Kapitel 1

Computerarchitektur

und PC-Aufbau

Von-Neumann · Systembus · BIOS/UEFI · POST · Bootvorgang · Bundeswehr-Systeme
FiSi Systemintegration · Sabel Schulen München · Bundeswehr

Lernziele – Kapitel 1

#	Lernziel	Niveau	IHK-relevant?
1	Von-Neumann-Modell: 5 Komponenten beschreiben und erklären	Verstehen	✓ **Häufig**
2	Systembus gliedern: Adress-, Daten-, Steuerbus – Funktion erläutern	Verstehen	✓ Ja
3	BIOS vs. UEFI: Unterschiede, Vor-/Nachteile benennen	Analysieren	✓ **Sehr häufig**
4	MBR und GPT erklären und vergleichen (Partitionen, Größe)	Verstehen	✓ Ja
5	POST-Ablauf beschreiben und Fehlercodes/Pieptöne interpretieren	Anwenden	✓ **Häufig**
6	Harvard- und Von-Neumann-Architektur vergleichen und Einsatzgebiete nennen	Analysieren	○ Möglich

GBS Schulen München

Kapitelübersicht – Was uns in Kapitel 1 erwartet

01 Was ist ein Computer?

Definition · Computertypen · ca. 5 Min.

02 Von-Neumann-Architektur

5 Komponenten · Stored-Program Concept · Harvard-Vergleich · ca. 15 Min.

03 Systembus

Adressbus · Datenbus · Steuerbus · Busbreite · ca. 10 Min.

04 BIOS und UEFI

Firmware · Secure Boot · MBR vs. GPT · Konfiguration · ca. 10 Min.

05 POST und Bootvorgang

Selbsttest · Fehlercodes · Pieptöne · Bootsequenz · ca. 10 Min.

06 Bundeswehr-Kontext

Ruggedized · Fanless · MIL-STD-810 · Secure Boot BW · ca. 5 Min.

30 Folien · Input ca. 60 Min. · IHK AP Teil 1 relevant

01

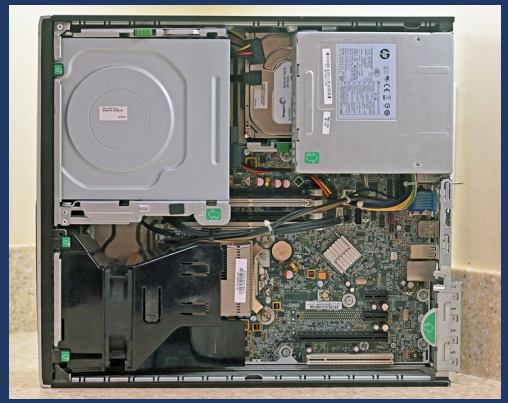


Bild: Bubba73, CC BY-SA 4.0, via Wikimedia Commons

Was ist ein Computer?

Definition · Computertypen · Einsatzbereiche

Was ist ein Computer? – Definition

Definition (IHK-konform)

Ein Computer ist ein elektronisches Gerät, das nach programmierten Anweisungen Daten verarbeitet, speichert und ausgibt – universell programmierbar und digital arbeitend.

Drei Grundeigenschaften:

Eigenschaft	Bedeutung	Beispiel
Programmierbarkeit	Verhalten durch Software steuerbar – nicht festverdrahtet	Gleicher PC: heute Office, morgen Bildbearbeitung
Universalität	Dasselbe Gerät löst verschiedenartige Aufgaben	PC: Texte schreiben, rechnen, Netzwerk verwalten
Digitale Verarbeitung	Alle Daten intern als Binärcode (0 und 1 = Bits)	Bild, Text, Audio – alles wird zu Bits

