GTI HS 20 Lösung Serie 3

Michael Baur, Tatjana Meier, Sophie Pfister

Die 3. Serie ist bis Montag, den 12. Oktober 2020 um 12:00 Uhr zu lösen und als PDF-Dokument via ILIAS abzugeben. Für Fragen steht im ILIAS jederzeit ein Forum zur Verfügung. Zu jeder Frage wird, falls nicht anders deklariert, der Lösungsweg erwartet. **Lösungen ohne Lösungsweg werden nicht akzeptiert**. Allfällige unlösbare Probleme sind uns so früh wie möglich mitzuteilen, wir werden gerne helfen.
Viel Spass!

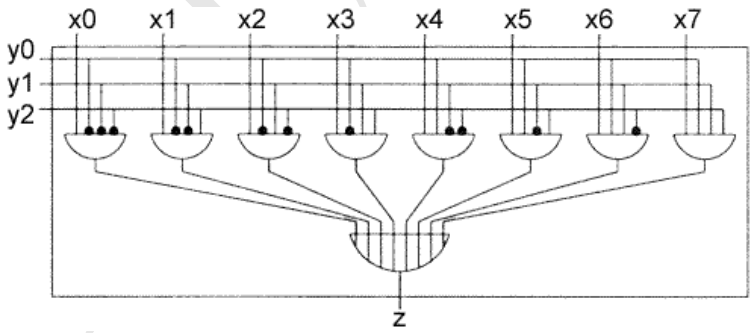
1 Mux und DeMux (4 Punkte)

(a) (2 Punkte) Stelle einen 3-Mux (8-to-1-Multiplexer) als Schaltung in disjunktiver Form (d. h. als zweistufige Schaltung, die eine Disjunktion von Konjunktionen ist) dar.

(b) (2 Punkte) Bestimme die Schaltfunktionen eines 3-DeMux (1-to-8-Demultiplexer).

Lösung

(a) Die Schaltfunktion in disjunktiver Form lautet $f(x_0, \ldots, x_7, y_0, y_1, y_2) = x_0(\neg y_0 \neg y_1 \neg y_2) + x_1(\neg y_0 \neg y_1 y_2) + x_2(\neg y_0 y_1 \neg y_2) + x_3(\neg y_0 y_1 y_2) + x_4(y_0 \neg y_1 \neg y_2) + x_5(y_0 \neg y_1 y_2) + x_6(y_0 y_1 \neg y_2) + x_7(y_0 y_1 y_2)$ und die Schaltung sieht somit wie folgt aus



(b) Die Schaltfunktionen lauten

$$
\begin{aligned}
z_0(x_0, y_0, y_1, y_2) &= x_0(\neg y_0 \neg y_1 \neg y_2) \\
z_1(x_0, y_0, y_1, y_2) &= x_0(\neg y_0 \neg y_1 y_2) \\
z_2(x_0, y_0, y_1, y_2) &= x_0(\neg y_0 y_1 \neg y_2) \\
z_3(x_0, y_0, y_1, y_2) &= x_0(\neg y_0 y_1 y_2) \\
z_4(x_0, y_0, y_1, y_2) &= x_0(y_0 \neg y_1 \neg y_2) \\
z_5(x_0, y_0, y_1, y_2) &= x_0(y_0 \neg y_1 y_2) \\
z_6(x_0, y_0, y_1, y_2) &= x_0(y_0 y_1 \neg y_2) \\
z_7(x_0, y_0, y_1, y_2) &= x_0(y_0 y_1 y_2)
\end{aligned}
$$

SOLUTION

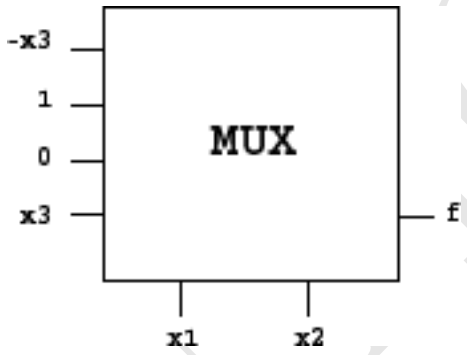
2 Mux realisiert Funktionstabelle (2 Punkte)

Realisiere die folgende Funktion $f(x_1, x_2, x_3)$ mittels eines 2-Muxs (4-to-1-Multiplexer). Zusätzliche Negationen bei Eingängen des Muxs sind erlaubt.

x_1	x_2	x_3	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Lösung

Eine mögliche Lösung sieht wie folgt aus:



3 Mux, DeMux und Decoder (2 Punkte)

(a) (0.5 Punkte) Wieviele Outputsignale hat ein Decoder mit d Steuersignalen?

(b) (0.5 Punkte) Wieviele Steuersignale hat ein Mux mit d Inputsignalen?

(c) (0.5 Punkte) Was ist die kleinste Anzahl Eingangssignale (Steuer- und Datensignale) für einen Mux (zusätzliche Negation bei Eingängen erlaubt), damit jede beliebige Funktion $f : B^4 \to B$ damit dargestellt werden kann? Begründe die Antwort.
(*Tipp*: Es sind weniger als $2^4 + 4 = 20$.)

