Hardware Kapitel 1 (Computerarchitektur und PC-Aufbau) – Zusammenfassung mit IHK-Bezug

Von-Neumann, Systembus, Zahlensysteme, Einheiten, RAM, BIOS/UEFI, MBR/GPT, Secure Boot, TPM, POST

💻 Hardware Kapitel 1 – Zusammenfassung mit IHK-Bezug

📚 Grundlage: Präsentation "HW Kap. 01 Computerarchitektur" und Lernskript Hardware, Kapitel 1. Die Zusammenfassung hat sieben Blöcke.

🧭 So liest du dieses Dokument

- 🔑 **Fette** Begriffe sind Prüfungs-Schlagwörter. Du solltest sie erklären können.
- 🎯 Jeder Abschnitt hat eine Zeile "IHK-Relevanz". Das ist eine eigene Einschätzung. Es ist keine amtliche Angabe.
- 📝 Hardware passt laut Verordnung vor allem zu zwei Prüfungsteilen:
  - **AP1** (Teil 1, "Einrichten eines IT-gestützten Arbeitsplatzes"): Hard- und Software auswählen. Arbeitsplatz konfigurieren und testen.
  - **AP2 Konzeption und Administration**: Speicherlösungen integrieren und verwalten. Systeme betreiben.
- 🔢 Zahlensysteme brauchst du später für Subnetting und IPv6 (AP1 und **AP2 Netzwerke**).
- ❓ Am Ende jedes Blocks stehen Übungsfragen. Die Lösungen stehen gesammelt am Ende.
- ✅ Alle Rechenbeispiele sind nachgerechnet. Wo die Quellen ungenau sind, steht "Korrektur".

✏️ Schreibweise: `2^10` heißt 2 hoch 10. Das Komma trennt Dezimalstellen. Der Punkt trennt Tausender.

🧠 Block 1: Der Computer und die Von-Neumann-Architektur

1.1 🤖 Was ist ein Computer?

🎯 IHK-Relevanz: gering bis mittel – Definitionen werden selten einzeln abgefragt, sind aber Basis für Architekturfragen in AP1.

Stell dir einen Koch mit Rezeptbuch vor. Das Rezept sagt ihm Schritt für Schritt, was er tun soll. Ein Computer arbeitet genauso.

💡 **Definition:** Ein Computer ist eine Maschine. Sie verarbeitet Daten automatisch nach einem **gespeicherten Programm**. Programm und Daten liegen im selben Speicher.

Die Definition hat drei Teile:

- 📄 **gespeichertes Programm**: Die Arbeitsanweisungen liegen als Daten im Speicher. Man kann sie austauschen.
- ⚙️ **automatisch**: Der Computer arbeitet ohne Eingriff. Er erledigt einen Befehl nach dem anderen.
- 🔄 **Daten verarbeiten**: Eingabe, Berechnung, Zwischenspeicherung, Ausgabe.

Grundeigenschaft	Bedeutung	Beispiel
Programmierbarkeit	Software steuert das Verhalten. Nichts ist fest verdrahtet.	derselbe PC: heute Office, morgen Bildbearbeitung
Universalität	Ein Gerät löst verschiedene Aufgaben.	Texte schreiben, rechnen, Netzwerk verwalten
Digitale Verarbeitung	Alle Daten liegen intern als Bits vor (0 und 1).	Bild, Text und Ton werden zu Bits

🖥️ **Computertypen** unterscheiden sich in Größe, Leistung, Robustheit und Anschlüssen. Die Grundarchitektur ist bei allen gleich.

Typ	Kennzeichen	Beispiele
Server	Dauerbetrieb, redundant, gut skalierbar	Rack-Server, Tower-Server, Mainframe
Workstation	hohe Rechenleistung, oft Spezial-Hardware	CAD-PC, Renderknoten
Desktop-PC	Standard-Arbeitsplatz	Office-PC, BWI-Standard-Client
Notebook	mobil, mit Akku	Dienst-Notebook, Ultrabook
Embedded System	in ein größeres Gerät eingebaut	Steuerungen, Sensorik, Funkgeräte
Mobilgerät	Smartphone, Tablet	Diensthandy

🕰️ Meilensteine:

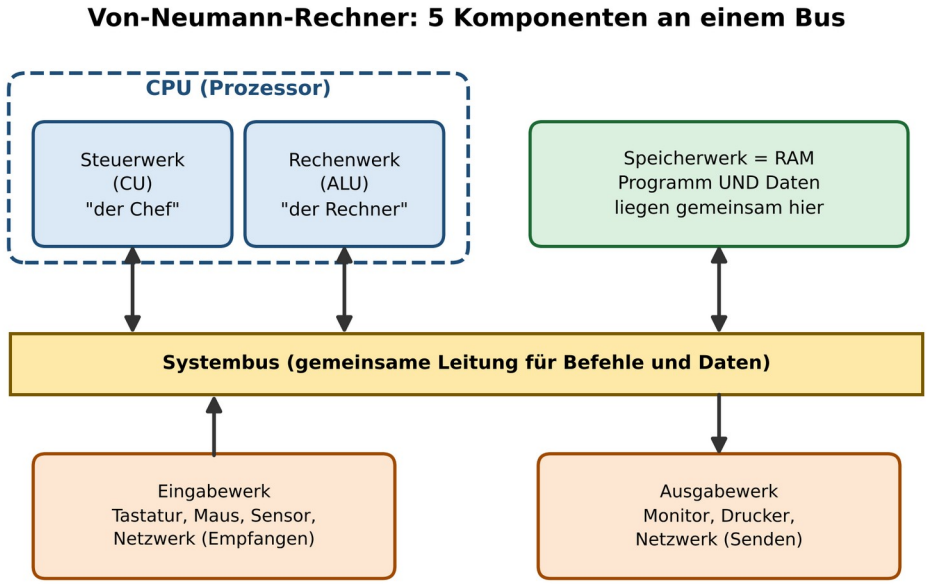
- 1941: Zuse Z3. Erster funktionsfähiger programmierbarer Rechner.
- 1945: Von-Neumann-Architektur.
- 1971: Intel 4004. Erster kommerzieller Mikroprozessor.
- 1981: IBM PC.
- 2003: AMD Opteron (x86-64).

1.2 🧩 Die fünf Komponenten nach Von Neumann

🎯 IHK-Relevanz: hoch – die Von-Neumann-Architektur ist ein klassisches Grundlagenthema; Zuordnungsaufgaben (“Welche Komponente macht was?”) sind gut prüfbar in AP1.

1945 beschrieb John von Neumann ein neues Rechnerkonzept. Seine Idee: Programm und Daten liegen gemeinsam im Arbeitsspeicher. Vorher war das anders. Rechner wie der ENIAC mussten für jede Aufgabe neu verdrahtet werden.

🏢 Vergleich mit einer Firma: Der Chef verteilt Aufgaben (Steuerwerk). Der Buchhalter rechnet (Rechenwerk). Das Archiv bewahrt alles auf (Speicherwerk). Die Post bringt Briefe rein (Eingabe) und raus (Ausgabe). Alle nutzen denselben Flur (Systembus).



Die fünf Komponenten der Von-Neumann-Architektur am gemeinsamen Systembus

Komponente	Aufgabe	Heute
Steuerwerk (Control Unit, CU)	holt und dekodiert Befehle, steuert den Ablauf	Teil der CPU
Rechenwerk (ALU, Arithmetic Logic Unit)	rechnet und vergleicht	Teil der CPU
Speicherwerk	speichert Programm und Daten	Arbeitsspeicher (RAM)
Eingabewerk	nimmt Daten von außen auf	Tastatur, Maus, Sensor, Netzwerk
Ausgabewerk	gibt Ergebnisse nach außen	Monitor, Drucker, Lautsprecher, Netzwerk

🔗 Alle fünf Komponenten hängen am gemeinsamen **Systembus**.

📌 **Merksatz:** Steuerwerk und Rechenwerk bilden zusammen die **CPU**. Das Speicherwerk ist der RAM.

🧮 Das Rechenwerk (ALU) im Detail

Die ALU ist der Taschenrechner der CPU. Sie kann:

- ➕ rechnen: Addition, Subtraktion, Multiplikation, Division.
- 🔀 logisch verknüpfen: AND, OR, NOT, XOR.
- ⚖️ vergleichen und schieben. Schieben heißt **Schiebeoperation** (Shift).

Woher kommen die Zahlen? Aus kleinen Speicherplätzen in der CPU. Diese heißen **Register**. Das Ergebnis geht wieder in ein Register.

Die ALU setzt außerdem Merker. Diese heißen **Status-Flags**:

- Zero: Das Ergebnis ist 0.
- Carry: Es gab einen Übertrag.
- Overflow: Es gab einen Überlauf.

Beispiel 1: Links-Shift verdoppelt

	Binär	Dezimal
Ausgangswert	`0000 0101`	5
1 Bit nach links	`0000 1010`	10
1 Bit nach rechts (vom Ausgangswert)	`0000 0010`	2

⚠️ Beim Rechts-Shift fällt das letzte Bit heraus. Deshalb ergibt 5 halbiert nur 2. Der Rest geht verloren.

