



Prof. Dr. Rolf Drechsler, drechsler@informatik.uni-bremen.de, MZH 3510
Dr. Robert Wille, rwille@informatik.uni-bremen.de, MZH 3485
Eleonora Schönborn, eleonora@informatik.uni-bremen.de, MZH 3350

2. Übungsblatt zur Vorlesung Technische Informatik 1

Aufgabe 1

(2 Punkte)

- Beschreibe den Unterschied zwischen synchroner und asynchroner Datenübertragung auf einem Bussystem.
- Welche Vorteile und Nachteile haben die jeweiligen Übertragungs-Schemata?

Aufgabe 2

(6 Punkte)

Hier soll die Steuerung einer Ampelanlage realisiert werden. Die gesamte Anlage besteht aus zwei Signalgruppen, deren Lichter unabhängig voneinander über die Steuersignale r_1, y_1, g_1 bzw. r_2, y_2, g_2 an- und ausgeschaltet werden können. Ein Steuersignal wird auf 1 gesetzt, wenn das Licht eingeschaltet ist, sonst auf 0. Dies ist exemplarisch in Abbildung 1 gezeigt.

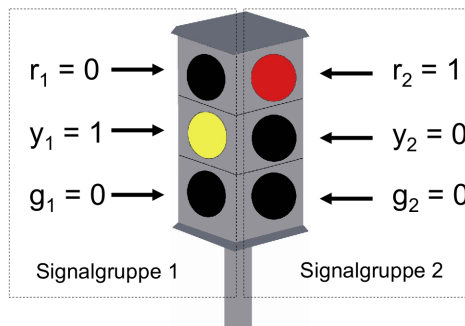


Abbildung 1: Ampelanlage mit zwei Signalgruppen

Die einzelnen Lichter werden entsprechend einem Dualzähler geschaltet, der damit die Funktion eines Taktgebers übernimmt. Der Zählerstand wird durch die Variablen x_3, x_2, x_1, x_0 repräsentiert. Zum Zeitpunkt 0 steht die Signalgruppe 1 auf Rot und die Signalgruppe 2 auf Rot-Gelb. Im nächsten Takt wird die Signalgruppe 2 auf Grün geschaltet. Die Rotphase dauert 8 Takte und die Grünphase jeweils 6 Takte. Rot und Gelb leuchten zusammen einen Zeittakt, Gelb alleine ebenfalls einen einzigen Takt. Schaltet eine Signalgruppe auf Gelb bzw. Rot-Gelb, steht die jeweils andere Signalgruppe auf Rot. Zu keinem Zeitpunkt dürfen beide Ampelgruppen auf Grün stehen.

Gib eine Boolesche Funktion $A : \mathbb{B}^4 \rightarrow \mathbb{B}^6$ durch eine Wahrheitstabelle und durch Boolesche Ausdrücke an, die diese Ampelsteuerung realisiert.

Aufgabe 3

(2 Punkte)

Zeige (z.B. in MIPS-Assembler), dass es möglich ist, eine Zuweisung mittels

- a) indirekter Adressierung ($R_{\text{dest}} := \text{Mem}[\text{Mem}[R_{\text{src}}]]$)
- b) indizierter Adressierung ($R_{\text{dest}} := \text{Mem}[R_{\text{src}} + \text{index}]$)

nur unter Verwendung von absoluter Adressierung ($R_{\text{dest}} := \text{Mem}[R_{\text{src}}]$) zu realisieren.

Aufgabe 4

(4 Punkte)

Es soll ein neuer Rechner für ein spezielles Einsatzgebiet entworfen werden. Die Rechengenauigkeit aller Operationen soll 32 Bit sein. Aufgrund der vorkommenden Operationen werden 166 Instruktionen benötigt. Der Rechner hat 42 Register bei einem Adressraum von maximal 750K. Es wird eine Load/Store-Architektur verwendet: Befehle, die auf den Speicher zugreifen, haben zwei Register als Operanden (Befehl 1: $R_{\text{dest}} := \text{Mem}[R_{\text{src}}]$, Befehl 2: $\text{Mem}[R_{\text{src}}] := R_{\text{dest}}$). Kein Befehl hat mehr als drei Operanden, mindestens zwei Operanden sind Register, der dritte kann eine 16-Bit-Konstante oder ein Register sein.

- a) Wie breit müssen die Register mindestens sein?
- b) Wie viele Bit werden benötigt, um eine Instruktion zu codieren?

Begründe deine Überlegungen.

Aufgabe 5

(6 Punkte)

Der folgende Algorithmus in C-Notation beschreibt den Euklidischen Algorithmus zur Berechnung des *größten gemeinsamen Teilers* (ggT) zweier natürlicher Zahlen.

```
int ggt ( int x, int y ) {
    int r;
    do {
        r = x % y;
        x = y;
        y = r;
    } while ( y != 0 );
    return x;
}
```

Gib ein MIPS-R2000-Assembler-Programm an, das den ggT zweier natürlicher Zahlen berechnet. Die Parameter x und y stehen zu Beginn in den Registern $\$t0$ und $\$t1$ zur Verfügung. Implementiere auch die Modulo-Operation und kommentiere dein Programm ausführlich!

Den Befehlssatz des MIPS R2000 findest du z. B. unter folgender URL:

http://www.iss.tu-darmstadt.de/student_area/tgdi/uebungen/mips-r2000.pdf

Um dein Programm zu testen, kannst du einen MIPS Simulator verwenden, z. B.:

<http://pages.cs.wisc.edu/~larus/spim.html>

<http://courses.missouristate.edu/KenVollmar/MARS/index.htm>

Abgabetermin: vor Beginn der Vorlesung am 10. Mai 2012