

DIGITALE SCHALTUNGEN



Robert Wille (robert.wille@jku.at)

Sebastian Pointner (sebastian.pointner@jku.at)

Institut für Integrierte Schaltungen

Abteilung für Schaltkreis- und Systementwurf

INHALT DER VORLESUNG



■ Grundlagen

- ☐ Beschreibungen über „0“ und „1“ (Boolesche Algebra)
- ☐ Beschreibungen von Schaltungen

■ Rechnen

- ☐ Darstellung von Zahlen
- ☐ Digitale Schaltungen für Addition, Subtraktion, Multiplikation

■ Speichern

- ☐ Sequentielle Schaltungen
- ☐ Speicherelemente

■ Steuern

- ☐ Endliche Automaten
- ☐ Synthese von Steuerwerken

■ Entwerfen

- ☐ Synthese von allgemeinen Schaltungen
- ☐ Logikminimierung

INHALT DER VORLESUNG



■ Grundlagen

- ☐ Beschreibungen über „0“ und „1“ (Boolesche Algebra)
- ☐ Beschreibungen von Schaltungen

■ Rechnen

- ☐ Darstellung von Zahlen
- ☐ **Digitale Schaltungen für Addition, Subtraktion, Multiplikation**

■ Speichern

- ☐ Sequentielle Schaltungen
- ☐ Speicherelemente

■ Steuern

- ☐ Endliche Automaten
- ☐ Synthese von Steuerwerken

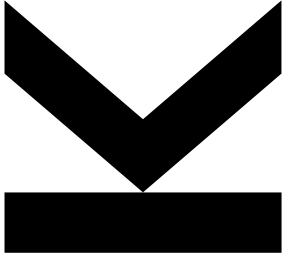
■ Entwerfen

- ☐ Synthese von allgemeinen Schaltungen
- ☐ Logikminimierung

SCHALTKREIS

- Hier: kombinatorische Schaltkreise
- Gerichteter, zyklener Graph
- Knoten repräsentieren
 - Primäre Eingänge
 - Primäre Ausgänge
 - Gatter (i.d.R. basierend auf vorher festgelegter Gatterbibliothek)
- Kanten repräsentieren
 - Signale zwischen den Gattern bzw. primären Eingängen/Ausgängen
- Gängige Kostenmaße
 - Anzahl der Gatter (Größe)
 - Tiefe, d.h. Zahl der Gatter auf dem längsten Pfad von einem primären Eingang zu einem primären Ausgang (Geschwindigkeit)

ADDITION



Robert Wille (robert.wille@jku.at)

Sebastian Pointner (sebastian.pointner@jku.at)

Institut für Integrierte Schaltungen

Abteilung für Schaltkreis- und Systementwurf

ADDIERER

Gegeben: 2 positive Binärzahlen $\langle a \rangle = \langle a_{n-1} \dots a_0 \rangle$ und $\langle b \rangle = \langle b_{n-1} \dots b_0 \rangle$,
Eingangsübertrag c aus $\{0,1\}$

Gesucht: Schaltkreis, der Binärdarstellung s von
 $\langle a \rangle + \langle b \rangle + c$ berechnet

Wegen $\langle a \rangle + \langle b \rangle + c \leq 2 \times (2^n - 1) + 1 = 2^{n+1} - 1$
genügen $n+1$ Ausgänge des Schaltkreises.

DER HALBADDIERER (*HA*)

Der Halbaddierer dient zur Addition zweier 1-Bit-Zahlen *ohne* Eingangsübertrag.

Er berechnet die Funktion:

$$ha : B^2 \rightarrow B^2$$

$$\text{mit } ha(a_0, b_0) = (s_1, s_0)$$

$$\text{mit } 2s_1 + s_0 = a_0 + b_0$$

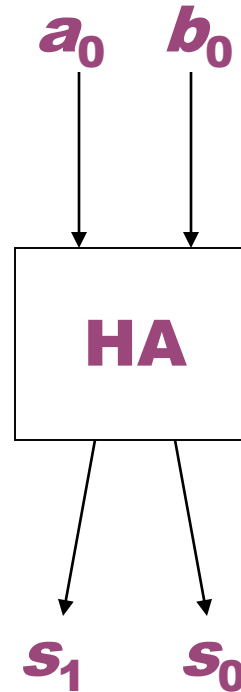
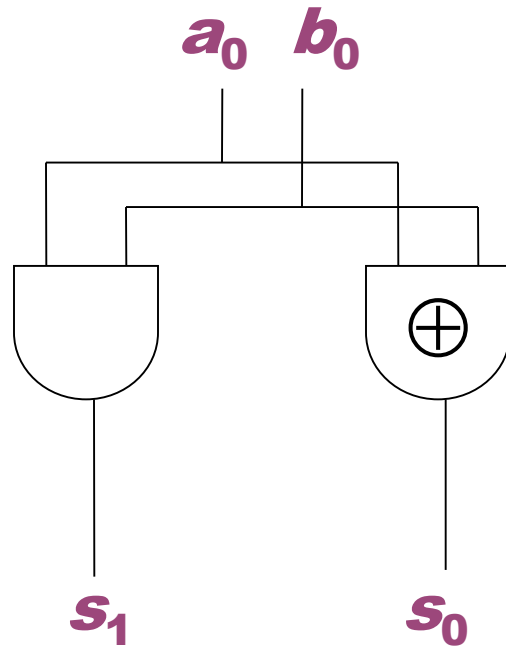
FUNKTIONSTABELLE DES HA

a_0	b_0	ha_1	ha_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Folglich:

$$ha_0 = a_0 \oplus b_0 \quad ha_1 = a_0 \wedge b_0$$

SCHALTKREIS EINES HALBADDIERERS



Kosten und Tiefe eines HA:

$$\mathbf{C(HA) = 2, \text{ depth(HA) = 1}}$$

DER VOLLADDIERER (*FA*)

Der Volladdierer dient zur Addition zweier 1-Bit-Zahlen *mit* Eingangsübertrag.

Er berechnet die Funktion:

$$\textbf{fa: } \mathbf{B^3} \rightarrow \mathbf{B^2}$$

$$\textbf{mit } \textbf{fa}(a_0, b_0, c) = (s_1, s_0)$$

$$\textbf{mit } 2s_1 + s_0 = a_0 + b_0 + c$$

FUNKTIONSTABELLE DES FA

a_0	b_0	c	fa_1	fa_0
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

VOLLADDIERER ALS FUNKTION VON HAS

Aus der Tabelle folgt:

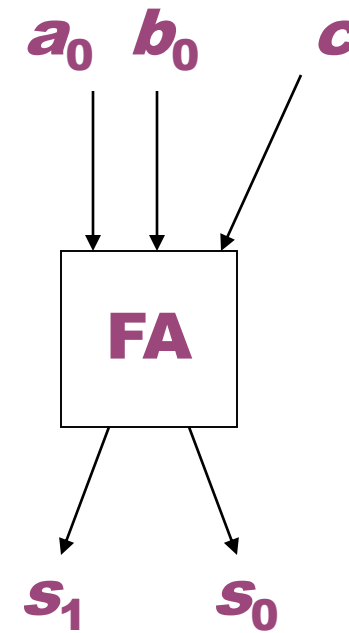
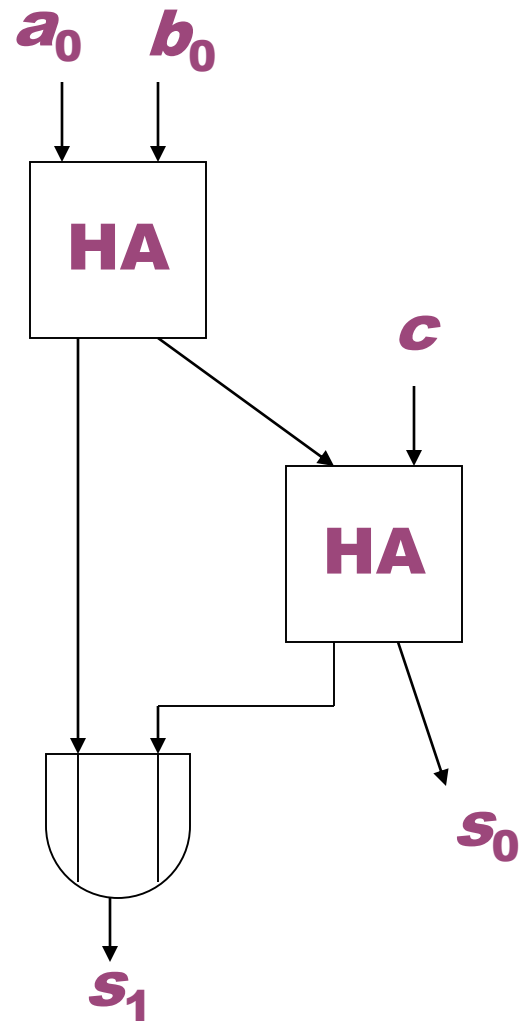
$$fa_0 = a_0 \oplus b_0 \oplus c = ha_0(c, ha_0(a_0, b_0))$$

$$\begin{aligned} fa_1 &= a_0 \wedge b_0 \vee c \wedge (a_0 \oplus b_0) \\ &= ha_1(a_0, b_0) + ha_1(c, ha_0(a_0, b_0)) \end{aligned}$$

Kosten und Tiefe eines FA:

$$\mathbf{C(FA) = 5, depth(FA) = 3}$$

SCHALTKREIS EINES VOLLADDIERERS



REALISIEREN DER SCHULMETHODE: CARRY RIPPLE ADDIERER (CR)

Hierarchisches Vorgehen:
(induktive Definition)

Für $n=1$: $CR_1 = FA$

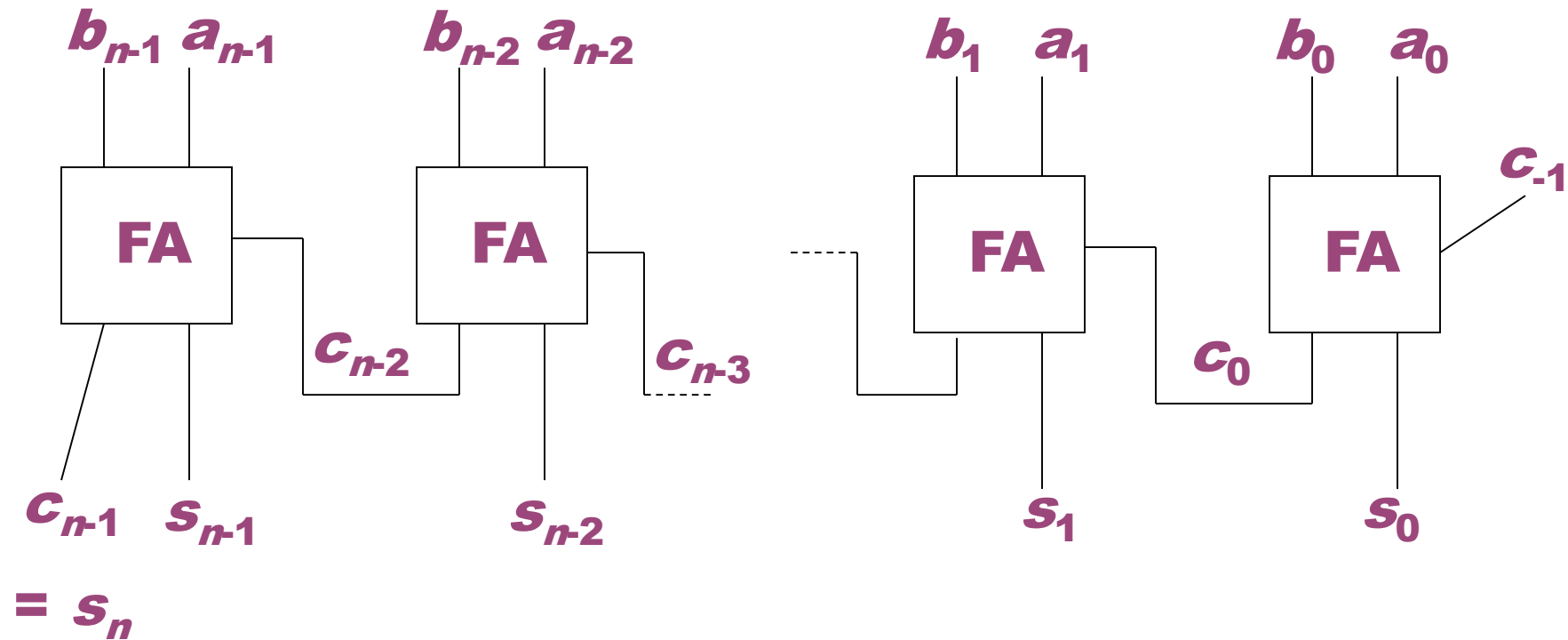
Für $n>1$: Schaltkreis CR_n wie folgt definiert

