For Windows

Please use HackMD to view the file.

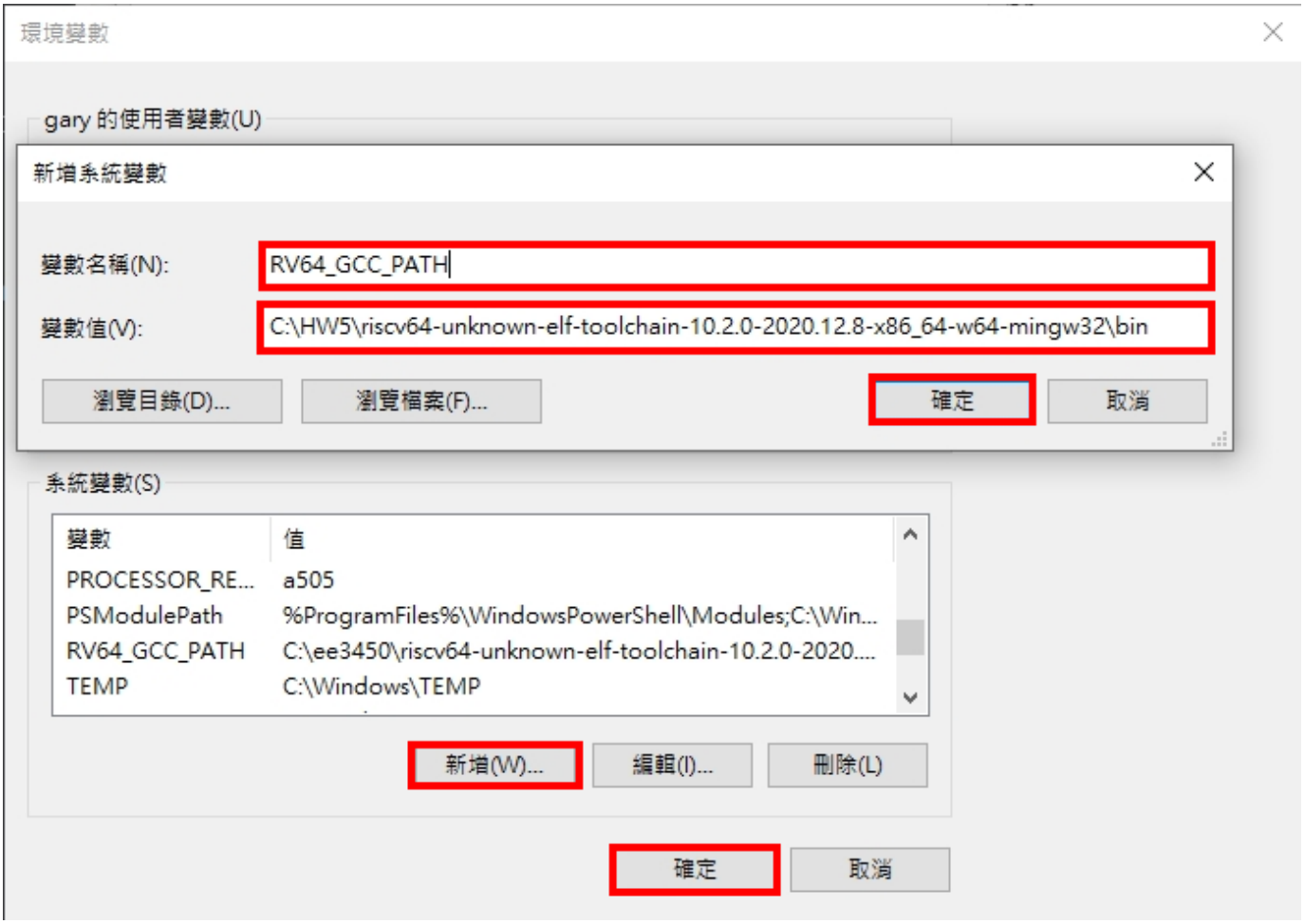
Install Ripes

1. Please download the release version of Ripes from Ripes Release build.
   `Ripes-v2.2.4-win-x86_64.zip` is the current version for Windows platform.
2. Please extract the file.
   Please right click and extract the file. Here the file was extracted in `C:\HW5`.

