Halbaddierer und Volladdierer

Die Ziffern einer im Dezimalsystem geschriebenen Zahl a ergeben sich als Aneinanderreihung der Koeffizienten aus der Dezimalzerlegung (Summe von Zehnerpotenzen) von a; entsprechend erhalten wir die Darstellung von a im Dualsystem als Aneinanderreihung der Koeffizienten aus der Dualzerlegung (Summe von Zweierpotenzen).

$87_{dezimal} = 8 \cdot 10^1 + 7 \cdot 10^0$

$87_{dezimal} = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1010111_{dual}$

Addition der Dualzahlen
$a = a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$ und $b = b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$:

		a_3	a_2	a_1	a_0
+		b_3	b_2	b_1	b_0
	s_4	s_3	s_2	s_1	s_0

		1	1	0	1
+		1	0	1	1
	1	1	0	0	0

Den Übertrag („carry"), der sich aus der i-ten Stelle ergibt und der bei der Addition in der (i + 1)-ten Stelle zu berücksichtigen ist, bezeichnen wir mit c_{i+1}; $i \geq 0$.

Für die 0-te Stelle genügt ein Halbaddierer mit den Eingängen a_0 und b_0 und den Ergebnissen s_0 und c_1; die Addition in der i-ten Stelle, $i \geq 1$, erfordert einen Volladdierer mit den Eingängen a_i, b_i, c_i und den Ergebnissen s_i und c_{i+1}.

Halbaddierer HA

Wahrheitstafel:

a_0	b_0	s_0	c_1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Wir ermitteln für $\mathbf{s_0}$ und $\mathbf{c_1}$ jeweils die disjunktive Normalform („Disjunktion der Konjunktionen"):

$$s_0 = \overline{a_0} \cdot b_0 + a_0 \cdot \overline{b_0} = a_0 \oplus b_0$$

$$c_1 = a_0 \cdot b_0$$

Volladdierer VA

Wahrheitstafel:

a_i	b_i	c_i	s_i	c_{i+1}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Wir ermitteln für $\mathbf{s_i}$ und $\mathbf{c_{i+1}}$ jeweils die disjunktive Normalform („Disjunktion der Konjunktionen") und vereinfachen ggf. die booleschen Funktionsterme:

$$s_i = \bar{a}_i \cdot \bar{b}_i \cdot c_i + \bar{a}_i \cdot b_i \cdot \bar{c}_i + a_i \cdot \bar{b}_i \cdot \bar{c}_i + a_i \cdot b_i \cdot c_i$$

ohne Index i geschrieben:

$$s = \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot \bar{c} + a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c$$

$$s = (\bar{a} \cdot b + a \cdot \bar{b}) \cdot \bar{c} + \bar{a} \cdot \bar{b} \cdot c + a \cdot b \cdot c$$

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

$$s = (a \oplus b) \cdot \bar{c} + (0 + \bar{a} \cdot \bar{b} + a \cdot b + 0) \cdot c$$

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

$$s = (a \oplus b) \cdot \bar{c} + [(\bar{a} + b) \cdot (a + \bar{b})] \cdot c$$

$$s = (a \oplus b) \cdot \bar{c} + [(\bar{a} + \bar{\bar{b}}) \cdot (\bar{\bar{a}} + \bar{b})] \cdot c$$

$$s = (a \oplus b) \cdot \bar{c} + [\overline{(a \cdot \bar{b})} \cdot \overline{(\bar{a} \cdot b)}] \cdot c$$

$$s = (a \oplus b) \cdot \bar{c} + [\overline{a \cdot \bar{b} + \bar{a} \cdot b}] \cdot c$$

$$s = (a \oplus b) \cdot \bar{c} + (\overline{a \oplus b}) \cdot c$$

$$s = (a \oplus b) \oplus c$$

mit Index i erhält man:

