

1 Einfache mathematische Berechnungen mit der Minimaschine umsetzen

Setzen Sie die folgenden Berechnungen mit der Minimaschine um. [xxx] bedeutet dabei, den Inhalt der Speicherzelle mit der Adresse xxx zu verwenden; Term \rightarrow [yyy] bedeutet, dass der Wert des Terms in der Speicherzelle mit der Adresse yyy abgelegt werden soll. Assemblieren Sie das Programm anschließend und führen Sie es für verschiedene Werte in den Speicherzellen aus.

- a $[100] - [101] + [102] \rightarrow [103]$.
- b $[100] + [101] \cdot [102] \rightarrow [103]$. Beachten Sie bei der Umsetzung die Regel „Punkt vor Strich“.
- c $[100] + 2 \cdot [101] + 3 \cdot [102] + 4 \cdot [103] \rightarrow [104]$
Tipp: Legen Sie die konstanten Werte in geeigneten Speicherzellen ab.
- d Erstellen Sie ein Programm, das den Inhalt von zwei Speicherzellen vertauscht.

2 Komplexe Terme mit der Minimaschine umsetzen

Setzen Sie die Berechnung folgender Terme mit Hilfe der Minimaschine um. Ordnen Sie den Platzhaltern Speicherzellen mittels symbolischer Adressierung zu und speichern Sie das Ergebnis in einer Speicherzelle mit einem Namen Ihrer Wahl.

- a $(a + b) / c$
- b $a + (b - c) \cdot d$
- c $a - (b + c) \cdot (b - c)$
- d $a^2 + b^2$

3 Befehlszyklus

Für die folgenden Teilaufgaben hat der Speicher die Vorbelegung wie im Screenshot rechts zu sehen.

90	0	0	0
100	4	7	0
110	100	101	0
120	0	0	0
130	0	0	0
140	0	0	0
150	0	0	0

- a Führen Sie die Befehle LOAD 100, LOADI 4 und LOAD (110) im Mikroschrittmodus aus. Notieren Sie für jeden Befehl die Phasen im Befehlszyklus, den Wert des Akkumulators am Ende jeder Phase, die Werte auf Adress- bzw. Datenbus, sowie eventuelle Signale auf dem Steuerbus. Nennen Sie die wesentlichen Unterschiede zwischen den drei Befehlsformen.
- b Führen Sie nun die Befehle ADD 101, ADDI 7 und ADD (111) im Mikroschrittmodus aus. Notieren Sie für jeden Befehl die Phasen im Befehlszyklus, den Wert des Akkumulators am Ende jeder Phase, die Werte auf Adress- bzw. Datenbus, sowie eventuelle Signale auf dem Steuerbus. Nennen Sie die wesentlichen Unterschiede zwischen den drei Befehlsformen.
- c Geben Sie die wesentlichen Unterschiede zwischen den entsprechenden Befehlen aus Teilaufgabe a) und Teilaufgabe b) an (LOAD vs. ADD, LOADI vs. ADDI, LOAD () vs. ADD ()).
- d Die Befehle LOAD und STORE haben den gleichen Ablauf im Befehlszyklus. Erläutern Sie anhand der Aufzeichnungen bei Teilaufgabe a), worin genau sich die Ausführung der Befehle unterscheidet, und verifizieren Sie Ihre Vermutung durch Ausführen eines STORE-Befehls.

4 Rudimentäre Felder

Basis für Felder ist die Speicherung der Feldelemente in aufeinanderfolgenden Speicherzellen. Im Folgenden wird ein Feld f mit 10 Elementen betrachtet, das in Bild ab Speicherzelle 100 angelegt ist. Die Indexwerte laufen von 0 bis 9.

