Vitor Amadeu Souza

FPGA

Aplicações em VHDL
Volume IX

© 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.

Julho 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. Memória RAM 62256.....7
2. Operação de leitura (read cycle).....10
3. Operação de escrita (write cycle).....15

Capítulo I
Metodologia de desenvolvimento

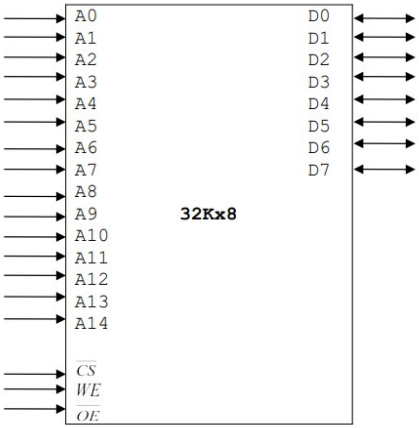
A proposta desta obra é continuar apresentando exemplos em VHDL para FPGA. Exemplos com acesso à memória para escrita e leitura são temas abordados neste volume.

Para isso, recomenda-se que o leitor leia as obras dos volumes passados como *FPGA Aplicações em VHDL (2014), FPGA Aplicações em VHDL Volume II (2023) até o Volume VIII (2023)* do mesmo autor e editora de modo a continuar seus estudos com eficiência.

Capítulo II
Exemplos no VHDL

1. Memória RAM 62256

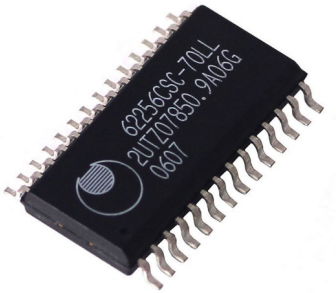
A memória RAM (Random Access Memory – Memória de Acesso Aleatório) é uma memória volátil, ou seja, sua informação é perdida na ausência de energia. É composta por três barramentos como visto anteriormente, neste caso o de endereço, dados e controle. Encontramos estas memórias tipicamente no mercado sob a designação 62XXX, onde XXX indica a capacidade em kbits da memória. Por exemplo, a memória 62256 possui capacidade de 256 kbits ou 32kB. A seguir é apresentado a simbologia típica desta memória. Observe que temos ao todo 15 linhas de endereçamento, que permitem escolher uma das posições dos 32 kB de memória.



A linha $\overline{CS}$ (Chip Select – Seleção de Chip) permite selecionar a memória no barramento de dados. Já a linha WE (Write Enable – Habilitação de Escrita) permite que a operação de escrita na memória seja habilitada e a linha $\overline{OE}$ (Output Enable – Habilitação da Saída) faz com que uma operação de leitura seja efetivada na memória. Na próxima tabela temos as configurações possíveis na memória RAM 62XXX.

Entradas				**Saídas**
Modo	WE	$\overline{CS}$	$\overline{OE}$	D_7-D_0
Não selecionado	X	1	X	Alta impedância
Saída desabilitada	1	0	1	Alta impedância
Leitura	1	0	0	Saída de dados
Escrita	0	0	1	Entrada de dados

A próxima figura mostra os diversos tipos de encapsulamento disponíveis para o 62256.



SOIC-28



TSOP-28



DIP-28

2. Operação de leitura (read cycle)

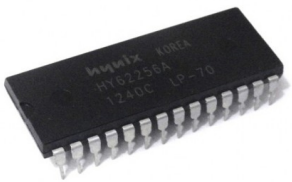
O primeiro passo para conectar uma memória ao FPGA é ligar o barramento de controle, dados e endereço conforme apresentado abaixo.



Endereço: 15 bits

Dados: 8 bits

Controle: 3 bits



São 15 linhas de endereçamento para mapear até 32kB de memória. Há 8 linhas de dados, pois o armazenamento é por bytes e 3 linhas de controle para seleção, escrita ou leitura da memória. No exemplo a seguir o objetivo é ler o endereço 127 da memória.

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

entity ex1 is
port(data : inout std_logic_vector(7 downto 0);
     ce,wr,oe : out std_logic;
     addr : out std_logic_vector(14 downto 0));
end ex1;

architecture read_cycle of ex1 is
signal dados : STD_LOGIC_VECTOR(7 downto 0);
```