**- Gerador de Números Pseudo-Aleatórios com LFSR –**

**- Introdução -**

O **LFSR** (Linear Feedback Shift Register) é um registrador de deslocamento que gera sequências pseudo-aleatórias a partir de uma entrada inicial (semente) e realimentação linear de seus bits. Ele opera deslocando bits e aplicando operações lógicas, como XOR, para determinar o próximo bit. LFSRs são usados em geradores de números pseudo-aleatórios, por serem eficientes e capazes de gerar sequências longas com baixa correlação entre os valores.

**- Metodologia -**

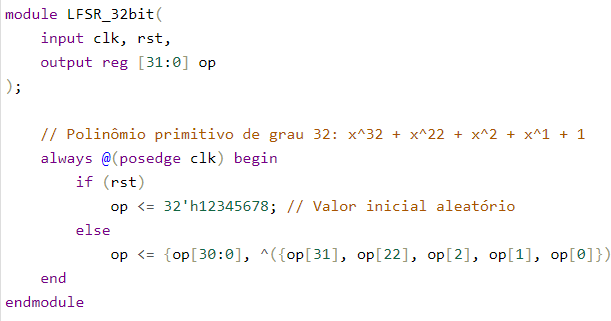
**Circuito LogiSim:**

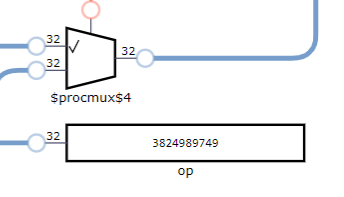
**Componentes:** 32 flip-flops D**,** 4 portas XOR**,** 1 componente clock**,** Conectores.

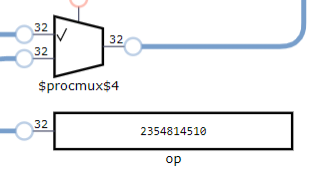
**Conexões: Flip-flops:** Conecte os 32 flip-flops em série, com a saída Q de cada flip-flop conectada à entrada D do próximo. **Portas XOR:** Conecte as saídas dos flip-flops 31, 22, 2, 1 e 0 às entradas de uma das portas XOR. Conecte a saída dessa porta XOR à entrada D do primeiro flip-flop. Conecte as saídas dos flip-flops 31 e 22 às entradas de uma segunda porta XOR. Conecte as saídas dos flip-flops 2 e 1 às entradas de uma terceira porta XOR. Conecte as saídas das três portas XOR a uma quarta porta XOR. A saída da quarta porta XOR será conectada à entrada D do primeiro flip-flop, juntamente com a saída da primeira porta XOR. **Clock:** Conecte o sinal de clock à entrada de clock de todos os flip-flops. **Reset:** Conecte o sinal de reset à entrada de reset de todos os flip-flops. **Funcionamento:** Um LFSR de 32 bits é um circuito sequencial que gera sequências pseudoaleatórias usando um registro de deslocamento e uma função de realimentação linear. Ele contém 32 flip-flops, onde a cada ciclo de clock os bits são deslocados para a direita, e um novo bit entra no registro. Este novo bit é gerado a partir de uma operação XOR entre certos bits específicos (tap positions) do registro. A sequência pode ter um comprimento máximo de **4.294.967.295** estados antes de se repetir. O LFSR é amplamente usado em geradores de números pseudoaleatórios, criptografia e verificação de erros.Foi se modificando o circuito e testando suas mudanças de estado até ser confirmado de que estava em pleno funcionamento.

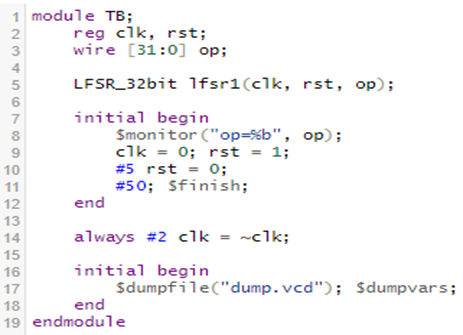
**Código System Verilog:** implementa um Registrador de Deslocamento com Realimentação Linear (LFSR) de 32 bits.

