

Professor Dr.-Ing. Stefan Kowalewski  
Dipl.-Inform. Andreas Polzer  
Dipl.-Inform. Ralf Mitsching

Aachen, 19. Januar 2007  
SWS: V2/Ü2, ECTS: 4

## Einführung in die Technische Informatik WS 2006/2007

### Blatt 8: Donald Ervin Knuth's MMIX

Ihre Lösung zu den mit (★) gekennzeichneten Übungen sollen Sie am **26.01.2007** in der Übung abgeben. Die Bearbeitung der Aufgaben in Lerngruppen ist sinnvoll. Bitte geben Sie nur eine Lösung pro Lerngruppe ab.

#### Aufgabe 1: (★) Einführung der MMIX-Umgebung

Diese Aufgabe soll in die Benutzung der MMIX-Umgebung einführen. Für weitere Informationen schauen Sie bitte auf folgende Webseite:

<http://www-kbsg.informatik.rwth-aachen.de/misc/RS2006/mmix.php>

- a) Erzeugen Sie die Datei `einfach.mms`, die folgendes Programm enthält (Zeilennummern und Kommentare nicht mit eingeben, Programm ist auch auf der Webseite verfügbar):

1	LOC #100	Start des Programms im Speicher
2	x IS \$42	Anweisung, 'x' im Programm durch \$42 zu ersetzen
3	y IS \$43	...
4	z IS \$44	...
5	Main SET x,40	Main: hier beginnt das Hauptprogramm
		Setze x (also das Register 42, \$42) auf den Wert 40
6	SET y,7	Setze y auf 7
7	MUL \$44,\$42,\$43	Multipliziere Reg. 42 mit Reg. 43, schreibe das Ergebnis nach Reg. 44
8	ADD z,z,3	...
9	DIV y,z,13	...
10	TRAP 0,Halt,0	Ende des Programms

- b) Erzeugen Sie den assemblierten Code `einfach.mmo`.
- c) Welchen Wert enthalten die Register 42, 43 und 44 nach der Ausführung von Zeile 9 durch MMIX? Wie kommt man an diese Information bei Benutzung von `mmix`? (Außer durch Ausrechnen im Kopf.)
- d) Ändern Sie das Programm so ab, dass anfangs  $x = 6$ ,  $y = 7$  und  $z = 20$  gilt, der Term  $xy(x + y)/(z - y)$  berechnet wird und das Ergebnis in Register 14 steht.

## Aufgabe 2: Kontrollstrukturen im MMIX

Geben Sie MMIX Assembler-Befehle zur Simulation folgender Kontrollstrukturen an:

a) `while X > Y do S;`

b) `if A = B then begin X:=X+1; Y:=Z end else A:=B;`

c) `for J:= LAST downto FIRST do S;`

d) Verwenden Sie die Aufgabenteile (a)–(c) zur MMIX-Assemblerprogrammierung des nebenstehenden sogenannten Bubblesort-Algorithmus zur Sortierung eines Arrays. Alle Zahlen seien vom Datentyp *wyde*.

```
last:=num;
while last > 0 do begin
  pairs:=last-1;
  last:=0;
  for j:=1 to pairs do
    if list[j] > list[j+1]
    then begin
      tmp:=list[j];
      list[j]:=list[j+1];
      list[j+1]:=tmp;
      last:=j;
    end;
  end;
end;
```

## Aufgabe 3: Mittelwertberechnung unter MMIX

Schreiben Sie ein MMIX-Programm, das den arithmetischen Mittelwert einer Folge von Zahlen, die sich im Speicher befinden, berechnet. Die Zahlen werden als ganze Zahlen betrachtet und eventuelle Rundungsfehler werden vernachlässigt. Das Programm soll die Anzahl der Zahlen aus der mit Marke `num` versehenen Speicherstelle laden. Als Vorlage dient nebenstehendes Fragment.

```
const
  num := 6;
var
  count : octa;
  sum    : octa;
  result : octa;
  data   : array [0..5] of octa;
begin
  count := num;
  sum   := 0;
  while count > 0
  begin
    count := count - 1;
    sum   := sum + data[count];
  end;
  result := sum DIV num;
end
```

### Aufgabe 4: (★) Wurzelberechnung unter MMIX

Schreiben Sie ein MMIX-Assemblerprogramm, das die folgende Aufgabenstellung löst: Das Programm soll den ganzzahligen Anteil der Quadratwurzel zu einer gegebenen ganzen Zahl berechnen, d.h. (mit  $\lfloor x \rfloor = \max\{z \in \mathbb{Z} \mid z \leq x\}$ , also  $x$  „nach unten abgerundet“)

$$\text{result} = \lfloor \sqrt{\text{value}} \rfloor \quad .$$

Dazu wird von der Intervallschachtelung ausgegangen, die sich aus der nebenstehenden Newton-Methode ergibt.

```

a := value
b := 0
while |a - b| > 1
    a := ⌊(a + b)/2⌋
    b := ⌊value/a⌋
end_while
result := min{a, b}

```

### Aufgabe 5: (★) Sattelpunkte berechnen unter MMIX

Gegeben sei eine  $9 \times 8$ -Matrix der folgenden Gestalt:

$$\begin{pmatrix} a_{00} & a_{01} & \cdots & a_{07} \\ a_{10} & a_{11} & \cdots & a_{17} \\ \cdots & \cdots & \cdots & \cdots \\ a_{80} & a_{81} & \cdots & a_{87} \end{pmatrix}$$

Ein Matrix-Element  $a_{ij}$  wird als Wyde angenommen. Im Speicher des MMIX-Prozessors ist diese Matrix als Folge von Wydes ab einer Basisadresse  $A$  abgelegt. Die Adresse des Matrix-Elements  $a_{ij}$  entspricht damit  $M_2[A + (i * 8 + j) * 2]$ .

Eine  $m \times n$ -Matrix besitzt einen sog. *Sattelpunkt*, falls es ein Matrix-Element gibt, das den kleinsten Wert in seiner Zeile und den größten in seiner Spalte darstellt. Formal:  $a_{ij}$  ist Sattelpunkt genau dann wenn

$$a_{ij} = \min_{1 \leq k \leq n} a_{ik} = \max_{1 \leq k \leq m} a_{kj}.$$

Entwickeln Sie ein MMIX-Programm, das eine gegebene  $9 \times 8$ -Matrix auf die Existenz (mindestens) eines Sattelpunktes untersucht. Im positiven Fall sind Zeile und Spalte des Elementes in den Registern \$0 und \$1 abzulegen. Existiert kein Sattelpunkt, ist Register \$0 mit -1 zu belegen.