02

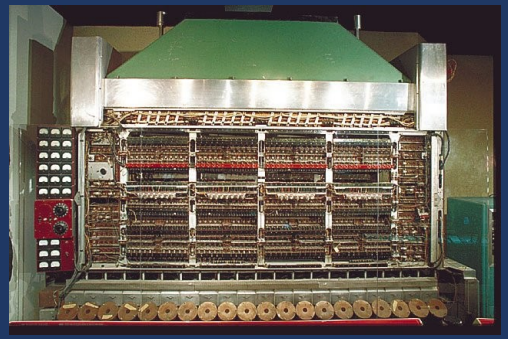
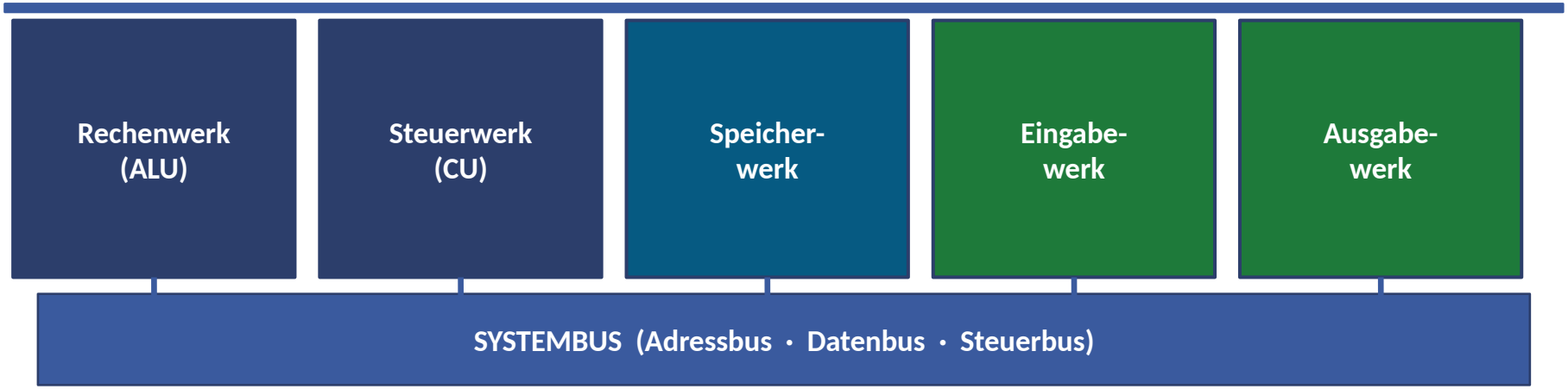


Bild: Smithsonian Institution, gemeinfrei, via Wikimedia Commons

Von-Neumann-Architektur

Das Grundmodell aller modernen Computer – 5 Komponenten

Das Von-Neumann-Modell – Die 5 Grundkomponenten



Merksatz:

Alle 5 Komponenten kommunizieren ausschließlich über den GEMEINSAMEN Systembus. Programme und Daten liegen im GLEICHEN Speicher (Stored-Program Concept).

Steuerwerk – Control Unit (CU)

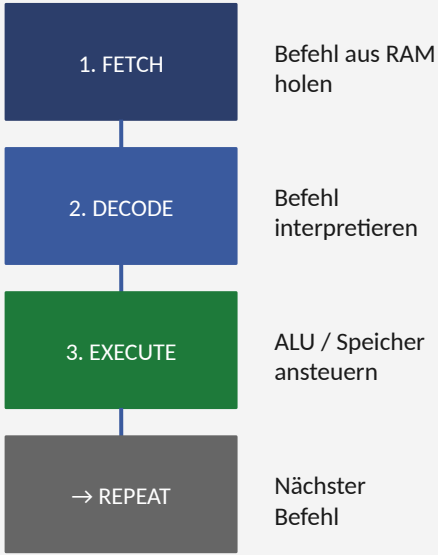
Steuerwerk – Control Unit (CU)

- Liest (fetcht) den nächsten Befehl aus dem Programmspeicher (RAM)
- Decodiert den Befehl (was soll getan werden?)
- Steuert ALU, Speicher, Ein-/Ausgabe mit Steuersignalen
- Verwaltet den Befehlszähler (Program Counter, PC)
- Verwaltet Interrupts (Unterbrechungsanforderungen)

Fetch-Decode-Execute-Zyklus:

