ECE 5960/6960-010: Hardware Cryptography and Security

Prepared by *Priyank Kalla*
Spring 2024, Homework # 2
Due Date: Monday Feb 19

1) **(40 points) LFSRs, primitive polynomials and stream ciphers:**

   a) Consider the polynomial $P(x) = x^5 + x^2 + 1 \in \mathbb{F}_2[x]$ with coefficients in the finite field $\mathbb{F}_2$. You are asked to check if this polynomial is a primitive polynomial. Describe an approach to test if $P(X)$ is primitive. [Refer to the lecture slides on primitive polynomials and LFSRs]. Write a program in SINGULAR to test if $P(x)$ is a primitive polynomial.

   b) Design a Type-I or Type-II linear feedback shift register (LFSR) using the above $P(x)$ as its characteristic polynomial. Starting with a non-0 seed value (reset values in the LFSR flip-flops should be non-zero), does your LFSR produce a maximal-length pseudo-random sequence? Show the 5-bit sequences produced by your LFSR. You can do this in Verilog (preferable), or Singular, or any other software.

   c) Refer to Fig. 1. Using a 5-bit plaintext P, and a seed (reset) value for your LFSR, demonstrate that your LFSR can indeed be used as a stream cipher to encrypt (compute C) and decrypt (get back P) one-bit at a time. Once again, it is convenient to demonstrate this using Verilog coding and simulation.

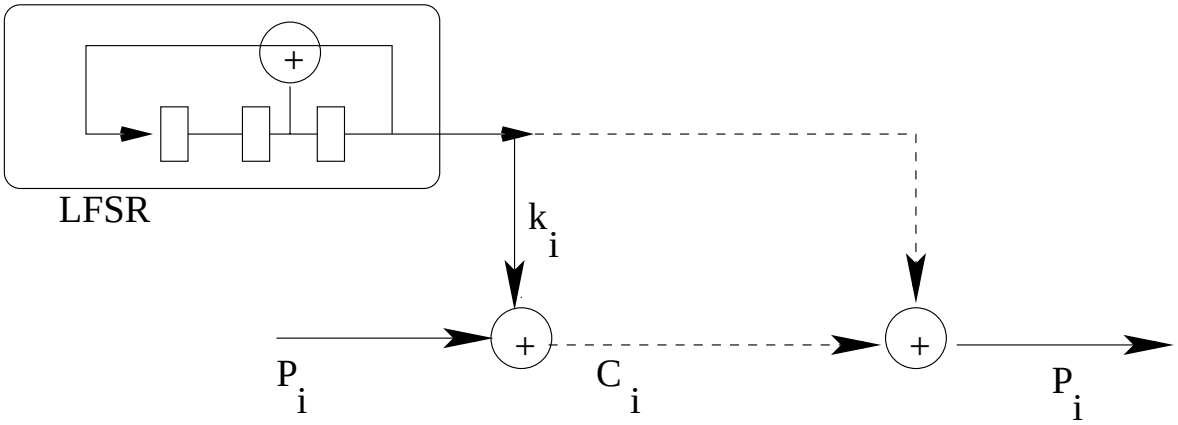

Fig. 1: LFSR and Stream Cipher

2) **(60 points) Mastrovito Multiplier Design:** In this question, you will design a digital logic circuit of a *Mastrovito* multiplier, i.e. the one that computes $A \cdot B \pmod{P(x)}$, as given in my slides. You will implement your design in Verilog, and demonstrate – by means of exhaustive simulation – that modulo-multiplication is being performed. Proceed as follows:

a) We will use the finite field $\mathbb{F}_8 \equiv \mathbb{F}_2[x] \pmod{P(x) = x^3 + x^2 + 1}$ with $P(\alpha) = 0$. Denote the degree of $P(x)$ as k; of course, here $k = 3$.

b) Design a $k = 3$ bit *finite field Mastrovito multiplier* that takes $A = \{a_2, a_1, a_0\}$ and $B = \{b_2, b_1, b_0\}$ as 3-bit inputs, and produces $Z = \{z_2, z_1, z_0\}$ as a 3-bit output. Note that we will have:

$$A = a_0 + a_1\alpha + a_2\alpha^2 \tag{1}$$

$$B = b_0 + b_1\alpha + b_2\alpha^2 \tag{2}$$

$$Z = z_0 + z_1\alpha + z_2\alpha^2 \tag{3}$$

Such that $Z = A \cdot B \pmod{P(\alpha)}$.

c) Give Boolean equations (or polynomial equations $\pmod 2$) of the outputs in terms of inputs, and draw the gate-level schematic. We covered Mastrovito multiplier design in the class when we studied GF circuits. It is given in the slides and in my Book Chapter that I've uploaded on Canvas.

d) Implement the design in Verilog (as a `GFMult(A, B, Z)` module) and demonstrate its correctness via exhaustive simulation.

e) Using any Verilog synthesis and simulation tool, synthesize the circuit into a netlist. Note: if you have taken the "ECE/CS 5710/6710 VLSI Design" course, then feel free to use the Synopsys Design Compiler and map the circuit to a gate library.

f) If you do not have access to the Design Compiler suite, then you may use the Quartus FPGA synthesis tool. The Verilog Quartus manual is uploaded on Canvas. Using any of the `Cyclone V` devices, synthesize the circuit for the selected FPGA architecture.

g) Note down the area and the delay of the circuit. The synthesis tools will give this information in the synthesis report. Area could be the actual area, or in terms of the number of gates, or in terms of the number of LUTs of the FPGA. Delay could be the actual delay, or the topological depth of the circuit.

3) **(60 points) Montgomery Multiplier Design:** Now you will design a MONTGOMERY multiplier for the same finite field as given above: $\mathbb{F}_8 \equiv \mathbb{F}_2[x] \pmod{P(x) = x^3 + x^2 + 1}$ with $P(\alpha) = 0$.

a) First, you will design a Montgomery Block $MM(A, B, \alpha^{-k}) = A \cdot B \cdot \alpha^{-k} \pmod{P(\alpha)}$, as shown in the slides as well as in my book chapter.

b) Then, you will put 4 of these blocks together to design a multiplication circuit that computes $G = A \cdot B \pmod{P(x)}$ – again, as shown in my slides.

c) For the design of a MM block, you should use Algorithm 1 in my textbook chapter, unroll the loop k-times and design a *combinational circuit*.

d) Design the circuit in Verilog.

e) Simulate the multiplier exhaustively to demonstrate its correct operation.

f) Synthesize the circuit, report the area and delay statistics, comparing it with the `Mastrovito` design.

4) **(40 points) This question is for ECE 6960 students. ECE 5960 students can solve it for extra credit**. Let $P(x) \in \mathbb{F}_2[x]$ be a *primitive* polynomial of degree k, with α as its root, i.e. $P(\alpha) = 0$. Let $P^*(x) = x^k \cdot P(\frac{1}{x})$ be the corresponding reciprocal polynomial. Prove (formally!) or disprove (show a counterexample) the following:

a) The element α^{-1} is a root of $P^*(x)$, where α^{-1} is the multiplicative inverse of α.

b) $P^*(x)$ is also a primitive polynomial.

Have fun. I just want to mention that please design these circuits carefully, and preserve these designs. In HW #4, we will make use of these circuits to design point-addition and point-doubling circuitry for elliptic curve cryptography.