



INSTITUTO SUPERIOR TÉCNICO
MESTRADO INTEGRADO EM ENGENHARIA ELECTROTÉCNICA E DE
COMPUTADORES

ARQUITECTURAS AVANÇADAS DE COMPUTADORES

Simulação de um processador μ Risc

Maria Margarida Dias dos Reis	n.º 73099
Nuno Miguel Rodrigues Machado	n.º 74236

Lisboa, 29 de Março 2014

Índice

1	Introdução	1
2	Características do Processador	1
3	Estrutura do Processador	1
3.1	Primeiro Andar - IF	1
3.2	Segundo Andar - ID e OF	1
3.3	Terceiro Andar - EX e MEM	1
3.4	Quarto Andar - WB	1

1 Introdução

Com este trabalho laboratorial pretende-se projectar um processador μ Risc, de 16 *bits* com arquitectura RISC. O processador possui 8 registos de uso geral e 42 instruções. O projecto do processador é feito com recurso a uma linguagem de descrição de *hardware* - VHDL.

2 Características do Processador

O processador elaborado foi simulado para uma placa Artix 7 e tem as seguintes características:

- 16 *bits*;
- 8 registos de uso geral de 16 *bits* de largura (R0, ..., R7);
- 42 instruções;
- instruções de 3 operandos;
- organização de dados na memória do tipo *big endian*;
- uma memória ROM de 8 KBytes (4k endereços \times 2 *bytes*) endereçada com palavras de 12 *bits* utilizada para as instruções/programa e uma memória RAM de 8 KBytes (4k endereços \times 2 *bytes*) endereçada com palavras de 12 *bits* para os dados.

3 Estrutura do Processador

O processador μ Risc que foi projectado encontra-se dividido em quatro andares - num primeiro andar é feito o *instruction fetch* (IF), no segundo andar é feito o *instruction decode* (ID) e o *operand fetch* (OF), no terceiro andar são executadas operações da ALU (EX) e de acesso à memória de dados (MEM) e, por fim, no quarto e último andar é feita a escrita no banco de registos, o *write back* (WB).

3.1 Primeiro Andar - IF

No primeiro andar a próxima instrução a ser executada é carregada da memória ROM.

3.2 Segundo Andar - ID e OF

3.3 Terceiro Andar - EX e MEM

3.4 Quarto Andar - WB

No último andar os resultados são escritos no banco de registos, sendo que os resultados

Todo list