🎛️ Das Steuerwerk im Detail

Das Steuerwerk ist der Chef. Es:

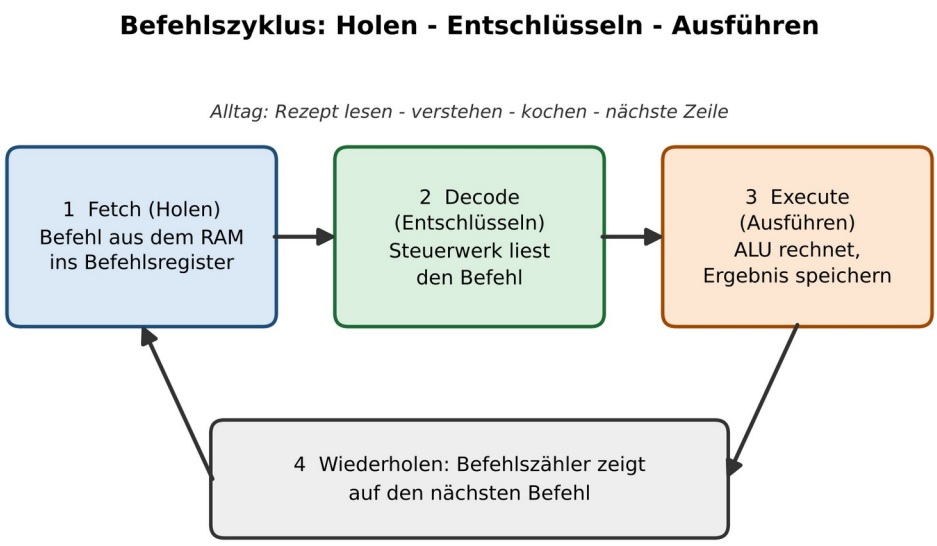
- holt den nächsten Befehl aus dem RAM und dekodiert ihn.
- steuert ALU, Speicher und Ein-/Ausgabe mit Steuersignalen.
- führt einen Zeiger auf den nächsten Befehl. Dieser Zeiger heißt **Befehlszähler** (Program Counter, PC). Er enthält die Adresse des nächsten Befehls.
- reagiert auf Unterbrechungen, z. B. von der Tastatur. Diese heißen **Interrupts**.

🌐 Sonderfall Netzwerkkarte (**NIC**): Sie gehört zu beiden E/A-Werken. Beim Empfangen ist sie Eingabewerk. Beim Senden ist sie Ausgabewerk.

1.3 🔁 Der Befehlszyklus: Fetch – Decode – Execute

🎯 IHK-Relevanz: mittel – der Ablauf in drei Schritten ist eine gute Kurzfrage für AP1.

🍳 Wie beim Kochen nach Rezept: Zeile lesen, verstehen, ausführen, nächste Zeile. Die CPU macht das milliardenfach pro Sekunde.



Befehlszyklus Fetch, Decode, Execute mit Wiederholung

Schritt	Name	Was passiert
1	**Fetch** (Holen)	Die CPU lädt den nächsten Befehl aus dem RAM in das Befehlsregister.
2	**Decode** (Dekodieren)	Das Steuerwerk entschlüsselt den Befehl. Es ermittelt die Operanden.
3	**Execute** (Ausführen)	Die ALU führt den Befehl aus. Das Ergebnis geht in ein Register oder den Speicher.
4	Wiederholen	Der Befehlszähler zeigt auf den nächsten Befehl. Der Zyklus beginnt neu.

💡 **Stored-Program Concept**: Programm und Daten liegen im selben Speicher. Deshalb ist der Computer universell. Für ein neues Programm muss man die Hardware nicht umbauen.

1.4 🍾 Der Von-Neumann-Flaschenhals

🎯 IHK-Relevanz: mittel – Ursache und Gegenmaßnahmen (Cache) sind ein typisches Erklärthema.

🚗 Stell dir eine einspurige Brücke vor. Autos in beide Richtungen müssen warten. So ist es auch hier:

- Programm und Daten teilen sich **einen** Bus zur CPU.
- Es geht also immer nur ein Befehl **oder** ein Datum gleichzeitig.
- Dazu ist der RAM viel langsamer als die CPU.

- Die CPU muss warten. Diese Wartetakte heißen **Wait States**.

Beispiel 2: Wie lange wartet die CPU auf den RAM?

Werte aus der Präsentation: CPU-Takt 5 GHz, RAM-Zugriffszeit ca. 40 ns.

- Dauer eines CPU-Takts: 1 / 5.000.000.000 s = 0,2 ns.
- Takte pro RAM-Zugriff: 40 ns / 0,2 ns = **200 Takte**.

ℹ️ Die Präsentation sagt "ca. 100-mal langsamer". Mit den genannten Werten sind es rechnerisch 200 Takte. Wichtig ist die Größenordnung: Die CPU wartet sehr lange.

🛠️ Gegenmaßnahmen:

Maßnahme	Wirkung
Cache (L1, L2, L3)	schneller Zwischenspeicher zwischen CPU und RAM; L1 ist am kleinsten und schnellsten
schnellere Speicherbusse	z. B. DDR5 im Dual-Channel-Betrieb mit ca. 100 GB/s
Prefetching	Daten werden vorab geladen, bevor die CPU sie braucht
getrennte L1-Caches für Befehle und Daten	innen arbeitet die CPU damit wie eine Harvard-Architektur
Pipelining, Out-of-Order-Execution, Branch Prediction	die CPU arbeitet mehrere Befehle überlappend ab

🧺 Cache im Alltag: Du legst das Werkzeug, das du oft brauchst, direkt neben dich. So musst du nicht jedes Mal in den Keller.

1.5 ⚖️ Von-Neumann und Harvard im Vergleich

🎯 IHK-Relevanz: mittel – klassische Abgrenzungsfrage mit Tabelle; passt zu AP1.

Merkmal	Von-Neumann	Harvard
Speicher	ein Speicher für Programm und Daten	getrennter Programm- und Datenspeicher
Busse	ein Bus	zwei getrennte Busse
Zugriff	Befehl und Daten nacheinander	Befehl und Daten gleichzeitig
Flaschenhals	ja	nein
Einsatz	PCs, Server, Notebooks, Smartphones	Mikrocontroller (z. B. AVR, PIC), digitale Signalprozessoren (DSP)

⚠️ Nachteile von Harvard:

- Selbstmodifizierender Code ist nicht möglich.
- Beide Speicher müssen getrennt geplant werden.

🔀 Moderne CPUs sind **Hybride**. Nach außen arbeiten sie nach Von Neumann. Innen, auf Cache-Ebene, arbeiten sie nach Harvard.

📌 **Merksatz:** Von-Neumann = gemeinsamer Speicher und Bus. Harvard = getrennte Speicher und Busse.

❓ Übungsfragen Block 1

Ü1.1 Nennen Sie die fünf Komponenten der Von-Neumann-Architektur. Ordnen Sie jeder Komponente ein heutiges Bauteil zu.

Ü1.2 Beschreiben Sie den Befehlszyklus einer CPU in drei Schritten.

Ü1.3 Erklären Sie den Von-Neumann-Flaschenhals. Nennen Sie zwei Gegenmaßnahmen.

🚌 Block 2: Systembus und Adressraum

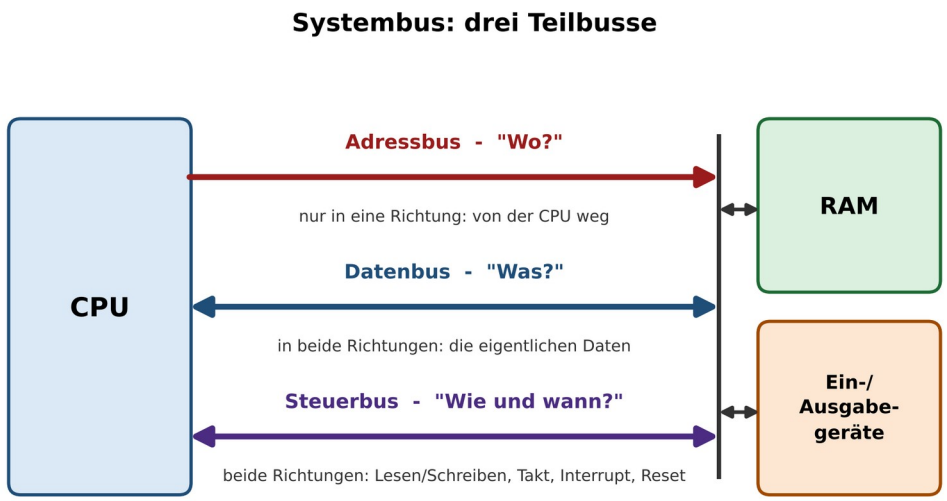
2.1 🛣️ Die drei Teilbusse

🎯 IHK-Relevanz: hoch – Adress-, Daten- und Steuerbus mit Richtung und Aufgabe sind eine typische Zuordnungsaufgabe in AP1.

Der **Systembus** verbindet CPU, Speicher und Peripherie. Physisch sind das Leiterbahnen auf dem Mainboard. Logisch besteht er aus drei Teilbussen.