80	0	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0
110	0	0	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0	0	0
130	0	0	0	0	0	0	0	0	0	0

Speicherzellen mit den Feldelementen

- a Schreiben Sie ein Programm, das die Adresse des fünften Feldelements berechnet; das Ergebnis soll im Akkumulator bleiben.
- b Ein Indexwert ist in einer Zelle mit dem Namen i abgelegt. Entwickeln Sie ein Programm, welches das i-te Feldelement ausliest und in Speicherzelle res ablegt.
Tipp: Nutzen Sie Teilaufgabe a) und die indirekte Adressierung.
- c Entwerfen Sie ein zweites Programm, welches den Wert in Speicherzelle wert an der i-ten Position im Feld ablegt.
- d Ein zweiter Indexwert wird mit j bezeichnet. Entwerfen Sie ein Programm, das die beiden Feldelemente an den Stellen i und j miteinander vertauscht. Testen Sie auch für den Sonderfall $i = j$.

- e Begründen Sie, warum auch bei Hochsprachen sehr oft Feldindizes ab 0 zählen. Geben Sie insbesondere an, was Sie bei den obigen Programmen verändern müssten, wenn der Indexwert von 1 bis 10 laufen würde.
- f Für Schnelle: Geben Sie an, wie die Teilaufgaben a) bis d) abgeändert werden müssen, wenn ein Feldelement nicht eine, sondern zwei Speicherzellen belegt. Setzen Sie Ihre Überlegungen um und testen Sie.

5 Zwei Zahlen sortieren

Die zwei Zahlen in den Speicherzellen 100 und 101 sollen sortiert werden.

- a Die kleinere Zahl soll in Speicherzelle 102, die größere in 103 abgelegt werden.
Formulieren Sie zunächst einen kurzen Pseudocode für den Algorithmus und setzen sie ihn dann in Assemblercode um.
- b Die kleinere Zahl soll in Speicherzelle 100 und die größere in Speicherzelle 101 abgelegt werden (also in denselben Speicherzellen wie die Ausgangszahlen).
Formulieren Sie einen kurzen Pseudocode und reflektieren Sie, inwiefern die Veränderung in der Aufgabenstellung die verwendete Kontrollstruktur beeinflusst.
Setzen Sie Ihren Algorithmus dann in Assemblercode um.

6 Multiplikation ohne MUL / Division ohne DIV

Für die folgenden Aufgaben werden zunächst nur positive Zahlen betrachtet, ein Überlauf braucht nicht kontrolliert zu werden.

- a Setzen Sie die Multiplikation als wiederholte Addition um, also durch Aufaddieren eines Faktors so oft, wie der andere Faktor angibt.
- b Setzen Sie analog dazu die Division zweier Zahlen durch wiederholte Subtraktion um.
- c Für Schnelle: Ergänzen Sie die Lösung von a) so, dass auch negative Faktoren berücksichtigt werden.

7 Fakultät!

Es soll die Fakultät einer Zahl n berechnet und das Ergebnis in Zelle `res` abgelegt werden

- a Setzen Sie das Programm um.
- b Führen Sie das Programm mit unterschiedlichen Werten für n aus. Finden Sie heraus, ab welchem Wert für n das Programm fehlerhafte Ergebnisse liefert und finden Sie (z. B. durch Nutzen des Einzelschrittmodus) die Programmstelle, an der der Fehler verursacht wird.
- c Ergänzen Sie das Programm so, dass im Fehlerfall die Berechnung abgebrochen und als Ergebnis in `res` der Wert -1 gespeichert wird.

8 Felddurchlauf

Es soll die Summe der Elemente eines Feldes mit 5 Elementen berechnet werden. Dazu wird – wie üblich – eine Zählwiederholung verwendet.

- a Legen Sie Speicherplatz und Inhalt des Feldes mit fünf `WORD`-Befehlen fest; der erste Befehl erhält die Marke `feld`.
- b Implementieren Sie eine Zählwiederholung, welche die Zählvariable i die Werte von 0 bis 4 durchlaufen lässt.
- c Ergänzen Sie das Programm so, dass in der Zählwiederholung die Summe der Feldelemente `feld[i]` in der Variablen `sum` berechnet wird.
- d Für Schnelle: Ändern Sie das Programm so ab, dass die Länge des Feldes in einer Variablen mit dem Namen `laenge` gespeichert wird.