Bezeichnung:

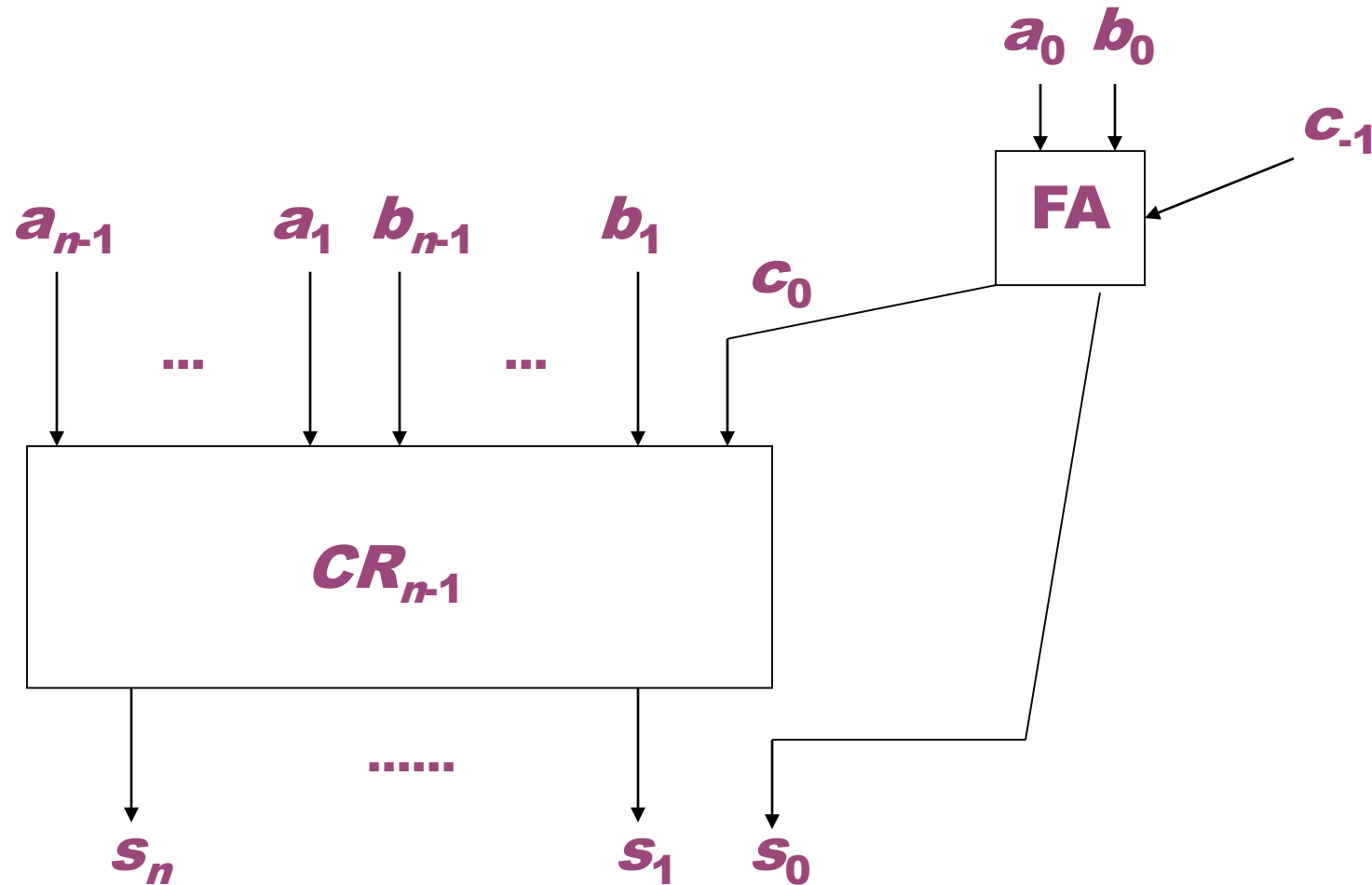
Bezeichne den Eingangsübertrag mit c_{-1} , den
Übertrag

von Stelle i nach $i+1$ mit c_i

AUFBAU EINES CARRY RIPPLE ADDIERERS



SCHALTBILD DES N-CARRY RIPPLE ADDIERERS (CR_N)



DATEN EINES CARRY RIPPLE ADDIERERS

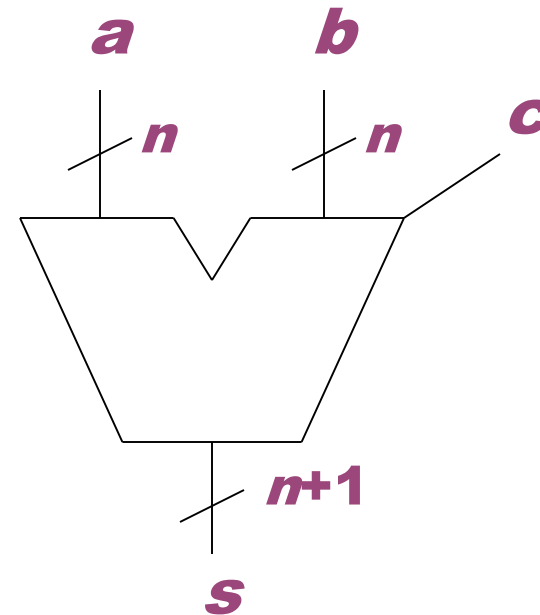
Kosten eines CR_n :

$$C(CR_n) = n \cdot C(FA) = 5n$$

Tiefe eines CR_n :

$$\text{depth}(CR_n) = 3 + 2(n-1)$$

Schaltbild eines n -Bit-Addierers:



SUBTRAKTION



Robert Wille (robert.wille@jku.at)

Sebastian Pointner (sebastian.pointner@jku.at)

Institut für Integrierte Schaltungen

Abteilung für Schaltkreis- und Systementwurf

BEHAUPTUNG

Zur Addition von $(n+1)$ -Bit Zweierkomplementzahlen kann man $(n+1)$ -Bit-Binäraddierer benutzen.

Der Test, ob das Ergebnis durch eine $(n+1)$ -Bit-Zweierkomplementzahl darstellbar ist, d. h. ob das Ergebnis aus $R_n = \{-2^n, \dots, 2^n-1\}$ ist, lässt sich zurückführen auf den Test $c_n = c_{n-1}$.

SUBTRAHIERER

Wegen $-[b] = [\bar{b}] + 1$ kann $[a] - [b]$ zurückgeführt werden auf $[a] + [\bar{b}] + 1$.

→ Schaltkreis für Subtrahierer aus Addiererschaltkreis

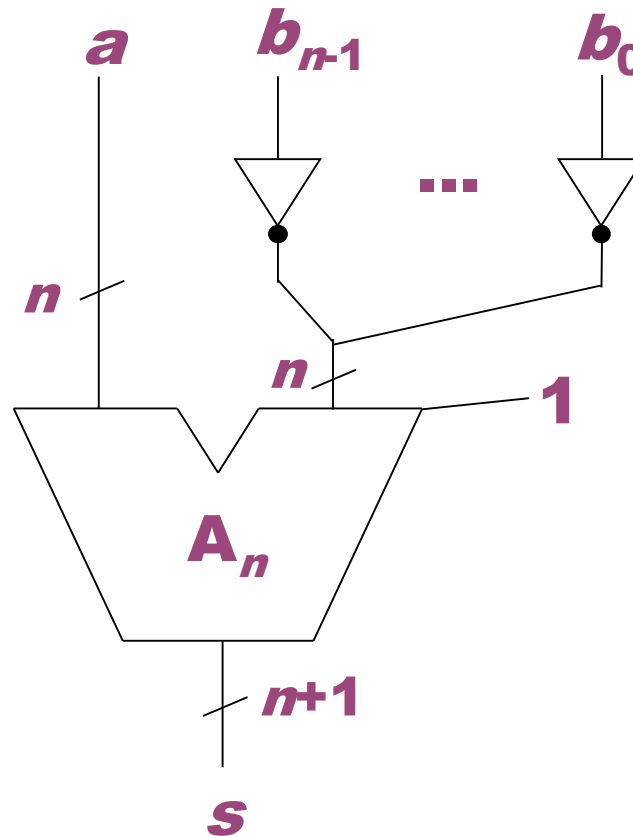
→ kombinierter Addierer/Subtrahierer

BEISPIEL

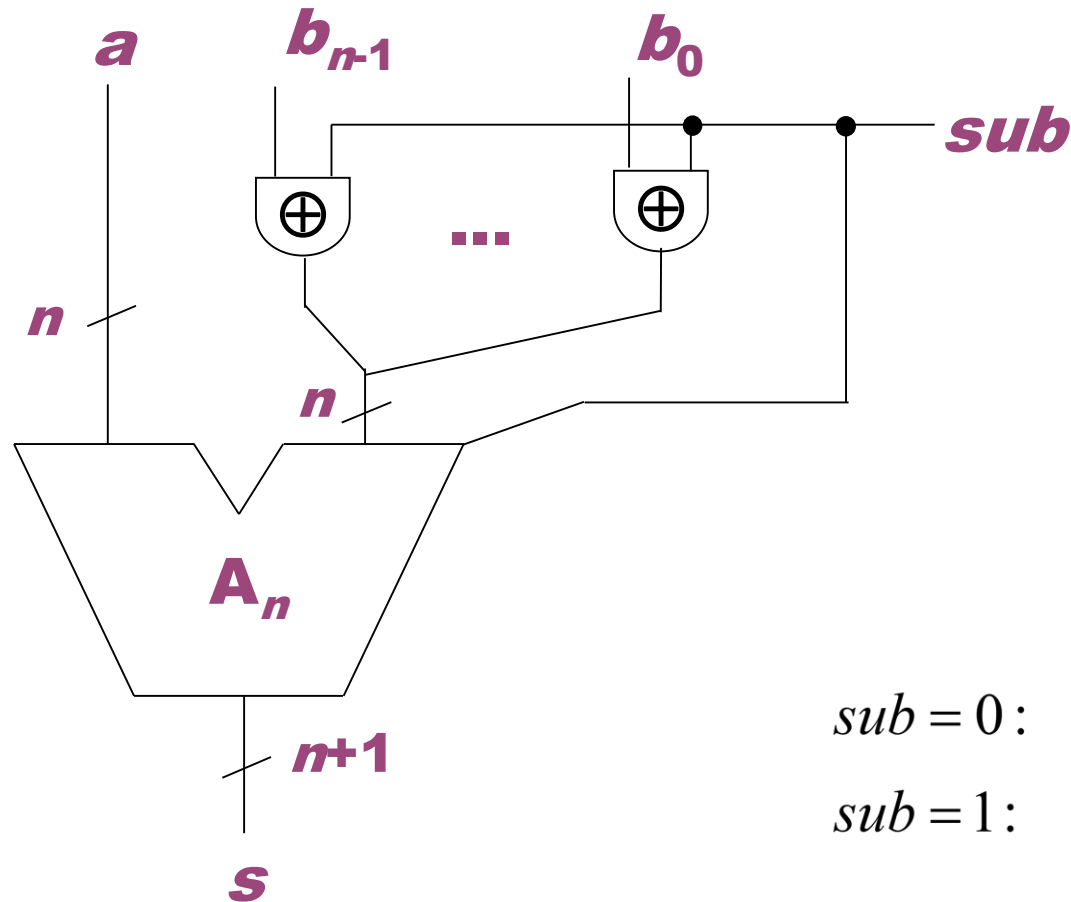
$$[a] = [0110] = 6_{10}, \quad [b] = [0111] = 7_{10}, \quad [\bar{b}] = [1000]$$

$$\begin{array}{rcccc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ \hline & & & 1 \\ 1 & 1 & 1 & 1 \\ = (-1)_{10} \end{array}$$

