zu Aufgabe 3.1) MIPS-Rate am Toy-Rechner:

Das Beispielprogramm:

```
;       Autor:   Martin Beck
;       Programmierdauer: ca. 30 min.
;
;       Mit diesem Quelltext soll die MIPS-Rate des ToyEmulator gemessen werden
;
$1010  ; Lade Akku mit Wert aus Zelle 10hex
$0011  ; Speichere Akku in Zelle 11hex
$1011  ; Lade Akku mit Wert aus Zelle 11hex
$a000  ; Dekrementiere Akku
$0011  ; Speichere Akku in Zelle 11hex
$2008  ; Falls Akku = 0 springe nach 08hex
$b000  ; Setze Akku = 0
$2002  ; (Falls Akku = 0) springe nach 02hex
$100f  ; Lade Akku mit Wert aus Zelle 0Fhex
$a000  ; Dekrementiere Akku
$000f  ; Speichere Akku in Zelle 0Fhex
$200e  ; Falls Akku = 0 springe nach 0Ehex
$b000  ; Setze Akku = 0
$2000  ; (Falls Akku = 0) springe nach 00hex
$f000  ; STOP
$000f  ; Aeusserer Schleifenwert U
$000f  ; Innerer Schleifenwert V
;
;       Zeile 02hex bis 07hex werden U-mal durchlaufen, bei einem aeusseren
        Durchlauf
;       Zeilen 00hex, 01hex, 08hex bis 0Dhex werden je V-mal durchlaufen
;       N = 8 * V + 6 * U * V  =  2 * V * (4 + 3 * U)        (Anzahl der Anweisungen)
;       fuer U = 15 und V = 15 ergaebe sich dann N = 1470
```

zu Aufgabe 3.1) Testprotokoll

Für dieses Programm wurde eine Zeit t gemessen für 1470 Befehle.
Es gilt also folgende Formel:

MIPS-Rate = 1470 / 1000000 / t

Für die Rechner ergeben sich demnach folgende MIPS-Raten:

Tipc001:
Gemessene Zeit: 268 s
MIPS-Rate : 5,4/1000000 mips

Tipc006:
Gemessene Zeit: 379 s
MIPS-Rate : 3,8/1000000 mips

Tipc010:
Gemessene Zeit: 167 s
MIPS-Rate : 8,8/1000000 mips

Tipc013:
Gemessene Zeit: 70 s
MIPS-Rate : 2,1/100000 mips

zu Aufgabe 3.2) Quelltexte

Die Beispielprogramme sehen wie folgt aus:

MIPSfast:

```
include macro.bib

Stack_ SEGMENT STACK
        dw 100h DUP(?)        ; Stack mit 256 words initialisiert
Stack_ ENDS

ASSUME CS:Code,SS:Stack_


Code SEGMENT

Start:          SHOW_TIME     ; Anzeige der Zeit und Zeilenvorschub
                mov al,10
                mov dl,al
                SHOW_DL
                mov al,13
                mov dl,al
                SHOW_DL
                xor ax,ax
                mov cx,ax
I1:             xor ax,ax
                mov dx,ax
I2:             inc al
                inc dx
                cmp dx,04fffh
                jne I2
                inc cx
                cmp cx,04fffh
                jne I1
                SHOW_TIME
                MSDOS


SHOW_AL PROC NEAR
                push ax
                push bx
                push cx
                push dx
                push ax
                mov cl,4
                shr al,cl
                mov bh,al
                pop ax
                mov cl,0fh
                and al,cl
                mov bl,al
                mov ax,bx
                add ax,3030h
                cmp ah,3ah
                jl M1
                add ah,07h
M1:         cmp al,3ah
                jl M2
                add al,07h
M2:         mov dl,ah
                SHOW_DL
                mov dl,al
```

```
                SHOW_DL
                pop dx
                pop cx
                pop bx
                pop ax
                RET
SHOW_AL ENDP


Code ENDS

END Start
```

