2. TASKS FROM SweRVref

TASK: Open file *[RVfpgaPath]/RVfpga/src/SweRVolfSoC/SweRVEh1CoreComplex/include/swerv_types.sv* and analyse it during the next descriptions of the structure types that group together the control bits.

Solution not provided.

TASK: Take a quick look at modules **dec_decode_ctl** and **dec_dec_ctl** to see how the fields of the control signals are assigned based on the 32 bits of the instruction. These two modules are very extensive and quite complex, so the idea is not to analyse them in detail. Moreover, see that module **dec_dec_ctl** is created automatically as explained in lines 2482-2495 of *dec_decode_ctl.sv*.

Solution not provided.

TASK: Analyse the remaining instructions from the loop body in terms of compressed/uncompressed instructions.

```
92:  4398                 lw    a4,0(a5)
94:  0791                 addi  a5,a5,4
96:  0729                 addi  a4,a4,10
98:  fee7ae23             sw    a4,-4(a5)
9c:  fed79be3             bne   a5,a3,92 <main+0xa>
```

The first instruction is in its compressed format: `c.lw`

funct3	imm	rs1'	imm	rd'	op	CL-Type
3 bits	3 bits	3 bits	2 bits	3 bits	2 bits	

0x4398 = 010 000 111 00 110 00

- funct3 = 010 → c.lw
- imm = 00000
- rs1` = 111 → x15 = a5
- rd` = 110 → x14 = a4
- op = 00 → c.lw

The second instruction is in its compressed format, as we analysed in the SweRVref document.

The third instruction is in its compressed format: `c.addi`

funct3	imm	rd/rs1	imm	op	CI-Type

0x0729 = 000 0 01110 01010 01

- funct3 = 000 → c.addi

- rd/rs1 = 01110 → x14 = a4
- imm = 001010 → 10
- op = 01 → c.addi

The fourth and fifth instructions are in their uncompressed formats.

TASK: Take a look inside module **ifu_compress_ctl** and try to get an idea about how it works.

Solution not provided.