SCHALTBILD EINES SUBTRAHIERERS



SCHALTBILD FÜR EINEN ADDIERER/SUBTRAHIERER



$$b_i \oplus 0 = b_i$$

$$b_i \oplus 1 = \bar{b}_i$$

$$sub = 0: \quad [a] + [b] + 0$$

$$sub = 1: \quad [a] + [\bar{b}] + 1 = [a] - [b]$$

MULTIPLIKATION



Robert Wille (robert.wille@jku.at)

Sebastian Pointner (sebastian.pointner@jku.at)

Institut für Integrierte Schaltungen

Abteilung für Schaltkreis- und Systementwurf

MULTIPLIZIERER

- Multiplikation zweier Bits

$$B_0 * A_0 = B_0 A_0$$

⇒ Und-Verknüpfung

- Multiplikation einer mehrstelligen Zahl mit einem Bit

$$(B_1, B_0) * A_0 = (B_1 A_0, B_0 A_0)$$

⇒ Und-Verknüpfung der Stellen mit dem Bit

MULTIPLIZIERER

■ Multiplikation zweier mehrstelliger Zahlen

$$(B_1, B_0) * (A_1, A_0)$$

$$= (B_1, B_0) * A_0 + (B_1, B_0) * A_1 * 2$$

$$= (B_1 A_0, B_0 A_0) + (B_1 A_1, B_0 A_1, 0)$$

$$\begin{array}{r}
 1\ 0\ 1\ 1\ * \ 0\ 1\ 0\ 1 \\
 \hline
 0\ 0\ 0\ 0 \\
 1\ 0\ 1\ 1 \\
 0\ 0\ 0\ 0 \\
 1\ 0\ 1\ 1 \\
 \hline
 1 \\
 \hline
 0\ 1\ 1\ 0\ 1\ 1\ 1
 \end{array}
 \quad
 \begin{array}{l}
 11 * 5 \\
 \\
 \\
 \\
 \\
 55
 \end{array}$$

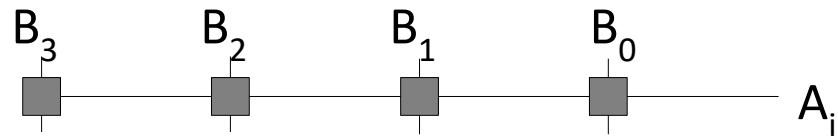
- ☐ Berechnung der Teilprodukte $(B_{n-1}, \dots, B_0) * A_i$, mit $0 \leq i < n$
- ☐ Stellengerechte Summenbildung
- ☐ Teilprodukt mit A_i muss um i Stellen nach links verschoben werden

MULTIPLIZIERER

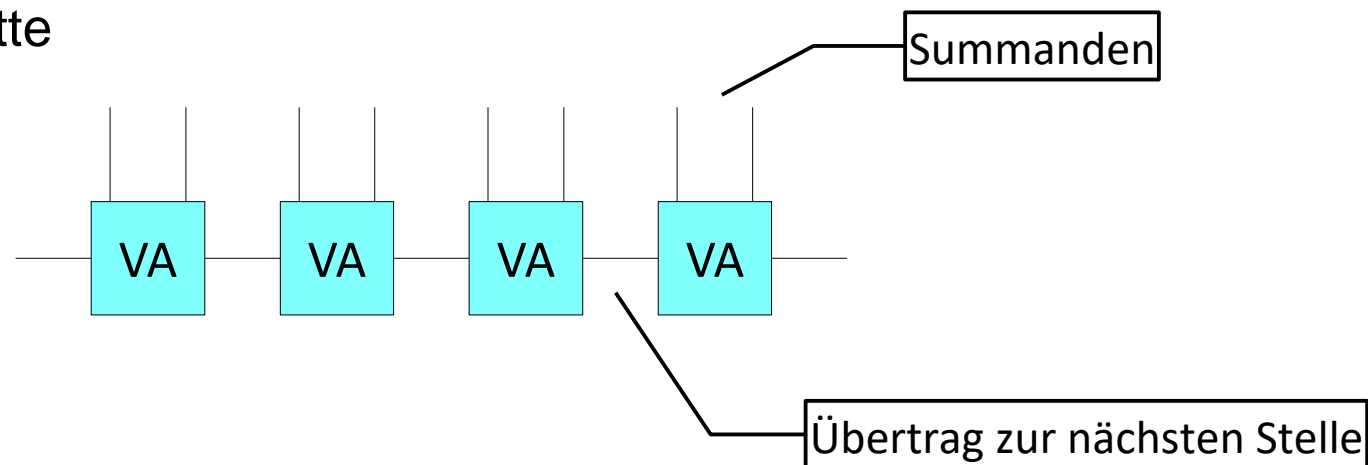
- Verschiedene Varianten möglich
 - ☐ Vollständig parallele Schaltung
 - ☐ Bildung und Aufsummieren der Teilprodukte nacheinander
 - ☐ Mischformen

