# 2. TAREFAS DO SweRVref

**TAREFA:** Abra o ficheiro *[RVfpgaPath]/RVfpga/src/SweRVolfSoC/SweRVEh1CoreComplex/include/swerv\_types.sv* e analise-o durante as próximas descrições dos tipos de estrutura que agrupam os bits de controle.

Solução não fornecida.

**TAREFA:** Dê uma leitura rápida nos módulos **dec\_decode\_ctl** e **dec\_dec\_ctl** para ver como os campos dos sinais de controlo são atribuídos com base nos 32 bits da instrução. Esses dois módulos são muito extensos e bastante complexos, portanto, a ideia é não analisá-los em detalhe. Além disso, veja que o módulo **dec\_dec\_ctl** é criado automaticamente, conforme explicado nas linhas 2482-2495 do *dec\_decode\_ctl.sv*.

Solução não fornecida.

**TAREFA:** Analisar as instruções restantes do corpo do ciclo em termos de instruções comprimidas/não comprimidas.

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>**

A primeira instrução está em seu formato comprimido: c.lw



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

A segunda instrução está em formato comprimido, conforme analisamos no documento SweRVref.

A terceira instrução está em formato comprimido: c.addi



0x0729 = 000 0 01110 01010 01

* + funct3 = 000🡪 c.addi
  + rd/rs1 = 01110🡪 x14 = a4
  + imm = 001010🡪 10
  + op = 01🡪 c.addi

A quarta e a quinta instruções estão em formatos não comprimidos.

**TAREFA:** Inspecione o módulo **ifu\_compress\_ctl** e tente ter uma ideia de como ele funciona.

Solução não fornecida.