

# Desenvolvimento de uma Unidade Lógica Aritmética de 8 Bits em Verilog

Gabriela Correia de Oliveira | Leonardo de Almeida Pereira | Luiz Anselmo dos Santos Oliveira

**Resumo** — Este artigo explora a construção e o funcionamento de uma Unidade Lógica Aritmética (ULA) de 8 bits. Motivados pelo interesse em compreender o funcionamento de circuitos digitais e suas aplicações práticas. A partir de um estudo detalhado, apresentamos o desenvolvimento de uma ULA do zero, destacando os princípios e técnicas envolvidos. Agradecemos especialmente ao professor Walter Oliveira, cujo ensino em um único semestre nos proporcionou o conhecimento necessário para realizar este projeto.

## I. INTRODUÇÃO

Neste artigo, apresentamos o desenvolvimento de uma Unidade Lógica e Aritmética (ULA) de 8 bits, projetada com a linguagem de descrição de hardware Verilog. A ULA desempenha um papel fundamental na manipulação de dados em sistemas digitais, realizando operações aritméticas básicas, como soma e subtração, além de operações lógicas (AND, OR, XOR, NOT, XNOR) e comparações (maior que, menor que, igual, maior ou igual, menor ou igual). Essas operações são controladas por uma tabela de opcode, que associa cada operação a um código binário específico, possibilitando o gerenciamento preciso das operações executadas.

A ULA é essencial no desempenho e na eficiência dos processadores, já que possibilita cálculos, comparações e tomadas de decisão — operações que formam a base da computação moderna. Em sistemas digitais, os circuitos lógicos recebem sinais de entrada binários e os processam em diferentes níveis de tensão, representando valores 0 e 1. Esse projeto, portanto, não apenas ilustra a aplicação do Verilog no design de circuitos digitais, mas também contribui para o entendimento das operações internas de processadores, destacando a importância da ULA em tecnologias de eletrônica digital e oferecendo uma base sólida para futuras implementações em hardware.

## II. METODOLOGIA

Para desenvolver uma Unidade Lógica e Aritmética (ULA) de 8 bits em nível de circuitos digitais, escolhemos uma linguagem de descrição de hardware Verilog. Essa escolha foi motivada pela capacidade do Verilog de modelar e simular circuitos digitais de maneira eficiente, o que nos permitiu criar um circuito detalhado e bem estruturado, composto por portas lógicas e flip-flops. Esse fluxo ágil facilitou e aprimorou o processo de síntese, permitindo-nos validar o design exclusivamente no ambiente de simulação.

As operações aritméticas implementadas incluem soma e subtração entre dois operandos de 8 bits, funções essenciais para cálculos binários básicos. Essas operações foram construídas utilizando somadores completos, garantindo precisão nos cálculos.

Além disso, a ULA possui um conjunto detalhado de operações lógicas, incluindo comparações — como maior que ( $>$ ), menor que ( $<$ ), igual ( $=$ ), maior ou igual ( $>=$ ) e menor ou igual ( $<=$ ) — que permitem comparações entre operandos. Cada uma dessas operações retorna um valor binário (1 ou 0), indicando se a condição é verdadeira ou falsa.

A ULA também executa operações lógicas bit a bit, como AND, OR, XOR, NOT e XNOR, manipulando diretamente cada bit dos operandos para transformar ou analisar dados de forma detalhada.

OPERAÇÃO	DESCRIÇÃO	OPCODE
SOMA	Resultado de $A + B$	0000
SUBTRAÇÃO	Resultado de $A - B$	0001
MAIOR QUE	Verifica se $A > B$	0010
MENOR QUE	Verifica se $A < B$	0011
MAIOR OU IGUAL	Verifica se $A \geq B$	0100
MENOR OU IGUAL	Verifica se $A \leq B$	0101
IGUAL	Verifica se $A = B$	0110
NOT	NOT bit a bit	0111
AND	AND bit a bit	1000
OR	OR bit a bit	1001
XOR	XOR bit a bit	1010
XNOR	XNOR bit a bit	1011

Para identificar cada operação, foi criada uma tabela de opcode que associa cada função a um código binário exclusivo. Essa tabela orienta a ULA na seleção e execução da operação correta para cada par de operandos.

Foi desenvolvido também um diagrama de blocos para fornecer uma visão clara da arquitetura da ULA de 8 bits. Nele, cada componente e sua função estão organizados para ilustrar o fluxo de dados e as operações realizadas. As entradas A e B de 8 bits são recebidas por registradores específicos, que

armazenam temporariamente os valores para processamento. Em seguida, o decodificador de opcode interpreta o código binário fornecido e seleciona a operação a ser executada — seja aritmética ou lógica. As operações aritméticas incluem: soma e subtração, as operações lógicas incluem: Maior que, Menor que, Maior ou igual, Menor ou igual, Igual, Not, And, Or, Xor, Xnor.

Os resultados dessas operações são enviados ao registrador de saída, que armazena o resultado final. Este registrador possui 9 bits, acomodando a saída dos cálculos e possíveis bits extras. Por exemplo, se somarmos 11111111 + 11111111, o resultado será 11111110, o que exige 9 bits para representar corretamente o valor. O sinal de clock controla todo o fluxo de dados e a sincronização entre os elementos do circuito, garantindo que cada operação ocorra de forma ordenada e precisa.