VEREINFACHTE DARSTELLUNG

- Und-Verknüpfung von B mit A_i

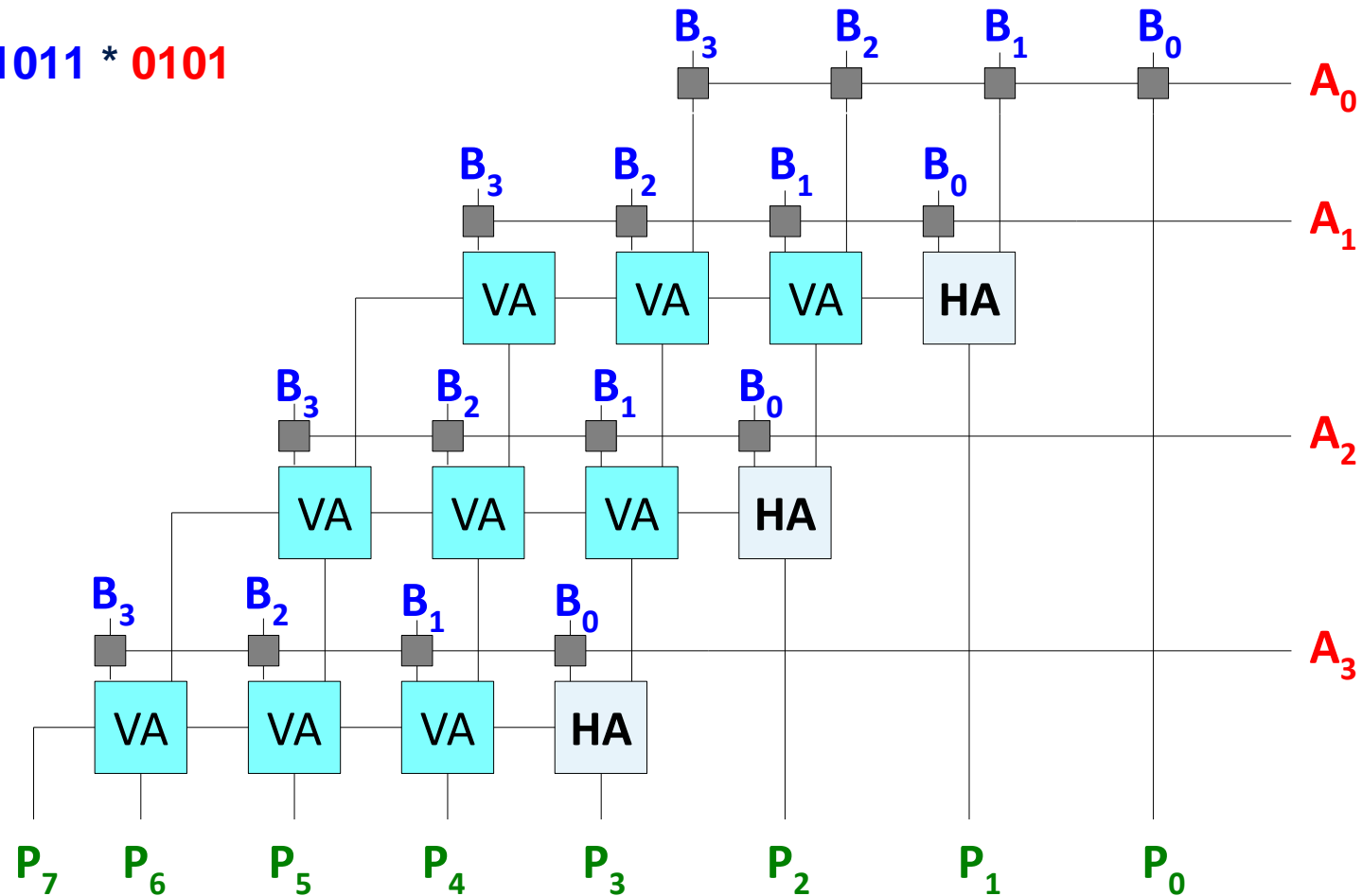


- Volladdierer-Kette

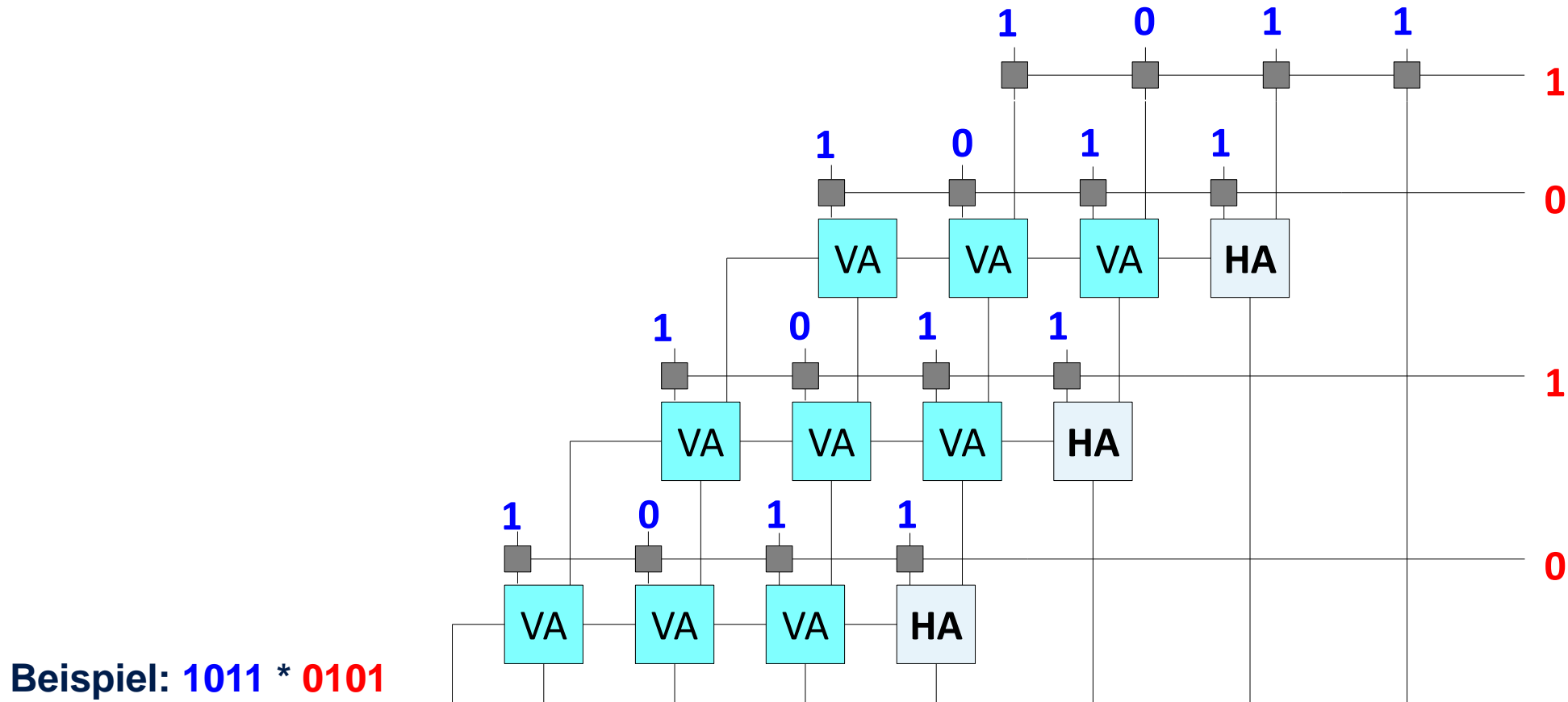


PARALLELER MULTIPLIZIERER #1

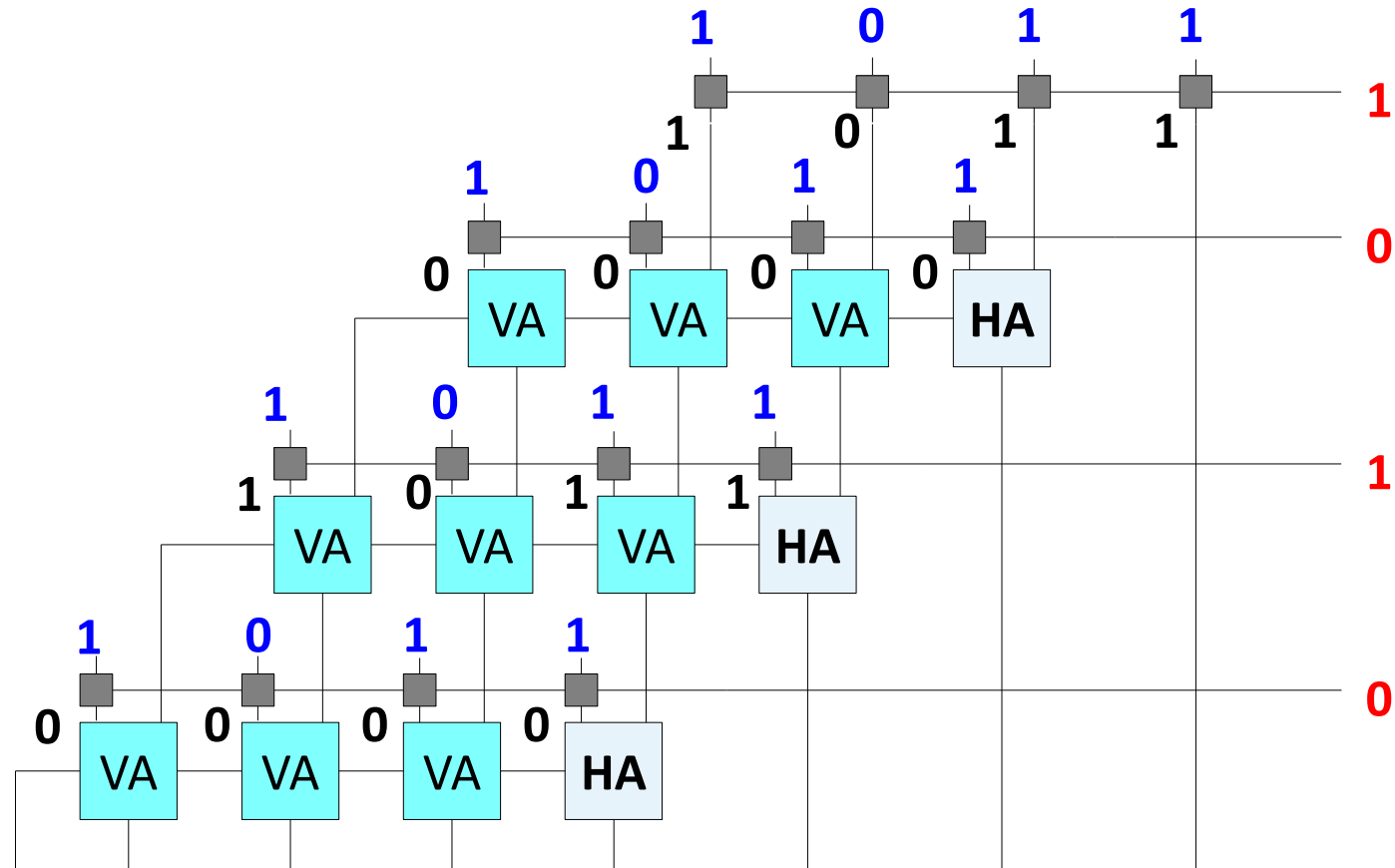
Beispiel: 1011 * 0101



PARALLELER MULTIPLIZIERER #2

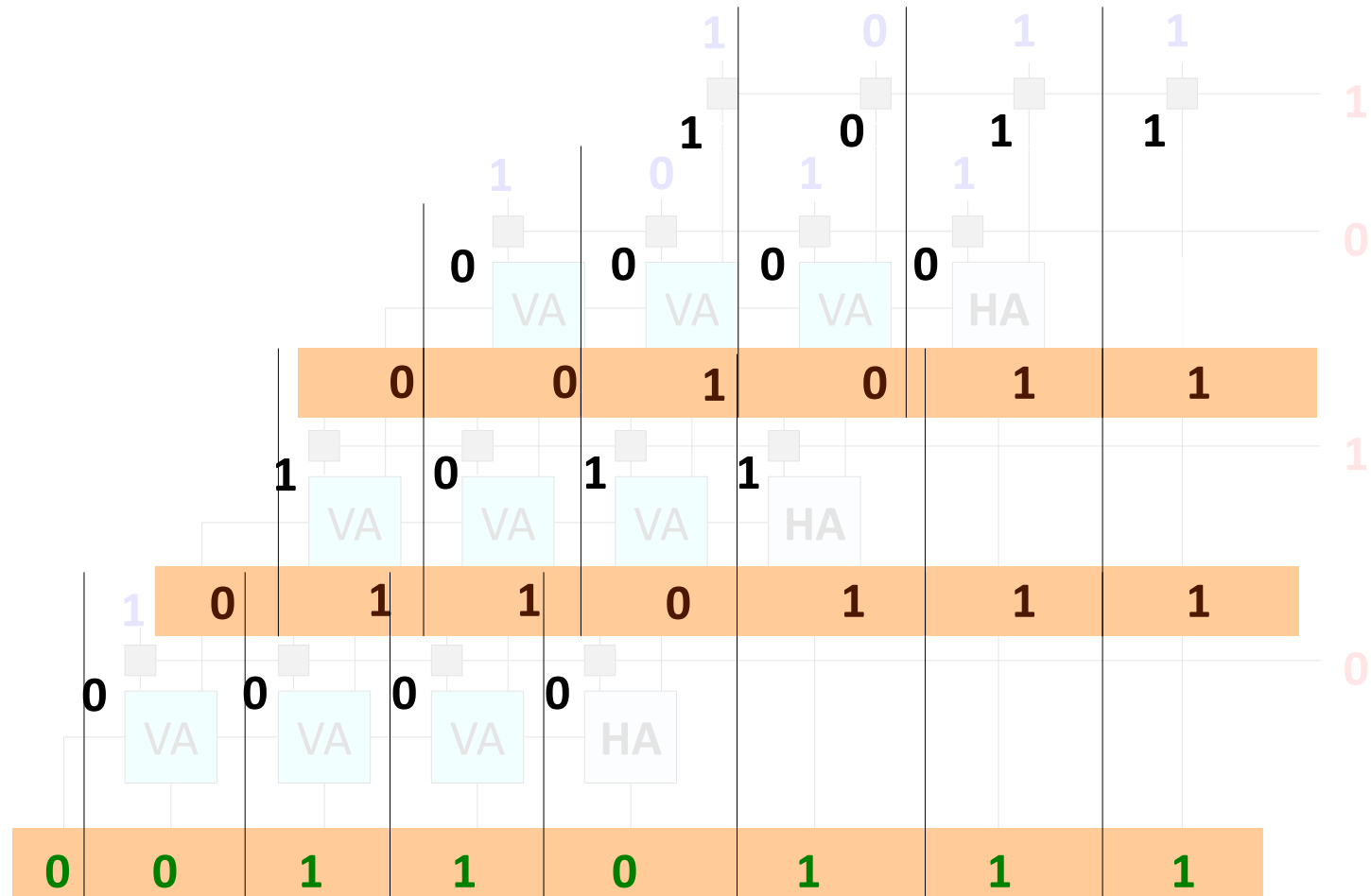


PARALLELER MULTIPLIZIERER #3

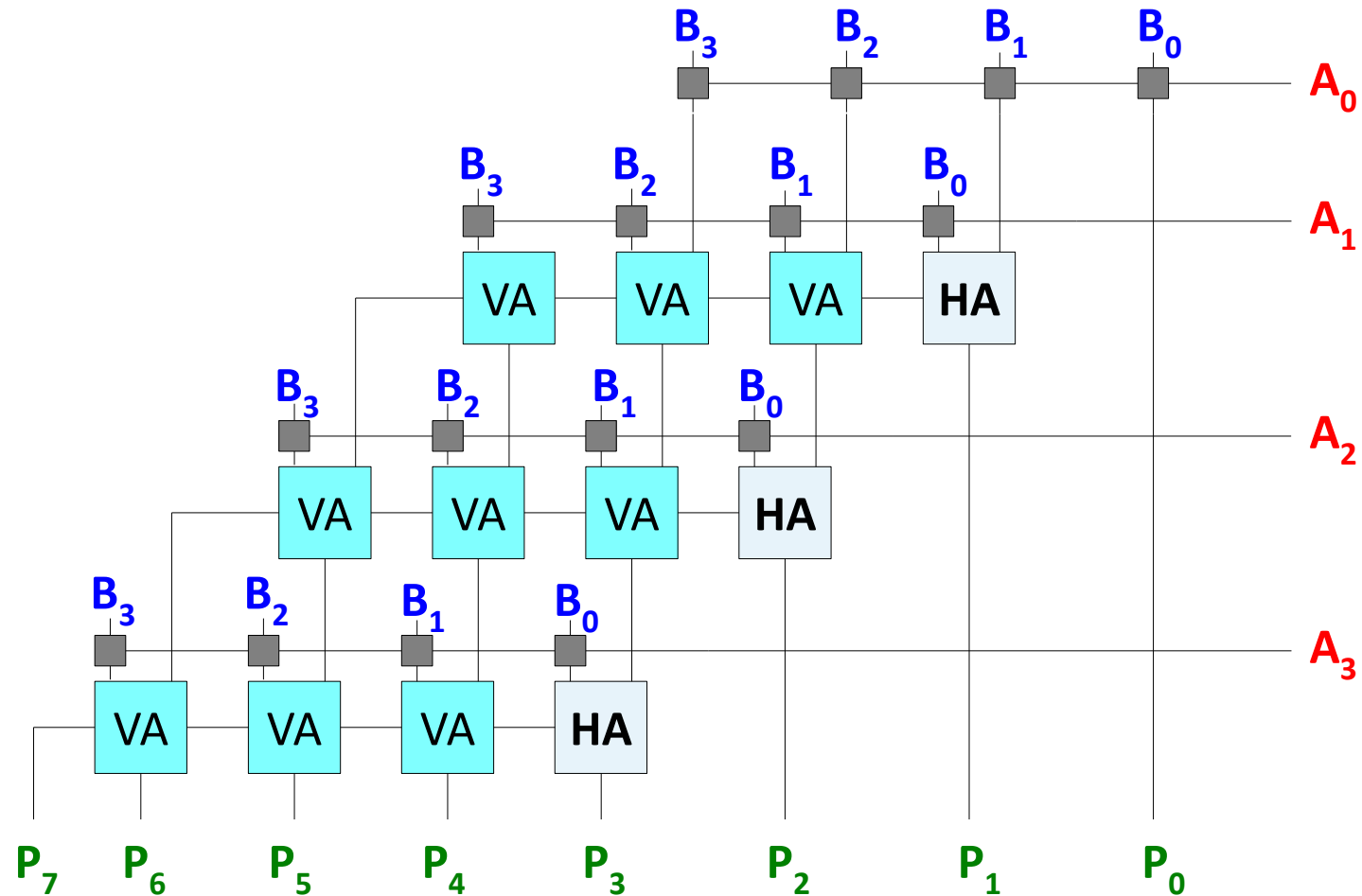


Beispiel: 1011 * 0101

Beispiel: 1011 * 0101

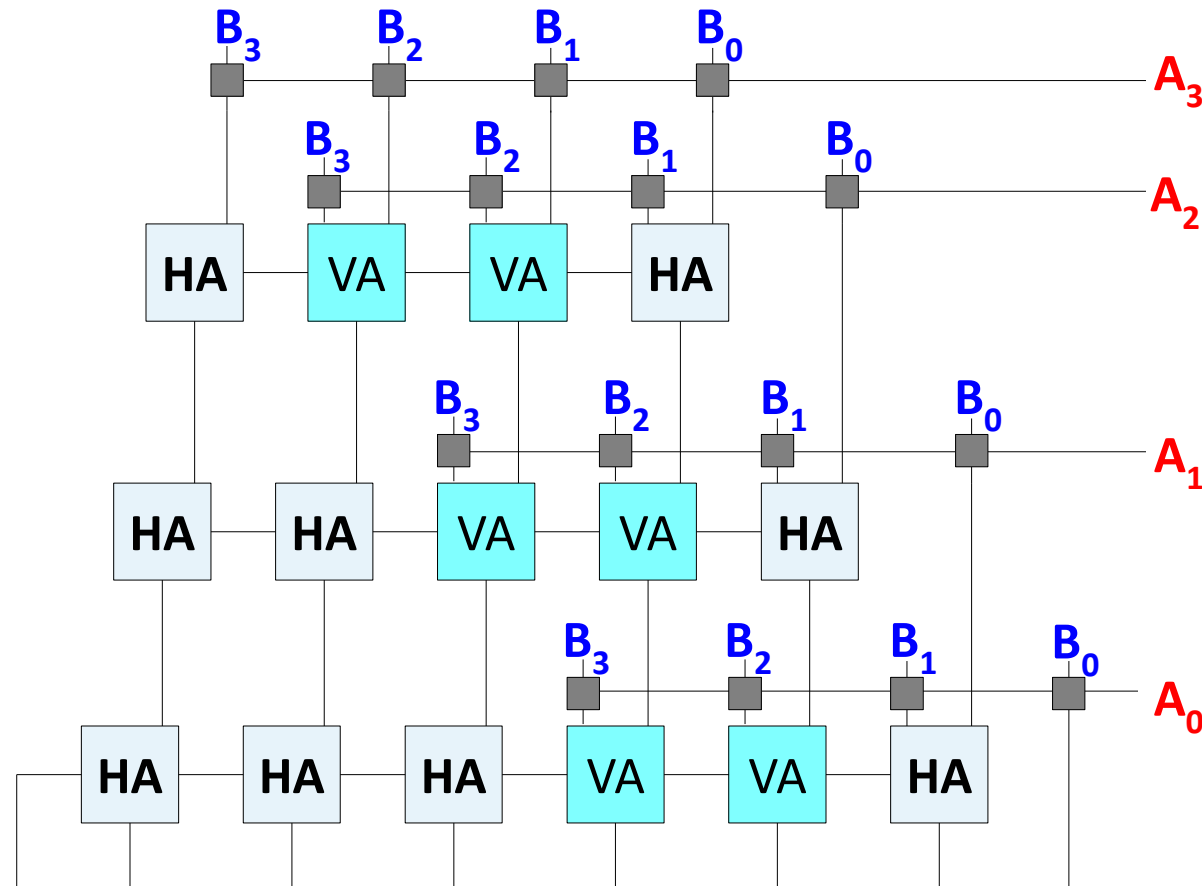


PARALLELER MULTIPLIZIERER #1



MULTIPLIZIERER: ALTERNATIVE STRUKTUR

- Beginn bei Teilprodukt mit MSB von A
- Ungünstiger, da unregelmäßige Struktur



DIVISION



Robert Wille (robert.wille@jku.at)

Sebastian Pointner (sebastian.pointner@jku.at)

Institut für Integrierte Schaltungen

Abteilung für Schaltkreis- und Systementwurf

