

Prof. Dr. Armin Biere  
Dr. Mathias Fleury

Freiburg, 17. Mai 2023

## Technische Informatik Übungsblatt 5 (v2)

**Achtung 1:** Nächste Woche (21. bis 24.) gibt es keine Vorlesung und keine Übungen. Das Übungsblatt für den 31. ist aber nicht länger als sonst.

**Information 1:** Damit alle Tutoren gleichviele Abgaben korrigieren, werden ich manche Abgabeteams von einem Tutorat zu einer anderen Tutoren-Gruppe bewegen in Ilias (wahrscheinlich Dienstag). Dies ändert nur bei wem Sie abgeben auf Ilias. Sie sollen ganz normalen weiter zum selben Tutorat gehen wo Sie jetzt schon gehen. Sie werden über Mail informiert werden.

### Aufgabe 1 (4 Punkte)

Finden Sie eine möglichst kostengünstige CMOS-Realisierung (bzw. Anzahl der  $p$ - und  $n$ -Kanal-Transistoren aus der Vorlesung) für ein neues Gatter, welches die folgende boolesche Funktion berechnet:

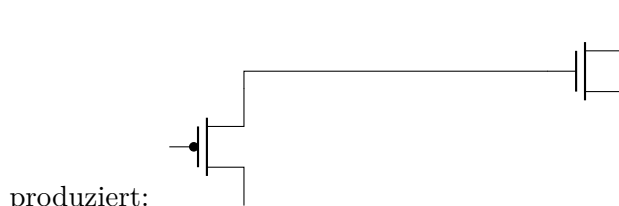
$$\bar{c} \wedge (\bar{a} \vee \bar{b})$$

Achten Sie darauf, dass es in Ihrem Gatter keinen Kurzschluss gibt, d.h. es gibt für keine Eingangsbelegung gibt es einen leitenden Pfad von der Spannungsquelle (1) zur Masse (0). Selbstzerstörerische Gatter bekommen keine Punkte.

*Hinweis 1:* Die volle Punktzahl erreichen Sie für eine ordentlich gezeichnete funktionierende Lösung mit maximal 6 Transistoren.

*Hinweis 2:* Zum Abbilden der Transistoren kann ich nur TikZ empfehlen

```
\begin{circuitikz}[transform shape, scale=1.0]
  \draw (0,0) node[nmos] (nmos_a) {};
  \draw (-5,-1) node[pmos] (pmos_b) {};
  \path (pmos_b.S) edge node[above] {something} (nmos_a.D);
\end{circuitikz}
```



*Hinweis 3:* Die Realisierung *darf* auch per Hand produziert werden. Es gibt einen Punkt dafür, dass die Realisierung ordentlich gezeichnet ist. Mit dem Computer produzieren erhöht die Wahrscheinlichkeit, dass Sie diesen Punkte bekommen.

## Aufgabe 2 (3 + 4 Punkte)

Wie Ihnen vielleicht aufgefallen ist, wurde in der Vorlesung nur für das *NAND*-Gatter und den Inverter (*NOT*) die *CMOS*-Realisierungen vorgestellt. Dies ist dadurch begründet, dass diese beiden Gatter ausreichen, um alle Booleschen Funktionen  $f : \mathbb{B}^2 \rightarrow \mathbb{B}$  zu realisieren. Tatsächlich ist das *NAND*-Gatter dafür ausreichend.

Realisieren Sie die folgenden Gatter der Standardbibliothek (*STD*), nur durch Verwendung von *NAND*-Gattern.

- *AND*
- *NOT*
- *OR*
- *NOR* (*negiertes OR*)
- *XOR*
- *XNOR* (*negiertes XOR*)

Die Kosten für die Realisierung eines Gatters dürfen dabei nicht größer als 5 sein.

## Aufgabe 3 (3 + 2 Punkte)

Die formale Beschreibung eines Schaltkreises  $C$  über der Bibliothek *STD* sei gegeben durch

