



5. Übungsblatt Informatik II

(Abgabe: 30.05.2003)

1. Aufgabe: (Zero-Berechnung)

(5 Punkte)

Da bei dieser Aufgabe eine Äquivalenz zu zeigen ist, muss man Hin- und Rückrichtung beweisen.

- $\Leftarrow$: Zu zeigen: $[a] = [b] \Rightarrow s[n-1:0] = 0^n$.

Beweis:

$$\begin{aligned}
 [a] = [b] & \stackrel{\text{Vor}}{\Rightarrow} [s] = 0 \\
 & \stackrel{\text{Def}}{\Leftrightarrow} -s_n \cdot 2^n + \langle s[n-1:0] \rangle = 0 \\
 & \stackrel{(*)}{\Rightarrow} \langle s[n-1:0] \rangle = 0 \\
 & \Leftrightarrow s[n-1:0] = 0^n
 \end{aligned}$$

Anmerkung:

Die Folgerung $(*)$ gilt, weil $-s_n \cdot 2^n = \langle s[n-1:0] \rangle = 0$ die einzig mögliche Lösung obiger Gleichung ist. Die theoretische andere Lösung, $\langle s[n-1:0] \rangle = s_n \cdot 2^n$, kann nicht auftreten, da $s_n \cdot 2^n$ für $s_n = 1$ außerhalb des Wertebereichs von $\langle s[n-1:0] \rangle$ liegt. (Im Falle von $s_n = 0$ sind wir wieder bei der zuerst genannten Lösung.)

- $\Rightarrow$: Zu zeigen: $s[n-1:0] = 0^n \Rightarrow [a] = [b]$.

Beweis:

$$\begin{aligned}
 s[n-1:0] = 0^n & \Leftrightarrow \langle s[n-1:0] \rangle = 0 \\
 & \stackrel{\text{Def}}{\Rightarrow} [s] = -s_n \cdot 2^n \\
 & \stackrel{\text{Vor}}{\Leftrightarrow} [a] - [b] = -s_n \cdot 2^n
 \end{aligned}$$

Nun muss zwischen zwei Fällen unterschieden werden:

- (a) $s_n = 0 \Rightarrow [a] - [b] = 0 \Rightarrow [a] = [b]$
- (b) $s_n = 1 \Rightarrow [a] - [b] = -2^n$

Der erste Fall ist genau die Folgerung, die es zu beweisen gilt. Um den Beweis zu vollenden, muss gezeigt werden, dass der zweite Fall niemals auf treten kann.

Es gilt offensichtlich: Die Differenz $[a] - [b]$ ist minimal, wenn $[a]$ den kleinstmöglichen und $[b]$ den größtmöglichen Wert annimmt. (Zur Erinnerung: $[a], [b] \in \{-2^{n-1}, \dots, 2^{n-1}-1\}$.)

Wir wählen also $[a]_{\min} = -2^{n-1}$ und $[b]_{\max} = 2^{n-1} - 1$. Dann gilt für alle $[a], [b]$ mit $[a] \geq [a]_{\min}$ und $[b] \leq [b]_{\max}$:

$$\begin{aligned}
 [a] - [b] & \geq [a]_{\min} - [b]_{\max} \\
 & = -2^{n-1} - (2^{n-1} - 1) \\
 & = -2^n + 1 \\
 & > -2^n
 \end{aligned}$$

Damit ist gezeigt, dass Fall 2. niemals auftreten kann. Fall 1. ist damit der einzig mögliche und der Beweis ist erbracht.



5. Übungsblatt Informatik II

(Abgabe: 30.05.2003)

2. Aufgabe: (Decoder)

(4 Punkte)

Seien $y'[2^n - 1 : 0]$ die Ausgänge von Dec_n . Dann gilt nach Konstruktion des Schaltkreises:

$$y'_j = \begin{cases} y_j \wedge \overline{x_{n-1}} & , j \in \{0, \dots, 2^{n-1} - 1\} \\ y_{j-2^{n-1}} \wedge x_{n-1} & , j \in \{2^{n-1}, \dots, 2^n - 1\} \end{cases}$$

Mit $y[2^{n-1} - 1 : 0]$ sind dabei die Ausgänge von Dec_{n-1} bezeichnet.

Es gilt nach Aufgabenstellung: $x_{n-1} = 1$. Damit:

$$\langle x' \rangle := \langle x_{n-1}, x \rangle = 2^{n-1} + \langle x \rangle$$

Sei nun $\langle x \rangle = i$, $i \in \{0, \dots, 2^{n-1} - 1\}$ beliebig. Dann gilt:

$$\langle x' \rangle = 2^{n-1} + i =: i'$$

Zu zeigen ist: $y'_{i'} = 1 \Leftrightarrow \langle x' \rangle = i'$.

