Mini Project Guidelines for Introduction to VLSI Design

We will use a progressive design, reporting and evaluation process for the project. You will use the following design methodology and reporting procedure, with opportunities to revise your designs and reports as you go. Accordingly, more weight is assigned to the later submissions than the earlier ones.

Following the submission of the reports, you may be asked meet with the instructor or the TA to further discuss your project if necessary. The progress reports are intended to make the flow of the design cycle a little easier. Design projects can become cumbersome if not managed well. The objective of the progress reports is to ensure that designers follow a systematic process, plan out ahead, anticipate potential problems and encourage exploration of design alternatives. If we find any obvious design errors, it is much easier to modify your design at a pre-layout stage rather than wasting time and effort in drawing the layout for the erroneous design.

1 Task Allocation Statement

Each team should meet as soon as possible and discuss the overall project planning, schedule and tasking. The technical work should be divided among the team members as evenly as possible. In addition, report writing responsibilities also must be divided as evenly as possible.

In addition, select one member of the team as the project coordinator. The coordinator will take the initiative to keep the project ticking along, ensuring that the deadlines are being met by the team members in accomplishing various subtasks. Identify coordinator on the coverpage. The coordinator will of course share the other task responsibilities in a fair manner.

Your first deliverable is a task allocation statement which should include the following:

1. Coverpage with students' names (identify who the coordinator is), email addresses, phone numbers, project title, name of the chip and the date.

2. Tasks, subtasks, responsibility assignments, milestones and schedule for your project. The project should be divided into tasks aligned with the progress reports and the final report/tapeout. Each task should be divided into detailed subtasks. Each subtask should be assigned to a member of the team. Inter task/subtask dependencies must be identified. Milestones are events that indicate the deliverable at the completion of a task or subtask and the sender/recipient of that deliverable and schedule is a time before which that event should happen (eg. VHDL model delivered by John to Joe by 6pm. Nov 21).

What I am looking in this report is as detailed a thought as possible into project planning towards successful completion. The report doesn't have to be long but should indicate careful planning and proper division of work.

Although the following progress reports are required by the deadlines announced, you can certainly plan to make more progress (and are advised to do so) than needed by these reports in order to complete the project earlier and avoid rush in the last few days.

2 First Progress Report

Your first progress report consists of the following elements:

1. Coverpage with students' names, email addresses, phone numbers, project title, name of the chip and date.
2. A name for your chip. Choose any name using no more than 12 characters from the alphabet (no blanks and special characters).
3. Pin-out diagram for your chip (interface).
4. Explanation of how the chip works from a user's perspective with reference to the pins. Use cycle-level timing diagrams as necessary.
5. Inclusion and explanation of the test mode in the above items.
6. Description of the major design decisions made, important design alternatives considered, if any, and rationale for the design decisions.
7. A logic level design for your entire chip presented hierarchically. The leaf-level gates should be implementable as single-stage CMOS gates.
8. Logic gate level VHDL models where each gate is modeled behaviorally and those components are instantiated and connected in a hierarchical fashion to make the structural model of the entire chip, including the test mode. Submit models and test benches with sufficient tests and annotated simulation results.

9. A top-level block diagram of your architecture, including an overview of the bit-slicing scheme used describing the in/out signals of the bit-slice and how the slices will be connected.

10. Explain how the work was divided among the team members.

3 Second Progress Report

Your second progress report consists of the following elements:

1. Coverpage.

2. The entire first progress report revised and updated as needed to match the progress to date and reflect any new decisions and changes made since then.

3. Transistor circuit diagrams and layouts for the bit-slices and their IRSIM/Spice simulation results.

4. VHDL models with back-annotated timing data. The leaf level components should include delays obtained by the above IRSIM/Spice simulations. Using the chip level simulations in VHDL, you should be able to predict the overall timing, clock speed, throughput rate etc. Submit annotated VHDL models, test benches and simulation results.

   Compare these simulation results with those done in the first progress report and comment. Revise the simulation models of the first progress report if necessary.

5. A floor plan (within the pad-frame) that shows the placement and routing among the major modules. In case of bit-sliced designs, inter-slice routing must be shown. All control logic and test logic must be shown as well.

6. Description of the major design decisions and/or design revisions made, important design alternatives considered, if any, and rationale for the decisions.

7. Explain how the work was divided among the team members.

4 Final Report

Your final report should consist of the following elements.

