



Dispositivos Lógicos Programáveis (DLP)

Ideia e Arquiteturas

Universidade Federal de Uberlândia
Faculdade de Computação
Prof. João Henrique de Souza Pereira

Créditos dos slides para o Prof. Dr. Daniel D. Abdala

Na Aula Anterior

- Mapa de Karnaugh;
- Funções incompletamente especificadas;
- Simplificação de funções incompletamente especificadas.

Nesta Aula

- DLPs – Ideia Geral
- Benefícios da Utilização de DLPs;
- Funcionamento geral de DLPs;
- Visão geral FPGAs;
- Introdução ao VHDL.

DLP	Dispositivo Lógico Programável
FPGA	Field Programmable Gate Array
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuits

Nesta Aula

- DLPs – Ideia Geral
- Benefícios da Utilização de DLPs;
- Funcionamento geral de DLPs;
- Visão geral FPGAs;
- Introdução ao VHDL.

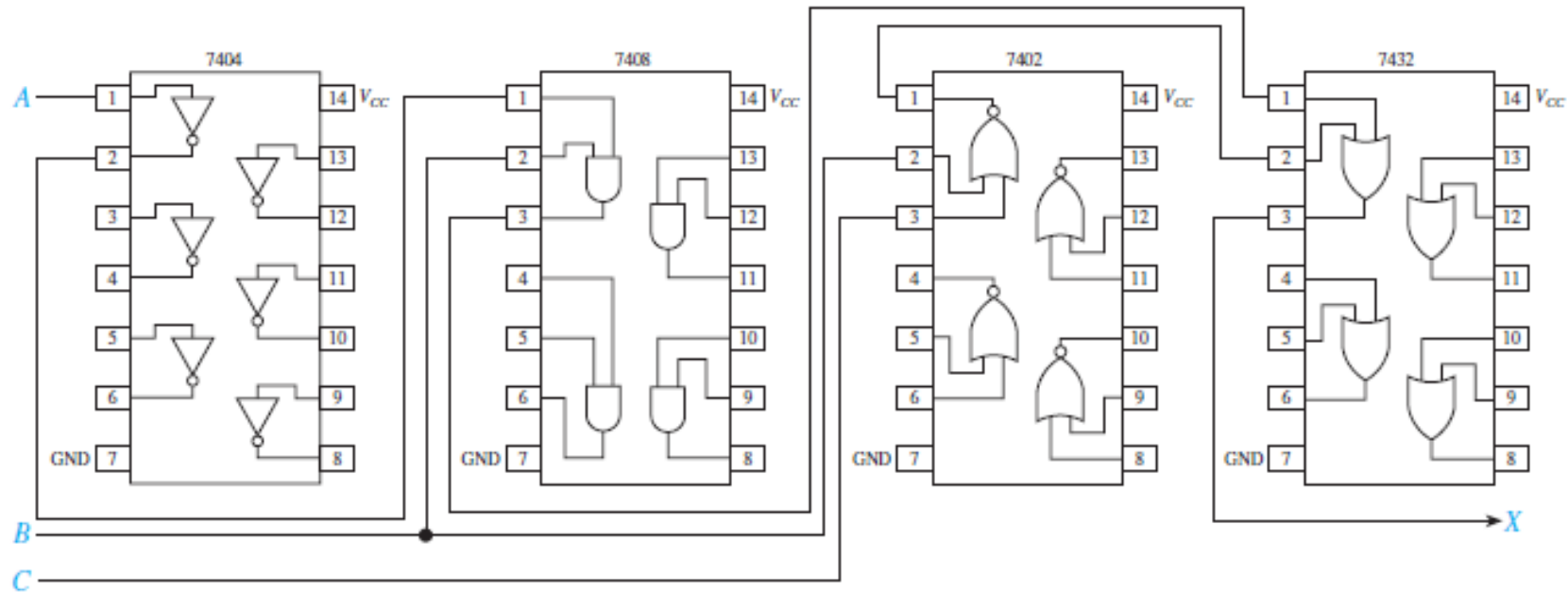
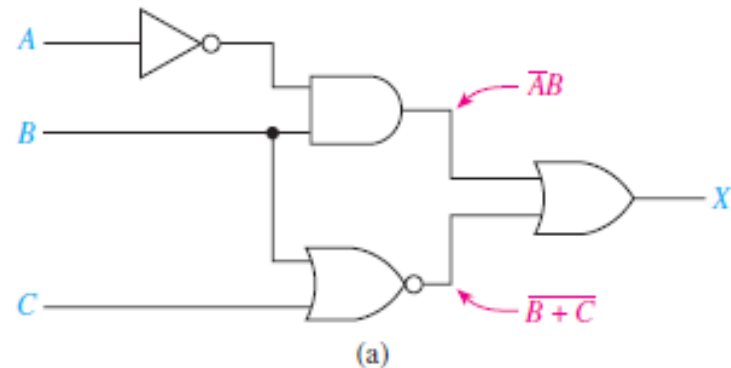
DLP	Dispositivo Lógico Programável
FPGA	Field Programmable Gate Array
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuits

