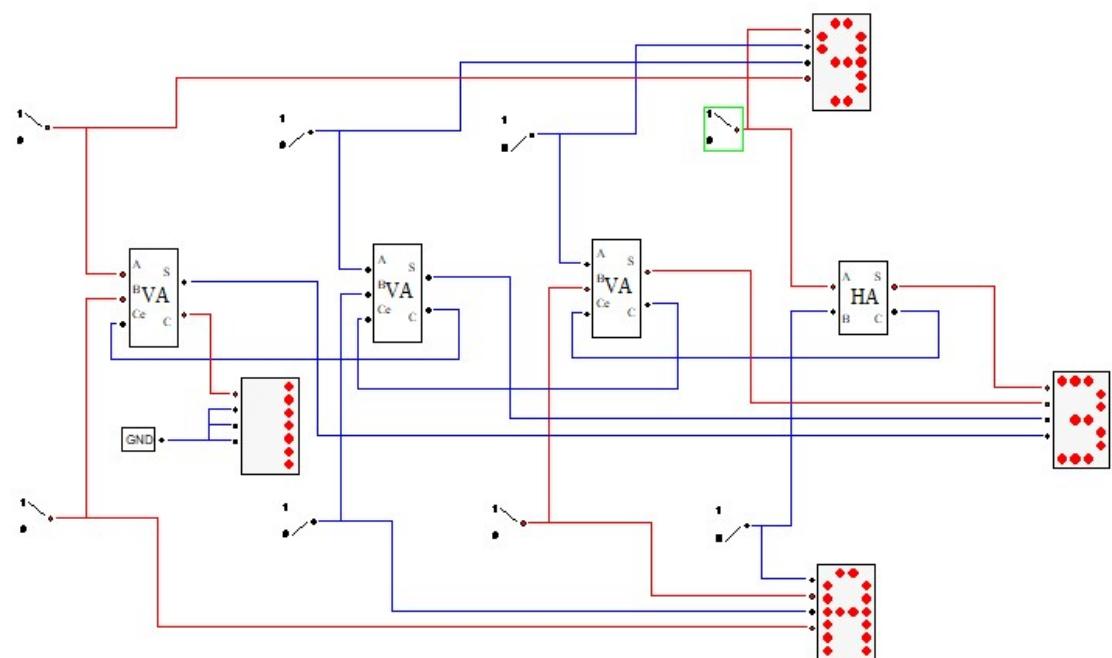
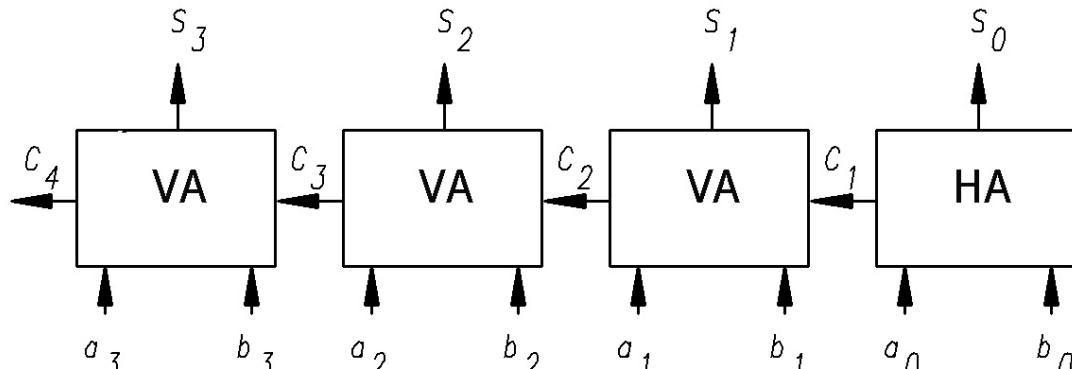


## Addier-Schaltungen für Dualzahlen

$$\begin{array}{r}
 & a_3 & a_2 & a_1 & a_0 \\
 + & b_3 & b_2 & b_1 & b_0 \\
 \hline
 s_4 & s_3 & s_2 & s_1 & s_0
 \end{array}$$

### 1. Paralleladdierer mit serielltem Übertrag (hier: 4-Bit-Addierer)

Für das Least Significant Bit (LSB) genügt ein Halbaddierer (HA); die höherwertigen Bits erfordern jeweils einen Volladdierer, da hier der Übertrag aus der vorherigen Stelle zu berücksichtigen ist.

