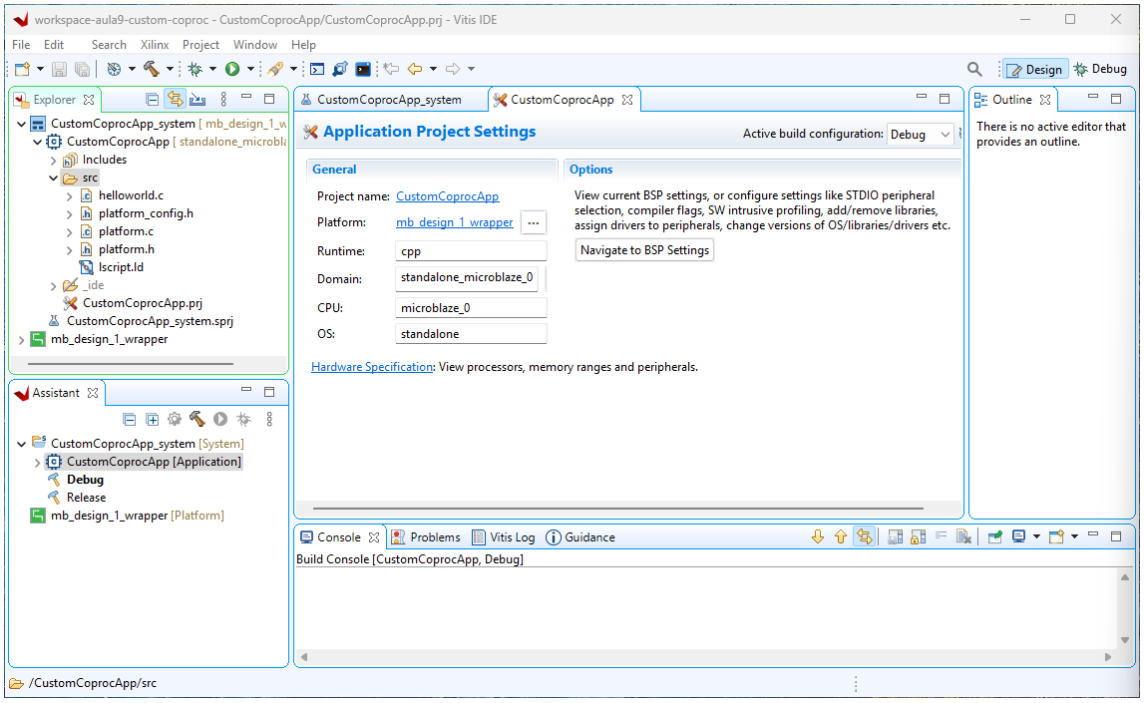
Tutorial 4

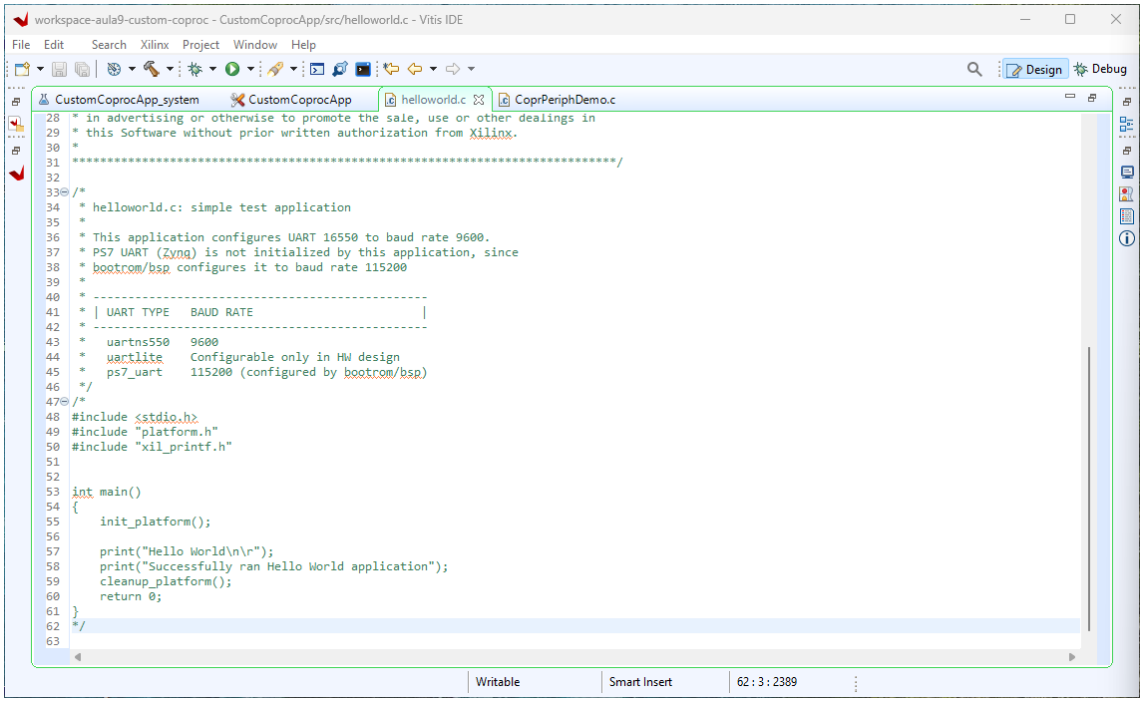
Update MicroBlaze Platform with a Custom Coprocessor (3-register adder) connected via AXI-Lite – SW Project (VITIS)

