

# **Introdução aos Sistemas Lógicos - TW**

## **Especificação do Trabalho Prático em Verilog**

**Profa. Michele Nogueira - michele@dcc.ufmg.br**

**Monitor: Gabriel Amarante - gabriel.amarante96@gmail.com**

<sup>1</sup>Departamento de Ciência da Computação - UFMG

**Data de Entrega: 07 de Julho de 2023**

**(Não serão aceitos trabalhos fora deste prazo!)**

**O trabalho deve ser feito individualmente.**

**O peso desse trabalho é de 30% da nota final.**

### **Visão Geral**

Este trabalho integra lógica combinatória, lógica sequencial e noções básicas relacionadas à criptografia. Considere o seguinte contexto sobre o chamado *Vernam Cypher*. Claude Shannon, da Bell Labs na época, provou em sua pesquisa que o *one-time pad*, devidamente implementado, é inquebrável, resultado publicado em outubro de 1949. Ele também provou que qualquer sistema inquebrável deve ter essencialmente as mesmas características que *one-time pad*: a chave deve ser verdadeiramente aleatória, tão longa quanto o texto simples, nunca reutilizada no todo ou em parte, e mantida em segredo. Em sua forma original, o sistema de Vernam era vulnerável porque a chave (a fita) era um loop, que era reutilizada sempre que o loop fazia um ciclo.

Apesar da prova de segurança de Shannon, o *one-time pad* tem sérias desvantagens na prática pois requer:

- Valores verdadeiramente randômicos, ao contrário de valores pseudo-aleatórios, sendo que isto é um requisito não trivial. Existem geradores de verdadeiros números aleatórios, mas são normalmente mais lentos e especializados.
- Geração e troca seguras dos valores do *one-time pad*, que devem ser de pelo menos do tamanho da mensagem.
- A segurança do *one-time pad* é tão forte quanto a segurança da troca dos valores do *one-time pad*, porque se um invasor for capaz de interceptar o valor do *one-time pad* e saber que ele é um *one-time pad*, o invasor pode decifrar a mensagem.
- Tratamento cuidadoso para garantir que os valores do *one-time pad* continuem secretos e são descartados corretamente, evitando qualquer reutilização no todo ou em parte - portanto verdadeiramente “one-time”.

### **Atividades**

1. Em Verilog, implementar um flip-flop do tipo D. Você deve apresentar a especificação descritiva e comportamental, e testbench. Deve entregar o código da implementação e o print screen do diagrama de tempo.
2. Em Verilog, implementar registradores e *stream cypher*. Assim necessário:
  - (a) Montagem dos registradores contendo *One-Time Pad (OTP)* e mensagem a ser cifrada.
  - (b) Operação XOR para cifragem de mensagens.
  - (c) Decifragem da mensagem.
  - (d) Entregar código, test bench e resultado.

### **Orientações para submissão do trabalho para correção**

- Para as questões que solicitam implementação, você deve entregar o código do seu programa e as respostas. A entrega será realizada pelo Moodle (UFMG Virtual).
- Descreva bem no seu arquivo de código.
- Você deve criar um arquivo .zip (não envie com outra extensão!) contendo todos os arquivos necessários para a correção do trabalho.
- Qualquer dúvida, marque um horário com a monitoria da disciplina ou envie um email para mim.