



## LABORÁTÓRIO DE CIRCUITOS – CODIFICAÇÃO E SIMULAÇÕES

**ATENÇÃO:** Descrever as soluções (no formato de um relatório técnico) com o máximo de detalhes possível, inclusive a forma como os testes (de unidades ou análise de tabela verdade) foram feitos. Todos os artefatos (relatório, código fonte de programas e outros) gerados para este trabalho devem ser adicionados em um repositório no site [github.com](https://github.com), com o seguinte formato AOC\_Nome1Nome2\_UFRR\_LabCircuitos\_2024.

Implemente os componentes abaixo usando o **Logisim** (simulador digital de circuitos), para cada componente apresente a: (i) Descrição (os pinos e a lógica) do componente e sua funcionalidade; (ii) Imagem dos componentes do circuito integrado; (iii) Os testes do componente; e (iv) Descrição dos testes, apresentando os pinos de entrada, as conexões ativas e o resultado dos pinos de saída.

[COMPONENTE 01]. Registrador Flip-Flop do tipo D e do tipo JK.

[COMPONENTE 02]. Multiplexador de quatro opções de entrada.

[COMPONENTE 03]. Porta lógica XOR usando os componentes: AND, NOT, e OR.

[COMPONENTE 04]. Somador de 8 bits que recebe um valor inteiro e soma com o valor 4.

[COMPONENTE 05]. Memória ROM de 8 bits.

[COMPONENTE 06]. Memória RAM de 8 bits.

[COMPONENTE 07]. Banco de Registradores de 8 bits.

[COMPONENTE 08]. Somador de 8 bits.

[COMPONENTE 09]. Construa um detector de sequência binária para identificar a sequência "101" em um fluxo de entrada.

[COMPONENTE 10]. ULA de 8 bits com as seguintes operações: AND, OR, NOT, NOR, NAND, XOR, SHIFT de 2 bits à esquerda, SHIFT de bits à direita, soma e subtração.

[COMPONENTE 11]. Extensor de sinal de 4 bits para 8 bits.

[COMPONENTE 12]. Implemente uma máquina de estados utilizando portas lógicas.

[COMPONENTE 13]. Contador Síncrono.

[COMPONENTE 14]. Combine portas AND, OR e NOT para criar a lógica de um detector de paridade ímpar (entrada com número ímpar de 1s)

[COMPONENTE 15]. Resolva um problema de otimização lógica utilizando mapas de Karnaugh e implemente o circuito otimizado.

[COMPONENTE 16]. Decodificador de 7 Segmentos: Projete um circuito que converta um número binário de 4 bits para os sinais necessários para acionar um display de 7 segmentos (formato hexadecimal).

[COMPONENTE 17]. Detector de Número Primo: Crie um circuito que detecte se uma entrada binária de 4 bits representa um número primo. Utilize portas lógicas e mapas de Karnaugh para simplificar o circuito.

