

Arquitetura e Organização de Computadores

Turma A - 2018/01

Projeto da Memória d Instruções do MIPS

Objetivo: projetar, simular e sintetizar a memória de instruções do MIPS.

Descrição:

O MIPS unicolor e pipeline utiliza uma memória para armazenar programa e outra para armazenar os dados. Neste trabalho deve-se instanciar um módulo de memória da biblioteca de macrofunções do FPGA e iniciá-lo com um código gerado pelo MARS.

O bloco de memória é ilustrado na figura 1. Provê a seguinte interface:

- um barramento de endereço (suportar 8 bits de endereço)
- um barramento de dados de saída (32 bits)
- sinal de habilitação de escrita no PC (wPC)
- seleção de entrada do PC (m1)
- relógio (clk)

O módulo de memória pode ser gerado através do MegaWizard Plugin Manager, em Tools. O wizard tem uma interface interativa que permite gerar uma descrição *vhdl* de um RAM a ser mapeada nos blocos de RAM do FPGA. O módulo de memória deve ser configurado com largura de dados de 32 bits, largura de endereços de 8 bits, 256 palavras de 32 bits de armazenamento. O conteúdo inicial da memória pode ser definido através de um arquivo .MIF, que é especificado durante a geração do módulo de memória.

Um arquivo de inicialização da memória contendo o código de um programa MIPS pode ser gerado simplesmente criando-se um arquivo tipo MIF no Quartus II e escrevendo nele a saída obtida no MARS, textual e em hexa. Pode-se editar diretamente o arquivo texto, com um editor de texto qualquer fora do Quartus, ou usar a própria interface do Quartus para inicializar o arquivo. O arquivo texto MIF gerado está no formato abaixo:

```
-- Quartus II generated Memory Initialization File (.mif)
```

```
WIDTH=32;  
DEPTH=256;
```

```
ADDRESS_RADIX=HEX;  
DATA_RADIX=HEX;
```

```
CONTENT BEGIN  
  00      :      20082000;  
  01      :      8d090000;  
  02      :      8d0a0004;  
  03      :      8d0b0008;  
  04      :      8d0c000c;  
  05      :      8d0d0010;  
  06      :      01498020;  
  07      :      01498822;  
  08      :      018d9024;  
  09      :      012b982a;  
  [0A..FF] :      00000000;  
END;
```

Note que o sinal de habilitação de leitura não é necessário nesse tipo de RAM.

O módulo automaticamente gerado deverá ter uma interface como segue:

```
ENTITY ramtest IS
  PORT
  (
    address      : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
    clock        : IN STD_LOGIC;
    data         : IN STD_LOGIC_VECTOR (31 DOWNTO 0);
    wren         : IN STD_LOGIC := '0';
    q            : OUT STD_LOGIC_VECTOR (31 DOWNTO 0)
  );
END ramtest;
```

O *testbench* deverá gerar os sinais para carga e incremento do PC visualizando-se os códigos das instruções através dos mostradores de 7 segmentos.

