Microwatt is an open-source POWER ISA softcore written in VHDL.
https://github.com/antonblanchard/microwatt

Microwatt can run bare-metal C programs either in simulation mode using GHDL or on an FPGA.

Step 1. Installing Microwatt with GHDL simulator:

Requirements: LLVM, Clang, and GHDL with LLVM backend

If you don't have clang and/or LLVM installed. Install them using the commands below

```
sudo apt-get install clang
sudo apt-get install llvm
```

Install GHDL with LLVM backend as shown here:

https://ghdl.github.io/ghdl/development/building/LLVM.html

Install micropython and microwatt as shown here:

https://github.com/antonblanchard/microwatt#simulation-using-ghdl

Step 2. Compiling and running a bare-metal C program:

Microwatt supports simulation of bare-metal C programs in POWER ISA

For ex. the instructions below show how to run hello_world.c from microwatt/hello_world directory

```
user:/microwatt$ cd hello_world
```

Compile the code:

```
user:/microwatt/hello_world$ make

powerpc64le-linux-gnu-gcc -Os -g -Wall -std=c99 -msoft-float -mno-string -mno-multiple -mno-vsx -mno-altivec -mlittle-endian -fno-stack-protector -mstrict-align -ffreestanding -fdata-sections -ffunction-sections -I../include   -c -o hello_world.o hello_world.c

powerpc64le-linux-gnu-gcc -Os -g -Wall -std=c99 -msoft-float -mno-string -mno-multiple -mno-vsx -mno-altivec -mlittle-endian -fno-stack-protector -mstrict-align -ffreestanding -fdata-sections -ffunction-sections -I../include   -c -o head.o head.S

powerpc64le-linux-gnu-gcc -Os -g -Wall -std=c99 -msoft-float -mno-string -mno-multiple -mno-vsx -mno-altivec -mlittle-endian -fno-stack-protector -mstrict-align -ffreestanding -fdata-sections -ffunction-sections -I../include -c ../lib/console.c -o console.o

powerpc64le-linux-gnu-ld -T powerpc.lds -o hello_world.elf hello_world.o head.o console.o

powerpc64le-linux-gnu-objcopy -O binary hello_world.elf hello_world.bin

../scripts/bin2hex.py hello_world.bin > hello_world.hex

user:/microwatt$ rm -rf main_ram.bin

user:/microwatt$ ln -s ./hello_world/hello_world.bin main_ram.bin
```

Run the compiled binary

```
user:/microwatt$./core_tb > logs_hello_world
```

`logs_hello_world` contains the logs for the instructions executed along with the warnings.

You can remove the warnings and store the simulation information of the instructions in a separate file.

For example, the command below stores all lines from logs_hello_world that do not have the text “assertion” in `instruction_logs`

```
User@/microwatt$ sed '/assertion/d' ./logs_hello_world > instruction_logs
```