Project Specification

1 Introduction

The aim of the project is to create a complete functional system on the DE2-115 FPGA board. The task consists of using the DE2-115 board to modify/analyze a sound signal (in stereo) that is generated by a sound source. The input sound signal shall be converted into digital form and modified on the FPGA. The output sound shall be sent to the loudspeakers. Information related to the signal will be displayed graphically on a VGA attached to the board. The red and green LED's and 7-segment displays on the DE2-115 board can also be used for this purpose. The settings will be controlled by a keyboard attached to the board and, optionally, the push buttons and switches of the board.

The project is done in groups of 6 people (in principle three lab groups together). Each project group will be supervised by one of the teachers. The supervisor will help the group to decide about the system that they will build and will support them during the entire project.

The project bases on the lectures, assignments and laboratory work. Thus, students can make use of their designs from the laboratories as part of the system in the project.

1.1 Project Timeline

The project is executed in three phases:

1. Planning phase: Defining the project. Here a design specification and project plan takes form.
2. Implementation phase: VHDL!!!!!
3. Round-of phase: Verify the product, write report, present the result.

More in details, the progression of the project is as follows (many of those steps/documents are explained in separate sections below):

1. **Form a project group**, discuss what to do (see section about Requirements below). Make sure all are registered in the course. Also make sure all passed lab 2 (this is a requirement to do the project). Inform the course responsible about your project group.
2. You are assigned a supervisor. Book the first meeting.
3. **First meeting** (“M1”) with supervisor: Write a **requirement specification**. Discuss the design.
4. Write the **design specification v0.1**. The project plan v0.1 is provided.
5. **Second meeting** (“M2”) with supervisor: Updated talk about design and project plan.
6. Write v1.0 of the **design specification** and **project plan**.
7. Do the **design presentation**.

8. Brief start of implementing the design according to design spec v1.0.
9. **Third meeting** (“M3”) with the supervisor: Discuss discovered problems in the design.
10. Write v2.0 of the **design specification** and **project plan**.
11. **Implement the design in VHDL** (this is the main part of the project).
    This step includes a **forth meeting** (“M4”) with the supervisor.
12. **Verify the product:** Product vs requirement specification, and check VHDL code style.
13. Write the **project report** and project presentation. This partly overlaps the end of the implementation phase.
14. Do the **project presentation** and **demonstration** of the product, for the teachers and another project group.
15. Good work. You passed the project.

The course website includes a “Dates and deadlines” page, that specifies the pace of this flow.

1.2 Documents

It may look heavy, but the documentation in the project is quite minimized. The aim of this is to keep focus on FPGA development rather than writing documents.

Apart from what is listed above, a **time report** has to be weekly written/updated. This helps keeping track of the progress of the project.

You may also, optionally, record a video of about a minute, where you present and sell in the product. If you publish the video on e.g. Youtube, feel free to mention the course, but do not show/mention any person without permission.

All documents, except slides to the presentations, have templates, that will help you. The templates are written in libreOffice format. You may change to Word or rewrite as LaTeX or anything you prefer. You can also arrange the documents with other sections if you want, as long as the required content is present.

1.3 Roles of Teachers and Students

During the project, the teacher help the students to understand the problems that they face and suggest them what to try or how to think, so that they can solve the problem themselves. During the initial meetings, the teachers also help the students to understand and focus their project.

The supervisors check from time to time (by either mail or going to the lab) the evolution of the project. However, the students take the main responsibility of their work. They are responsible to ask or report about any problem or difficulty they find, any internal problem in the group, any excessive delay with respect to the planned time, etc. By default, if no information from the group is received, the supervisors will assume that the project is running smoothly. Especially if the group gets stuck with some problem, we encourage the students to not delay it longer and communicate it to the supervisors, so that the project is not delayed too much.

2 Requirements

The requirement specification are divided into three sections:

1. General requirement, that all groups must do
2. Alternative tasks, with a specific set of requirements. Select one. They are listed below.
3. Distinctive features. Customize the product by making up some additional requirements.

You will write the requirement specification together with the supervisor during the first meeting.

