Projeto e implementação em VHDL de um softcore

Prof. Renan Augusto Starke

1 Introdução

RISC-V é uma nova arquitetura de conjunto de instruções (ISA) que foi originalmente projetada para dar suporte à comunidade científica em pesquisas voltadas a arquiteturas de computadores e também para edução. Atualmente está tornando-se um padrão de implementação livre (*open hardware*) de arquiteturas para aplicações industriais [1].

Este projeto final utiliza uma implementação em VHDL com fins diádicos do conjunto de instruções RISCV RV32I. A ideia é criar um microcontrolador com periféricos comuns como I2C, USART, SPI e GPIOs. Uma visão geral da implementação pode ser visualizada na Figura 1.

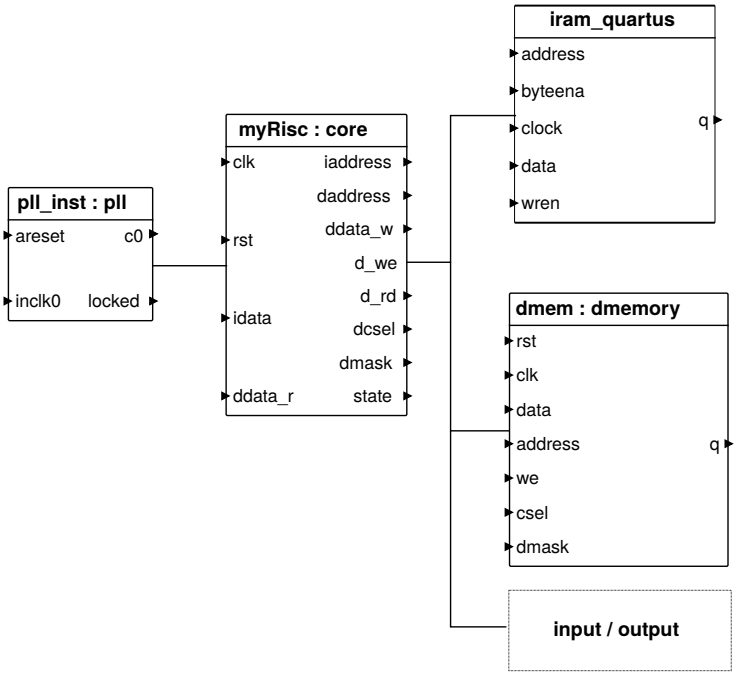

Figura 1: Visão geral da implementação VHDL (*top entity*)

2 Componentes internos

Essa implementação RISCV RV32I particular não implementa um pipeline e foi criada seguindo a especificação disponível em [2]. Como referência, também é possível consultar [3].

A implementação VHDL fornecida é composta pelos componentes principais apresentados pela Figura 2, onde:

- *iram_quartus*: IP de memória RAM de duas portas 32-bits x 1024 words. Código executado pelo núcleo deve ser gravado nessa memória.
- *pll*: PLL para redução e manutenção do clock da CPU. A frequência pode ser ajustada, recomenda-se 1MHz durante o desenvolvimento.

- *iregister*: pré decodificação das instruções: *bit slicing* da palavra de instruções conforme [2]: pág 103.
- *decoder*: máquina de estados responsável pela decodificação das instruções conforme [2]: Capítulo 2 e pág 103.
- *register_file*: banco de 32 registradores.
- *ula*: unidade lógica e aritmética.
- *dmemory*: memória de dados.