1. Fetch: Befehl aus RAM holen → 2. Decode: Befehl interpretieren → 3. Execute: ALU ausführen lassen → (zurück zu 1.)

Befehlsausführungszyklus



Der Von-Neumann-Flaschenhals

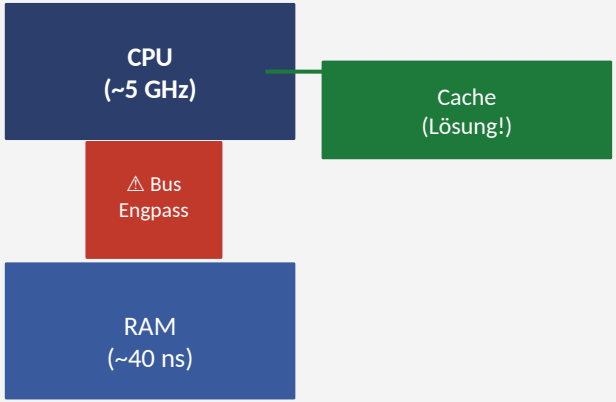
Problem: Von-Neumann-Flaschenhals

- CPU und RAM teilen DENSELBEN Bus für Befehle UND Daten
- CPU kann nur so schnell arbeiten wie Daten aus dem RAM ankommen
- RAM ist ~100× langsamer als die CPU (Taktfrequenz-Unterschied)
- Der Bus wird zum Engpass ('Flaschenhals') des Gesamtsystems
- Resultat: CPU wartet auf Daten – 'Wait States'

Lösungsansätze in modernen Systemen:

- Cache-Hierarchie L1/L2/L3: Puffer zwischen CPU und RAM
- Schnellere Speicherbusse: DDR5 im Dual-Channel (ca. 100 GB/s)
- Prefetching: CPU lädt Daten vor, bevor sie gebraucht werden

Visualisierung Flaschenhals



Cache L1/L2/L3 puffert Daten vor → CPU muss seltener auf langsamen RAM warten

GBS Schulen München

Harvard-Architektur – Die Alternative

Von-Neumann Allzweck-Computer	vs.	Harvard Aufgabenspezifische Systeme
Programm + Daten im gleichen Speicher		Getrennter Programm- und Datenspeicher
1 Bus für Befehle und Daten		2 separate Busse
Befehl und Daten nur nacheinander		Befehl und Daten parallel
Flaschenhals: Bus ist Engpass		Kein Flaschenhals
PCs, Server, Smartphones, Laptops		Mikrocontroller (AVR, PIC), DSP

Kernunterschied: gemeinsamer oder getrennter Speicher und Bus

03

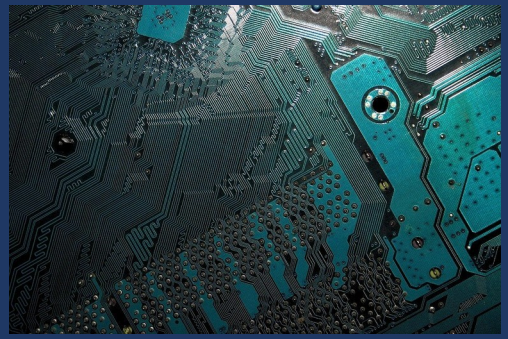


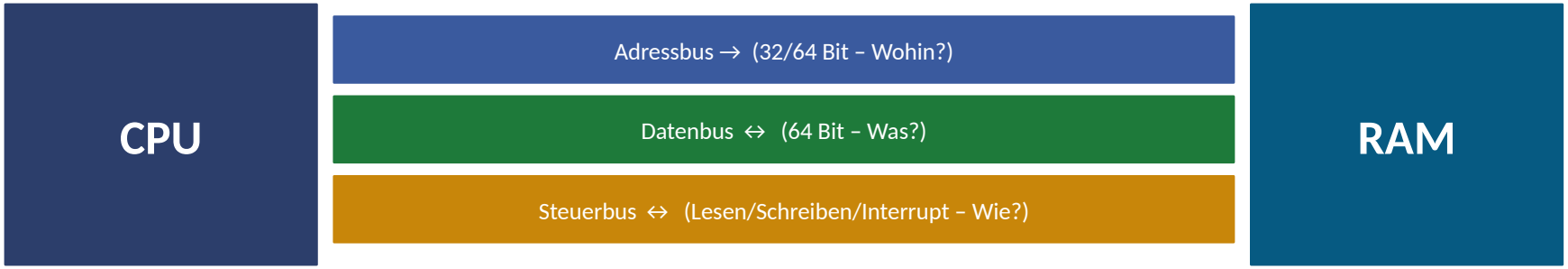
Bild: Chrihern, gemeinfrei, via Wikimedia Commons