Install Toolchain

1. Please download the SiFive RISC-V toolchain from https://github.com/sifive/freedom-tools/releases.
   For example, `riscv64-unknown-elf-toolchain-10.2.0-2020.12.8-x86_64-w64-mingw32.zip` is the current version for Windows.
2. Please extract the file to a folder.
   Please right click and select extract on windows. At the location you extract the Ripes, for example `C:\HW5`.
3. Please find out from the extract folder where `riscv64-unknown-elf-gcc.exe` is. In this example the file location is `C:\HW5\riscv64-unknown-elf-toolchain-10.2.0-2020.12.8-x86_64-w64-mingw32\bin`.
4. This folder is denoted as `$env:RV64_GCC_PATH` below. You may use following steps to setup environment variable: `$env:RV64_GCC_PATH=C:\HW5\riscv64-unknown-elf-toolchain-10.2.0-2020.12.8-x86_64-w64-mingw32\bin`.
   (1)Please set the environment variables on your computer

(2)Add a new variable with the name of `RV64_GCC_PATH` and the value of `C:\HW5\riscv64-unknown-elf-toolchain-10.2.0-2020.12.8-x86_64-w64-mingw32\bin`



(3)Please shut down your computer and reopen it to activate the environment variable.
Please note that Reboot will not work !

