

Circuito divisor de ponto fixo (inteiro)

Arthur Navarro Guedes - RA: 11085314

Caio Cedran Colombo - RA: 11201722553

Universidade Federal do ABC (UFABC)

{guedes.arthur, cedran.c}@aluno.ufabc.edu.br

1. Introdução ou Motivação

A divisão de ponto fixo é uma operação imperativa na realização de cálculos de divisão com números inteiros. O ponto fixo, uma representação numérica que aloca uma quantidade fixa de bits para a parte inteira e fracionária de um número real, permite abordar eficientemente esse tipo de cálculo.

A importância da implementação de tal circuito é respaldada pela sua relevância em cenários práticos, como no controle de motores de passo.[1] Esses dispositivos eletromecânicos, conhecidos por converter pulsos elétricos em movimentos mecânicos incrementais, são amplamente utilizados em diversas aplicações que exigem precisão e controle no posicionamento. A utilização da operação de divisão é essencial para a utilização de uma gama de fórmulas físicas e trigonométricas, tornando-se essencial para essa e outras aplicações, como por exemplo impressoras, plotters e equipamentos de coordenadas.

2. Objetivos

O objetivo geral deste projeto consiste em conceber e desenvolver um circuito divisor de ponto fixo com capacidade de calcular o quociente e o resto de uma divisão entre dois números inteiros de 4 bits.

Para atingir o objetivo geral proposto, o trabalho busca cumprir os seguintes

objetivos específicos: Implementar o circuito divisor de ponto fixo utilizando a linguagem VHDL, assegurando a correta execução das operações de divisão, tanto para a parte inteira quanto para a parte fracionária dos números envolvidos; Realizar testes do circuito em diferentes cenários e condições, a fim de verificar a corretude de seu funcionamento; realizar uma proposição de visualização dos resultados obtidos na Placa Intel DE10-Lite, utilizando seus switches como entradas (dividendo e divisor) e seus displays de 7 segmentos para representar os valores do algoritmo (divisor, dividendo, quociente e resto), permitindo uma avaliação em tempo real das operações do circuito.

Fora isto, será utilizado o visualizador de ondas GTKWave, para ilustrar diferentes cenários de divisão a fim de validar o algoritmo por completo.

3. Justificativa

O desenvolvimento do circuito divisor de ponto fixo apresenta uma contribuição significativa, destacando-se por sua capacidade de proporcionar cálculos de divisões inteiras com elevada precisão e eficiência. Ao optar pela implementação em VHDL, torna-se possível obter uma descrição clara e compreensível do hardware, simplificando sua análise, manutenção e otimização.

A escolha da linguagem VHDL proporciona uma abstração de alto nível, permitindo a especificação do comportamento do

hardware de forma estruturada e logicamente coesa. Essa abordagem confere uma vantagem substancial ao circuito, permitindo que seu funcionamento seja compreendido e validado antes da implementação em hardware real.

A utilização do testbench para realizar simulações é um elemento essencial na validação do circuito divisor. Por meio dessas simulações, é possível verificar o correto funcionamento em diversos cenários e condições, garantindo que o circuito opere conforme o esperado, livre de erros e imprecisões.

A possibilidade de validar o circuito por meio de simulações antes da implementação física proporciona benefícios significativos, permitindo a detecção e correção de problemas de projeto antes de serem manifestados no hardware real. Essa abordagem evita gastos desnecessários com prototipagem e o refazer do projeto, contribuindo para a redução de custos e prazos no desenvolvimento do circuito divisor de ponto fixo.

Dessa forma, a combinação da implementação em VHDL com a validação por meio de simulações utilizando o testbench fortalece a importância do circuito divisor de ponto fixo como uma ferramenta de alto desempenho e confiabilidade. Sua capacidade de realizar cálculos precisos e eficientes em divisões inteiras o torna essencial em aplicações que exigem resultados confiáveis e consistentes, como em sistemas de controle, processamento de sinais e outras áreas de grande relevância.

Diante desse contexto, a justificativa apresentada reforça a pertinência e o valor do circuito divisor de ponto fixo, enfatizando sua importância em cenários onde a precisão, a eficiência e a confiabilidade são fundamentais para o sucesso das aplicações em que ele será empregado.