Systembus

Adressbus · Datenbus · Steuerbus · Busbreite und Adressraum

Systembus – Drei Busse, eine Aufgabe

Bus	Richtung	Aufgabe	Beispiel (32-Bit)
Adressbus	CPU → Speicher (unidirektional)	Gibt an, WELCHE Speicheradresse gelesen/geschrieben wird	32 Bit: 2^{32} = 4 GB max. Adressraum
Datenbus	Bidirektional (beide Richtungen)	Überträgt die eigentlichen NUTZDATEN zwischen Komponenten	32 Bit: 4 Byte je Transfer
Steuerbus	Bidirektional	Überträgt Steuersignale: Lesen/Schreiben, Interrupts, Takt	Lese-Signal, Schreib-Signal, Interrupt



Adressbus – Wie viel Speicher ist adressierbar?

Adressbus – Details

- Unidirektional: Nur CPU → Speicher (nur eine Richtung)
- Breite bestimmt maximalen adressierbaren Speicher
- Formel: 2^n Byte (n = Busbreite in Bit)
- 32-Bit-Bus: 2^{32} = 4.294.967.296 Byte = 4 GB
- 64-Bit-Bus: 2^{64} = 16 Exabyte (theoretisch)

⚠ **Warum 32-Bit-Windows max. 4 GB RAM:**

32-Bit-Adressbus = max. 4 GB Adressen. Selbst mit 8 GB RAM kann ein 32-Bit-System nur 4 GB ansprechen. Deshalb: Alle neuen Systeme sind 64-Bit!

Busbreite	Max. Adressraum	Bemerkung
16 Bit	2^{16} = 64 KB	8086-Ära
20 Bit	2^{20} = 1 MB	x86 Real-Mode
32 Bit	2^{32} = 4 GB	Pentium-Ära, Win32
36 Bit	2^{36} = 64 GB	PAE (Physical Address Extension)
48 Bit	2^{48} = 256 TB	x86-64 (aktuell, physisch)
64 Bit	2^{64} = 16 EB	x86-64 (virtuell, theoretisch)

Vom Bit zum Byte – Datendarstellung

Einheit	Abkürzung	Größe (Dezimal / Binär)	Beispiel
Bit	b	1 Stelle (0 oder 1)	Kleinste Informationseinheit
Byte	B	8 Bit	1 ASCII-Zeichen
Kilobyte / Kibibyte	kB / KiB	1.000 B / 1.024 B	Kleine Textdatei
Megabyte / Mebibyte	MB / MiB	1.000.000 B / 1.048.576 B	MP3-Song ca. 4 MB
Gigabyte / Gibibyte	GB / GiB	10^9 B / 2^{30} B	Film ca. 4–8 GB
Terabyte / Tebibyte	TB / TiB	10^{12} B / 2^{40} B	HDD 1–4 TB

Merkhilfe: 1 Byte = 8 Bit. 1 KiB = 2^{10} = 1.024 Byte. 1 MiB = 2^{20} = 1.048.576 Byte. IHK-Hinweis: Kibibyte (KiB) ist die korrekte binäre Einheit – Kilobyte (kB) = 1.000 Byte (dezimal).

Hexadezimalsystem – Warum Techniker hex sprechen

Dezimal	Binär (4 Bit)	Hexadezimal	IT-Einsatz
1	0000	0	
2	0001	1	
3	1000	8	
4	1010	A	Hex-Ziffern A bis F = Dezimal 10 bis 15
5	1111	F	1 Hex-Ziffer = 4 Bit = 1 Nibble
6	11111111	0xFF	Beispiel: IP 192.168.1.1 = C0.A8.01.01 hex

Binärsystem – Wie Computer rechnen

Warum Binär? Transistoren haben nur 2 Zustände: AN (1) und AUS (0). Alle Informationen werden als Folge von 0 und 1 dargestellt – das nennt man Binärdarstellung.

