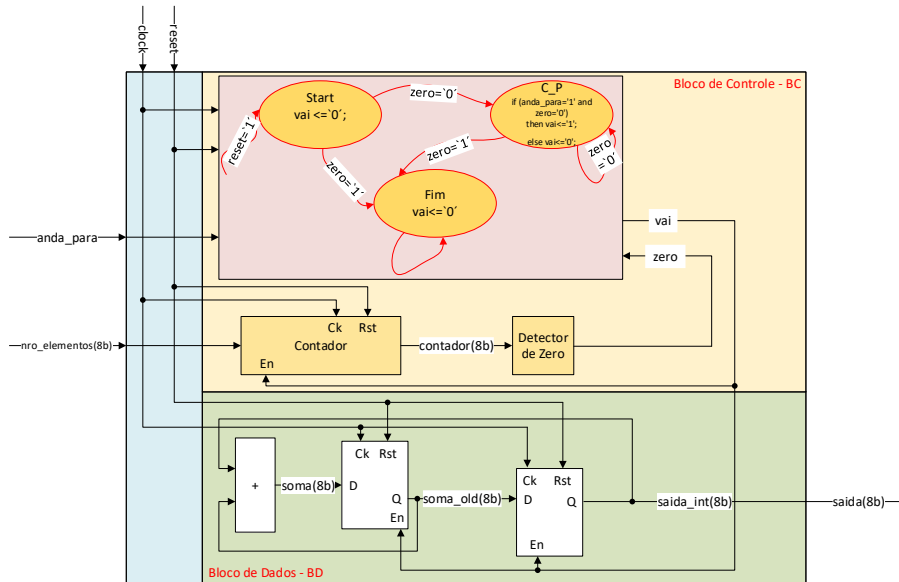


Aluno:

10/dezembro/2021

1. (4,0 pontos). Um hardware usado na disciplina teórica de Organização e Arquitetura de Processadores deste semestre foi o Fibonacciero, cujo diagrama de blocos é dado abaixo.



Disponibiliza-se junto com este enunciado o material de desenvolvimento deste hardware, incluindo seu código VHDL, um *testbench* em VHDL, as versões em esquemáticos da implementação do mesmo e mesmo um arquivo de configuração de formas de onda. Esta questão versa sobre como modificar este hardware para aumentar minimamente sua capacidade de operação.

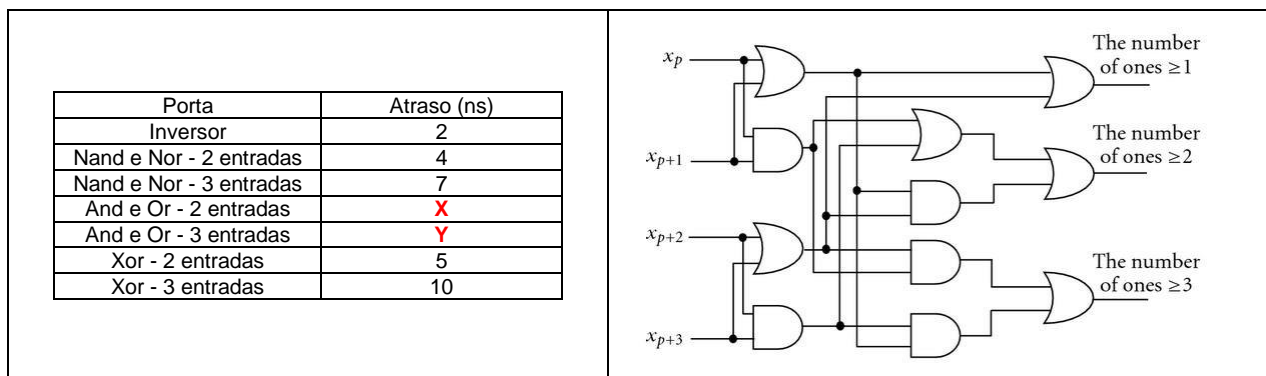
A ideia é gerar uma nova versão do Fibonacciero, com uma nova funcionalidade, descrita assim: cada vez que o hardware está parado sem ter atingido o último valor previsto para ser gerado no reset do sistema, deve ser possível entrar com um novo valor de contagem do número de elementos, alterando desta forma o último elemento a ser gerado.