DIVISION #1

Beispiel:

$$11_{10} / 3_{10} = 3_{10} \text{ Rest } 2_{10}$$

$$\begin{array}{r} 1011 / 11 = 011 \\ - \quad \color{red}{11} \\ \hline 101 \\ - \quad 11 \\ \hline 0101 \\ - \quad 11 \\ \hline 010 \end{array}$$

DIVISION #2

■ Dividend: A
2n-stellige Zahl

■ Divisor: B
n-stellige Zahl

Beides positive Zahlen

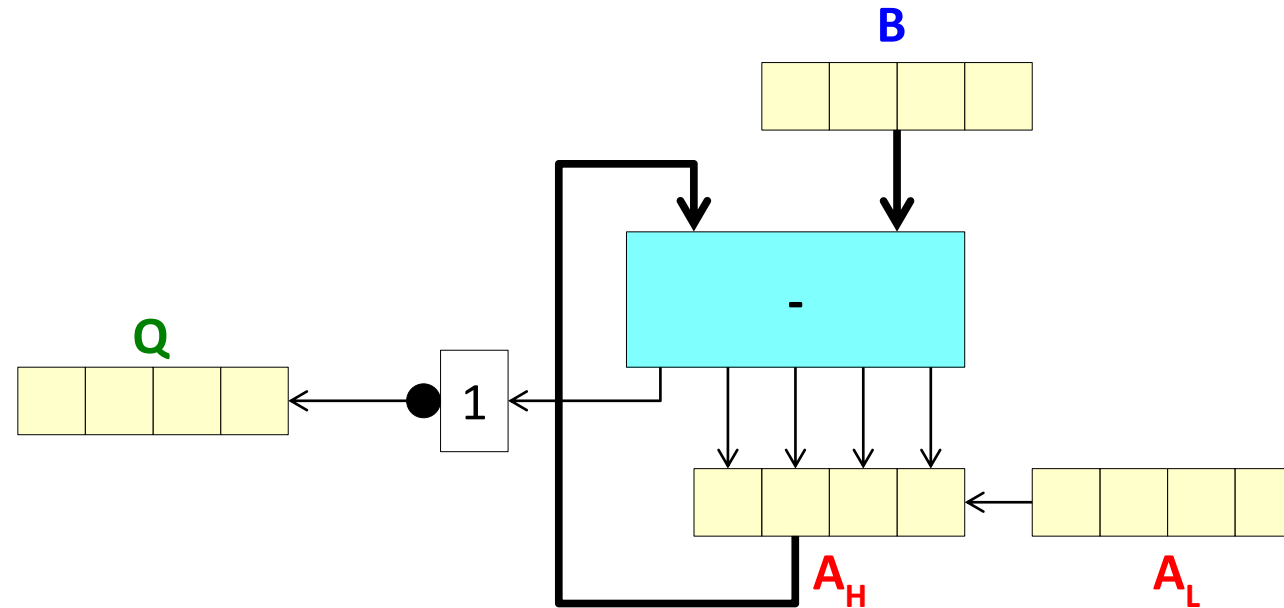
■ Vorgehensweise

1. B von A stellengerecht subtrahieren (also beim MSB von A beginnen)
2. Berechnung der Subtraktion liefert n+1 Stellen (2er-Komplement)
 - a) Ergebnis negativ (Vorzeichen-Bit 1):
A unverändert übernehmen, 0 ins Ergebnis übernehmen
 - b) Ergebnis positiv (Vorzeichen-Bit 0):
Betroffene Stellen von A ändern, 1 ins Ergebnis übernehmen
3. A um eine Stelle nach links schieben
4. Bei 1. fortfahren, bis alle Stellen behandelt sind