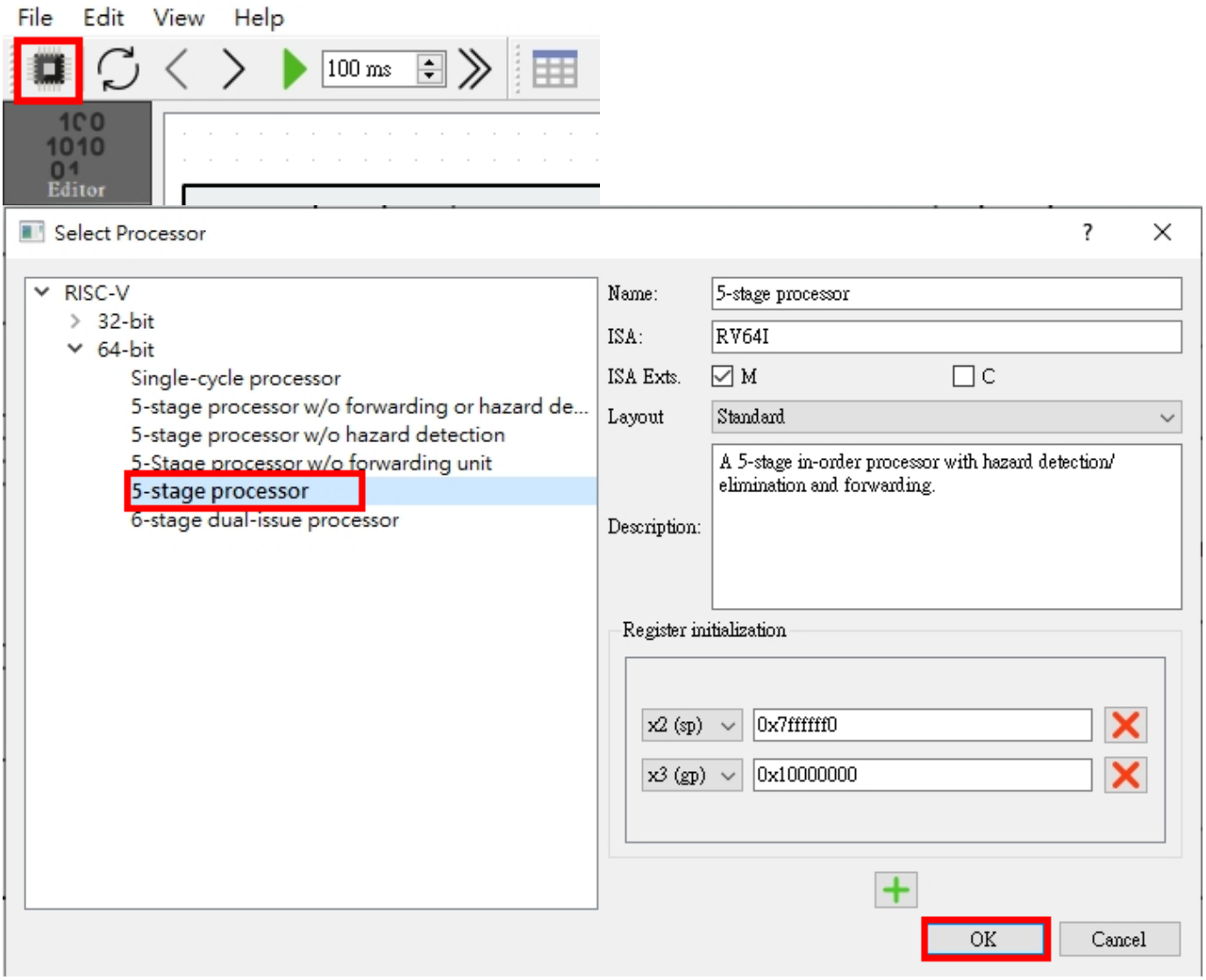
Test Set

1. Download cmul.S from HW5.
2. Start a Powershell.
3. Go to the download folder with cmul.S, e.g., `cd C:\HW5\`.
   Before compiling the file, please double check $RV64GCCPATH is set with the folder location with riscv64-unknown-elf-gcc.
4. Generate an rv64im executable, cmul.elf in this case:

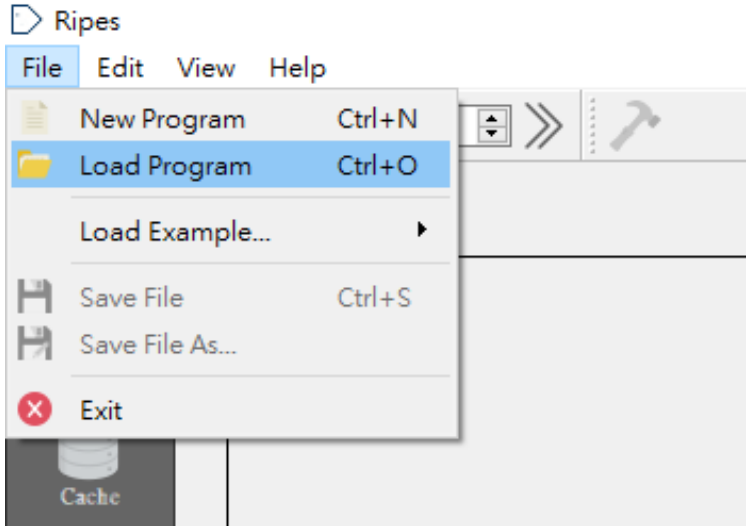
```
>& $env:RV64_GCC_PATH\riscv64-unknown-elf-gcc.exe -march=rv64im -mabi=lp64 -s -static -nostdlib -o cmul.elf cmul.S

"-march=rv64im" : to use an ISA version with 64-bit architecture and integer ("i") and multiply ("m") supports.
"-mabi=lp64" : to specify the language data model. In this setup, long ("l"), and pointer ("p") are all 64-bit.
"-s" : to strip symbols from binary.
"-static" : to link statically to produce a complete executable.
"-nostdlib" : do no use stdlib.
"-o" : specify output name.
```