Vivado version: 2022.2

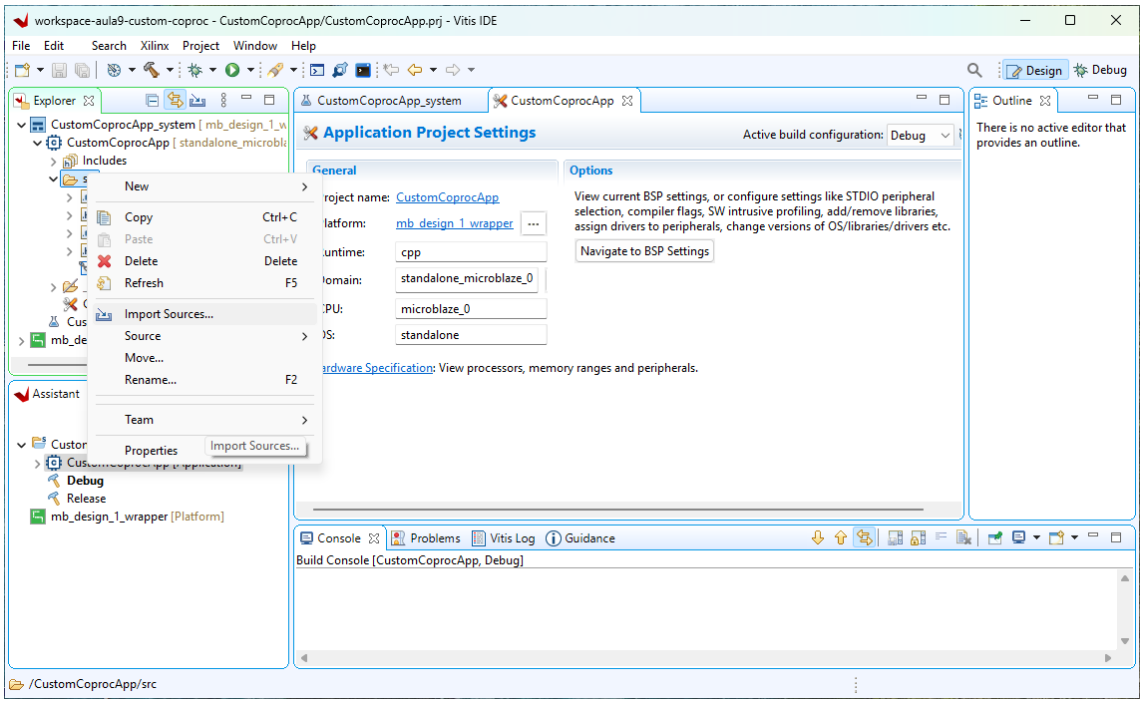
1. Open **Vitis**. Create a new application project. Import the XSA of the project you just created in **Vivado**. The application project template can be an empty C++ project or, if in doubt, the **Hello World** application template. Name the application at your discretion (e.g., **CustomCoprocApp**).



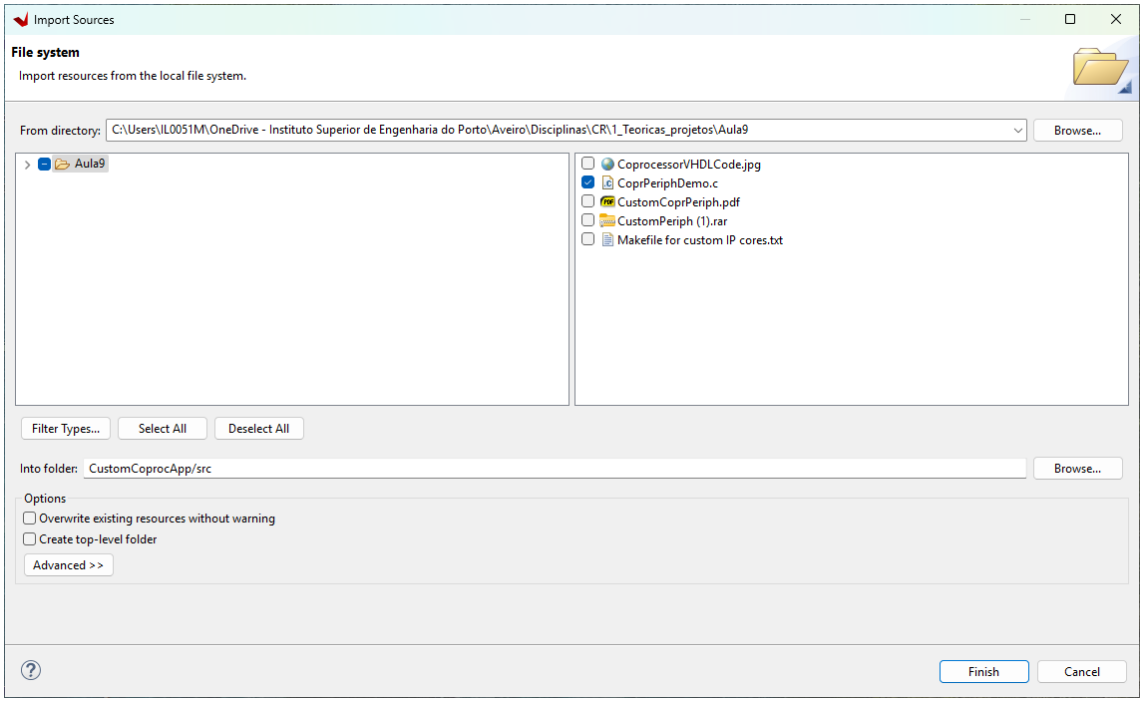
2. In case you selected the **Hello World** application template, open the **helloworld.c** file and comment it fully.