DIVIDIERER

Schaltung

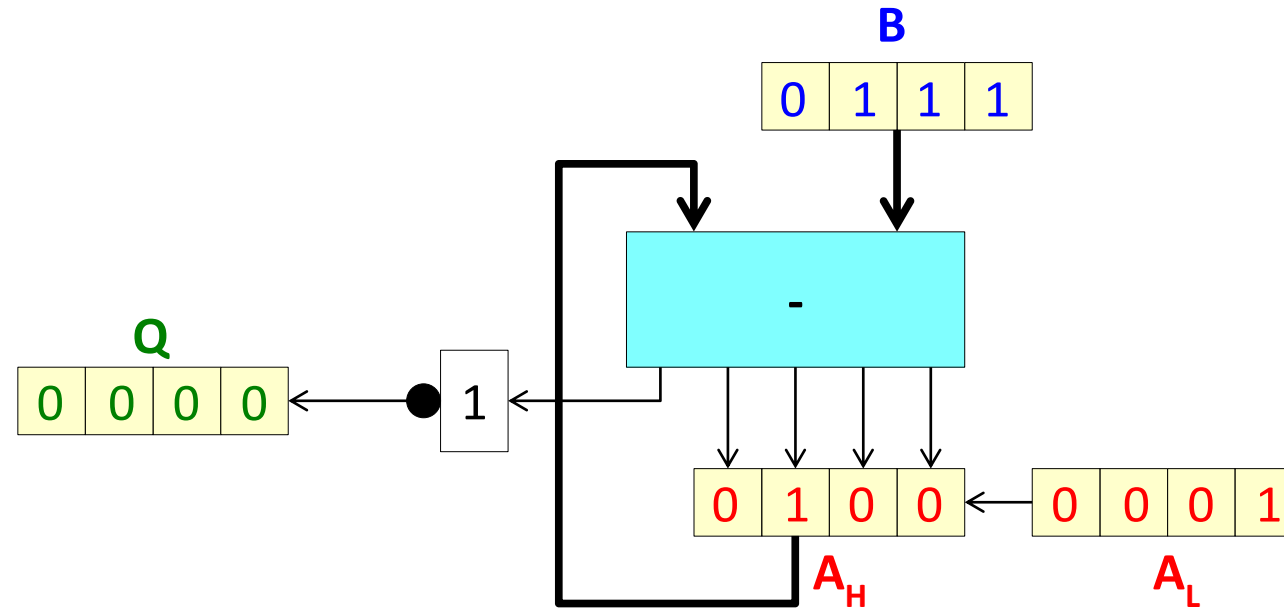
- A_H übernimmt nur das Ergebnis der Subtraktion wenn Vorzeichen positiv
- Rest verbleibt in A_H nach $n+1$ Takten



DIVIDIERER – BEISPIEL 1 #1

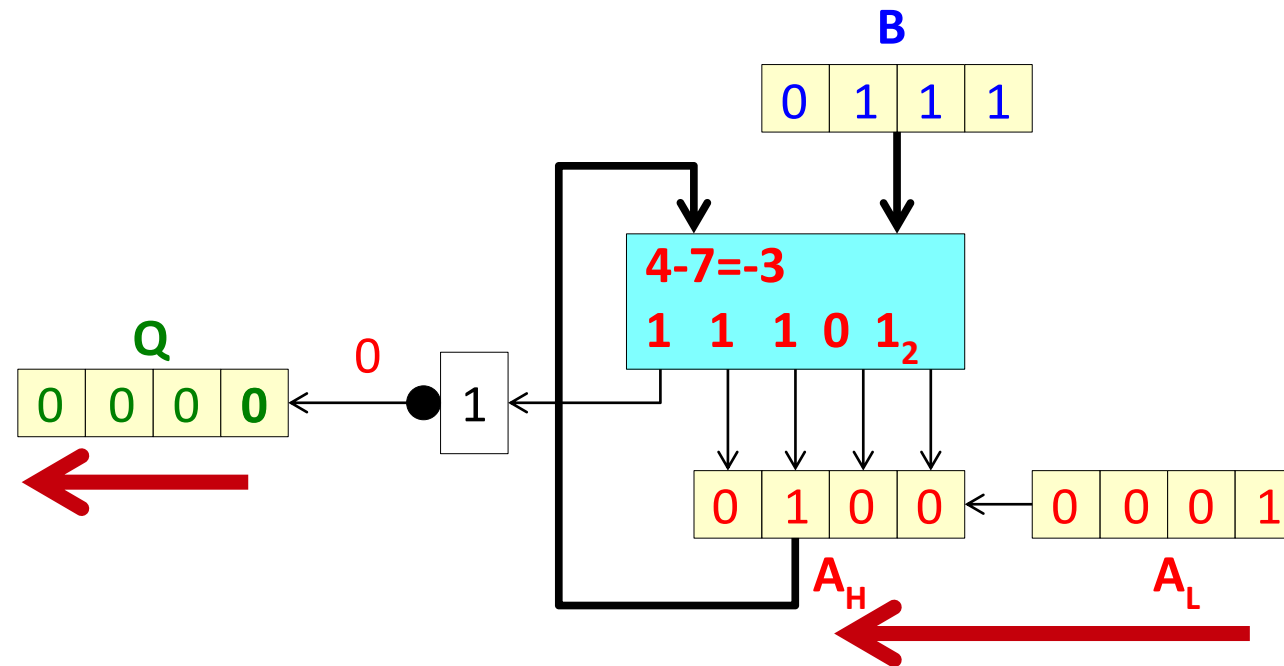
■ $A=65_{10}=0100\ 0001_2$, $B=7_{10}=0111_2$

