I wanted to provide you with some additional details about the final project modality. The project consists of 4 steps/deliverables as highlighted below. Note that we are also going to have a project checkpoint right after Thanksgiving (12/1).

|  |  |  |
| --- | --- | --- |
| **Phase** | **Date** | **% of project grade** |
| Pitch presentation | 11/10 | 10% |
| Project proposal | 11/15 | 10% |
| Final presentation | 12/13 | 30% |
| Project report | 12/23 | 50% |

**Pitch presentation:**prepare a short presentation (3-5 minutes) highlighting the main goal of your project, background and motivation, proposed resources and deliverables, expected outcomes. Following each presentation, we will have a Q&A session. When appropriate, the feedback you receive during the Q&A session should be incorporated in your project proposal.

**Project proposal:**The proposal should be 2 pages in length and address the following points:

* + What are your objectives? What are the problems you are trying to investigate?
  + Background and Motivation: What have others done in this area? What is the significance of this work? Include relevant references.
  + How do you plan to implement your project? For instance, what software tools do you intend to use/develop to implement this? What experiments do you plan to perform to evaluate your idea?
  + What kind of resources will you need for your project?
  + What do you expect to discover from your project?

**Final presentation:**The target for your final presentation should be 15 minutes + 5 minutes for Q&A. The outline should touch the following topics:

* + Introduction and motivation
  + Your proposed solution/implementation for the problem you are addressing
  + Discussion of your experimental results
  + Conclusions and possible future directions

You should put particular emphasis on your methodology and results (even if some experiments did not turn out as expected).

**Project report:**The report should be 6 pages (including references) in a two column format and using 10pts font. If you are looking for a template for your report, you can find one [here](https://www.overleaf.com/latex/templates/ieee-conference-template/grfzhhncsfqn). The outline is going to depend on the nature of your project. However, you can use the following list of sections as a guideline to structure your report:

* + Abstract
  + Introduction
  + Background and related work
  + Methodology
  + Experimental results and discussion
  + Conclusions

Make sure all your plots and figures are readable by using colors with good contrast and appropriate font size for your axes and labels.

Hi Morgan,

I wanted to follow up with some guidelines for your project. Based on what you proposed I think it makes more sense to compare two compatible simulators, i.e. NVSim and Destiny. I am attaching the papers that discuss the two simulators, and I have placed an archive for the destiny code on the EECS server (/ee/193EMT/destiny\_v2.tar.gz).

<https://ieeexplore.ieee.org/document/6218223>

<https://ieeexplore.ieee.org/document/7092634>

Possible steps for your projects would be:

- evaluate NVSim and Destiny across the same memory configurations to capture any potential difference between the two implementations.

- evaluate NVM models exclusive to Destiny (SOTRAM, racetrack)

- evaluate 2D vs 3D memory architectures in Destiny

The goal would be to create a database of results for all the different memory configurations and tools so that different design points can be easily compared when choosing an implementation target for a given compute architecture.

*- Limiting the technology node to 22nm for now should be fine.*

*- I would replace cache with RAM for your comparison, since that will simplify the analysis and remove the cache associativity from the set of parameters you need to sweep.*

*The original NVSim repository (*[*https://github.com/SEAL-UCSB/NVSim*](https://github.com/SEAL-UCSB/NVSim)*) also includes samples for PCM.*

*You may want to add that to the list as well.*

- Beyond comparing the two tools, it’s also important to evaluate the additional features of destiny and run the same simulations for these additional points. In particular, you want to add the following:

- 3D memories (which technologies are supported? RRAM, SRAM(?), STT(?)….)

- Additional technologies (SOT-RAM, eDRAM, DWM)

- Some sample files in destiny include reference to MLC. How is that modeled? Which technologies are supported?

**Goal**:

* 1) Compare NVSim and Destiny cache simulators and create a database of results
* 2) Model destiny 3D functionalities
  + See if MLC actually works

**Experiment**:

Input Params

Process Node: **22nm**

Dimensions: 2D for head to head comparisons

Design Target: replace cache with **RAM** (\*.cfg file)

Cache Sizing: {16, 32, 64, 128, 256, 512, 1024, 2048, 4096} KB sweep

Memory Cells: {SRAM, STTRAM, RRAM, PCRAM}

Optimization Target {either read latency, write latency, read energy, write energy, leakage, or area}

(InputParameter.h)

Output Data

Read Latency

Write Latency

Area

Read Energy

Write Energy

Leakage Power

Calculations

                Percent Difference

Visuals

Tables

Raw results?

Results with percent difference

Charts

Read Latency

Write Latency

Area

Read Energy

Write Energy

Leakage Power

**3D Part 2**

Use [config](https://github.com/rockett-m/EECE0193/tree/main/Final_Project/destiny_v2/config)/sample\_3DReRAM.cfg as template

Change cell files to SRAM.cell / STTRAM.cell / RRAM.cell / PCRAM.cell

See if output is compatible / no errors / works with 3D

config/sample\_3D\_eDRAM.cell

config/sample\_3D\_eDRAM.cfg

config/sample\_3D\_ReRAM.cell

config/sample\_3DReRAM.cfg

**Conclusion**

                Comment on the experiment process

                Highlight notable outliers

                Talk about trends

                Suggest reasoning for any shortcomings or suspicious data

                Mention flaws with tools if apparent

                Make a statement on which tool seems like a better option for 22nm and these cache sim settings

                Talk about the value of this dataset and how it can help researchers (save time)

                Discuss runtime and experiment hurdles (slurm etc)

                Point out configurability differences between cell and cfg files – v important

                Future work possibilities

                Extrapolate trends to smaller process nodes like 14nm?

                Acknowledgements etc