

Relatório Implementação FPGA do Caminho de Dados

Pedro Paulo (5937)¹, Erich Pinheiro (5915)²

¹Ciência da Computação – Universidade Federal de Viçosa - Campus Florestal (UFV)

Abstract. *This work describes the implementation and adaptation of a simple datapath, initially designed and simulated in Verilog, for an FPGA integrated circuit board. The main objective is to demonstrate the datapath's functionality in hardware, allowing real-time visualization of the Program Counter (PC) and all 32 register values on the FPGA's integrated seven-segment displays. This approach not only validates the logical design but also provides an interactive tool for understanding data flow and the processor's internal state, facilitating debugging and analysis of system behavior in a physical environment.*

Resumo. *Este trabalho descreve a implementação e adaptação de um caminho de dados simples, inicialmente projetado e simulado em Verilog, para uma placa de circuito integrado FPGA. O objetivo principal é demonstrar a funcionalidade do caminho de dados em hardware, permitindo a visualização em tempo real dos valores do Program Counter (PC) e dos 32 registradores em displays de sete segmentos integrados à FPGA. Esta abordagem não só valida o design lógico, mas também proporciona uma ferramenta interativa para a compreensão do fluxo de dados e do estado interno do processador, facilitando a depuração e a análise do comportamento do sistema em um ambiente físico.*

1. Introdução

Este trabalho tem como objetivo adaptar um projeto de um processador RISC-V, que foi anteriormente desenvolvido e implementado em Verilog, para possibilitar sua simulação e execução em uma placa FPGA por meio do software Quartus. A proposta consiste em realizar a visualização do valor atual dos registradores e do contador de programa (PC) diretamente no display de 7 segmentos da placa, permitindo o acompanhamento do funcionamento do processador em tempo real. Para isso, será executada uma instrução por vez, facilitando a análise detalhada do comportamento do sistema e a validação do caminho de dados implementado. Essa abordagem visa proporcionar uma compreensão mais clara da arquitetura RISC-V e seu funcionamento prático em hardware.

Para concretizar essa implementação, o foco principal recaiu sobre a adaptação do caminho de dados simples do processador RISC-V na placa FPGA. A interatividade com o sistema é garantida pela utilização dos switches da placa, que permitem a seleção dinâmica dos registradores a serem visualizados. Adicionalmente, para o controle preciso da execução, foram designados botões específicos para as funções de reset e clock. O botão de clock é crucial para avançar a execução do processador, permitindo que cada instrução seja processada individualmente. É importante ressaltar que, para o correto funcionamento do reset, é necessário acioná-lo e, em seguida, pressionar o botão de clock, garantindo assim a inicialização adequada do sistema antes de cada ciclo de instrução.

2. Implementação

Além dos módulos implementados anteriormente, foram adicionados quatro novos módulos, e foram adaptados os módulos `registrador.v` e `main.v` que serão explicados abaixo.

Main.v: Foram adicionadas duas novas portas, conectadas ao registrador, que são responsáveis por receber o endereço do registrador a ser exibido e envia o seu conteúdo, que sera interpretado pelo módulo de display.

Registrador.v: Foram adicionas as mesmas duas portas do modulo main, permitindo o acesso a cada registrador sem alterar a logica de leitura e escrita.

Display: Controla a interface com o usuário, colocando botões para reset e para clock do processador, além de receber qual será o registrador via switches. Faz o debounce dos botões, gera pulsos para o clock, e converte os valores do PC e do registrador selecionado para sinais do displays de 7 segmentos.

HexTo7Seg: Converte um valor hexadecimal de 4 bits para o displays de 7 segmentos, permitindo a visualização dos números hexadecimais.

SinglePulse: Gera um pulso único de clock para cada pressionamento de botão, garantindo que o processador avance uma instrução por vez.

Debouncer: Garante estabilidade do sinal para que cada pressionamento do botão seja registrado apenas uma vez.

Todo o código foi compilado com sucesso no software Quartus e executado na placa FPGA, demonstrando a funcionalidade do processador e a visualização dos dados em tempo real.

3. Resultados

Após a adaptação e criação dos novos módulos implementados, foi possível a realização da simulação e execução do processador RISC-V na placa FPGA. Utilizando o conjunto de instruções apresentado na Figura 1, foi possível visualizar o funcionamento do caminho de dados em tempo real. Os resultados obtidos confirmaram a correção dos valores dos registradores (Figura 2), que foram exibidos nos displays de 7 segmentos da placa.

