

Aufgabe 1

(a) 3+3

$$\begin{array}{r} 00011 \\ + 000_11_11 \\ \hline 00110 \end{array}$$

Es ergibt sich $3_{10} + 3_{10} = (0110)_{2K} = 6_{10}$.

3-4

$$\begin{array}{r} 00011 \\ + 11100 \\ \hline 11111 \end{array}$$

Es ergibt sich $3_{10} - 4_{10} = (1111)_{2K} = -1_{10}$.

-3-5

$$\begin{array}{r} 11101 \\ + 1_11_10_11_11 \\ \hline 11000 \end{array}$$

Es ergibt sich $-3_{10} - 5_{10} = (1000)_{2K} = -8_{10}$.

7+5

$$\begin{array}{r} 00111 \\ + 00_11_10_11 \\ \hline 01100 \end{array}$$

Es ergibt sich $7_{10} + 5_{10} \neq (1100)_{2K} = -4_{10}$. Überlauf!

-3-6

$$\begin{array}{r} 11101 \\ + 1_11010 \\ \hline 10111 \end{array}$$

Es ergibt sich $-3_{10} - 6_{10} \neq (0111)_{2K} = 7_{10}$. Unterlauf!

(b) Der Befehlsatz unserer ALU wird folgendermaßen kodiert:

Befehl	Bedeutung	Kodierung CMD
ADD	$ERG := OP0 + OP1$	0
SUB	$ERG := OP0 - OP1$	1

Die Realisierung einer ALU nutzt den beschriebenen Befehlsatz. Hierbei ist die gewählte Bezeichnung der Eingaben und Ausgaben zu beachten:

$$\begin{aligned} OP0 &= (OP0_3 \ OP0_2 \ OP0_1 \ OP0_0), \\ OP1 &= (OP1_3 \ OP1_2 \ OP1_1 \ OP1_0) \text{ und} \\ ERG &= (ERG_3 \ ERG_2 \ ERG_1 \ ERG_0). \end{aligned}$$

Für die Berechnung des Overflow-Flags OVF wird zusätzlich ein Sicherungsbit berechnet. Für das Overflow-Flag können sowohl Carry-Leitungen oder Summen-Leitungen mittels XOR verglichen werden.

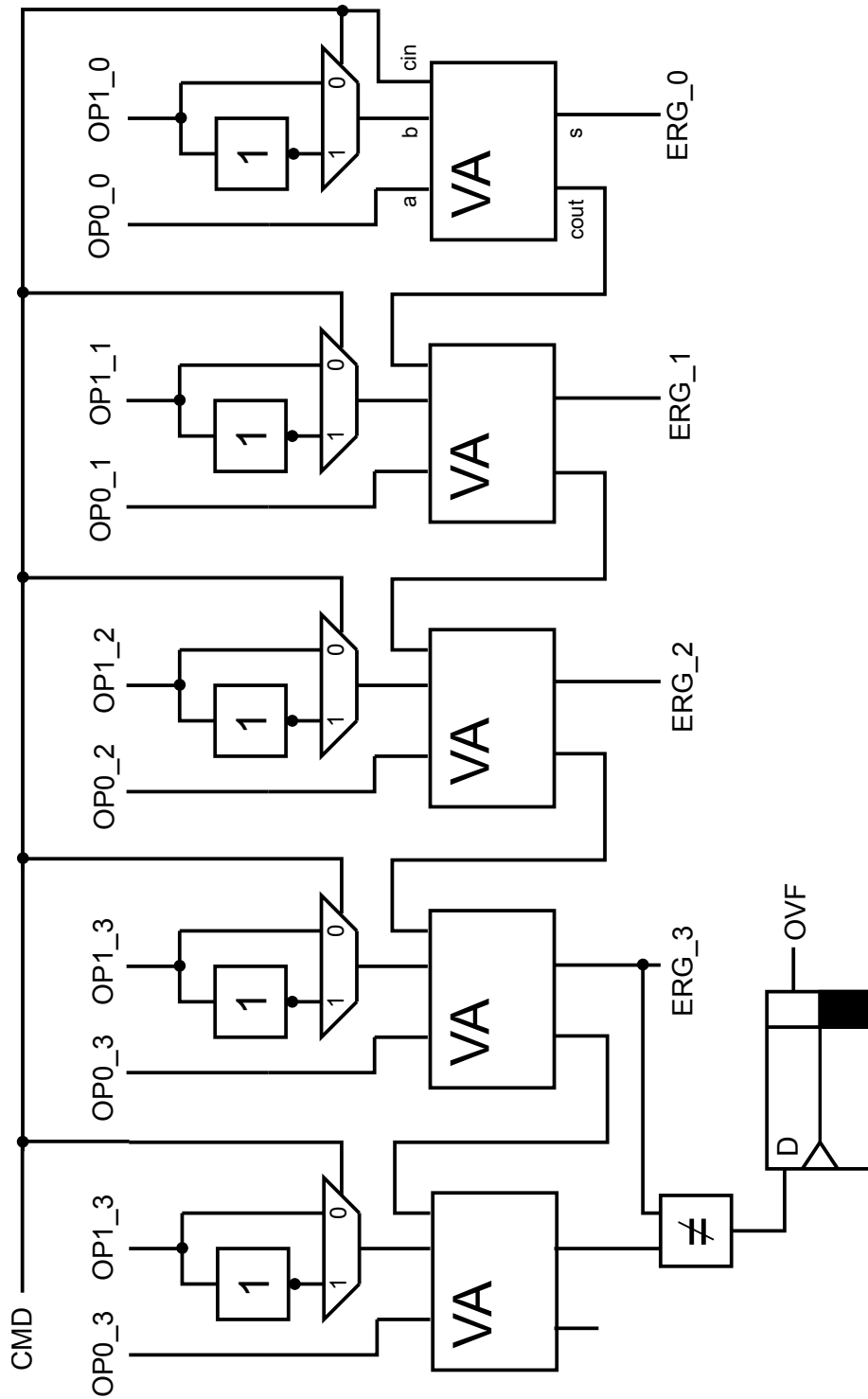


Abbildung 1: 4-Bit-ALU