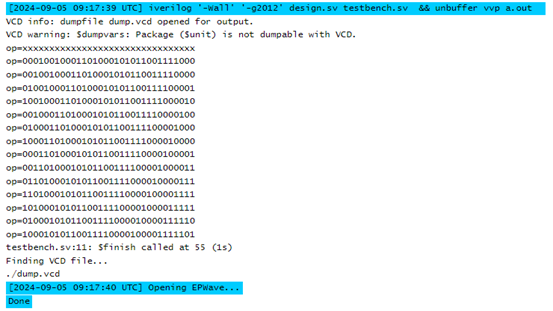
**Componentes:** Entradas e Saídas: **clk:** Entrada de clock. Cada ciclo de clock faz o LFSR atualizar seu valor. **rst:** Entrada de reset. Quando rst é ativado, o LFSR é inicializado com um valor fixo. **op:** Saída de 32 bits que mantém o estado atual do LFSR. **Reset (if (rst)):** Se o sinal de reset estiver ativado, o registro de saída é inicializado com o valor hexadecimal 32'h12345678, que é uma semente inicial aleatória. **Atualização do LFSR (else):** Se o reset não estiver ativo, o LFSR atualiza seu valor da seguinte forma: A nova saída op será composta por 31 bits da saída anterior, deslocados para a direita. O bit mais à esquerda será preenchido com o valor gerado pela operação XOR dos bits op[31], op[22], op[2], op[1] e op[0]. O novo bit a ser realimentado é gerado pela expressão. Aqui, os bits das posições 31, 22, 2, 1 e 0 são combinados com uma operação XOR para produzir o novo bit de entrada. **Funcionamento:** O módulo LFSR\_32bit implementa um LFSR de 32 bits que usa um polinômio primitivo para gerar uma sequência pseudoaleatória. A cada ciclo de clock, o conteúdo do registro é deslocado e atualizado com um novo bit, que é calculado usando a operação XOR de bits específicos do registro anterior. Se o sinal de reset for ativado, o registro é inicializado com um valor fixo. Esse tipo de circuito é útil em aplicações de geração de números pseudoaleatórios e criptografia. **Test Bench:** Um **t**estbenché um módulo Verilog usado para simular e testar outro módulo, que geralmente é o design em verificação. O testbench cria um ambiente de teste para o design, fornecendo sinais de entrada, como clock e reset, e monitorando as saídas do design sob diferentes condições. Ele permite verificar o comportamento funcional do design sem precisar de hardware físico.







****

****

O testbench descrito tem o objetivo de testar o módulo **LFSR\_32bit**. Ele define os sinais de **clock (clk)** e **reset (rst)**, além de uma saída de 32 bits (**op**). O módulo **LFSR\_32bit** é instanciado e conectado a esses sinais. Inicialmente, o clock é definido como 0, e o reset é ativado para inicializar o LFSR com o valor **32'h12345678**. Após 5 unidades de tempo, o reset é desativado, permitindo que o LFSR comece a gerar a sequência pseudoaleatória. O valor da saída é monitorado a cada ciclo de clock, e a simulação é encerrada após 50 unidades de tempo. Um arquivo de simulação chamado **dump.vcd** é gerado para ser visualizado em um analisador de ondas, permitindo observar o comportamento do LFSR durante a simulação. O testbench testa o funcionamento do LFSR fornecendo um clock, controlando o reset e monitorando a saída.

**- Resultados -**

O trabalho implementou um LFSR (Linear Feedback Shift Register) de 32 bits no Logisim, utilizando 32 flip-flops, quatro portas XOR e a estrutura de realimentação adequada. Durante a simulação, o circuito gerou uma sequência pseudoaleatória conforme esperado. O código em SystemVerilog também foi testado com um testbench apropriado, onde o LFSR foi inicializado com o valor hexadecimal 32’h12345678. A sequência gerada foi verificada, garantindo que não houvesse repetição antes dos ciclos previstos. A análise do arquivo dump.vcd, gerado pela simulação e visualizado em um visualizador de ondas, confirmou o funcionamento correto do LFSR e a qualidade da sequência pseudoaleatória, adequada para aplicações que exigem aleatoriedade.

**- Conclusão –**

O trabalho apresentou a implementação eficiente de um LFSR (Linear Feedback Shift Register) de 32 bits, utilizado para gerar sequências pseudoaleatórias. A implementação foi realizada tanto no Logisim quanto em SystemVerilog, e a simulação confirmou seu funcionamento correto de acordo com o projeto. O LFSR gerou sequências pseudoaleatórias de qualidade, essenciais para aplicações em criptografia e geradores de números aleatórios. Sua capacidade de produzir longas sequências sem correlação entre valores consecutivos o torna uma ferramenta valiosa em várias áreas. Apesar da eficiência do LFSR de 32 bits, o trabalho sugere melhorias, como a exploração de diferentes polinômios primitivos e a análise de LFSRs com outras configurações, para aprimorar a adequação da ferramenta a diferentes aplicações. A conclusão destaca o sucesso da implementação e a validade dos testes realizados tanto em ambiente gráfico quanto em código.