Especificamente, os 32 registradores foram apresentados nos displays HEX5 e HEX4. O valor atual do Program Counter (PC) foi exibido nos displays HEX7 e HEX6. A interação com o sistema foi facilitada pela implementação de botões e switches dedicados: o botão KEY3 foi configurado para o clock (clk), enquanto o botão KEY2 foi utilizado para o reset. A seleção do registrador a ser exibido foi realizada por meio dos switches SW4 a SW0.

A Figura 3 detalha a implementação física do projeto, demonstrando a pinagem da FPGA para os periféricos utilizados. O principal desafio técnico durante a implementação foi lidar com o ruído elétrico (bouncing) gerado pelos botões mecânicos. A solução foi desenvolver um módulo "Debouncer", que filtra os sinais indesejados e garante que apenas um pulso estável seja enviado ao sistema, transformando um obstáculo técnico em um importante aprendizado prático.

```

addi x1, x0, 42
addi x2, x0, 16
addi x3, x0, 8
addi x7, x0, 1
srl x3, x3, x7
sb x2, 0(x3)
lb x2, 0(x0)
addi x4, x4, 4
beq x4, x3, 8
sub x1, x1, x3
srl x6, x1, x7
ori x10, x6, 0x20

```

Figura 1: Instruções executadas em RISC-V

```

Registrador [ 0]: 0
Registrador [ 1]: 42
Registrador [ 2]: 0
Registrador [ 3]: 4
Registrador [ 4]: 4
Registrador [ 5]: 0
Registrador [ 6]: 21
Registrador [ 7]: 1
Registrador [ 8]: 0
Registrador [ 9]: 0
Registrador [10]: 53
Registrador [11]: 0
Registrador [12]: 0
Registrador [13]: 0
Registrador [14]: 0
Registrador [15]: 0
Registrador [16]: 0
Registrador [17]: 0
Registrador [18]: 0
Registrador [19]: 0
Registrador [20]: 0
Registrador [21]: 0
Registrador [22]: 0
Registrador [23]: 0
Registrador [24]: 0
Registrador [25]: 0
Registrador [26]: 0
Registrador [27]: 0
Registrador [28]: 0
Registrador [29]: 0
Registrador [30]: 0
Registrador [31]: 0

```

Figura 2: Valores dos registradores após execução

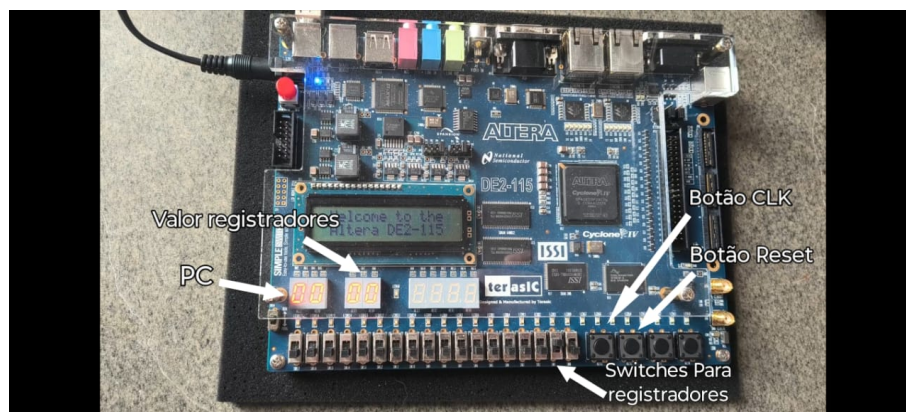


Figura 3: Implementação física e pinagem

4. Conclusão

A adaptação do processador RISC-V para execução em uma placa FPGA possibilitou a análise em tempo real do seu funcionamento, por meio da exibição dos valores dos registradores e do contador de programa (PC) no display de 7 segmentos. A execução de uma instrução por vez permitiu observar de forma detalhada o comportamento do caminho de dados, facilitando a verificação do projeto e o entendimento dos fluxos do processador. Além de aprofundar o conhecimento sobre a arquitetura RISC-V, esse trabalho também reforçou a aplicação prática da linguagem Verilog no desenvolvimento de sistemas voltados para hardware.