Vitor Amadeu Souza

FPGA

Aplicações em VHDL
Volume II

© 2023 by Cerne Tecnologia e Treinamento Ltda.

© 2023 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Maio de 2023

Direitos reservados por:

Cerne Tecnologia e Treinamento Ltda

Produção: *Cerne Tecnologia e Treinamento*

E-mail da Empresa: *cerne@cerne-tec.com.br*

Home Page: *www.cerne-tec.com.br.com.br*

Atendimento ao Consumidor: *sac@cerne-tec.com.br*

Contato com o Autor: *vitor@cerne-tec.com.br*



FEITO NO BRASIL

“Melhor é repelir dez verdades do que admitir uma única falsidade.”

Allan Kardec

Kits Didáticos e Gravadores da Cerne Tecnologia

A Cerne tecnologia têm uma linha completa de aprendizado para os microcontroladores da família PIC, 8051, Holtek, dsPIC, ARM e etc. Veja os detalhes de cada um nas figuras abaixo:



Kit Cerne CPLD

- CPLD ALTERA
- Botões
- Leds
- Gravação ICSP
- Display 7 Segmentos
- E muito mais!

Uma linha completa de componentes para o desenvolvimento de seus projetos eletrônicos como displays, PICs, botões, leds, cristais e etc.

Visite a nossa página na Internet, no endereço **www.cerne-tec.com.br** e conheça melhor nossos serviços e produtos.

Cerne
Conhecimento para o Desenvolvimento

www.cerne-tec.com.br

Sumário

Capítulo I – Metodologia de desenvolvimento.....6

Capítulo II – Exemplos no VHDL.....7

1. Lógica AND com Nibble.....7
2. Lógica OR com Nibble.....8
3. Lógica XOR com Nibble.....9
4. Lógica NAND com Nibble.....11
5. Lógica NOR com Nibble.....12
6. Lógica XNOR com Nibble.....13
7. Lógica AND com Byte.....14
8. Lógica OR com Byte.....16
9. Lógica XOR com Byte.....18
10. Lógica NAND com Byte.....19
11. Lógica NOR com Byte.....21
12. Lógica XNOR com Byte.....22
13. Somador com Nibble.....24
14. Representação complemento 2.....29
15. Subtrator com Nibble.....31
16. Unidade Lógica Aritmética (ULA).....35

Capítulo I
Metodologia de desenvolvimento

A proposta desta obra é continuar apresentando exemplos em VHDL para FPGA. Exemplos com portas lógicas fundamentais fazendo uso de barramento, somador, subtrator e unidade lógica e aritmética (ULA) são temas abordados neste volume.

Para isso, recomenda-se que o leitor leia a obra do volume passado *FPGA Aplicações em VHDL (2014)* do mesmo autor e editora de modo a continuar seus estudos com eficiência.

Capítulo II
Exemplos no VHDL

1. Lógica AND com Nibble

O intuito deste exemplo é fazer uma operação AND (&) entre dois *nibbles* (conjunto de 4 bits) no qual o primeiro *nibble* (c1) é carregado com 2#0101# em binário e o segundo *nibble* (c2) com 2#1100#. Logo em seguida a operação AND é feita e o resultado salvo no *nibble* res e tal valor atribuído ao conjunto de saída representado por 4 leds conectados ao FPGA para representar o resultado da operação. Acompanhe o exemplo:

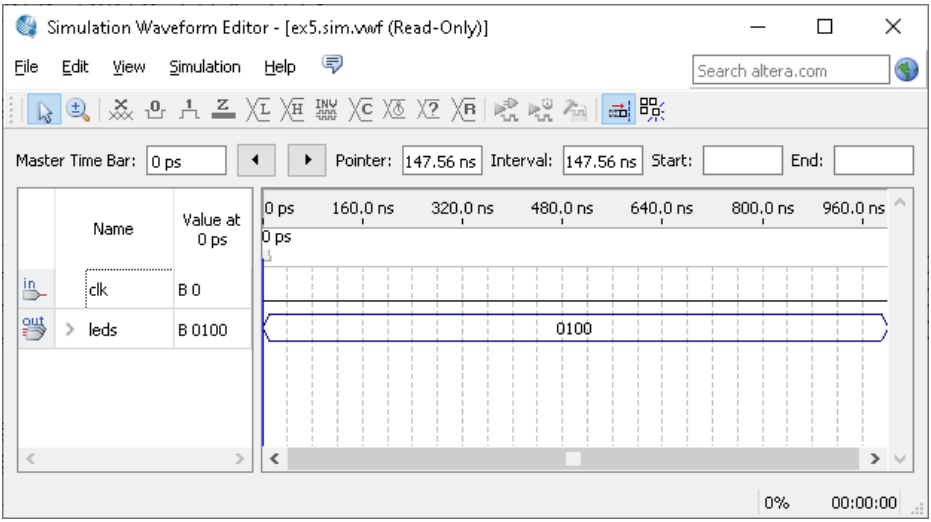
```
library ieee;
use ieee.std_logic_1164.all;

entity ex5 is
	port(clk :in std_logic;
		leds: out STD_LOGIC_VECTOR(3 downto 0));
end ex5;

architecture OperatorTest of ex5 is
	signal c1 :  STD_LOGIC_VECTOR(3 downto 0) := ('0','1','0','1');
	signal c2 :  STD_LOGIC_VECTOR(3 downto 0) := ('1','1','0','0');
	signal res : STD_LOGIC_VECTOR(3 downto 0) := ('0','0','0','0');

begin
	res<=c1 and c2;
	leds<=res;
end OperatorTest;
```