Por exemplo, imagine que no reset do hardware (**reset**=‘1’) o valor 5 estava colocado na entrada **nro\_elementos**. Assim depois que o **reset** vai para ‘0’, o hardware passa a gerar os elementos da série de Fibonacci, do primeiro ao quinto, um a cada borda de subida do **clock**. Mas imagine que depois de gerar o terceiro elemento, o sinal **anda\_para** vai para ‘0’. Em seguida a nova entrada **recarga** (veja item a abaixo) vai para ‘1’ por 1 ou mais clocks e ao mesmo tempo o sinal **nro\_elementos** recebe o valor 10. Depois disto, **anda\_para** e **recarga** voltam para ‘1’ e ‘0’, respectivamente. O que se espera então é que ao invés de parar após gerar o quinto elemento da série, para-se apenas depois de gerar o décimo terceiro elemento dela.

Para produzir o novo hardware, a interface externa e a arquitetura interna do hardware devem, é claro, ser alteradas. O trabalho nesta questão consiste em gerar esta nova versão do hardware, e mostrar que ele funciona, usando um novo *testbench*. Algumas dicas para realizar a questão são:

- Crie um sinal externo novo, chamado por exemplo de **recarga**, que vai habilitar a nova funcionalidade. Assuma que todas as entradas (**anda\_para**, **recarga** e **nro\_elementos** são sincronizadas pelo **clock**).
- Altere os módulos internos do Fibonacciero para que a nova funcionalidade opere corretamente. Deixe claro na resposta a esta questão (ou seja, descreva com suas palavras) que módulos foram alterados ou criados, e como os módulos existentes foram alterados para dar suporte à nova funcionalidade.
- Crie um *testbench* que seja capaz de validar que o circuito pode operar exatamente como antes, mas que também mostre que ele dá suporte à nova funcionalidade.

- d) Anexe à sua resposta os novos arquivos .vhd e/ou .wcfg, identificando-os adequadamente para fins de avaliação. Dê também imagem(ns) de forma(s) de onda gerada(s) com seu *testbench*, demonstrando que o hardware opera corretamente.
2. (3,0 pontos). Um circuito combinacional interessante é um que capaz de contar o número de dígitos em 1 (ou 0) em uma entrada de múltiplos bits. Refira-se à Figura abaixo. No lado direito, ela apresenta um diagrama de portas lógicas que gera esta informação (o número de 1s) para vetores de 4 bits. No lado esquerdo ela possui uma tabela de valores de atrasos para cada tipo de porta. Os valores **X** e **Y** serão definidos de forma individual para cada aluno realizando esta prova. Interajam com o professor da disciplina para obter os valores específicos de **X** e **Y** a usar.
- Deve-se descrever em VHDL um módulo que implemente este circuito, usando os dados de atraso mostrados na Tabela do lado esquerdo da Figura.
  - Deve-se gerar um *testbench* em VHDL para este circuito e deve-se testar sua funcionalidade por simulação da forma mais extensa possível, documentando o comportamento, com formas de onda, nas respostas da prova, com comentários.
  - Deve-se responder o seguinte: (1) O circuito pode gerar *glitches* na saída ou não? (2) É possível estimar a velocidade máxima de operação correta deste circuito com os atrasos aqui definidos? Explique como fazer isto e responda isto para o seu caso de circuito.
- Anexe os arquivos .vhd e/ou .wcfg gerados à sua resposta, identificando-os adequadamente para fins de avaliação.



3. (3 pontos). Dada a descrição VHDL abaixo, deve-se fazer o que se pede. Os valores **M** e **N** serão definidos de forma individual para cada aluno realizando esta prova. Deve-se interagir com o professor da disciplina para obter os valores específicos de **M** e **N** a usar.

```

1.  entity Dut is
2.      port (a, b: in std_logic;
3.            c: out std_logic;
4.            d: in std_logic_vector(3 downto 0);
5.            e: out std_logic_vector(3 downto 0));
6.  end Dut;
7.  architecture Dut of Dut is
8.      signal x: std_logic;
9.      signal y: std_logic_vector(3 downto 0) := "0001";
10. begin
11.     x <= a nand b after Mns;
12.     c <= x;
13.     y <= "000" & x after 5ns;
14.     e <= d + y after Nns;
15. end Dut;

```

- Deve-se desenhar o circuito acima, usando convenções adequadas para representar o número de fios em cada sinal e para representar o atraso de portas e/ou de fios do circuito;
- Deve-se descrever um *testbench* em VHDL para testar este circuito o mais extensamente possível, e simular o mesmo por tempo suficiente;

c) Deve-se desenhar o que se espera obter como comportamento para cada um dos sinais ao simular o *testbench*. Claro, sempre é possível gerar uma forma de onda via simulação do *testbench* criado e mostrá-la como resposta.