



Einheit H3 – Funktionsweise eines Computers

H-AB3.1

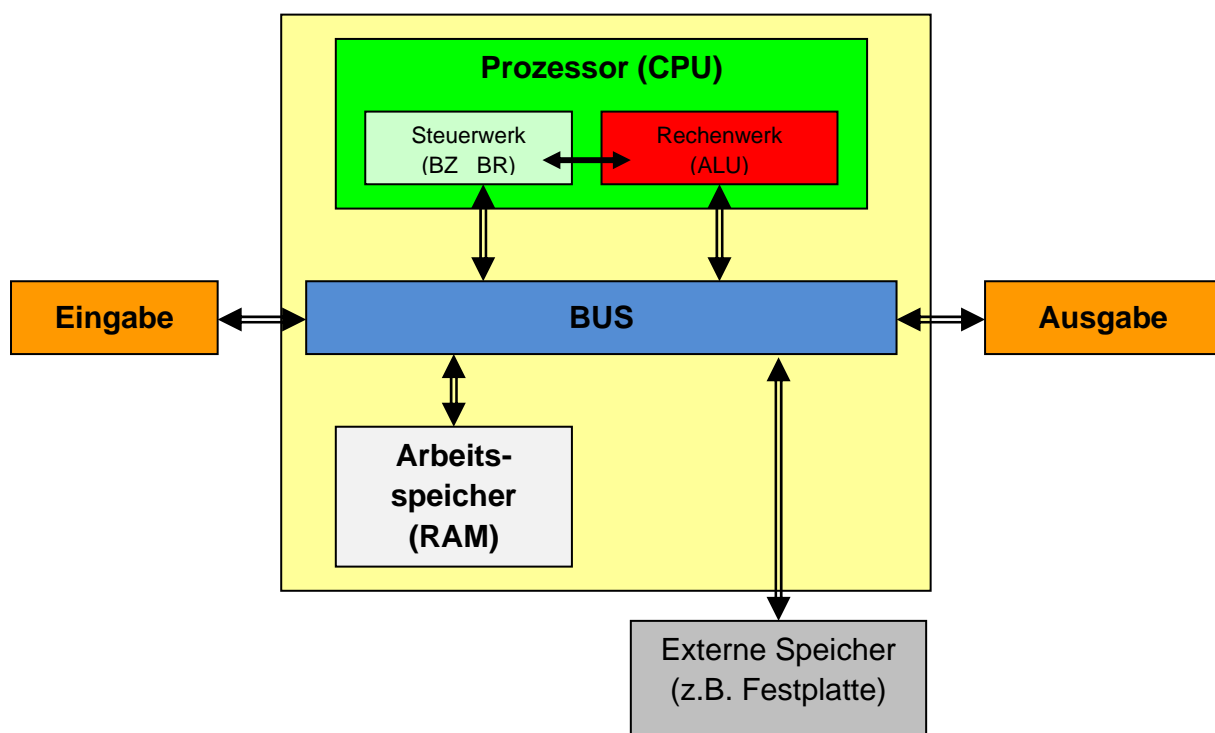
Auf diesem Arbeitsblatt zeigen wir Euch den Aufbau und die Funktionsweise eines Computers.

Die wesentlichste Eigenschaft eines Computers ist, dass er durch Programmierung von einer Universalmaschine zur Lösung beliebiger Berechnungs- oder Informationsverarbeitungsaufgaben zu einer Spezialmaschine gemacht werden kann. Die dafür erforderlichen Prinzipien hat der ungarische Mathematiker John von Neumann in den Vierzigerjahren des 20. Jahrhunderts vorgeschlagen.

Aus diesen Prinzipien ergibt sich eine Architektur, die aus

- Eingabeeinheit,
- Ausgabeeinheit,
- Speicher,
- Steuereinheit und
- Rechenwerk, sowie
- Verbindungsstrukturen zwischen diesen Komponenten (heute üblich, ein Bus)

besteht. Im Speicher befinden sich sowohl die Daten, die der Berechnung zu Grunde liegen als auch das Programm (die entsprechend dargestellte Berechnungsformel), das auf diese Daten angewandt werden soll. Die folgende Abbildung zeigt schematisch die Beziehungen zwischen den Komponenten eines von Neumann-Rechners .



So wie uns Menschen nur verstehen, wenn wir ihre Sprache sprechen, versteht uns auch der Computer letztlich nur, wenn wir seine Maschinensprache sprechen. Wir wollen hier ein Beispiel für eine sehr einfache Maschinensprache, den [Befehlsvorrat](#) unseres Computers angeben.



LIES Speicheradresse	---	Liest einen elementaren Wert von Eingabe in angegebene Speicheradresse.
SCHREIB Speicheradresse	---	Schreibt den in der angegebenen Speicheradresse enthaltenen Wert auf der Ausgabeeinheit
LADE Speicheradresse, Reg	---	Holt den in der angegebenen Speicheradresse gespeicherten Wert in das angegebene CPU-Register Register.
SPEICHERE Reg, Speicheradresse	---	Speichert den von der ALU berechneten im angegebenen Register abgelegten Wert in die angegebene Speicheradresse.
ADD R1, R2, R3	---	Addiert die in Register1 und Register 2 gespeicherten Werte. Ergebnis wird nach Register 3 geschrieben.
MULT R1, R2, R3	---	Multipliziert die in Register1 und Register 2 gespeicherten Werte. Ergebnis wird nach Register 3 geschrieben.
MULT* Reg, Wert	---	Multipliziert den Inhalt des angegebenen Registers mit einem konstanten Wert.
STOP	---	Beendet das Programm. Damit kommt der Rechner tatsächlich allerdings nicht zum Stillstand sondern die Kontrolle wird an das Betriebssystem übertragen.



