/write word at address

ROM: provide address, get word at address

Two-port RAM: allows read and write at the same time

CAM: used to search a word in the entire memory

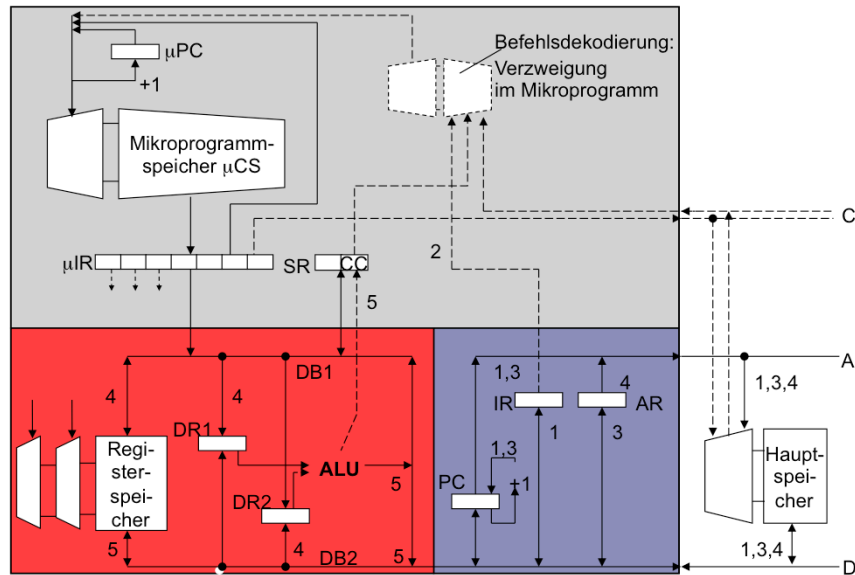
Application: e.g., routers (ternary CAM compares 0s, 1s and X (don't care))

Flik, S. 20

Zweiport (Multiport) RAM: Früher Grafikspeicher (der eine Port war ReadOnly).

Heute wichtig in Registerspeicher: Flik p51: gleichzeitig zwei Operanden holen und das Resultat der Vorvorherigen Operation schreiben (RISC Pipeline).

Prozessorstruktur



Rechnerarchitektur

11

CPU instruction sets were hardwired.

Each step needed to fetch, decode, and execute the machine instructions (including any operand address calculations, reads, and writes)

was controlled directly by combinational logic and rather minimal sequential state machine circuitry.

Hard-wired processors very efficient, but difficult to design and debug

Microcodes are a simpler method of developing the **control logic** for a computer.

Structure of the processor

Instruction coding: branching in the microprogram

PC program counter

SR status register

IR instruction register

AR address register

Mikroprozessor

Operationswerk

□ Rechenwerk

- Aufgabe: Ausführen von Berechnungen
- Registerspeicher (2-Port-Speicher, für Zwischenergebnisse)
- Operandenregister DR1, DR2
- 2 unabhängige interne Datenbusse DB1, DB2 (mit D verbunden)
- Prozessorstatusregister SR (Overflow, Carry)
- Arithmetic and Logical Unit (ALU)
 - verknüpft die zu Ausführungsbeginn in DR1 und DR2 geladenen Operanden
 - erzeugt Resultat u. Statusinformation (Condition Code für Programmverzweigungen)
 - Quelle und Ziel von Datentransporten: Haupt- oder Registerspeicher

Steuerwerk

- Festwertspeicher für Befehlsdekodierung (Erzeugen von Mikroprogramm-Startadressen)
- Mikroprogramm-speicher (μ CS)
- Mikrobefehlszähler (μ PC)
- Mikrobefehls-register (μ IR)

Leitwerk

- Befehlszähler (PC)
- Instruktionsregister (IR)
- Adressregister (AR)

Rechnerarchitektur

12

Flik, S. 26-28

Operation mode

Arithmetic unit

Task: Perform calculations

Register memory (2-Port memory, for intermediate results)

Operand register DR1, DR2

2 independent internal data buses
DB1, DB2 (connected to D)

Processor status register SR (overflow, carry)

Arithmetic and Logical Unit (ALU)