2.1 General Requirements

All groups must implement a digital volume control and balance for the incoming sound signal. Each group can decide the resolution of these controls (minimum is a volume control with 10 levels and 10 levels of left-right balance). The settings are controlled from the keyboard. All current setting must be indicated on the VGA screen.

All the project files must be written by the students. Automatic generation of VHDL files is generally not allowed (memory content may be generated).

2.2 Alternative Tasks

Apart from the general requirements, each group shall select one among the following alternative tasks.

Task 1 – Digital Oscilloscope

Use a VGA screen to describe the shape of the sound signals. The students can decide about how the information is exactly displayed on the VGA.

Task 2 – Signal Level Indication

Use a VGA screen to show two bar diagrams indicating the average magnitude (in dB) of the signal in each output channel. Some form of averaging algorithm must be used to avoid flickering of the displayed signal. The peak level of the signal for a certain period of time should also be displayed.

Task 3 – Echo

Add echo to the sound. The user should be able to change the strength and length of the echo. The students can also consider the option to include several echoes.

Task 4 – Equalizer

The stereo signal is divided into (at least) two frequency bands. Each band has a linear volume control. All bands are merged together when sending to the speakers. Information about the configuration should be shown on the VGA.

For this task, the students should be well familiar with digital filters.

Task 5 – Create your own project

For the project, you can choose your own task that makes use of any components of the DE2-115. Which is your dream system? The supervisor will be happy to listen to your proposal, give you feedback and support you in the project. The supervisor or course responsible must approve the proposal.

2.3 Distinctive Feature

You should add more requirements that makes your system unique. This can be e.g. visual effects, something related to to selected alternative task, or something completely different.

2.4 Requirement Priority

The requirements are specified in three priority levels:

Level 1 – All level 1 requirements must be implemented.

Level 2 – Some level 2 requirements must be implemented.

Level 3 – Implement if you have hours left to spend in the project after the level 2 requirements.

The exact number of implemented level 2 requirements is not specified, and depends upon the quality of the fulfilled requirements. Keep the dialog with the supervisor open toward the end of the project, until the he/she is pleased with the product. This rather informal level of requirement specification aims to reduce the documentation burden.

3 Design Specification

The design specification has three aims: First aim: You solve a big potion of the project just by writing it. Second aim: The specification will work as a reference for you when implementing VHDL. Third aim: to help the supervisor see that you actually have a plan of how to proceed.

The document quality of the design specification is of less importance. Spend as little time as possible on e.g. creating fancy figures (paint or a photo of a whiteboard is enough). Do not waste words on trivial things that everybody already knows. **Focus on what is actually hard**.

The document is created in three versions, as described below. There is a template for the design specification, that include instructions corresponding to the information below.

3.1 Version 0.1 of Design Specification

Aim for version 1.0, but expect to not manage it. You may leave gaps with comments like “How to do this?”. Take note of what you want to talk about with the supervisor in the second meeting.

3.2 Version 1.0 of Design Specification

The design specification template contains lots of information about what to do and write about. Some major parts:

First the **introduction**: Describe in a few sentences what the product shall do.

You need a top level **block diagram**, where you specify what blocks you have, and how they are interconnected. 3-5 blocks are suitable.

Specify the **communication** between the blocks, e.g. “here comes a stream of samples to the volume control”.

You also need a **block description** – what’s their tasks etc. Brief ideas of how this is implemented.

Finally, you need to spend some imagination of future **challenges** and how to manage them: What will be hard to solve, and how will you attack the problem to not get stuck. The earlier you face the problems, the less likely that they risk a project failure.

When done, have a glance at the requirement specification: Can you tell where each requirement will be solved? E.g. “There are at least 10 levels of volume” partly requires a register to keep track of the current level, and partly requires a block that perform the actual volume modification.

After this version, you should start a **draft implementation** in Quartus. This aims to let you face any aspects you may have missed. It’s just a scratch/draft, and be prepared to discard much of this work (although not required).

3.3 Version 2.0 of the Design Specification

This version includes updates due to found problems, discussions with the supervisor etc. Especially, it includes major updates with implementation details.

Moreover, the **communication** part must now contain much more detailed information, e.g. word lengths, timing, enable signals, bit definitions etc.

