

Übung 1

Abgabe bis **Donnerstag, 7. April 10:00** via EPIIC: <http://ep.iic.jku.at>.

1. Adressierungsarten (4 Punkte)

Verwende möglichst wenig Assembler Befehle um den Wert 0xAD mit Hilfe von

| | | |
|---------------------------------|---------|------|
| (a) unmittelbarer Adressierung, | address | data |
| (b) absoluter Adressierung, | 6 | 0xFF |
| (c) indirekter Adressierung, | 5 | 0xAD |
| (d) indizierter Adressierung | 4 | 0x01 |
| | 3 | 0xAB |
| | 2 | 0x43 |
| | 1 | 0x05 |
| in das Register \$s0 zu laden. | 0 | 0x10 |

2. Logische Befehle (6 Punkte)

Schreibe ein MIPS Programm, das die Permutation $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix}$ auf die vier Bytes des Werts in \$s0 anwendet. Verwende dazu ausschließlich logische Befehle und Rotationsbefehle, und teste deine Implementierung in *QtSpim* für \$s0 = 0xDEADBEEF.

3. Arithmetische Befehle (6 Punkte)

Übersetze $a = b * 19 + c * 17$; in ein MIPS Programm, das die Berechnung mit möglichst wenig Additionen, Rotationsbefehlen und logischen Befehlen ausführt. Andere Befehle sind nicht erlaubt. Nimm an, dass die Variablen a , b und c in den Registern \$s1, \$s2 und \$s3 liegen. Die Werte von b und c werden nach der Berechnung nicht mehr benötigt.

4. MIPS Programmierung (8 Punkt)

Das folgende C Programm konvertiert alle Kleinbuchstaben in Großbuchstaben. Übersetze das Programm in MIPS. Die Abgabe beinhaltet das kommentierte MIPS Programm und einen Screenshot von der Ausführung in *QtSpim*.

```
char str[] = "Computer Architecture 1!";
```

```
int main() {  
    for(char* s = str; *s != '\0'; s++) {  
        if(*s >= 'a' && *s <= 'z') {  
            *s += 'A' - 'a';  
        }  
    }  
}
```

| Char | Dezimal |
|------|---------|
| '\0' | 0 |
| 'a' | 97 |
| 'z' | 122 |
| 'A' | 65 |