FINAL PROJECT REPORT

This is a long handout. Certain items are marked "**Only if...**", and may not apply to you, but most of the handout applies to everyone. Read it slowly and carefully, and make sure you and your partner understand it.. **The project is 20% of your grade.**

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.

This does not have to be a long report. Think carefully about what you will include, write crisply and clearly.

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.

Gather these documents into a set of appendices (1-7) to your final report. Bind them together into one unit and submit separately from the narrative part of your report.

1. System level diagrams

(a) **Front panel:** An annotated digital photo of your project (Basys3 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) As needed, 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) **Only if** you designed and built any "custom" circuits as part of your project (e.g., on a breadboard), then include these in your documentation:

(1) 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 $n = 1, 2, \ldots$), 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.

(2) A **package map** showing where each component is located on your board. This only applies to *off-board* circuits that you built. An easy way to create a package map is to annotate a digital photo of the circuit. It should clearly show 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. If, on the logic diagram, that chip is labeled U8, you look up U8 on the package map and it shows you precisely where to look on your circuit board to find the suspicious part.

(d) **Only if** you used parts other than the Basys 3, 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 Basys3 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. Edit the final XDC file to remove the lines for unused pins (currently commented out).

(c) Post-implementation **resource utilization** for your FPGA

(d) **Analysis of residual warnings**, explaining why they are benign (non-benign warnings must be cleared).

3. Memory map: Only if you are using a memory chip or a block memory in the FPGA as a look-up table, give the dimensions of the memory (locations × bits per location). If you partitioned the address or data (eight bits for x, eight bits for y, etc), explain how the memory is used, and what the stored bits signify.

4. Waveform graphs from the simulations of your components, showing all important signals, significant features annotated. Each component that you design (.vhd file) should be separately simulated. If you can, print your waveforms on a light background rather than black.

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

6. Computer programs (*e.g.*, Matlab scripts 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. Collect these pages into one PDF file and submit with a separate PDF for the appendices.

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. List any inputs such as buttons, keypads, sensors, etc., and any outputs such as displays, LEDs, actuators, etc. Specify the clock speed, audio

sampling rate, video image size, etc, as applicable to your project. This information is conventionally written as a table or a bulleted list.

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. Refer to the block diagram and "control panel" of your circuit, showing the locations of buttons, displays, lights, etc., relative to each other.

2.3 Theory of operation: Discuss in detail how your project works. Start with a high-level overview of your design, illustrated by the block diagram. Describe, in prose, what each block does and how it works. 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 As you write, refer the reader to the various supporting documents you collected: block diagrams, state diagrams, simulation waveforms, VHDL files, *e.g.*, "As shown in Figure 4…" or "Please see Appendix B."

3. **Evaluation** of your design: What do you like best about your design? What was particularly challenging? 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. students, TAs, instructors) for design ideas and other assistance.

   In addition, 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, websites) 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.

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 must be *typed, 1.5 or double-spaced*. Use 8.5" x 11" (A-size) page layout, with approximately 1" margins. Microsoft Word or LaTeX is typically used.

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 diagrams, we recommend Helvetica or a similar plain *sans serif* (no spurs) font, no smaller than 9 point for details. Typewriter font (*e.g.*, `Courier` or `Monaco`) is commonly used for VHDL listings.

4. 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, include a state diagram and the VHDL code. Here are things you should do to maximize the impact of your exhibits.

1. Every figure is numbered and has a descriptive caption, either typed or neatly handwritten, **in ink not pencil.**
2. Block diagrams and state diagrams may be drawn by hand, or using a drawing program. There are some good free applications on the Web. For style, imitate the drawings you've seen in readings and handouts.
3. 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 for a rotary decoder, 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.
4. 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.
5. Make sure your VHDL code is well-commented, with accurate header blocks at the top. It is common in development to comment-out fragments of code that are temporarily unused. These fragments must be completely removed in the final working version of your projectUse.
6. 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…")
7. 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.
8. Use color schemes for waveform printouts that are easy to read. 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.

Additional notes on online submission

1. Submit your report as two PDF files, one containing the narrative and the other containing the appendices. This makes it easier for the reader to go back and forth between them.
2. Make sure your report is complete. I will not notify you and ask for a resubmission if I see something missing; I will just mark off points. (Ask me about the student who left half of his—and his partner's—report on the copy machine.)
3. Make sure your submission is readable, of sufficiently high contrast, pages in the right order, right-side up, etc. There are 18 reports to read, and no time to decipher a hard-to-read report. I will just mark off points.
4. Before the report is submitted, both partners should separately inspect it online and make sure they are satisfied with how it looks. After the report is submitted, both partners should again inspect the uploaded files.