📦 Vergleich mit einem Paketdienst:

- Die Adresse auf dem Paket sagt: **Wo** hin? Das ist der Adressbus.
- Der Inhalt des Pakets ist das **Was**. Das ist der Datenbus.
- Der Auftrag “abholen” oder “zustellen” und die Uhrzeit sind das **Wie und wann**. Das ist der Steuerbus.



Systembus mit Adressbus, Datenbus und Steuerbus und ihren Richtungen

Bus	Richtung	Aufgabe	Merkfrage
Adressbus	unidirektional, von der CPU weg	wählt die Speicheradresse oder das Gerät aus	Wo?
Datenbus	bidirektional	überträgt die eigentlichen Daten	Was?
Steuerbus	bidirektional	Steuersignale: Lesen/Schreiben, Takt, Interrupts, Reset	Wie und wann?

ℹ️ Unidirektional heißt: nur in eine Richtung. Bidirektional heißt: in beide Richtungen.

📏 Die **Datenbusbreite** sagt, wie viele Bits pro Takt gleichzeitig übertragen werden.

- 32-Bit-Datenbus: 32 / 8 = **4 Byte** je Transfer.
- 64-Bit-Datenbus: 8 Byte je Transfer.

2.2 🏠 Adressbusbreite und Adressraum

🎯 IHK-Relevanz: hoch – die Formel 2^n ist eine typische Rechenaufgabe mit Rechenweg; dazu die Erklärung der 4-GB-Grenze bei 32 Bit.

🏘️ Stell dir eine Straße vor. Jedes Haus braucht eine Hausnummer. Je mehr Ziffern die Hausnummer haben darf, desto mehr Häuser gibt es. Beim Adressbus ist jede Leitung eine “Ziffer”. Sie kann nur 0 oder 1 sein.

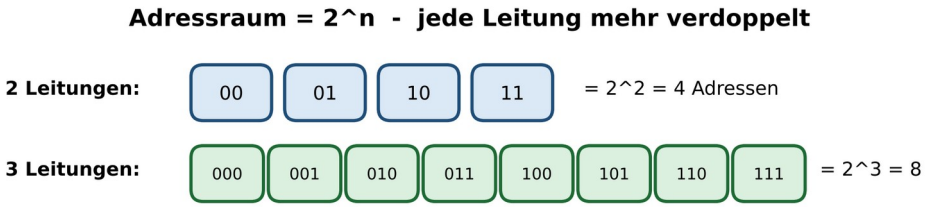
Die Breite des Adressbusses bestimmt also, wie viele Speicherzellen die CPU ansprechen kann.

📌 **Formel: Adressraum = 2^n Speicherzellen** (n = Breite des Adressbusses in Bit)

Bei PCs ist eine Speicherzelle 1 Byte groß. Dann gilt: Adressraum = 2^n Byte.

❔ Warum 2^n?

- Jede Adressleitung kann 0 oder 1 sein.
- Bei 2 Leitungen gibt es 4 Kombinationen: 00, 01, 10, 11.
- Jede weitere Leitung verdoppelt die Anzahl.



Busbreite n	2^n Byte	binär
16 Bit	65.536 Byte	**64 KiB**
20 Bit	1.048.576 Byte	**1 MiB**
32 Bit	4.294.967.296 Byte	**4 GiB**
64 Bit	18.446.744.073.709.551.616 Byte	**16 EiB**

Adressraum: jede zusätzliche Adressleitung verdoppelt die Zahl der Adressen

Busbreite n	2^n Byte	Binäre Einheit	Beispiel
16 Bit	65.536	64 KiB	8-Bit-Prozessoren wie Intel 8080
20 Bit	1.048.576	1 MiB	Intel 8086, x86 Real Mode
24 Bit	16.777.216	16 MiB	Intel 80286
32 Bit	4.294.967.296	4 GiB	32-Bit-Systeme, z. B. Windows 32 Bit
36 Bit	68.719.476.736	64 GiB	PAE (Physical Address Extension)
48 Bit	281.474.976.710.656	256 TiB	x86-64, virtuelle Adressen heute
64 Bit	18.446.744.073.709.551.616	16 EiB	theoretisches Maximum bei 64 Bit

✏️ Korrektur zu den Quellen: Dort steht teils “64 KB”, “4 GB” oder “16 EB”. Zweierpotenzen ergeben aber genau binäre Einheiten (KiB, GiB, EiB). Die Präsentation ordnet 16 Bit der “8086-Ära” zu. Der 8086 hatte aber einen 20-Bit-Adressbus.

Beispiel 3: Adressraum bei 24 Bit berechnen

1. Formel: 2^24.
2. Zerlegen: 2^24 = 2^20 × 2^4 = 1.048.576 × 16.
3. Ergebnis: **16.777.216 Byte = 16 MiB**.

Beispiel 4: Warum sieht ein 32-Bit-Windows nur 4 GiB?

1. 2^32 = 4.294.967.296 Byte.
2. 4.294.967.296 / 1.073.741.824 (= 2^30) = **4 GiB**.
3. Mehr Adressen gibt es nicht. Auch 8 GiB eingebauter RAM bleiben zur Hälfte ungenutzt.

⚠️ In der Praxis sieht ein 32-Bit-Windows oft nur etwa 3 bis 3,5 GiB. Ein Teil der Adressen ist für Geräte reserviert, z. B. für die Grafikkarte.

ℹ️ 64-Bit-CPUs nutzen heute nicht alle 64 Adressbits. Üblich sind 48 oder 52 Bit für virtuelle bzw. physische Adressen.

2.3 🔌 Moderne Punkt-zu-Punkt-Verbindungen

🎯 IHK-Relevanz: gering bis mittel – Datenraten helfen bei Auswahl- und Vergleichsaufgaben; exakte Werte werden selten verlangt.

Heute gibt es keinen einzelnen gemeinsamen Bus mehr. Die Komponenten haben eigene, schnelle Direktverbindungen. Diese heißen Punkt-zu-Punkt-Verbindungen. Das Bild aus Adress-, Daten- und Steuerbus bleibt aber das Denkmodell.

Verbindung	Datenrate (ca.)	Einsatz
PCIe 5.0 x16	64 GB/s je Richtung	Grafikkarte
DDR5 Dual-Channel	100 GB/s	Arbeitsspeicher
USB4	40 bzw. 80 Gbit/s	externe Geräte
Thunderbolt 5	80 Gbit/s, Boost bis 120 Gbit/s	Docks, Displays
SATA III	600 MB/s	HDD, SATA-SSD

⚠️ Achtung bei den Einheiten: **GB/s** (Byte) und **Gbit/s** (Bit) unterscheiden sich um den Faktor 8. Beispiel: 40 Gbit/s sind 40 / 8 = 5 GB/s.

❓ Übungsfragen Block 2

Ü2.1 Nennen Sie die drei Teilbusse des Systembusses. Geben Sie jeweils Richtung und Aufgabe an.

Ü2.2 Eine CPU hat einen 20-Bit-Adressbus. Berechnen Sie den maximalen Adressraum in Byte und in einer passenden binären Einheit.

Ü2.3 Ein Kunde hat 8 GiB RAM in einem PC mit 32-Bit-Windows. Er sieht nur etwa 3,5 GiB. Erklären Sie die Ursache.

🔢 Block 3: Zahlensysteme und Speichereinheiten

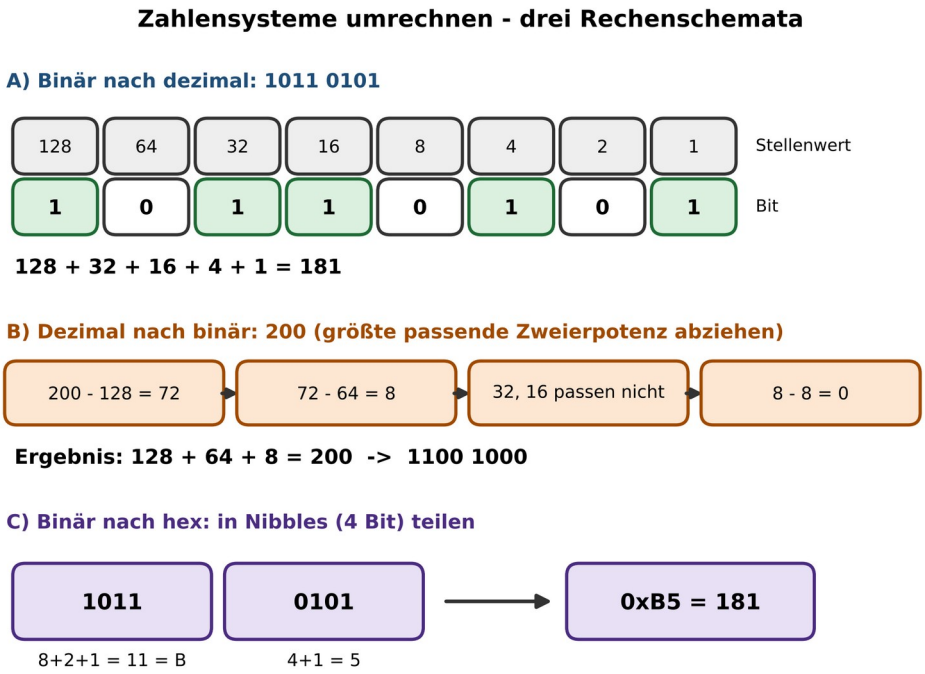
3.1 💡 Das Binärsystem

🎯 IHK-Relevanz: mittel bis hoch – Umrechnungen sind Grundlage für Subnetting und IPv6 (AP1, AP2 Netzwerke); Lernportale nennen Zahlensysteme als AP1-Thema.

