Lukas Batschelet, 16-499-733
Paulo Rangel Garcia, 23-111-461

GTI HS 23 Serie 3

Tobias Kohler, Nicolas Wyss, Maya Nedir

Die 3. Serie ist bis Mittwoch, den 18. Oktober 2023 um 16:00 Uhr zu lösen und in schriftlicher Form in der Übungsstunde 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, DeMux, Encoder und Decoder (4 Punkte)

(a) (1 Punkt) Welcher der vier Begriffe *Mux, DeMux, Encoder, Decoder* passt jeweils zu den folgenden Aussagen

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

(b) (1 Punkt) Stelle einen 2-Mux (4-to-1-Multiplexer) als Schaltung in disjunktiver Form (d. h. als zweistufige Schaltung, die eine Disjunktion von Konjunktionen ist) dar.

(c) (1 Punkt) Bestimme die Schaltfunktionen eines 2-DeMux (1-to-4-Demultiplexer).

(d) (0.5 Punkte) Wie viele Inputsignale hat ein $d-$Mux, also ein Multiplexer mit d Steuersignalen, insgesamt?

(e) (0.5 Punkte) Wie viele Outputsignale hat ein Encoder mit n Inputsignalen?

2 Für jede Funktion ein Mux (1 Punkt)

Erkläre, wie eine beliebige Funktion $f : B^n \to B$ mittels eines $(n-1)-$Multiplexers dargestellt werden kann.

3 Mit Mux, OR und Decoder (2 Punkte)

Gegeben sei ein 2x4 Decoder, ein OR-Gatter und ein 2-Mux. Bilde damit eine Schaltung, die genau für die folgenden Eingabewerte (x_0, x_1, x_2, x_3) den Wert 1 am Ausgang y liefert.

x_0	x_1	x_2	x_3
0	0	0	1
0	1	0	0
0	1	1	0
1	0	1	1
1	1	1	1

(*Tipp:* Leite (x_0, x_1) an den Decoder und (x_2, x_3) an den Steuereingang des Multiplexers.)

Lukas Batschelet, 16-499-733
Paulo Rangel Garcia, 23-111-461

4 Nullenzähler (6 Punkte)

Die Funktion $f : B^3 \to B^2$ berechne die Anzahl Nullen (in Binärdarstellung) in einem 3-bit Input. (z.B. $f(0,0,1) = (1,0)$, weil $(0,0,1)$ zwei Nullen enthält und $(2)_{10} = (10)_2$)

(a) (1 Punkt) Ergänze die Wertetabelle.

x_0	x_1	x_2	f_0	f_1
0	0	0		
0	0	1	1	0
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

(b) (2 Punkte) Realisiere eine Schaltung, die f berechnet, mittels zweier Halbaddierer und eines OR-Gatters. Zusätzliche Negationen sind erlaubt.
(*Tipp:* Vergleiche f mit der Funktion eines Volladdierers.)

(c) (3 Punkte) Realisiere eine Schaltung, die f berechnet, mittels zweier 2-Multiplexer (4-to-1). Zusätzliche Negationen bei Eingängen der Muxs sind erlaubt.

5 Addierer (3 Punkte)

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

(b) (2 Punkte) Bilde aus zwei Halbaddierern und zwei Volladdierern ein Addierwerk, das drei zweistellige Dualzahlen addiert.

Freiwillige Aufgaben

Volladdierer mittels Decoder

Realisiere einen Volladdierer mittels eines 3x8 Decoders und OR-Gattern.

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: Versuche, die Aussage durch Induktion nach d zu zeigen.