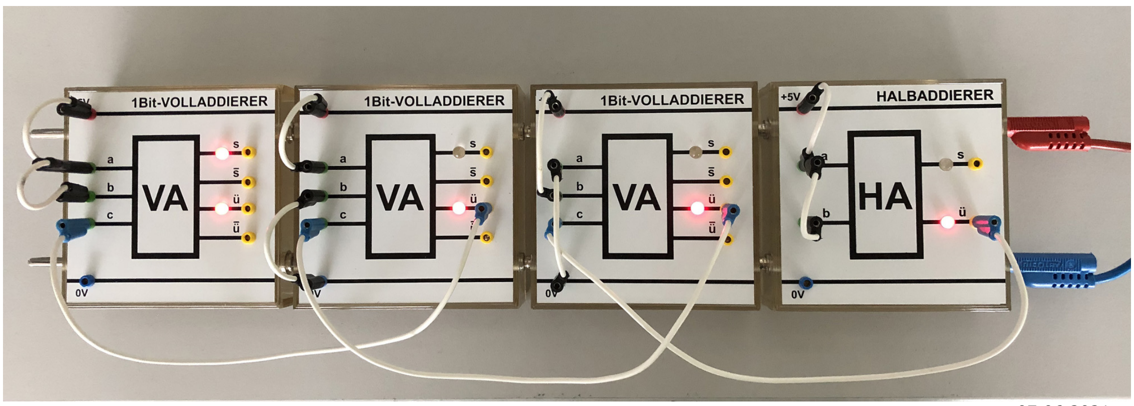
$$s_i = (a_i \oplus b_i) \oplus c_i$$

$$c_{i+1} = \bar{a}_i \cdot b_i \cdot c_i + a_i \cdot \bar{b}_i \cdot c_i + a_i \cdot b_i \cdot \bar{c}_i + a_i \cdot b_i \cdot c_i$$

$$c_{i+1} = (\bar{a}_i \cdot b_i + a_i \cdot \bar{b}_i) \cdot c_i + a_i \cdot b_i \cdot (\bar{c}_i + c_i)$$

$$c_{i+1} = (\bar{a}_i \cdot b_i + a_i \cdot \bar{b}_i) \cdot c_i + a_i \cdot b_i \cdot 1$$

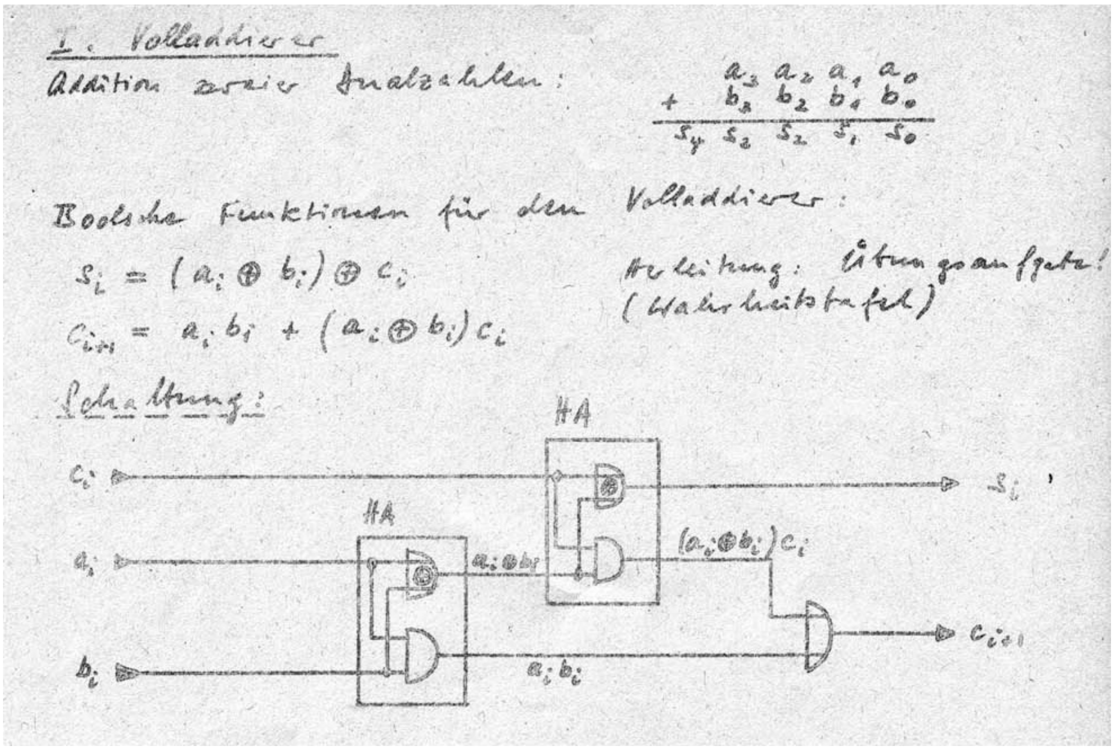
$$c_{i+1} = (a_i \oplus b_i) \cdot c_i + a_i \cdot b_i$$