💡 Ein Lichtschalter kennt nur “an” und “aus”. Ein Transistor auch: an (1) und aus (0). Deshalb rechnet der Computer mit nur zwei Ziffern. Das heißt **Binärsystem** (Basis 2). Eine Binärstelle heißt **Bit**.

Jede Stelle hat einen festen Wert. Dieser Wert ist eine Zweierpotenz. Von rechts nach links verdoppelt er sich:

Stelle	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Wert	128	64	32	16	8	4	2	1



Drei Rechenschemata: binär nach dezimal, dezimal nach binär, binär nach hex

Beispiel 5: Binär nach dezimal

`1011 0101` umrechnen. Regel: Alle Stellen mit einer 1 addieren.

Stelle	128	64	32	16	8	4	2	1
Bit	1	0	1	1	0	1	0	1
Wert	128	0	32	16	0	4	0	1

128 + 32 + 16 + 4 + 1 = **181**.

Beispiel 6: Dezimal nach binär (Subtraktionsverfahren)

200 umrechnen. Regel: Immer die größte passende Zweierpotenz abziehen.

💶 Wie beim Bezahlen mit Scheinen: Erst den größten Schein nehmen, der passt.

Schritt	Rechnung	Bit gesetzt bei
1	200 − 128 = 72	128
2	72 − 64 = 8	64
3	32 und 16 passen nicht in 8	–
4	8 − 8 = 0	8

Ergebnis: 128 + 64 + 8 = 200, also **`1100 1000`**.

Weitere Beispiele aus der Präsentation:

Dezimal	Binär	Rechnung
5	`0000 0101`	4 + 1
10	`0000 1010`	8 + 2
42	`0010 1010`	32 + 8 + 2
127	`0111 1111`	64 + 32 + 16 + 8 + 4 + 2 + 1
255	`1111 1111`	128 + 64 + 32 + 16 + 8 + 4 + 2 + 1

📌 **Merke:** Mit 8 Bit gibt es 2^8 = 256 Werte, von 0 bis 255.

3.2 🔤 Das Hexadezimalsystem

🎯 IHK-Relevanz: mittel bis hoch – Hex braucht man für MAC- und IPv6-Adressen; Umrechnungen sind schnelle Rechenpunkte.

Lange Binärzahlen sind schwer zu lesen. Deshalb gibt es eine Kurzschrift mit 16 Ziffern. Sie heißt **Hexadezimalsystem** (Basis 16). Nach der 9 folgen die Buchstaben A bis F.

Hex	A	B	C	D	E	F
Dezimal	10	11	12	13	14	15

- 🧱 Eine Hex-Ziffer entspricht genau 4 Bit.
- 4 Bit heißen **Nibble**.
- Ein Byte hat also genau zwei Hex-Ziffern.
- Die Vorsilbe `0x` kennzeichnet eine Hex-Zahl.

Beispiel 7: Hex nach dezimal

0xAF = A × 16 + F × 1 = 10 × 16 + 15 = 160 + 15 = **175**.

Beispiel 8: Dezimal nach hex

173 umrechnen:

1. 173 / 16 = 10 Rest 13.
2. 10 = A, 13 = D.
3. Ergebnis: **0xAD**.

✅ Probe: 10 × 16 + 13 = 173.

Beispiel 9: Binär nach hex über Nibbles

`1011 0101` in zwei Nibbles teilen (siehe Bild oben, Teil C):

- `1011` = 8 + 2 + 1 = 11 = B.
- `0101` = 4 + 1 = 5.
- Ergebnis: **0xB5** (dezimal 181, siehe Beispiel 5).

Beispiel 10: IP-Adresse in hex

192.168.1.1 = C0.A8.01.01 (hex).

- 192 = 12 × 16 + 0 = C0.
- 168 = 10 × 16 + 8 = A8.

3.3 💾 Speichereinheiten: dezimal oder binär?

🎯 IHK-Relevanz: hoch – Umrechnungen zwischen TB und TiB sowie die Frage “Warum zeigt Windows weniger an?” sind typische Rechen- und Erklärungsaufgaben.

- Ein **Bit** ist die kleinste Informationseinheit.
- Ein **Byte** besteht aus 8 Bit.

📏 Für größere Einheiten gibt es zwei Maßsysteme. Das ist wie Meilen und Kilometer: gleiche Strecke, andere Zahl.

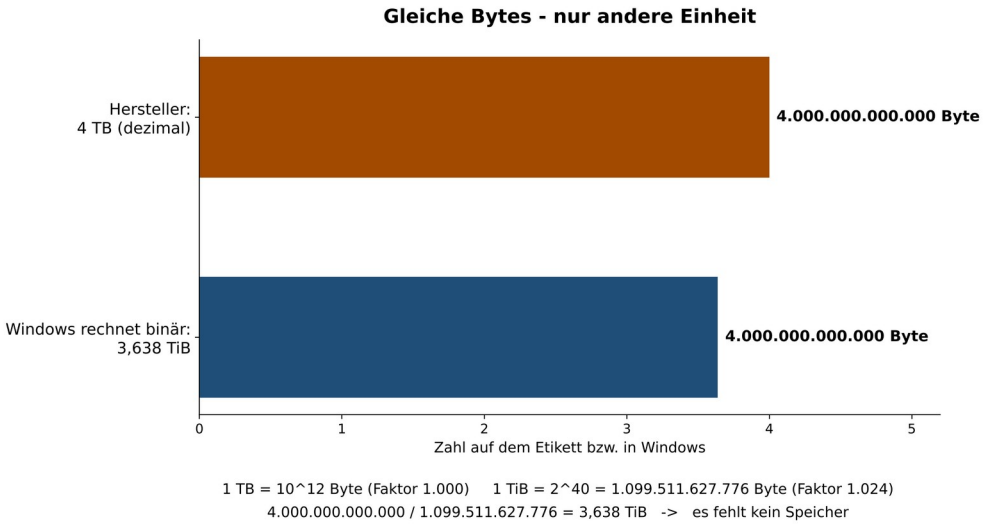
- **Dezimal (SI-Präfixe)**: Faktor 1.000. Einheiten kB, MB, GB, TB. Festplattenhersteller rechnen so.
- **Binär (IEC-Präfixe)**: Faktor 1.024 = 2^10. Einheiten KiB, MiB, GiB, TiB.

Dezimal	Größe	Binär	Größe
1 kB (Kilobyte)	10^3 = 1.000 Byte	1 KiB (Kibibyte)	2^10 = 1.024 Byte
1 MB (Megabyte)	10^6 = 1.000.000 Byte	1 MiB (Mebibyte)	2^20 = 1.048.576 Byte
1 GB (Gigabyte)	10^9 = 1.000.000.000 Byte	1 GiB (Gibibyte)	2^30 = 1.073.741.824 Byte
1 TB (Terabyte)	10^12 Byte	1 TiB (Tebibyte)	2^40 = 1.099.511.627.776 Byte

✏️ Kleines k bei kB, großes K bei KiB. So steht es in den Normen.

Beispiel 11: 4-TB-Festplatte in TiB

1. Herstellerangabe in Byte: 4 TB = 4 × 10^12 = 4.000.000.000.000 Byte.
2. 1 TiB = 2^40 = 1.099.511.627.776 Byte.
3. 4.000.000.000.000 / 1.099.511.627.776 = **3,638 TiB** (gerundet).



Dieselbe Bytezahl ergibt 4 TB dezimal oder 3,638 TiB binär

❔ Warum zeigt Windows weniger an?

- Windows rechnet binär.
- Windows schreibt aber “TB” statt “TiB”.
- Es zeigt also etwa “3,63 TB” an.
- ✅ Es fehlt kein Speicher. Nur die Einheit ist eine andere.
- Dazu kommt ein kleiner Anteil für Partitionstabelle und Dateisystem.

Beispiel 12: 1-TB-SSD in GiB

1 × 10^12 / 2^30 = 1.000.000.000.000 / 1.073.741.824 = **931,32 GiB**.

📈 Der Unterschied wächst mit der Größe:

Präfix	Faktor binär / dezimal	Abweichung
Kilo / Kibi	1,024	2,4 %
Giga / Gibi	1,0737	7,4 %
Tera / Tebi	1,0995	10,0 %

📝 Rechenweg-Tipp für die Prüfung: Immer zuerst in Byte umrechnen. Dann durch die Zieleinheit teilen.

❓ Übungsfragen Block 3

Ü3.1 Rechnen Sie um: a) 0x3C nach dezimal, b) 173 nach binär, c) `1111 1111` nach hex.

