Date: 9/24/17

# Project Plan
# for
# Blue's Clues

**Distribution and Project Leads**

Bucknell University
Cole Whitley - Scrum Master
Andrew Capuano - Developer
Jason Corriveau - Developer
Sienna Mosher - Developer

# Overview

Blue's Clues is a team consisting of three developers and a scrum master that has one objective: create a pipelined CPU in Verilog that can appropriately process machine code. Our goal, as the team members of Blue's Clues, is to complete this project within four weeks for our customer, Professor Alan Marchiori. Furthermore, we aim to build off of our experience and code from project 1, the single cycle MIPS processor in order to create the pipelined CPU.

The rest of this document goes into how our team plans to remain organized, stay on schedule, and verify that our code is working as expected throughout the development process.

# Roles

Team members will be split into one of two roles: Scrum Master and Developer. As said above, Cole will be the team's Scrum Master and Andrew, Jason and Sienna will work as developers. The role of the scrum master is to facilitate the scrum methodology within the group. Cole will lead bi-weekly sprint meetings and ask team members about their progress, what difficulties they have run into (so that the group can help overcome them), and their plans for what work they will complete next. He will help the team decide what the content and scope of their next sprint will be, as well as keep the team on track and avoid distractions. As developers, Andrew, Jason and Sienna will take Cole's direction and use them to plan, write code, or complete a task that helps the team finish the sprint as quickly as possible. All team members will be responsible for developing the software, creating tests for all modules and documenting the code as they write it. Also, all team members will work together and communicate often so that work is completed successfully and in a timely manner.

# Organization

In order to keep our team organized, we plan on leveraging multiple process management tools. First and foremost, we will be using Slack for team communication. We plan on implementing a simple form of Kanban using Trello as our Kanban board. Github will be used for revisioning and we plan to use pull requests to ensure that we do not muck up out git. Furthermore, team members will work in branches off of master at all times. These branch names will be formated "feature/<last-name>/<feature-name>". We plan on having at least two meetings a week - on Tuesday and Thursday. During these meetings we will discuss the work that we have completed and any roadblocks that we may have encountered. This will be very collaborative and will help us brainstorm as a team to solve this problem. There should never be more than 4 cards on the Trello board "in progress" section at a time (1 card per person). When the card is determined finished by whoever is working on it, the card will move to "review" column where a different team member will check the feature submitted. This will server as a type of code review to ensure we are always committing and merging

quality product. As a way to analyze our work, all members will work together to create weekly team status reports that will detail the progression of the project throughout the week. This will be used in order to see where the group starts at the beginning of the week, what the accomplish throughout the week, what obstacles they find as they work, and where the group stands at the end of the week.

## Schedule

The following is a proposed layout as to our projected milestones in order to stay on schedule for the duration of this semester. However, as tends to be the case in development projects, it is difficult to predict the future regarding timelines. As such, this listed out schedule is subject to change, however, we will strive to stick to it as much as possible in order to produce our project on time.

| Milestones | Description | Milestone Criteria | Planned Date |
|---|---|---|---|
| M0 | Start Project | Clear Plan and Organization | 2017-09-21 |
| | Project goals and scope defined, set up Git and organization | Work Contract and Work Plan created, Git, Slack, and Trello created | 2017-09-24 |
| M1 | Transfer From Project 1, Clock Modules Working | Salvage any and all applicable code | 2017-09-28 |
| | Create a base project from project 1, while adding in clock modules for easy debugging later on | Tested and functional code base with project one modules and project 2 clock modules | 2017-09-04 |
| M2 | Add Additional Modules to Project 1 | All three programs function as expected | 2017-10-05 |
| | Add in additional modules to support test functions | Tested and functional code base that supports test functions | 2017-10-11 |
| M3 | Hazards | Fully functioning hazards module, integrated | 2017-10-12 |
| | Hazard Handling and Forwarding properly supported | Architecture reviewed and stable | 2017-10-18 |
| M4 | Create Presentation | Create and Organize a Presentation of Project | 2017-10-15 |
| | Project is “Presentation-Ready”, Clear Documentation, Presentation is created | Presentation created and finalized, ready to go for in-lab presentation | 2017-10-18 |
| M5 | Release and Close Project | Present and Turn in Project | 2017-10-19 |

A detailed Project Schedule. This schedule will possibly vary depending on struggles.

## Testing plan

Our testing plan consists of writing small testbenches for most of our modules as well as a large testbench for the whole module. As we find errors in

certain module connection, we can write small test benches as integration tests of our modules. I say some of the modules only because we plan to reuse many modules written by team members for project 1. The modules that we plan on using from these have already been thoroughly tested by the individual that is providing them. GTKWave will also come in handy as we attempt to debug our program. We can use it to ensure that we are producing expected values at the proper time in the clock cycle as well as define any unknown values. This form of testing is quite incremental, which will help us to ensure that we do not get to the end and have too much difficulty locating any bug.

# Team Policies

The following is a list of the team policies that have been laid out in the team contract and are necessary for the completion of our project. Under each team policy is a quick explanation as to why the policy is important and any other relevant details.

- Java-style comments (detailed inputs / outputs and overall purpose of modules)
    - As we are each developing separate modules, it is imperative that we are all able to understand the purpose of each module. Along with this, the inputs and outputs of the module are imperative to be fully understood as other developers utilize the modules.
- Well named variables that are easy to understand
    - Similar to important Java-style comments, having clearly named variables will help towards developers understanding how the other developers' modules are organized.
- Keep to your assigned tasks - don't work on another part of the project unless other group members explicitly give you permission
    - This includes ensuring each member only has one card "in progress" on the Trello board at a time
- Meet on Tuesdays at 9:00PM and Sundays at 2:00PM meeting in Dana
    - As a rule, we need to set meeting times that we stick to in order to stay on task and establish proper communication with each other. If we do not set weekly meeting times, it is likely that we will have much more difficulty in establishing meetings on the fly.
- Cole will act as our scrum master; he will help the rest of the team in ensuring each team member is on task
    - As a development team, it is extremely important to stay organized. As our scrum master, Cole will help us to accomplish this goal by staying on top of the rest of the group to ensure they are accomplishing their tasks. Furthermore, it is also on the rest of the team to express to Cole if they are having any difficulty and need assistance with their tasks.
- Stay in communication and do your work - we are all busy with other classes and otherwise busy on campus, but we all need to do our fair share of work to complete the assignment on time
    - Most important is to stay active on slack. Try and respond within an hour or two

- Everyone should always be working in their own branch (not directly in master). Then, after other's approve of your work, you are able to merge it into master.
    - Branches will be named "feature/<last-name>/<feature-name>"
- Code reviews will occur during lab periods; the purpose of such code reviews is to ensure code is readable and operating.
    - The code to be reviewed comes from the "review" column of our Trello board
    - One teammate that did not write the module is expected to review before moving the card to the "done" column