und MIPSslow.asm:

```
include macro.bib

Stack_ SEGMENT STACK
        dw 100h DUP(?)        ; Stack mit 256 words initialisiert
Stack_ ENDS

ASSUME CS:Code,SS:Stack_


Code SEGMENT

Start:          SHOW_TIME     ; Anzeige der Zeit und Zeilenvorschub
                mov al,10
                mov dl,al
                SHOW_DL
                mov al,13
                mov dl,al
                SHOW_DL
                xor ax,ax
                mov cx,ax
I1:             xor ax,ax     ; U
                mov dx,ax     ; U
I2:             mul bl        ; V
                inc dx        ; V
                cmp dx,04fffh ; V       ; Innere Schleifenzahl
                jne I2        ; V
                inc cx        ; U
                cmp cx,04fffh ; U       ; Äußere Schleifenzahl
                jne I1        ; U
                SHOW_TIME
                MSDOS


SHOW_AL PROC NEAR
                push ax
                push bx
                push cx
                push dx
                push ax
                mov cl,4
                shr al,cl
                mov bh,al
                pop ax
                mov cl,0fh
                and al,cl
```

```
                mov bl,al
                mov ax,bx
                add ax,3030h
                cmp ah,3ah
                jl M1
                add ah,07h
M1:             cmp al,3ah
                jl M2
                add al,07h
M2:         mov dl,ah
                SHOW_DL
                mov dl,al
                SHOW_DL
                pop dx
                pop cx
                pop bx
                pop ax
                RET
SHOW_AL ENDP


Code ENDS

END Start
```

zu Aufgabe 3.2) Testprotokoll

Bei Vernachlässigung der Zeitausgabe ergibt sich die Anzahl der Befehle aus folgender Gleichung:

$N = 5 * U + 4 * U * V = U * (5 + 4 * V)$
$N = 1.677.660.159$

Die Formel zur Bestimmung der MIPS-Rate lautet demzufolge:
MIPS-Rate = $1677660159/1000000/\Delta t$

Es ergeben sich also die MIPS-Raten für folgende Rechner:

tipc 001:
MIPSfast: Startzeit = 11:43:40
Endzeit = 11:43:47
Zeitdifferenz (Δt): 7s
MIPS-Rate = 239 MIPS

MIPSslow: Startzeit = 11:44:12
Endzeit = 11:44:58
Zeitdifferenz (Δt): 46s
MIPS-Rate = 36 MIPS

tipc 006:
MIPSfast: Startzeit = 11:57:19
Endzeit = 11:57:30
Zeitdifferenz (Δt): 11s
MIPS-Rate = 152 MIPS

MIPSslow: Startzeit = 11:57:51
Endzeit = 11:59:05
Zeitdifferenz (Δt): 74s
MIPS-Rate = 22 MIPS

tipc 010:
MIPSfast: Startzeit = 12:03:05
Endzeit = 12:03:09
Zeitdifferenz (Δt): 4s
MIPS-Rate = 419 MIPS

MIPSslow: Startzeit = 12:04:07
Endzeit = 12:04:34
Zeitdifferenz (Δt): 27s
MIPS-Rate = 62 MIPS

tipc 013:
MIPSfast: Startzeit = 12:48:25
Endzeit = 12:48:27
Zeitdifferenz (Δt): 2s
MIPS-Rate = 838 MIPS

MIPSslow: Startzeit = 12:49:16
Endzeit = 12:49:20
Zeitdifferenz (Δt): 4s
MIPS-Rate = 419 MIPS

Der Unterschied ist leicht zu erklären, er beruht auf der geringen Komplexität der Anweisung in MIPSfast.

Daher ist ein MIPS-Test nicht sehr gut geeignet, die Rechnergeschwindigkeit zu vergleichen, denn je nach Schwierigkeit der Befehle fiele so ein Test unterschiedlich aus.

Ein weiteres Manko ist, daß damit nicht die Geschwindigkeit des restlichen Systems gemessen werden kann.