**Abstract**

s

**Goals**

The purpose of this experiment is twofold

Determining if one of the two state-of-the-art simulators, NVSim or Destiny, is better overall – or perhaps in certain domains – given the task of RAM cell modeling.

Also, there is interest in exploring the 3D and high-density capabilities of Destiny to evaluate the tool’s features.

**Methods**

A comprehensive sweep of capacities from 16 - 4096KB, incrementing by each power of two, was used as a consistent x-variable despite other variations.

All eight supported optimization targets ('Total Area', 'Read Latency', 'Write Latency', 'Read Dynamic Energy', 'Write Dynamic Energy', ‘Read Energy Delay Product’, ‘Write Energy Delay Product’, 'Leakage Power') were iterated through in each simulation. In the 2D realm, the RAM cell types: (STTRAM, SRAM, RRAM, and PCRAM) were tested on NVSim and Destiny. For Destiny high-density testing, 3D RRAM cells and MLC RRAM cells were used for simulations.

As for output results, the following fields were parsed from output files: ('Total Area (um^2)', 'Read Latency (ns)', 'Write Latency (ns)', 'Read Bandwidth (GB/s)', 'Write Bandwidth (GB/s)', 'Read Dynamic Energy (pJ)', 'Write Dynamic Energy (pJ)', 'Leakage Power (uW)'). Generally all eight output fields were used in graphing; however a few combinations did not work with the simulators, which is discussed in the ‘Limitations’ section.

The script generated configuration files for each unique simulation, and used the 22nm technology node for the smallest available process technology supported, for consistency.

**Results**

The

For certain metrics - which is more favorable - 3d vs mlc

Point out expectations

What is in line

What is not in line and why?

could just but outlier but might not be

Pick one chart of the 8 optimizations for each result (outlier or most interesting result)

Include in each of the results sections?

* 1. **2D** **NVSim vs Destiny** [**STTRAM SRAM RRAM] {8 results}**

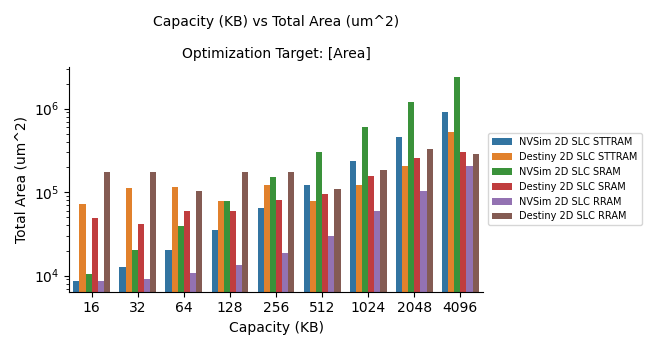
Results: ('Total Area (um^2)', 'Read Latency (ns)', 'Write Latency (ns)', 'Read Bandwidth (GB/s)',

'Write Bandwidth (GB/s)', 'Read Dynamic Energy (pJ)', 'Write Dynamic Energy (pJ)',

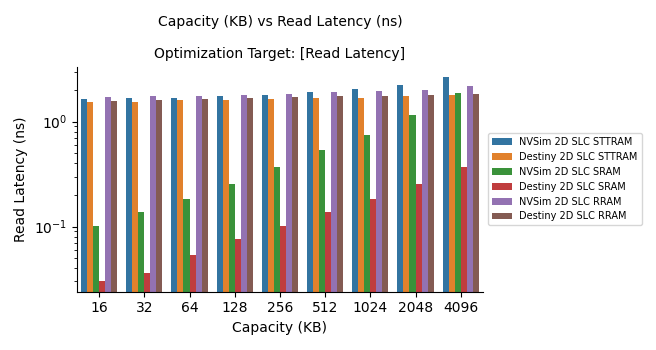
'Leakage Power (uW)')