Dezimal → Binär Umrechnung:

Dezimalzahl	Binärdarstellung	Stellenwerte	Rechenweg
5	0000 0101	$2^2 + 2^0$	4 + 1 = 5
10	0000 1010	$2^3 + 2^1$	8 + 2 = 10
42	0010 1010	$2^5+2^3+2^1$	32+8+2 = 42
127	0111 1111	$2^6+2^5+...+2^0$	64+32+16+8+4+2+1 = 127
255	1111 1111	$2^7+...+2^0$	128+64+32+16+8+4+2+1 = 255

04

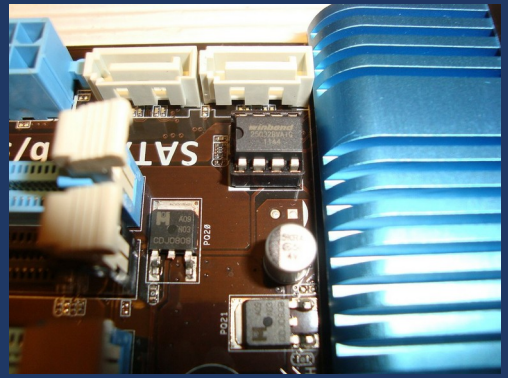


Bild: Mixabest, CC BY-SA 3.0, via Wikimedia Commons

BIOS und UEFI

Firmware · Aufgaben · Einstellungen · Secure Boot · MBR vs. GPT

UEFI Secure Boot – Sicherheitsmerkmal

Secure Boot prüft, ob der Bootloader digital signiert und in der UEFI-Datenbank als vertrauenswürdig eingetragen ist. Nicht signierte oder unbekannte Bootloader werden ABGELEHNT.

Aspekt	Beschreibung
Funktionsprinzip	UEFI prüft Signatur des Bootloaders gegen db (allowed) und dbx (revoked) Datenbanken
Plattform Key (PK)	Oberster Vertrauensanker – i. d. R. vom Mainboard-Hersteller (OEM)
Key Exchange Keys (KEK)	Microsoft und OEM-Schlüssel – dürfen db/dbx aktualisieren
db (Signature Database)	Liste erlaubter Bootloader-Signaturen (z.B. shim.efi für Linux)
dbx (Blacklist)	Widerrufene, kompromittierte Signaturen
Custom Mode	Eigene Schlüssel eintragen – für Organisationen wie BW möglich

Bundeswehr-Praxis

Typisch in großen Behördennetzen: Secure Boot ist per Richtlinie vorgeschrieben. Es verhindert, dass Soldaten unautorisierten Code (z.B. von privatem USB-Stick) booten. Zusammen mit TPM 2.0 und BitLocker sichert es das System bereits vor dem Betriebssystemstart.

MBR und GPT – Partitionierungsstandards

Merkmal	MBR (Master Boot Record)	GPT (GUID Partition Table)
Max. Partitionen	4 primäre (oder 3 primär + 1 erweitert mit logischen)	128 primäre Partitionen (Windows-Limit)
Max. Laufwerksgröße	2 TiB ≈ 2,2 TB (2^{32} Sektoren × 512 Byte)	ca. 8 ZiB (Zebibyte) – praktisch unbegrenzt
Redundanz	Kein Backup – Verlust = Datenverlust!	Header und Partitionstabelle doppelt gespeichert
Bootloader-Ort	Im MBR (erste 512 Byte des Laufwerks)	Separate EFI System Partition (ESP, 100–500 MB)
Kompatibilität	Alle BIOS-Systeme, Legacy-Modus bei UEFI	Windows bootet GPT nur per UEFI; als Datenlaufwerk auch an BIOS-PCs
Empfehlung	Nur für Altsysteme und externe Laufwerke < 2 TB	Standard für alle neuen Systeme und Laufwerke

GBS Schulen München

BIOS vs. UEFI – Der große Vergleich

BIOS (Legacy) Nur noch Altsysteme	vs.	UEFI (Modern) Standard auf neuen Systemen
Start im 16-Bit Real Mode		Start im 32-Bit Protected Mode bzw. 64-Bit Long Mode
MBR: max. 4 Partitionen, 2 TiB		GPT: 128 Partitionen, ca. 8 ZiB
Textbasiert, nur Tastatur		Grafisch, Maus, mehrsprachig
Langsamer Start		Schneller Start (Fast Boot)
Kein Secure Boot		Secure Boot (Signaturprüfung, PKI)

