

# Project Planning, Proposal, Milestones

ECE532S: Digital Systems Design

Spring 2011

**Note: This document is being updated. The Proposal section is complete.**

## Timeline

**Draft Proposal Due** Monday, January 17, at start of lecture.

**Proposal Feedback and Set Milestone 1** Wednesday, January 19. During the lab, a TA will discuss your lab proposal and help you determine the first milestone that you will incorporate into your formal proposal and demonstrate later.

**Formal Proposal Due** Wednesday, January 26 in lab. This is the formal proposal as outlined below. Keep a copy for yourself as we will keep the proposal to start a file for your project.

**Weekly Progress and Milestones** Each week you will negotiate milestones for the next week. Each member will be responsible for at least some aspect of a milestone and you will be evaluated on your actual progress the following week.

**Demonstration** April 6 lab.

## Draft Project Proposal

Briefly describe the project you want to implement. The basic guideline is that your project should incorporate at least one MicroBlaze processor and a hardware block of your own design.

Give a list of functional requirements and features.

Provide enough details so that you can discuss your proposal with a TA and so that the TA can give you some feedback on whether it might be too hard, too easy or does not meet the requirements. This can be very rough and even in point form. A few sketches of block diagrams will usually help explain your ideas much more easily. You might do this work on a napkin, but it might be better to copy it onto another piece of paper so that it is easier to read!

## Proposal

This is a template for your project proposal. It does not have to be in perfect prose (point form is okay). However, it should clearly convey the intent and planning for the project. Please provide **all** of the information requested in this template.

The proposal is worth 10%. Marks (2 per day) will be deducted for late submission. Please pay attention to the requirements for each section. All information requested is required.

## Project Title

## Project Team

It is expected that a typical team will consist of three members.

List your team members.

## Project Description

Describe the project to be implemented. The basic guideline is that your project should incorporate at least one MicroBlaze processor and a hardware block of your own design.

Describe the **functional requirements and features** that you wish to achieve. Describe the **acceptance criteria** that will be used to determine whether the requirements and features have been satisfied.

Include a **system block diagram** that shows all of the major components in the system. See the link on the course web site:

<http://www.eecg.toronto.edu/~pc/courses/432/2008/handouts/blockdiagram.pdf>

for a good example of what is expected in a block diagram. Indicate which components will be the reuse of existing IP, and which components will have to be implemented. Briefly describe each component.

Note that a component can be hardware or software, i.e., a block in your block diagram could be hardware or software. Depending on the project, you may need one diagram to describe the hardware system, and one to describe the structure of your software.

The diagram and its description shows the work that you have put in to figuring out how your system is supposed to work and it should help you estimate the amount of work required.

## Resource Requirements

List other resources you will need such as microphones, VGA monitors, or video cameras. Microphones and VGA monitors are readily available in the lab. You will have to provide anything else you require, like cameras, but it is good to know the list.

## Milestones

Give an initial estimate of the milestones you hope to accomplish each week for the project starting from now till the demo. If you know how to do a Gant chart, this would be a good addition, but you should also provide a bit of explanation of each milestone.

**Feb 10** Choose this milestone well. Indicate your planned progress towards your first milestone and what you plan to show or demo.

**Feb 24** By this milestone you should have significant progress in your design most likely demonstrated by a working simulation of some hardware. We will build and test blocks *widget A* and *widget B*

**March 3** Complete and test code for software *XYZ Control*

**March 10** etc.

**March 17** etc.

April 7 Done! Final demo.

## Milestone Progress Reports

For each lab, starting Feb. 10, you should prepare a one-page progress report and email it to the TA managing your group by midnight of the evening before the lab or else you will lose two marks if it is late. The report should be in plain text format, no pdf, no Word files, just plain old text! This will make it easier for the TA to annotate comments.

The progress report should be based on the milestones that you set with your TA the previous week. For the first lab, your first milestone will be what you wrote in your formal proposal, or based on a subsequent revision agreed to by your TA. Each week, you will set new milestones for the following week and update your future timeline of milestones if necessary

At times, it may make sense that individuals in the group have different deliverables for the milestones. Separate reports should be sent, or the individual deliverables should be clearly identified. In this case, separate grades will be assigned to each member.

You may also find that you will have a very busy week and will not get anything done. It is acceptable to promise no milestones, but no grade will be assigned either. In this case, milestones should be set for two weeks, instead of one week, and the grade value for those milestones will be doubled. It is at the discretion of the TA to allow this depending on his/her belief that you can maintain the average pace required to finish your project.

The milestone progress report grades will account for 10% of the final grade.

The report should provide the following information:

- Progress towards that week's milestones. If a significant delay is incurred, discuss how you will accommodate it. It may be something like, "Will have to delete feature ABC."
- Proposed milestones for the next week. This may be amended after discussion with the TA.

It is quite possible that you will not meet a milestone. In that case, be prepared to explain what you did during the week. Struggling with a bug and describing how you wrestled with it is acceptable as a delay in your milestones as long as your approach was significantly better than random guessing or trial-and-error.

After reviewing the milestones with the TA, a grade will be assigned according to the following rubric:

Grade	Criteria
10	All milestones met
9	Milestones not met, but delay is understandable and well-explained Suitable adjustments to milestones have been made
8	Milestones not met, but delay is understandable, not well-explained and/or no suitable adjustments to milestones have been made
7	Milestones partially met and probably could have been met with more work Good reasons for delay, catch up plan or adjustments to milestones presented
6	Milestones partially met and probably could have been met with more work
3	No milestones met, but some effort demonstrated
0	No milestones met, no work done

## Overall Project Strategy

Your overall project grade will depend on how well you meet your original milestones, or, if you encounter difficulties, how well you make adjustments by the end of the project. It is most important to be able to show something working by the final demonstration.

Choose your project in a way that you can afford to *slip* (four-letter word that managers hate to hear) a few weeks and still show something that works at the end.

This means that you should have several working stages, each with some additional functionality or features. If you have some unforeseen delay, you should be able to still show something that works in the end, even though it is

not the full project you initially proposed. This is an important part of planning a project: get some basic stuff working as soon as possible. Add fancy features, optimize your circuits, and make it beautiful later. Time to market (time to marks?) is often more critical (how to make money) than being perfect (the dreaded perfectionist engineer). It is better to build something that works with basic functionality, than build something pretty that does nothing.