PROJECT 2A GUIDELINES

1) **Project 2 is an Group project. You are allowed to have Design specification / general verification related discussions with other groups. However, discussions relating to BUGS or any CODE SHARING is strictly prohibited and would be considered a violation of academic integrity.**

2) **THE FULL LC3 INSTRUCTION SET, including but not limited to ALU, Control and Memory instructions are valid for this project, and you are required to verify it.**

3) **The Design Specs for this project can be found at :**
   http://www.ece.ncsu.edu/asic/asic_verification/shared/Project2/Project2_Spec.pdf

4) **The common class message board is only for specification/testbench/verification related questions. No BUG related queries should be posted to the message board.**

5) **You testbench should be based on the methodology/environment/architecture suggested in LABS 3 and 4. Different layers of the testbench should be distinctly visible as in the Labs. Inter-process communication / mailboxes and object oriented approach both on the Transmit (Like LAB3) and Receive side (Like LAB4) of your TB would be strictly enforced.**

6) **It is compulsory to have a Golden reference model of each block of the DUT for Project 2.**

7) **This project would require you to get feedback from the TAs for each bug you find. Each group would have their own message board, and the Bugs have to be reported on their own message board. The procedure for this is discussed in details in a later section in this document.**

Project Requirements:

The Project 2A has been divided into 2 subparts. Each part would have a separate deadline, and hence would need to be submitted separately.

Submissions:

1) **Submission 1: DUE - Friday, November, 7th, 2008 at 10 pm EST.**
   This part requires you to have modeled your complete testbench as per Lab3 and Lab 4, plus a Golden reference model for at-least one DUT sub-block. This part also

requires you to have reported at-least 2 bugs to the TAs using your group message board.

2) **Final Submission: DUE - Monday, November, 17th, 2008 at 10 pm EST.**
This is the final deadline for the Project 2A. All the bugs you have found would be frozen here. You would be required to submit a Fully Commented "re-usable" Testbench by this deadline , to the ToTA folder, and upload a Bug report including all bugs using wolfware submit utility on the course webpage. The Final TB should be strictly as per Lab3 and Lab4 as described above, and should contain Golden Reference models for all DUT blocks verified.

The design blocks for the project would be uploaded to your respective group lockers in a folder called FromTAs. Please do not edit the contents / write anything to this folder. You are expected to copy the contents from this folder to use for your design. For each valid bug, the TAs would upload the fix here.

BUG REPORTING PROCEDURE

For each bug you find, create a new topic in your respective group message board, and name the BUG #. For example, for Bug 1 create a Topic named "BUG 1". Use the bug format given below. For the first 2 bugs, you would get 3 turns to report the correct analysis. For each subsequent bug, you would get just one turn. It is on the discretion of the TA to give you additional turns, if he finds that you are "almost there". After your turns are over, the TA gives you the fix. You get points for analysis only if you report the correct analysis within the given turns. All communication for a given Bug should be held within the respective Bug# message board.

Also, for each bug you find, move your TB with the bug and the active checker to ToTA under a folder named Bug# (# is the bug number). The TAs would be running a script which would access /GrpX/ToTA/Bug#, and not finding your files in this location would call for penalty.

Within 24 hours of posting your Bug, you can expect a response from the TAs. For quality response from the TAs, you are suggested to start early.

cont…

BUG REPORTING FORMAT.

a) Design Input for Bug to Appear:

E.g. ADD, R4, R3, R2 or more instructions if necessary

b) Expected Behavior:

E.g. In Execute, Aluout should be ________ for this instruction because .....

c) Observed Behavior

E.g. Aluout was found to be__________

d) Detailed Description of error/ Bug Analysis

Good bug reports are as important as catching the bugs. Remember that the errors need to be very specific. You CAN NOT say things like "aluout is wrong" (that's too generic). You will have to try multiple inputs and output combinations to determine the exact error. The marks each bug would be distributed among all part in the given format.

IT IS COMPULSORY TO HAVE A COMPILATION COMMAND NAMED "compile.do" AND A SIMULATION COMMAND named "simulation.do" PRESENT WITH THE FILES FOR EACH BUG YOU REPORT.

compile.do

```
setenv MODELSIM modelsim.ini

rm –rf mti_lib

vlib mti_lib

vlog ….

vlog –sv …..
```

<u>simulation.do</u>

```
vsim –c –novopt …..

run xxxx.

// No waveforms please. Runtime should be enough to display tour error for the bug.

// remember to use the –c with the vsim
```