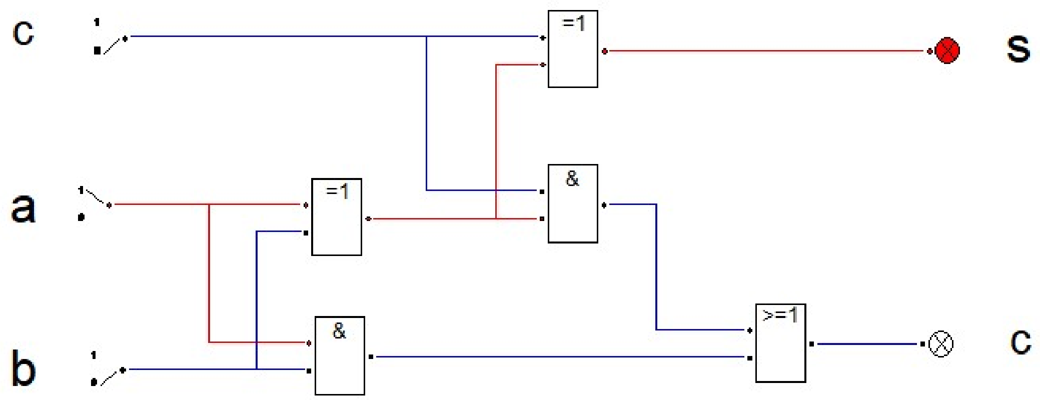


07.06.2021

Halbaddierer (HA) und Volladdierer (VA)