■ $Q=9_{10}=1001_2$, $R=A_H=2_{10}=0010_2$



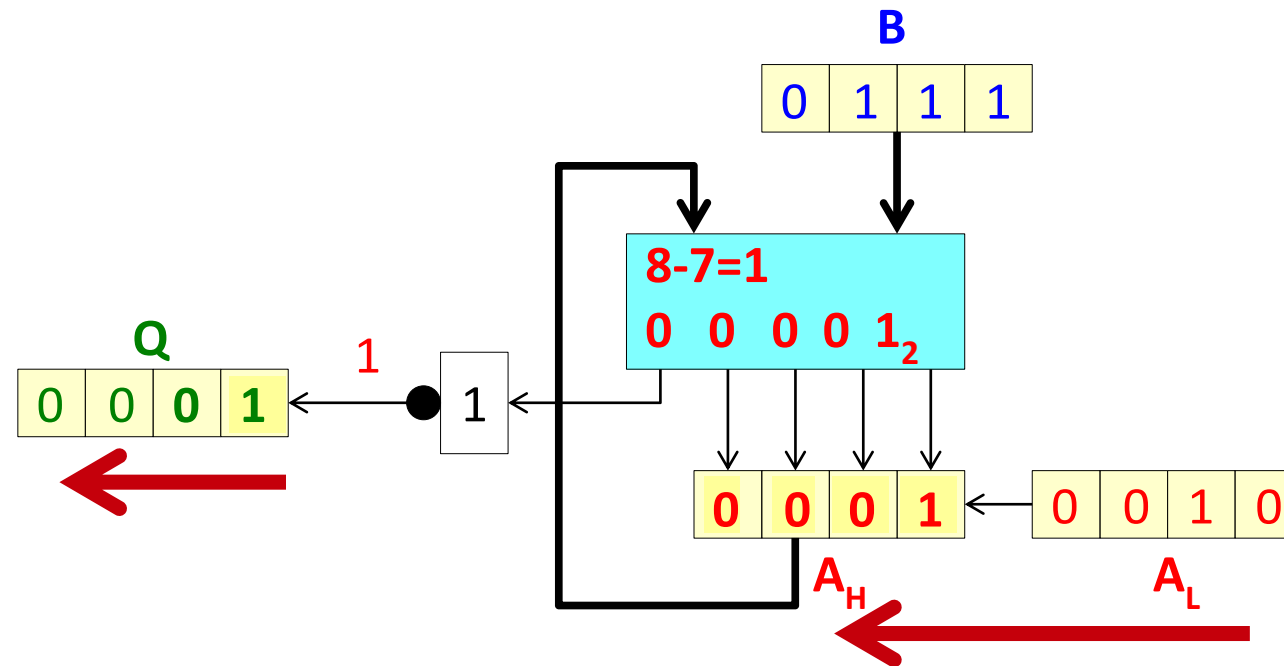
DIVIDIERER – BEISPIEL 1 #2

- 1. Stelle berechnen



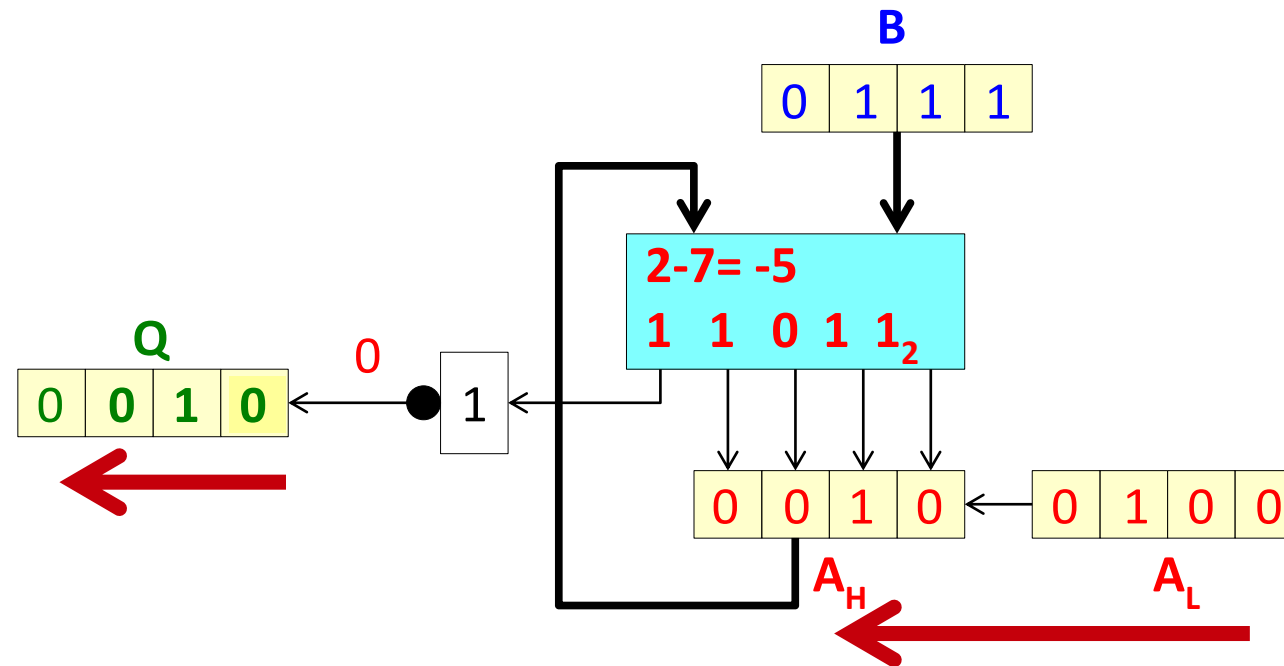
DIVIDIERER – BEISPIEL 1 #3

■ 2. Stelle berechnen



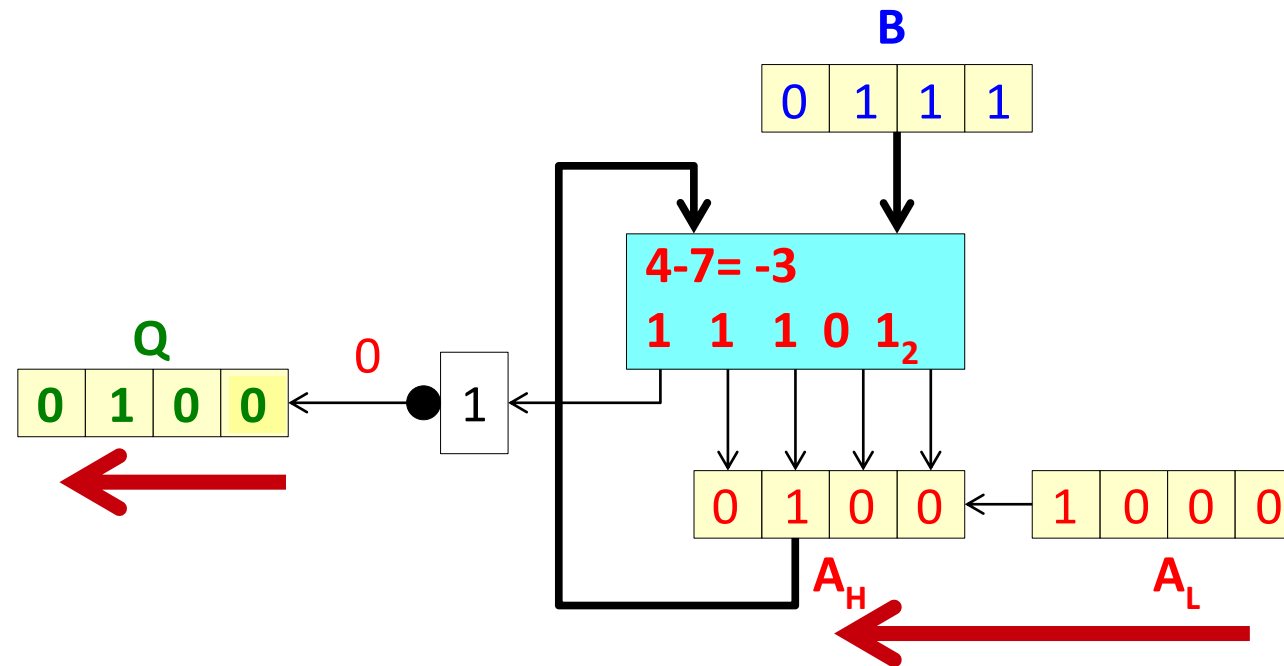
DIVIDIERER – BEISPIEL 1 #4

■ 3. Stelle berechnen



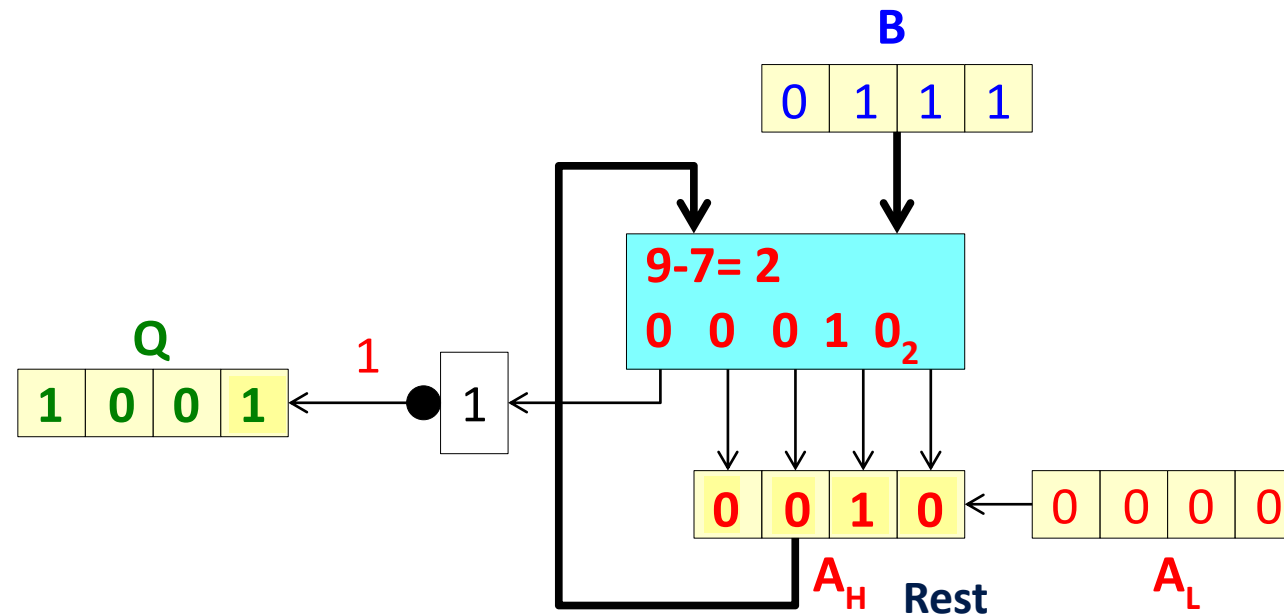
DIVIDIERER – BEISPIEL 1 #5

■ 4. Stelle berechnen



DIVIDIERER – BEISPIEL 1 #6

- „5.“ Stelle berechnen

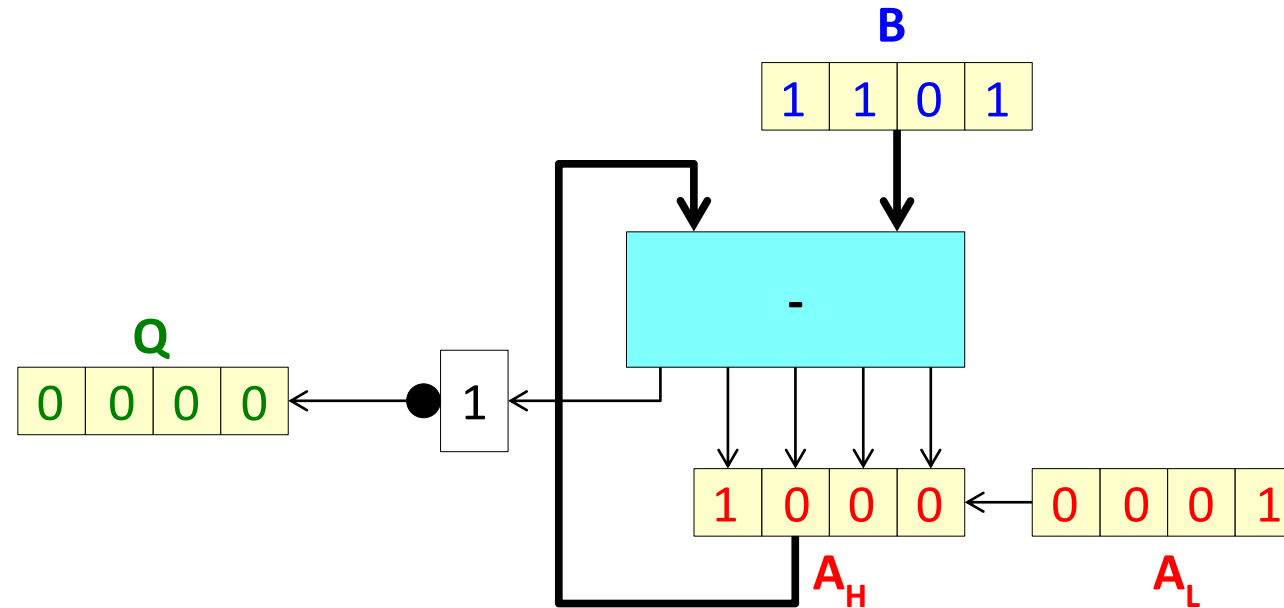


DIVIDIERER – BEISPIEL 2 #1

Beispiel 2:

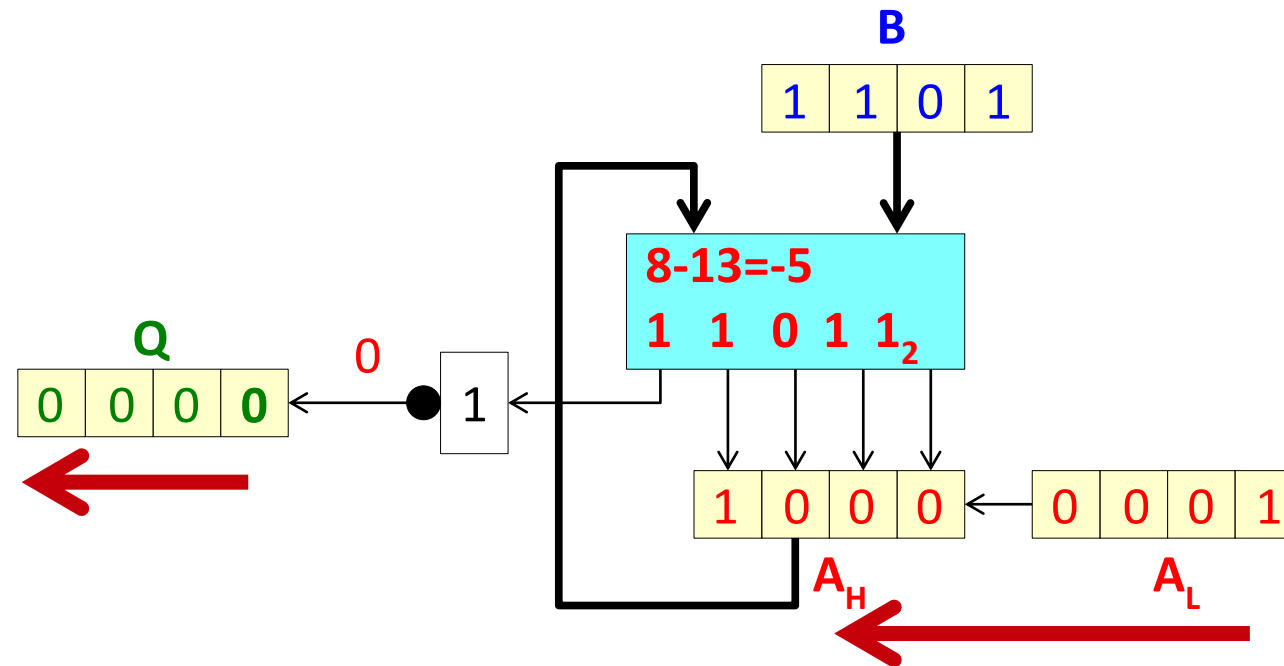
■ $A=129_{10}=1000\ 0001_2$, $B=13_{10}=1101_2$