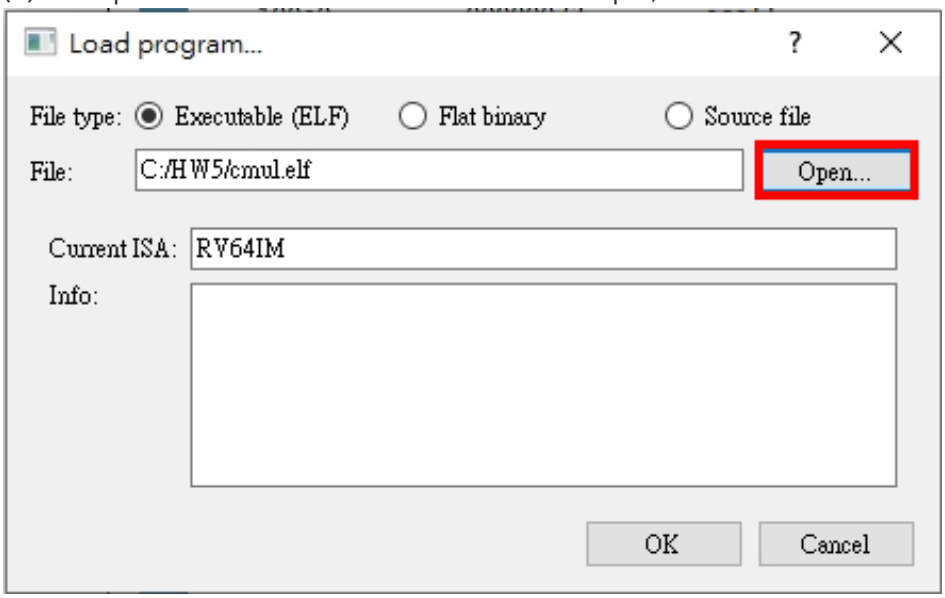
5. Start Ripes GUI by double clicking C:\HW5\Ripes-v2.2.4-win-x86_64\Ripes.exe
6. Click on `processor selection button` at top-left and choose Processor RISC-V 64-bit 5-stage processor



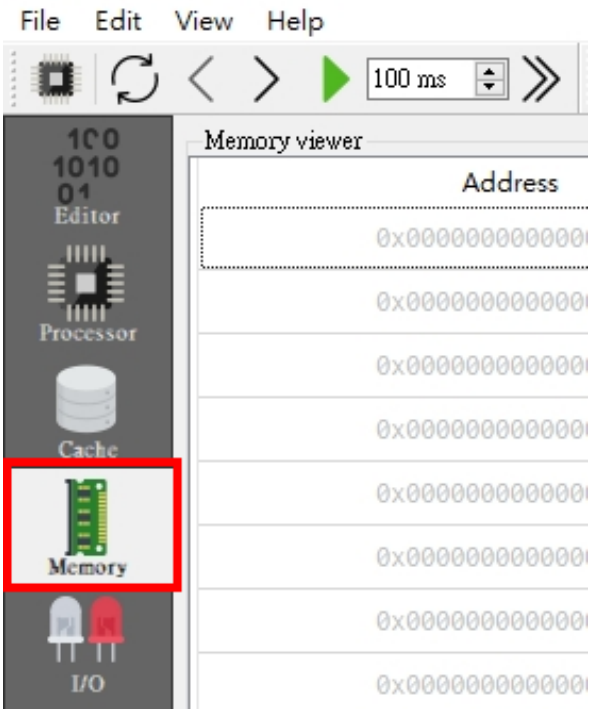
7. Load the rv64im executable
   (1)File -->Load Program



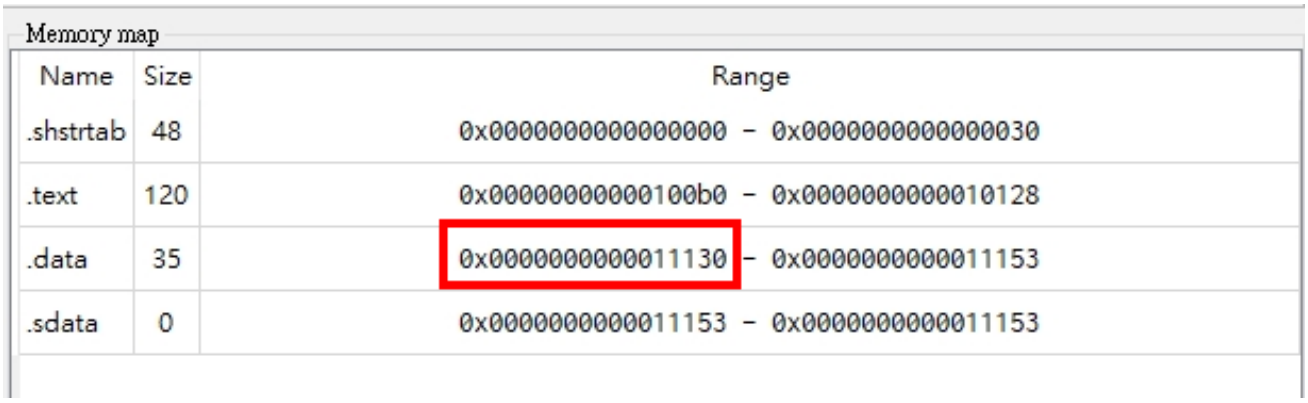
(2)Use Open to search for cmul.elf. In our example, cmul.elf is in C:/HW5/.



