Project for Hardware-Software Codesign

Assigned: Tue., Nov 25, 2103
Due: Last day of class

Description: Combine the components that you have designed to implement a PUF system.

- Use the LFSRs to generate random pairings of numbers which serve as addresses into a table.
- You C program should 'look-up' these random numbers (addresses) in your data file that maps from addresses (can be treated as row/col) into VDC numbers.
- Your sample analysis engine creates a sum from one or more 'samples' (a power of two) and then divides or multiplies the sum to add 3 digits of precision.
- Once two VDC nums are available from the sample analysis engine, start the Bit Generation engine to generate a bit.
- The SRAM public data memory must store the valid bits if the operation is enrollment. Regeneration uses the valid bit memory to re-produce the bitstring.
- Continue this process until a bitstring of length 64 is produced.

Your PUF engine accepts several parameters including (but may not be exhaustive)

- The size of the bitstring
- Seed for the LFSRs
- Number of samples to average
- Function: Enrollment, Regeneration or Authentication
- XMR level

You in class demo should use the following parameters:

- Data table: Use the data table from lab3
- Seed: 1
- Threshold: 60
- Log number of samples: 0 and 4 (1 and 16 samples)
- Number of bits: 64

Basic functionality requirements:

1) Perform Enrollment
2) Store PD in C or RAM -> Bonus for RAM
3) Perform Regeneration

Bonus functionality:

1) XMR
2) Authentication

3) RAM -- PD
4) Band Enrollment
5) Chunks -- Buffered Output

Reporting (bonus points)
1) # of valids comparisons
2) # of invalids comparisons
3) # of redundants
4) PD size
5) Overall statistics -- Performance

Report Requirements:

1) Short 5 to 10 minute presentation.
2) Carry out a live demo of your project in class on the due date.