UB 1

A 1

a) $55_{10} \overset{7}{=} 106_7$

$55 : 7 = 7 \quad R\ 6$

$7 : 7 = 1 \quad R\ 0$

$1 : 7 = 0 \quad R\ 1$

b) $42_7 \overset{3}{=} 1010_3$

1) Rechne in 10er Sys um

$4 \cdot 7^1 + 2 \cdot 7^0 = 30_{10}$

2) Rechne in 3-System um

$30 : 3 = 10 \quad R0$

$10 : 3 = 3 \quad R1$

$3 : 3 = 1 \quad R0$

$1 : 3 = 0 \quad R1$

c) $12321_4 \overset{2}{=} 01\ 10\ 11\ 10\ 01_2$

$q = 4 \quad p = 2$ Basen

$q = p^2 \rightsquigarrow$ Dadurch kann mit 2 Ziffern von p eine Ziffer aus q dargestellt werden

| q=4 | | p=2 |
|---|---|---|
| 0 | $\hat{=}$ | 00 |
| 1 | $\hat{=}$ | 01 |
| 2 | $\hat{=}$ | 10 |
| 3 | $\hat{=}$ | 11 |

d) $17HAI_{26} \overset{36}{=} CY53_{36}$

wähle $\mathbb{Z}_{26} = \{0, \dots, 9, A, \dots P\}$

wähle $\mathbb{Z}_{36} = \{0, \dots, 9, \overset{10}{A}, \dots, Z\}$

Wir nennen nun die Ziffern A,... P in 10... 26 um

1) Schreibe ins 10er System

$1 \cdot 26^4 + 7 \cdot 26^3 + 18 \cdot 26^2 + 10 \cdot 26^1 + 19 \cdot 26^0 = 592455_{10}$

2) Rechne nach 36 System um

$592455 : 36 = 16457 \quad R3 \quad \hat{=}\ 3$

$16457 : 36 = 457 \quad R5 \quad \hat{=}\ 5$

$457 : 36 = 12 \quad R25 \quad \hat{=}\ Y$

$12 : 36 = 0 \quad R12 \quad \hat{=}\ C$

Die Zahlen 10, ..., 36 werden auf A, ..., Z gemapt

A2

A4

5 Bits: 1 Vorzeichen + 4 Wertbits

a) $15 + (-5)$ im 2er Kompliment

1) Berechne $-5$: $00101 \overset{\neg}{\leadsto} 11010 \overset{+1}{\leadsto} 11011$

2) Addition

$$\begin{array}{r} 15_{10} \\ +(-5_{10}) \\ \hline 10_{10} \end{array} \quad = \quad \begin{array}{r} 01111 \\ +\ 11011 \\ \hline 1\,01010 \end{array}$$

$16 \;\; 8 \;\; 4 \;\; 2 \;\; 1 = 10_{10}$

überlauf fällt weg

3) Subtraktion von $15_{10} - 5_{10}$

$$\begin{array}{r} 01111 \\ -\ 00101 \\ \hline 01010 \end{array}$$

Warum Rückführung der Subtraktion auf die Addition?

Der Vorteil ist, dass so nicht eine Extra regel für den Fall $0-1=-1$ eingeführt werden muss. Bei diesem Fall hat man als ergebnis eine negative Zahl, die zwar wieder als Übertrag aufgefasst werden kann, aber es mussten hier für 2 Arten von Addierern geschrieben/gefertigt werden. So ist nur eine Operation mit allen 4 Fällen notwendig $\leadsto$ Spart Komponenten und Laufzeit

b) 3+(−2) 3+(−3) 3+(−4)

00010 ⇝ 11101 ⇝ 11110 00011 ⇝ 11100 ⇝ 11101 00100 → 11011 ⇝ 11100

im einer Kompliment

$$\begin{array}{r} 00011 \\ 11110 \\ + \\ \hline 100001 \end{array} \quad \begin{array}{r} 00011 \\ 11101 \\ + \\ \hline 100000 \end{array} \quad \begin{array}{r} 00011 \\ 11101 \\ + \\ \hline 100000 \end{array} \quad \begin{array}{r} 00011 \\ 11100 \\ + \\ \hline 11111 \end{array} \quad \begin{array}{r} 00011 \\ 11100 \\ + \\ \hline 11111 \end{array} \quad \begin{array}{r} 00011 \\ 11011 \\ \\ \hline 11110 \end{array}$$

Im zweier Kompliment wird bei negativen Zahlen eine 1 hinzuaddiert, damit es keine doppelte Darstellung der 0 geben kann als +0 und −0. Dadurch ist jede Zahl eindeutig und kann eindeutig berechnet werden.

c)

$$\begin{array}{rl} 01111 & 15 \\ 00101 & 5 \\ + \\ \hline 10100 & = -12 \end{array} \quad \begin{array}{rl} 10001 & -15 \\ 11011 & -5 \\ + \\ \hline 101100 & = 12 \end{array}$$

01111 ⇝ 10000 ⇝ 10001 $\boxed{-15}$

Es stehen uns 5 Bits zur Zahlendarstellung zur verfügung in den natürlichen Zählen sind damit 32 Zahlen (0,…,31) darstellbar. Da wir aber noch negative Zahlen zusätzlich darstellen wollen halbiert sich (im groben) die Anzahl der positiv darstellbaren Zahlen (im negativen ist eine Zahl mehr darstellbar).

Daraus folgt (−16,…,0,…15) sind nur darstellbar.

Da wir nun aber versuchen eine Addition mit dem Ergebnis 20 bzw. −20 durchzuführen, entsteht ein Überlauf. Das Most Significant Bit, welches im groben das Vorzeichen repräsentiert, wird durch die Addition mit einem falschen Wert überschrieben und verletzt damit seine Funktion. Damit gelingt keine korrekte Interpretation mehr