Laboratório de Sistemas Digitais

Trabalho Prático nº 9

Modelação de Memórias ROM e RAM em VHDL

Objetivos

- Modelação de memórias em VHDL com escrita síncrona e leitura assíncrona ou síncrona.
- Implementação de memórias de um ou mais portos (multi-porto).
- Parametrização de memórias.

Sumário

Este trabalho pretende introduzir vários tipos de memórias, a sua descrição em VHDL e implementação em FPGA. Na primeira parte vai ser implementada uma memória ROM (*Read Only Memory*), que tem armazenado um conjunto de valores predefinidos, os quais vão inicialmente ser lidos de forma manual e seguidamente de forma sequencial utilizando um contador. Na segunda parte vai ser modelada uma memória RAM (Random Access Memory), de leitura/escrita, em que estas operações vão ser controladas manualmente através de *switches* e o valor lido da memória vai ser visualizado diretamente nos LEDs.
Na terceira parte é solicitada a implementação de uma RAM de dois portos, um dos quais será utilizado para escrita através de *switches* e um de leitura assíncrona utilizando para o efeito um contador. A quarta e quinta partes são dedicadas à construção de modelos de memórias parametrizáveis. Finalmente a sexta parte é dedicada à leitura síncrona.

Parte I

1. Abra a aplicação "Altera Quartus II" e crie um novo projeto para a FPGA Altera Cyclone IV EP4CE115F29C7. Poderá designar o projeto e a entidade *top-level* como "**`ROM_Demo`**".

2. Descreva em VHDL uma memória ROM de 16 posições com um comprimento de palavra de 8 bits (entidade ROM_16_8). A informação armazenada na ROM deverá ser definida através de uma constante inicializada com os valores que entender adequados.

3. Crie um novo ficheiro, chamado “ROM_Demo.vhd”, onde deverá construir uma entidade e uma arquitetura para instanciar a ROM_16_8 e realizar a sua ligação a dispositivos do kit DE2-115, para que seja possível estabelecer o endereço de leitura através de *switches* e a visualização nos LEDs do valor lido da memória (valor armazenado no endereço especificado pelos *switches*).

4. Repita o ponto anterior utilizando agora um contador incrementado à frequência de 1Hz para endereçar a memória. Repita para frequências de 10Hz e 100Hz.

Parte II

1. Crie um novo projeto para a FPGA Altera Cyclone IV EP4CE115F29C7. Poderá designar o projeto e a entidade *top-level* como "**`RAM1_Demo`**".

2. Implemente em VHDL uma memória RAM de 16 posições com um comprimento de palavra de 8 bits e com um porto de acesso para escrita síncrona e leitura assíncrona (com um único endereço). Ao ficheiro poderá dar o nome de RAM_16_8.vhd. Os portos

de entrada e saída da entidade poderão seguir a nomenclatura adotada nas aulas teórico-práticas, ou seja **clk**, **writeEnable**, **address**, **writeData** e **readData**.

3. Simule a RAM criando um ficheiro com uma *testbench* adequada. O processo para aplicação dos vetores de simulação deverá permitir a escrita e leitura do conteúdo de vários endereços de memória (idealmente todos) – ver slides da aula TP para um exemplo de construção de uma *testbench* para uma memória.

4. Crie um novo ficheiro, chamado "RAM1_Demo.vhd", para instanciar a RAM_16_8, permitindo que a escrita/leitura seja controlada pelos *switches* e visualizada nos LEDs do kit DE2-115, isto é, utilizando os *switches* para definir o endereço da memória a aceder (**address**), o valor a escrever (**writeData**) e o sinal que habilita a escrita (**writeEnable**). Para o sinal de relógio deve usar um pulsador (KEY(x)). O valor da posição de memória endereçada (**readData**) deverá ser mostrado nos LEDs da placa.

Parte III

1. Crie um novo projeto para a FPGA Altera Cyclone IV EP4CE115F29C7. Poderá designar o projeto e a entidade *top-level* como " RAM2_Demo".

2. Implemente em VHDL uma memória RAM de 16 posições com um comprimento de palavra de 8 bits e com dois portos de acesso (um para escrita síncrona e outro para leirura assíncrona, existindo entradas de endereçamento separadas para escrita e leitura). Os portos de entrada e saída da entidade VHDL poderão seguir a nomenclatura adotada nas aulas teórico-práticas, ou seja **clk**, **writeEnable**, **writeAddress**, **writeData**, **readAddress** e **readData**. Ao ficheiro poderá dar o nome RAM_2port_16_8.vhd.

3. Crie um novo ficheiro, chamado "RAM2_Demo.vhd" para instanciar a RAM_2port_16_8. A escrita na RAM é controlada através de *switches* (tal como descrito na parte II deste trabalho). Para a leitura deverá utilizar um contador incrementado a um ritmo de 1Hz, que vai "varrer" (endereçar) toda a memória de forma cíclica. A saída **readData** deverá ser ligada a LEDs para visualização do valor lido da memória.

Parte IV

Repita a parte II fazendo a parametrização da memória, isto é, permitindo através de *generics* de VHDL especificar a profundidade (número de palavras de armazenamento) e a largura (número de bits por palavra). De seguida instancie no ficheiro *top-level* a memória parametrizável com a mesma dimensão que a de tamanho fixo usada anteriormente.

Parte V

Repita a parte III fazendo a parametrização da memória, isto é, permitindo através de *generics* de VHDL especificar a profundidade (número de palavras de armazenamento) e a largura (número de bits por palavra). De seguida instancie no ficheiro *top-level* a memória parametrizável com a mesma dimensão que a de tamanho fixo usada anteriormente.

Parte VI

Repita a parte III alterando o porto de leitura para que esta operação passe a ser realizada de forma síncrona com uma entrada **readClk**.

PDF criado em 29/04/2015 às 11:55:38