ND*nano* Undergraduate Research Fellowship (NURF) 2015 Project Summary

1. Student name: Katherine M Sanders

2. Faculty mentor name: Dr. X. Sharon Hu

3. Project title: Stochastic Computing with Nanomagnetic Logic (NML)

4. Briefly describe any new skills you acquired during your summer research: Working with MuMax3 on a CUDA platform, background research

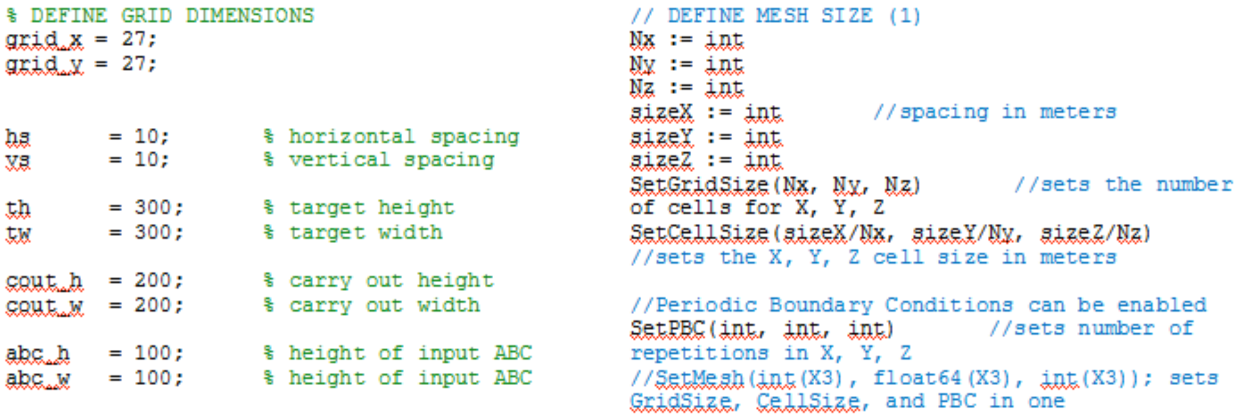
5. Briefly share a practical application/end use of your research: This summer I honed my critical thinking skills by applying different approaches to my research. I read through complex literature to find pertinent information and analyzed two similar types of code. Coding skills will be instrumental in my education as an electrical engineer.

Publications (papers/posters/presentations): N/A

Begin two-paragraph project summary here (~ one type-written page) to describe problem and project goal and your activities / results:

Leveraging stochastic computing (SC) paradigm and nanomagnetic logic (NML) devices is difficult and requires simulations to both lead toward to correct design and to test the design for viability. The majority of our research relies on determining parameters of the design to test, either in isolation or in tandem. Using micromagnetic simulation tools, simulations are run to compare the effect these parameters have on different logic cases and on the adder as a whole (based on the clocking margins), test different NML structures, and use the results to create a functional counter. The micromagnetic simulator we use is Object Oriented MicroMagnetic Framework (OOMMF). This program creates 3D array of the magnetic fields of the design. OOMMF, which is CPU based, takes hours to finish just one nanomagnetic simulation. During last semester I had completed 100 simulations for the 000 case (both incorrect to correct and correct to incorrect) using OOMMF. This summer I continued this by running a handful of simulations for the 001 and 010 cases. The same as last semester, these simulations originated with a MATLab code and were then submitted to OOMMF through the CRC system. The MATLab code required changing the ranges of the magnetic field applied in order for the magnet to switch from incorrect to correct state and vice versa. These simulations take hours to run and pose an impractical approach when hundreds of simulations need to create and validate a design. However, MuMax3 is another micromagnetic simulation program, can run on GPU and significantly accelerates simulation runs. The other part of my summer NURF project is to learn and use MuMax3 with the hope that a nanomagnetic simulation would take just a few minutes.

To begin comparing OOMMF and MuMax3 I first had to learn how to code/ run simulations with MuMax3. It was a new platform that I needed to learn the API for. I started by looking at and running a couple sample simulations to make sure the GPU units were working and providing me with accurate results. I then studied the application program interface (API) and examples provided on the MuMax3 website. Once I had worked my way through it I began comparing the code with the MatLab file used to run OOMMF simulations. Figure 1 shows what the side by side comparison looks like.



```
% DEFINE GRID DIMENSIONS
grid_x = 27;
grid_y = 27;


hs       = 10;        % horizontal spacing
vs       = 10;        % vertical spacing

th       = 300;       % target height
tw       = 300;       % target width

cout_h   = 200;       % carry out height
cout_w   = 200;       % carry out width

abc_h    = 100;       % height of input ABC
abc_w    = 100;       % height of input ABC
```

```
// DEFINE MESH SIZE (1)
Nx := int
Ny := int
Nz := int
sizeX := int         //spacing in meters
sizeY := int
sizeZ := int
SetGridSize(Nx, Ny, Nz)        //sets the number
of cells for X, Y, Z
SetCellSize(sizeX/Nx, sizeY/Ny, sizeZ/Nz)
//sets the X, Y, Z cell size in meters

//Periodic Boundary Conditions can be enabled
SetPBC(int, int, int)        //sets number of
repetitions in X, Y, Z
//SetMesh(int(X3), float64(X3), int(X3)); sets
GridSize, CellSize, and PBC in one
```

Figure 1. One section of the side by side comparison of OOMMF and MuMax3

With what I understood of the codes I successfully analyzed/ recreated about one third of the code from MatLab to MuMax3. This project will take more time and a reintroduction to OOMMF's platform before it can be completed.