True or False.

* \_F\_From generation to generation, the number of the registers inside a CPU increases exponentially as Moore’s law predicted.
* \_F\_Let a0 point to the start of an array x (considering all data types: word, char, half word). lw s0, 4(a0) will always load x[1] into s0.
  + False. This only holds for data types that are four bytes wide, like int or float. For data-types like char that are only one byte wide, 4(a0) is too large of an offset
* \_T\_PC-relative addressing: Uses the PC and adds the immediate value of the instruction to create an address (used by branch and jump instructions)

**Short answer questions:**

1. For the following C statement, write the corresponding RISC-V assembly code.

Assume that the C variable f,g, and h, have already been placed in register x5,x6, and x7 respectively. Use a minimal number of RISC-V assembly instructions.

f = a + (h - 5);

A picture containing chart

Description automatically generated

1. Write a single C statement that corresponds to the two RISC-V assembly instructions below.

add f, g, h

add f, i, f

Diagram

Description automatically generated

1. Provide the instruction type and hexadecimal representation

of the following instruction:

SW x5, 32 (×30)



**Essay question:**

1. Translate the following C code to RISC-V. Assume that the variables f,g, h, i, and j are assigned to registers x5, x6, x7, x28, and x29, respectively. Assume that the base address of the arrays A and B are in registers ×10 and x11, respectively. Assume that the elements of the arrays A and B are 8-byte words:

B[8] = A[i] + A[j];

Table

Description automatically generated