

Übungsblatt 1

Ausgabe: Mi, 27.10.99

Abgabe: Di, 02.11.99, 18 Uhr

Aufgabe 1: Rechneraufbau [10 Punkte]

In Abbildung 1 sehen Sie eine sogenannte „Random Access Machine“ (RAM). Der Prozessor der RAM kann Programme verarbeiten, die in der „Random Access Language“ (RAL) geschrieben sind. Die Basisaktionen, die der Prozessor versteht, sind in Tabelle 1 zusammengefaßt.

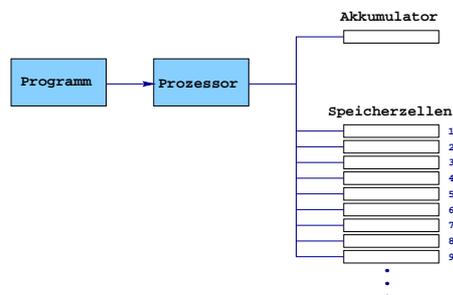


Abbildung 1: Random Access Machine

Jede Speicherzelle (auch der Akkumulator) kann eine Ganzzahl beliebiger Länge speichern. Jede Speicherzelle besitzt eine eindeutige Adresse mit der sie angesprochen werden kann. Der Akkumulator ist eine spezielle Speicherzelle, mit deren Hilfe der Prozessor Rechenbefehle ausführen kann.

Befehl	Argument	Bedeutung
LDA	X	lade den Akkumulator mit dem Inhalt der Speicherzelle X
LDI	X	lade den Akkumulator (indirekt) mit dem Inhalt der Speicherzelle, deren Adresse (Nummer) in der Speicherzelle X steht
STA	X	speichere den Inhalt des Akkumulators in Speicherzelle X
STI	X	speichere den Inhalt des Akkumulators (indirekt) in die Speicherzelle, deren Adresse (Nummer) in der Speicherzelle X steht
ADD	X	addiere den Inhalt von Speicherzelle X zum Akkumulator hinzu
SUB	X	subtrahiere den Inhalt von Speicherzelle X vom Akkumulator
JMP	Z	Springe zum Befehl mit der Nummer Z
JMZ	Z	Springe zum Befehl mit der Nummer Z, falls Akkumulator = 0
HLT		Beende das Programm

Tabelle 1: Die Sprache RAL

Ein Programm besteht aus einer Folge von durchnummerierten Befehlen. Die Befehle werden sequentiell abgearbeitet. Nur durch die Sprungbefehle JMP und JMZ, die angeben, mit welchem Befehl der Prozessor weitermachen soll, kann diese Abarbeitungsreihenfolge durchbrochen werden. Den Akkumulator indirekt mit LDI X zu laden bedeutet nicht den Inhalt von X zu

holen, sondern den Inhalt von X (nehmen wir einmal an es ist Y) als Adresse zu interpretieren und auf diese Stelle zuzugreifen (also quasi LDA Y auszuführen). STI funktioniert analog dazu.

- (a) [7 Punkte] Was macht das Programm unten mit der danebenstehenden Speicherbelegung? (Geben Sie für die ersten zwanzig Schritte jeweils die Belegung des Akkumulators und der Speicherzellen 1 bis 4 an.

1.	LDI	3	1	<table border="1"><tr><td>01</td></tr></table>	01
01					
2.	ADD	2	2	<table border="1"><tr><td>00</td></tr></table>	00
00					
3.	STA	2	3	<table border="1"><tr><td>05</td></tr></table>	05
05					
4.	LDA	3	4	<table border="1"><tr><td>10</td></tr></table>	10
10					
5.	SUB	4	5	<table border="1"><tr><td>12</td></tr></table>	12
12					
6.	JMZ	11	6	<table border="1"><tr><td>07</td></tr></table>	07
07					
7.	LDA	3	7	<table border="1"><tr><td>02</td></tr></table>	02
02					
8.	ADD	1	8	<table border="1"><tr><td>25</td></tr></table>	25
25					
9.	STA	3	9	<table border="1"><tr><td>18</td></tr></table>	18
18					
10.	JMP	1	10	<table border="1"><tr><td>04</td></tr></table>	04
04					
11.	HLT		11	<table border="1"><tr><td>17</td></tr></table>	17
17					

- (b) [3 Punkte] Wieviele Schritte führt das Programm insgesamt aus, bevor es beendet wird? Angenommen in der Speicherzelle 4 steht zu Anfang des Programmes n . Geben Sie die Anzahl der ausgeführten Programmschritte in Abhängigkeit von n an.

Aufgabe 2: Java-Programmierung [10 Punkte]

In dieser Aufgabe sollen Sie die Unterschiede zwischen einer Application und einem Applet herausarbeiten sowie Erfahrungen im Umgang mit dem Abstract Window Toolkit (AWT) sammeln.

Anmerkung: Um die Korrektur der Aufgaben zu erleichtern, benutzen Sie bitte den Klassennamen `Aufgabe<#><?>`. Dabei steht `<#>` für die Nummer des Übungsblatts und `<?>` für die Aufgabennummer. Daraus ergibt sich automatisch, daß die Datei mit dem Quelltext Ihres Programmes den Namen `Aufgabe<#><?>.java` erhält. Für Unteraufgabe (a) sollen Sie beispielsweise den Klassennamen `Aufgabe2a` benutzen.

- (a) [3 Punkte] Schreiben Sie ein Programm (Application), welches den Initialbuchstaben Ihres Namens mit „*“ schreibt. Benutzen Sie zur Ausgabe die Methode `println(...)` der Klasse `System.out`. Für den Buchstaben „A“ soll die Ausgabe beispielsweise folgendermaßen erfolgen:



- (b) [3 Punkte] Schreiben Sie nun das Programm so um, daß die Ausgabe mit Hilfe der Methode `drawString(...)` aus der Klasse `Graphics` erfolgt.
- (c) [4 Punkte] Schreiben Sie nun die beiden Programme aus den vorherigen Teilaufgaben so um, daß diese als Applet in einem Web-Browser oder mit dem Applet-Viewer laufen. Wo erfolgt die Ausgabe der „Print-Statements“, wenn ein Web-Browser zur Ausführung benutzt wird?

Anmerkung: Benutzen Sie bitte die Klassennamen `Aufgabe2ca` und `Aufgabe2cb`.