

Übungsblatt 9

Rechnerarchitektur im SoSe 2020

Zu den Modulen L

Besprechung: Besprechung der Übungsaufgaben in den Übungsgruppen vom 29. Juni – 3. Juli 2020

Aufgabe H19: Addiernetze in arithmetisch-logischen Einheiten

(11 Pkt.)

- Nehmen Sie einen **Carry-Look-Ahead-Addierer** mit einer Größe der Bit-Gruppen von $g = 3$ an. Leiten Sie den logischen Ausdruck her, mit dem der ausgehende Übertrag U_{out} bereits vor Abarbeitung des Addiernetzes bestimmt werden kann. Bezeichnen Sie dabei die beiden eingehenden Binärzahlen als $x_2x_1x_0$ und $y_2y_1y_0$ und den eingehenden Übertrag als U_{in} .
- Zeichnen Sie das Schaltnetz für einen Carry-Look-Ahead-Addierer für eine Größe von Bit-Gruppen von $g = 3$. Vorkommende Volladdierer können dabei durch ihr entsprechendes Schaltsymbol dargestellt werden. Hierbei können Sie annehmen, dass AND-Gatter und OR-Gatter zur Verfügung stehen, die mehr als zwei Eingaben gleichzeitig verarbeiten können. Achten Sie darauf, die Verbindung von zwei Leitungen explizit zu kennzeichnen.

