

1 Project Scope

The final project of the class is to implement the famous PONG game with a changing background based on the Mandelbrot fractal (see Figure 1). The game and the background image computation will be implemented on the FPGA. The project will allow you to exercise your skills in control logic, finite state machines, and design and optimization of a datapath.



Figure 1: PONG game project outcome illustration.

2 Project Organization

The project is partitioned into multiple steps that build on each other with a new step of the project introduced every week. The project steps are introduced more rapidly than most can likely follow to provide you all with greater flexibility as to when you can start and when you can stop working on the project steps. The steps are designed to build a more complex system while having a running system even before or without completing all steps. The four specific steps (individual labs) are as follows:

1. Output basic graphics on a screen (using a self-made VGA interface)
2. Adding a fixed background using a frame buffer
3. A basic PONG game with simple graphics generated online by logic
4. Computing a Mandelbrot fractal online on the FPGA as background (and potentially updating it during the game)

These steps corresponds to Labs 5-8 and are illustrated in Figure 2.



Figure 2: Steps toward completion of the full project.

Even though you can successively add these building blocks to the initial design, **we propose that for each step, you create a new independent project and import the design files from the previous step (project)**. In this way, you always maintain the previous steps and can go back to them as needed.

2.1 File Organization

Figure 3 shows the general file structure of the handouts we have used in the course. **It is important when working with different projects over time that you have a clean and organized workspace, especially for the final project of the class!** We have noted that many students are not very organized in how and where the code is stored, which causes various issues. The reason to keep everything organized like this is that:

- It makes it very easy to know where your code is if you are consistent.
- Having the source code separate from the Vivado working directory means that in cases where the Vivado project is somehow misconfigured, it is easy for us to just delete the project instead of having to find your code inside the working directory.

This means that when you create your own VHDL files, please put them in the `src` directory instead of letting Vivado put it inside the project (working directory) or putting them somewhere else like your desktop. Additionally, you should have **no spaces in your paths!** This causes issues with the Vivado project in some cases.

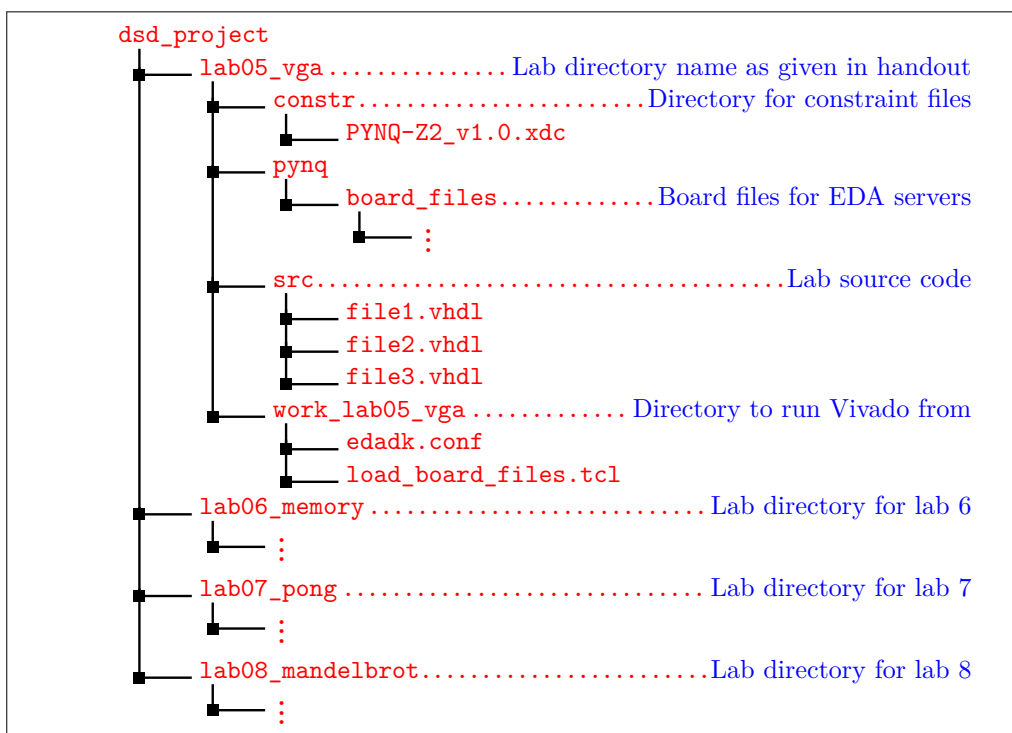


Figure 3: Recommended (strongly) directory structure for the project.

