University of California, Berkeley
College of Engineering
Computer Science Division
Electrical Engineering and Computer Science Department

CS 150 R. H. Katz
Fall 2000 Norm Zhou

FINAL PROJECT GRADING CRITERIA AND REPORT FORMAT

A critical aspect of your final project is to document what you have done in a final report. A perfectly working design is worthless unless it can be duplicated and built upon, and this is the role of proper documentation. Long after the components are returned and the wiring is torn up, your report will remind you of the design experiences that you had and the many things that you learned. It is not uncommon for students to bring their CS 150 Final Report to job interviews (be sure to make a copy for each project partner). This course has long had an excellent reputation among corporate recruiters from such companies as AMD, Compaq, HP, IBM, Intel, Sun Microsystems, etc.

We understand that you have worked hard on your project. We don't want the final report to represent too much of an additional burden. The key thing is for you to evaluate your design and to document your special features and lessons learned. The report sections and suggested lengths are described below. The typical report will be around 5 pages with quite a few additional pages dedicated to additional appendices, to hold your block diagrams, schematics, state diagrams, etc.) and tables (register transfer operations, microoperations, ROM contents, etc.). Use the following page lengths as guidelines, though we do want to keep written descriptions short and crisp.

The overall report outline is as follows:

1. **Introduction**
2. **Theory of Operation**
3. **Control Description**
4. **Datapath Design**
5. **Control Design**
6. **Evaluation**
7. **Lessons Learned and Conclusions**

Appendix A: Control State Diagram
Appendix B.1: SRAM/FIFO Subsystem Schematics
Appendix B.2: UART Transmitter/Receiver Schematics
Appendix B.3: Audio Interface Schematics
Appendix B.4: Packet Transmission/Reception Schematics
Appendix C: Controller Schematics
Appendix D: Project Checkpoint Check-off Sheets

In more detail, the following describes in more detail the expectations for each subsection:

1. Introduction (0.5 Page)

What is the function of your project, in general terms? We know what it is—we assigned it to you!—so it is not necessary to go into details. This is to make the report able to stand on its own. Concentrate on how you differ from the original specification, especially in terms of extra credits

work. What aspects of your implementation make you particularly proud (e.g., few control states, able to run at a fast clock rate, unusual datapath or control organization, cool user interface, etc.)? Why did you choose to implement that aspect of your design in that way?

2. Theory of Operation (0.5 page)

Describe how your project would be used by someone not intimately familiar with its detailed implementation. What do you do on power up? How do you reset it? What is the detailed procedure for getting it going? What assumptions about the supporting hardware environment are you making, e.g., memory system organization? What are the detailed timing constraints on your processor, e.g., clock frequency or period and duty cycle? Include a diagram that shows the layout of your user's console switches and LEDs.

3. Control Description (1 page)

Describe the basic control sequencing of your system, including your user interface. Attach a state diagram, a program-like description, or similar method for describing your control as Appendix A. In this section, the output of the control should be described in high level register transfer operations, not detailed control signals.

4. Datapath Design (1 page)

The major subsystems of your design map onto the four checkpoints we gave you: (1) SRAM/FIFOs, (2) UART Transmitter/Receiver, (3) Audio Interfaces, and (4) Packet Transmission/Reception. The purpose of this subsection is to document these components in terms of their control signals and their timing behavior, and to describe any changes you may have made to the original checkpoint specifications. Include block diagrams with documented input/output signal names for each subsystem in Appendix B.1, Appendix B.2, Appendix B.3, and Appendix B.4, one section for each of the four checkpoints.

If you made significant change to a subsystem, include in this section a short description of its theory of operation (the most likely checkpoint that you changed is probably Checkpoint #1—tell us about your changes!). Include a high-level register transfer diagram showing busses, registers, and functional units. Focus on how you have interfaced the four checkpoint components.

Include in Appendix B a more detailed schematic, showing all library components and the detailed implementation of interconnections such as tri-state buffers or multiplexers. The detailed diagram should show all microoperation control signals. Appendix B should include a table that correlates the register transfer operations of Section 3 with the microoperations in your datapath. Very briefly describe the function performed by each register transfer operation.

5. Control Design (0.5 Page)

Give a short description of your controller's theory of operation. Did you use random logic, a ROM-based design, or other strategy? Describe your approach for state assignment, if any. In Appendix C, (1) give a block diagram description showing the controller inputs, state register, and outputs, (2) describe any equations or microcode formats, and (3) for ROM-based designs, include a table showing the ROM contents in terms of your chosen format. Briefly describe what is happening in each ROM word.

6. Evaluation (1 Page)

Describe the overall system timing considerations for your design. What is your critical path? What is the maximum speed you believe you can transmit and receive data? Justify your answer based on identifying your critical path and determining its speed. In addition to time, consider space. Include in your answer the number of Xilinx CLBs used in your design. What percentage of the Xilinx component did your design consume?

7. Lessons Learned and Conclusions (0.5 Page)

What have you learned from you experience implementing your project? What would you do differently if you had the ability to start over from scratch? Include suggestions on how to improve the project specification, checkpoints, and process for future CS 150 offerings.