Ü3.2 Ein Kunde kauft eine 500-GB-SSD. Windows zeigt etwa 465 GB an. Berechnen Sie den Wert und erklären Sie den Unterschied.

Ü3.3 Wie viele verschiedene Werte lassen sich mit 12 Bit darstellen?

🧮 Block 4: Arbeitsspeicher (RAM)

4.1 📋 Der RAM in der Architektur

🎯 IHK-Relevanz: mittel bis hoch – RAM-Auswahl gehört zu "Hardware auswählen" (AP1); Kapitel 1 liefert dafür die Grundlage, Details folgen im RAM-Kapitel.

🗂️ Der RAM ist wie dein Schreibtisch. Dort liegt alles, woran du gerade arbeitest. Der Arbeitsspeicher heißt **RAM** (Random Access Memory). Er ist das **Speicherwerk** der Von-Neumann-Architektur. Er hält Programme und Daten, mit denen die CPU gerade arbeitet.

📌 Wichtige Merkpunkte aus Kapitel 1:

- ⚡ Der RAM ist **flüchtig**. Ohne Strom gehen die Daten verloren.
- 🐢 Der RAM ist viel langsamer als die CPU. Daraus folgt der Von-Neumann-Flaschenhals (Abschnitt 1.4).
- 📏 Die nutzbare RAM-Menge hängt von der Adressbusbreite ab. Ein 32-Bit-System adressiert höchstens 4 GiB (Abschnitt 2.2).

Begriff	Bedeutung
DDR5	aktuelle RAM-Generation (Double Data Rate 5)
Dual-Channel	zwei Speicherkanäle arbeiten parallel, die Bandbreite verdoppelt sich
ECC-RAM	Error-Correcting Code: erkennt und korrigiert Bitfehler; braucht ein ECC-fähiges Mainboard, typisch für Server
XMP / EXPO	gespeicherte Profile für höheren RAM-Takt (XMP bei Intel, EXPO bei AMD); im UEFI aktivieren
Cache L1/L2/L3	kleiner, sehr schneller Speicher in der CPU, puffert RAM-Zugriffe

🛣️ Dual-Channel im Alltag: zwei Fahrspuren statt einer. Es passen doppelt so viele Autos durch.

4.2 📊 Bandbreite berechnen

🎯 IHK-Relevanz: mittel – Bandbreitenrechnungen passen zu Auswahl- und Vergleichsaufgaben; die Formel ist Ergänzung, nicht aus Kapitel 1.

ℹ️ Ergänzung, nicht auf den Folien:

- Ein DDR-Speicherkanal ist 64 Bit = 8 Byte breit.
- Die Datenrate steht in **MT/s**. Das heißt Megatransfers pro Sekunde.

📌 **Formel:** Bandbreite = Transferrate × 8 Byte × Anzahl Kanäle

Beispiel 13: DDR5-6400 im Dual-Channel

1. Ein Kanal: 6.400.000.000 Transfers/s × 8 Byte = 51.200.000.000 Byte/s = 51,2 GB/s.
2. Zwei Kanäle: 51,2 GB/s × 2 = **102,4 GB/s**.

✅ Das passt zur Präsentation: DDR5 Dual-Channel ca. 100 GB/s.

🔭 Ausblick: Das RAM-Kapitel behandelt DRAM und SRAM, DDR3 bis DDR5, MT/s und MHz, SPD, Timings, SODIMM und MemTest86.

❓ Übungsfragen Block 4

Ü4.1 Ein Server soll ausfallsicher arbeiten. Welche RAM-Art empfehlen Sie, und welche Voraussetzung muss das Mainboard erfüllen?

Ü4.2 Berechnen Sie die Bandbreite von DDR5-4800 im Dual-Channel-Betrieb in GB/s.

🔐 Block 5: Firmware – BIOS, UEFI, MBR/GPT, Secure Boot, TPM

5.1 🧓 BIOS

🎯 IHK-Relevanz: mittel – BIOS wird vor allem als Gegenstück zu UEFI abgefragt.

Software, die fest auf einem Chip im Gerät sitzt, heißt **Firmware**. Sie läuft beim Einschalten als Erstes. 🔑 Sie ist wie der Hausmeister, der morgens aufschließt und das Licht anmacht.

Die alte PC-Firmware heißt **BIOS** (Basic Input/Output System). Sie stammt aus den 1980er-Jahren. Sie liegt in einem Flash-Baustein auf dem Mainboard.

Aufgaben des BIOS:

1. 🩺 **POST** ausführen (Selbsttest, siehe Block 6).
2. ⚙️ Hardware initialisieren.
3. 🔎 Bootloader suchen. Er steht in den ersten 512 Byte der Bootplatte. Dieser Bereich heißt **MBR**.

🔋 Einstellungen wie Boot-Reihenfolge und Uhrzeit liegen im **CMOS**-Speicher. Eine Knopfzelle versorgt ihn mit Strom (**CR2032**). Ist sie leer, gehen Uhrzeit und Einstellungen verloren.

⌨️ Aufruf beim Start: Entf, F2, F10 oder Esc. Das hängt vom Hersteller ab.

Grenze des BIOS	Folge
16-Bit Real Mode	nur 1 MiB direkt adressierbar, langsame Initialisierung

Grenze des BIOS	Folge
nur MBR	max. 2 TiB (ca. 2,2 TB) und max. 4 primäre Partitionen
kaum Treiber	nur Grundfunktionen
kein Secure Boot	Schadsoftware (Bootkits) kann sich vor dem Betriebssystem einnisten

5.2 🆕 UEFI und der Vergleich mit BIOS

🎯 IHK-Relevanz: hoch – der Vergleich BIOS/UEFI ist ein Standardthema bei Arbeitsplatzeinrichtung in AP1.

Der Nachfolger des BIOS heißt **UEFI** (Unified Extensible Firmware Interface).

- Es läuft im 32- oder 64-Bit-Modus.
- 🖱️ Es hat eine grafische Oberfläche mit Maus.
- Es kann Treiber und Netzwerk nutzen.

📁 UEFI startet den Bootloader aus einer eigenen Partition. Sie heißt **EFI System Partition (ESP)**. Die ESP ist mit FAT32 formatiert. Sie enthält `.efi`-Dateien.

ℹ️ Die Quellen nennen unterschiedliche Jahre. Skript: UEFI "seit etwa 2005". Präsentation: "seit ca. 2011 Standard auf neuen Systemen". Beides passt: Die Spezifikation ist älter. Standard auf PCs wurde UEFI erst später.

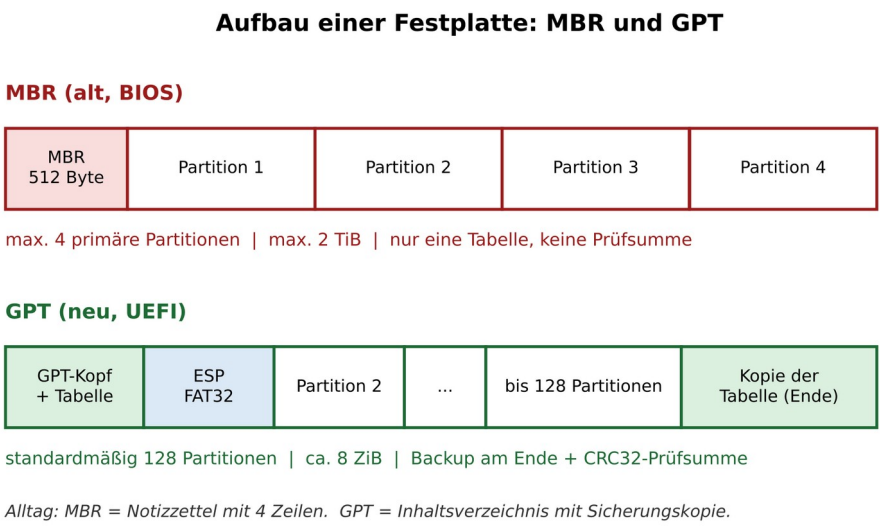
Merkmal	BIOS (Legacy)	UEFI
Modus	16-Bit Real Mode	32-Bit oder 64-Bit
Startzeit	langsam	schnell (Fast Boot)
Partitionsschema	MBR: 4 primäre Partitionen, 2 TiB	GPT: standardmäßig 128 Partitionen, ca. 8 ZiB
Bedienung	Text, nur Tastatur	grafisch, mit Maus
Sicherheit	keine Signaturprüfung	**Secure Boot**, Measured Boot
Treiber	Grundfunktionen	modulare UEFI-Treiber
Einsatz heute	nur Altsysteme	Standard

🔙 UEFI hat einen Legacy-Modus für alte BIOS-Systeme. Er heißt **CSM** (Compatibility Support Module). ⚠️ Für Secure Boot muss CSM aus sein.

5.3 🗄️ MBR und GPT

🎯 IHK-Relevanz: hoch – Grenzen von MBR, Vorteile von GPT und die 2-TiB-Rechnung passen zu AP1 und zu "Speicherlösungen integrieren" in AP2 Konzeption/Administration.

