Laboratório de Sistemas Digitais

Trabalho Prático nº 5

Circuitos sequenciais elementares: registos de deslocamento
(comparação com blocos combinatórios de deslocamento – em TPC)

Objetivos

- Modelação em VHDL, simulação, implementação em FPGA e teste de circuitos combinatórios e sequenciais elementares de deslocamento/rotação.

Sumário

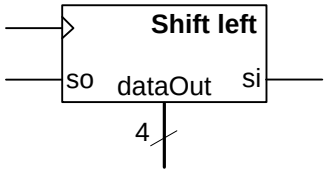
Os blocos funcionais de deslocamento – combinatórios ou sequenciais – permitem produzir uma saída de "**n**" bits deslocada à esquerda ou à direita (em relação ao índice dos bits que compõem o vetor de saída). Este deslocamento pode ser meramente combinatório (a saída é deslocada relativamente a um vetor de entrada), ou sequenciais (a saída é deslocada relativamente ao anterior valor do vetor de saída).
O deslocamento, à esquerda ou à direita, quando o vetor de saída é interpretado como uma quantidade numérica inteira (com ou sem sinal) representam respetivamente uma operação de multiplicação por 2^n ou o quociente de uma divisão por 2^n. No caso da multiplicação, "**n**" é igual ao número de deslocamentos à esquerda e pressupõe-se que os bits acrescentados à direita são zeros. No caso da divisão, "**n**" é igual ao número de deslocamentos à direita e pressupõe-se que os bits acrescentados à esquerda são iguais ao valor lógico do bit mais significativo original (para quantidades em complemento para dois), ou é '**0**' (para quantidades representadas como inteiros positivos – sem sinal).

Este trabalho prático está dividido em duas partes. Na primeira parte é implementado um *shift-register* sequencial do tipo *serial in / parallel out* com dimensão parametrizável. A segunda parte é dedicada à descrição comportamental e implementação de um *shift-register* genérico, com *parallel load*, capaz de efetuar operações de deslocamento e rotação à esquerda e à direita, e suportando igualmente o deslocamento aritmético à direita e à esquerda.

Parte I

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

2. Descreva em VHDL um *Shift Register left* de 4 bits, com a interface representada na figura ao lado. Guarde o seu código num ficheiro com o nome "**ShiftRegister4.vhd**".



3. Efetue a simulação do componente modelado.

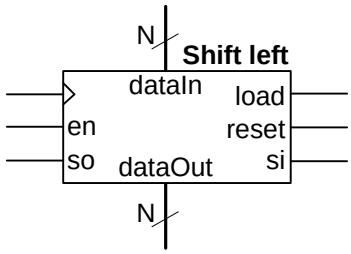
4. Copie o ficheiro "**ShiftRegister4.vhd**" para "**ShiftRegisterN.vhd**" e altere-o de modo a que a dimensão do *shift register* seja parametrizável, definida com uma

constante genérica "**size**" fixada aquando da instanciação do registo. Esta constante poderá ter o valor por omissão 4.

5. Crie um novo ficheiro VHDL, chamado "**ShiftRegister_Demo.vhd**", de modo a que instancie o *shift register* parametrizável com uma dimensão de 8 (i.e. deve atribuir, aquando da instanciação e usando a construção **generic map**, o valor 8 à constante genérica "**size**"). Ligue a entrada "**clk**" à saída de um divisor de frequência que gere um relógio com a frequência de 1Hz e as restantes entradas a interruptores e as saídas a LEDS (NOTA: No caso de não ter chegado a instanciar um divisor de frequência no guião anterior, pode usar o código disponível no final deste guião).
6. Efetue a síntese e implementação do projeto, programe a FPGA e teste o funcionamento do componente.

TPC:

1. Copie o ficheiro "**ShiftRegister4.vhd**" para o ficheiro "**ShiftRegisterLoadN.vhd**" e altere-o de modo a modelar um *Shift Register* de N bits, de acordo com o bloco lógico da figura, em que as entradas de "**reset**" e "**load**" são síncronas.
2. Efetue a simulação do novo componente.