4. Metodologia de Projeto ou de Simulação

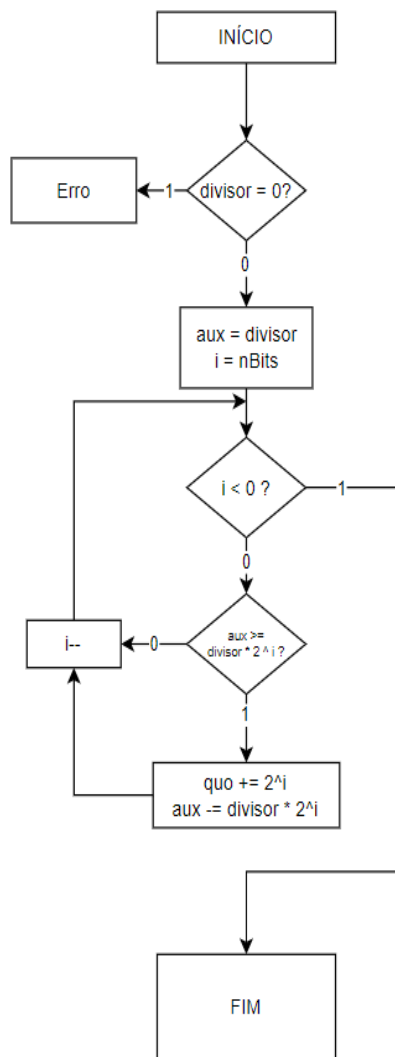


Diagrama 1. Fluxograma lógico da divisão binária

O desenvolvimento foi dividido em dois arquivos: O arquivo ***divider.vhd*** contém toda lógica de divisão binária e entradas e saídas na placa, e o arquivo ***divider_vhd_tst.vhd*** que contém os testes que serão feitos no programa GTKWave.

| | | | | |
|---|-------------|----|----|-----|
| 1 | $11 \div 3$ | 3 | 2 | Não |
| 2 | $15 \div 2$ | 7 | 1 | Não |
| 3 | $13 \div 5$ | 2 | 3 | Não |
| 4 | $10 \div 0$ | 0 | 0 | Sim |
| 5 | $2 \div 2$ | 1 | 0 | Não |
| 6 | $15 \div 1$ | 15 | 0 | Não |
| 7 | $0 \div 0$ | 0 | 0 | Sim |
| 8 | $2 \div 10$ | 0 | 10 | Não |

Tabela 1. Cenários de teste

| Teste | Motivação do Cenário |
|-------|-------------------------------------|
| 1 | Contido nos material para o projeto |
| 2 | Contido nos material para o projeto |
| 3 | Contido nos material para o projeto |
| 4 | Divisão por zero |
| 5 | Dividendo e divisor iguais |
| 6 | Divisão por elemento neutro (1) |
| 7 | Divisão zero por zero |
| 8 | Divisor maior que dividendo |

Tabela 2. Motivação de cada Cenário

5. Apresentação dos Dados e Análise dos Resultados

Logo abaixo temos os resultados das divisões dos testes GTKWave e uma proposição de ilustração do algoritmo utilizando-se da Placa Intel DE10-Lite GTKWave:

| Teste | Operação | Quociente | Resto | Erro |
|-------|----------|-----------|-------|------|
|-------|----------|-----------|-------|------|

