LAB Assignment #7 for ECE 525

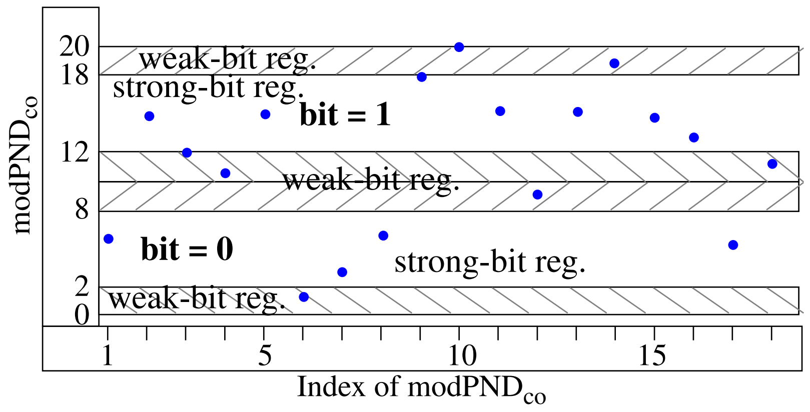
Assigned: Tue., Mar. 21, 2017
Due: Thur., March. 23, 2017

Description: Process modPND$_{co}$ into strong bitstrings

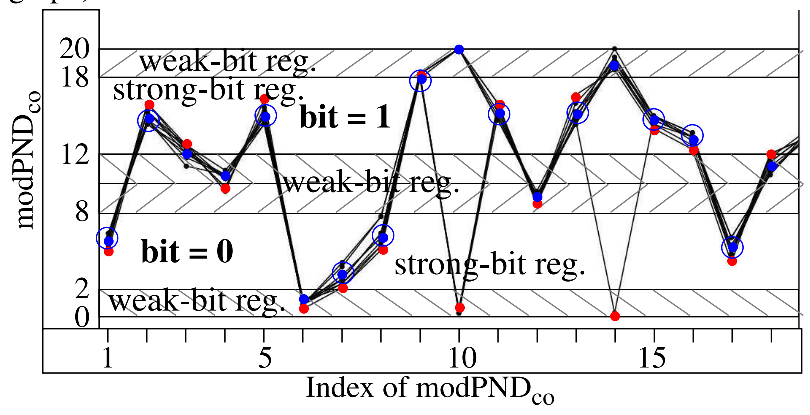
1) This lab adds to the code you created for lab6.

2) From lab6, you have an array of floating point values, modPNDco, of size 2048. Write a routine that computes strong bitstrings as follows:

> The modPND$_{co}$ values you produced from lab6 can be represented as a graph shown below. The x-axis plots the index of consecutative values in your array while the y-axis represents the computed values. In order to avoid bit flip errors, you need to apply a margin technique that excludes specific modPNDco from participating in the bitstring generation process. The margin in the following illustration is set to 2. The margin creates *weak-bit regions* around the bit-flip lines 0, 10 and 20. modPND$_{co}$ that fall within these *weak-bit regions* have a higher probability of introducing a bit flip error during regeneration than those in the *strong-bit regions*.



> For example, if we were to re-collect the modPND$_{co}$ under other environmental conditions, i.e., with different supply voltage values and/or temperature conditions, the values will *shift* as shown by the following set of curves. The red dots represent a ‘worst-case’ shift that occurs to each modPND$_{co}$. The vertical shift in modPND$_{co}$ that are close to the bit-flip lines can result in the modPND$_{co}$ value being re-classified from a ‘1’ to a ‘0’ or vise versa (identify where this occurs in the graph).



Therefore, modPND$_{co}$ that fall within the *weak-bit regions* are to be skipped when generating the bitstring. For example, the *strong* bitstring (SBS) generated by processing the modPND$_{co}$ shown the graph is as follows (NOTE: values that fall ON the line are considered weak):

0	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**0**

So only 11 bits of the possible 18 are actually used. A **helper data bitstring** is also generated that identifies which bits are classified as *strong-bits* during the enrollment process (blue points).

1	1	0	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0
1				5					10					15			

The helper data bitstring is used during regeneration to ‘select’ the modPND$_{co}$ values to be used for generating the bitstring, i.e., **the margins are NOT used during regeneration**. The helper data bitstring does NOT reveal any information about the bitstring itself, which is the secret generated by the PUF.

Apply this technique to the set of 2,048 modPND$_{co}$ values and print out the SBS and helper data bitstrings.