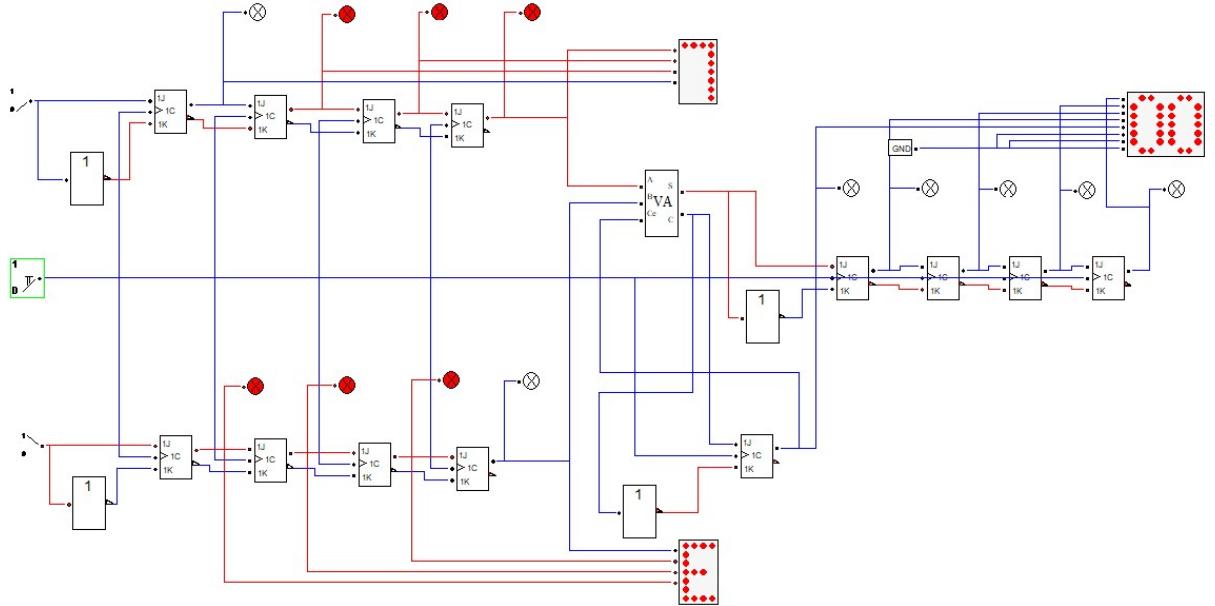


$$\begin{array}{r}
 \text{Dezimal: } 09 \quad \text{Hexadezimal: } 09 \quad \text{Dual: } 0000 \ 1001 \\
 + 10 \qquad \qquad \qquad + 0A \qquad \qquad \qquad + 0000 \ 1010 \\
 \hline
 19 \qquad \qquad \qquad 13 \qquad \qquad \qquad 0001 \ 0011
 \end{array}$$

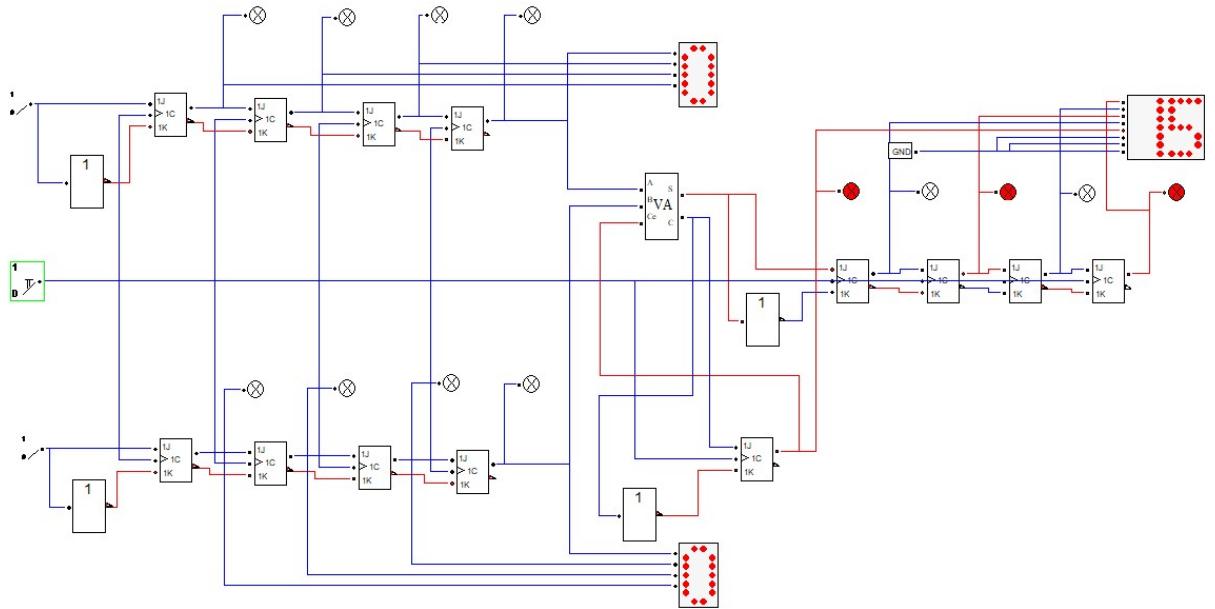
### 2. Serieller 1-Bit-Addierer für 4-stellige Dualzahlen

Die Operanden werden jeweils in einem 4-Bit-Schieberegister abgelegt, nach 4 Taktimpulsen finden wir das Ergebnis (hier: die Summe) in einem weiteren 4-Bit-Schieberegister.

Da der Übertrag aus der vorherigen Stelle für die Addition in der aktuellen Stelle zu berücksichtigen ist, wird er in einem Flip-Flop zwischengespeichert. Dieses Flip-Flop liefert auch das Most Significant Bit (MSB) des Ergebnisses.



Nach 4 Taktimpulsen (hier: Triggerung der Flip-Flops auf der steigenden Taktflanke):



$$\begin{array}{r}
 \text{Dezimal: } 07 \\
 + 14 \\
 \hline
 21
 \end{array}$$

$$\begin{array}{r}
 \text{Hexadezimal: } 07 \\
 + 0E \\
 \hline
 15
 \end{array}$$

$$\begin{array}{r}
 \text{Dual: } 0000 \ 0111 \\
 + 0000 \ 1110 \\
 \hline
 0001 \ 0101
 \end{array}$$