

Desenvolvimento de uma ULA de 16 Bits: Somador e Multiplicador

Estela Vidal Platero: estelavidalplatero@gmail.com

Mateus de Souza Nascimento: mateus.leno@yahoo.com.br

Welinton Thiago Fечи Sandrin: thiagosandrin2@gmail.com

Wesley Santos Silva: silva.wes2003@gmail.com

Centro Universitário Adventista de São Paulo (UNASP), Faculdade de Ciência da
Computação, São Paulo, Brasil

Estr. de Itapecerica, 5859, Capão Redondo

CEP 05890-020

2024

Resumo do artigo (Abstract).

Este trabalho apresenta o desenvolvimento, simulação e validação de uma Unidade Lógica e Aritmética (ALU) de 16 bits projetada em VHDL. A ALU implementa duas operações básicas: soma e multiplicação, selecionadas por meio de um sinal de controle (Mode). O projeto é composto por módulos específicos para cada operação e um multiplexador que gerencia o acesso ao barramento compartilhado, assegurando a correta seleção e execução das operações.

As simulações realizadas no ambiente ModelSim demonstraram a precisão e eficiência do sistema, com os resultados esperados para diferentes combinações de operandos e modos de operação. O design modular da ALU permite fácil compreensão e potencial para futuras expansões, como a adição de novas operações ou a parametrização da largura de bits.

O artigo também apresenta uma análise dos resultados e discute limitações e melhorias possíveis, como otimizações de desempenho e ampliação do conjunto de operações. O projeto comprova a viabilidade de implementação de ALUs funcionais em linguagem VHDL, representando uma contribuição significativa para o design de sistemas digitais e processamento aritmético em arquiteturas embarcadas.

Keywords: ALU (Arithmetic Logic Unit), ULA, Somador, Multiplicador, 16 Bits, Multiplexador, VHDL, ModelSim, Simulação de Circuitos

SUMÁRIO

1. INTRODUÇÃO

2. METODOLOGIA

- 2.1 Descrição do Sistema
- 2.2 Arquitetura dos Módulos
- 2.3 Processo de Simulação

3. RESULTADOS E DISCUSSÃO

- 3.1 Gráficos da Simulação
- 3.2 Discussão dos Resultados
- 3.3 Limitações e Melhorias
- 3.2 Conclusão da Discussão

4. CONCLUSÃO

5. REFERÊNCIAS

1. INTRODUÇÃO

PROJETO DISPONÍVEL EM: <https://github.com/estelavidal/projeto-vdhl>

As Unidades Lógicas e Aritméticas (ALUs) são componentes centrais de processadores e sistemas digitais que desempenham operações fundamentais, possibilitando o processamento de dados [3]. ALUs modernas precisam equilibrar precisão, eficiência e otimização de recursos de hardware, especialmente em dispositivos embarcados, onde a área do chip e o consumo de energia são críticos.

O objetivo deste trabalho é projetar e validar uma ALU básica e funcional, com foco na implementação de duas operações essenciais: soma e multiplicação de números de 16 bits. O projeto tem como objetivos:

1. Desenvolver uma arquitetura que seja modular e flexível, permitindo ajustes e expansões de maneira simples e eficiente.
2. Realizar testes para assegurar que o sistema funcione de forma confiável e atenda aos requisitos de desempenho.
3. Explorar as possibilidades de integração da ALU em arquiteturas de processadores de uso geral, ampliando seu potencial de aplicação em sistemas reais.

Além disso, o projeto destaca a utilização do VHDL [1], uma linguagem amplamente utilizada no design de sistemas digitais, para descrever, implementar e simular o sistema.

2. METODOLOGIA

2.1 Descrição do Sistema

A arquitetura da ALU consiste em três módulos principais:

- Somador 16 bits: Realiza a soma de dois números de 16 bits.
- Multiplicador 16 bits: Executa a multiplicação de dois números de 16 bits, gerando uma saída de 32 bits.
- Multiplexador: Seleciona entre as saídas do somador e do multiplicador com base no sinal de controle Mode.

A integração desses componentes é gerenciada pelo módulo ALU Control, responsável por definir qual operação será realizada com base no valor do sinal Mode, assim como mostrado na tabela abaixo:

Sinal Mode	Operação	Saída (ALU_Result)
0	Soma	Sinal redimensionado para 32 bits
1	Multiplicação	Saída direta do multiplicador

2.2 Arquitetura dos Módulos

1. Somador 16 bits:
Implementado para realizar somas de forma eficiente, utilizando vetores de 16 bits. A lógica soma os dois operandos como números sem sinal (unsigned).
2. Multiplicador 16 bits:
O multiplicador usa aritmética sem sinal para gerar uma saída de 32 bits, necessária devido à expansão do número de bits resultante da multiplicação.