(d) (0.5 Punkte) Welcher der drei Begriffe *Mux, DeMux, Decoder* passt jeweils zu den folgenden Aussagen

1. Kann durch Ersetzen des Inputsignales mit einer 1 zu einem Decoder gemacht werden.
2. Ist universell, d. h. damit lässt sich jede boolesche Funktion realisieren (mit entpsrechend genügend vielen Signalleitungen).
3. Erzeugt genau an einer Outputstelle eine 1.

Lösung

(a) 2^d Outputsignale

(b) $\log_2(d)$ Steuersignale

(c) 11 Eingangssignale, bestehend aus 8 Datensignalen und 3 Steuersignalen. (Stelle f als Funktion dar, die von nur drei Variablen abhängt, diese drei Variablen werden als Steuersignal verwendet, die vierte Variable kann als Inputsignal verwendet werden, vgl. Folie 27)

SOLUTION

(d) Welcher der drei Begriffe *Mux, DeMux, Decoder* passt jeweils zu den folgenden Aussagen

(a) **DeMux:** Kann durch Ersetzen des Inputsignales mit einer 1 zu einem Decoder gemacht werden

(b) **Mux:** Ist universell, d.h. damit lässt sich jede boolesche Funktion realisieren (mit entsprechend genügend vielen Signalleitungen).

(c) **Decoder:** Erzeugt genau an einer Outputstelle eine 1

4 Multiplexer (1 Punkt)

Erkläre wie mittels eines Multiplexers mehrere parallele Signale in ein serielles Signal umgewandelt werden können (Serialisierer). Welche Bedingung muss gegeben sein, damit der Empfänger der Nachricht das serielle Signal mit einem Demultiplexer wieder in das genau gleiche parallele Signal umwandeln kann?

Lösung

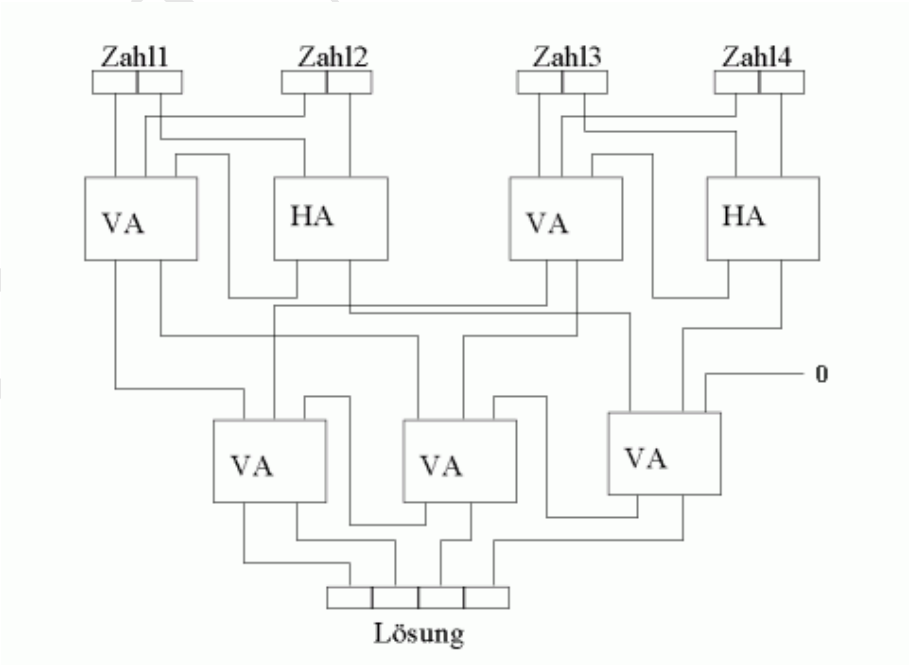
Mittels eines zyklischen Ablaufes kann das parallele Signal in ein serielles umgewandelt werden. Es ist eine Zeitsynchronisation zwischen dem Multiplexer und dem Demultiplexer nötig, so dass die Schalter in den gleichen Zeitabständen umschalten. Dadurch wir dann das Signal korrekt umgewandelt.

5 Addier-Netzwerk (2 Punkte)

Gegeben seien fünf Volladdierer und zwei Halbaddierer. Bilde daraus ein Addier-Netzwerk, das die Addition von vier zweistelligen Dualzahlen realisiert.

Lösung

Eine mögliche Lösung sieht wie folgt aus:



SOLUTION

6 Halbaddierer (4 Punkte)

(a) (1 Punkt) Realisiere einen Halbaddierer mit AND- und OR-Gattern. Du darfst Gatter mit eingebauter Negation im Eingang benutzen.

