Projeto 1 AC

Indice

1. **Introdução**

Este relatório apresentará os objetivos relacionados ao primeiro trabalho prático da unidade curricular de Arquitetura de Computadores assim como o seu desenvolvimento, discussão de resultados e a conclusão a que os alunos chegaram no fim.

Os processadores são as unidades centrais dos sistemas computacionais. Se comparássemos um sistema computacional a uma pessoa, o processador seria o “cérebro” pois é ele que executa as instruções de máquina, que são qualquer tarefa que o processador possa executar utilizando uma série de cálculos e decisões.

1. **Objetivos**

O objetivo deste trabalho é realizar um processador básico, com um conjunto mínimo de instruções, em linguagem de descrição de hardware (VHDL). Para este fim, utilizou-se o programa ISE da Xilinx com a simulação sendo efetuada no ISim e o teste em FPGA (Spartan 3E e Artix 7).

1. **Desenvolvimento**

O processador desenvolvido é constituído por vários módulos que, quando conectados com a memória de dados e a memória de instruções, formam a placa mãe. Cada módulo foi implementado em separado, como será descrito de seguida, de modo a facilitar a implementação, sendo só necessário no fim ligar os diferentes módulos.

* 1. **Placa-mãe**
     1. **Memória de Dados (RAM)**

A RAM (Random-Access Memory) é um tipo de memória que permite a escrita/leitura de dados, individual e aleatória (como o nome indica), através do seu endereço. É uma memória volátil, ou seja, os seus dados perdem-se quando a memória perde a alimentação elétrica.

No caso desta placa-mãe, a memória de dados guarda os dados presentes no sinal de entrada ***Operando1,*** de 8 bits, quando o sinal ***WR*** está a ‘1’ na transição ascendente do sinal de relógio (***clk***), no endereço indicado pelo sinal de entrada ***Constante***, de 8 bits. Quando o sinal ***WR*** está a ‘0’ é feita a leitura dos dados, na posição de memória indicada por ***Constante*** e o valor lido é atribuído ao sinal de saída ***Dados\_M***, de 8 bits.

* + 1. **Memória de Instruções**

É neste módulo que ficam armazenas as instruções do programa a ser executado. Apresenta uma dimensão de 14 bits, onde o endereço da instrução é determinado pelo sinal ***Endereço***, de 8 bits, e à saída é disponibilizado o ***opcode***, de 5 bits, o sinal ***SEL\_R***, de 1 bit, e o sinal ***Constante***, de 8 bits.

* + 1. **Processador**
       1. **Periférico de Entrada**

É neste módulo que é feita a comunicação do processador com o exterior, permitindo ao utilizador inserir dados para posteriormente serem realizadas operações com os mesmos. Alguns exemplos deste tipo de periféricos são o teclado e o rato.

Este módulo é controlado pelo sinal , de 1 bit, que quando está a ‘1’ é feita uma leitura dos dados de entrada, ***PIN***, de 8 bits, colocando-os na saída do periférico, ***Dados\_IN***, de 8 bits.

* + - 1. **Periférico de Saída**

Este módulo permite que o utilizador veja os dados e informações processados pelo computador. Alguns exemplos deste tipo de periféricos são o monitor, a impressora e colunas de som.

Este periférico é controlado pelo sinal ***ESCR\_P***, de 1 bit, que quando está a ‘1’, na transição ascendente do relógio (clk), escreve no sinal de saída, ***POUT***, de 8 bits, o valor do sinal à entrada do módulo, ***Operando1***, também de 8 bits.

* + - 1. **Multiplexer dos Registos (Mux R)**

Este módulo é responsável por encaminhar um dos quatro sinais disponíveis, de 8 bits, à sua entrada (***Resultado, Dados\_IN, Dados\_M e Constante***) para apresentar na sua saída, ***Dados\_R***, de 8 bits. O sinal a encaminhar depende do valor do sinal de entrada ***SEL\_Data***, de 2 bits.

|  |  |
| --- | --- |
| **SEL\_DATA** | **DADOS\_R** |
| 00 | Resultado |
| 01 | Dados\_IN |
| 10 | Dados\_M |
| 11 | Constante |

Tabela Sinal de saída do Mux R em função do sinal SEL\_DATA.

* + - 1. **Registos A e B**

A escrita nos registos A e B é controlada pelo sinal ***ESCR\_R***, de 1 bit. Quando o sinal está a ‘1’ o valor presente no sinal de entrada ***Dados\_R***, de 8 bits, é guardado no registo especificado pelo sinal ***SEL\_R***, de 1 bit, na transição ascendente do sinal de relógio (clk).

Estes registos estão continuamente a efetuas leituras. As saídas ***Operando1*** e ***Operando2***, ambas de 8 bits, apresentam os valores guardados nos registos A e B, respetivamente.

|  |  |
| --- | --- |
| **SEL\_R** | **Registo a ser escrito** |
| 0 | Registo A |
| 1 | Registo B |

Tabela Sinal de seleção de escrita nos registos A e B

* + - 1. **Unidade Aritmética e Lógica (ALU)**

Este módulo permite realizar operações aritméticas e lógicas, tal como o nome indica. No caso desta placa-mãe, a unidade aritmética e lógica do processador, é capaz de realizar as operações soma, subtração, AND, OR e XOR, com os sinais de entrada ***Operando1*** e ***Operando2***, ambos de 8 bits, que representam números inteiros com sinal. Os sinais de saída da ALU são determinados pelo sinal de seleção ***SEL\_ALU***, de 3 bits, A saída ***Resultado***, de 8 bits, será atualizada no caso de cada operação e, a saída ***COMP\_RES***, de 5 bits, será atualizada apenas quando é realizada uma comparação, cada um dos seus bits indicando o resultado de uma das cinco comparações apresentadas no módulo *Comparação*.

|  |  |
| --- | --- |
| **SEL\_ALU** | **Operação** |
| 000 | Operando1 **+** Operando2 |
| 001 | Operando1 **–** Operando2 |
| 010 | Operando1 **AND** Operando2 |
| 011 | Operando1 **OR** Operando2 |
| 100 | Operando1 **XOR** Operando3 |
| 101 | Operando1 > Operando2  Operando1 >= Operando2  Operando1 = Operando2  Operando1 <= Operando2  Operando1 < Operando2 |

Tabela Operações da ALU