

Die wesentlichen Komponenten einer **CENTRAL PROCESSING UNIT (CPU)** bestehen aus der **CONTROL UNIT (CU)** und der **ARITHMETIC LOGIC UNIT (ALU)**.

Die **ALU** berechnet arithmetische und logische Funktionen, die **CU** decodiert die im Arbeitsspeicher abgelegten Befehle und führt sie aus.

In der Minimalkonfiguration beherrscht die **ALU** die arithmetische Funktion „**Addition**“ sowie die logischen Operationen „**Negation**“ (NOT) und „**Konjunktion**“ (AND). Zu Lasten der Rechenzeit lassen sich die übrigen arithmetischen und logischen Funktionen auf die genannten, minimal verfügbaren Operationen zurückführen.

## 1. Subtraktion

Die duale Subtraktion

$$\begin{array}{r} \phantom{-} \quad a_3 \quad a_2 \quad a_1 \quad a_0 \\ - \quad b_3 \quad b_2 \quad b_1 \quad b_0 \\ \hline d_3 \quad d_2 \quad d_1 \quad d_0 \end{array}$$

läßt sich auf eine duale Addition nach folgendem Verfahren zurückführen:

- Bilde das Einerkomplement des Subtrahenden  $b_3 \ b_2 \ b_1 \ b_0$ , indem man alle Ziffern negiert (invertiert; aus 0 wird 1 und aus 1 wird 0).
  - Addiere das Einerkomplement und die Zahl 1 zum Minuenden.
  - Das Ergebnis ist die gesuchte Differenz; dabei bleibt der Überlauf unberücksichtigt.
- a) Verdeutliche das genannte Verfahren anhand einiger selbst gewählter Beispiele (ein Beweis des Verfahrens ist nicht erforderlich.).
- b) Ergänze die Schaltung „4-bit-Paralleladdierer.dsim“ so, daß man nach entsprechender Umschaltung wahlweise eine duale Addition oder eine duale Subtraktion durchführen kann.  
Hinweise:
- Ersetze den HA für das least significant bit (LSB) durch einen VA, um erforderlichenfalls eine „1“ als Summand einspeisen zu können (wie?).
  - Die Invertierung der Ziffern des Subtrahenden gelingt z. B. durch den geeigneten Einsatz von XOR-Gattern.

## 2. Weitere Rechenoperationen

Gegeben sind die (im einfachsten Fall positiven ganzzahligen) Operanden a und b. Um zu verdeutlichen, wie man die „höheren“ Rechenoperationen mittels geeigneter Iteration auf die Grundoperationen „Addition“ und „Subtraktion“ zurückführen kann, schreibe und teste ein Python-Programm, welches die Operationen „Multiplikation“ ( $a*b$ ), „Division“ ( $a/b$ , ganzzahlige Division) und „Potenzierung“ ( $a**b$ ) realisiert.

## 3. Logische Operationen

Zeige exemplarisch, daß sich die logischen Verknüpfungen

- a)  $a + b$
- b)  $a \oplus b$
- c)  $a \cdot (b + \bar{c})$

auf die Operationen NOT und AND zurückführen lassen.