


Aufgabe 1: MIPS-Makrobefehle

a)


1. # Kopiert Wert von %rs nach %rd

```
.macro mov (%rd, %rs)
    add %rd, %rs, $zero
.end_macro
```




2. # Schiebt Wert %reg auf den Stack

```
.macro push (%reg)
    sw %reg, 0(%sp)
    subi %sp, %sp, 4
.end_macro
```




3. # Holt den obersten Wert vom Stack

```
.macro pop (%reg)
    add %sp, %sp, 4
    lw %reg, 0(%sp)
.end_macro
```




4. # %rd = %rs * %rt; %rd: least 32 bit

```
.macro mult (%rd, %rs, %rt)
    mult %rs, %rt
    mflo %rd
.end_macro
```




5. %rd = %rs / %rt

```
.macro div (%rd, %rs, %rt)
    div %rs, %rt
    mflo %rd
.end_macro
```




6. # %rd = %rs % %rt

```
.macro mod (%rd, %rs, %rt)
    div %rs, %rt
    mfhi %rd
.end_macro
```




7. # %reg = ~% reg

```
.macro not (%reg)
    nor %reg, %reg, %reg
.end_macro
```



8. # %reg = 0

```
.macro clear (%reg)
    xor %reg, %reg, %reg
.end_macro
```



9. # Rotiert %imm Bits nach rechts

```
.macro ror (%reg, %imm)
    srl $at, %reg, %imm
    sll %reg, %reg, 32-%imm
    or %reg, %reg, $at
.end_macro
```

Das geht nicht unbedingt



10. # Rotiert %imm Bits nach links

```
.macro rol (%reg, %imm)
    sll $at, %reg, %imm
    srl %reg, %reg, 32-%imm
    or %reg, %reg, $at
.end_macro
```



11. # if (%rs > %rt) goto % label

```
.macro bgt (%rs, %rt, %label)
    slt $at, %rt, %rs
    bne $at, $zero, %label
.end_macro
```



12. # if (%rs <= %rt) goto % label

```
.macro ble (%rs, %rt, %label)
    slt $at, %rt, %rs
    beq %at, $zero, %label
.end_macro
```



b)

In einer RISC Architektur werden die Makrobefehle von dem Compiler in Mikrobefehle übersetzt, nachdem der Sourcecode in Tokens unterteilt wurde.



Viele CISC-Architekturen haben außerdem komplexe Makrobefehle, die während der Laufzeit für die Ausführung von kleinen Mikroprogrammen sorgen. Dies kann in verschiedenen Stufen einer Pipeline passieren.



Mikroprogramme sind hardwarenäher und -orientierter als normale Assembly-Befehle.

Mikroprogramme kommen auch in RISC-Architekturen vor, werden aber üblicherweise mit CISC-Architekturen assoziiert.

3/3

Quellen

- Vorlesung
- <https://binaryterms.com/cisc-processors.html>
- <https://www.quora.com/Computer-Architecture-Does-only-CISC-instruction-set-system-have-microprogram>
- <https://github.com/netwide-assembler/nasm> Im Code kann man nachlesen, wann Macros ausgeführt werden.
- <http://what-when-how.com/microcontrollers/macro-assemblers-microcontrollers/>