Aus diesen Befehlen können wir nun ein Programm in Maschinensprache (Assembler) entwickeln.

Beginnen wir mit der Formel $O = 2 * (l * b + (l+b) * h)$

Dazu ist erst wichtig, festzustellen, dass wir in der Formel 4 Variable haben (O, l, b, h), für die wir Speicherplatz reservieren müssen. Wir nehmen dabei an, dass wir für jede dieser Variablen, wie auch für jeden dann zu schreibenden Befehl ein Speicherwort benötigen. Also reservieren wir

```
S1      reserviere Platz für O in Zelle z1
S2      reserviere Platz für l in Zelle z2
S3      reserviere Platz für b in Zelle z3
S4      reserviere Platz für h Zelle z4
```

Sodann stellen wir fest, dass wir wohl erst die Werte für l, b und h von der Eingabeeinheit einlesen müssen.

```
S5      LIES z2      lies l von Eingabeeinheit
S6      LIES z3      lies b von Eingabeeinheit
S7      LIES z4      lies h von Eingabeeinheit
```

An dieser Stelle können wir festhalten, dass unser Programm nicht sehr benutzerfreundlich ist, da wir uns merken müssen, in welcher Reihenfolge l, b und h eingegeben werden müssen. Das soll uns momentan aber noch nicht sehr stören. Da wir nun die Daten dort haben, wo sie sein sollten, können wir nun mit der eigentlichen Berechnung beginnen.

```
S8      LADE z2, r1   lade Wert von l in Register 1
S9      LADE z3, r2   lade Wert von b in Register 2
S10     MULT r1, r2, r3 führe auf die in Register 1 und Register 2 gespeicherten Werte eine
                          Multiplikation aus und schreibe das Ergebnis nach Register 3
```

Nun hängt es davon ab, wie viel Register unsere CPU hat ob wir weitere Werte in anderen noch freien Registern abspeichern können oder ob wir Zwischenergebnisse in den Speicher zurückschreiben und bei Bedarf frisch holen müssen. Wir nehmen an, wir hätten nur 3 Register, also müssen wir das Ergebnis der Multiplikation zurückschreiben. Dafür können wir die bisher noch unbenutzte Speicheradresse $z1$ verwenden.

```
S11     SPEICHERE r3, z1 speichere das Zwischenergebnis l*b einstweilen in die Speicherzelle z1
```

Als nächstes kommt wohl die Addition $(l+b)$ an die Reihe. Doch da wir die Werte von l und b bereits in Registern haben, müssen wir sie nicht holen, sondern können sofort addieren.

```
S12     ADD r1, r2, r3 führe auf den noch immer in Register 1 und Register 2 gespeicherten
                          Werten eine Addition aus und schreibe das Ergebnis nach Register 3
S13     LADE z4, r2   lade den Wert von h in das nun nicht mehr benötigte Register 2
S14     MULT r3, r2, r1 führe auf die in Register 3 (Ergebnis des Klammerausdrucks) und Register
                          2 (Wert von h) gespeicherten Werte eine Multiplikation aus und
                          schreib das Ergebnis in das inzwischen nicht mehr benötigte Register 1
```

Wir haben somit in Register 1 den Flächeninhalt der Vorderseite und eines Seitenteils, also des halben Mantels stehen. Wir müssen nun noch den Flächeninhalt der Grundfläche, die wir zwischenzeitlich in Zelle $z1$ abgespeichert haben holen, addieren und die so erhaltene Fläche mit dem konstanten Wert 2 multiplizieren.

```
S15     LADE z1, r2   lädt das in z1 abgespeicherte Zwischenergebnis in das inzwischen nicht
                          mehr benötigte Register 1
S16     ADD r1, r2, r3 addiert den in Register 1 gespeicherten Wert des halben Mantels zum
                          eben in Register 2 geladenen Wert der Grundfläche. Somit steht nun
                          der Wert der halben Oberfläche in Register 3
S17     MULT* r3, 2   Diese Multiplikation ist eigentlich ein neuer Befehl. Hier wird nicht mit
                          einem in einer Variablen sondern mit der Konstanten „2“ multipliziert.
                          Daher lautet der Befehlscode auch nicht MULT sondern MULT*.
S18     SPEICHERE r3, z1 speichert das nunmehr in Register 3 enthaltene Endergebnis in Zelle z1
```



S19 SCHREIB z1 gibt den in z1 enthaltenen Wert auf der Ausgabe aus.
S20 STOP beendet dieses Programm

Ihr könnt nun versuchen, an dieser Stelle das Programm für die Berechnung der Fläche eines Dreiecks nach der Formel $F = g * h / 2$ zu entwerfen und dieses dann mit Partnern so wie das Programm zur Oberflächenberechnung zu simulieren.

- S1 _____
- S2 _____
- S3 _____
- S4 _____
- S5 _____
- S6 _____
- S7 _____
- S8 _____
- S9 _____
- S10 _____
- S11 _____