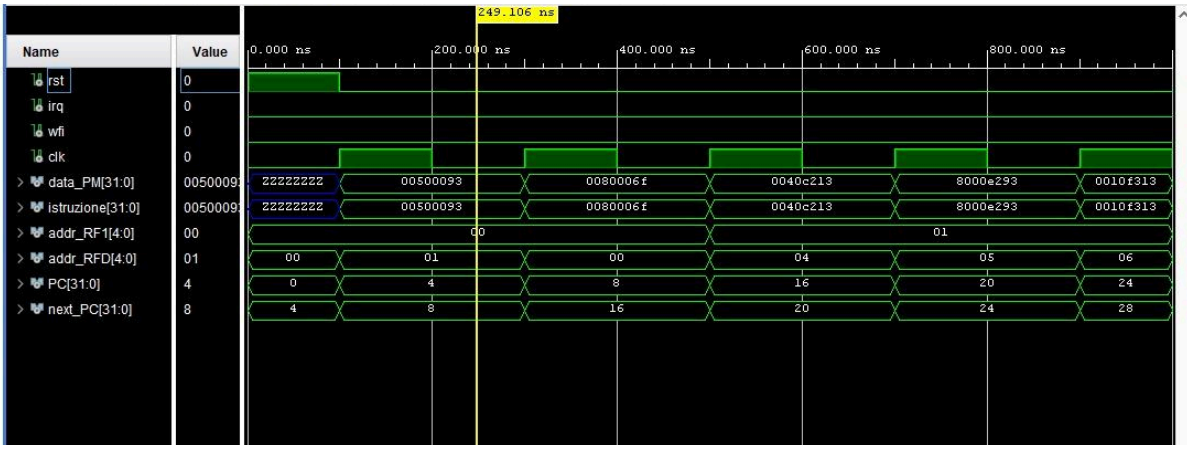
# Compilation and simulation

In the following image, we see a simulation of the testbench:




It's important to know that in order to correctly function, our processor needs to execute a NOP as the first instruction, so the simulation starts with a PC of 0 and an increment of 4 for the computation of the next program counter (next_PC).
In the figure it is possible to see that at every rising edge of the clock signal the core fetches a new instruction by acquiring the data coming from the program memory and placing it in the "istruzione" signal. It then decodes the instruction by analyzing the bits of istruzione. At the same time the core updates the current program counter in order to prepare the fetch for the next instruction.
By observing the figure it's possible to see the instructions fetched: for example, after the first clock cycle, "istruzione" has the hexadecimal value "00500093" that correspond to the ADDI instruction placed in the second position of the program memory and the program counter has in fact the value 4. In this case the next program counter is correctly computed as 4+4= 8.
The following instruction is "0080006f" (JAL) with the current program counter being 8 and then incremented by an offset of 8 as written in the instruction.

In the following figure it's possible to notice how an WFI instruction makes the core stop the usual execution and it makes it wait for an IRQ signal. In particular the program counter is updated just once and then left unchanged in order to have it ready whenever an interrupt arrives.