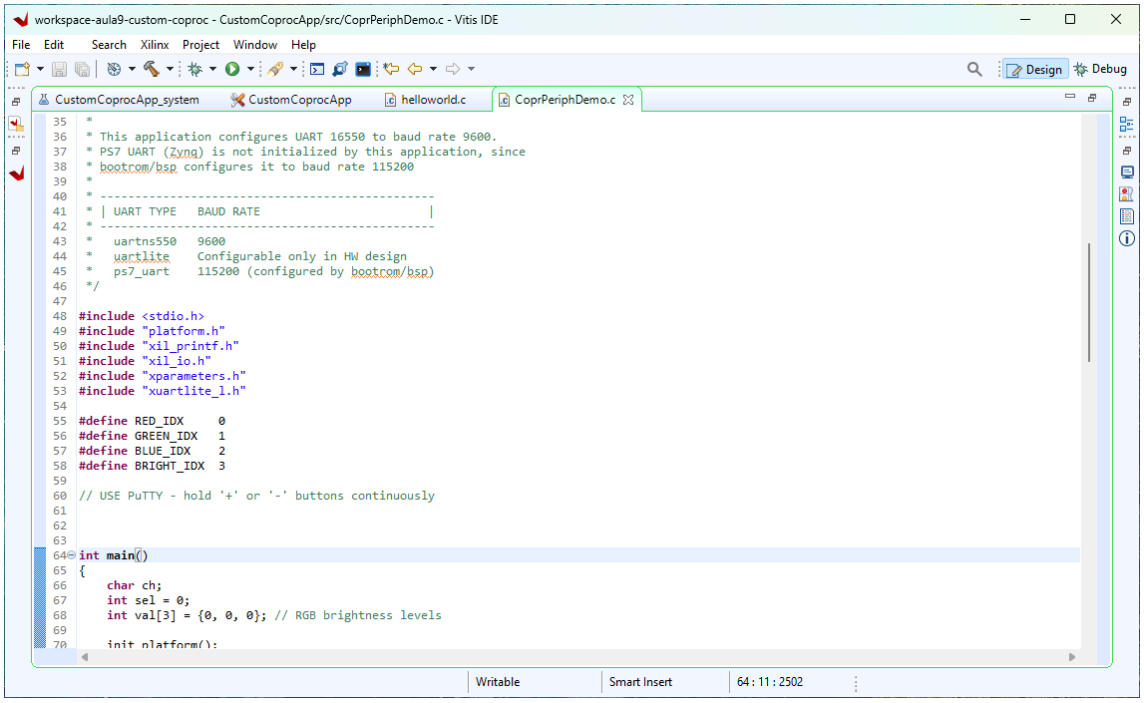
3. Click on **Import Sources**, on the **src** folder.



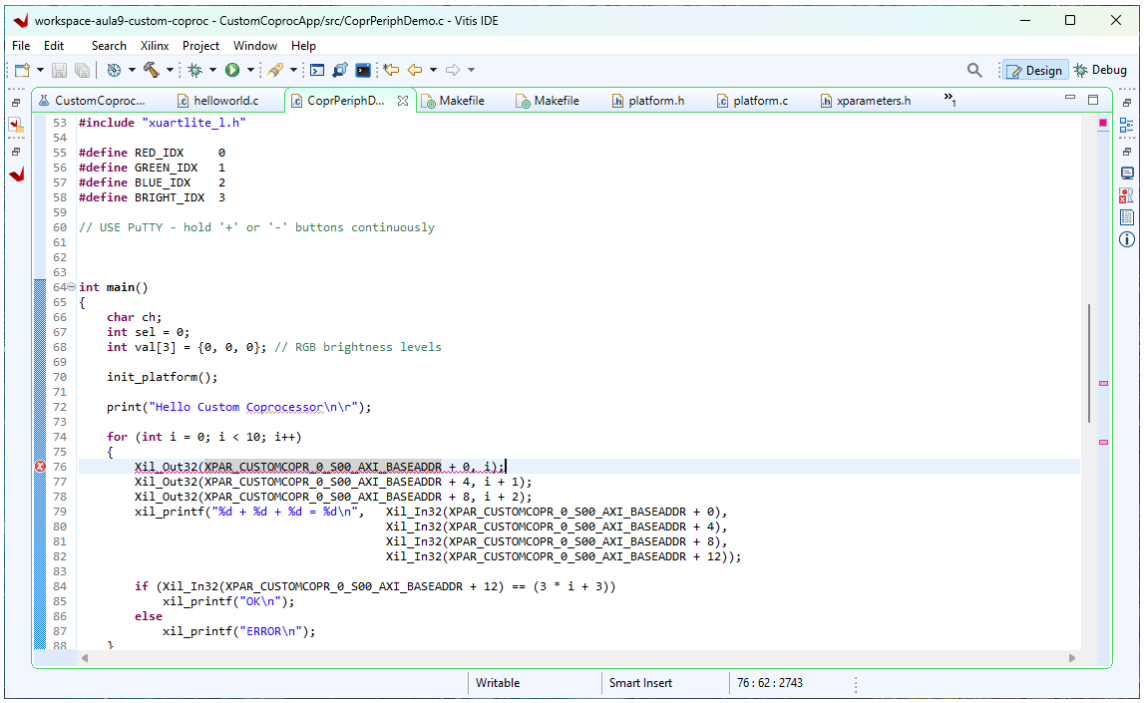
4. Import file **CoprPeriphDemo.c** – available in the Moodle.



