

Sistemas Reconfiguráveis – Eng. de Computação

Especificações para o terceiro trabalho

1º semestre de 2024

1. Objetivo

Nesse trabalho serão feitos dois projetos independentes de periféricos para microcontrolador: **port_io**, especificado no item 2 e **ram_256x8**, especificado no item 3. Cada projeto consiste em descrever em linguagem VHDL, comentar, simular o funcionamento e comentar os resultados da simulação, de acordo com as especificações apresentadas. Poderá ser usado código concorrente ou sequencial, e todas as entradas e saídas deverão ser do tipo **std_logic** ou **std_logic_vector**.

Deverá ser entregue um relatório do trabalho na forma de um documento padrão ABNT para trabalhos acadêmicos (Capa, folha de rosto, índice de figuras, etc, etc), em um arquivo no formato pdf, via Canvas. Nesse relatório, cada projeto deverá estar em um capítulo próprio.

Além do relatório, deverá ser entregue um arquivo compactado (.zip ou .rar), com **todos** os arquivos dos projetos gerados no ambiente Quartus. Nesse arquivo, cada projeto deverá estar em uma pasta própria.

2. Port_io

Esse bloco é uma porta de entrada/saída paralela de 8 bits, que permite o interfaceamento do controlador com sensores e atuadores digitais. Cada pino pode ser individualmente programado para funcionar como entrada ou saída, através do registrador **dir_reg**, que pode ser lido e escrito em um endereço definido mais adiante. O valor '0' em um bit desse registrador (valor *default* após o *reset*) implica que o pino correspondente da porta é uma entrada e o valor '1' configura o pino correspondente como saída. Os pinos configurados como saída recebem o valor escrito no registrador **port_reg**, que é um registrador apenas para escrita. O estado de todos os pinos pode ser lido no mesmo endereço usado para escrever no registrador **port_reg**.

2.1. Lado do processador

nrst	Entrada de <i>reset</i> assíncrono. Quando ativada (nível lógico baixo), todos os registradores deverão ser zerados. Esta entrada tem preferência sobre todas as outras.
clk_in	Entrada de <i>clock</i> do sistema. Todo os registradores devem ser escritos em sincronia com a borda de subida desse <i>clock</i> .
abus[7..0]	Entrada de endereçamento para os registradores internos. Os endereços dos registradores são especificados através de um "generic", conforme detalhado no item 2.3.
dbus[7..0]	Barramento bidirecional de dados, com 8 bits. Durante as operações de leitura, comporta-se como saída e durante as operações de escrita, como entrada. Quando não em uso, fica em alta impedância.
wr_en	Entrada de habilitação para escrita nos registradores. Quando ativada (nível lógico alto), um dos registradores dir_reg ou port_reg será escrito, síncrono com clk_in , com o valor de dbus , desde que o endereço correspondente esteja presente em abus . Caso contrário, nenhuma ação será efetuada.
rd_en	Entrada de habilitação para leitura. Quando ativada (nível lógico alto), o barramento dbus deverá receber o conteúdo do registrador dir_reg ou do latch , desde que o endereço correspondente esteja presente em abus . Quando desativada ou quando o endereço não corresponder a um dos registradores, a saída deverá ficar em alta impedância ("ZZZZZZZZ").

Obs.: As entradas **wr_en** e **rd_en** nunca serão ativadas simultaneamente.

2.2. Lado da porta

`port_io[7..0]` Porta bidirecional ou seja, modo = INOUT, com 8 bits. A direção será configurada, bit a bit, através do registrador **dir_reg**. Para cada bit de **port_io**, se o bit correspondente no registrador **dir_reg** for igual a '0', o pino será uma entrada. Se for igual a '1', será uma saída.