8. After loading the elf file, we need to setup global pointer. First, check the current `global pointer(gp)`
   (1)Select Memory Tab.



(2)In Memory Tab we check the `data segment base memory address` on the right.



Here we find that the base is 0x0000000000011130.

Since the assembler assume the global pointer (x3) to be set at the base + 0x800,

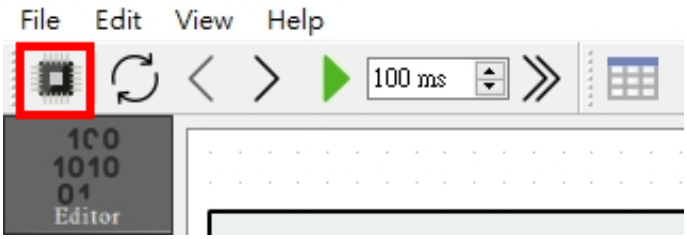
we will use 0x0000000000011130+0x800=0x0000000000011930 to setup global pointer.

9. Setup global pointer(gp)

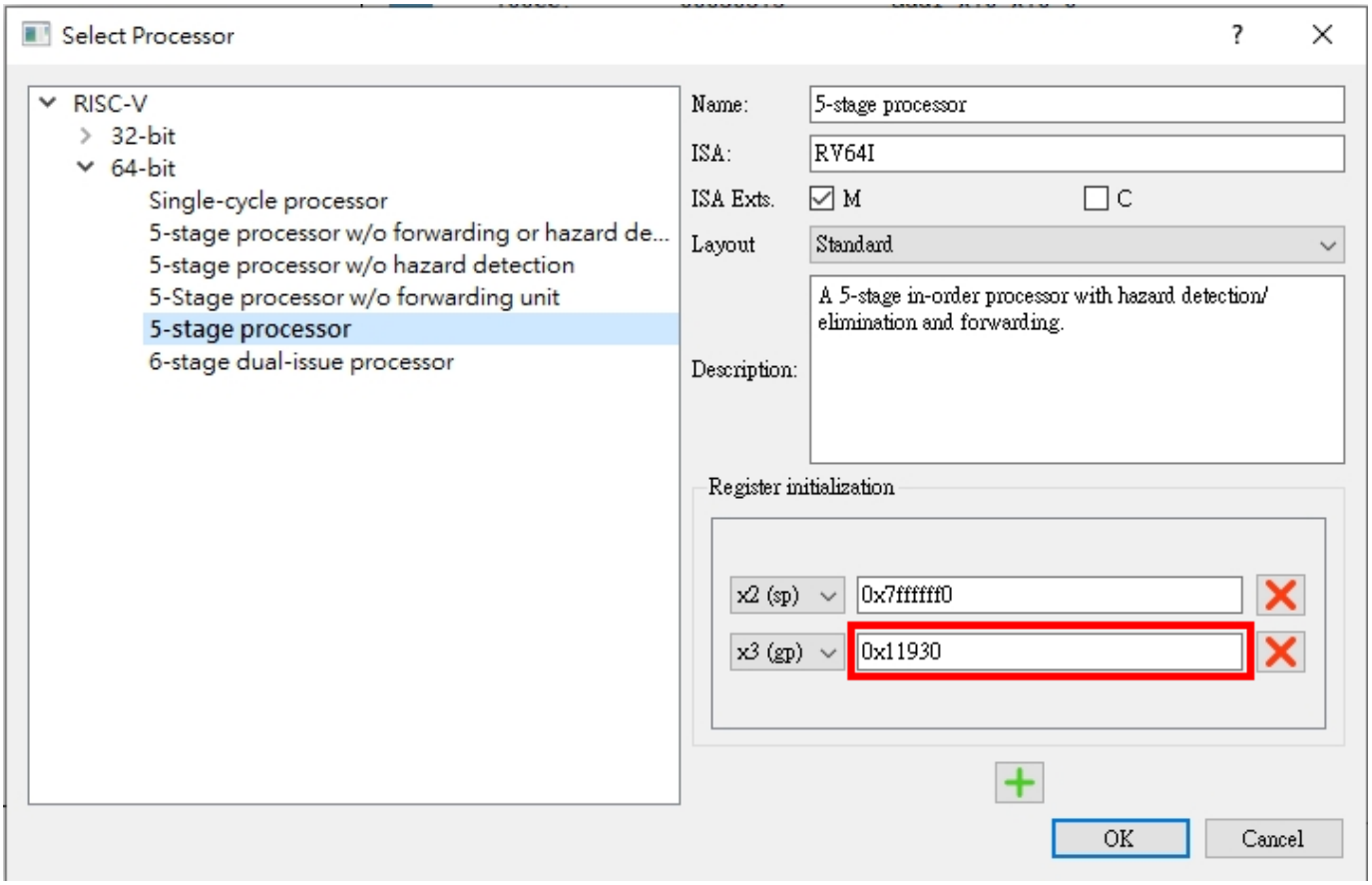
   There are two methods,

   Method1

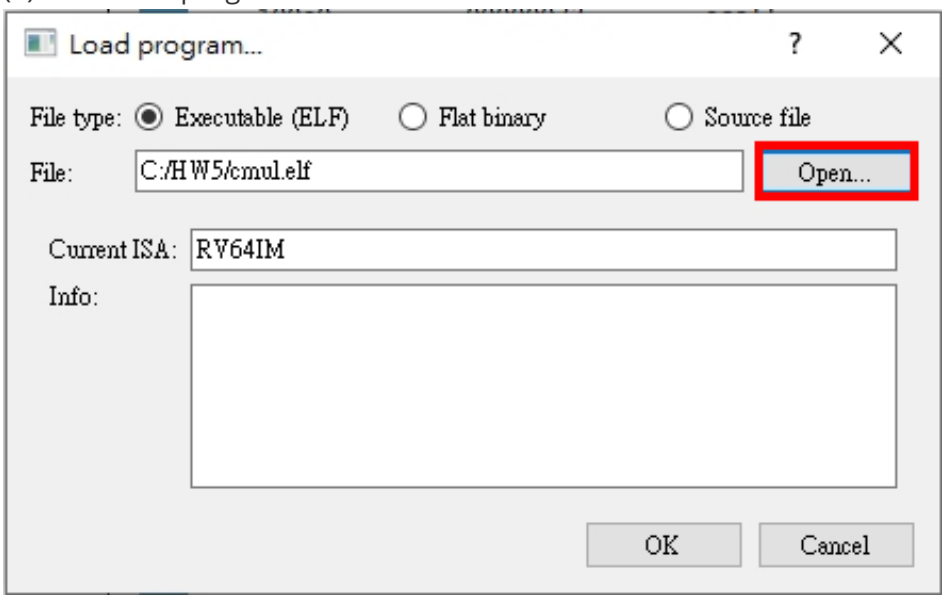
   (1)Click on `processor selection button` again