IMPORTANTE: Este conteúdo é adicional à Ficha de Disciplina GSI008. Será ministrado para conhecimento complementar. Por ser adicional, não será cobrado em avaliação.

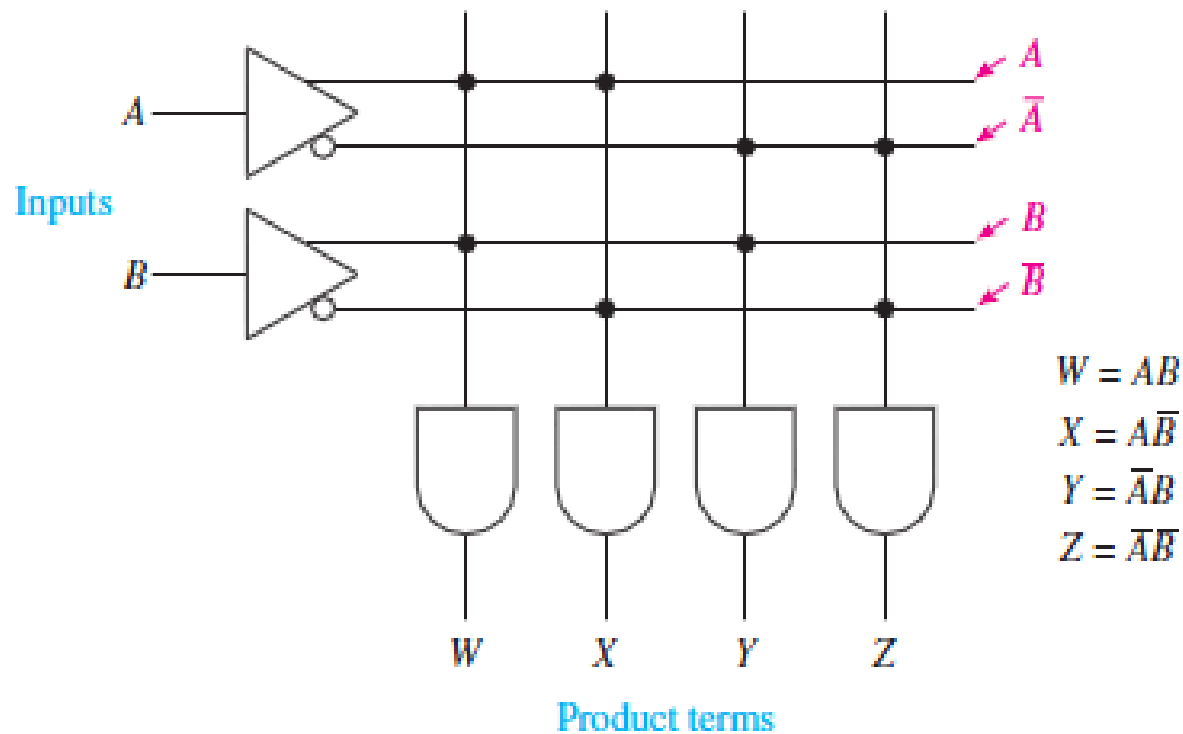
Introdução

- Projeto digital demanda diferentes componentes;
- Manter em estoque todos os componentes potencialmente necessários pode ser custoso;
- Projeto digital usando as famílias 7400 e 4000 requerem muito espaço em circuitos impressos;
- Muitas vezes apenas uma ou duas portas são utilizadas em um CI;
- Solução: CIs reconfiguráveis (programáveis) pelo usuário para executar as operações específicas requeridas;
- Estes CIs são chamados DLPs – Dispositivos Lógicos programáveis.

