PSI-3451 Projeto de CI Lógicos Integrados

Aula 5- Conceitos relacionados aos circuitos modelados em VHDL

Nesta aula, o aluno receberá os arquivos de 3 módulos combinatórios utilizados no ***Snake***:

- somador *ripple-carry* (*rc_adder_2*) em um modelo estrutural com a construção *generate*
- somador *ripple-carry* (*rc_adder_3*) em um modelo estrutural com um *shifter*
- unidade lógica aritmética (*alu_1*)

A seguir apresentamos uma breve descrição funcional e temporal de cada módulo.

Somador *ripple-carry* estrutural com *generate* (módulo rc_adder_2.vhd)

Este somador é equivalente ao circuito que o aluno construiu na Aula 4. Trata-se de um módulo somador do tipo *ripple-carry* com tamanho de palavra parametrizado em "*width*". O circuito, como ilustrado na figura 1, contém dois vetores de entrada *a_i* e *b_i*, e uma entrada de 1 bit *carry-in* (*c_i*). Na saída, temos o vetor de soma, *z_o* e o sinal de 1 bit *carry-out* (*c_out*).

O arquivo VHDL deste somador (*rc_adder_2.vhd*) corresponde a um modelo estrutural utilizando como componente qualquer uma das descrições *full_adder* já trabalhadas nas aulas anteriores. Em particular, no projeto ***Snake***, usaremos o modelo comportamental *dataflow* testado na aula 2.

O modelo *rc_adder_2.vhd* é um somador *ripple-carry* tradicional, porém utilizando a sintaxe *generate*, uma construção alternativa do VHDL para montar de maneira simplificada operações com vetores.

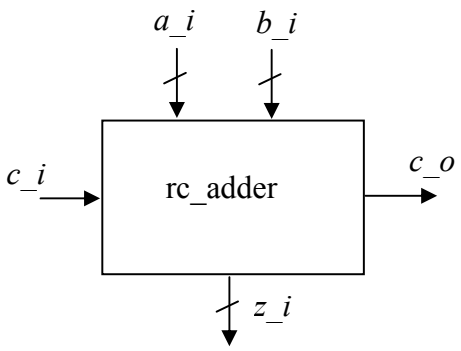

Figura 1. Entradas e saídas do módulo somador *ripple_carry*

Unidade Lógica Aritmética (módulo *alu.vhd*)

O módulo *ULA* do projeto ***Snake*** é um circuito especializado que realiza duas operações: soma e deslocamento (*shift*). A funcionalidade detalhada e seus objetivos constam do documento de descrição do ***Snake***. Na descrição VHDL, a operação de soma é

realizada pelo módulo *rc_adder* descrito acima, enquanto a operação *shift* é realizada através de uma descrição *dataflow*.

Como ilustrado na Figura 2, o somador realiza a soma do valor em *rb_op* com o valor de *op_first.* Este último pode estar deslocado ou não[1], sendo a seleção feita pelo *mux* com seleção de mux, *ctrl_x_y*, em 1 e 0, respectivamente. A saída *ofc_result* recebe o valor *rb_op* ou o resultado da soma *result_s*, sendo a seleção feita pelo *mux*, com seleção *ctrl_pass_calc*, em 0 e 1, respectivamente.

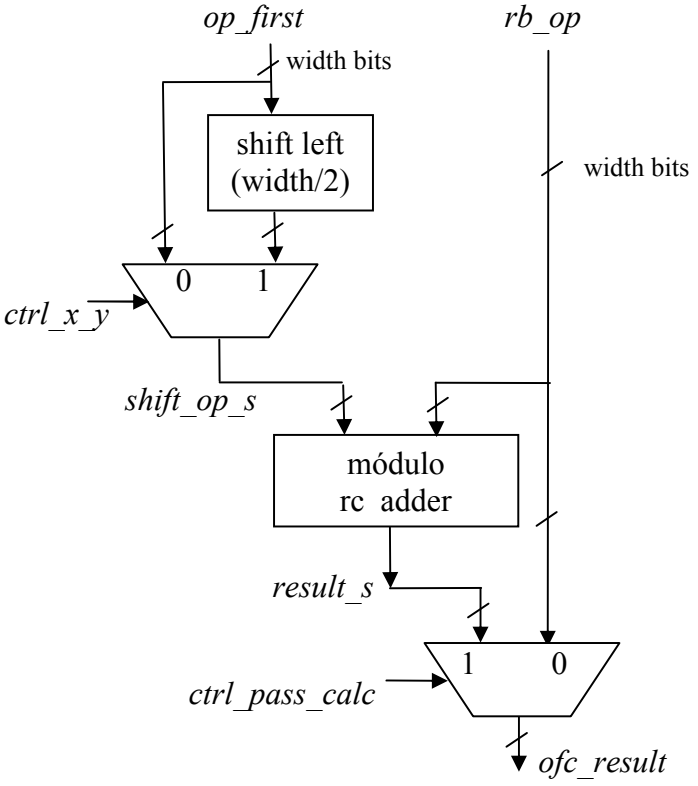

Figura 2 - Módulo *ALU*

Somador *ripple-carry* estrutural com *shifter* (módulo rc_adder_3.vhd)

Para que o aluno exercite a construção *generate*, é dado ao aluno código do arquivo *rc_adder_3.vhd* que corresponde ao circuito da Figura 3. A implementação do *shifter* é realizada *bit-a-bit* junto aos *full-adders* do circuito *ripple-carry*.

Pode-se observar que para *ctrl_x_y*= 0, o vetor *a_i* será somado ao vetor *b_i*, enquanto, para o caso de *ctrl_x_y*= 1, a metade dos bits (menos significativos) de *b_i* será somada a 0 e a outra metade dos bits (mais significativos) será somada à metade dos bits (menos significativos) de *a_i*.

[1] O sinal *of_first* carrega os valores de incremento (-1, +1 e +3). O deslocador tem a função de passar estes valores de incremento nas coordenadas x da tela do jogo ***Snake*** (a metade dos bits menos significativos) para a as coordenadas y (a metade dos bits mais significativos)

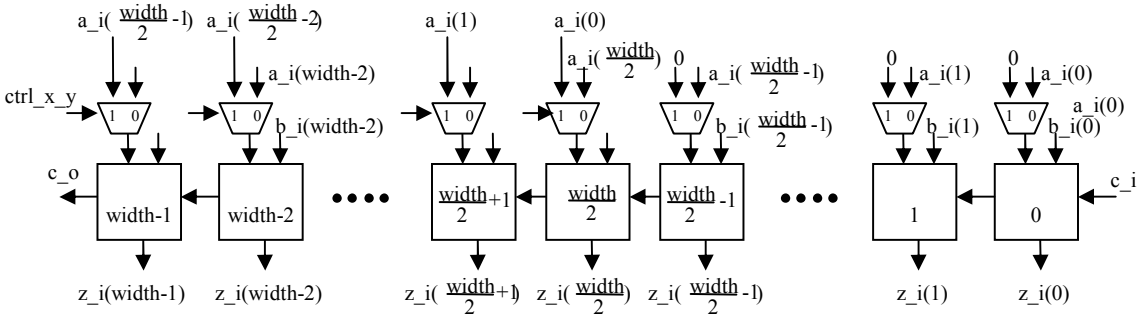

Figura 3. Módulo somador *ripple_carry* com shifter, composto de *muxes* e *full_adders*