

**Question 1:** (0 points)  
Bank of Questions

## RISC-V Assembly

In the following two questions you will write two functions: `fun` and `maxfun`.

**Question 2:** (20 points)  
(V0C, V0D)

The first function is called `fun`. Given the value of an integer  $i$ , `fun` computes the value of an integer  $f_i$ , that is defined by the following equations:

$$\begin{aligned} f_0 &= 1 \\ f_1 &= 2 \\ f_i &= f_{i-2} + (-1)^i \times f_{i-1} \end{aligned} \tag{1}$$

Hint: another way to write the expression for  $f_i$  is as follows:

$$f_i = \begin{cases} 1 & \text{if } i = 0 \\ 2 & \text{if } i = 1 \\ f_{i-2} - f_{i-1} & \text{if } i \neq 1 \text{ and } i \text{ is odd} \\ f_{i-2} + f_{i-1} & \text{if } i \neq 0 \text{ and } i \text{ is even} \end{cases} \tag{2}$$

The specification for `fun` is as follows:

- **parameters:**  
a0:  $i$
- **return value:**  
a0:  $f_i$
- **guarantee:**
  - The value of  $i$ , all the intermediate values, and of  $f_i$  can be expressed as 32-bit integers.

Your implementation of `fun` must follow all the RISC-V calling conventions for saving/restoring registers.

**Solution:**

```
5  # parameter:      a0 = i
6  # return value:   a0 = fun_i
7  # register usage: s0: i
8  #                s1: fun_{i-1}
9  # guarantee: All intermediate values and the result are signed
10 #               integers that can be expressed in 32 bits.
11 fun:
12     addi    sp, sp, -12
13     sw      ra, 0(sp)
14     sw      s0, 4(sp)
15     sw      s1, 8(sp)
16     add     s0, zero, a0    # i <- a0
17     addi    a0, zero, 1     # a0 <- 1
18     beq     s0, zero, done  # if i == 0 goto done
19     addi    a0, zero, 2     # a0 <- 2
20     addi    t0, zero, 1     # t0 <- 1
21     beq     s0, t0, done    # if i == 1 goto done
22     addi    a0, s0, -1      # a0 <- i-1
23     jal     ra, fun
24     add     s1, zero, a0    # s1 <- fun_{i-1}
25     addi    a0, s0, -2      # a0 <- i-2
26     jal     ra, fun
27     andi    t1, s0, 1      # t0 <- 1 if i is odd; t0 <- 0 if i is even
28     beq     t1, zero, even
29     sub     a0, a0, s1      # a0 <- fun_{i-2} - fun_{i-1}
30     jal     zero, done
31 even:     add     a0, a0, s1  # a0 <- fun_{i-2} + fun_{i-1}
32 done:     lw      ra, 0(sp)
33           lw      s0, 4(sp)
34           lw      s1, 8(sp)
35           addi    sp, sp, 12
36           jalr    zero, ra, 0
```

Figure 1: A solution for fun.

**Question 3:** (20 points)

(V0B, V0C) The second function that you will write is **maxfun**. Given an integer  $k$ , **maxfun** returns the maximum value of  $f_i$  for interval  $[0, k]$ . The  $[ ]$  indicates that the limits of the interval are included in the computation of the maximum. To compute  $f_i$  **maxfun** must call the function **fun**. The specification for **maxfun** is as follows:

- **parameters:**

a0:  $k$

- **return value:**

a0: maximum value of  $f_i$  in the interval  $[0, k]$ .

- **guarantee**

– the value of  $f_i$  in all points for interval  $[0, k]$  can be expressed as a 32-bit integer.

**Solution:**

```
7  # maxfun: given a positive integer k, returns the maximum value of fun in [0,k]
8  # parameter:      a0: k
9  # return value:   a0: maximum value of fun in interval [0,k]
10 # guarantee:      k >= 0 and the value of fun_i fits into 32 bits for all i in [0,k]
11 # register usage: s0: k
12 #                  s1: max
13 #                  s2: i
14 #
15 maxfun:
16     addi    sp, sp, -16
17     sw      ra, 0(sp)
18     sw      s0, 4(sp)
19     sw      s1, 8(sp)
20     sw      s2, 12(sp)
21     add     s0, zero, a0      # s0 <- k
22     li      s1, 1            # max <- 1 (lowest value of f_i)
23     addi    s2, zero, zero    # i <- 0
24 for_i:     bgt     s2, s0, done # if i > k goto done
25           add     a0, zero, s2 # a0 <- i
26           jal     ra, fun
27           bgt     s1, a0, MaxOk # if max > fun_i goto MaxOk
28           add     s1, zero, a0  # max <- fun_i
29 MaxOk:     addi    s2, s2, 1    # i <- i+1
30           jal     zero, for_i
31 done:      addi    a0, s1, zero  # a0 <- max
32           lw      ra, 0(sp)
33           lw      s0, 4(sp)
34           lw      s1, 8(sp)
35           lw      s2, 12(sp)
36           addi    sp, sp, 16
37           jalr    zero, ra, 0
```

Figure 2: A solution to the `maxfun` function.