



9. Übungsblatt Informatik II

(Abgabe: 04.07.2003)

Als Programmiersprache für unsere DLX-Maschine führen wir die sogenannte Register Transfer Language RTL ein. Diese beschreibt den Effekt der Instruktion auf die sichtbaren Register und sieht dementsprechend fast so aus wie die Effect-Spalte im Instruktionssatz.

Aus Effizienzgründen schreiben wir $R\langle a \rangle$ für $GPR(a)$, $a \in \{0, 1\}^5$ bzw. für $\langle GPR(a) \rangle$.

Wir nehmen an, dass Programme immer ab Speicherzelle 0 im Instruktionsspeicher liegen.

Beispiel: Folgendes Programm liest 2 Words aus den Speicherzellen 4 und 8 und speichert den Wert 42 in Speicherzelle R8, falls die beiden gelesenen Werte gleich groß sind:

```

R1 = M4(R0 + 4)           // R1 ist jetzt Inhalt von Zelle 4
R2 = M4(R0 + 8)           // R2 ist jetzt Inhalt von Zelle 8
R3 = (R1 = R2 ? 1 : 0)     // R3 ist 1 wenn die beiden Werte gleich sind
R4 = R0 + 42               // R4 ist jetzt 42
PC = PC + (R3 = 0 ? 8 : 4) // Führe den nächsten Befehl nicht aus, wenn R3 = 0, also
wenn die Werte unterschiedlich groß sind
M4(R8 + 0) = R4           // Schreibe 42 in Speicherzelle R8

```

Beachte: Kommentiere jede Zeile deiner Programme! Unkommentierte Programmzeilen werden NICHT korrigiert!

1. Aufgabe: (Memory-Tester) (4 + 6 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.
- Schreibe ein RTL-Programm, das für jede Speicherzelle von 1024 bis 32767 überprüft, ob der Inhalt der Speicherzelle gleich ihrer Adresse modulo 2^8 ist. Ist diese Bedingung für alle Zellen erfüllt, dann soll eine 1 in Speicherzelle 1020 abgelegt werden, sonst eine 0.

2. Aufgabe: (Zählerautomat) (5 + 5 + 5 Punkte)

Zu entwerfen ist ein Automat, der folgendes Verhalten zeigt:

- Der Automat hat die Eingangssignale up und $output$ und die Ausgangssignale $x[1:0]$.
- Die Zustandsmenge ist $Z = \{0, 1, 2, 3\}$ mit $z_0 = 0$.
- Für $z' = \delta(z, up)$ soll gelten:

$$z' = \begin{cases} z + 1 & (\text{mod } 4); up = 1 \\ z - 1 & (\text{mod } 4); up = 0 \end{cases}$$

$$4. \langle x \rangle = \begin{cases} z; output = 1 \\ 0; output = 0 \end{cases}$$

- Spezifiziere den Automaten über einen Graphen.
- Gib einen booleschen Ausdruck an, der die Next-State-Funktion berechnet.
- Gib einen booleschen Ausdruck an, der die Ausgabefunktion berechnet.



9. Übungsblatt Informatik II (Abgabe: 04.07.2003)

3. Aufgabe: (Zykluszeit)

(3 + 10 + 2 Punkte)

- (a) Schätze die Gesamt-Tiefe der DLX_S aus der Vorlesung ab.

Voraussetzungen:

- Die Tiefe der ALU beträgt 28.
 - Die Tiefen des Instruktionsmemory und des Datenmemory betragen jeweils 25.
 - Als Incrementer sind Conditional-Sum Incrementer, als Addierer sind Conditional-Sum Addierer zu benutzen.
 - Für jedes Register auf einem Pfad ist die Tiefe dieses Pfades um 2 zu erhöhen.
- (b) Schätze die Tiefen der 5 einzelnen Stufen der DLX_σ unter den gleichen Voraussetzungen ab.
- (c) Um welchen Faktor ist die DLX_σ demzufolge langsamer als die DLX_S ?

4. * Aufgabe: (Burst-Automat)

(4 + 6 Punkte *)

- (a) Entwirf auf Basis des Memory-Zugriffs-Automaten aus der Vorlesung einen neuen Automaten, der Burst-Zugriffe der Länge 4 unterstützt. Spezifiziere den Automaten entweder als 6-Tupel oder als Graph.

Erinnerung: Das Ende des Burtszugriffs erkennt man an $/reqp \wedge brdy$.

- (b) Entwirf einen Automaten, der das gleiche tut wie der Automat aus Teil (a). Dieser Automat soll jedoch bei Burst-Zugriffe die Nummer des jeweiligen Datenpaketes mitzählen. Dazu gibt es ein weiteres Ausgangssignal $nr[2:0]$, so dass $\langle nr[2:0] \rangle$ immer die laufende Nummer des im aktuellen Takt gültigen Datums ist. Ist kein Datum gültig, so soll 000 ausgegeben werden.