Lukas Batschelet, 16-499-733
Paulo Rangel Garcia, 23-111-461

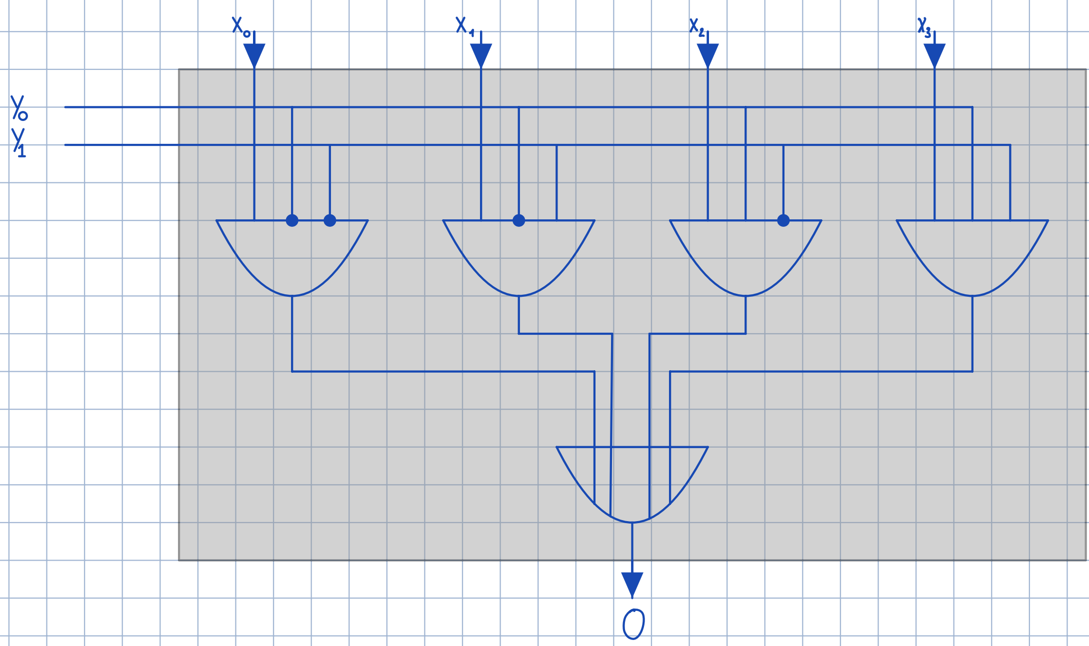
1a) 1. DeMux

2. Encoder

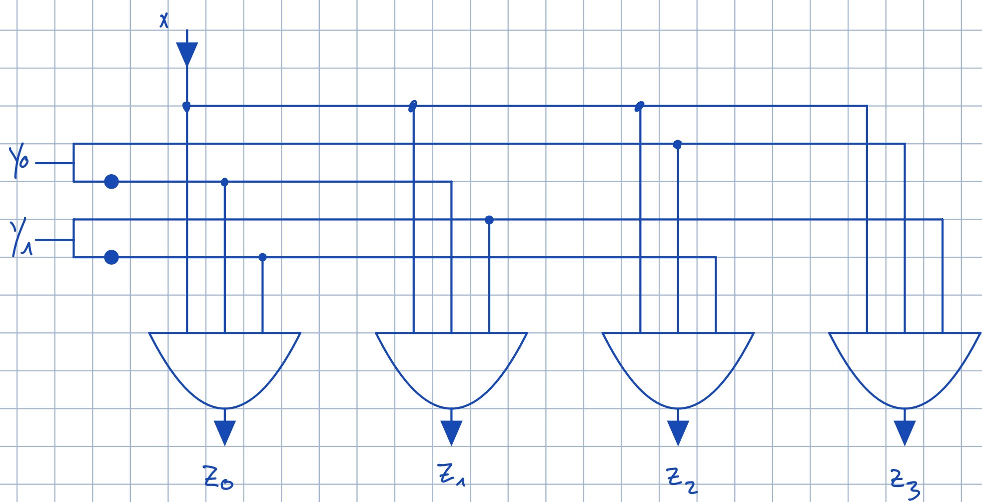
3. Mux (rein durch ausschliessen, ergibt sich mir nicht wirklich wie man bspw. NOR bauen will...)

4. Decoder

1b) 2 Mux



1c) 2 DeMux



Schaltfunktion: $x\,\bar{y}_0\,\bar{y}_1 + x\,\bar{y}_0\,y_1 + x\,y_0\,\bar{y}_1 + x\,y_0\,y_1$

1d) Ein d-Mux hat d Steuersignale und 2^d Dateninputs $\Rightarrow$ $d + 2^d$

1e) $\lceil \log_2(n) \rceil$

Lukas Batschelet, 16-499-733
Paulo Rangel Garcia, 23-111-461

2 Für jede Funktion ein Mux (1 Punkt)

Erkläre, wie eine beliebige Funktion $f : B^n \to B$ mittels eines $(n-1)$–Multiplexers dargestellt werden kann.

Ich vermute ein ähnliches Vorgehen wie auf Folie 27

Die ersten $(n-1)$ Dateneingänge werden als Steuersignale verwendet

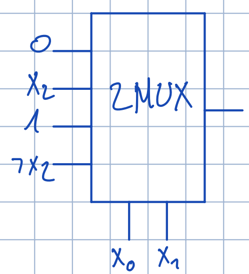
So entstehen immer 2er-Gruppen, welche vordefiniert durch das letzte Bit entschieden werden können. Für jede 2er-Gruppe muss „hinterlegt" sein wie das letzte Bit zum Output beiträgt $(0, x_n, 1, \neg x_n)$. Diese Werte können dann als jeweilige Eingangssignale verwendet werden

Bsp. $B^3 \to B$ mit einem 2-MUX

x_0	x_1	x_2	z	
0	0	0	0	→ 0
0	0	1	0	
0	1	0	0	→ x_2
0	1	1	1	
1	0	0	1	→ 1
1	0	1	1	
1	1	0	1	→ $\neg x_2$
1	1	1	0	