2.3. Endereçamento

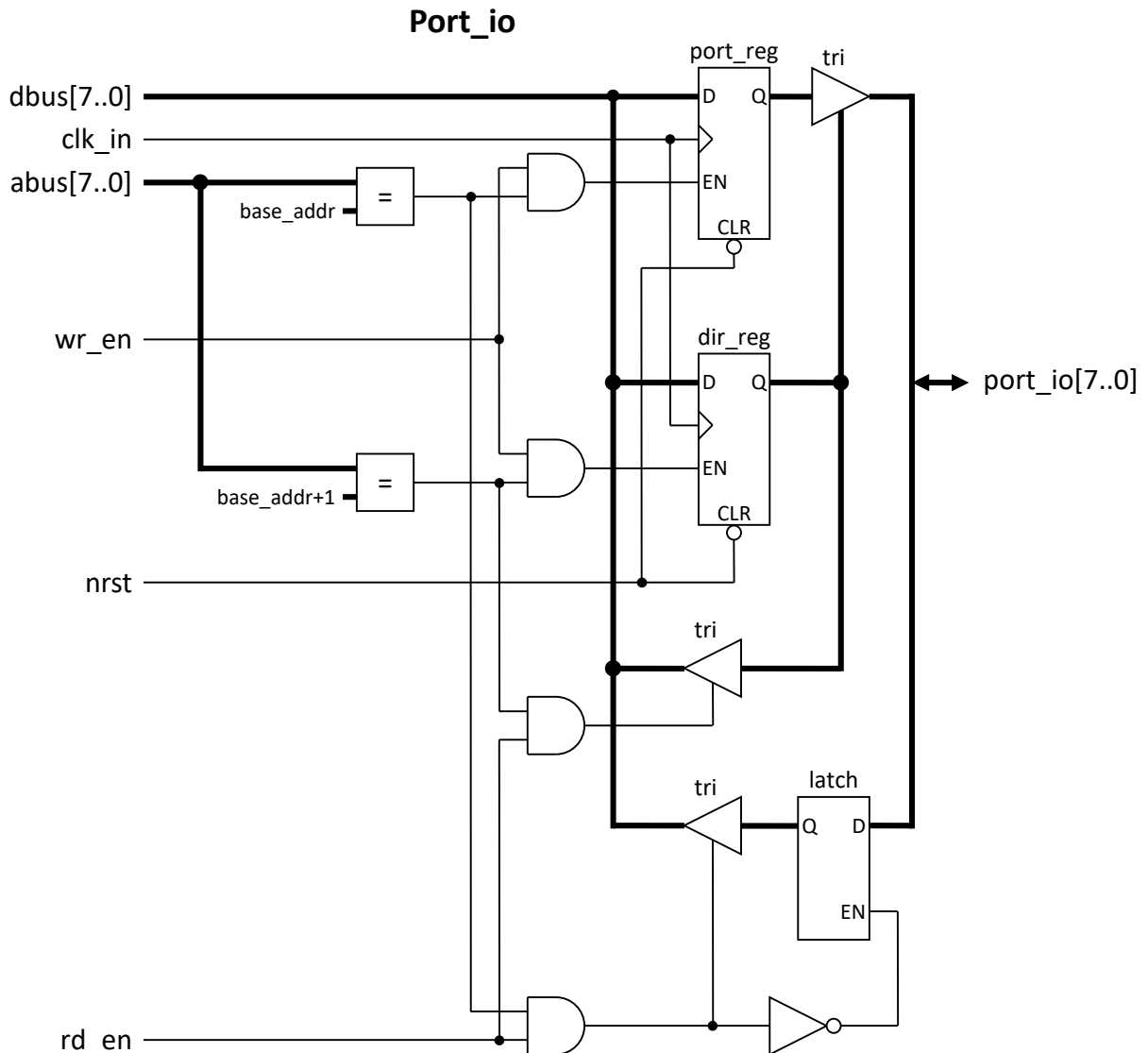
Para permitir o instanciamento de mais de um bloco da porta de entrada e saída, o endereço base de cada bloco será especificado através de um “generic” **base_addr** incluído na “entity” do código. O endereço de cada registrador está definido na tabela abaixo:

abus	wr_en	rd_en	operação
base_addr	1	0	escrita no port_reg
base_addr	0	1	leitura no latch
base_addr + 1	1	0	escrita no dir_reg
base_addr + 1	0	1	leitura no dir_reg

2.4. Diagrama simplificado

Existem dois registradores e um *latch* nesse bloco. O registrador **dir_reg**, (endereço **base_addr** + 1) configura a função de cada bit da porta, se entrada ou saída. Inicialmente, após o *reset*, o **dir_reg** tem todos os bits em '0', configurando, assim, todos os bits da porta como entrada. Para configurar um bit da porta como saída, deve ser escrito '1' no bit correspondente do **dir_reg**. O conteúdo de **dir_reg** pode ser lido no mesmo endereço, com o sinal **rd_en** ativado. O registrador **port_reg** (endereço **base_addr**) armazena os dados de saída da porta. Após o *reset*, todos os bits ficam zerados. A entrada da porta pode ser lida no mesmo endereço do registrador **port_reg**, com o sinal **rd_en** ativado. Durante a operação de leitura na porta, os dados de entrada são travados em um *latch*.

O diagrama é apresentado na próxima página.



3. Ram_256x8

Esse bloco é uma memória RAM de 256 bytes.

3.1. Entradas e saídas

nrst	Entrada de <i>reset</i> assíncrono. Quando ativada (nível lógico baixo), todas as posições de memória deverão ser zeradas. Esta entrada tem preferência sobre todas as outras.
clk_in	Entrada de <i>clock</i> do sistema. A memória deve ser escrita em sincronia com a borda de subida desse <i>clock</i> .
addr[7..0]	Entrada de endereçamento para a memória. Serve tanto para escrita quanto para a leitura.
dio[7..0]	Barramento bidirecional de dados, com 8 bits. Durante as operações de leitura, comporta-se como saída e durante as operações de escrita, como entrada. Quando não em uso, fica em alta impedância (“ZZZZZZZZ”).
mem_wr_en	Entrada de habilitação para escrita na memória. Quando ativada (nível lógico alto), a posição de memória endereçada por addr será escrita, síncrona com clk_in , com o valor de dio que, nessa operação, funciona como entrada. Quando desativada, nenhuma escrita é realizada.
mem_rd_en	Entrada de habilitação para leitura. Quando ativada (nível lógico alto), o barramento dio deverá receber o conteúdo da posição de memória endereçada por addr . Quando não

ativada, o barramento **dio** fica em alta impedância (“ZZZZZZZZ”). A leitura é uma operação assíncrona, ou seja, não depende de **clk_in**.

Obs.: As entradas **mem_wr_en** e **mem_rd_en** nunca serão ativadas simultaneamente, não necessitando, portanto, de uma proteção lógica para ativação simultânea.

3.2. Diagrama