Beweis:

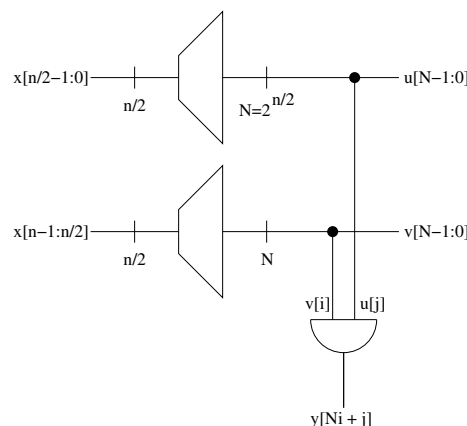
$$\begin{aligned} y'_{i'} = 1 & \Leftrightarrow y_{i'-2^{n-1}} = 1 \\ & \stackrel{(*)}{\Leftrightarrow} y_i = 1 \\ & \stackrel{\text{Vor}}{\Leftrightarrow} \langle x \rangle = i \\ & \Leftrightarrow \langle x \rangle + 2^{n-1} = i + 2^{n-1} \\ & \Leftrightarrow \langle x' \rangle = i + 2^{n-1} \\ & \Leftrightarrow \langle x' \rangle = i' \end{aligned}$$

Anmerkung: Die Äquivalenz $(*)$ gilt, weil $i' - 2^{n-1} \geq 2^{n-1}$ (nach Definition von y'_j) und weil $i' - 2^{n-1} = i$.

3. Aufgabe: (Schneller Decoder)

(4 + 5 + 7 Punkte)

(a) Der Schaltkreis ist rekursiv wie folgt aufgebaut:



Der Schaltkreis besteht aus zwei parallelen $FDec_{n/2}$, und 2^n parallele UND-Gattern. Somit hat er die Tiefe $D(FDec_n) = D(FDec_{n/2}) + 1$ und den Kosten $C(FDec_n) = 2 \cdot C(FDec_{n/2}) + 2^n$



5. Übungsblatt Informatik II

(Abgabe: 30.05.2003)

(b) Induktion über $k = \log_2(n)$:

IA: $k = 0$ Korrektheit schon bewiesen (da $FDec_1 = Dec_1$).

IV: Sei $y_i = 1 \Leftrightarrow \langle x \rangle = i$ für $FDec_{k-1}$

IS: $k - 1 \rightarrow k$

$$y[N \cdot i + j] = 1 \Leftrightarrow v[i] = 1 \wedge u[j] = 1$$

$$\stackrel{IV}{\Leftrightarrow} \langle x[n-1 : n/2] \rangle = i \wedge \langle x[n/2-1 : 0] \rangle = j$$

$$\Leftrightarrow \langle x[n-1 : n/2]x[n/2-1 : 0] \rangle = N \cdot i + j$$

(c) $C(n) = 2 \cdot C(n/2) + 2^n$

$$C(n) = 2 \cdot (2 \cdot C(n/4) + 2^{n/2}) + 2^n$$

$$C(n) = 2 \cdot (2 \cdot (2 \cdot C(n/8) + 2^{n/4}) + 2^{n/2}) + 2^n$$

$$C(n) = 2 \cdot (2 \cdot (2 \cdot (2 \cdot C(n/16) + 2^{n/8}) + 2^{n/4}) + 2^{n/2}) + 2^n$$

$$C(n) = 2 \cdot 2 \cdot 2 \cdot 2 \cdot C(n/16) + 2 \cdot 2 \cdot 2 \cdot 2^{n/8} + 2 \cdot 2 \cdot 2^{n/4} + 2 \cdot 2^{n/2} + 2^n$$

$$C(n) = 2^4 \cdot C(n/2^4) + 2^3 \cdot 2^{n/2^3} + 2^2 \cdot 2^{n/2^2} + 2^1 \cdot 2^{n/2^1} + 2^0 \cdot 2^{n/2^0}$$

Jetzt können wir eine Zahl k für unsere Rekursionsschritte wählen. Und dann die Gleichung mit dieser Zahl umformen:

$$C(n) = 2^k \cdot C(n/2^k) + 2^{k-1} \cdot 2^{n/2^{k-1}} + 2^{k-1} \cdot 2^{n/2^{k-1}} + 2^{k-1} \cdot 2^{n/2^{k-1}} + 2^{k-1} \cdot 2^{n/2^{k-1}}$$