$$C := (\vec{X}_3, (V, E), typ, IN, \vec{Y}_2),$$

wobei

$$\begin{aligned} V &= \{0, 1\} \cup \{x_1, x_2, x_3\} \cup \{v_0, v_1, v_2, v_3, v_4\} \\ E &= \{(v_2, v_4), (x_3, v_3), (x_2, v_1), (v_0, v_4), (v_1, v_3), (x_1, v_0), (x_3, v_2), (x_1, v_1), (x_2, v_0), (v_1, v_2)\} \\ \vec{X}_3 &= (x_1, x_2, x_3) \\ \vec{Y}_2 &= (v_3, v_4) \\ typ &= \{(v_i \mapsto \wedge) \mid i \in \{0, 2\}\} \cup \{(v_i \mapsto \oplus) \mid i \in \{1, 3\}\} \cup \{(v_4 \mapsto \vee)\} \\ IN &= \{(v_0 \mapsto ((x_1, v_0), (x_2, v_0))), (v_1 \mapsto ((x_1, v_1), (x_2, v_1))), (v_2 \mapsto ((v_1, v_2), (x_3, v_2))), \\ &\quad (v_3 \mapsto ((v_1, v_3), (x_3, v_3))), (v_4 \mapsto ((v_0, v_4), (v_2, v_4)))\}. \end{aligned}$$

a) Zeichnen Sie  $C$ .

- b) Führen Sie für  $C$  eine symbolische Simulation durch (geben Sie für jedes Gatter die an seinem Ausgang berechnete Funktion an).

#### Aufgabe 4 (5 Punkte)

Nutzen Sie das `retiquiz` Program von Prof. Biere, dass Sie letzte Woche installiert haben.

Nutzen Sie es mit `./retiquiz <ihre-matrikel-nummer>`. Schreiben Sie hier das Resultat auf, mit der Anzahl an Punkten die Sie bekommen haben. Falls Sie zu zweit abgeben, reicht es wenn es die Matrikelnummer von einem von ihnen ist. Geben Sie ihre ganze Lösung hab, also sowohl der Aufruf, das Programm mit den fehlendes Bits und auch die Punktzahl, wie zum Beispiel (wo ich immer '0' geraten habe):

```
$ ./retiquiz 42
ReTI Machine Code Quiz Version 0.0.2
retiquiz 42 16
Enter hexadecimal digits as an answer or
space ' ' to skip a question or 'q' to quit.
For irrelevant '*' in the machine code use '0'.
Asking 16 questions.
INSTRUCTION          ; PC      CODE
STOREIN1 13          ; 00000000 0000000d X expected 9 in 9000000d at I[31:28]
LOADIN2 IN1 5         ; 00000001 61000005 ✓
STOREIN2 12          ; 00000002 0000000c X expected a in a000000c at I[31:28]
LOADI IN1 28          ; 00000003 0100001c X expected 7 in 7100001c at I[31:28]
LOADIN1 ACC 17        ; 00000004 53000001 X expected 1 in 53000011 at I[7:4]
JUMP!= -28            ; 00000005 e80fffe4 X expected f in e8ffffe4 at I[23:20]
JUMP> -7              ; 00000006 c8f0fff9 X expected f in c8fffff9 at I[19:16]
JUMP> -27             ; 00000007 08ffffe5 X expected c in c8ffffe5 at I[31:28]
LOADIN2 ACC 3         ; 00000008 63000003 ✓
LOADIN1 ACC 19        ; 00000009 50000013 X expected 3 in 53000013 at I[27:24]
LOADIN2 IN1 13        ; 0000000a 6000000d X expected 1 in 6100000d at I[27:24]
LOADIN2 ACC 0         ; 0000000b 63000000 ✓
STORE 29              ; 0000000c 8000000d X expected 1 in 8000001d at I[7:4]
LOADIN2 IN2 24        ; 0000000d 62000008 X expected 1 in 62000018 at I[7:4]
JUMP!= -27            ; 0000000e e8fff0e5 X expected f in e8ffffe5 at I[11:8]
LOADIN2 IN1 8         ; 0000000f 61000008 ✓
RESULT
asked      100%   16/16
answered   100%   16/16
skipped     0%    0/16
correct    ✓ 25%    4/16
incorrect  X 75%   12/16
POINTS
```

0 points (more answers incorrect than correct)

TIME

18.10 seconds

Wenn Sie `./retiquiz` aufrufen kommt jedes mal ein neues Quiz raus, aber mit `./retiquiz <ihre-matrikel-nummer>` werden sie immer über dasselbe Programm abgefragt. So können Sie zum Beispiel das Quiz nochmal machen.

**Hinweis:** Falls Sie LaTeX benutzen müssen Sie ✓ und ✗ definieren, zum Beispiel mit:

```
\usepackage{pifont}
\usepackage{newunicodechar}
\newunicodechar{✓}{\ding{51}}
\newunicodechar{✗}{\ding{55}}
```

Abgabe: 31. Mai 2023, 17<sup>00</sup>