Parte II

1. Crie um projeto para a FPGA Altera Cyclone IV EP4CE115F29C7. Poderá designar o projeto e a entidade *top-level* como "**SeqShiftUnit_Demo**".
2. Dentro deste projeto crie um novo ficheiro VHDL com o nome "**SeqShiftUnit.vhd**".
3. Transcreva para o seu ficheiro o código que se apresenta em seguida. Complete o código por forma a que, para além do *parallel load*, este circuito possa efectuar as seguintes operações: *shift left* lógico; *shift right* lógico; *shift left* aritmético; *shift right* aritmético; *rotate left*; *rotate right*.

   NOTA: "**siLeft**" e "**siRight**" são respetivamente as entradas série para os *shifts* lógicos à esquerda e à direita. Relembre a matéria de ISDig e das aulas TP de LSDig, e anote no seu *log book* uma breve descrição funcional de cada uma das operações indicadas neste ponto.
4. Sintetize o circuito criado, verifique a sua correção sintática e efetue a simulação do componente modelado.
5. Crie um novo ficheiro VHDL, chamado "**SeqShiftUnit_Demo.vhd**", de modo a que instancie o *shift register* que programou. Ligue a entrada "**clk**" à saída do divisor de frequência que usou na *Parte I*, as restantes entradas a interruptores e as saídas a LEDs à sua escolha. Teste o funcionamento do circuito na sua placa de desenvolvimento.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;

entity SeqShiftUnit is
  port(clk       : in  std_logic;
       dataIn    : in  std_logic_vector(7 downto 0);
       siLeft    : in  std_logic;
       siRight   : in  std_logic;
       loadEn    : in  std_logic;
       rotate    : in  std_logic;
       dirLeft   : in  std_logic;
       shArith   : in  std_logic;
       dataOut   : out std_logic_vector(7 downto 0));
end SeqShiftUnit;

architecture RTL of SeqShiftUnit is
  signal s_shiftReg : std_logic_vector(7 downto 0);

begin
  process(clk)
  begin
    if (falling_edge(clk)) then

      if (loadEn = '1') then
        s_shiftReg <= dataIn;

      elsif (rotate = '1') then
        if (dirLeft = '1') then
          s_shiftReg <= s_shiftReg(6 downto 0) & s_shiftReg(7);
        else
          s_shiftReg <= s_shiftReg(0) & s_shiftReg(7 downto 1);
        end if;

       elsif (shArith = '1') then
--
--    Complete o código aqui
--
       end if;
    end if;
  end process;

  dataOut <= s_shiftReg;
end RTL;
```

TPC:

1. Crie um projeto para a FPGA Altera Cyclone IV EP4CE115F29C7. Poderá designar o projeto e a entidade *top-level* como "**`CombShiftUnit_Demo`**".
2. Dentro deste projeto crie um novo ficheiro VHDL com o nome "**`CombShiftUnit.vhd`**".
3. Escreva o código necessário para instanciar um módulo de deslocamento **combinatório** de 8 bits, capaz de efetuar as mesmas operações que o *shift register* da *Parte II.*
5. Sintetize o circuito criado, verifique a sua correção sintática e efetue a simulação do componente modelado.
6. Crie um novo ficheiro VHDL, chamado "**`CombShiftUnit_Demo.vhd`**", onde deverá instanciar o *shifter* anteriormente modelado e associar os respetivos portos aos seguintes pinos concretos da FPGA do kit de desenvolvimento: **`dataIn->SW(7:0)`**, **`rotate->KEY(0)`**, **`dirLeft->KEY(1)`**, **`shArith->KEY(2)`**, **`shAmount->SW(17:15)`** e **`dataOut->LEDR(7:0)`**.
7. Efetue a síntese e compilação do projeto, programe a FPGA e teste o funcionamento do componente.

:::

```vhdl
--
-- Divisor de frequência parametrizável
--
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;

entity ClkDividerN is
  generic(divFactor : integer := 10);
  port(clkIn        : in  std_logic;
       clkOut       : out std_logic);
end ClkDividerN;

architecture Behav of ClkDividerN is
  signal s_divCounter : integer := 0;
begin
  process(clkIn)
  begin
    if (rising_edge(clkIn)) then
      if (s_divCounter = (divFactor - 1)) then
        clkOut       <= '0';
        s_divCounter <= 0;
      else
        if (s_divCounter = (divFactor / 2 - 1)) then
          clkOut <= '1';
        end if;
        s_divCounter <= s_divCounter + 1;
      end if;
    end if;
  end process;
end Behav;
```