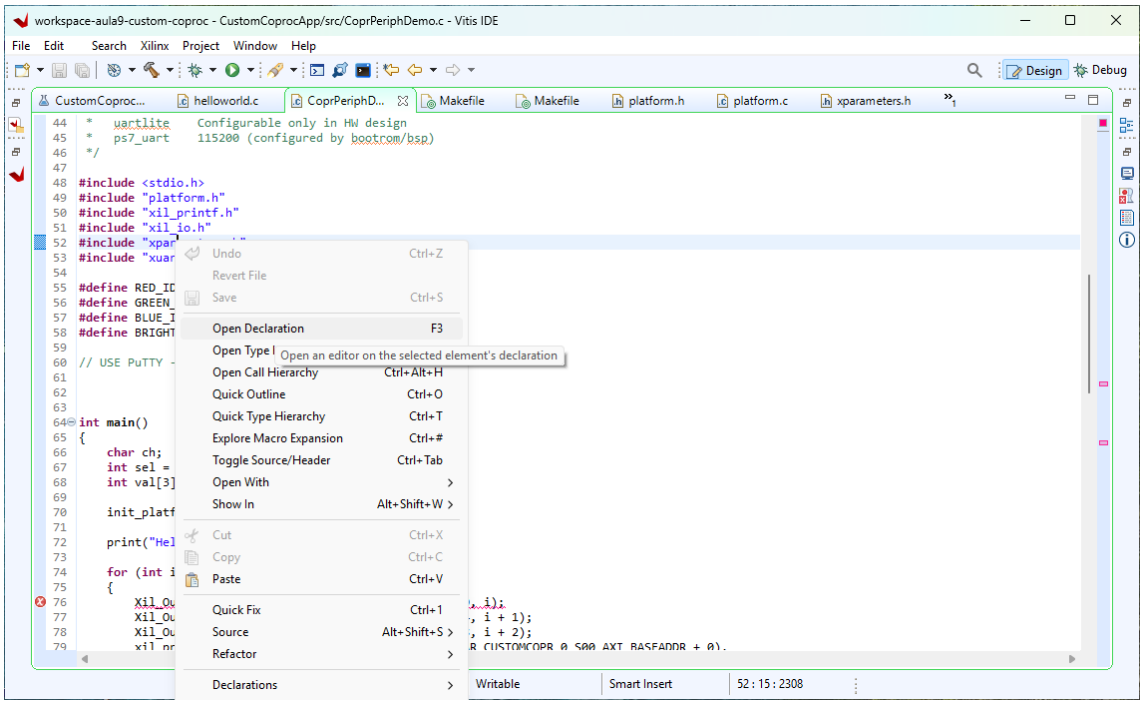
5. Confirm that the file has been imported into the project.



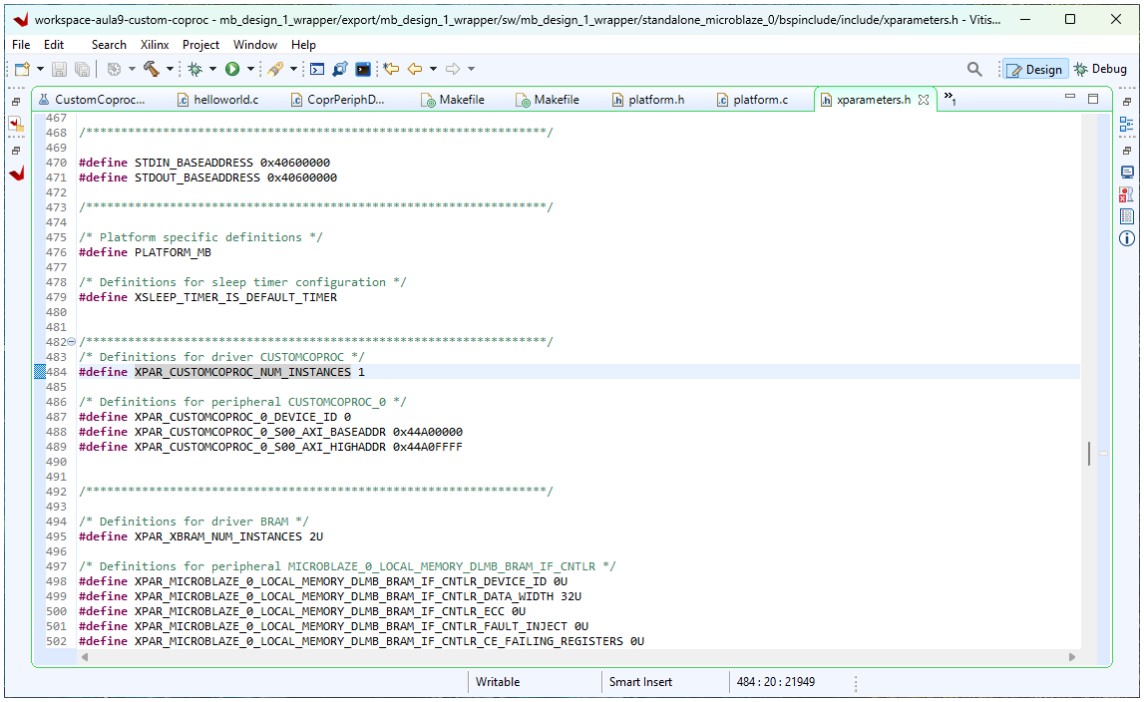
6. There may be errors flagged in the file, concerning macros that represent the co-processor parameters (e.g., base address). This occurs because this example file assumes a name for the co-processor that may not match the one you choose.



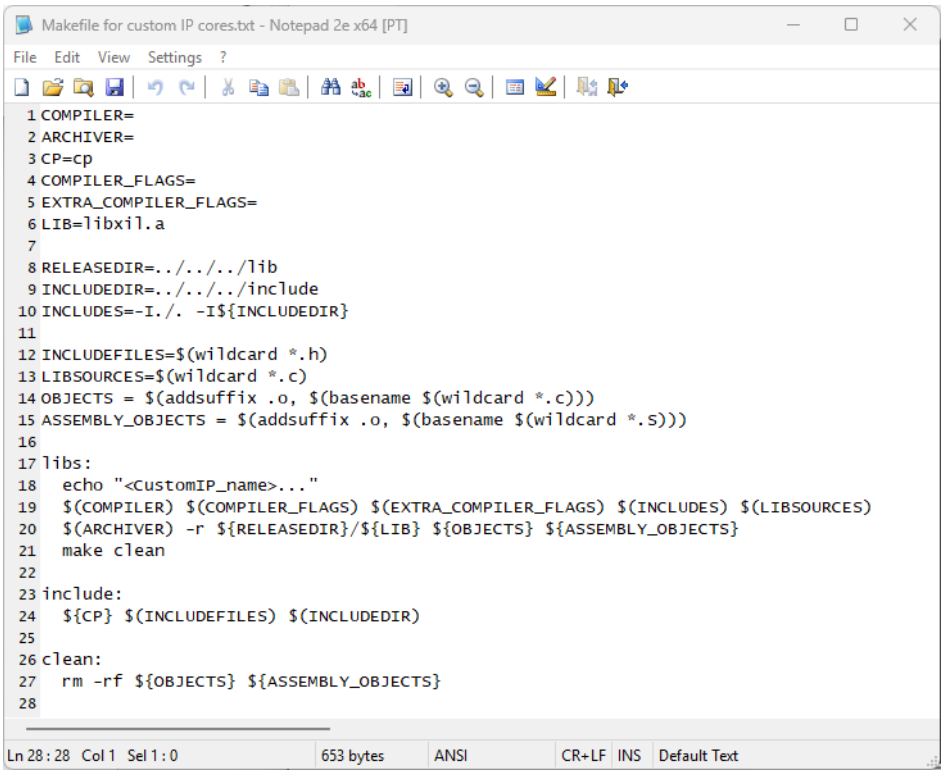
7. Right-click on **#include** "xparameters.h", and select **Open Declaration**.