UEFI ist seit ca. 2011 Standard auf allen neuen Systemen

BIOS/UEFI – Praktische Konfiguration

Aufgabe	BIOS/UEFI-Taste	Menü-Pfad (typisch)	Bundeswehr-Relevanz
BIOS/UEFI öffnen	Del, F2, F10, Esc (je Hersteller)	Beim POST: Hinweis auf Bildschirm	Standard-Wartungsaufgabe
Boot-Reihenfolge	–	Boot → Boot Priority	PXE-Boot für Neuinstallation
Secure Boot aktivieren	–	Security → Secure Boot → Enabled	In Behördennetzen typisch Pflicht
TPM aktivieren	–	Security → TPM → Enabled	BitLocker-Voraussetzung
XMP/EXPO aktivieren	–	Advanced → Memory → XMP Profile 1	RAM-Takt aktivieren
Virtualisierung aktivieren	–	Advanced → CPU → Intel VT-x oder AMD-V	Für VMs (Hyper-V, VirtualBox)

05



Bild: Hans Haase, CC BY-SA 3.0, via Wikimedia Commons

POST und Bootvorgang

Power-On Self Test · Fehlercodes · Pieptöne · Bootsequenz

POST – Power-On Self Test

POST (Power-On Self Test) ist die automatische Hardware-Selbstprüfung durch BIOS/UEFI beim Einschalten – vor dem Betriebssystem-Start. Fehler werden durch Pieptöne, LED-Codes oder Q-Code-Displays signalisiert.

Was POST prüft:

Komponente	Was wird geprüft?	Fehlerreaktion bei Problem
CPU	Grundfunktion des Prozessors	Kein Start, POST hält an
RAM	Speicherzellen auf Lese-/Schreibfehler	z. B. 3 kurze Pieptöne (AMI), herstellerabhängig
Grafikkarte	Grundfunktion, Videoausgabe	1 langer + 2 kurze Pieptöne
Tastatur	Anschluss und Grundfunktion	Kontinuierlicher Piepston
CMOS/NVRAM	Gespeicherte Konfiguration lesbar	Warnung: CMOS-Batterie leer?
Bootgerät	Vorhanden und bootfähig?	Kein bootfähiges Medium gefunden

POST-Fehlercodes – Pieptöne und Q-Codes

Pieptone / Code	BIOS-Typ	Bedeutung	Maßnahme
1 kurz	AMI/Award	POST erfolgreich – kein Fehler	Normal, kein Handlungsbedarf
1 lang + 2 kurz	AMI/Award	Grafikkarten-Fehler	GPU prüfen, PCIe-Slot kontrollieren
3 kurz	AMI	RAM – Lese-/Schreibfehler	RAM tauschen oder Slot wechseln
Kein Ton, kein Bild	–	CPU nicht erkannt / Mainboard-Defekt	CPU-Sitz prüfen, Mainboard prüfen
Kontinuierlich	AMI	Tastaturfehler oder schwerwiegender Fehler	Tastatur prüfen oder Mainboard
Q-Code: A2	UEFI (MSI)	SATA-Laufwerk nicht erkannt	SATA-Kabel, Laufwerk prüfen

06



Bild: Museum of Transport and Technology, CC BY 4.0, via Wikimedia Commons

Bundeswehr-Kontext

Ruggedized Systems · Fanless · MIL-STD-810 · IT-Anforderungen

IT-Systeme der Bundeswehr – Anforderungen

Ruggedized (MIL-STD-810)
Verstärktes Gehäuse, Vibrationsdämpfung, IP-Schutzklasse

Fanless Design
Passiv gekühlte CPUs mit niedriger TDP (ca. 15 W)

Secure Boot + TPM 2.0
UEFI mit Secure Boot, TPM 2.0 – Schutz vor Bootkits

ECC-RAM
Fehlerkorrektur; ECC-fähiges Mainboard (Server-Klasse)

