

3. Übungsblatt

Grundlagen der Anwendungsunterstützung

Christine Jakobs und Laura Morgenstern

Der Lehrcomputer 1 (LC1) [1] stellt einen einfachen Von-Neumann-Rechner dar. Der LC1 verfügt über einen Speicher aus 64 Worten zu 10 Bit. Eine E/A-Einheit besitzt der LC1 aus Gründen der Einfachheit nicht. Die Ein- bzw. Ausgaben werden direkt in den Speicher geschrieben bzw. gelesen. Der LC1 besitzt 16 Maschinenbefehle. Eine Kurzeinführung zur Architektur und den Befehlssatz des LC1 ist im Download-Archiv des LC1 enthalten.

Die Befehlseingabe kann über die LC1-Kommandozeile erfolgen. Die Befehle werden dabei in folgender Form eingegeben.

$$\langle \text{Adresse} \rangle \langle \text{Befehl} \rangle \langle \text{Operand} \rangle$$

Damit wird an Adresse $\langle \text{Adresse} \rangle$ im Speicher der Befehl $\langle \text{Befehl} \rangle$ eingesetzt. Sofern ein Befehl einen Operand erfordert, steht dieser an dritter Stelle. Zusätzlich können Speicherstellen über folgendes Kommando direkt modifiziert werden.

$$\langle \text{Adresse} \rangle \text{DEF} \langle \text{Wert} \rangle$$

Dabei wird die Dezimalzahl $\langle \text{Wert} \rangle$ an Adresse $\langle \text{Adresse} \rangle$ im Speicher abgelegt.

Der Ablauf des LC1 kann über einfache Tastendrücke gesteuert werden. Nach einem Reset des LC1 beginnt die Befehlsabarbeitung bei Adresse 0. Folgende Tastencodes sind für die Übung relevant.

Tastencode	Beschreibung
x	Führt den nächsten Befehl aus.
r	Führt alle Befehle bis zu nächsten HLT-Befehl aus.
g $\langle A \rangle$	Wie r, nur wird mit dem Befehl an Adresse $\langle A \rangle$ begonnen.
n	Die weitere Befehlsausführung erfolgt ohne Anzeige der Zwischenschritte.
d	Bei der weiteren Befehlsausführung werden die einzelnen Zwischenschritte angezeigt.
w	Wie d, nur wird nach jedem Zwischenschritt eine Pause eingelegt und der Nutzer zur Fortsetzung aufgefordert.
c	Führt einen Reset aus (löscht den Speicherinhalt und alle Register).
l $\langle D \rangle$	Lädt das Programm aus der Datei $\langle D \rangle$ in den Speicher.
pc $\langle A \rangle$	Setzt das PC-Register (Program Counter) auf die Adresse $\langle A \rangle$.
s $\langle D \rangle$	Speichert das aktuelle Programm in der Datei $\langle D \rangle$.
v $\langle A \rangle$	Zeigt den Speicherinhalt an Adresse $\langle A \rangle$.
e	Beendet den LC1-Simulator.

Aufgabe 1

- (i) Wie breit sind der Daten- und der Adressbus des LC1?
- (ii) Jeder Befehl wird in einem Wort kodiert. Wieviel Bits eines Befehlswords werden benötigt um alle LC1-Befehle zu kodieren? Wieviel Bits bleiben für einen Operanden übrig?
- (iii) Wie kann die LC1-CPU feststellen, ob ein Additionsergebnis gültig ist?
- (iv) Welche Adressierungsarten unterstützt der LC1?

Aufgabe 2

Laden Sie folgendes Programm in den LC1.

```
0 LDA 10 - Lade Inhalt aus #10 in Register A
1 LDB 11 - Lade Inhalt aus #11 in Register B
2 ADD   - Addiere A + B
3 MOV 12 - Speichere das Ergebnis (liegt in A) in #12
4 HLT   - Stoppe den Prozessor
10 DEF 23 - Definition der Variablen (#10=23)
11 DEF 42 - #11 = 42
12 DEF 0  - definierte 0 ist immer wichtig!
```

- a) Was macht dieses Programm?
- b) Notieren Sie den Inhalt der Register A und B nach jedem Befehl.
- c) Wieviele Speicherzugriffe finden statt? Unterscheiden Sie zwischen lesenden und schreibenden Speicherzugriffen.
- d) Führen Sie das Programm Schritt für Schritt im LC1-Simulator aus. Wie wirken sich Veränderungen in den Speicherzellen 10, 11 und 12 auf den Programmablauf aus?

Aufgabe 3

Schreiben Sie ein LC1-Programm, das vier Zahlen (an den Adressen 30, 31, 32 und 33) addiert und die Summe an Adresse 34 ablegt. Tritt während der Addition ein Überlauf ein, soll stattdessen 0 als Ergebnis abgespeichert werden.

Literatur

- [1] Lehrcomputer 1,
<http://www.tu-chemnitz.de/informatik/friz/Grundl-Inf/Rechnerarchitektur/LC1/index.php>
Nutzung unter Linux mit Hilfe des Programms dosbox möglich.