

## IHW 3

1.) Compare:

addi \$sp, \$sp, -4  
sw \$ra, 0(\$sp)  
jal sub  
li \$t0, 0  
bltz \$v0, exit+  
li \$t0, 1

Exit+:

mov \$v0, \$t0  
lw \$ra, 0(\$sp)  
addi \$sp, \$sp, 4  
jr \$ra

Sub :

sub \$v0 \$a0, \$a1  
jr \$ra

Stack = Comp function \$sp = 0x7fffffff8  
return address 0x7fffffff8

HW3

2. fib\_iter:

```
bne $a2, $0, else
    mov $a0, $a1
    jr $ra
```

else:

```
addiu $sp, $sp, -4
    sw $ra, 0($sp)
    mov $t0, $a0
    addu $a0, $a0, $a1
    nori $a1, $tc
    addiu $a2, $a2, -1
    jal fib_iter
    lw $ra, 0($sp)
    addiu $sp, $sp, 4
    jr $ra
```

Stack Contents assume fib\_iter is called with n=3

n	Address
3	0x7fffffc8
2	0x7fffffc4
1	0x7fffff0
0	0x7fffffec

### HW3

3.) a) func:

```
addiu $sp, $sp, -8  
sw $ra, 0($sp)  
sw $a2, 4($sp)  
jal func  
mov $a0, $v0  
lw $a1, 4($sp)  
jal func  
lw $ra, 0($sp)  
addiu $sp, $sp, 8  
jr $ra
```

b.) \$ra is equal to return address & \$sp = \$s3  
had when func was invoked.

4.) func:  
addiu \$sp, \$sp  
sw \$ra, 0(\$sp)  
sw \$a1, 4(\$sp)  
sw \$a2, 8(\$sp)  
jal ffunc  
lw \$a0, 4(\$sp)  
lw \$a1, 8(\$sp)  
sw \$v0, 4(\$sp)  
jal ffunc  
lw \$t0, 4(\$sp)  
addiu \$v0, \$t0, \$v0  
lw \$ra, 0(\$sp)  
addiu \$sp, \$sp, 12  
jr \$ra

HW3

S. func - Can i:

li \$t6, 0x30  
li \$t7, 0x39  
li \$v0, 0  
mov \$t9, \$a0  
lb \$t1, (\$t9)

Loop:

blt \$t1, \$t6, set  
bge \$t1, \$t7, set  
subu \$t1, \$t1, \$t6  
mul \$v0, \$v0, 10  
add \$v0, \$v0, \$t1  
addiu \$t9, \$t9, 1  
lb \$t1, (\$t9)  
bne \$t1, \$0, loop  
jr \$r9

Set:

li \$v0, -1  
jr \$r9

### MW 3

6. Hex-func:

```
li    $t4 0x41  
li    $t5 0x46  
li    $t6 0x30  
li    $t7 0x39  
li    $v0 , 0  
mov   $t0 , $a0  
lb    $t1 , ($t0)
```

Loop:

```
blt  $t1 , $t6 , set  
bge  $t1 , $t7 , set_  
subu  
j     half
```

Set\_hex:

```
blt  $t1 , $t4  
bge  $t1 , $t5  
addiu $t1 , $t1  
sll   $v0 ; $v0 , 4  
add   $v0 , $v0 , $t1  
lb    $t1 , ($t0)  
bne  $t1 , $0 , loop  
jr    $ra
```

Set:

```
li    $v0 , -1  
jr    $ra
```