Wenn wir hier annehmen, dass $k = \log_2 n$ ($n = 2^k$) ist, dann erhalten wir die Gleichung:

$$C(n) = 2^k \cdot C(1) + \sum_{i=0}^{k-1} 2^i \cdot 2^{n/2^i}$$

$$C(n) = n + \sum_{i=0}^{\log_2 n - 1} 2^i \cdot 2^{n/2^i}$$

Korrektheitsbeweis per Induktion über k :

IA: $k = 0 \Rightarrow$

$$C(1) = 2^0 + \sum_{i=0}^{-1} 2^i \cdot 2^{n/2^i} = 1$$

IV: Sei $C(2^k) = 2^k + \sum_{i=0}^{k-1} 2^i \cdot 2^{n/2^i}$

IS: $k - 1 \rightarrow k$

$$\begin{aligned} C(n) &\stackrel{nD}{=} 2 \cdot C(n/2) + 2^n \\ &\stackrel{IV}{=} 2 \cdot (n/2) + \sum_{i=0}^{k-1} 2^i \cdot 2^{n/2^i} + 2^n \\ &= n + \sum_{i=0}^{k-2} 2^{i+1} \cdot 2^{n/2^{i+1}} + 2^n \\ &= n + \sum_{i=1}^{k-1} 2^i \cdot 2^{n/2^i} + 2^0 \cdot 2^{n/2^0} \\ &= n + \sum_{i=0}^{k-1} 2^i \cdot 2^{n/2^i} \end{aligned}$$

4. Aufgabe: (Incrementer)

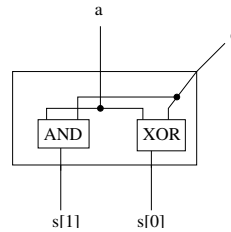
(2 + 3 + 5 Punkte)

(a) Wie in der Aufgabenstellung bereits erwähnt kann ein 1-bit Incrementer durch einen Halbaddierer dargestellt werden.

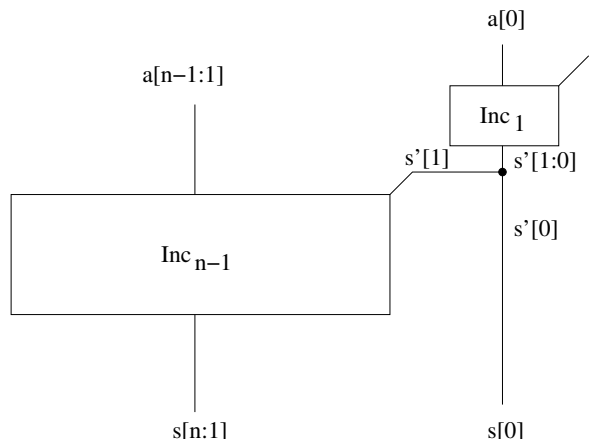


5. Übungsblatt Informatik II

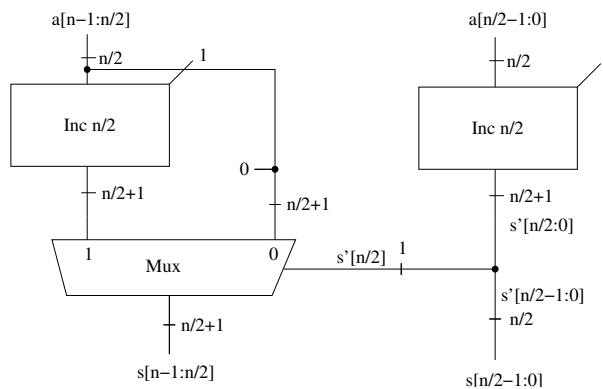
(Abgabe: 30.05.2003)



(b) Das folgende Schaltbild zeigt einen n-bit-Carry-Chain-Incrementer



(c) Das folgende Schaltbild zeigt einen n-bit-Conditional-Sum-Incrementer



Beachte für die Kostenbestimmung: Das oberste Bit des $(n/2+1)$ -Muxes ist ein Mux mit einer konstanten 0 und geht deshalb mit einem einfachen UND-Gatter, daher die Kosten des Muxes $(3(n/2) + 1) + 1$.

5. Aufgabe: (ALU)

(10 Punkte)

Es gibt verschiedene korrekte Lösungen, da an manchen Stellen alternative Implementierungen an der Tafel standen. Alle sind richtig, solange vernünftig begründet wird.

