FINAL PROJECT REPORT

Introduction

Next to making something work, the most important part of any project is documenting it. The process of documentation consists not only of the final report, but includes all written communication produced during the course of a project's development — from the original specifications and functional block diagrams right on through to the detailed circuit schematics and parts lists. Good documentation makes the design process go smoother and enables others to modify or troubleshoot your design after you've gone. Incomplete or inaccurate documentation is frustrating and can cause customers to lose confidence in a product or company.

Collect documentation as you go

Your final report should enable someone to understand and to reproduce your design. It will be easier to collect and update the supporting documentation as you go through the design and development of your project, rather than trying to put it all together at the very end. These documents will be placed in appendices to your final report.

1. System level diagrams

(a) **Front panel:** An annotated digital photo of your project (Nexys2 or Nexys3 board + add-ons) is helpful to show how you use the buttons, switches, LEDs, etc, and also where PMods plug in, etc.

(b) A **functional block diagram** which gives an overview of the hardware functional modules. By looking at this diagram, anyone else who has had the course should be able to understand *what* your project does. It is not necessary to use a drawing program to create your block diagram. A piece of graph paper and a ruler work just fine.

Each block, each input / output port in a block, and each net connecting blocks should be labeled with a descriptive name. Input ports for a drawing are always at the left edge of the drawing, and output ports are at the right edge. Additionally, ports that connect to/from Pmods must be labeled with a connector and pin number, *e.g.*, JA-2 for pin 2 of the JA Pmod connector.

Here is a suggested hierarchy for your block diagrams:

(1) The top level that shows the main functional blocks, including switches, displays, etc. At this level, the logic in your FPGA can be one big "black box" with ports labeled to match the port declarations in the top level of your logic design (whether VHDL or schematic).

(2) The top level of logic in your FPGA. Do not use the RTL schematic generated by the Xilinx tool (it leaves things out).

(3) Work down "recursively" into each block. Make sure your drawings are labeled in such a way that their place in the hierarchy is clear.

(c) A **schematic diagram** with *signal names* and pin numbers for any *off-board* circuitry that you designed and built (*i.e.*, on the Pmod breadboard). The purpose of this diagram is to show how your project is wired up and to help in debugging if it doesn't work. Make sure you include detailed schematic diagrams for any special circuits (*e.g.* LED arrays). Each package should be given a unique identifier, known as the reference designator: U*n* for integrated circuits (where *n* = 1, 2, …), R*n* for resistors, C*n* for capacitors, Q*n* for transistors, S*n* for switches, P*n* and J*n* for plugs and jacks, respectively, D*n* for diodes. LED*n* is sometimes used for LEDs or arrays of LEDs (like 7-segment displays). If you have other parts, ask me.

(d) A **package map** showing where each component is located on your board. This only applies to *off-board* circuits that you built. The package map should resemble an "aerial photograph" showing clearly all connectors (labeled J*n* or P*n*), sockets, ICs (U*n*), switches (S*n*), displays, etc. The package map is used as an intermediate stage between the logic diagram and your circuit. Suppose you are troubleshooting your project. By studying the symptoms and your logic diagram, you deduce that there may be a wiring error at a particular part, *e.g.*, a keypad encoder. On the logic diagram, that chip is labeled U8. You find U8 on the package map and it shows you precisely where to look on your circuit board to find the suspicious part. An easy way to create a package map is to annotate a digital photo of the board.

(e) **Parts list**: List all external components, including Pmods and parts that you wired up on external boards. You don't need a parts list if everything is confined to the Basys or Nexys board. Use sensible groupings (IC's, discrete components, switches, connectors, etc.), and keep this list compact but complete. Omit trivial details like wire and sockets.

2. **Programmed logic**

   (a) **State diagrams** for all your state machines.

   (b) **VHDL code** for each module.

   (c) **Resource utilization** for your FPGA

   (d) **Critical timing path**, explained

   (e) **Analysis of residual warnings**, explaining why they are benign

3. **Memory map:** If you are using a memory chip or block memory in the FPGA as a complex look-up table, show how the address and data bit patterns are partitioned among the logical variables involved (*e.g.*, for storing sound clips, address bits 12-13 are which sound, and address bits 0-11 are samples within a sound).

4. **Waveform graphs** from your major simulations, showing all important signals, significant features annotated.

5. Copies of **data sheets** for any special components you used that aren't already on the class website or on Digilent's website.

6. **Computer programs** (*e.g.*, Matlab codes used to generate lookup tables, etc).

7. **Other documentation**. Include anything else that might be needed to understand or reproduce your design.

Write the Report

Your project report must contain the sections described below. Use the same section numbers (where specified). The abstract and table of contents do not have section numbers.

Abstract: Give a concise summary of the goals and results from your project.

Table of Contents: Include page numbers of all sections and appendices.

1. **Introduction:** State the problem to be solved.

2. **Design solution**

   2.1 Specifications: Describe *what* your circuit does. This section will be similar to the project proposal. List any inputs such as buttons, keypads, sensors, etc., and any outputs such as displays, LEDs, actuators, etc. Include a block diagram of your circuit as well as a diagram of the control panel, showing the locations of buttons, displays, lights, etc., relative to each other.