→

x_0	x_1	z
0	0	0
0	1	x_2
1	0	1
1	1	$\neg x_2$

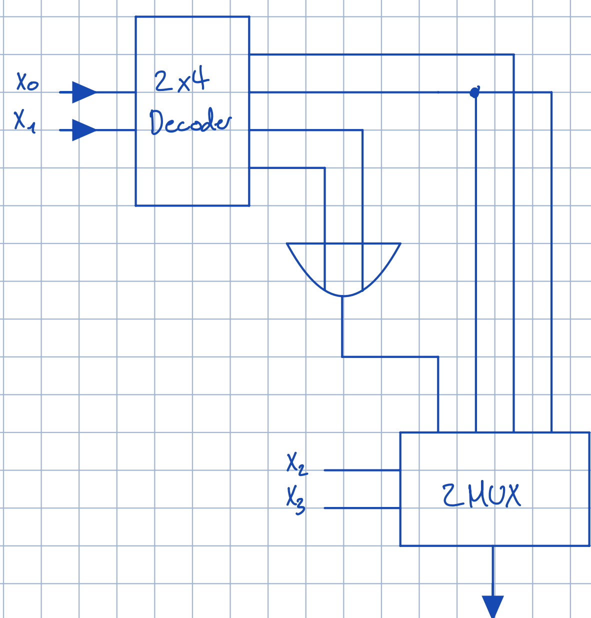


3 Mit Mux, OR und Decoder (2 Punkte)

Gegeben sei ein 2x4 Decoder, ein OR-Gatter und ein 2-Mux. Bilde damit eine Schaltung, die genau für die folgenden Eingabewerte (x_0, x_1, x_2, x_3) den Wert 1 am Ausgang y liefert.

x_0	x_1	x_2	x_3
0	0	0	1
0	1	0	0
0	1	1	0
1	0	1	1
1	1	1	1

(*Tipp:* Leite (x_0, x_1) an den Decoder und (x_2, x_3) an den Steuereingang des Multiplexers.)

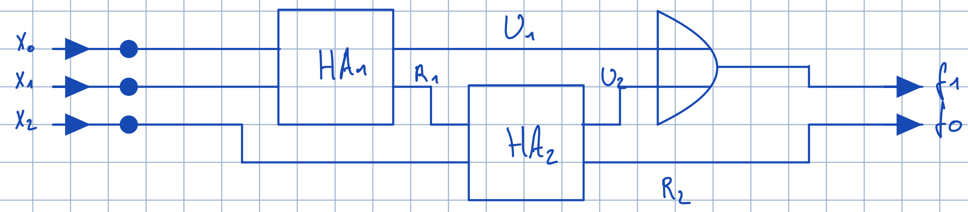


Lukas Batschelet, 16-499-733
Paulo Rangel Garcia, 23-111-461

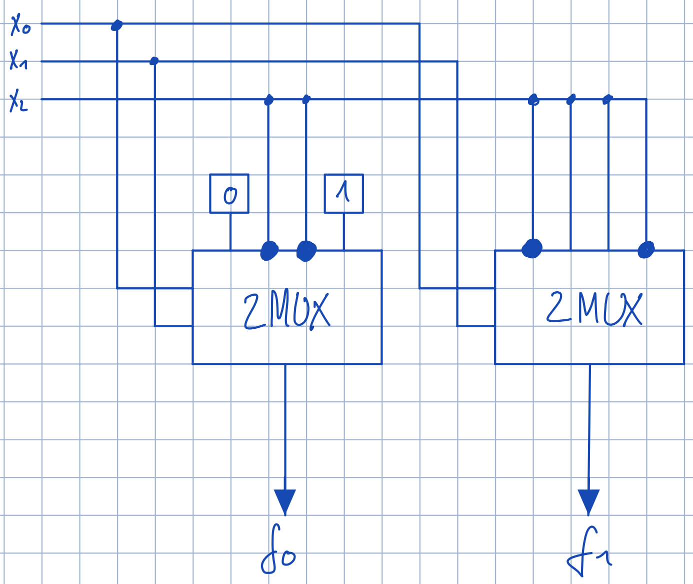
4a)

x_0	x_1	x_2	f_0	f_1
0	0	0	1	1
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	0	0

4b) Volladdierer mit negierten Eingängen...



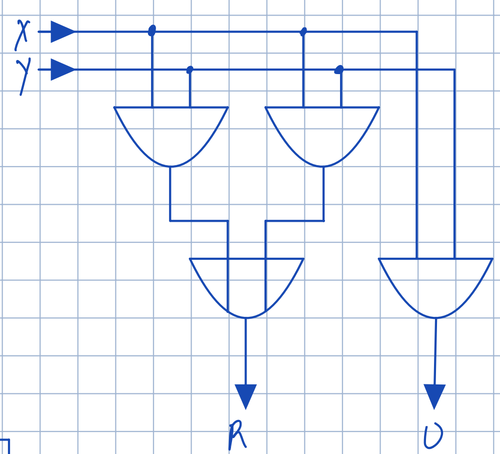
4c)



Ähnlich wie Aufgabe 2: x_2 als Schlüssel.

x_0	x_1	f_0	f_1
0	0	1	$\overline{x_2}$
0	1	$\overline{x_2}$	x_2
1	0	$\overline{x_2}$	x_2
1	1	0	$\overline{x_2}$

a) $x \oplus y = (\bar{x} \cdot y) + (x \cdot \bar{y})$



b)