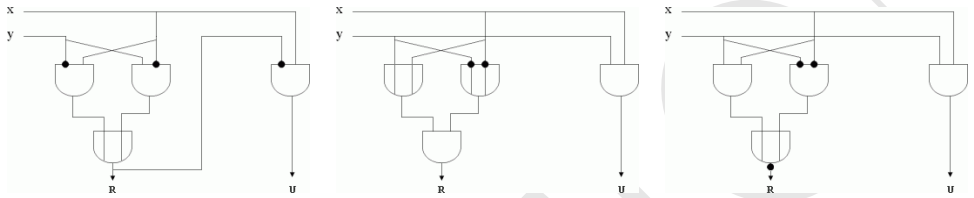
(b) (2 Punkte) Zeige (durch Induktion nach Aufbau der Schaltung), dass Schaltungen, welche nur aus AND- und OR-Gattern (ohne eingebaute Negation) bestehen, immer 1 ausgeben, wenn alle Inputsignale auf 1 gesetzt sind. Gib einen mathematischen Beweis für deine Antwort!

(c) (1 Punkt) Zeige, dass es unmöglich ist, einen Halbaddierer ausschliesslich mit AND- und OR-Gattern ohne eingebaute Negation zu realisieren.

Tipp: Betrachte die Wertetabelle für den Resultatsausgang eines Halbaddierers und benutze die Aussage von (b).

Lösung

(a) Mögliche Lösungen sehen wie folgt aus:



(b) Sei n die Grösse einer Schaltung die nur aus AND- und ODER-Gatter besteht.

Für $n = 1$ ist die Aussage klar.

Für $n > 1$ entfernen wir die letzte Gatter aus der Schlatung und wir haben zwei Schaltungen mit $< n$ Gatter. Aus der Induktionsanahme haben wir dass diese zwei Schaltungen den Wert 1 ausgeben, wenn alle Eingänge auf 1 gesetzt sind. Deswegen, gibt die originale Schaltung, auch den Wert 1 im Ausgang wenn alle Eingänge auf 1 gesetzt sind.

(c) Wir wissen von (b), dass eine Schaltung die nur aus AND- und OR-Gattern besteht immer den Wert 1 im Ausgang hat, wenn alle Eingänge auf 1 gesezt sind. Deswegen ist es nicht möglich einen Halbaddierer ausschliesslich mit AND- und OR-Gattern ohne eingebaute Negation zu realisieren, da man für den Resultateausgang den Output 0 benötigt, wenn beide Inputs 1 sind.

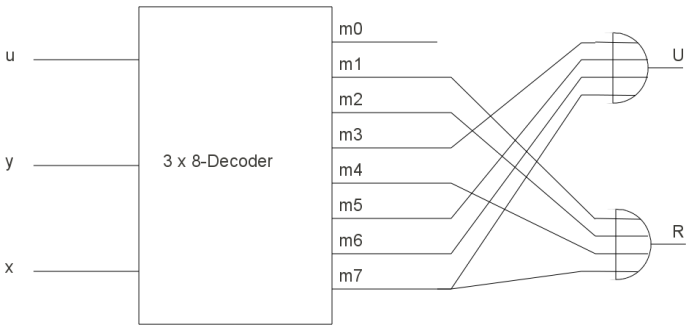
7 Decoder (1 Punkt)

Realisiere einen Volladdierer mittels eines Decoders und OR-Gatter.

Lösung

Wir haben folgende Wertetabelle

x	y	z	U	R
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



SOLUTION

Freiwillige Aufgabe

Grösse eines rekursiv aufgebauten Mux

Für $k = 1, 2, 4, 8, 16, \ldots$ können wir einen $2k$-Mux aus k-Muxen "rekursiv aufbauen", d.h. unter ausschliesslicher Verwendung von k-Muxen realisieren.

Zeige, dass für $d = 1, 2, 4, 8, 16, \ldots$ gilt:

$$G(d) = 3 \cdot (2^d - 1)$$

wobei $G(d)$ die Anzahl `AND`- und `OR`-Gatter in einem d-Mux ist.

Tipp: Versuch die Aussage durch Induktion nach d zu zeigen.

Lösung

Wir leiten zuerst eine rekursive Gleichung her. Offensichtlich gilt $G(1) = 3$, da der 1-Mux aus zwei `AND`- und einem `OR`-Gatter besteht.

Ein $2d$-Mux kann aus 2^d und einem d-Mux zusammengesetzt werden (wobei beim "zusätzlichen" d-Mux nicht notwendigerweise alle Eingänge benötigt werden), es gilt also, zusammenfassend:

$$\begin{aligned} G(1) &= 3 \\ G(2d) &= (2^d + 1) \cdot G(d) \end{aligned}$$

Wir zeigen die Behauptung mittels vollständiger Induktion, dass $G(d) = 3 \cdot (2^d - 1)$ gilt:

- $G(1) = 3 = 3 \cdot (2^1 - 1)$.
- $G(2d) = (2^d + 1) \cdot G(d) = (2^d + 1) \cdot 3 \cdot (2^d - 1) = 3 \cdot (2^{2d} - 1)$

SOLUTION