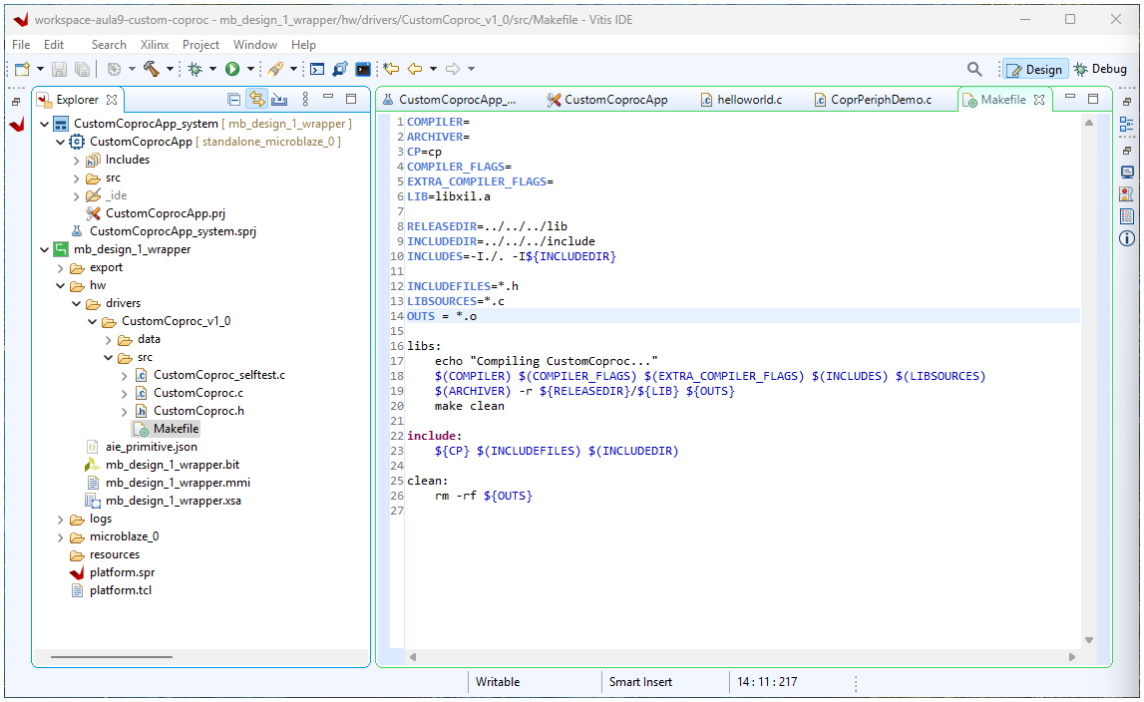
8. Navigate the file until you find the right name of your macro. Replace the names correspondingly in the **CoprPeriphDemo.c** file.



9. At this point, due to a bug in Vitis, we need to update two **Makefiles** used for compilation of the Custom Coprocessor. In the Moodle, you should find the file **Makefile for custom IP cores.txt**. Open it. **You should use it for all projects involving co-processors.**



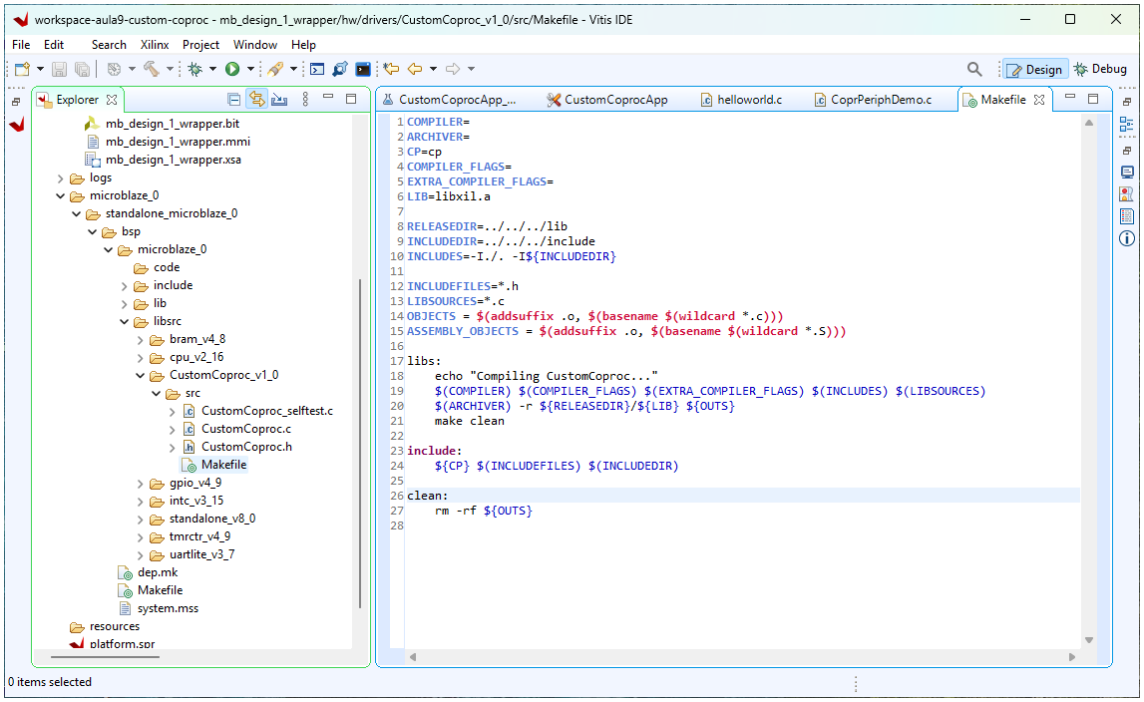
10. The first **Makefile** can be found in the following location.



