CS536 HW9

Lingfei Li

11/28

Question 1:

- Assume the following layout of the memory:

(low address)
Control link
Return address
Param3
Param2
Param1
[Caller's AR]
(high address)

- Assume that the caller has already placed the parameters on the stack. The stack pointer points to one byte lower than param3.
- Assume the return address is stored in $ra by caller.
- Assume the return value needs to be stored in register $v0.
- Assume an integer takes 4 bytes.
- Results of computations are directly put to registers. No stack operations involved.

The MIPS code for the function is as follows:

```
#function entry:
sw $ra 0($sp)        #push return address
subu $sp $sp 4       #move stack pointer

sw $fp 0($sp)        #push control link (caller frame pointer)
subu $sp $sp 4       #move stack pointer

add $fp $sp 20       #set callee frame pointer (sp+8+sizeof(param))

#function body:
lw $t0 0($fp)        #load param a to $t0
lw $t1 -4($fp)       #load param b to $t1
add $t2 $t0 $t1      #calculate a + b
sw $t2 -8($fp)       #save the result of a+b to param c
li $t3 2             #save immediate value 2 to $t3
mult $t2 $t3         #calculate the result of c*2, save to $LO
mflo $v0             #save multiply result value from $LO to $v0


#function return:
lw $ra -12($fp)      #load return address to $ra
```

```
move $sp $fp         #restore caller’s stack pointer
lw $fp -16($fp)      #restore caller’s $fp from control link
jr $ra               #jump to return address
```

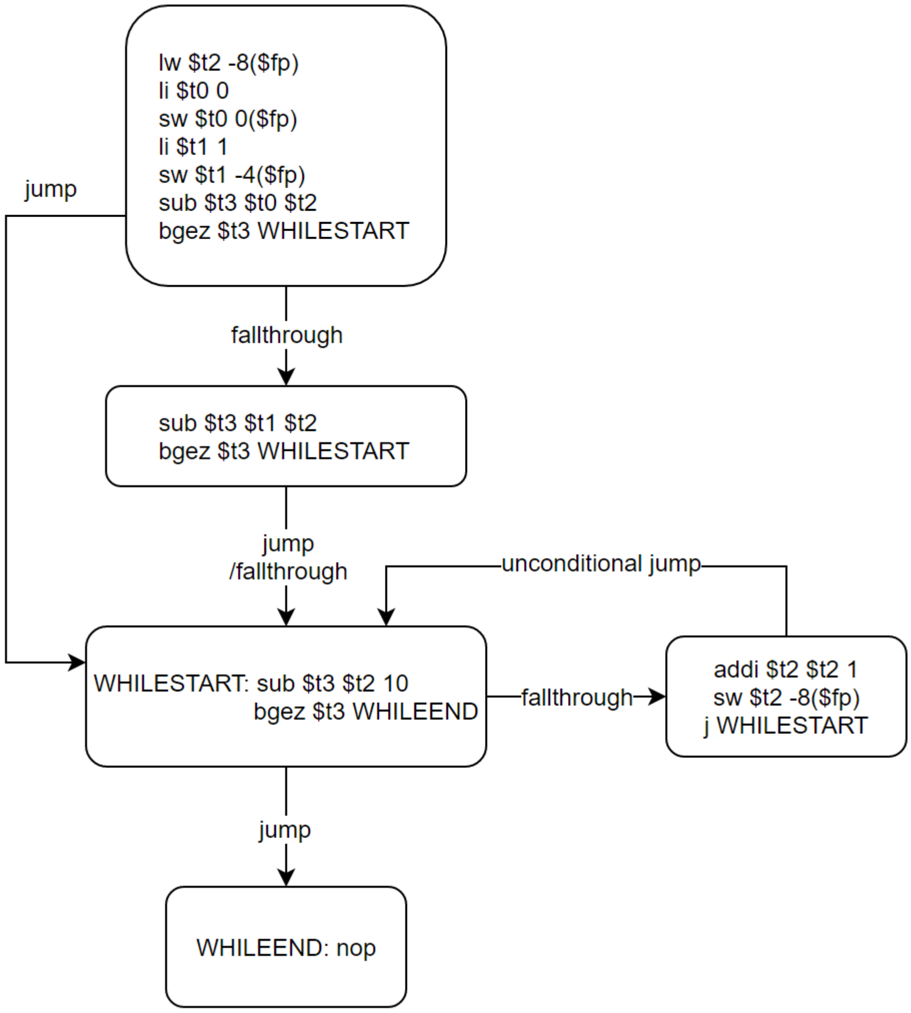
Question 2:

- Assume a, b, c are the 1st, 2nd, 3rd function parameters of integer type. Hence their addresses are 0, 4, 8 bytes lower than $fp respectively.

The MIPS code for the code snippet is:

```
             lw $t2 -8($fp)      #load c from 3rd parameter
             li $t0 0            #a
             sw $t0 0($fp)       #save a to memory
             li $t1 1            #b
             sw $t1 -4($fp)      #save b to memory
             sub $t3 $t0 $t2     #a-c
             bgez $t3 WHILESTART #if a-c >= 0, the condition fails
             sub $t3 $t1 $t2     #b<c <=> b-c < 0
             bgez $t3 WHILESTART #if b-c >= 0, the condition fails
WHILESTART:  sub $t3 $t2 10      #c-10
             bgez $t3 WHILEEND   #if c-10 >= 0, the condition fails
             addi $t2 $t2 1      #increment c
             sw $t2 -8($fp)      #save c to memory
             j WHILESTART
WHILEEND:    nop
```

The CFG is:



```
lw $t2 -8($fp)
li $t0 0
sw $t0 0($fp)
li $t1 1
sw $t1 -4($fp)
sub $t3 $t0 $t2
bgez $t3 WHILESTART
```

```
sub $t3 $t1 $t2
bgez $t3 WHILESTART
```

```
WHILESTART: sub $t3 $t2 10
            bgez $t3 WHILEEND
```

```
addi $t2 $t2 1
sw $t2 -8($fp)
j WHILESTART
```

```
WHILEEND: nop
```

Question 3:

$t0	6
$t1	4
$t2	8
$t3	undefined
$ra	0
PC	0