EE193 – Emerging Memory Technologies
Fall 2021
Prof. Marco Donato

Lab assignment #3 – MLC memories in NVSim

In Lab #2, you have taken the first steps to simulate non-volatile caches and memory macros. In this assignment we will continue working on NVSim by customizing the memory model for simulating multi-level cell (MLC) memories.

While the design of MLC memories requires ad-hoc peripheral circuits that would need to be included in the model, for this assignment we are going to adopt some simplifying assumptions that will allow us to simulate MLC memories modifying the existing NVSim code. We will look at two aspects of the memory design that are affected by using MLC devices, namely capacity, sensing. Use the RRAM cell model for all your experiments. By doing so, you can make all your code additions conditional to the `MemCell` class attribute `memCellType` being equal to `memristor`.

Part 1: Improving storage density

The primary reason for implementing MLC memories is that they provide an opportunity for increasing storage density beyond the limit set by scaling the memory cell physical area. By looking at a memory array in isolation, MLC should provide a reduction in array size proportional to $\log_2(N)$ where N is the number of levels encoded in the memory cell. In other words, the physical design of an MLC array having a capacity *C*, is equivalent to the one of a single-level cell (SLC) array having a capacity $C' = C/\log_2(N)$.

1) In the `MemCell` class, add the appropriate attributes to flag an MLC implementation, and the number of bits per cell (these parameters should be set in the memory cell file)
2) Modify the code to compute the *physical* array capacity based on the MLC configuration. The `main.cpp` file is convenient place to insert this function
3) Run iso-capacity simulations for SRAM, SLC RRAM, and MLC RRAM macros by choosing a set of appropriate target capacities. Comment on the results in terms of overall array area.

Part 2: MLC sensing approaches

As we have discussed in class, there are two main approaches for sensing multi-level cell memories, each with its own trade-offs. In order to model the impact of parallel and sequential sensing we are going to focus on the implementation of the `SenseAmp` class provided by NVSim. You can assume that the same sense amplifier design will work for MLC sensing as well.

1) Modify the implementation of the area and read latency functions in the `SenseAmp` class to describe both sequential and parallel MLC sensing. Ignore the contribution of additional logic (latches, muxes, etc.) and assume that the sequential sensing uses uniform sampling intervals.
2) Run iso-capacity simulations for SRAM, SLC RRAM, and MLC RRAM macros by choosing a set of appropriate target capacities. Compare the results with Part 1. How is the sensing circuitry affecting your model area estimates? Does your model correctly estimate the trade-off between parallel and sequential sensing?

In your report, state all the assumptions you have made for creating your model and describe the simulation settings you have used for your experiments. The deliverables should include an archive containing ONLY the modified files and a sample cell configuration file.