1. Coverpage. Should include name, date, project name, and phone numbers where the students can be reached after the project is submitted. The

full path of the CIF file should be included as well. Please make sure this path is correct.

2. The entire first and second progress reports revised and updated if needed to match the progress to date and reflect any new decisions and changes made since then. Revise the logic design, VHDL models (especially delays), floor plan and component layouts to match the final state of the project.

3. Chip level layout in the form of a color plot (you will get instructions about how to get these plots later).

4. Detailed Users' Guide for the chip including description of the function of all the pins and the timing diagrams that show how the chip should be used.

5. Complete test strategy for the test engineer who will test the chip later.

6. Description of the chip architecture, design decisions and the rationale for the decisions. Revisions made from the second report should be specifically addressed with suitable explanations.

7. IRSIM/Spice simulation results for the normal as well as test mode operation of the chip, including the pads. Annotate the results with comments so that they can easily be understood. Make sure that your simulations cover all functionality of the chip. Explain the simulation strategy you have used in identifying your simulation tests and what the critical paths are. Report the expected maximum clock frequency (based on your simulations of the critical paths).

8. Final VHDL simulation results with an explanation of any timing/event discrepancies between those and the IRSIM simulations.

9. Explain how the work was divided among the team members.

5 A Note on Test Logic and Test Mode

These will be discussed in the class, but here is a summary.

1. Include two standard cells (an inverter and a flip-flop) on the spare pads, *completely* isolated from the rest of your chip. You can test these cells independently of your design to ensure that the die has no major structural flaw and that it is alive.

2. In case of bit-sliced designs, include a stand-alone bit-slice on the spare pads so that it that can tested in complete isolation from the rest of your design.

3. Provide a test-mode/normal-mode control input and provide a complete scan path in the test mode. That is, in the test mode, *all* the flip-flops in your design should form a shift register with the shift-in and shift-out ends accessible from the pins. More on scanning will be discussed in the class.

4. Tap various interesting and strategic signals (such as clock at remote locations from the pad, global buses if any, outputs of the first and second slices, outputs of other selected slices, outputs of selected flip-flops/registers etc.) and bring them out to the spare pins to make them observable. By observing these signals you might be able to say whether your chip is partially working.

6 Some Notes on the Design

1. You must do Spice simulations for all building blocks and bit-slices. For the overall chip testing IRSIM simulations are acceptable.

2. You can use the standard cells made available via the Blackboard system. You may also develop your own full-custom layouts. If you do, make sure to include well contacts.

3. Make sure that you have progressive sizing of global wires such as VDD, GND, Clocks, etc.. They should be at least 8 lambda wide coming out from the pads and can not be reduced to less than 4 wide. (In other words, 4 is the smallest width for these global wires) Also, buffer the global wires as necessary.

   Note that the pads in the pad frame already have buffers in them.

4. Do *NOT* flip or rotate the pad frame. This can lead to confusion and it is easier to keep all the chips in a standard format so that they are easy to test. This can also lead to rejection of your design by MOSIS or requirement to supply additional bonding diagrams.

5. When you get the final design done. You *MUST* do the following steps to ensure your chip operates correctly. You must verify that your circuit being submitted to fabrication operates correctly.

   After finishing your design, simulate in IRSIM. Put the design inside the pad frame and then simulate "pad to pad". (This means simulating your design inside of the pad frame) If this works correctly, write your file out as a CIF file. (:cif write NAME ostyle lambda ....) Exit magic. Reload magic and read the CIF file in. (:cif read NAME istyle lambda ..), extract it, and re-simulate it. If this works, then congratulations, you have a chip that is ready to be submitted to fabrication. Submit the cif file.

Warning: :cif write does hierarchical writing. If this is giving you problems with labels or design rules, use a :cif flat to produce a flattened CIF file. Since :cif flat removes labels, you must relabel for simulation. You might find it convenient to keep a cell with only the labels that you can overlay on top the chip layout without labels.

Make sure you use proper istyle/ostyle format with :cif read and write corresponding to the technology you are using.

6. Logos can be displayed in metal 1 or metal 2 only. Do NOT connect the logos to any part of your design.

7 Grading

We will use the following scheme to grade your projects. Your project grade is divided as follows. Your task allocation statement has 5% weight, first progress report has 25% weight. Your second progress report (which includes a suitably revised first report as described above) has 30% weight. Your final report (which includes revised first and second reports) and demo together have 40% weight.

8 Other Logistics

Keep checking your email for additional instructions, clarifications, FAQ etc.