if 0 0 \Rightarrow there is no way to get overflow

(a)
$$x_2x_1x_0$$

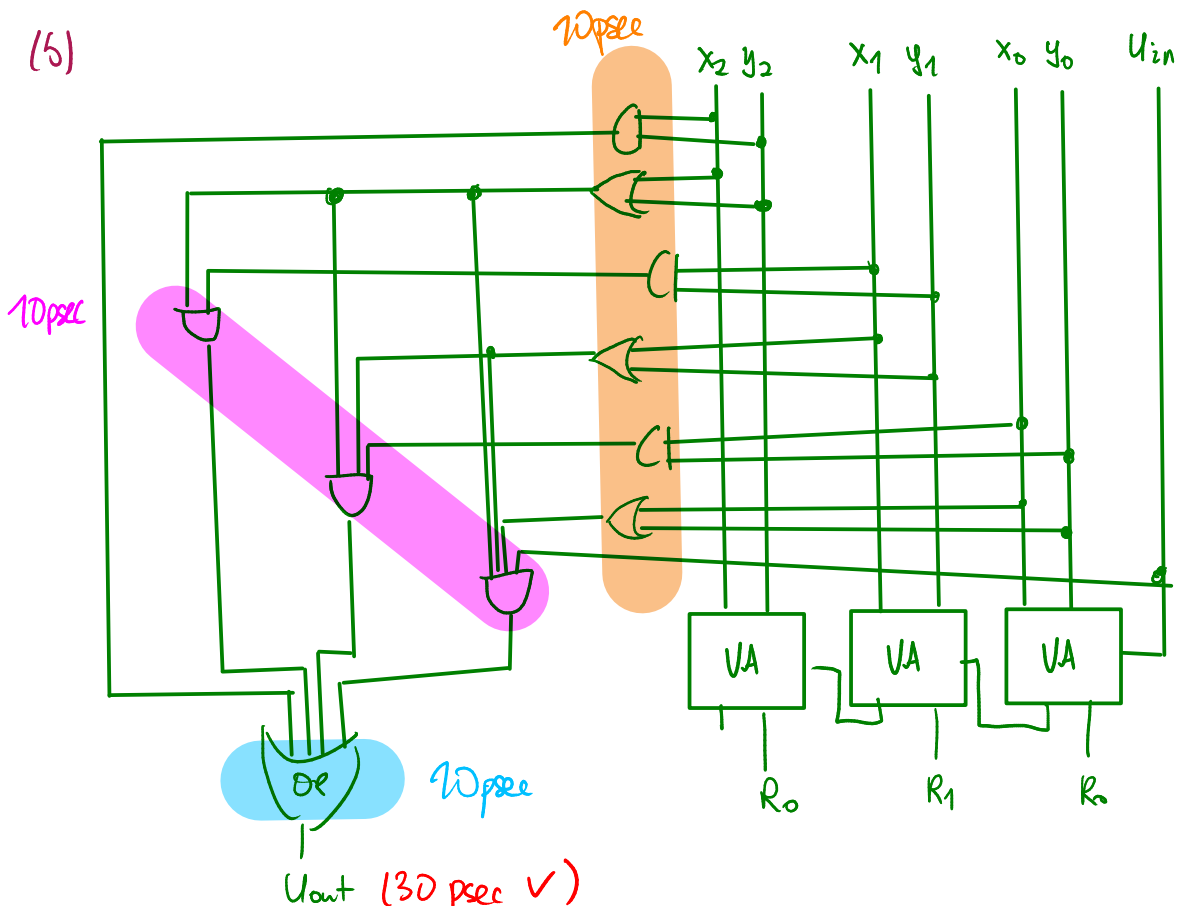
$$+ y_2y_1y_0$$

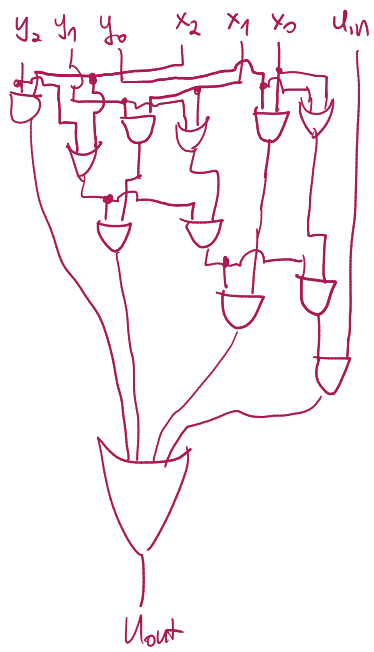
$$U_{out} = x_2y_2$$

$$+ (x_2 + y_2)x_1y_1$$

$$+ (x_2 + y_2)(x_1 + y_1)x_0y_0$$

$$+ (x_2 + y_2)(x_1 + y_1)(x_0 + y_0)U_{in} \Rightarrow \text{Important.}$$





[VA] [VA] [VA]

- c. Gehen Sie nun von der Addition zweier Dualzahlen der Länge 6-Bit aus. Berechnen Sie die Ausführungsdauer der Addition für Carry-Look-Ahead-Addierer mit einer Größe der Bit-Gruppen von $g = 3$, d.h. es werden zwei Carry-Look-Ahead-Addierer aus den vorhergehenden Aufgabenteilen hintereinander geschaltet. Berechnen Sie zudem die Ausführungsdauer für ein angenommenes Ripple-Carry-Addiernetz, das zwei 6-stellige Dualzahlen addieren kann. Nehmen Sie hierbei an, dass ein Volladdierer eine Verzögerung von 70 psec, ein AND-Gatter und OR-Gatter jeweils eine Verzögerung von 10 psec verursachen. AND-Gatter und OR-Gatter mit mehr als zwei Eingängen sollen ebenfalls mit einer Verzögerung von 10 psec veranschlagt werden.

$VA = 70 \text{ psec}$
 $AND/OR = 10 \text{ ps}$

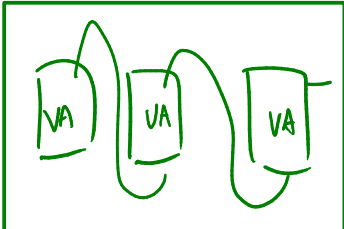
Bei Look-Ahead-Übertrag:

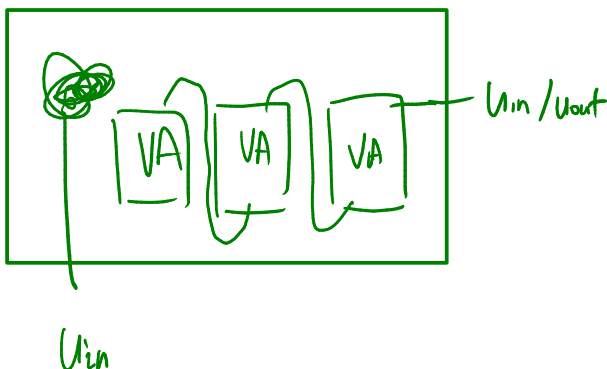
$\left. \begin{array}{l} \text{All } x_i y_i \text{ und } x_i + y_i = 10 \text{ psec} \\ \text{All AND-Gatter} = 10 \text{ psec} \\ \text{OR-Übertrag teigler} = 10 \text{ psec} \end{array} \right\} 30 \text{ psec}$

$A_3 = VA + VA + HA$
 $= 70 + 70 + 30 = 170 \text{ psec}$
 $\tilde{A}_3 = VA + VA + VA$
 $= 3(70) = 210 \text{ psec}$