Pick best of the 8 opt targets for each result category and include chart

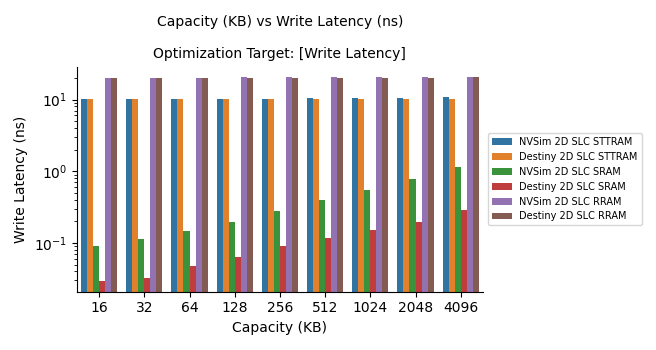
1. 'Total Area (um^2)'



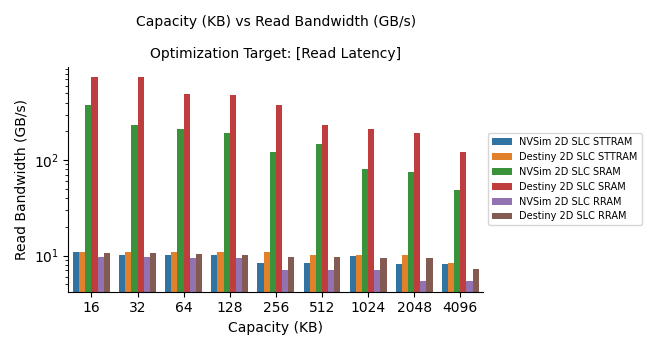
1. 'Read Latency (ns)'



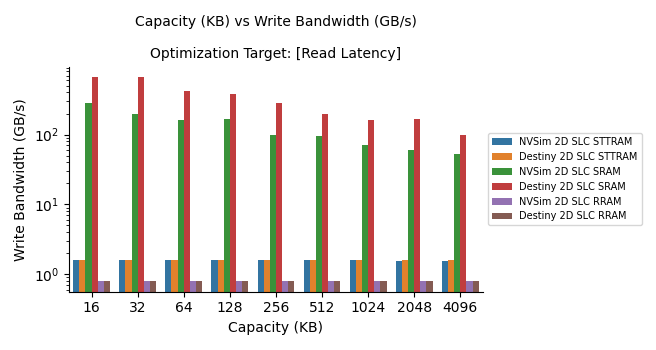
1. 'Write Latency (ns)',



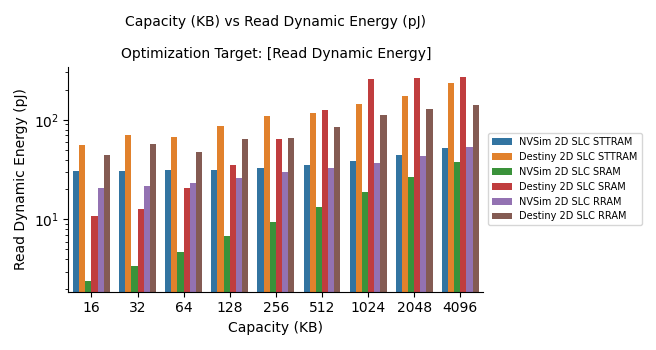
1. 'Read Bandwidth (GB/s)'



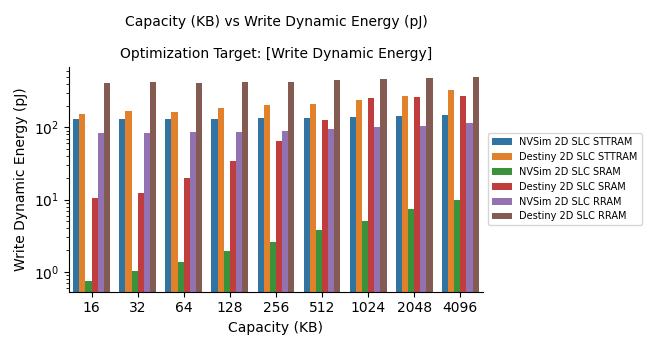
1. ‘Write Bandwidth (GB/s)'