This is things you anyway need to decide, in the project, and the design specification is a good place to document it in. Keep it as simple as possible, without loosing details.

The **block description** is refined to include possible sub blocks, FSM states, timing diagrams etc.

4 Project Plan

The project plan helps you plan what should be done and when. In this way, you avoid a situation like “it’s just one week remaining, and we have plenty to do in other courses”, when 80% of the project remains.

The project plan is handed in together with the design specification.

Project plan content, briefly:

- Member of the group, “customer” (examiner), supervisor etc.
- Documentation overview – which documents/presentations are written and what’s their deadline?
- Time plan: A number of tasks are specified, and scheduled. A task can be to e.g. implement a design how a module is to be implemented, another to implement it, or to test it. The plan specifies those, and who will do them when.

- How to handle misc problems, like illness, collaboration problems, or disagreements about something.

Expect plenty of time to be spent on merging all modules into one project. Also, miscellaneous problems and silly mistakes are known to consume too much working hours. Remember to plan for trouble.

There is a template also for the project plan, that include instructions corresponding to the information below.

4.1 Version 0.1 of Project Plan

This version is already written for you, and covers the start of the project, up until version 1.0 of the design specification and project plan. You only have to fill in e.g. group number and members.

4.2 Version 1.0 of Project Plan

Now update version 0.1 to cover the development of documentation version 2.0, including a Quartus draft implementations. Also include/correct a draft of how much time is available for the rest of the project.

4.3 Version 2.0 of Project Plan

The final version: A complete list of tasks, and who will do them when and when. This has three aims: Getting a plan, get an early warning if the project lags, and mainly: Get experience in doing time estimations.

5 Time Reports

After each project week, you should (at latest next Monday at 12:00) send a time report, where you specify what have been done during the last week, and compare this with the project plan.

6 Design Presentation

Around the third meeting (M3), you should hold a presentation of your current design (version 1.0 of the design specification), for the teachers and another project group. Since it covers an incomplete design specification, you are not expected to have solutions to all problems.

All group members must participate in the presentation, and use powerpoint/PDF/something, on a beamer. You can use a computer of your own, or a teachers computer.

Aim to let the entire presentation be about 15 minutes. Make the smallest possible effort to produce the presentation. Do not practice too much (you have better things to spend time on). It’s better to be nervous due to lack of practice, than to waste loads of valuable time in preparation.

After each presentation, the other group, and the teachers have the opportunity to ask questions.

There is no template for the presentation.

The purpose is to reduce the fear of presentations, and to exchange ideas/thoughts with other groups.

7 Implementation phase

The implementation phase have a sneak draft start after the design 1.0 is done. The real work starts in project week 4, and you have a few weeks to implement the design in VHDL and on the DE2-115 board.

Remember to keep a good coding style. Decide together how to handle e.g. tabs, to not make an indentation mess when mixing editors. See Lab2 manual, appendix B about coding style.

During the first week in the phase (project week 4), you are to have the fourth meeting (M4) with the supervisor, to just discuss how it is going, ask for help in getting started (if need be) etc.

8 Verification

In the round-up phase, you verify the product against the requirement specification together with your supervisor. A teacher will also verify that your code is not a shame.

9 Project Report

You are to write a project report. There is a template, that includes a description of what is expected of the report. The intended reader is the supervisor, and, where relevant, a fictive future project group that will continue your work. Consider your own knowledge in the beginning of the course – how would you need the text to be written in order to understand it?

10 Project Delivery (Presentation, Demonstration and Code)

Finally, you should present the product for the teachers and another project group. First with PowerPoint or similar, and then with a demonstration of the product.

The presentation have the staff and another group as intended audience. A touch of selling is welcome (be proud of your awesome product). There is no template. Include the following:

- Introduction: What does the product do? Emphasis on your specific parts from the requirement specification.
- Design: How does it work. Block diagram etc. You can skip loads of details here. Details can be included if they are relevant to discuss challenges later in the presentation.
- Challenges and experiences: Solving problems is a great source of experience. What do you bring to your future life? Personal experiences are optional.
- Time summary: How many hours have you spent on the project?
- A user manual, in case anyone else than you want to test the product.
- (thanks for the applause. Any questions?)