📖 Eine Festplatte braucht ein Inhaltsverzeichnis. Es sagt, wo welche Partition beginnt. Dieses Verzeichnis heißt Partitionstabelle. Es gibt zwei Arten: MBR (alt) und GPT (neu).



Aufbau einer Festplatte mit MBR und mit GPT im Vergleich

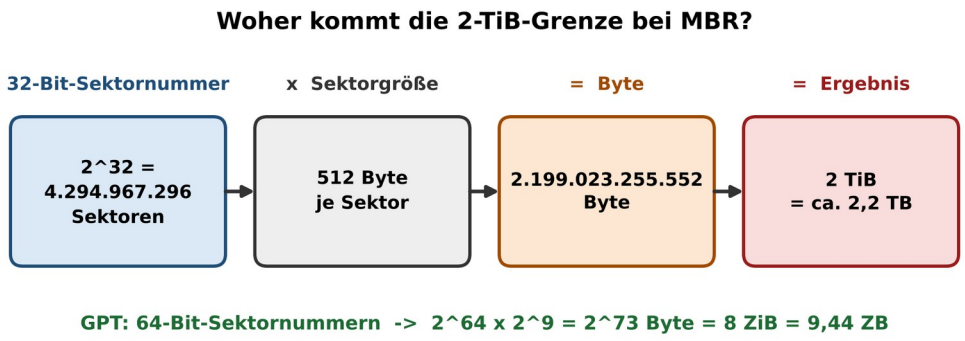
Merkmal	**MBR** (Master Boot Record)	**GPT** (GUID Partition Table)
eingeführt	1983 (DOS)	um 2000, Teil der UEFI-Spezifikation
Partitionen	4 primäre, oder 3 primäre + 1 erweiterte mit logischen Laufwerken	standardmäßig 128
max. Größe (512-Byte-Sektoren)	2 TiB ≈ 2,2 TB	8 ZiB ≈ 9,4 ZB
Redundanz	keine, nur eine Tabelle	Kopie der Tabelle am Plattenende
Prüfsumme	keine	CRC32
Bootloader	im MBR (erste 512 Byte)	in der ESP (ca. 100 bis 500 MB)
Startmodus	BIOS bzw. CSM	UEFI
Empfehlung	nur Altsysteme	Standard für neue Systeme

ℹ️ Hinweis zu den Quellen: Die Präsentation nennt für GPT “2005 (UEFI 2.0)”. GPT entstand schon um 2000 mit EFI. Seit UEFI 2.0 ist GPT Teil der Spezifikation. 📌 Für die Prüfung zählt: GPT gehört zu UEFI.

⚠️ Windows startet von einer GPT-Platte nur im UEFI-Modus. Als reine Datenplatte geht GPT auch an BIOS-PCs mit 64-Bit-Windows.

Beispiel 14: Woher kommt die 2-TiB-Grenze bei MBR?

Der MBR speichert Sektornummern mit 32 Bit. Ein Sektor hat 512 Byte.



Rechenweg zur 2-TiB-Grenze bei MBR und zur 8-ZiB-Grenze bei GPT

1. Anzahl Sektoren: 2^32 = 4.294.967.296.
2. Größe: 4.294.967.296 × 512 Byte = 2.199.023.255.552 Byte.
3. In TiB: 2.199.023.255.552 / 2^40 = **2 TiB**.
4. In TB: 2.199.023.255.552 / 10^12 = **2,2 TB** (gerundet).

Beispiel 15: Grenze bei GPT

GPT nutzt 64-Bit-Sektornummern.

1. 2^64 × 512 Byte = 2^64 × 2^9 = 2^73 Byte.
2. 2^73 / 2^70 = **8 ZiB** (Zebibyte).
3. Dezimal: 2^73 / 10^21 = **9,44 ZB**.

✅ Das erklärt die zwei Angaben in den Quellen: 8 ZiB und 9,4 ZB sind derselbe Wert.

5.4 🛡️ Secure Boot

🎯 IHK-Relevanz: hoch – Secure Boot verbindet Hardware mit IT-Sicherheit; IT-Sicherheit nennt die Verordnung für AP1 ausdrücklich.

🛂 Vergleich mit einer Ausweiskontrolle: Nur wer einen gültigen, echten Ausweis hat, darf rein.

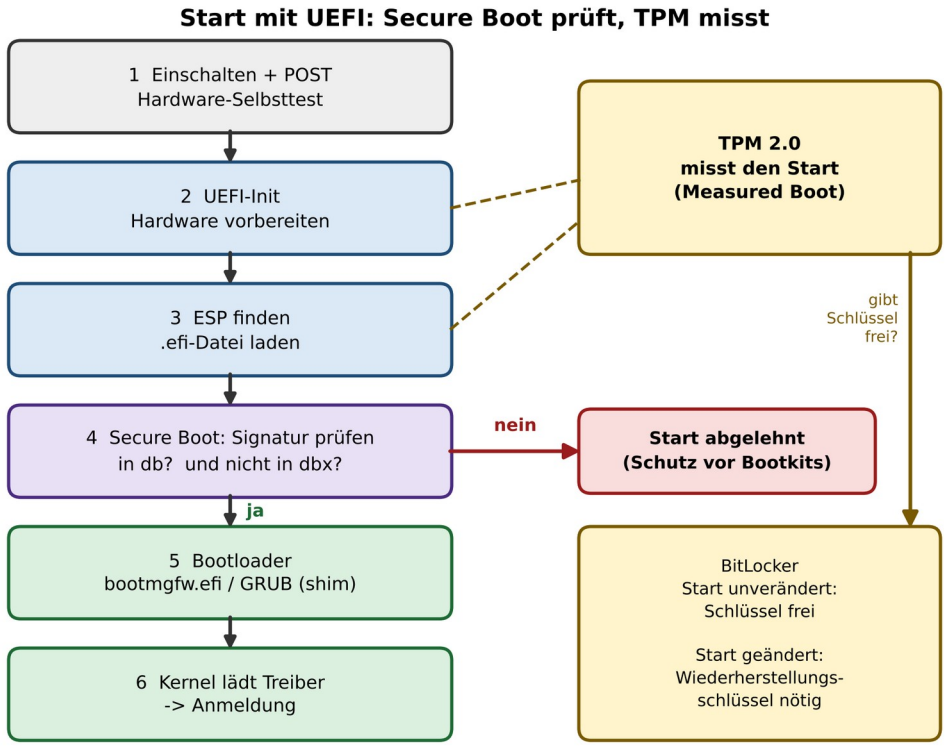
Secure Boot ist eine Funktion von UEFI.

- Es startet nur Bootloader und Kernel mit gültiger digitaler Signatur.
- 🚫 Unbekannte oder manipulierte Software wird abgelehnt.
- Das schützt vor Schadsoftware im Startvorgang: **Bootkits** und **Rootkits**.

Baustein	Aufgabe
PK (Platform Key)	oberster Vertrauensanker, meist vom Mainboard-Hersteller
KEK (Key Exchange Key)	Schlüssel von Microsoft und Hersteller; dürfen db und dbx ändern
db (Signature Database)	Liste erlaubter Signaturen, z. B. für

Baustein	Aufgabe
	`shim.efi` unter Linux
dbx (Forbidden Signature Database)	Sperrliste widerrufener Signaturen
Custom Mode	eigene Schlüssel eintragen, z. B. für eine Organisation

🔎 Ablauf: Die Firmware prüft die Signatur des Bootloaders. Steht sie in db und nicht in dbx? Dann startet der Bootloader.



UEFI-Startablauf mit Secure-Boot-Prüfung, TPM-Messung und BitLocker

🐧 Linux-Distributionen wie Ubuntu oder RHEL nutzen einen signierten Vor-Bootloader. Er heißt **shim**.

5.5 🔒 TPM und BitLocker

🎯 IHK-Relevanz: mittel bis hoch – TPM und Festplattenverschlüsselung gehören zu IT-Sicherheit am Arbeitsplatz (AP1).

🗝️ Stell dir einen kleinen Tresor auf dem Mainboard vor. Dieser Sicherheitschip heißt **TPM** (Trusted Platform Module).

- Er speichert kryptografische Schlüssel geschützt.
- Er misst den Startvorgang (Measured Boot).
- Er kann auch in die CPU-Firmware eingebaut sein: **Intel PTT** bzw. **AMD fTPM**.

🤝 Zusammenspiel:

- **Secure Boot** sorgt für einen vertrauenswürdigen Start.
- Das **TPM** gibt den Schlüssel für **BitLocker** nur frei, wenn der Start unverändert war.
- Ergebnis: Das System entsperrt sich automatisch. Ein geänderter Start fällt auf.

🪟 Windows 11 setzt UEFI mit Secure-Boot-Fähigkeit und TPM 2.0 voraus.

⚠️ Wichtig für die Praxis:

- Nach Hardwaretausch oder Firmware-Update kann das TPM den Start als verändert ansehen.
- Dann fragt BitLocker nach dem **Wiederherstellungsschlüssel**.
- Diesen Schlüssel holt man vorher aus AD oder Intune.