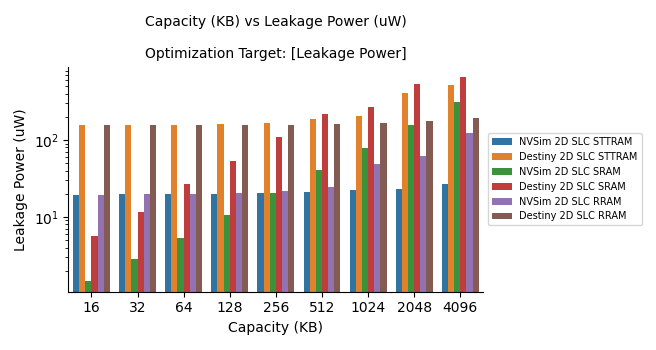
1. 'Read Dynamic Energy (pJ)'



1. Write Dynamic Energy (pJ)'



1. 'Leakage Power (uW)'



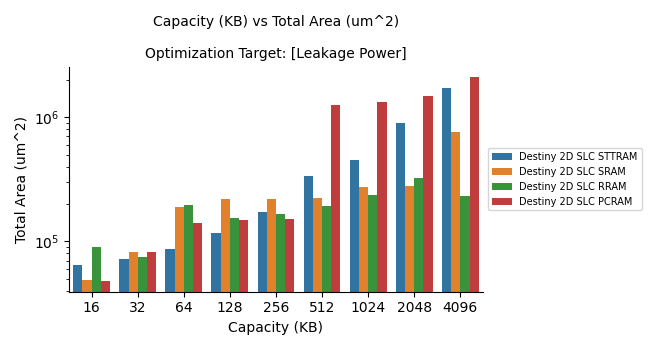
1.2 **2D Destiny [STTRAM SRAM RRAM PCRAM] {8 results}**

Results: ('Total Area (um^2)', 'Read Latency (ns)', 'Write Latency (ns)', 'Read Bandwidth (GB/s)',

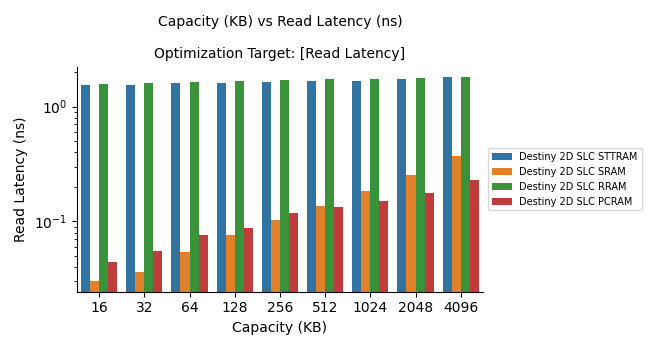
'Write Bandwidth (GB/s)', 'Read Dynamic Energy (pJ)', 'Write Dynamic Energy (pJ)',

'Leakage Power (uW)')

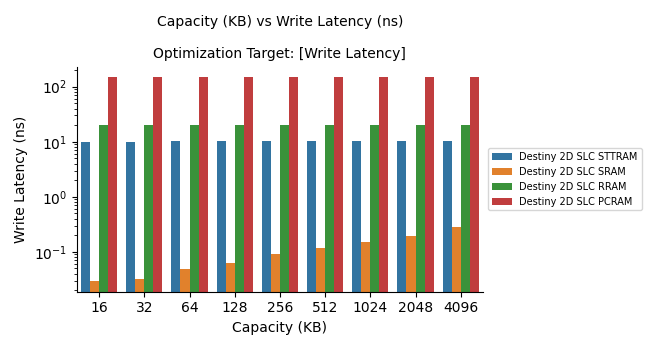
1. 'Total Area (um^2)'



