

Códigos de Detecção de Erros e Circuitos de Paridade

1 Objetivos deste tópico

Ao final do estudo deste tópico você saberá:

- O conceito de Cubo-n e Distância de Hamming;
- o conceito de códigos para detecção e correção de erros;
- As portas OU-Exclusivo e NOU-Exclusivo;
- Os conceitos de paridade, paridade par e paridade ímpar;
- Os circuitos de geração e detecção de paridade.

Leitura recomendada : seções do livro do Wakerly

- 2.14 - n-Cubes and Distance
- 2.15 - Codes for Detecting and Correcting Errors
 - 2.15.1 - Error Detecting Codes
 - 2.15.2 - Error Correcting and Multiple-Error-Detecting Codes
- 6.8 - Exclusive-OR Gates and Parity Circuits
 - 6.8.1 - Exclusive-OR and Exclusive-NOR Gates
 - 6.8.2 - Parity Circuits
 - 6.8.3 - The 74x280 9-Bit Parity Generator
 - 6.8.4 - Parity-Checking Applications
 - 6.8.6 - Exclusive-OR Gates and Parity Circuits in VHDL

Keywords: n-cube, Hamming distance, m-subcube, error, failure, error model, single error, multiple error, error detecting code, noncode word, minimum distance, information bit, parity bit, even-parity bit, odd-parity bit, check bits, error correction, error correcting code, odd-parity circuit, even-parity circuit, 74x280, exclusive-or gate

2 Exercícios

1. Mostre que $a \oplus b = b \oplus a$.
2. Mostre que $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
3. Considere um código com palavras de 4 bits e que o bit mais significativo seja o bit de paridade. Projete um detector de paridade ímpar, usando as estruturas *daisy-chain* e em árvore. Mostre a Tabela Verdade e o diagrama lógico dos circuitos resultantes. Como seria o detector de paridade par?
4. Analise o *datasheet* do 74x280 (<http://www.ti.com/lit/ds/sdas038c/sdas038c.pdf>) e verifique se a descrição VHDL seguinte corresponde a uma possível implementação deste componente. Simule o circuito usando as arquiteturas *parity9b* e *parity9s* e compare o resultado com o da Figura ???. Para simular o circuito, use o testbench *parity9_tb.vhd*.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity parity9 is
  port (
    I : in std_logic_vector (1 to 9);
    EVEN, ODD : out std_logic
  );
end entity parity9;

architecture parity9b of parity9 is
begin
  process (I)
    variable p : std_logic;
  begin
    p := I(1);
    for j in 2 to 9 loop
      if I(j) = '1' then
        p := not p;
      end if;
    end loop;
    ODD <= p;
    EVen <= not p;
  end process;
end architecture parity9b;

architecture parity9s of parity9 is
  component vxor3
  port (
    A, B, C : in std_logic;
    Y : out std_logic
  );
```

```

    end component;
    signal Y1, Y2, Y3, Y3N : std_logic;
begin
    U1 : vxor3 port map (I(1), I(2), I(3), Y1);
    U2 : vxor3 port map (I(4), I(5), I(6), Y2);
    U3 : vxor3 port map (I(7), I(8), I(9), Y3);
    Y3N <= not Y3;
    U4 : vxor3 port map (Y1, Y2, Y3, ODD);
    U5 : vxor3 port map (Y1, Y2, Y3N, EVEN);
end architecture parity9s;

library IEEE;
use IEEE.std_logic_1164.all;

entity vxor3 is
    port (
        A, B, C : in std_logic;
        Y : out std_logic
    );
end entity vxor3;

architecture vxor3_arch of vxor3 is
begin
    Y <= A xor B xor C;
end architecture vxor3_arch;

```

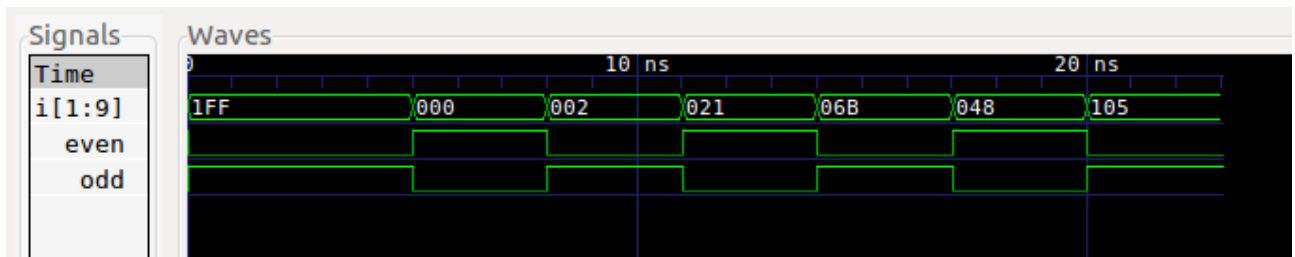


Figura 1: Diagrama de Tempos do detector de paridade de 9 bits

- Projete um detector de paridade ímpar de 25 bits usando o componente 74x280. Use mais de um se for necessário.