Linked to the state of the DR1 und DR2
operands at the start of the execution

Mikroprozessorbefehlssatz

- ❑ Programm (im Hauptspeicher) legt die Funktionsweise eines Mikroprozessorsystems fest.
 - Folge von Binärzahlen nach festem Format (Maschinencode), schwer lesbar
 - benutzerfreundliche Darstellung: Assemblersprachen mit speziellem Mnemocode für jeden Befehl
- ❑ Befehle für
 - Datentransport
 - arithmetische und logische Verknüpfungen
 - Änderung der Abarbeitungsreihenfolge
- ❑ Befehlssatz legt Art der möglichen Befehle fest.
 - Complex Instruction Set Computer (CISC)
 - Beispiele: Intel i386, Motorola MC680x0, MIPS R3000, Intel Pentium
 - Reduced Instruction Set Computer (RISC)
 - Beispiele: Sun (Ultra)SPARC, DEC Alpha, Motorola PowerPC, MIPS R10000

Rechnerarchitektur

13

Flik, S. 23

Program (in main memory) specifies the operation of a microprocessor system.

Sequence of binary numbers for fixed-format (machine code), difficult to read

user-friendly presentation: assembly languages with special Mnemocode for each command

Commands for

Data transport

Arithmetic and logic operations

Changing the order of execution

The instruction set defines the nature of the possible commands

Mikrobefehle (CISC)

- ❑ Befehlszähler enthält aktuelle Befehlsadresse.
- ❑ Inkrementieren oder Laden mit Sprungadresse
- ❑ Ablaufschritte (Mikrobefehlssequenzen) von Befehlszugriff und Befehlsausführung durch Mikroprogramm vorgegeben
- ❑ Durchlaufen von Mikrobefehlssequenzen
- ❑ Startadressen der Mikrobefehlssequenzen
 - hängen von Befehlscode, Bedingungsbits (CC) u. Steuersignale ab
 - werden durch Festwertspeicher der Befehlsdekodierung erzeugt
 - werden in Mikrobefehlsregister zur Ausführung geladen
- ❑ Adressierung der einzelnen Mikrobefehle durch Mikrobefehlsregister
- ❑ Mikrobefehl wirkt auf
 - Operationswerk (Durchschalten d. Datenwege, ALU-Operationen)
 - prozessorexterne Komponenten über Steuerleitungen

(Maschinen-)Befehlszyklus

- = Ablauf für Befehlszugriff und -ausführung
- ❑ Taktgenerator erzeugt Prozessor-(Maschinen-)takt (typische Taktfrequenz: mehrere 100 MHz / 1 GHz)
- ❑ Beispiel: zweistellige Operation
 1. Transport des Befehls vom Hauptspeicher in Befehlsregister, Erhöhen des PC
 2. Befehlsdekodierung
 3. Transport des 1. Operanden von Haupt- oder Registerspeicher in Operationswerk
 4. Transport des 2. Operanden in Operationswerk
 5. Operationsausführung (Verknüpfen der Operanden)
 6. Transport des Resultats vom Rechenwerk in Haupt- oder Registerspeicher

Rechnerarchitektur

15

Flik, S. 28-29

Machine (Instruction Cycle)

= run command for access and execution

Clock generator processor (machine); clock
(typical frequency: several 100 MHz / 1
GHz)

Example: two-term operation

1. Transport of the command from the main memory into the instruction register, increasing the PC
2. Instruction decoding
3. Transport of the 1st operand from main

Beispiel Befehlszyklus

□ ADD SPADR, R5

• PC → Hauptspeicher → IR, PC+1 → PC	• Lesen des 1. Befehlswords
• Befehlsdekodierung	• Auswerten Op-Code und Adressierungsarten
• PC → Hauptspeicher → AR, PC+1 → PC	• Lesen Adresse 1. Operand
• AR → Hauptspeicher → DR2, Registerspeicher → DR1	• Lesen der Operanden
• DR2 + DR1 → Registerspeicher, Statusinformation → SR	• Addition, Schreiben des Resultats in Registerspeicher u. Status-info nach SR (CC-Bits)