A seguir o resultado da simulação obtendo o valor esperado pela lógica estudada.



2. Lógica OR com Nibble

O intuito deste exemplo é fazer uma operação OR (Ou) entre dois *nibbles* (conjunto de 4 bits) no qual o primeiro *nibble* (c1) é carregado com 2#0101# em binário e o segundo *nibble* (c2) com 2#1100#. Logo em seguida a operação OR é feita e o resultado salvo no *nibble* res e tal valor atribuído ao conjunto de saída representado por 4 leds conectados ao FPGA para representar o resultado da operação. Acompanhe o exemplo:

```
library ieee;
use ieee.std_logic_1164.all;

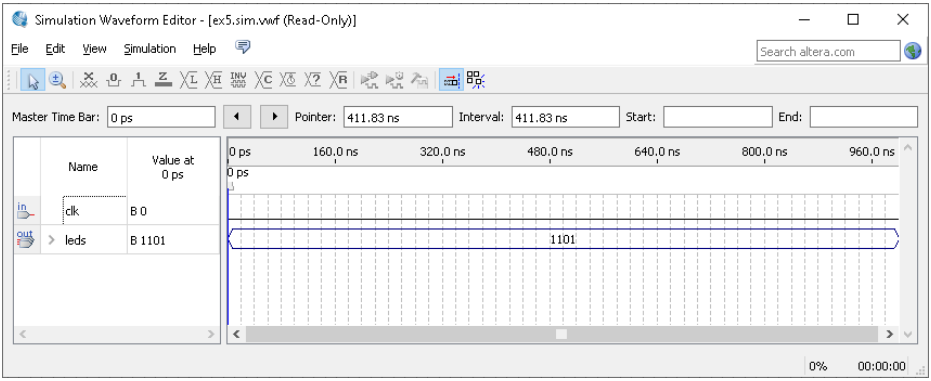
entity ex5 is
        port(clk :in std_logic;
```

```
                leds: out STD_LOGIC_VECTOR(3 downto 0));
end ex5;

architecture OperatorTest of ex5 is
        signal c1 :  STD_LOGIC_VECTOR(3 downto 0) := ('0','1','0','1');
        signal c2 :  STD_LOGIC_VECTOR(3 downto 0) := ('1','1','0','0');
        signal res : STD_LOGIC_VECTOR(3 downto 0) := ('0','0','0','0');

begin
        res<=c1 or c2;
        leds<=res;
end OperatorTest;
```

A seguir o resultado da simulação obtendo o valor esperado pela lógica estudada.



3. Lógica XOR com Nibble

O intuito deste exemplo é fazer uma operação XOR entre dois *nibbles* (conjunto de 4 bits) no qual o primeiro *nibble* (c1) é carregado com 2#0101# em binário e o segundo *nibble* (c2) com 2#1100#. Logo em seguida a operação XOR é feita e o resultado salvo no *nibble* res e

ltal valor atribuído ao conjunto de saída representado por 4 leds conectados ao FPGA para representar o resultado da operação. Acompanhe o exemplo:

```
library ieee;
use ieee.std_logic_1164.all;

entity ex5 is
        port(clk :in std_logic;
                 leds: out STD_LOGIC_VECTOR(3 downto 0));
end ex5;

architecture OperatorTest of ex5 is
        signal c1 :  STD_LOGIC_VECTOR(3 downto 0) := ('0','1','0','1');
        signal c2 :  STD_LOGIC_VECTOR(3 downto 0) := ('1','1','0','0');
        signal res : STD_LOGIC_VECTOR(3 downto 0) := ('0','0','0','0');

begin
        res<=c1 xor c2;
        leds<=res;
end OperatorTest;
```

A seguir o resultado da simulação obtendo o valor esperado pela lógica estudada.