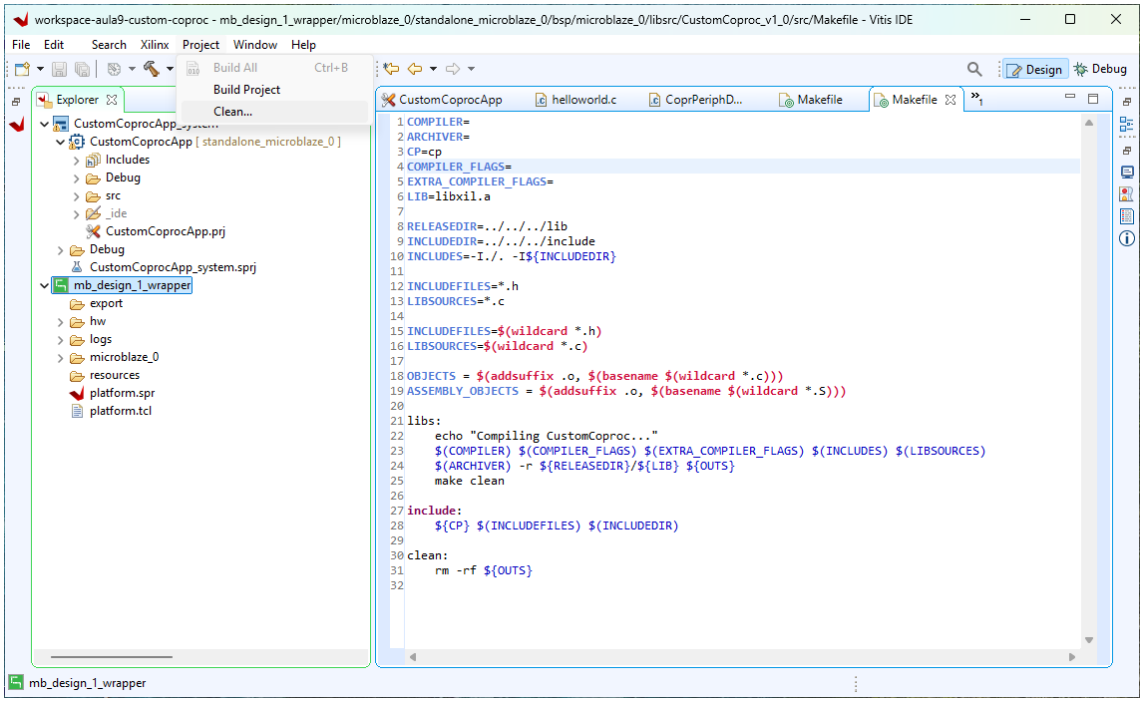
11. Replace line 14 with the following lines. Also, select and copy the whole file (Ctrl-A + Ctrl-C).

```
OBJECTS = $(addsuffix .o, $(basename $(wildcard *.c)))
ASSEMBLY_OBJECTS = $(addsuffix .o, $(basename $(wildcard *.S)))
```

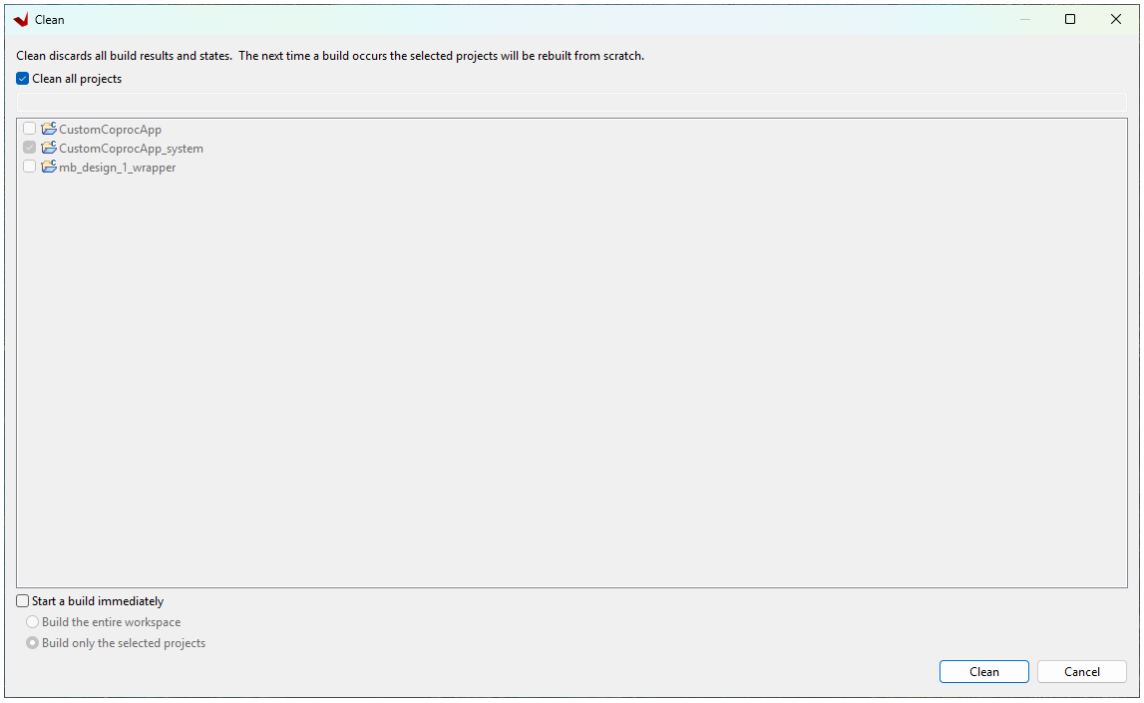
12. The second **Makefile** can be found in the following location. Replace its contents with the previous **Makefile** (Ctrl-A + Ctrl-V).



