Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl. Inform. M. Klein
Dipl. Inform. T. In der Rieden

11. Übungsblatt Informatik II

(Abgabe: 18.07.2003)

Vereinfachungen bei der RTL-Programmierung:
Folgende einfachere Schreibweisen dürfen ab sofort und in der Klausur benutzt werden:

- Die expliziten Berechnungen `+0` und `+R0` müssen nicht mehr hingeschrieben werden. Man darf also schreiben: `R1 = 42` oder `R1 = M1(R4)`.
- Die Immediate-Konstante bei Sprungbefehlen kann durch ein Label ersetzt werden. Dieses Label muss dann vor der Zeile des Source-Codes stehen, in die gesprungen werden soll. Beispiel:

```
start:  R1 = R2 + R3                      // Kommentar
            ⋮
        PC = PC + (R4 = 0? start : 4)     // springe zurück zur Zeile start wenn R4 = 0
```

1. **Aufgabe: (Stall-Engine)** **(10 Punkte)**

   Betrachte einen Prozessor mit n Pipelinestufen. Wie in der Vorlesung besprochen berechnet sich das Stall-Signal für Stufe $i,\ 0 \leq i < n$, als:

   $stall_i := (lstall_i \vee stall_{i+1}) \wedge full_i$

   wobei $stall_n$ immer 0 ist.

   Seien alle $lstall_i$ und $full_i$ Signale bekannt. Gib einen Schaltkreis an, der alle $stall_i$ Signale berechnet mit Kosten $O(n)$ und Tiefe $O(\log n)$.

2. **Aufgabe: (Ausdrucksauswertung)** **(18 Punkte)**

   Es soll ein stackbasiertes RTL-Programm geschrieben werden, welches unvollständig geklammerte arithmetische Ausdrücke prioritätengestützt auswertet. Dazu sind folgende Voraussetzungen gegeben:

   (a) Der auszuwertende Ausdruck liegt ab Stelle 1000 im Speicher und reicht maximal bis Stelle 9999. Der Ausdruck beginnt mit einer öffnenden Klammer und endet mit einer schließenden Klammer.

   (b) Da man nicht unterscheiden kann, ob es sich bei dem Inhalt einer Speicherzelle um eine Konstante (also eine Zahl), eine Variable oder einen Operator handelt, fasst man immer zwei aufeinanderfolgende Speicherworte zu einem Symbol zusammen und codiert die Bedeutung dieser Paare wie folgt:

erstes Wort	zweites Wort	Bedeutung des Symbols	Anmerkungen
0x00000000	$X[31:0]$	Konstante X	
0x00000001	$X[31:0]$	Variable X	wert$(X) = \mathrm{M}_4(X)$
0x00000010	0x00000000	Operator “(”	beachte Sonderregel
0x00000010	0x00000001	Operator “)”	prio(“)”) = 0
0x00000010	0x00000010	Operator “$\vee$”	prio(“$\vee$”) = 1
0x00000010	0x00000011	Operator “$\wedge$”	prio(“$\wedge$”) = 2
0x00000010	0x00000100	Operator “=”	prio(“=”) = 3
0x00000010	0x00000101	Operator “+”	prio(“+”) = 4
0x00000010	0x00000110	Operator “–”	prio(“–”) = 4

Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl. Inform. M. Klein
Dipl. Inform. T. In der Rieden

11. Übungsblatt Informatik II

(Abgabe: 18.07.2003)

(c) Die Auswertungsregeln sind:

alter Stack	nächstes Symbol	neuer Stack	n. Symbol
*	Konstante C *'	* C	*'
*	Variable V *'	* wert(V)	*'
* C_1 op C_2	op' *' mit prio(op') > prio(op)	* C_1 op C_2 op'	*'
* C_1 op C_2	op' *' mit prio(op') ≤ prio(op)	* auswerten(C_1 op C_2)	op' *'
*	(*'	* (	*'
* (C	) *'	* C	*'

3. **Aufgabe: (Integer-Division)** **(12 Punkte)**

Schreibe ein Programm, das eine vorzeichenlose Integer-Division ausführt (ganzzahliger Anteil und Divisionsrest). Dabei soll gelten:

- Die Source-Operanden stehen in den Registern `R1` und `R2` mit $\langle$`R2`$\rangle \neq 0$.
- Das Programm berechnet `R3` und `R4` mit $\langle$`R4`$\rangle < \langle$`R2`$\rangle$, so dass gilt:
  $\langle$`R1`$\rangle = \langle$`R3`$\rangle \cdot \langle$`R2`$\rangle + \langle$`R4`$\rangle$

4. * **Aufgabe: (Selbstmodifizierender Code)** **(10 Punkte *)**

Schreibe ein RTL-Programm, das in jede Speicherzelle von 1024 bis 32767 die Adresse der jeweiligen Speicherzelle modulo 2^8 als neuen Inhalt hineinschreibt. Dabei gelten folgende Einschränkungen:

- Es sind nur absolute Adressierungen erlaubt, d.h. bei allen Speicherzugriffen berechnet sich der *mem*-Teil als (`R0 + imm`).
- Das Programm darf maximal 8 Befehle lang sein (es geht mit 6 Befehlen).

Gehe davon aus, dass es einen gemeinsamen Speicher für Instruktionen und Daten gibt und dass das Programm ab Speicherzelle 0 in diesem Speicher liegt.

Noch einige allgemeine Hinweise:

1. Um zur 2. Teilklausur bzw. zur Nachklausur zugelassen zu werden, muss man zweimal vorgerechnet haben und insgesamt über 225 Übungspunkte erreicht haben. Eine vorläufige Liste aller zugelassenen Studenten findet ihr wieder unter

   http://www-wjp.cs.uni-sb.de/lehre/vorlesung/info2/gruppen.php

2. Dies ist das letzte Übungsblatt. Es sollte euch klar sein, was dies für die Punktegrenze und für das Vorrechnen bedeutet.

3. Die 2. Teilklausur findet am Sa., 09.08.2003 von 9 Uhr s.t. bis 13 Uhr s.t. in den Hörsälen der Mathematik im Geb. 27 statt. Die Zuordnung Studenten → Hörsaal wird wieder an den Türen ausgehängt. Bitte seid wieder rechtzeitig da.

4. Die Nachklausur ist am Sa. den 04.10.2003 von 9 Uhr s.t. bis 13 Uhr s.t. ebenfalls in den Hörsälen der Mathematik im Geb. 27. Alles weitere wie oben.