Redundante Netzwerkkarte
Dual-Port-NIC oder zwei separate NICs

Lange Verfügbarkeit
Industrielle Mainboards, 5–10 Jahre lieferbar

Typische Anforderungen und ihre Auswirkung auf die Hardware-Auswahl

07

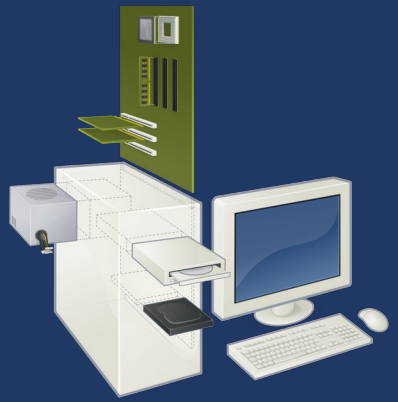
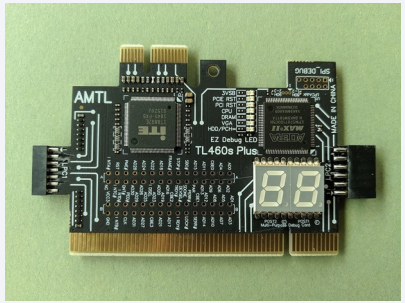


Bild: Gustavb, CC BY 2.5, via Wikimedia Commons

Zusammenfassung

Hardware Kap. 01 · Computerarchitektur und PC-Aufbau

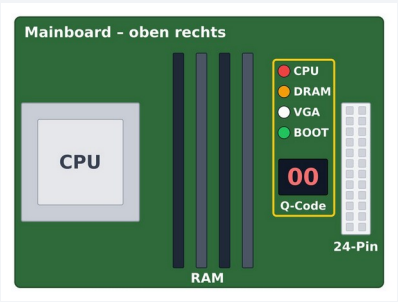
Bauteile erkennen – Was ist was?



POST-Karte

Debug-LEDs + Code-Anzeige zum Einstecken

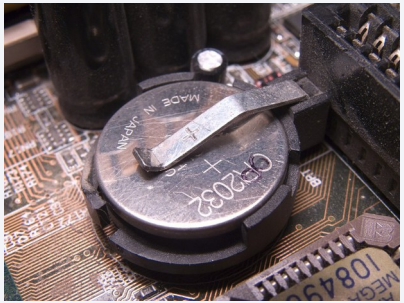
Bild: Markus Kuhn, CC0



Debug-LEDs und Q-Code

meist oben rechts neben RAM und 24-Pin

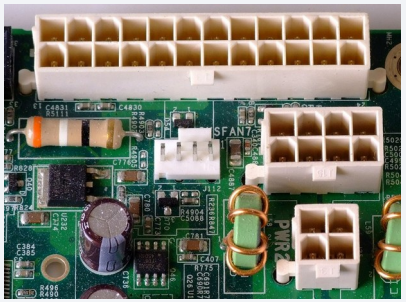
Eigene Darstellung



CMOS-Batterie

CR2032: hält Uhr und UEFI-Einstellungen

Bild: Lukas A, CZE, gemeinfrei



ATX-Strom am Board

24-Pin Mainboard, 8- und 4-Pin CPU

Bild: Tobias Maier, CC0

Kapitel 1 – Zusammenfassung aller Kernthemen

Thema	Kernaussage	IHK-Relevanz
Von-Neumann-Modell	5 Komponenten über Systembus – Stored-Program Concept	★★★ Sehr hoch
Systembus	Adressbus (Wo?), Datenbus (Was?), Steuerbus (Wie?)	★★ Hoch
Busbreite	Adressbus 32 Bit = 4 GB | 64 Bit = 16 EB | Formel: 2^n	★★★ Sehr hoch
BIOS	16-Bit, MBR, max. 4 Partitionen, max. 2 TiB, kein Secure Boot	★★★ Sehr hoch
UEFI	64-Bit, GPT, 128 Partitionen, Secure Boot, grafisch, schnell	★★★ Sehr hoch
MBR vs. GPT	MBR: 4 Partitionen, 2 TiB | GPT: 128 Partitionen, ca. 8 ZiB	★★★ Sehr hoch
POST	Selbsttest beim Start: CPU → RAM → GPU → Keyboard → Boot	★★ Hoch
POST-Fehlercodes	1 lang + 2 kurz = GPU, 3 kurz = RAM (AMI), kein Ton = CPU/Board	★★ Hoch

Kapitel 1 abgeschlossen

Computerarchitektur und PC-Aufbau

Von-Neumann · Systembus · BIOS/UEFI · POST · Bootvorgang · BW-Systeme
30 Folien · Kapitel 1 von 10 · Modul: Hardware · LF 3

Thomas Seuß – Sabel Schulen München · FiSi Systemintegration · Bundeswehr · IHK-Externenprüfung § 45 BBiG