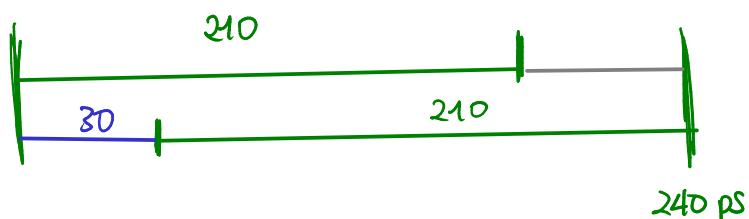
$\tilde{A}_3 \Rightarrow CLH = 210 \text{ ps} + 30 \text{ ps} = 240 \text{ ps}$
 $\tilde{A}_3 \Rightarrow \text{there's } U_{in}$

$RC \Rightarrow 170 + 210 = 380 \text{ psec}$

$\tilde{3A}$

 $RC = 6 \cdot 70 \text{ psec} = 420 \text{ psec}$



$CLA = 3 \cdot 10 \text{ psec} + (U_{out} \text{ fertig})$
 $3 \cdot 70 \text{ psec} (= 210 \text{ psec})$
 $= 240 \text{ psec}$



Aufgabe H20: Schaltung für Successor-Funktion

(8 Pkt.)

In dieser Aufgabe sollen Sie das Schaltnetz für eine Binärschaltung entwerfen, welche die Successor-Funktion für 2-Bit-Zahlen realisiert.

Sei $i \in \{0, \dots, 3\}$ eine Dezimalzahl und $d(i)$ die zweistellige Dualdarstellung von i . Die Successor-Funktion soll die Funktion

$$f_{\text{succ}}(d(i)) = d(i + 1 \bmod 4) \quad \text{mit} \quad f_{\text{succ}} : B^2 \rightarrow B^2$$

realisieren.

nachfolger

Neben zwei Dateneingänge x_0 und x_1 sowie zwei Datenausgänge y_0 und y_1 soll die Schaltung einen Steuereingang s besitzen. Nur wenn gilt $s = 1$ soll die Schaltung den Wert von $f_{\text{succ}}(d(i))$ als Ergebnis an den Datenausgängen liefern. Für $s = 0$ soll die Schaltung die Dateneingänge unverändert an die Datenausgänge weiterleiten.

- Stellen Sie zunächst die Funktionstafel für die gewünschte Schaltung auf.
- Leiten Sie von der Funktionstafel die beiden Schaltfunktionen f_{y_0} und f_{y_1} für die Datenausgänge y_0 und y_1 ab. Minimieren Sie beide Funktion so weit wie möglich.
- Oft ist es billiger eine Schaltung aus komplexeren schon existierenden Bausteinen zusammen zu setzen, anstatt sie von Grund auf neu zu konstruieren.

Gehen Sie davon aus, dass Ihnen aus Kostengründen nur Halbaddierer zur Verfügung stehen und konstruieren Sie die Schaltung mittels Halbaddierer.

(a)

s	x ₁	x ₀	y ₁	y ₀
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

$$\begin{aligned}
 (b) \quad f_{y_1} &= \bar{s}(x_1 \bar{x}_0 + x_1 x_0) + s(\bar{x}_1 x_0 + x_1 \bar{x}_0) \\
 &= \bar{s}x_1 + s\bar{x}_1 x_0 + s x_1 \bar{x}_0 + s x_1 x_0 \\
 &= \bar{s}x_1 + s\bar{x}_1 x_0 + x_1 \bar{x}_0
 \end{aligned}$$

$$\begin{aligned}
 f_{y_0} &= \bar{s}(\bar{x}_1 x_0 + x_1 x_0) + s(\bar{x}_1 \bar{x}_0 + x_1 \bar{x}_0) \\
 &= \bar{s}x_0 + s\bar{x}_0
 \end{aligned}$$

Tip: Highlight the resolved Terms.

(c) HA:

x	y	R	U
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{aligned}
 R &= \bar{x}y + x\bar{y} \\
 U &= xy
 \end{aligned}$$

$$x_1 \bar{s} \bar{x}_0 + s x_0 \bar{x}_1$$

$$\begin{aligned}
 &= (\bar{s} + \bar{x}_0) x_1 \\
 &= \bar{s}x_1 + x_1 \bar{x}_0
 \end{aligned}$$

$$\begin{aligned}
 x_0 &\leftrightarrow x_1 \\
 y_0 &\leftrightarrow y_1
 \end{aligned}$$

Vertauscht!