Notes on Extra Credit:

We are limiting it to no more than 20% of the final project grade. First and foremost, *you must have a working base design before you can obtain any extra credit points*. Furthermore, there is no such thing as partial credit on extra credit: either you get the extra credit feature to work completely or you receive no extra credit.

You can get half of that extra credit simply by making the Thanksgiving early demo deadline of 22 November (10 of 20 points). If you have made the early checkpoint deadlines, you can get an additional 4% (4 of 20)! Some of the bells and whistles for extra credit include the following features (or you can invent your own, but consult the Head TA or the Instructor before you endeavor to pursue it).

Don't forget, you must have a strategy for testing your optional extra to convince us that it will work—you can't assume that anyone else in the class will have made a compatible version with which you will be able to interoperate!

Here are some suggests for extra credit:

- *Most Compact Design* (smallest CLB count): For the base level design, we will award 5 extra credit points for the design that uses the fewest Xilinx resources (without cheating by moving too much functionality to off Xilinx components).

- *Broadcast Mode:* Demonstrate a wildcard delivery mechanism that allows one station to talk to many. One way to do this is to reserve station ID 0000 to match every station on the wire. This one is pretty easy, so you only get 2.5 Extra Credit points for making it work.

- *Output Audio Interpolation for Better Output Audio Quality*: In this enhanced feature, the audio packets need to be timestamped at the point of sampling (e.g., assigned a two bit sequence number), and then linearly interpolated for enhancing the playback quality or "filling in" for packets that are lost in transmission. This one is worth 5 extra credit points.

- *Multiple Sample Rate Under Different Modes*: This feature supports high resolution and low resolution audio. When in the low res mode, audio is sampled, sent, and played back at coarse grain sampling rate, e.g., 4 Khz. When placed in the high res mode, the packet stream has twice the resolution, e.g., 8 Khz sampling rate. This will require defining two different packet formats, and modifying the sampling and playback path throughout the input and output processing of the audio. We will award 5 extra credit points for a completely working implementation.

- *Digital Audio Mixing to Support Multiple Incoming Streams:* This feature has the ability to digitally combine incoming samples from multiple speakers. It requires maintaining separate FIFOs, one per source, and some method of time stamping them. The samples are extracted from the FIFOs and combined before output. This one could be quite complex, and we will award 10 extra credit points if you can show us that your implementation works.

- *Call User Interface* (handshaking for calls): Develop a user interface for your project. Define a "call set-up" protocol that works along the following lines. The user enters the station ID of the receiver. She presses a button. This sends a specially encoded control packet to the receiver. Upon receipt, he knows that there is an incoming call. The receiving station illuminates an LED to indicate this situation. By pressing a push button, the receiving user accepts the call, and a response packet travels back to the call initiator. The sender's LED now is illuminated and the connection is established. Audio packets flow in both directions. Packets for the receiver or transmitter from any other station than the opposite end of the call are ignored. You also need to define a handshake for "tearing down" the existing call so a new one can be started with a new station. We will award 10 extra credit points for a completely working user interface along these lines.

- *Your Good Idea for a Feature Here:* Run your idea by us to get its feasibility checked and to arrange for us to assign it a certain number of extra credit points. Don't forget that you must have some rationale way to demonstrate your design feature—especially when you can't demonstrate it with anyone else in the class. And remember that there is no extra credit for features that don't work!

University of California, Berkeley
College of Engineering
Computer Science Division
Electrical Engineering and Computer Science Department

CS 150 R. H. Katz
Fall 2000 Norm Zhou

Final Project Grading Template

Name Partner #1: _____ Teaching Assistant: _____

Name Partner #2: _____ Lab Section: _____

Design & Demonstration (50 pts. + 10 extra credit pts.) Points Awarded: _____

Exceptional	(50 pts.)
Outstanding	(40 pts.)
Very Good	(30 pts.)
Good	(20 pts.)
Fair	(10 pts.)
Poor	(0 pts.)
Early Demo Bonus	(+ 10 pts.)

Checkpoints (20 pts. + 4 extra credit pts.) Points Awarded: _____

- Checkpoint #1 (5 pts. plus 1 pt. early bonus)
- Checkpoint #2 (5 pts. plus 1 pt. early bonus)
- Checkpoint #3 (5 pts. plus 1 pt. early bonus)
- Checkpoint #4 (5 pts. plus 1 pt. early bonus)
- No points for using Norm's implementation.

Oral Project Debrief (20 pts.) Points Awarded Partner #1: _____

Points Awarded Partner #2: _____

Written Project Report (10 pts.) Points Awarded: _____

Exceptional	(10 pts.)
Outstanding	(8 pts.)
Very Good	(6 pts.)
Good	(4 pts.)
Fair	(2 pts.)
Poor	(0 pts.)

Extra Credit (NOT TO EXCEED +20 pts. TOTAL) Points Awarded: _____
+20 pts. max includes early bonuses and additional functionality

Total Points Partner #1: _____

Total Points Partner #2: _____