■ $Q=9_{10}=1001_2$, $R=A_H=12_{10}=1100_2$



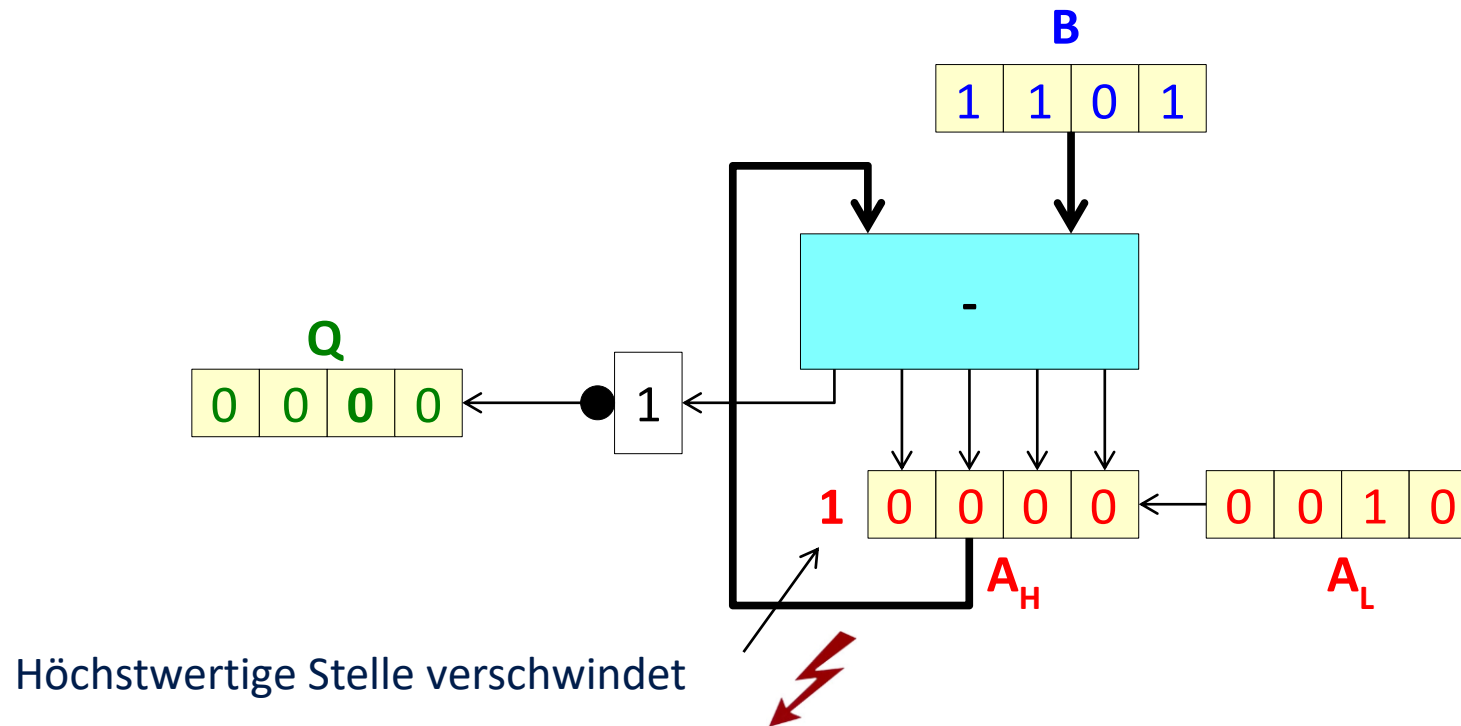
DIVIDIERER – BEISPIEL 2 #2

- 1. Stelle berechnen



DIVIDIERER – BEISPIEL 2 #3

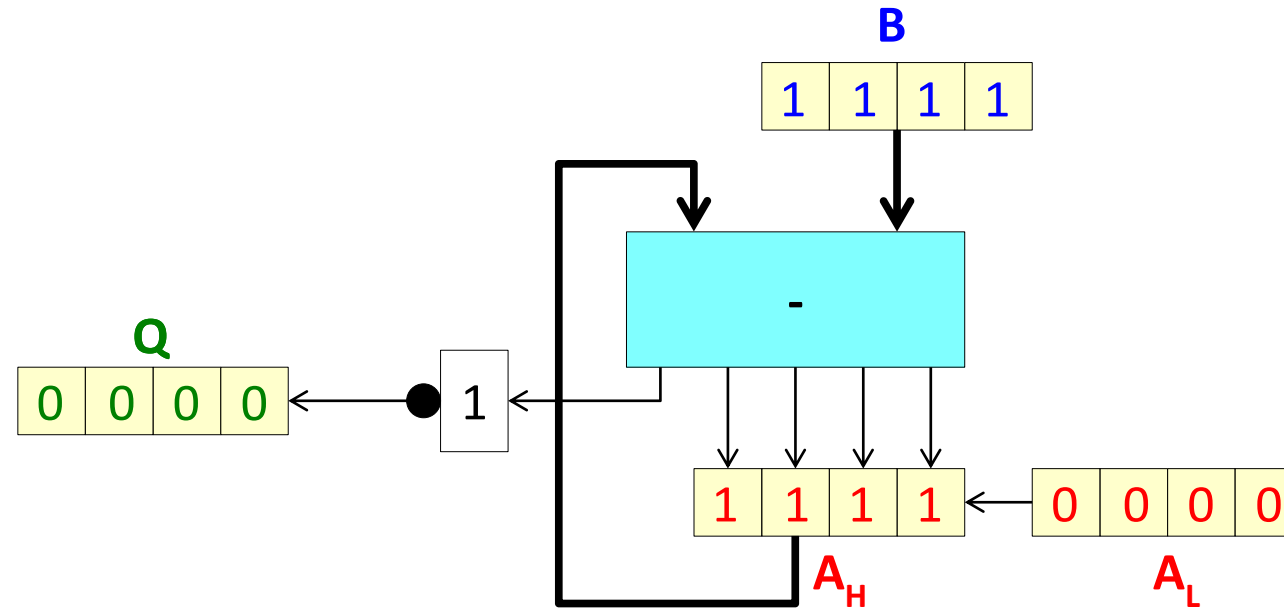
- 1. Stelle berechnen



DIVIDIERER – BEISPIEL 3 #1

■ $A=240_{10}=1111\ 0000_2$, $B=15_{10}=1111_2$

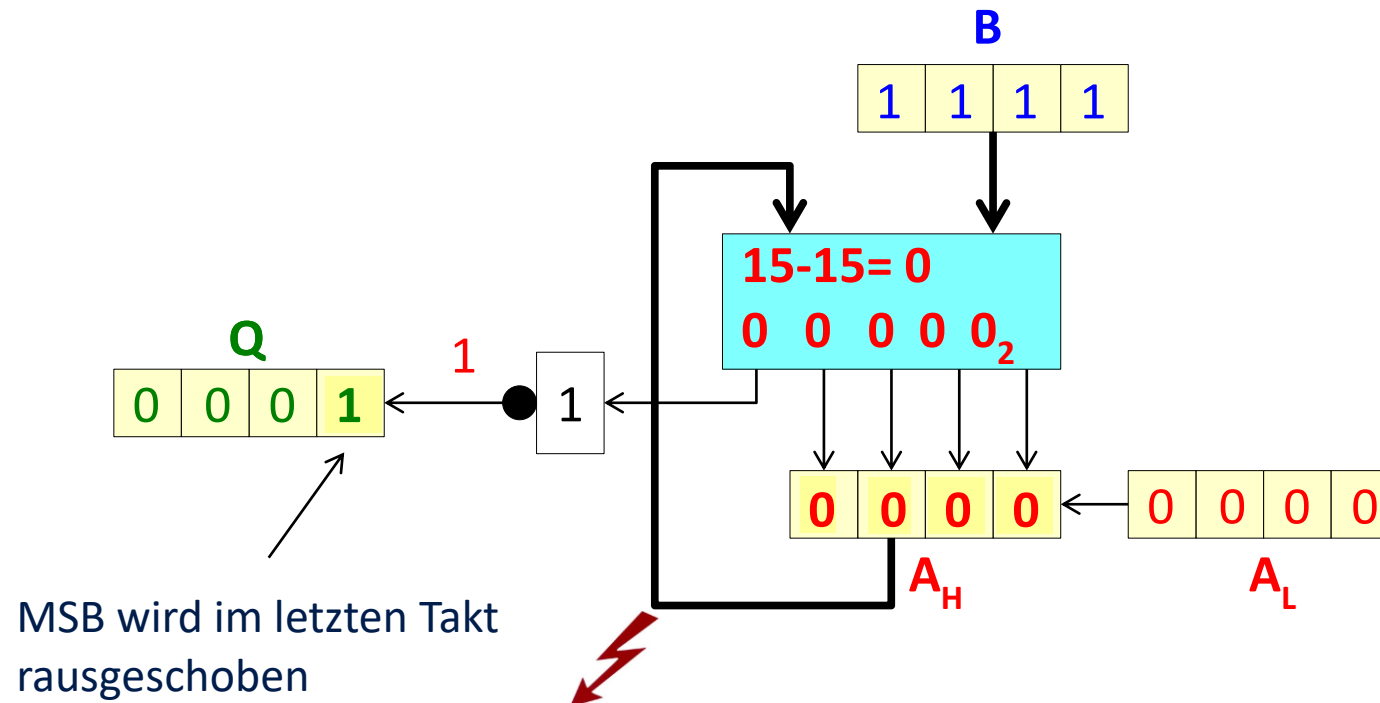
■ $Q=16_{10}=10000_2$, $R=A_H=0_{10}=000_2$



DIVIDIERER – BEISPIEL 3 #2

■ $A=240_{10}=1111\ 0000_2$, $B=15_{10}=1111_2$

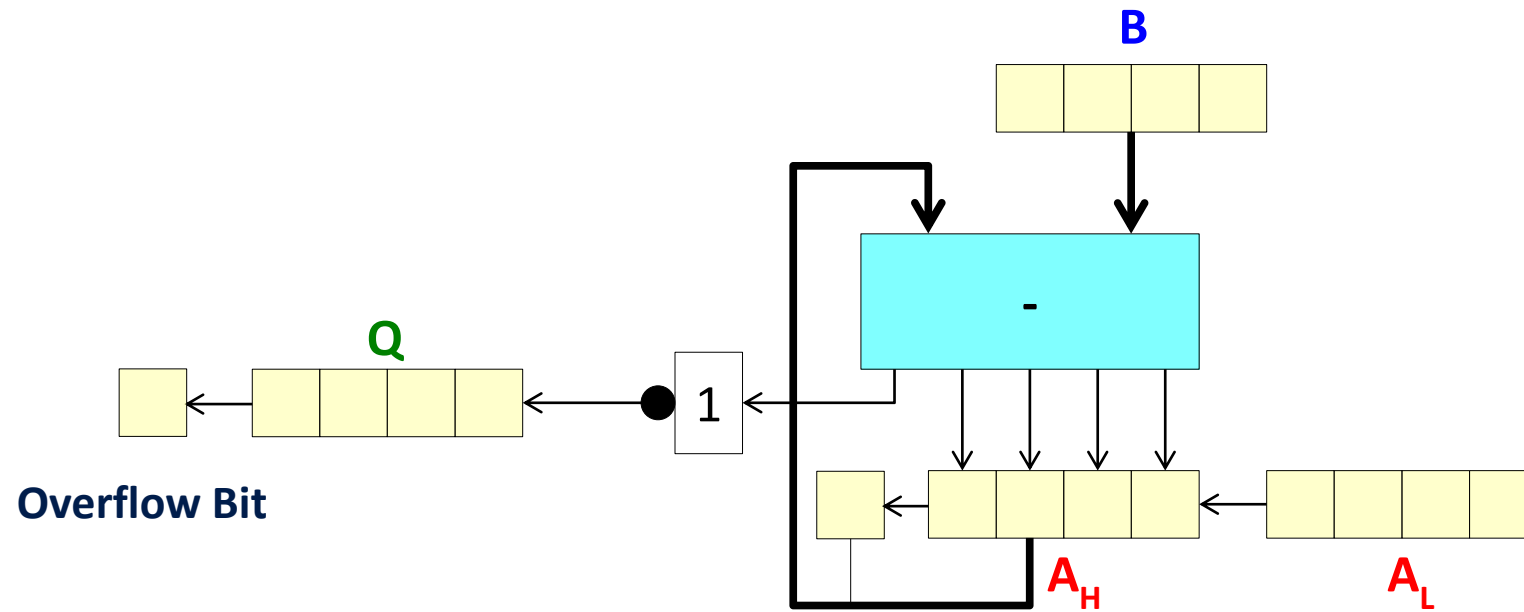
■ $Q=16_{10}=10000_2$, $R=A_H=0_{10}=000_2$



PROBLEME

- MSB des Quotienten muss berücksichtigt werden
 - ☐ signalisiert Überlauf
 - ☐ Tritt auf, wenn $A_H \geq B$
- $n+1$. Position von A_H muss ebenfalls berücksichtigt werden
 - ☐ A_H kann so immer größer als B werden
 - ☐ Zusatz-Bit kann am Ende nicht 1 sein

KORRIGIERTE SCHALTUNG



VARIANTEN

- Subtraktion von **B** durch 2er-Komplement-Addition
 - Zusätzliche Stelle in **B** für Komplement-Darstellung vorsehen
(kann wegfallen, da immer 1)
 - Zusätzliche Stelle in **A** für Vorzeichen nicht erforderlich
(da immer positiv)
- Variante ohne „Rückstellen des Restes möglich“
- Voll parallele Version der Division möglich
(eigener Subtrahierer für jeden Zeitschritt)