Rechnerarchitektur

16

Flik, S. 29-30

Bild auf Tafel

PC → Main memory → IR, PC+1 → PC

Instruction decoding

PC → Main memory → AR, PC+1 → PC

AR → Main memory → DR2,
Register memory → DR1

DR2 + DR1

→ Register memory, status information →
SR

CISC-Mikroprozessor

- ❑ in den 70er Jahren: Ausstattung der Prozessoren mit immer mächtigeren Befehlssätzen
 - Ziel: Verringern der semantischen Lücke zwischen höheren Programmiersprachen und einfachen Maschinenbefehlen
- ❑ typisch: > 200 Befehle
- ❑ grosse Anzahl von Adressierungsarten
- ❑ viele Kombinationen von Befehlen und Adressierungsarten
- ❑ Microcode für jeden Befehl in Steuerwerk
- ❑ Mikroprogrammierung des Steuerwerks ist langsamer als feste Verdrahtung.
- ❑ Versuch, CPU durch komplexe Instruktionen stärker zu belasten (Speicherbus als Flaschenhals)
- ❑ Viele Instruktionen und Adressierungsformen werden sehr selten verwendet.

Rechnerarchitektur

17

Oberschelp, S. 228

in the 70s: Equipping processors with increasingly powerful instruction sets

Objective: Reduce the semantic gap between high-level programming and simple machine instructions

typically > 200 commands

large number of addressing modes

many combinations of instructions and addressing modes

Microcode for each command in controller

Microprogramming of the controller is slower than hard-wiring

RISC-Mikroprozessor

- ❑ Reduced Instruction Set Computer
- ❑ Fixe Befehlslänge
- ❑ Wenige Befehle
- ❑ Wenige Addressierungsarten
- ❑ Optimiert, damit Compiler schnellen Code produzieren können

MIPS, Sun SPARC, HP PA-RISC, IBM PowerPC, Intel (Compaq) Alpha, ...

Oberschelp, S. 228

Reduced Instruction Set Computer

Fixed instruction length

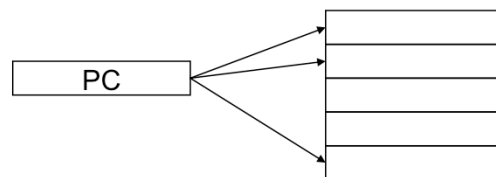
Fewer commands

Fewer addressing modalities

Optimized so that compilers can produce fast code

Befehlszähler

- ❑ Program Counter, PC
- ❑ enthält Adresse des nächsten Befehls
- ❑ Vielfaches von Bytes oder Halbworten
- ❑ Verändern des PC
 - Inkrementieren
 - Überschreiben bei Sprungbefehl



Rechnerarchitektur

19

Flik, S. 68-70

Instruction counter

Program Counter, PC

Stores address of the next instruction

Multiple of bytes or halfwords

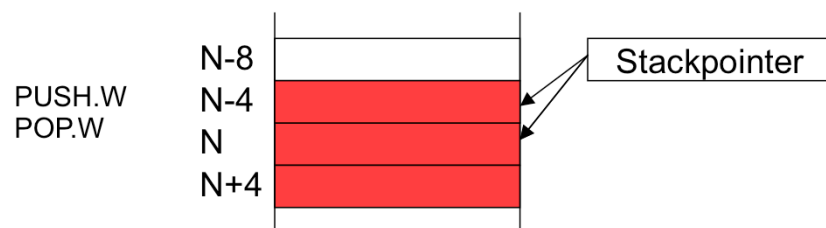
Change of the PC

Incrementing

Overwriting in jump instruction

Stackpointerregister

- ❑ Adressierung eines Keller-(Stapel-)speichers
- ❑ Stapелеlemente können nur oben aufgelegt und entnommen werden.
- ❑ Inkrementieren/Dekrementieren des Stackpointers
- ❑ Benutzung auch bei Unterprogrammssprüngen
- ❑ oft: getrennte Benutzer/System-Stacks (USP/SSP)



Rechnerarchitektur

20

Flik, S. 69

Addressing a base (stack) memory

Stack elements may be placed only at the top and removed

Increment/decrement of the stack pointer

Use also in subroutine jumps

Often: separate user/system stack (USP/SSP)