

UNIVERSITY OF CALIFORNIA AT BERKELEY
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Design Reviews

Please read through this carefully. There will be NO exceptions.

Due: (CLARIFICATION)

During the **first 10 minutes** of your lab section the week immediately following the lab lecture that describes the checkpoint. “Your lab” means the lab section we have recorded for you on the grade book, which *you* chose during the first 5 weeks of the course. There will be no more lab changes. The TAs for that lab section will collect all the designs and come around to speak with each team (10-15 minutes per team), allowing you the opportunity to demonstrate your understanding.

You will receive a **maximum of 50%** on your design review up to a week late. After the solution is discussed in the following lab lecture, **you will receive no credit.**

Note: You may still come to office hours or other lab sections for help or advice, but you may only turn in design reviews and get checked off as described above.

Example:

Week 0: Friday – Lab Lecture 1: Checkpoint 1 (Lab 4 solutions discussed)

Week 1: DR1 due in labs for maximum of 100%

Friday – Lab Lecture 2: Checkpoint 2 (Lab 5 solutions discussed)

Week 2: Checkpoint 1 due in labs for a maximum of 100%

DR2 due in labs for a maximum of 100%

DR1 due in labs for a maximum of 50%

Friday – Lab Lecture 3: Checkpoint 3 (Checkpoint 1 solutions discussed)

Week 3: Checkpoint 2 due in labs for a maximum of 100%

DR3 due in labs for a maximum of 100%

DR2 due in labs for a maximum of 50%

DR1 earns 0%

Week 4: ...

Grading:

Unlike the method of lab grading you have seen so far, the Design Reviews will be graded on a **continuous scale from 0 to 100**. The grade from the DR will account for 40% of the total grade for each checkpoint. Before you ask your TA for a review, **make a copy of your design**. This way you can get started with coding while we grade your design.

Requirements:

For design reviews, we want a pencil and paper design of your next checkpoint. This means drawing block diagrams or schematics (**NO VERILOG**). If you are designing a state machine, draw the state transition diagram with all inputs/outputs/states labeled

clearly. For all other logic, including state machine controllers, draw schematics as detailed as possible, including wire/bus names and widths whenever possible. But this does not mean you need to draw a million gates and/or registers for a complicated design. For example, if you going to be instantiating multiple copies of a module, first draw out how to implement that module. Then after that, you can just use a block with the module name to represent it for each instantiation.

The key is to make your diagrams **as clear and self-explanatory as possible**. We should be able to understand the full functionality of your design without you to explain it. It should be detailed enough such that we would be able to implement your design without further instruction. This will not only show us that you actually know what you plan to do, but it will also help you tremendously when it comes time to write the Verilog.

In summary:

GOOD

- Block diagrams
- Bubble and arc
- Schematics

BAD

- Verilog
- Pseudocode, etc.