3. Multiplexador:
Realiza a seleção entre as duas saídas disponíveis (soma e multiplicação) com base no sinal Mode.
4. Controle ALU:
Gerencia o fluxo do sistema, conectando as saídas do somador e do multiplicador ao barramento de dados.

2.3 Processo de Simulação

Para validar o funcionamento do sistema, foi elaborado um banco de testes (testbench) que fornece diferentes estímulos de entrada e monitora as saídas. As entradas e as condições de teste foram:

Teste 1: Soma com A = 00000000000000011 (3 em decimal) e
B = 00000000000000101 (5 em decimal).

Teste 2: Multiplicação com os mesmos valores.

As simulações foram realizadas no ModelSim Intel FPGA Standard Edition [4].

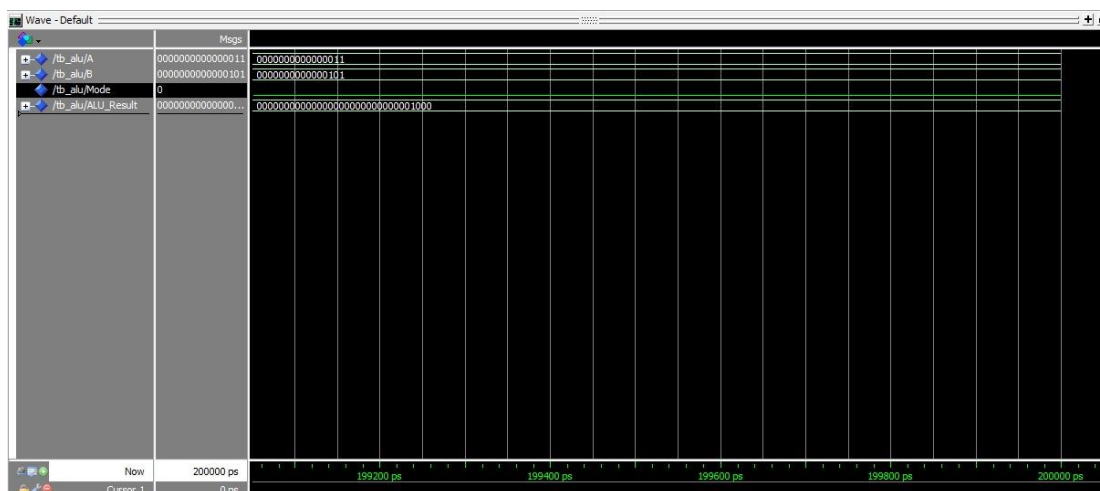
3. RESULTADOS E DISCUSSÃO

3.1 Gráficos de Simulação

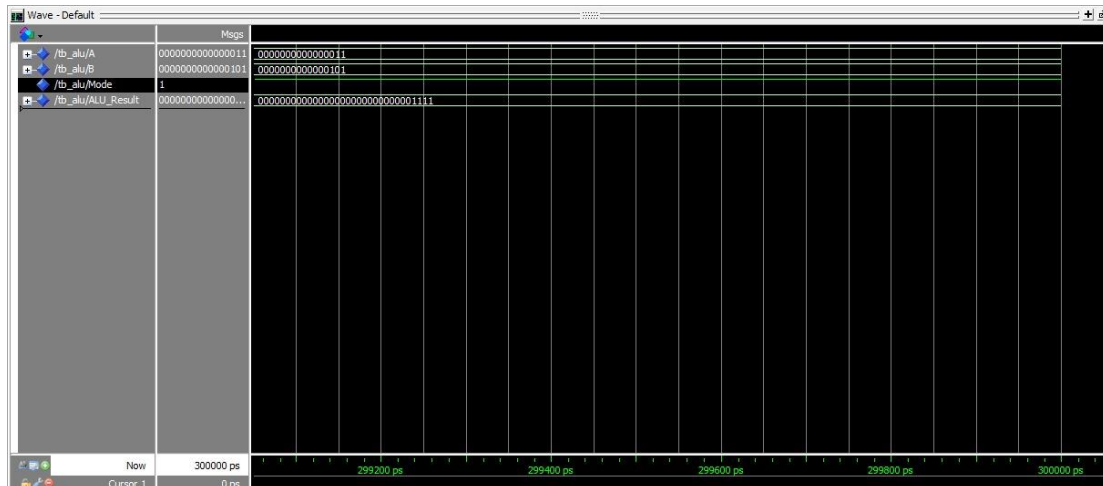
Os resultados obtidos através da simulação no ModelSim [4] confirmaram o funcionamento correto da Unidade Lógica e Aritmética (ALU) de 16 bits, conforme esperado. A análise dos gráficos de simulação permitiu validar as operações de soma e multiplicação em diferentes modos de operação (controlados pelo sinal Mode), mostrando que o sistema realiza as operações de forma precisa.

