

Übung 7 - Lösung

1. Multiplikation

Entwickeln Sie ein Programm welches folgenden Ausdruck berechnet:

$$\text{ERG} = (\text{ZAHL} * \text{MUL1}) * \text{MUL2}$$

- a) Erstellen Sie ein Flussdiagramm für den Algorithmus
- b) Erstellen Sie das Assembler Programm
- c) Testen Sie ihr Programm mit folgenden Zahlen:

ZAHL: 429496729 (dezimal)

MUL1: 22 (dezimal)

MUL2: 8 (dezimal)

Als ERG sollten Sie nun 75591424304 (dezimal), bzw. 1199999930 (hexadezimal) erhalten.

2. Division

Erweitern Sie das Program aus 1. Es sollen mit dem Ergebnis nun folgende weitere Berechnungen durchgeführt werden:

$$\text{ERG} = (\text{ZAHL} / \text{DIV1}) / \text{DIV2}$$

ZAHL entspricht hierbei dem Ergebnis aus 1.

- a) Erweitern Sie dazu ihr Flussdiagramm
- b) Erweitern Sie das Assembler Programm
- c) Testen Sie das fertige Program mit folgenden Werten:

ZAHL: 429496729 (dezimal)

MUL1: 22 (dezimal)

MUL2: 8 (dezimal)

DIV1: 2 (dezimal)

DIV2: 2 (dezimal)

Als ERG sollten Sie nun 18897856076 (dezimal), bzw. 46666664C (hexadezimal) erhalten.

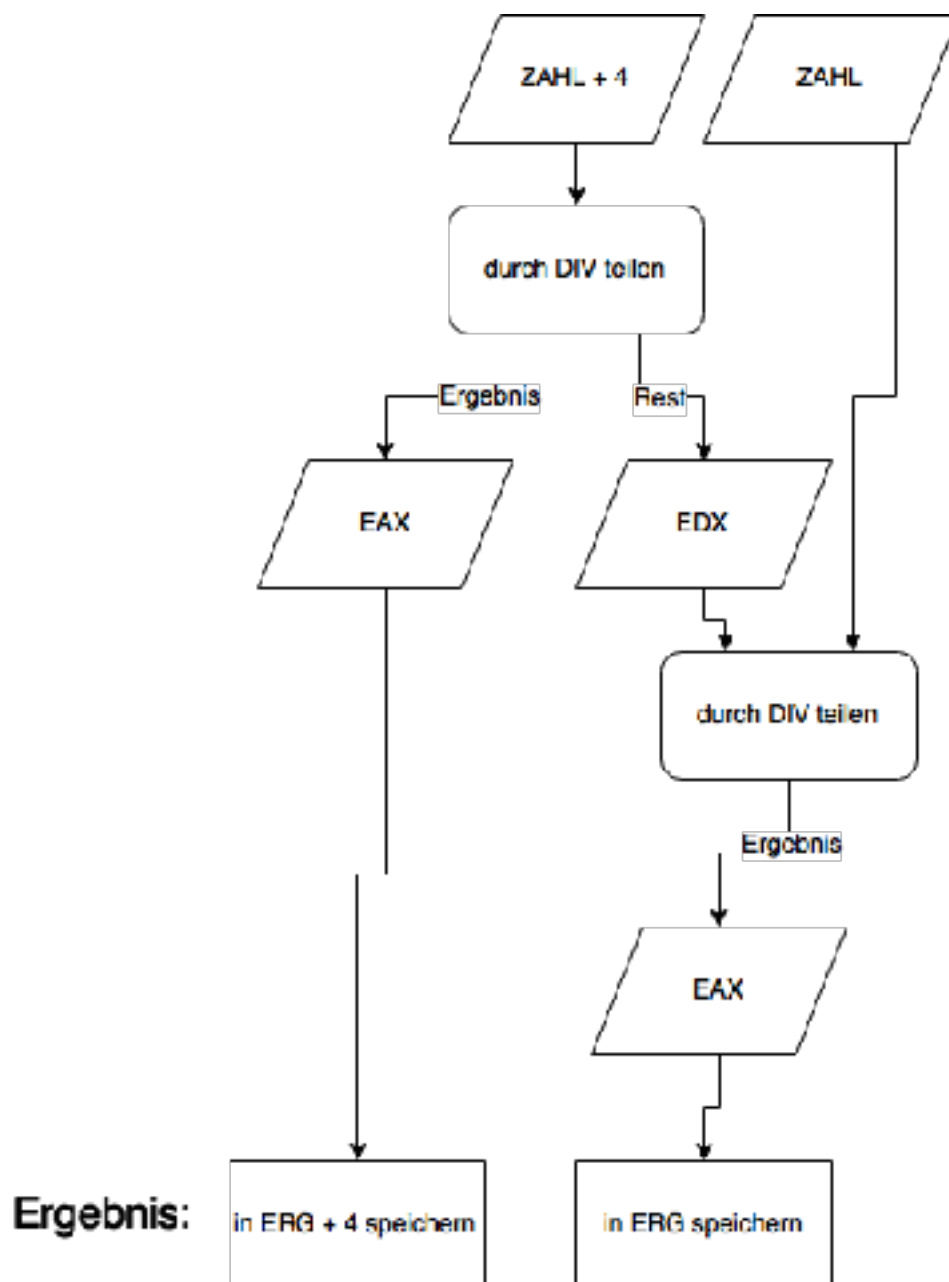
```
%include "io.inc"
section .data
    ERG DQ 0
    ZAHL DQ 429496729
    MUL1 DD 22
    MUL2 DD 8
    DIV1 DD 2
    DIV2 DD 2
    Save DD 0
section .text
global CMAIN
CMAIN:
    mov ebp, esp; for correct debugging
    ;write your code here
    xor eax, eax
    xor ebx, ebx
    xor ecx, ecx
    xor edx, edx

    ;Multiplizieren
    mov eax, [ZAHL]
    mul dword [MUL1]
    mov [Save], edx
    xor edx, edx
    mul dword [MUL2]
    mov [ERG], eax
    mov eax, [Save]
    mov [Save], edx
    xor edx, edx
    mul dword [MUL2]
    add eax, [Save]
    mov [ERG+4], eax

    ;Dividieren
    xor edx, edx
    mov eax, [ERG+4]
    div dword [DIV1]
    mov [ERG+4], eax
    mov eax, [ERG]
    div dword [DIV1]
    mov [ERG], eax

    xor edx, edx
    mov eax, [ERG+4]
    div dword [DIV2]
    mov [ERG+4], eax
    mov eax, [ERG]
    div dword [DIV2]
    mov [ERG], eax

    xor eax, eax
    ret
```

Ablauf Division:

Ablauf Multiplikation: