

# ERA-Übungsblatt 02

2 b)

- xor <sup>dest</sup> a0, <sup>src1</sup> a0, <sup>src2</sup> a0
- addi a0, zero, 0
- add a0, zero, zero
- sub a0, a0, a0
- and a0, zero, a0
- ...

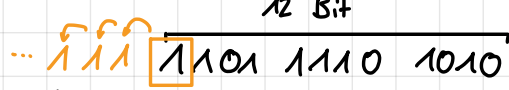
weil  $1 \text{ xor } 1 = 0$ ,  $0 \text{ xor } 0 = 0$   
... i : 12bit Immediat

weil  $0 \& ? = 0$

- li t1, 0xF000000
- lui t1, 0xF0000
- addi t1, t1, 0xF
- slli t1, t1, 28
- ...

(Achtung: Pseudo-Befehl)  
lui setzt die oberen 12 Bit auf 0

{ lui a3, 0xABAD1  
addi a3, a3, 0xDEA } funktioniert nicht!

addi sign-extended:  $0 \times \text{DEA} =$  ...   
sign-extension kopiert das oberste Bit an alle oberen 20 Bit!

Methode 1: auf  $0 \times \text{ABAD1}$  1 draufaddieren, Overflow der sign-extended Bits außerhalb des Registers

Methode 2:  $0 \times \text{ABAD1DEA} - 0 \times \text{FFFFFDEA} = ?$   
 $= 0 \times \text{ABAD1DEA} - (-0 \times 216)$  (Zweierkomplement)  
 $= 0 \times \text{ABAD2000}$

also:

- lui a3, 0xABAD2
- addi a3, a3, -534

$-534 = 0 \times \text{DEA}$  (unsigned hex mag der Assembler hier nicht)

3. a) durch lui a0, 0 werden die oberen 20 Bits auf 0 gesetzt und die unteren gelöscht (auch auf 0 gesetzt). D.h.  $a0 := 0$

- b)
- sub a0, a0, a1
  - addi a0, a0, 1024
  - addi t0, zero, 1
  - slli t0, t0, 11
  - add a0, a0, t0
  - slli a0, 1
  - addi a0, a0, 123
  - slli t0, a1, 2
  - add a1, t0, a1

$2^{11}$  hält als Immediat das Problem aus b)

$\rightarrow t0 := a1 \cdot 4$

4. siehe Musterlösung

5. dte) Bei Addition und Subtraktion hat das Ergebnis die gleiche Anzahl an Nachkommastellen wie die einzelnen Summanden (Subtrahend/Minuent). Bei der Multiplikation verschiebt sich das Komma: Anzahl Nachkommastellen Ergebnis =  $\sum$  Anz. Nachst. Multiplikanden

Beispiel: 4.4 Festkomma

Addition:

$$\begin{array}{r} 1010.0111 \\ + 0011.1000 \\ \hline 1101.1111 \end{array}$$

4 Nachkommast.

Multiplikation:

$$\begin{array}{r} 1001.1110 \cdot 11.0111 \\ \hline 1001.1110 \\ 1001.1110 \\ 00000000 \\ 1001.1110 \\ 1001.1110 \\ \hline 100001.11110010 \end{array}$$

4+4 = 8 Nachkommastellen!

→ um das Ergebnis der Multiplikation wieder im 4.4-Format zu bekommen, müssen Bits abgeschnitten werden (Shifts)

f) allgemeines: Wie kann Zahl  $x$  in eine ?-N-Festkommazahl umgewandelt werden?

Methode 1: Wie üblich in Binär umschreiben, nach dem Komma sind die Wertigkeiten der Binärstellen  $2^{-1} (= \frac{1}{2})$ ,  $2^{-2} (= \frac{1}{4})$ ,  $2^{-3} (= \frac{1}{8})$ , ...

Methode 2: Wir verschieben das Komma um  $N$  Stellen nach links: Wie im Dezimalsystem  $\cdot 10$ , so im Binärsystem  $\cdot 2$   
$$z = \lfloor x \cdot 2^N \rfloor$$