Schaltungen



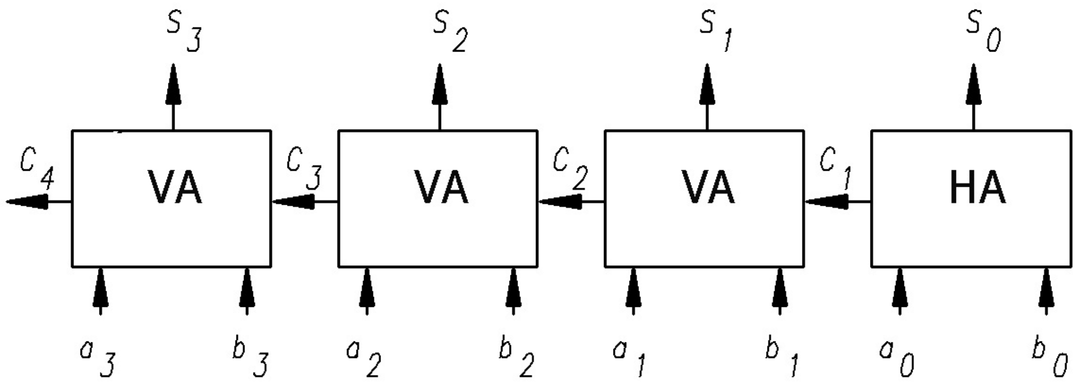


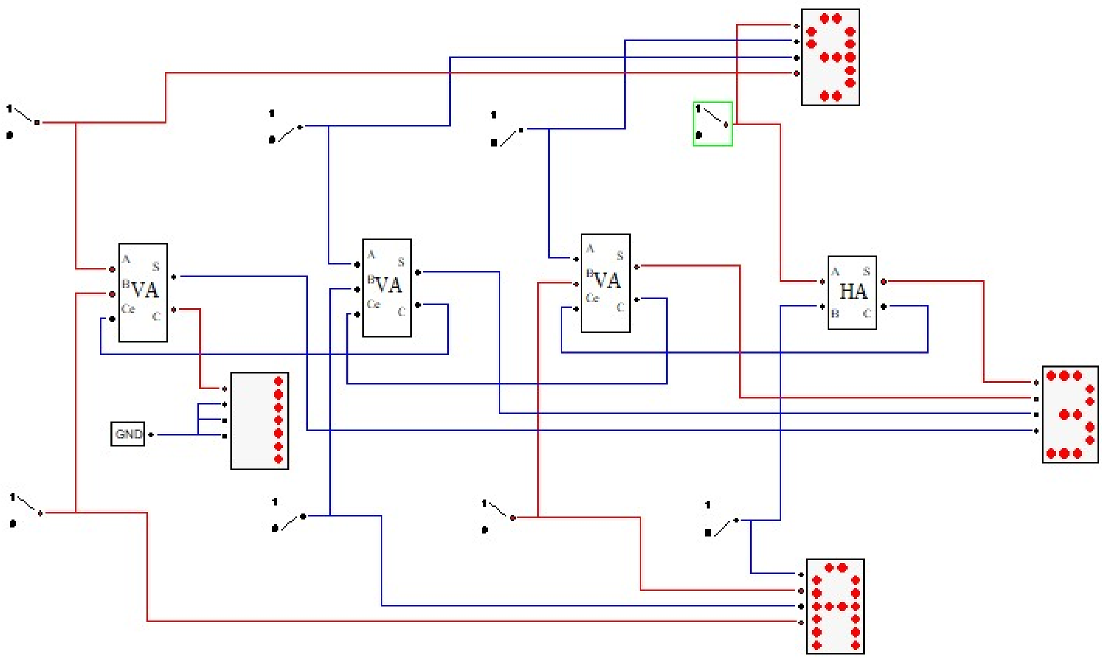
Addier-Schaltungen für Dualzahlen

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

1. Paralleladdierer mit seriellem 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.



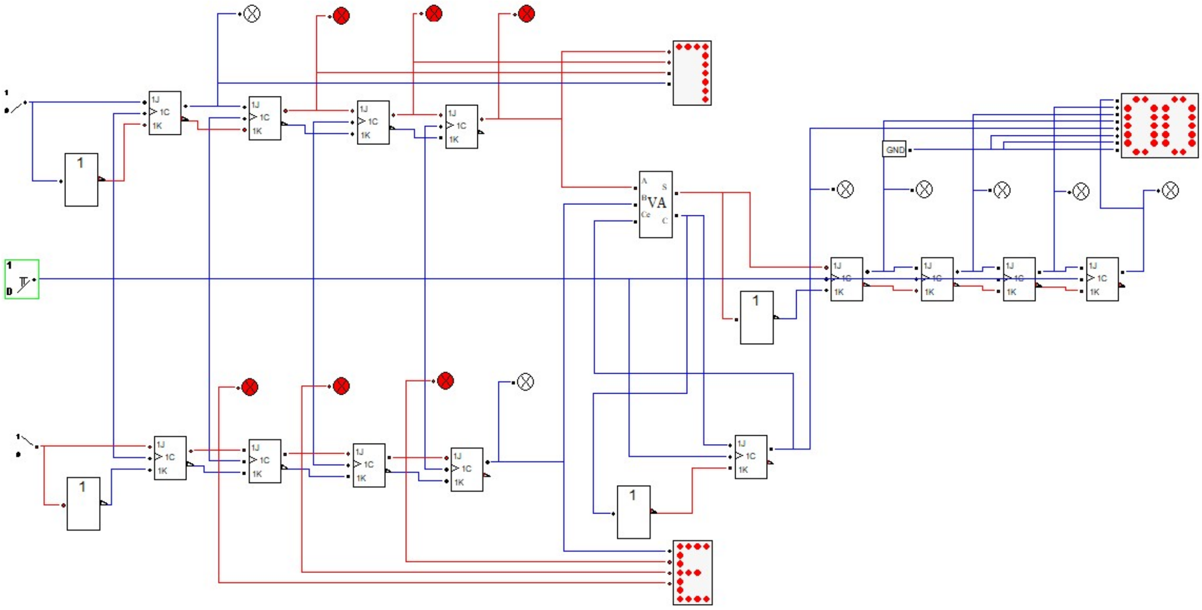



```
Dezimal:   09   Hexadezimal:   09     Dual:  0000 1001
         + 10                + 0A          + 0000 1010
           19                  13            0001 0011
```