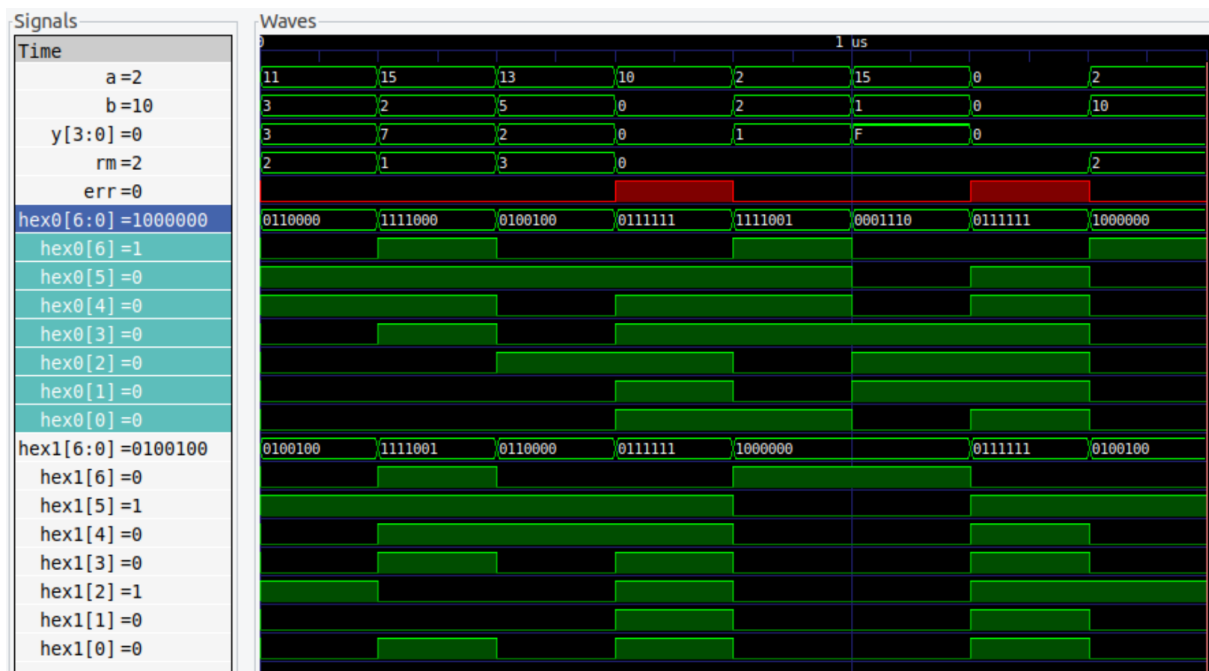


Figura 1. Testes GTKWave

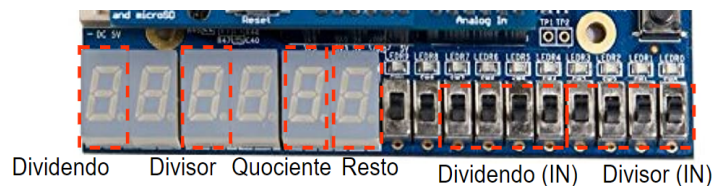


Figura 2. Proposição de visualização do algoritmo utilizando-se da Placa Intel DE10-Lite

6. Apresentação de um Exemplo de Funcionamento do Programa

Os códigos do projeto estão contidos no seguinte repositório do GitHub:

<https://github.com/GuedesArthur/SistemasDigitais-2023.2>

Baixe os arquivos *src/divider.vhd* e *src/divider_vhd_tst.vhd*. Certifique-se de usar sistema linux e que o programa GTKWave está instalado.

Compile-os utilizando o comando:

```
ghdl -a src/divider.vhd
src/divider_vhd_tst.vhd
```

Após compilar, gere um executável utilizando o comando:

```
ghdl -e src/divider_vhd_tst
```

Por fim, injete os vetores de entrada no testbench utilizando o comando:

```
ghdl -r src/divider_vhd_tst
-wave=resposta.ghw
```

e execute-o com o programa GTKWave com o comando:

```
gtkwave resposta.ghw &
```

7. Conclusão

A análise dos resultados demonstrou que o circuito divisor de ponto fixo alcançou os

objetivos propostos, fornecendo resultados de acordo com o esperado em todas as divisões realizadas. Os dados coletados nos testes realizados no GTKWave confirmaram a funcionalidade do circuito, corroborando com a metodologia de projeto ou simulação aplicada.

Referência Bibliográfica

[1]-FPGAParaTodos. Controle de um motor de passo. Disponível em:
<http://fpgaparatodos.com.br/2012/03/30/controle-de-um-motor-de-passo/>
Acesso em: [25/07/2023]

[2]- PEDRONI, Volnei A. Circuit Design with VHDL. Cambridge, Massachusetts; London, England: MIT Press, 3^a.ed., 2004. (páginas 198-202).