

## 5EIC0 - Computer Architecture 1

### Exercise RISC-V: Positive Power of Positive numbers

Write a RISC-V program that requests the user to input two non-negative numbers (zero is allowed) and the program computes the power of these numbers. The corresponding C program looks as follows:

```
1: int powpos(int a, int b)
2: {
3:     int c = b, d = 1;
4:
5:     while (c > 0) {
6:         d = d * a;
7:         c = c - 1;
8:     }
9:
10:    return d;
11: }
12:
13: void main(void)
14: {
15:     int a, b, c;
16:
17:     a = read_int();
18:     b = read_int();
19:
20:     c = powpos(a, b);
21:
22:     print_string("result = ");
23:     print_int(c);
24:     print_string("\n");
25: }
```

This program computes the positive power of a positive number. The main function from this C program can be translated into the following RISC-V assembler code.

```
1: .data
2: str1:    .ascii "result = \0"
3: str2:    .ascii "\n\0"
4:
5: .text
6:
7: .global main
8:
9: main:
10: addi sp,sp,-32
11: sw ra,28(sp)
12: sw s0,24(sp)
13: addi s0,sp,32
14: call read_int
15: mv s0,a0
16: call read_int
17: mv s1,a0
```

```
18: mv a0,s0
19: mv a1,s1
20: call gcd
21: mv s1,a0
22: la a0,str1
23: call print_string
24: mv a0,s1
25: call print_int
26: la a0,str2
27: call print_string
28: lw ra,28(sp)
29: lw s0,24(sp)
30: addi sp,sp,32
31: call show_pc
32: call exit
33: ret
34:
35: powpos:
36: ## put your code here ##
```

**Task 1.** Convert the powpos function to assembler and test the whole program using the SPIKE simulator. Obviously you should make sure that your RISC-V program produces exactly the same output as the above C program.