No primeiro cenário de teste, com Mode = '0', a ALU foi configurada para realizar a soma. Com os operandos A = 3 e B = 5, o resultado esperado era 8, o que foi corretamente refletido no sinal ALU_Result, conforme mostrado no gráfico da simulação. Já no segundo cenário, com Mode = '1', a ALU foi configurada para realizar multiplicação, o resultado esperado era 15 e novamente corretamente o refletiu.

Cenário Teste Soma:



Cenário Teste Multiplicação:



3.2 Discussão dos Resultados

Os gráficos de simulação apresentam uma transição clara entre os estados dos sinais de entrada e saída, evidenciando que o sistema responde corretamente às mudanças de modo e valores de entrada. O uso do multiplexador foi essencial para selecionar a operação adequada entre o somador e o multiplicador, e a arquitetura modular facilitou a integração dos componentes.

A precisão dos resultados indica que os componentes básicos, como o somador e o multiplicador, foram projetados corretamente. A conversão de vetores de sinais para números inteiros usando bibliotecas da IEEE, como `NUMERIC_STD`, garantiu que as operações aritméticas fossem realizadas sem inconsistências [2, 5].

Além disso, o tempo de propagação foi observado como satisfatório para as condições de simulação, embora a análise de desempenho para frequências mais altas ou em implementações físicas (FPGA ou ASIC) ainda precise ser realizada. A modularidade do projeto também se mostrou um ponto positivo, permitindo ajustes futuros e expansões, como a adição de novas operações aritméticas ou lógicas.

3.3 Limitações e Melhorias

Embora os resultados sejam promissores, algumas limitações foram identificadas. O sistema atualmente suporta apenas duas operações, o que restringe sua aplicabilidade em sistemas mais complexos. Além disso, a largura fixa dos operandos (16 bits) pode não ser suficiente para aplicações que demandam maior precisão ou maiores intervalos de valores.

Como melhorias futuras, sugere-se:

1. **Ampliação do Conjunto de Operações:** A inclusão de operações lógicas (AND, OR, XOR), deslocamento de bits e subtração pode aumentar a flexibilidade da ALU.
2. **Parametrização da Largura de Dados:** Permitir que a largura dos operandos seja configurável, tornando o projeto mais versátil.
3. **Otimização de Recursos:** Explorar técnicas para reduzir a área de silício e o consumo de energia em implementações físicas.
4. **Análise de Desempenho:** Realizar testes para determinar o desempenho da ALU em altas frequências e seu impacto no tempo de propagação.

3.4 Conclusão da Discussão

Os resultados obtidos validam o correto funcionamento da ALU e destacam a eficácia do uso de VHDL para o desenvolvimento de sistemas digitais modulares. A análise também aponta para o potencial de melhorias no projeto, indicando caminhos promissores para adaptações futuras. O projeto, além de atingir seus objetivos, serve como base para outros mais avançados e para o desenvolvimento de ALUs mais robustas e completas.

4. CONCLUSÃO

O desenvolvimento da ALU de 16 bits apresentado neste trabalho demonstrou a eficácia do uso de VHDL no design e validação de sistemas digitais modulares e escaláveis [1]. A implementação das operações de soma e multiplicação foi realizada de forma eficiente, comprovando que o projeto é funcional e adequado para aplicações práticas. Além disso, o uso de simulações no ModelSim permitiu validar o comportamento do sistema em diferentes cenários, garantindo precisão e confiabilidade nos resultados.

As simulações evidenciaram a robustez do projeto e abriram caminho para melhorias e otimizações. O sistema proposto possui potencial de aplicação em dispositivos embarcados, sistemas de controle e até mesmo em arquiteturas de processadores de propósito geral. Entretanto, algumas limitações foram identificadas, como a restrição a apenas duas operações e a largura fixa de 16 bits para os operandos.

Em resumo, este trabalho contribui para o campo da engenharia digital, apresentando uma solução prática e eficiente para a implementação de uma ALU simples, mas funcional, que pode ser utilizada como base para projetos mais avançados. Essa abordagem reforça a importância de metodologias estruturadas no design de circuitos digitais, bem como a utilização de ferramentas de simulação para garantir a confiabilidade e a qualidade do sistema.

5. REFERÊNCIAS

- [1] BROWN, S.; VRANESIC, Z. **Fundamentals of Digital Logic with VHDL Design**. 3ª edição. New York: McGraw-Hill, 2008.
- [2] PEDRONI, V. A. **Circuit Design with VHDL**. 2ª edição. Cambridge: MIT Press, 2008.
- [3] ZWOLINSKI, M. **Digital System Design with VHDL**. 2ª edição. London: Pearson Education, 2004.
- [4] INTEL. **ModelSim Intel FPGA Edition User Guide**. Disponível em: <https://www.intel.com/content/www/us/en/software-kit/750368/modelsim-intel-fpgas-standard-edition-software-version-18-1.html>. Acesso em: [26/11/2024].
- [5] IEEE. **IEEE Standard for VHDL Language Reference Manual (IEEE Std 1076-2008)**. New York: IEEE, 2009.