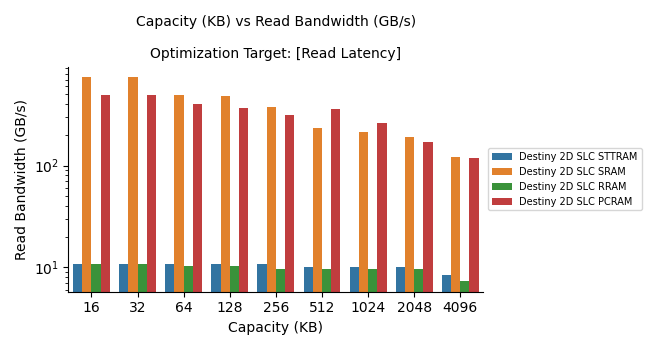
1. 'Read Latency (ns)'



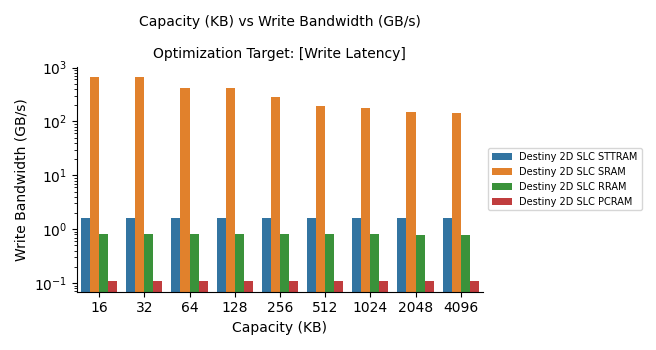
1. 'Write Latency (ns)',



1. 'Read Bandwidth (GB/s)'



1. ‘Write Bandwidth (GB/s)'



Capacity appears to have little influence on Write Bandwidth, if anything is decreases slightly as capacity increases, which could have to do with data access taking up minor amounts of time. This graph stands out the most of the batch, but it makes sense that reducing latency would be among the strongest of contributors to more bandwidth.

1. 'Read Dynamic Energy (pJ)'

Chart, bar chart

Description automatically generated

This optimization target has the largest difference between cell types.

It is interesting to note that RRAM actually uses less energy as capacity increases. STTRAM appears unaffected with capacity, and SRAM energy increases as it scales up. PCRAM is the most dramatic compared to the other cell types but does follow a high correlation with capacity.

1. Write Dynamic Energy (pJ)'

Chart, bar chart

Description automatically generated

The most dramatic of all the graphs has the optimization target of write latency. PCRAM suffers the most here; however none of the memory cell types appear to increase as capacity grows.

1. 'Leakage Power (uW)'

Chart, bar chart

Description automatically generated

This graph shows that RRAM has no correlation with capacity when it comes to leakage power. STTRAM doesn’t show linear correlation with capacity until the larger capacities such as 512KB are simulated. The closest log correlation with leakage power and capacity occurs at SRAM, showing that on smaller capacities, this is the best cell type to use if leakage power is the primary concern. The PCRAM has the strangest correlation to the capacity as it jumps in series every few capacity increases.

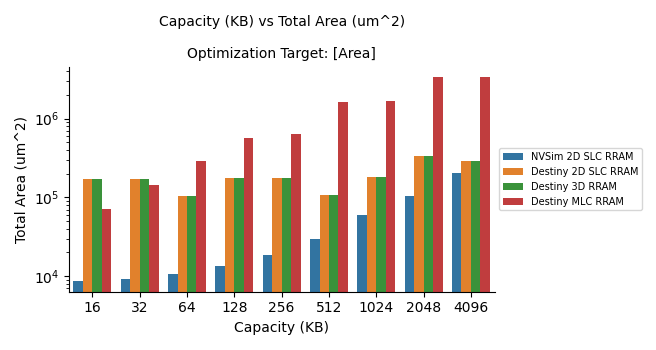
1.3 **2D** **NVSim vs 2D Destiny vs 3D Destiny vs MLC Destiny [RRAM] {6 results}**