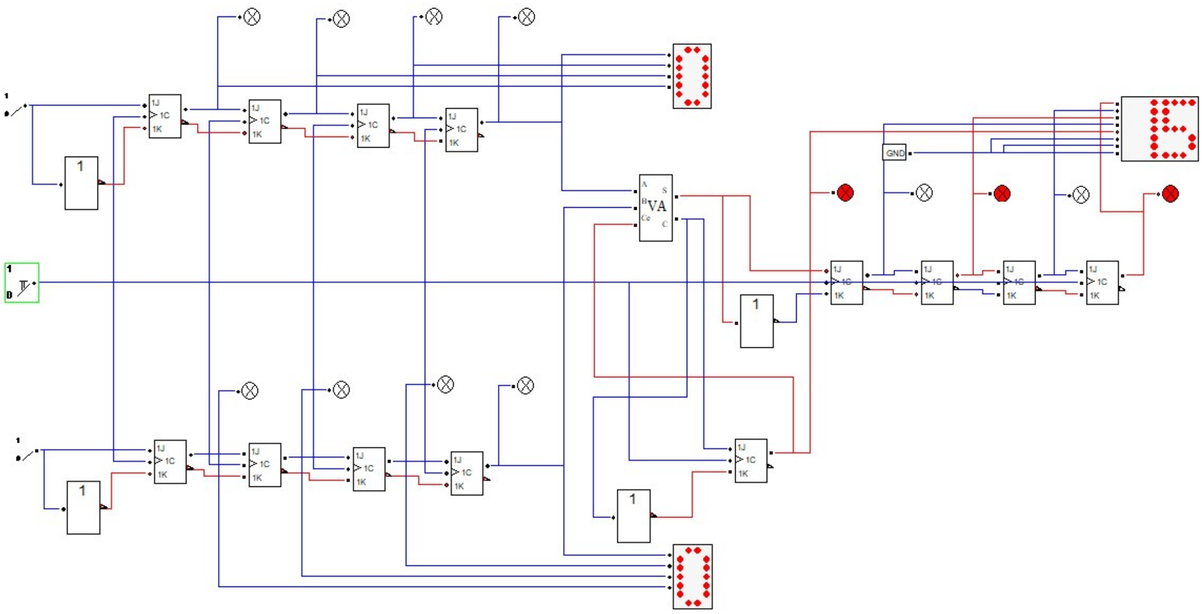
2. Serieller 1-Bit-Addierer für 4-stellige Dualzahlen

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):



```
Dezimal:   07   Hexadezimal:  07    Dual:   0000 0111
         + 14               + 0E          + 0000 1110
           21                 15            0001 0101
```