3 Project Architecture

The project is designed to allow you to re-use and build on the previous steps and simply extend and modify your design to add functionality. You may not need all the functionality of the previous steps in each step, but the core components will come in handy. The top-level structure of the final design is illustrated in Figure 4. You can see from the figure the individual building blocks as they correspond to the individual steps of the project.

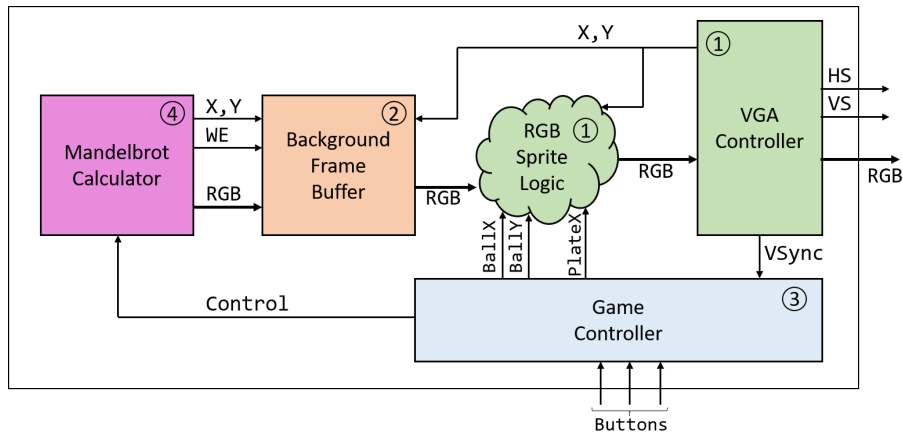


Figure 4: Final schematic with components of the individual steps.

4 Project Timeline

New steps of the project will be presented in the first four weeks (sessions) of the project along with some supporting material (VHDL files to start from for the individual steps). All the necessary background material has already been presented in classes before the start of the project and no new knowledge will be needed for the project. The project deliverables are described in Section 5.

5 Project Deliverables

The project should be completed in groups of 2-3 people. At the end of the project, you will need to deliver the following material for the lab project and your final presentation. Groups must submit a single presentation/set of files.

- Lab project hand-in: You must hand in the following for your project implementation
 - **Vivado project:** An archive (zip file) of your final Vivado project in its final state in a version that compiles and downloads to the FPGA. **Before creating a .zip file, please run the reset_project project command in Tcl console of the Vivado project.** If you have followed the directory structure in Figure 3 it means that all you have to do is just create a .zip file from the lab08_mandelbrot directory, which then includes both the source files and the Vivado project itself.
 - **Source files:** You must include your source files, ideally together with the zip file for the Vivado project so we can directly unzip and have a working Vivado project that includes your source files.
 - **Bit/bitstream file:** A final separate bit/bitstream file that can be downloaded to the FPGA for testing your implementation.

- Presentation hand-in: You must hand in the following for your project presentation
 - **Video presentation:** A final (approximately) 8 minute video presentation must be uploaded where you present your final design and show your implementation running on an FPGA. **Your video presentation is required to show the implementation running on the FPGA and clearly indicate the group members.** Additional requirements are:
 - * You should use screen recording for the video. **Do not use a camera to film your computer monitor as the quality is quite low.**
 - * Your video must have on the first page the group members and the group number.
 - **Slides:** The slides used for your video presentation must also be uploaded. Your slides must also indicate your group members. **You must submit your slides as a PDF, not a powerpoint file.**

The Vivado project, source files, bit file, video presentation, and slides must all be uploaded to Moodle by the 22nd of December at 20:00. All the files should adhere to the following:

- Groups must submit a single presentation/set of files
- All submitted files must include the group number.
- All submitted files and videos should ideally be in English. Please request an exception from the TAs if you wish to submit in French.

6 Project Grading

The project will be graded based on the deliverables (i.e., the code and the report). The grade range depends on the steps you have completed. The following list provides a guideline on which (maximum) grade you can achieve with which step for the project. Note that the final grade can deviate from this range depending on the midterm, the quality of the code, the report, and additional or missing functionality or features:

Project step	Grade range
VGA Controller / test image	4.0 / 4.25
Pong game	4.75
Pong game with background image	5.0
Pong with mandelbrot background	5.5
Extra (examples given in lecture)	6.0