**YODA – INTRODUCTION & BACKGROUND**

**Introduction**

Our project, P08, involves designing a Function Samples Generator. This will generate a sequence of samples which are determined by a function which has rules on the manner in which the sequence will be produced. The samples given will be encompassed within a given range of values passed in the function.

**Background**

The Function Samples Generator will be based on underlying logic. The framework of the project at hand involves a mathematical approach to the calculation of these samples. A good illustration is probability sampling.

Probability sampling is a systemic approach to select a value out of a given population sample(a range in our case). Four methods are available, which are simple random sample, systematic sample, stratified sample and cluster sample.

A simple random sample involves a sampling method that is unpredictable, every number within the given range has a chance of being selected. Systematic sampling involves choosing values at regular intervals. Stratified sampling is grouping specific intervals and selecting random values from those intervals. Cluster sampling groups the values/population randomly and randomly selects from that population.

While all these sampling methods have their properties, we will be exploring systematic sampling approach. This involves selection of samples governed by a rule. In statistics the samples are chose at regular intervals. In this project, we intend to implement a somewhat edited version of the systematic sampling approach, mimicking the regular intervals approach with formula of our own making or choosing.

The second aspect deals with the characteristics of our samples. Waveform generation functions are perfect for providing a good range of varying values within their given limits. Taking the sine wave as an approach to providing a framework on which we can make our samples from is the simplest solution because a sine wave is the most predictable.

The project is being tested and implemented on a Nexys A7 FPGA, which has numerous capabilities including Block RAM, LED Display Blocks, Push Buttons and switches. Our project will utilise these resources in different ways to optimise the use of the sampling function to the user.

**Objectives**

Given the project essentially involves most aspects of the FPGA, we have decided to tackle the project as follows:

* **A formula to generate an n number of samples within a given start and end range –** This is the core functionality of the project. The number of samples are going to be made, as previously stated from predetermined rules. These rules can range from simple mathematical equations, to token key generation, for example a One Time Pin used for secure Credit Card Transactions.
* **User input capability of the number of samples, the start point and the end point -** although the YODA Project suggests the function be hard-coded, we added a dynamic functionality that allows the user to select a range of values ( from the possible range of integers ) and the number of samples. This is done so the user can have more access and flexibility to the functionality of the generator compared to hard coding the limits.
* **Storage of these values in the FPGA for future retrieval and/or processing –** Saving these values in the system storage is essential. This is going to be done through use of the Block RAM in Vivado which is a dynamic storage space used for temporary and volatile data.
* **Display output(the samples) on the display given by the FPGA –** the FPGA has hardware that provide visuals in the form of Segment LEDs. These will be utilised in our project to display both the input and the output of the system upon implementation.
* **Navigation through storage contents for user analysis of the samples –** Navigation can be achieved through the input of the buttons embedded on the FPGA. These buttons will be configured to navigate through the storage contents of the BRAM and also choose the input parameters of our core function in the system.