Universität Bremen – Fachbereich 3
Mathematik & Informatik

Prof. Dr. Rolf Drechsler, drechsler@informatik.uni-bremen.de, MZH 3510
Dipl.-Inf. Mathias Soeken, msoeken@informatik.uni-bremen.de, MZH 3560

2. Übungsblatt zur Vorlesung

Technische Informatik 1

Aufgabe 1 (2 Punkte)

Zeige (z.B. in MIPS Assembler), dass es möglich ist, eine Zuweisung mittels
a) relativer Adressierung (PC := Mem[PC] + offset)
b) indizierter Adressierung (Rdest := Mem[Rsrc + index])
nur unter Verwendung von absoluter Adressierung (Rdest := Mem[Rsrc]) zu realisieren.

Aufgabe 2 (2 Punkte)

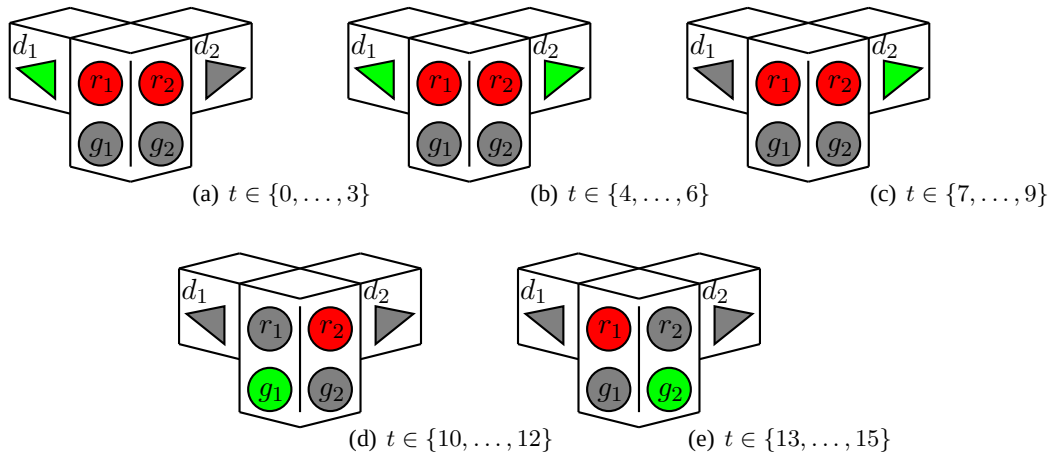
a) Beschreibe den Unterschied zwischen synchroner und asynchroner Datenübertragung auf einem Bus.
b) Welche Vorteile haben die jeweiligen Übertragungs-Schemata?

Aufgabe 3 (6 Punkte)

Hier soll die Steuerung einer Ampelanlage realisiert werden. Die gesamte Anlage besteht aus sechs Lichtern, die unabhängig voneinander über die Steuersignale $d_1, d_2, r_1, r_2, g_1, g_2$ 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.

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, bzw. $t = \sum_{i=0}^{3} 2^i x_i$ repräsentiert. Die Schaltzyklen der Ampel sind in Abbildung 1 ersichtlich.

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



(a) $t \in \{0, \ldots, 3\}$ (b) $t \in \{4, \ldots, 6\}$ (c) $t \in \{7, \ldots, 9\}$

(d) $t \in \{10, \ldots, 12\}$ (e) $t \in \{13, \ldots, 15\}$

Abbildung 1: Ampelschaltung

Aufgabe 4 (6 Punkte)

Implementiert einen Algorithmus zur Erkennung von Palindromen in MIPS-Assembler. Ein Palindrom ist ein Wort in dem alle Zeichen ausgehend von der Mitte symmetrisch angeordnet sind. Trotz Umkehrung der Zeichen gleicht das Wort also seiner ursprünglichen Form. Beispiele: rotor, lagerregal, anna. Der Algorithmus soll 0 für *false* und 1 für *true* zurückgeben:

```
1  public static boolean is_palindrom( String w )
2  {
3    boolean rval = true;
4    for ( int i = w.length() - 1; 0 <= i; --i )
5      rval &= w.charAt( i ) == w.charAt( w.length() - 1 - i );
6    return rval;
7  }
```

Achtung: Zusätzlich zum Papierausdruck soll eine digitale Abgabe eurer Implementierung via eMail an eure Tutoren erfolgen. Euer Programm soll auf dem Simulator `spim` der X-Rechner des FB3 lauffähig sein. Unkommentierte Programme werden nicht bewertet.

Tutorials für den Assembler, sowie Referenzen des Instruktionssatzes können hier gefunden werden:

- http://www.mrc.uidaho.edu/mrc/people/jff/digital/MIPSir.html
- http://refcards.com/docs/waetzigj/mips/mipsref.pdf
- http://www.cs.unibo.it/~solmi/teaching/arch_2002-2003/AssemblyLanguageProgDoc.pdf

Aufgabe 5 (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 126 Instruktionen benötigt. Der Rechner hat 65 Register bei einem Adressraum von maximal 8 GByte. Es wird eine Load/Store-Architektur verwendet: Befehle, die auf den Speicher zugreifen, haben zwei Register als Operanden (Befehl 1: Rdest := Mem[Rsrc], Befehl 2: Mem[Rsrc] := Rdest). Kein Befehl hat mehr als drei Operanden, mindestens zwei Operanden sind Register, der dritte kann eine 10-Bit-Konstante oder ein Register sein.

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

Begründe deine Überlegungen.

Abgabetermin: zu Beginn der Vorlesung am 29.04.2010