D(ALU)

$$= \max\{D(\text{Addierer}), D(\text{Subtrahierer}), D(\text{AND}), D(\text{OR}), D(\text{XOR}), D(\text{MUX}), D(\text{Test})\}$$

$$= D(\text{Test}) \quad (\text{da } D(\text{Test}) > D(\text{Subtrahierer}) > D(\text{Addierer}) > 3 > 1)$$

$$= \max\{D(\text{b-Eingang}), D(\text{neg-Signal})\} + D(\text{Addierer}) + D(\text{Testbox})$$

$$= D(\text{b-Eingang}) + D(\text{Addierer}) + D(\text{Testbox}) \quad (\text{da } D(\text{b-Eingang}) = D(\text{neg-Signal}) + 1)$$

$$= (4 \text{ bzw. } 3 \text{ bzw. } 2) + 18 + \max\{D(\text{neg}), D(\text{zero}), D(\text{pos})\} + (3 \text{ bzw. } 2)$$



5. Übungsblatt Informatik II

(Abgabe: 30.05.2003)

$$\begin{aligned}
 &= (4 \text{ bzw. } 3 \text{ bzw. } 2) + 18 + D(\text{pos}) + (3 \text{ bzw. } 2) \quad (\text{da } D(\text{pos}) = \max\{D(\text{neg}), D(\text{zero})\} + 1) \\
 &= (4 \text{ bzw. } 3 \text{ bzw. } 2) + 18 + \max\{3, (6 \text{ bzw. } 0)\} + 1 + (3 \text{ bzw. } 2) \\
 &= (4 \text{ bzw. } 3 \text{ bzw. } 2) + 19 + (6 \text{ bzw. } 3) + (3 \text{ bzw. } 2)
 \end{aligned}$$

6. * Aufgabe: (BCD-Addierer)

(7 + 3 Punkte *)

(a) Für $n=1$ gilt folgendes:

$$a, b \in \{0, 1\}^4, c \in \{0, 1\}, s \in \{0, 1\}^8$$

die Zahlen a, b (in Dezimaldarstellung) liegen im Bereich $\{0, \dots, 9\}$ und die Zahl s im Bereich $\{0, \dots, 19\} \Rightarrow s_7 = s_6 = s_5 = 0$

$$\text{Sei } \langle t \rangle = \langle a \rangle + \langle b \rangle$$

Unterscheide 2 Fälle:

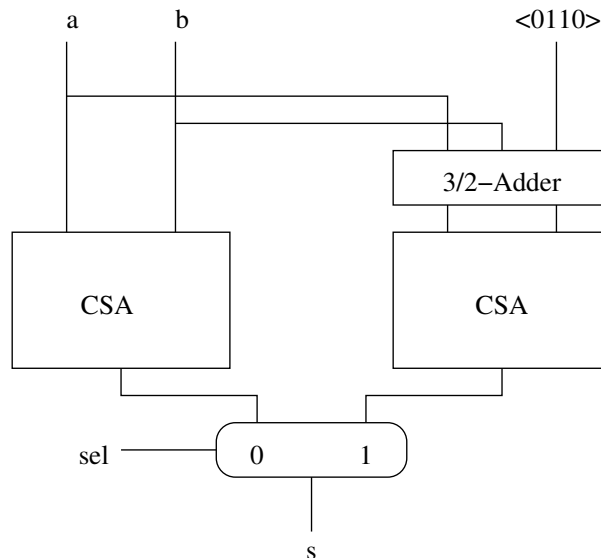
$$1. \langle t \rangle \leq 9 \Rightarrow s[4:0] = t[4:0]$$

$$2. \langle t \rangle > 9 \Rightarrow s_4 = 1,$$

$$\langle s[3:0] \rangle = \langle t[4:0] \rangle - \langle 10 \rangle_{10} = \langle t[4:0] \rangle - \langle 1010 \rangle = \langle t[4:0] \rangle + \langle 0110 \rangle \bmod 2^4$$

Das Signal sel gebe dabei an, ob $\langle t \rangle > 9$:

$$sel = t_4 \vee (t_3 \wedge t_2) \vee (t_3 \wedge t_1)$$



Tiefe:

1. Die Tiefe des Conditional Sum Adders ist 9.

2. Das sel -Signal hat Tiefe 3.

3. Der 3/2-Addierer hat ebenfalls Tiefe 3.

3. Der Mux hat Tiefe 3.

$$D(\text{BCD-Adder}) =$$

$$\max\{D(\text{MuxInput bei } 0), D(\text{MuxInput bei } 1), D(\text{sel-Signal})\} + D(\text{Mux}) = \max\{9, 12, 12\} + 3 = 12 + 3 = 15.$$

(b) Es gilt $BCD^{-1}(s) = BCD^{-1}(a) + BCD^{-1}(b) + c$

$$\leq (10^n - 1) + (10^n - 1) + 1$$

$$= 2 \cdot 10^n - 1 = \langle 19^{n-1} \rangle_{10}$$

$$\text{also ist } BCD^{-1}(s[4n+3:4n]) \leq 1 \Rightarrow s_{4n+3} = s_{4n+2} = s_{4n+1} = 0$$