



Roteiro 4 Introdução ao VHDL

| Equipe: | |
|-----------|------|
| Matricula | Nome |
| | |
| | |
| | |

1 Objetivos

- Travar um primeiro contato com a ferramenta para simulação de descrições em VHDL (ModelSim);
- Descrever sistemas digitais **combinacionais** em VHDL;
- Entender a estrutura básica de uma descrição em VHDL;
- Entender o esquema de subsistemas;

2 Descrição do Problema

Neste roteiro projetaremos dois subsistemas. O primeiro deles tem como função calcular a paridade par de um conjunto de quatro bits de entrada 'a', 'b', 'c' e 'd'. A paridade será indicada na saída 's'.



Figura 1 – diagrama esquemático da entidade `parityEven`

O segundo deles produzirá três saídas nomeadas `s2`, `s1`, `s0` que representará em binário o número de 1s existentes nas entradas 'a', 'b', 'c' e 'd'.



O primeiro dos circuitos a ser projetado é apresentado a seguir:

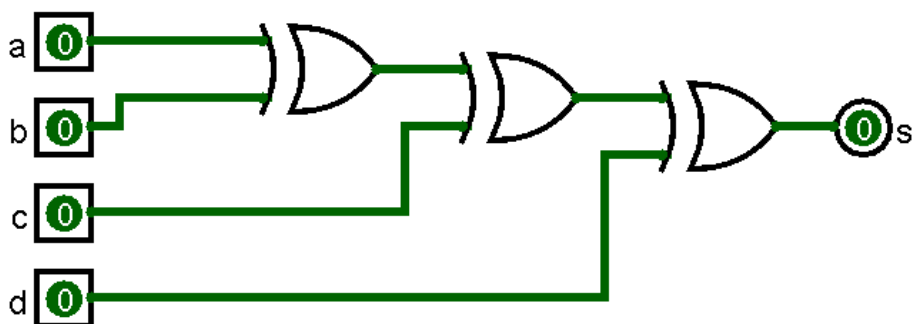


Figura 2 – circuito do gerador de paridade par

O "esqueleto" do subsistema parityEven é apresentado a seguir:

```
-----  
--comentários gerais do arquivo  
--parityEven.vhd  
  
-----  
--declaração das bibliotecas  
library ieee;  
use ieee.std_logic_1164.all;  
  
--declaração da entidade  
  
entity parityEven is  
port(  
  
    );  
end entity parityEven;  
  
--declaração da arquitetura  
  
architecture comb of parityEven is  
begin  
  
  
end architecture comb;
```

Utilize o programa "ModelSim" instalado nas máquinas do laboratório para implementar os subsistemas do roteiro. Crie um projeto chamado "lab4" e quando apresentado, escolha a criação de um arquivo VHDL. A seguir digite o "esqueleto" do sistema apresentado acima e por fim preencha os campos port da entidade e o corpo da arquitetura. (Na dúvida acerca da sintaxe busque ajuda na internet)

O segundo subsistema a ser implementado deve ser capaz de contar o número de 1s que ocorrem nos bits de entrada. A Tabela 1 lista todos os possíveis estados do subsistema "numOnes".

Tabela 1 – tabela verdade do circuito contador de sinais iguais a '1'

| a | b | c | d | s2 | s1 | s0 |
|---|---|---|---|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 |

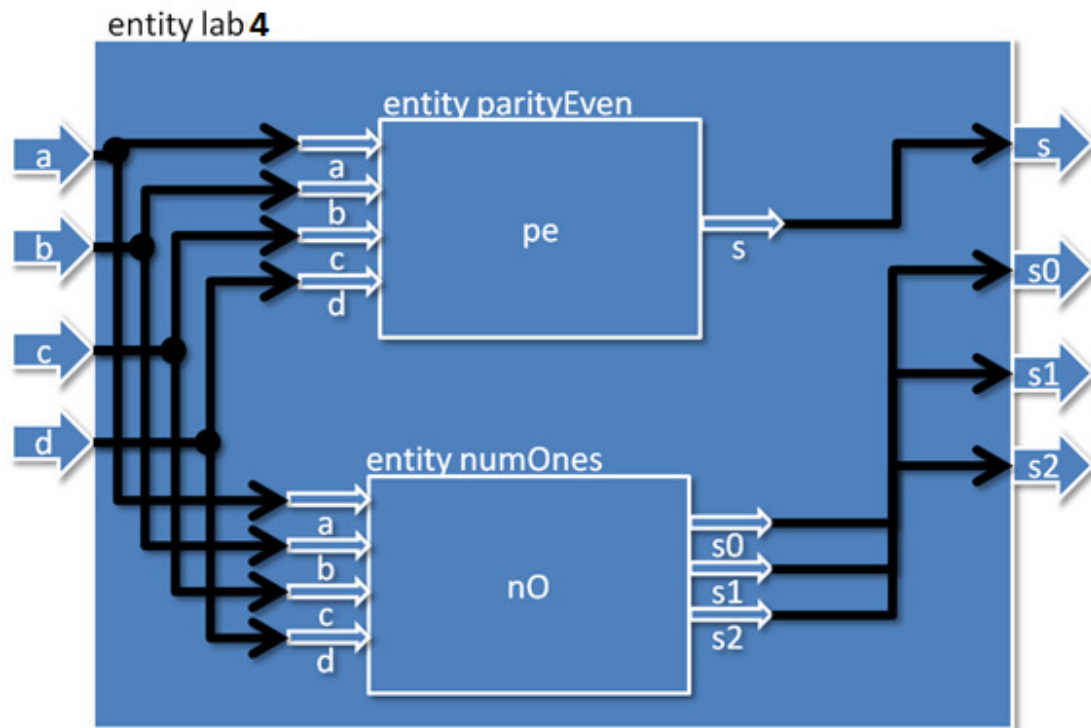
Levante a soma de produtos a partir da tabela acima e a implemente na arquitetura do subsistema numOnes.

\sum numOnes = _____

A seguir simplifique a soma de produtos pelo método de sua preferência:

numOnes_{min} = _____

Por fim una ambos os subsistemas previamente criados em um único sistema chamado "lab4" (nome da entidade) tal como apresentado no diagrama a seguir. Note que as entradas (a,b,c,d) e saídas (s, s0, s1, s2) devem ser mapeadas no campo "PORT" da entidade. As linhas em preto devem ser implementadas utilizando "signals" declarados na arquitetura de "lab4".



Boa Diversão Pessoal!