   2.2 Operating instructions: Provide any information needed to set up and run your circuit. This is the User's Manual part of your report.

2.3 Theory of operation: **Discuss in detail** how your circuit works. Start with a high-level overview of your design, illustrated by the block diagram. Give a prose description of each of the blocks and the interfaces between them. Describe what each block does and how it works. Make sure you discuss any tricks or functions that are not obvious from the logic diagram (*e.g.*, ROMs used as lookup tables). **From the information given in this section** (with references to additional diagrams in the appendices), **we should be able to reconstruct your design.**

2.4 Construction and debugging: Describe how you built and debugged your circuit. Which components did you build first? How did you test them? What problems did you have?

3. Justification and evaluation of your design: Why is your solution better or worse than some reasonable alternatives? What might you do differently if you had it to do over again?

4. Conclusions: Summarize the goals and accomplishments of your project. Give a comparison of your original proposal (and final specifications) and the degree to which you achieved your proposed objectives. What recommendations do you have for future groups considering such a project? What words of general advice would you give to future groups?

5. Acknowledgments: Technical reports and papers often have an acknowledgments section that thanks people other than the authors for their contributions. Use this section (if you wish) to acknowledge various sources (e.g. course staff, students, faculty, or others) for design ideas and other assistance.

In addition, you should use this section to explain the contributions of each partner to the project. (This is **required**.) For example, if each project partner was responsible for designing a different block of the system, state that here. If one partner had more responsibility for design, another for debugging, etc., describe that in this section. If different partners were responsible for different parts of the report, note that here.

6. References: Cite any references (books, application notes) used in your design (*e.g.*, for some clever circuit design). It is not necessary to acknowledge parts specifications (data sheets), unless you are quoting.

7. Appendices: All the diagrams, etc, you collected go here.

Written Style

We are interested in promoting good, efficient, and complete technical writing; prose style, grammar, and spelling all count in the project grade evaluation. The content of your report is more important than the thickness, so keep the information content per page as high as possible, consistent with clarity. The following guidelines are to be followed to ensure readability of your report.

1. Reports should be *typed, double-spaced*. Use 8.5″ x 11″ (A-size) paper, with approximately 1″ margins. You may bind your report in a folder, but a single staple in the upper left corner is sufficient. Paper clips are **not** acceptable. Do not use binders that are larger than 8.5″ x 11″.

2. Use 10-, 11-, or 12-point fonts in plain text style. Prepare your report in Times New Roman, Palatino, or similar *serif* font (*i.e.*, having spurs at the ends of the letters). The font for this sentence is 10-point Palatino.

3. For larger diagrams, you may use 11″ x 17″ (B-size) foldouts or plotter printouts, but they must be folded up properly to 8.5″ x 11″ for insertion in your report. For ease of discussion in the main body of your report, put segments of your diagram on 8.5″ x 11″ paper — the larger printouts will go into the appendices. Reduce drawings with a photocopier if necessary.

4. For diagrams, we recommend Helvetica or a similar plain *sans serif* (no spurs) font. Typewriter font (*e.g.*, `Courier` or `Monaco`) is commonly used for VHDL listings.

5. The discreet use of *italics* or **bold face** for emphasis is permitted, and **bold face** is recommended for section headings. Underlining is not to be used — it is a relic from the days of typewriters.

Figures

The purpose of any exhibit is to support the narrative in your report. When you write about a simulation, the corresponding exhibits include the testbench and a printout of the simulation waveform. When you write about a state machine, you want to include a state diagram and the VHDL code. Here are things you should do to maximize the impact of your exhibits.

1. Every figure has a descriptive caption, either typed or neatly handwritten, in ink not pencil.

2. Every graph, including simulation waveforms and oscilloscope screenshots, has a legend explaining what each graph or oscilloscope trace is. For example, instead of "1, 2, 3, 4" labeling the analog channels of the oscilloscope printout, say that channel 1 is the debounced A switch, channel 2 is the debounced B switch, channel 3 is the CW output, and channel 4 is the CCW output. You can do this in a caption, a classic legend, or by relabeling the names of the graphs with a pen.

3. If there are particular things your figure is intended to show, bring it to the reader's attention through the caption ("Note the narrow pulse in CW following the falling edge of B") or by "calling out" the feature with a circle or arrow, and a label.

4. Make sure your VHDL code is well-commented.

5. Refer to each exhibit by name ("Appendix 1", "Figure 2", …) at a relevant place in the text of your report, and again direct the reader to what the exhibit contributes to the report. ("Appendix A shows the timing of the quadrature encoder…")

6. Make sure your figures agree with one another—the state diagram matches the VHDL model, and the simulation waveform matches the state diagram. They won't support the argument in your text if they contradict each other.

7. Adjust the color scheme of simulation waveforms (you can do that in the simulator preferences) so that the background isn't black, and the line colors facilitate reading. This helps conserve toner in the printers. Make sure that all the letters in your figures are large enough to be easily read; compare their size with the font size of your main report. Maybe you can't change the fontsize, but you can use an enlarging copier if necessary.