

Homework 03

1. (改编自 *COD_CH, P116, 2.9*)

对于下述每条 *RISC-V* 指令，写出操作码 (*op*)、源寄存器 (*rs1*) 和目标寄存器 (*rd*) 字段的值。对于 *I* 型指令，写出立即数字段的值，对于 *R* 型指令，写出第二个源寄存器 (*rs2*) 的值。对于非 *U* 型和 *UJ* 型指令，写出 *funct3* 字段，对于 *R* 型和 *S* 型指令，写出 *funct7* 字段。

```
addi x30, x10, 8
addi x31, x10, 0
sd    x31, 0(x30)
ld    x30, 0(x30)
add   x5,  x30, x31
```

2. (改编自 *COD_CH, P116, 2.11*)

假设寄存器 *x5* 保存值 128_{10} ，寄存器均为64位。

- 1). 对于指令 `add x30, x5, x6`，求出导致结果溢出的 *x6* 值的范围？
- 2). 对于指令 `sub x30, x5, x6`，求出导致结果溢出的 *x6* 值的范围？
- 3). 对于指令 `sub x30, x6, x5`，求出导致结果溢出的 *x6* 值的范围？

3. (改编自 *COD_CH, P118, 2.22*)

假设程序计数器 (*PC*) 置为 $0x20000000$ ，指令采用RV32I。

- 1). 使用 *RISC-V* 跳转-链接 (*jal*) 指令可以到达的地址范围是什么？（换句话说，跳转指令执行后 *PC* 的可能值是多少？）
- 2). 使用 *RISC-V* 的相等则分支 (*beq*) 指令可以到达的地址范围是什么？（换句话说，分支指令执行后 *PC* 的可能值是多少？）

4. (改编自 *COD_CH, P118-119, 2.27、2.28*)

1). 将以下循环转换为 *C* 代码。假设 *C* 语言级的整数(可以认为是int类型) *i* 保存在寄存器 *x6* 中，*x5* 中保存名为 *result* 的 *C* 语言级的整数，*x10* 保存整数 *MemArray* 的基址。

```
addi x6, x0, 0
addi x29, x0, 100
LOOP: ld    x7, 0(x10)
      add   x5, x5, x7
      addi x10, x10, 4
      addi x6, x6, 1
      blt  x6, x29, LOOP
```

2). 重写第一问中的循环以减少执行的 *RISC-V* 指令数。提示：注意变量 *i* 仅用于循环控制。

5. (改编自 *COD_CH*, P119, 2.35)

考虑以下代码：

```
lb x6, 0(x7)
sd x6, 8(x7)
```

假设寄存器 *x7* 包含地址 $0x10000000$ ，且地址中的数据是 $0x1122334455667788$ 。(寄存器位宽为 64)

- 1). 在大端对齐的机器上 $0x10000008$ 中存储的是什么值？
- 2). 在小端对齐的机器上 $0x10000008$ 中存储的是什么值？