   (2)Fix the value directly

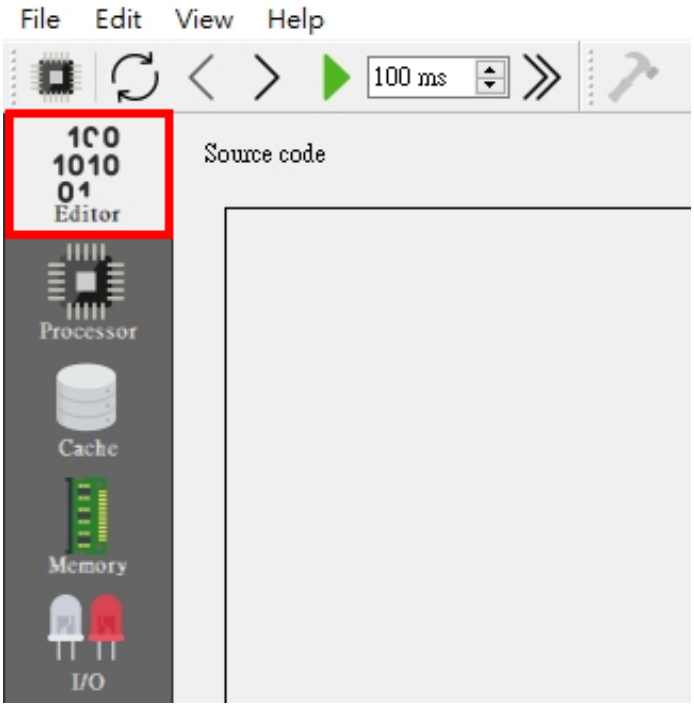


(3)Reload the program

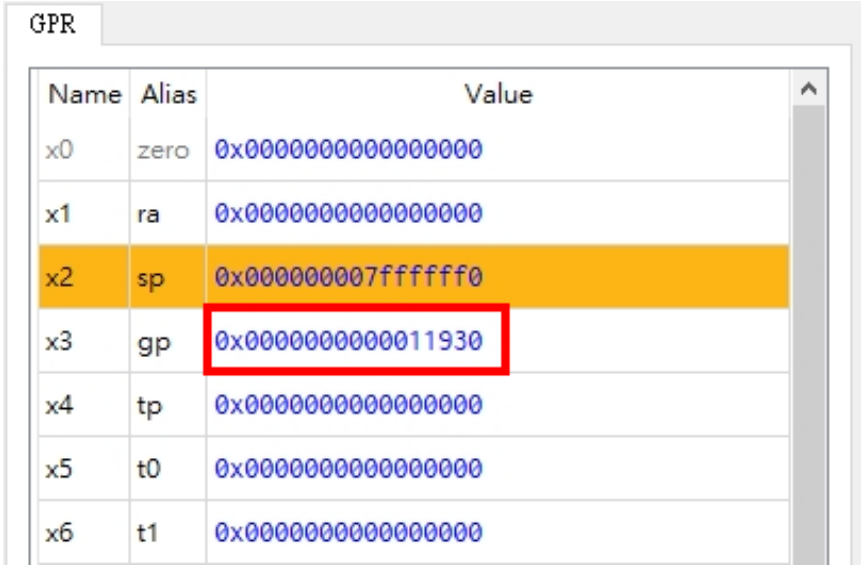


Method2

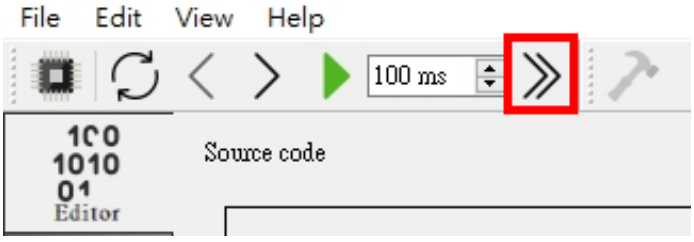
(1)Select Editor Tab.



(2)Fix the value of gp on the right side by clicking on the value of gp.



10. Click `>> button` on top-right to run and simulate the binary in Ripes without GUI updates.



11. We should see the program finishes and print `11 + i* 17` on the console at the bottom

Note that if we need to set the global pointer for each new elf program load (since the data segment changes according to the text segment).