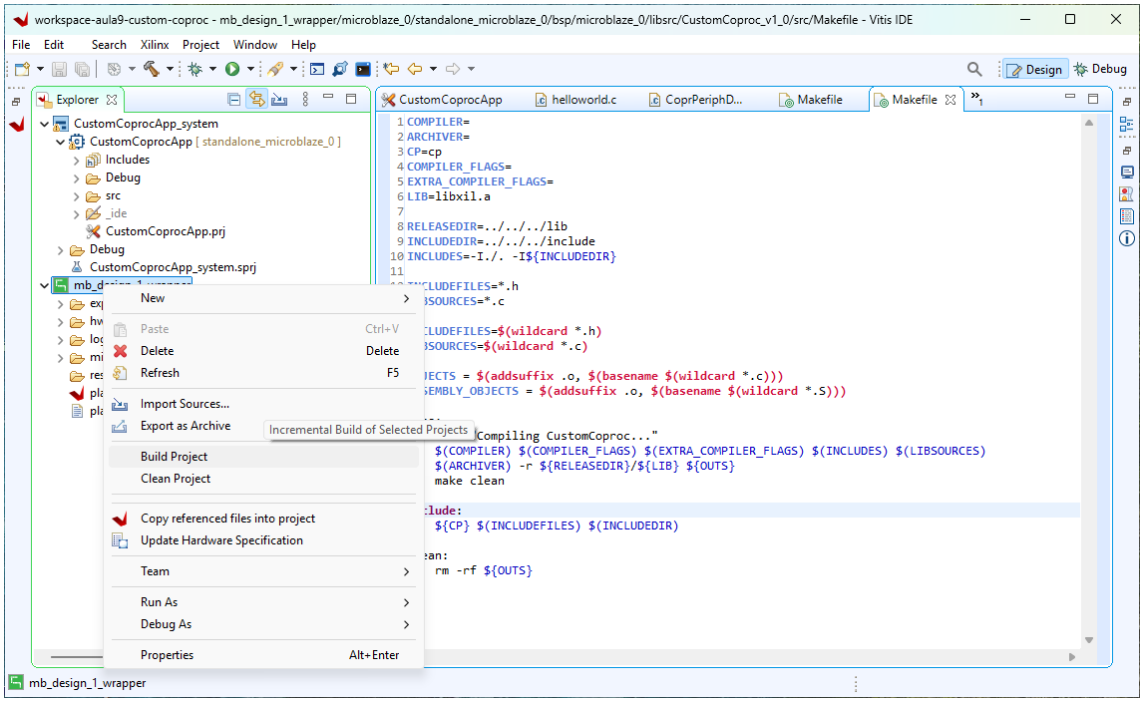
13. Click on **Project** > **Clean**.



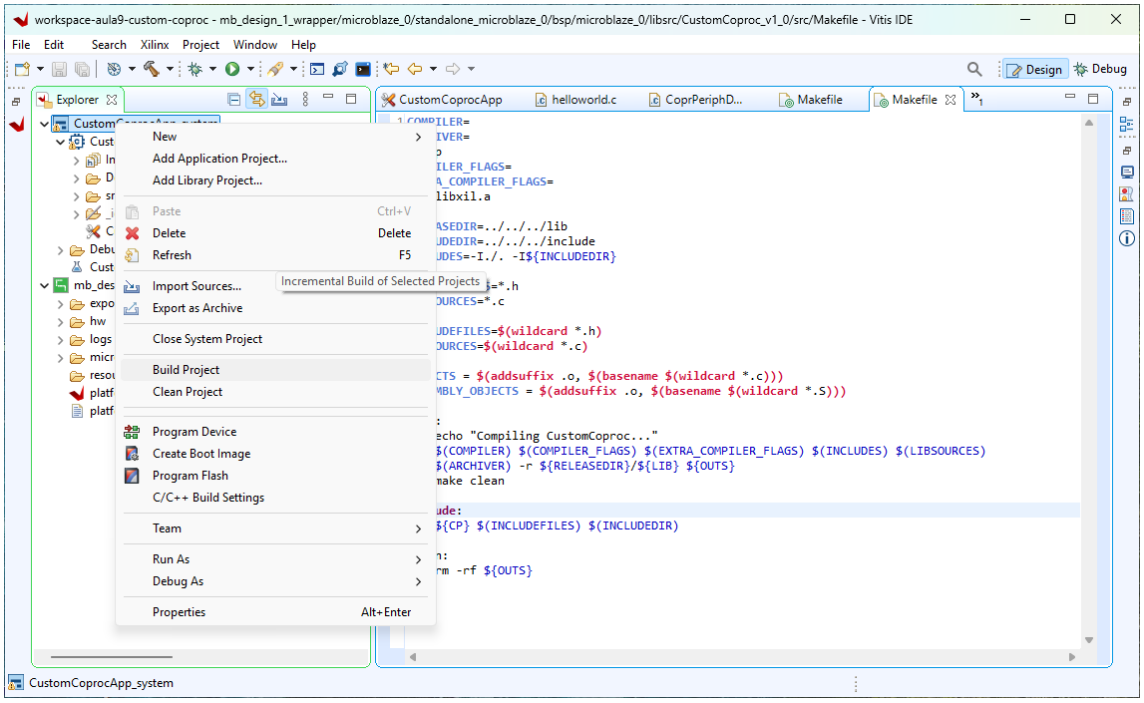
14. Clean all projects.



