- **Project Title**
  Error-Tolerant Image Processing Application Based On Stochastic Logic

- **Team Name**
  Silicon

- **Problem to be addressed**
  Aggressive scaling of semiconductor devices exacerbate issues of reliability. High density of devices leads to more probable occurrences of hard errors during manufacturing. In addition, soft errors including particles striking and ray radiation will also have a larger impact on the high-density chips compared to the chips less progressive scaling and lower densities. A particle striking on the chip, for example, can cause a bit flip. Therefore, it is important to develop a method to enhance the reliability of the system.

- **Why does this problem matter?**
  The modern techniques that we adapt now are limited in ensuring absolute reliability. An error-tolerant system is in urgent need to eliminate various external factors that can cause errors. A single transient fault may cause the whole system to perform erroneously. Unreliability in systems can lead to huge loss in all fields, including economic loss of company as well as potential hazards in real-world applications.

- **Idea/solution to be investigated by the project**
  Our idea is to use stochastic logics to implement image processing hardware such as sharpening/smoothing filters and make it more error-tolerant for transient faults.

  Stochastic logic is error-tolerant due to its nature and can create analog signal states in digital-based systems. In stochastic logic, a normal input will take advantage of randomness and be transformed to a bit stream. The bit streams or wire bundles are digital, carrying zeros and ones, and the signal is conveyed through the statistical distribution of the logical values.

  The advantages of the stochastic logic are:

  1. Error tolerance. A single binary value is presented with a series of bit stream which reduces the impact when transient fault happens. In normal operations, if the significant bit of one value is corrupted, it is likely to have a large impact on the result. However, in the stochastic logics, the impact is minimized by equally distributing value weight to all bits in the bit stream.
  2. Fast computation in specific applications. While the stochastic creates more bits than traditional logics, in some cases it will be faster by taking advantage of statistic concept. For example in Figure 1, the multiplication $y = x_1 \cdot x_2$ in directly computed by using a simple AND gate. Another example is scaled addition shown in Figure 2. Using a single MUX, the stochastic logic can perform a weighted average easily.

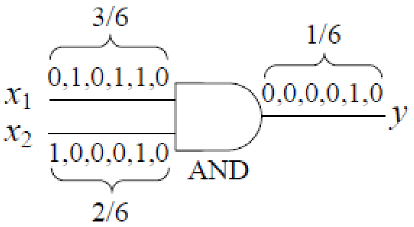

Figure 1. Multiplication in Stochastic Logic

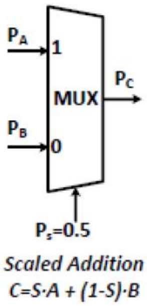

Figure 2. Scaled Addition

How do you plan to develop the project?

We will be using Verilog to simulate and synthesize the implemented hardware. In this way, performance and error-tolerant capability can be analyzed and presented.

How do you plan to evaluate the project's results?

1. Error tolerance: Same proportions of input bits will be flipped intentionally for both the baseline and our design. With the pseudo-error, some pixel values of both designs will be different with the golden output. The similarity (measured by structural similarity SSIM, for example) between original input image and our design/baseline in compared to evaluate the error tolerance.
2. Area: Both the baseline and the project design will be synthesized. We will compare the size of these two designs to evaluate the hardware cost.
3. Performance: We will evaluate the speed of these two designs by comparing the execution time (number of cycles * clock period) for the same picture.

Timeline -should include deliverables at least for each checkpoint.

- Stage1: Create baseline system Oct 23 CK1
    - Stage1-1: Implement basic stochastic logic units (randomizer, de-randomizer) and understand mathematics in stochastic logic
    - Stage1-2: Develop normal image processing hardware (smoothing and sharpening) using standard method (lowpass filter Verilog code downloaded)
- Stage 2: Implement stochastic image processing hardware Nov13 CK2
- Stage 3: Analysis & evaluation Dec 04 CK3

Paper References

1. Qian, W., Li, X., Riedel, M. D., Bazargan, K., & Lilja, D. J. (2011). An architecture for fault-tolerant computation with stochastic logic. *Computers, IEEE Transactions on*, *60*(1), 93-105.
2. Li, P., Qian, W., & Lilja, D. J. (2012, September). A stochastic reconfigurable architecture for fault-tolerant computation with sequential logic. In *Computer Design (ICCD), 2012 IEEE 30th International Conference on* (pp. 303-308). IEEE.
3. Teja, K. B. R., Warrier, A. S., Belvadi, A. S., & Gawhane, D. R. Design and Implementation of Neighborhood Processing Operations on FPGA using Verilog HDL.
4. Brown, B. D., & Card, H. C. (2001). Stochastic neural computation. I. Computational elements. *Computers, IEEE Transactions on*, *50*(9), 891-905.