🖥️ Prüfen unter Windows (PowerShell, als Administrator):

```
Confirm-SecureBootUEFI    # True = Secure Boot aktiv
Get-Tpm                   # zeigt TpmPresent, TpmReady, TpmEnabled
```

5.6 ⚙️ Typische Einstellungen im UEFI

🎯 IHK-Relevanz: mittel – passt zu "Arbeitsplatz konfigurieren und testen" (AP1); Menüpfade sind herstellerabhängig.

Aufgabe	Typischer Menüpfad	Hinweis
UEFI öffnen	Entf, F2, F10 oder Esc beim Start	je nach Hersteller
Boot-Reihenfolge	Boot → Boot Priority	PXE-Boot für Neuinstallation über das Netz
Secure Boot	Security → Secure Boot → Enabled	in Behördennetzen meist Pflicht
TPM	Security → TPM → Enabled	Voraussetzung für BitLocker
XMP/EXPO	Advanced → Memory → XMP Profile 1	höheren RAM-Takt aktivieren
Virtualisierung	Advanced → CPU → Intel VT-x oder AMD-V	nötig für VMs (Hyper-V, VirtualBox)

❓ Übungsfragen Block 5

Ü5.1 Nennen Sie drei Unterschiede zwischen BIOS und UEFI.

Ü5.2 Ein Kunde möchte eine 4-TB-Festplatte als Systemlaufwerk mit MBR einrichten. Beurteilen Sie das Vorhaben mit Rechnung.

Ü5.3 Ein neuer Client soll BitLocker mit automatischer Entsperrung nutzen. Welche drei Einstellungen prüfen Sie im UEFI?

🚦 Block 6: POST und Bootvorgang

6.1 🩺 Der POST

🎯 IHK-Relevanz: mittel – Fehlerdiagnose beim Start passt zu “Arbeitsplatz testen” in AP1; konkrete Pieptöne sind herstellerabhängig und eher Praxiswissen.

✈️ Wie der Pilot vor dem Start: Er geht eine Checkliste durch. Erst wenn alles passt, hebt er ab.

Beim PC heißt diese Checkliste **POST** (Power-On Self-Test).

- Es ist ein automatischer Selbsttest beim Einschalten.
- Die Firmware führt ihn aus, vor dem Start des Betriebssystems.
- Erst nach einem fehlerfreien POST startet der Bootloader.

Ablauf (Skript):

1. ⚡ Die Stromversorgung stabilisiert sich. Ein Reset-Signal startet die CPU.
2. Die CPU lädt die ersten Befehle aus dem Firmware-Flash.
3. Der Speichercontroller initialisiert und prüft den RAM.
4. 🖥️ Die Grafik wird initialisiert. Die erste Ausgabe erscheint.
5. ⌨️ Tastatur, USB und Onboard-Geräte werden geprüft.
6. 💽 Massenspeicher und Boot-Reihenfolge werden eingelesen.
7. 🚀 Der Bootloader auf dem ersten bootfähigen Medium startet.

📌 Kurzform: CPU → RAM → Grafik → Tastatur → Boot.

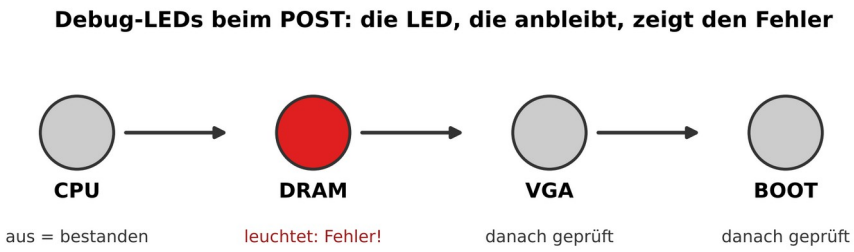
🚨 Fehler meldet der POST über:

- **Pieptöne**,
- **Debug-LEDs**,
- **Q-Codes** (zweistellige Anzeige auf dem Mainboard).

Signal	Bedeutung (Beispiel AMI/Award)
1 kurz	POST erfolgreich
1 lang + 2 kurz	Grafikkartenfehler
3 kurz	RAM-Fehler (AMI)
kein Ton, kein Bild	CPU nicht erkannt oder Mainboard defekt
Dauerton	Tastaturfehler oder schwerer Fehler

⚠️ Die Codes hängen vom Hersteller ab. Die verlässliche Quelle ist immer das Handbuch des Mainboards.

💡 **Debug-LEDs** leuchten der Reihe nach: CPU → DRAM → VGA → BOOT. Nach jedem bestandenen Test erlischt die LED. Bleibt eine LED an, liegt dort der Fehler.



Beispiel 16: DRAM bleibt an -> RAM-Module prüfen

Debug-LEDs beim POST: die DRAM-LED bleibt an und zeigt einen RAM-Fehler

Beispiel 16: DRAM-LED leuchtet dauerhaft

1. 🔧 RAM-Module fest einsetzen. Beide Halteclips müssen einrasten.
2. Module einzeln testen, auch in anderen Slots.
3. XMP/EXPO im UEFI deaktivieren.

6.2 🚀 Der Bootvorgang mit UEFI

🎯 IHK-Relevanz: mittel – die Reihenfolge vom Einschalten bis zur Anmeldung ist gutes Grundlagenwissen für Fehlersuche.

Den Ablauf zeigt auch das Bild in Abschnitt 5.4.

Phase	Ablauf
UEFI-Init	Firmware initialisiert die Hardware und prüft Secure Boot
ESP-Suche	Firmware findet die ESP und lädt die `.efi`-Datei
Bootloader	Windows Boot Manager (`bootmgfw.efi`) oder GRUB lädt den Kernel
Kernel	Kernel lädt Treiber und startet Dienste
Anmeldung	Anmeldebildschirm, danach Programme wie Explorer oder Shell

🧰 Diagnose-Werkzeuge:

- Mainboard-Handbuch,
- POST-Karte,
- Multimeter am 24-Pin-ATX-Stecker,
- Knopfzelle entfernen (CMOS-Reset),
- **Minimalkonfiguration**: nur CPU, ein RAM-Modul, Grafik, Netzteil,
- Komponenten einzeln tauschen.

❓ Übungsfragen Block 6

Ü6.1 Was ist der POST, und wann findet er statt? Nennen Sie drei geprüfte Komponenten.

Ü6.2 Ein PC zeigt kein Bild und gibt einen langen und zwei kurze Pieptöne aus (AMI-BIOS). Nennen Sie die wahrscheinliche Ursache und zwei Prüfschritte.

🪖 Block 7: Bundeswehr-Kontext

7.1 🏕️ Anforderungen an Hardware in der Bundeswehr

🎯 IHK-Relevanz: gering – die Prüfung fragt keinen BW-Kontext ab; die Prinzipien (Robustheit, Lebenszyklus, Sicherheit) helfen aber bei Auswahlbegründungen und im Projekt.

Anforderung	Umsetzung
Robustheit	stoß-, vibrations-, staub- und feuchtigkeitsfest
Temperatur	Betrieb im Feld, laut Skript von −32 °C bis +60 °C
Manipulationsschutz	versiegelte Gehäuse, TPM, Secure Boot, BSI-Zertifizierung
Lebenszyklus	lange Ersatzteilversorgung (Skript: 10+ Jahre, Präsentation: 5 bis 10 Jahre)
Lieferkette	zugelassene Hersteller, sichere Herkunft

📖 Begriffe:

- 🧱 **Ruggedized**: verstärkte Geräte gegen Stoß, Staub, Wasser und Temperatur. Oft lüfterlos und mit versiegelten Anschlüssen.
- 🇺🇸 **MIL-STD-810**: US-Militärstandard für Umweltprüfungen. Die Angabe “tested” ist meist eine Selbstauskunft des Herstellers.
- 💧 **IP-Schutzklasse**: Schutz gegen Staub und Wasser. Ergänzt MIL-STD-810.
- 🌬️ **Fanless**: passiv gekühlt, ohne Lüfter. Die CPU hat eine niedrige Verlustleistung (**TDP**, z. B. ca. 15 W).

🔒 Auf BWI-Clients ist Secure Boot Pflicht. Ungeprüfte Live-Linux-Sticks starten deshalb nicht ohne Weiteres.

❓ Übungsfragen Block 7

Ü7.1 Ein Notebook soll im Fahrzeug im Feld eingesetzt werden. Nennen Sie drei Anforderungen und je eine technische Umsetzung.

✅ Lösungen

Block 1

Ü1.1 Steuerwerk (Control Unit in der CPU), Rechenwerk (ALU in der CPU), Speicherwerk (RAM), Eingabewerk (z. B. Tastatur, Maus), Ausgabewerk (z. B. Monitor, Drucker). Alle sind über den Systembus verbunden.

Ü1.2

1. Fetch: Die CPU holt den nächsten Befehl aus dem RAM in das Befehlsregister.
2. Decode: Das Steuerwerk dekodiert den Befehl und ermittelt die Operanden.
3. Execute: Die ALU führt den Befehl aus. Das Ergebnis geht in ein Register oder den Speicher.

Danach zeigt der Befehlszähler auf den nächsten Befehl.