Results: ('Total Area (um^2)', 'Read Latency (ns)', 'Write Latency (ns)',

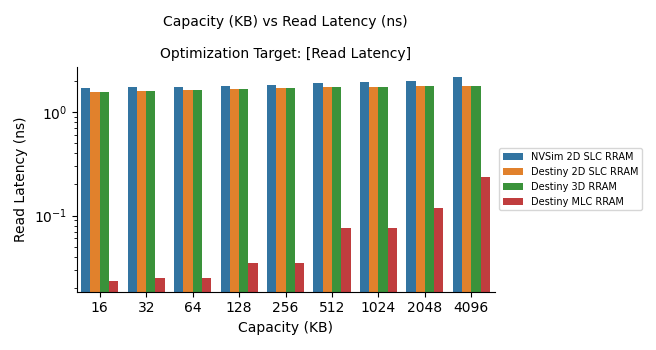
'Read Dynamic Energy (pJ)', 'Write Dynamic Energy (pJ)', 'Leakage Power (uW)')

Pick best of the 8 opt targets for each result category and include chart

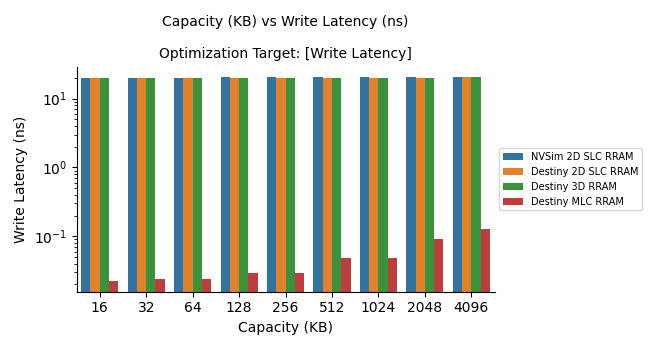
1. 'Total Area (um^2)'



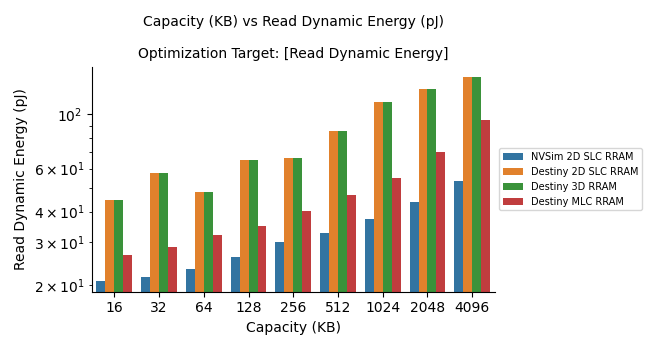
1. 'Read Latency (ns)'



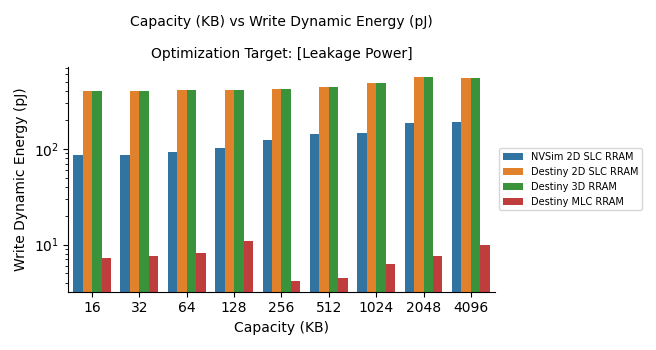
1. 'Write Latency (ns)',



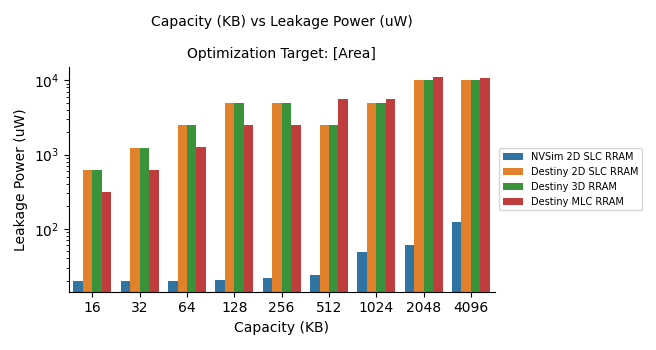
1. 'Read Dynamic Energy (pJ)'



