



Musterlösung 13. Gruppen- und Hausübung

Digitaltechnik und Rechnersysteme • Wintersemester 2022/2023

1 Gruppenübung

1.1 MIPS Maschinencode

- a) Der opcode für addi lautet 001000_2 . Mit $15_{10} = 01111_2$ und $16_{10} = 10000_2$ sowie $CAFE_{16} = 1100101011111110_2$ lautet der Maschinencode

001000 10000 01111 1100 1010 1111 1110

- b) 000000 00111 01111 00011 00000 100100

Da opcode=000000 handelt es sich um einen R-Befehl mit function= $100100_2 = 36$ und shamt= 00000. Dieser steht für die and operation. Die zugehörigen Register lauten rs=00111₂ = 7₁₀ rt=01111₂ = 15₁₀ und rd=00011₂ = 3₁₀. Dies führt zur Assembler Anweisung

and \$3, \$7, \$15

bzw.

and \$v1, \$a3, \$t7

1.2 MIPS Simulation

Programm	Registerinhalt			
	\$1	\$2	\$3	\$4
ori \$1,\$0,42	42	?	?	?
ori \$2,\$0,23	42	23	?	?
add \$3,\$1,\$2	42	23	65	?
and \$4,\$1,\$2	42	23	65	2
lui \$4,0x1001	42	23	65	0x10010000
sw \$1,4(\$4)	42	23	65	0x10010000
lw \$2,4(\$4)	42	42	65	0x10010000

$42_{10} = 101010_2$ $23_{10} = 010111_2$

1.3 MIPS Assembler aus C-Ausdruck

```

lui $1,0x1001    #load base address into s0
lw $2,8($1)      #load h into s1
addi $2,$2,5     #s1 <- s1 + 5
lw $3,4($1)      #load g into s2
add $3,$2,$3     #s2 <- s1 + s2
sw $3,0($1)      #store s2 into f

```

2 Hausübung

2.1 C-Ausdruck aus MIPS Assembler (4 Punkte)

Kommentierter Quellcode:

```
addi $3,$1,5      # $3 <- $1 + 5 = x + 5
mul  $2,$3,$1     # $2 <- $3 * $1 = x * (x + 5)
addi $2,$2,2      # $2 <- $2 + 2 = x * (x + 5) + 2 (=x^2+5x+2)
```

Daraus ergibt sich folgender C-Code:

```
y = x*(x+5)+2;
```

2.2 Fibonacci-Folge (6 Punkte)

```
or $1,$0,$0       #load base address into s0
lui $1,0x1001     #load base address into s0
ori $2,$0,1       #s1=1

sw $2,0($1)       #store f1
sw $2,4($1)       #store f2

ori $3,$0,0       #s3=i=0
ori $4,$0,28      #s4=loop limit-2

loop:
lw $2,0($1)       #get fi-2
lw $5,4($1)       #get fi-1
addu $6,$5,$2     #compute fi=s3=s2+s1
sw $6,8($1)       #store fi
addiu $1,$1,4     #i = i + 1

addiu $3,$3,1     #i=i+1
bne $3,$4,loop    #loop until loop limit reached
```