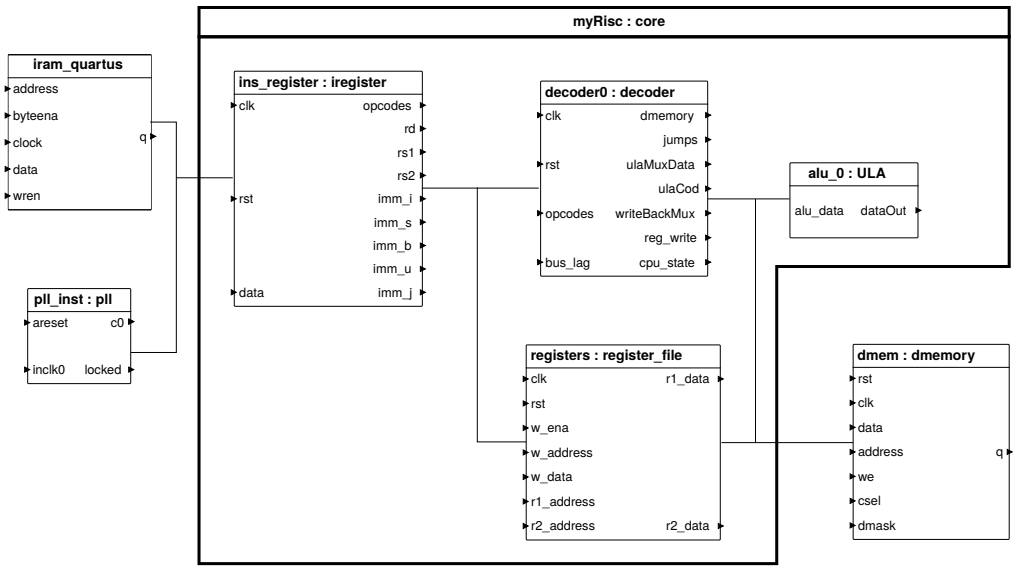

Figura 2: Visão geral dos componentes internos

3 Especificação e requisitos do projeto

O objetivo deste projeto é implementar um microcontrolador com periféricos comuns como I2C, USART, SPI, GPIOs ..., de forma colaborativa. O código do hardware está disponível em [4] em conjunto com as instruções de síntese e compilação do código.

O projeto deve ser implementado em algum kit FPGA, DE10-Lite por exemplo, e cada aluno será responsável por implementar um periférico para o núcleo RISC-V, como por exemplo:

- extensão de instruções para multiplicação RV32I+M;
- extensão de instruções de ponto flutuante;
- extensão de instruções DSP;
- portabilidade e testes para a biblioteca padrão C;
- interface de depuração (JTAG);
- controlador de interrupções;
- conversor analógico digital;
- temporizadores;
- controlador de SDRAM;
- controlador VGA;

- interface UART;
- interface SPI;
- interface I2C;
- controlador de display LCD (gráfico ou caractere).

O periférico deverá ser implementado em VHDL e deve:

- seguir o modelo hierárquico: elaborem componentes e integre-os no de maior nível;
- conter modelo de simulação: *testbench* no ModelSim;
- permitir síntese em hardware: utilizando as ferramentas apresentadas na disciplina, sintetizar o sistema no kit de desenvolvimento;
- executar um software: uma aplicação deverá ser desenvolvida mostrando a integração do núcleo com o *hardware* externo: UART, LEDs e botões.

Além da implementação do hardware, será necessário criar a camada de abstração de hardware (HAL), ou seja, mapear os registradores e criar as funções para que o periférico possa ser utilizado pelo programa C.

4 Avaliação

O projeto será composto por diversos componentes com funções específicas implementados em VHDL para a FPGA, assim, o projeto será avaliado conforme os itens abaixo:

- cumprimento das especificação;
- qualidade do código;
- inferência dos componentes adequados: memórias, *latches* e registrados;
- qualidade do *testbench*: geração de arquivo com *trace* de execução;
- qualidade da aplicação.
- apresentação do funcionamento em *hardware*.
- colaboração com o projeto no *github*: código fonte, simulação e documentação.

Referências

[1] RISC-V Foundation,
`https://riscv.org/`

[2] The RISC-V Instruction Set Manual v2.2,
`https://github.com/xtarke/riscv-multicycle/blob/master/docs/riscv-spec-v2.2.pdf`

[3] D. Patterson and J.Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. Morgan Kaufmann, 2005.

[4] RISC-V IFSC Code
`https://github.com/xtarke/riscv-multicycle`