1. Write Dynamic Energy (pJ)'



1. 'Leakage Power (uW)'

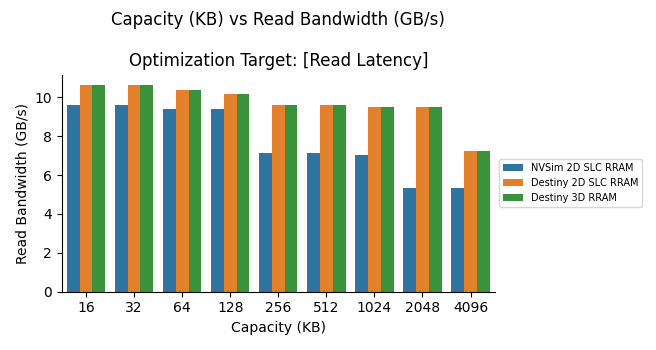


1.4 **2D** **NVSim vs 2D Destiny vs 3D Destiny [RRAM] {2 results}**

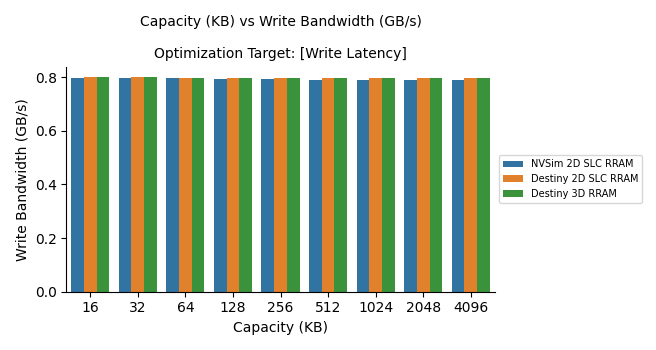
Results: (‘Read Bandwidth (GB/s)', 'Write Bandwidth (GB/s)',)

Pick best of the 8 opt targets for each result category and include chart

1. 'Read Bandwidth (GB/s)'



1. ‘Write Bandwidth (GB/s)'



**Limitations**

Destiny 2D simulations on STTRAM, SRAM, RRAM, and PCRAM worked flawlessly.

NVSim on the other hand, failed to generate data for PCRAM simulations. Looking into the source code of this tool, PCRAM had a note that it was currently unsupported in simulations. So in comparisons, Destiny and NVSim were able to go head to head with three RAM cell types: STTRAM, SRAM, and RRAM.

In the high-density category, the 3D RRAM simulations worked without issue.

The MLC RRAM simulations did not work with Read Bandwidth and Write Bandwidth as negative and infinity values were displayed. The likely cause of this shortcoming is

**Analysis**

3d and mlc both increase density

How can these features improve performance?

Drawbacks and improvements with

With higher density models what do we notice for pros and cons?

For example total area down and leakage power down

Ideally latency down as well etc

**Conclusions**

ddd

**References**

Two papers -

UCSB Seal Library/github

NVSim Source Code

Destiny Source Code

My Github Repo

M. Poremba, S. Mittal, D. Li, J. S. Vetter and Y. Xie, "**DESTINY**: A tool for modeling emerging 3D NVM and eDRAM caches," *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, pp. 1543-1546, doi: 10.7873/DATE.2015.0733.

X. Dong, C. Xu, Y. Xie and N. P. Jouppi, "**NVSim**: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 994-1007, July 2012, doi: 10.1109/TCAD.2012.2185930.