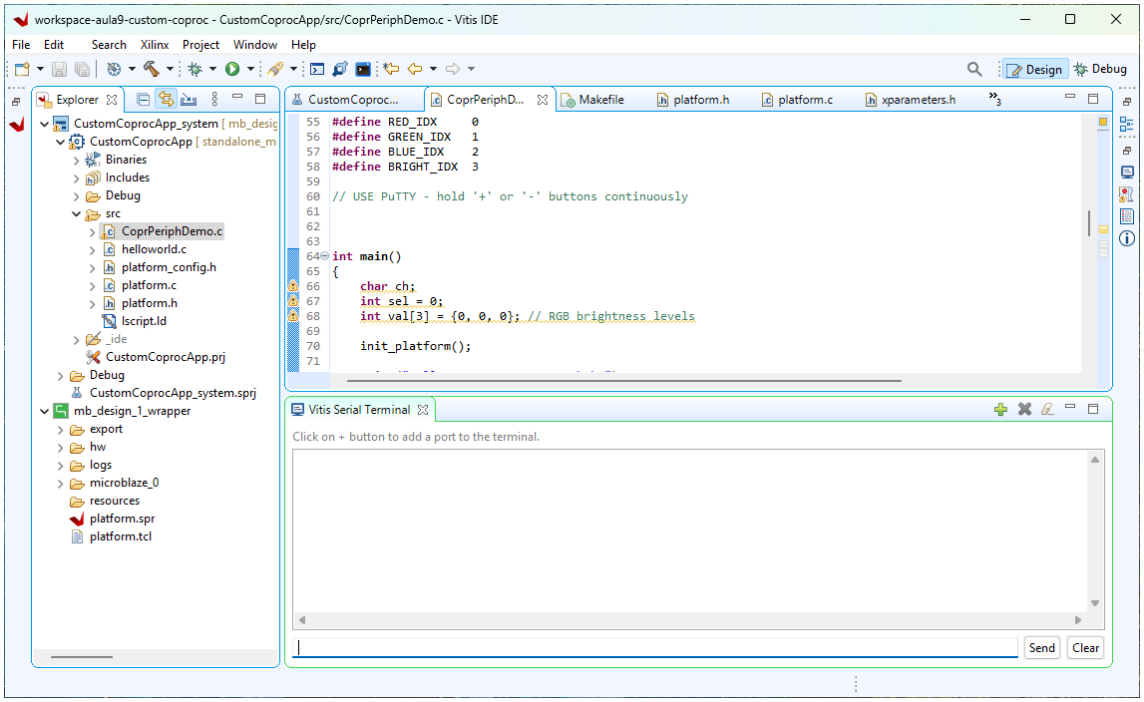
15. Build the platform



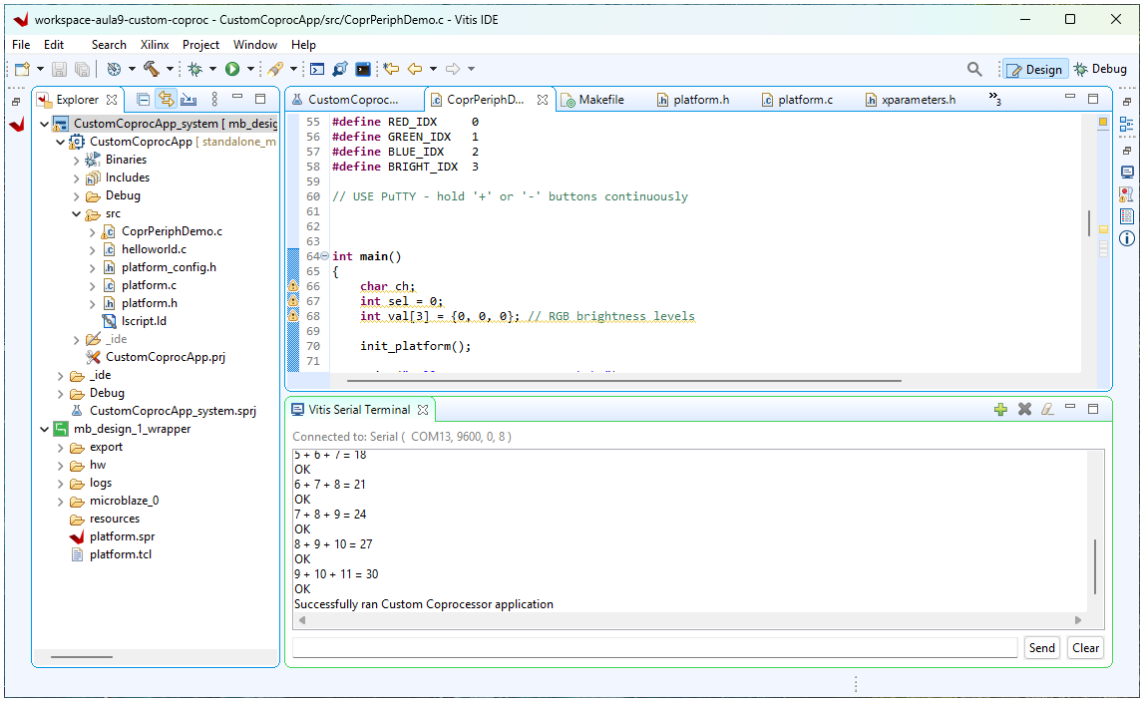
16. Build the application.



17. Open the Vitis Serial Terminal. Select the right configuration parameters.



18. You should observe the program output.



Congratulations!

Sources: AMD Xilinx documentation.