Ü1.3 Programm und Daten teilen sich einen Bus. Befehle und Daten gehen nur nacheinander. Der RAM ist zudem viel langsamer als die CPU. Die CPU wartet. Gegenmaßnahmen (zwei davon): Cache L1/L2/L3, Prefetching, schnellere Speicherbusse (DDR5, Dual-Channel), getrennte L1-Caches für Befehle und Daten.

Block 2

Ü2.1

- Adressbus: unidirektional, von der CPU weg. Er wählt die Speicherstelle oder das Gerät aus.
- Datenbus: bidirektional. Er überträgt die Daten.
- Steuerbus: bidirektional. Er überträgt Steuersignale wie Lesen/Schreiben, Takt, Interrupts und Reset.

Ü2.2 2^20 = 1.048.576 Byte. 1.048.576 / 1.048.576 = **1 MiB**.

Ü2.3 32-Bit-Windows nutzt 32-Bit-Adressen. 2^32 = 4.294.967.296 Byte = 4 GiB. Mehr Speicher kann es nicht adressieren. Ein Teil dieser Adressen ist für Geräte reserviert, z. B. die Grafikkarte. Deshalb bleiben etwa 3,5 GiB. Lösung: ein 64-Bit-Betriebssystem installieren.

Block 3

Ü3.1

- a) 0x3C = 3 × 16 + 12 = 48 + 12 = **60**.
- b) 173: 173 − 128 = 45; 45 − 32 = 13; 13 − 8 = 5; 5 − 4 = 1; 1 − 1 = 0. Gesetzt sind 128, 32, 8, 4, 1. Ergebnis: **`1010 1101`**. Probe: 128 + 32 + 8 + 4 + 1 = 173.
- c) `1111` = 15 = F. Zwei Nibbles: **0xFF** (dezimal 255).

Ü3.2

1. 500 GB = 500 × 10^9 = 500.000.000.000 Byte.
2. 1 GiB = 2^30 = 1.073.741.824 Byte.
3. 500.000.000.000 / 1.073.741.824 = **465,66 GiB**.

Der Hersteller rechnet dezimal, Windows binär. Windows schreibt dabei "GB", meint aber GiB. Es fehlt kein Speicher.

Ü3.3 2^12 = **4.096** Werte, von 0 bis 4.095.

Block 4

Ü4.1 ECC-RAM. Er erkennt und korrigiert Bitfehler. Mainboard und CPU müssen ECC unterstützen. Das ist bei Server-Plattformen üblich.

Ü4.2

1. Ein Kanal: 4.800.000.000 Transfers/s × 8 Byte = 38.400.000.000 Byte/s = 38,4 GB/s.
2. Dual-Channel: 38,4 GB/s × 2 = **76,8 GB/s**.

Block 5

Ü5.1 Drei von diesen:

- BIOS arbeitet im 16-Bit-Modus. UEFI arbeitet im 32- oder 64-Bit-Modus.
- BIOS nutzt MBR (max. 2 TiB, 4 primäre Partitionen). UEFI nutzt GPT (128 Partitionen, ca. 8 ZiB).
- UEFI bietet Secure Boot, BIOS nicht.
- UEFI hat eine grafische Oberfläche mit Maus.
- UEFI startet schneller und kann modulare Treiber laden.

Ü5.2 Das ist nicht sinnvoll. MBR nutzt 32-Bit-Sektornummern mit 512-Byte-Sektoren. 2^32 × 512 Byte = 2.199.023.255.552 Byte ≈ 2,2 TB. Von 4 TB wären nur etwa 2,2 TB nutzbar. Empfehlung: GPT im UEFI-Modus. Damit ist die ganze Platte nutzbar. Außerdem ist Secure Boot möglich.

Ü5.3

- UEFI-Modus statt Legacy (CSM deaktiviert).
- Secure Boot aktiviert.
- TPM 2.0 aktiviert (bzw. Intel PTT oder AMD fTPM).

Prüfen unter Windows mit `Confirm-SecureBootUEFI` und `Get-Tpm`.

Block 6

Ü6.1 Der POST ist der Power-On Self-Test. Die Firmware führt ihn direkt nach dem Einschalten aus. Das ist vor dem Start des Betriebssystems. Geprüft werden z. B. CPU, RAM, Grafik, Tastatur, CMOS und Bootgerät.

Ü6.2 Wahrscheinliche Ursache: Grafikkartenfehler. Prüfschritte (zwei davon):

- Sitz der Karte im PCIe-Slot prüfen.
- PCIe-Stromkabel prüfen.
- Anderen Slot oder andere Karte testen.
- Handbuch mit der Beep-Code-Tabelle prüfen.

Block 7

Ü7.1 Drei von diesen:

- Robustheit: Ruggedized-Gerät nach MIL-STD-810, verstärktes Gehäuse.
- Staub- und Wasserschutz: passende IP-Schutzklasse.
- Temperaturbereich: lüfterlos, passive Kühlung, CPU mit niedriger TDP.
- Manipulationsschutz: TPM 2.0 und Secure Boot.
- Lange Lieferbarkeit: Industrie-Mainboards, langer Lebenszyklus.

🔑 Schlagwortliste für die Prüfung

Begriff	Kurzdefinition	Block
Computer	Maschine, die ein gespeichertes Programm automatisch abarbeitet	1
Von-Neumann-Architektur	ein gemeinsamer Speicher und Bus für Programm und Daten	1
Steuerwerk (CU)	holt und dekodiert Befehle, steuert den Ablauf	1
Rechenwerk (ALU)	arithmetische und logische Operationen	1
Speicherwerk	speichert Programm und Daten, heute RAM	1
Ein-/Ausgabewerk	Datenaustausch mit der Außenwelt	1
Register / Status-Flags	Speicherplätze in der CPU / Zero, Carry, Overflow	1
Befehlszähler (PC)	zeigt auf die Adresse des nächsten Befehls	1
Interrupt	Unterbrechungsanforderung an die CPU	1
Fetch – Decode – Execute	Befehlszyklus: holen, dekodieren, ausführen	1
Stored-Program Concept	Programm liegt als Daten im Speicher und ist	1

Begriff	Kurzdefinition	Block
	austauschbar	
Von-Neumann-Flaschenhals	gemeinsamer Bus und langsamer RAM bremsen die CPU	1
Cache L1/L2/L3	schneller Zwischenspeicher zwischen CPU und RAM	1
Harvard-Architektur	getrennte Speicher und Busse für Programm und Daten	1
Systembus	Adressbus, Datenbus, Steuerbus	2
Adressbus	unidirektional, wählt Adresse aus (Wo?)	2
Datenbus	bidirektional, überträgt Daten (Was?)	2
Steuerbus	Steuersignale wie Lesen/Schreiben, Takt, Interrupt (Wie?)	2
Adressraum 2^n	Anzahl adressierbarer Speicherzellen bei n Adressbits	2
Bit / Byte	kleinste Einheit / 8 Bit	3
Binärsystem	Basis 2, Ziffern 0 und 1	3
Hexadezimalsystem	Basis 16, Ziffern 0–9 und A–F	3
Nibble	4 Bit, eine Hex-Ziffer	3
kB, MB, GB, TB	dezimale Einheiten, Faktor 1.000	3
KiB, MiB, GiB, TiB	binäre Einheiten, Faktor 1.024	3
RAM	flüchtiger Arbeitsspeicher, Speicherwerk	4
Dual-Channel	zwei Speicherkanäle parallel, doppelte Bandbreite	4
ECC-RAM	RAM mit Fehlerkorrektur, typisch für Server	4
XMP / EXPO	RAM-Profile für höheren Takt (Intel / AMD)	4
MT/s	Megatransfers pro Sekunde, Datenrate des RAM	4
Firmware	Software fest auf dem Mainboard, startet zuerst	5

Begriff	Kurzdefinition	Block
BIOS	alte Firmware, 16 Bit, MBR	5
UEFI	moderne Firmware, 32/64 Bit, GPT, Secure Boot	5
CMOS / CR2032	Speicher für Firmware-Einstellungen / Knopfzelle	5
CSM	Legacy-Modus im UEFI	5
MBR	altes Partitionsschema, max. 2 TiB, 4 primäre Partitionen	5
GPT	modernes Partitionsschema, 128 Partitionen, Backup, CRC32	5
ESP	EFI System Partition, FAT32, enthält `.efi`-Bootloader	5
Secure Boot	nur signierte Bootloader und Kernel starten	5
PK, KEK, db, dbx	Schlüssel und Signaturlisten von Secure Boot	5
shim	signierter Vor-Bootloader für Linux	5
TPM 2.0	Sicherheitschip für Schlüssel, misst den Start	5
BitLocker	Festplattenverschlüsselung unter Windows, nutzt TPM	5
POST	Power-On Self-Test beim Einschalten	6
Beep-Code / Q-Code / Debug-LED	Fehlersignale des POST	6
Minimalkonfiguration	nur CPU, ein RAM, Grafik, Netzteil zur Fehlersuche	6
Ruggedized	robuste Hardware für raue Umgebungen	7
MIL-STD-810	US-Militärstandard für Umweltprüfungen	7
TDP	thermische Verlustleistung einer CPU	7