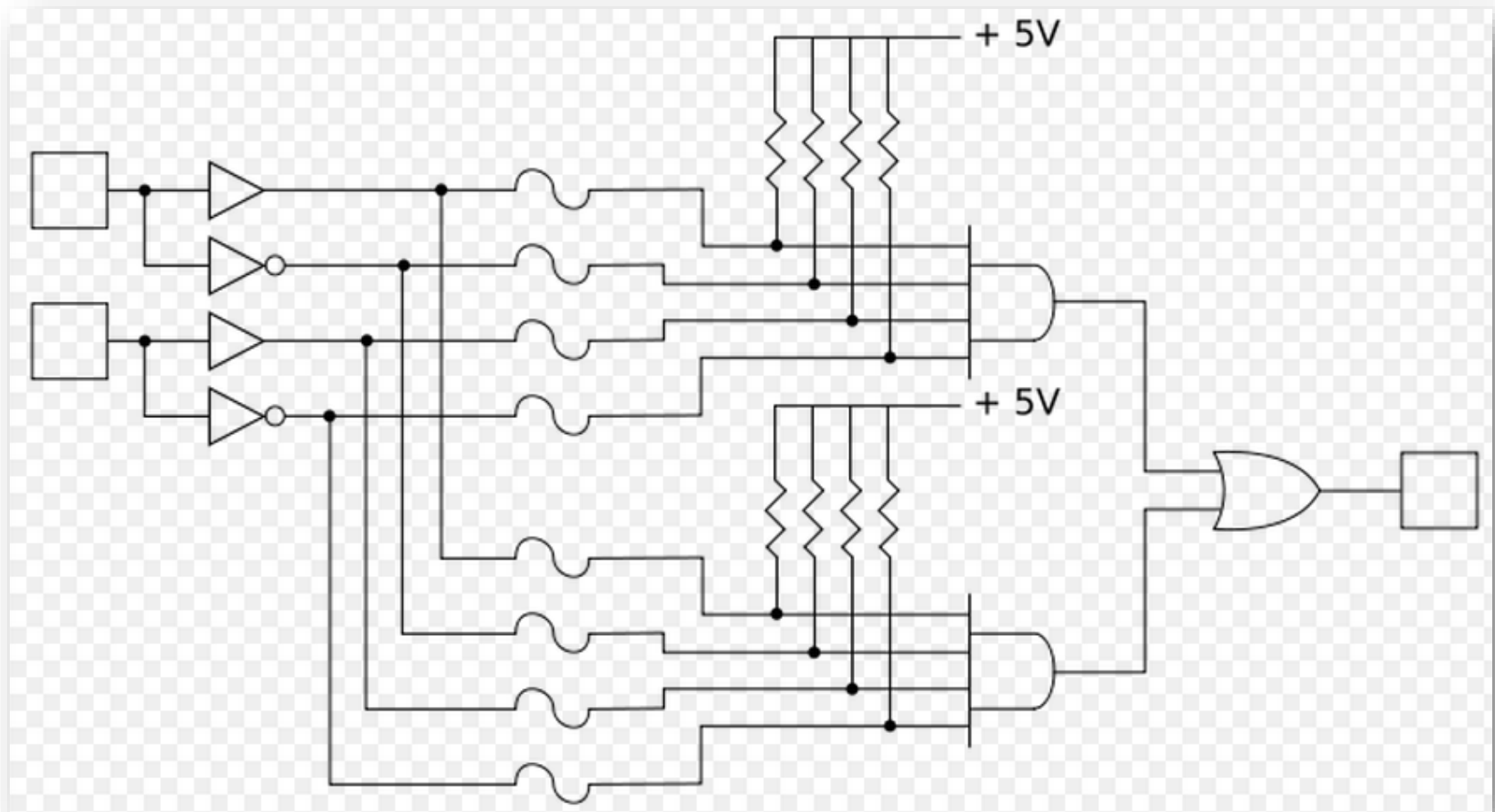
Projeto de SDs usando CIs Individuais



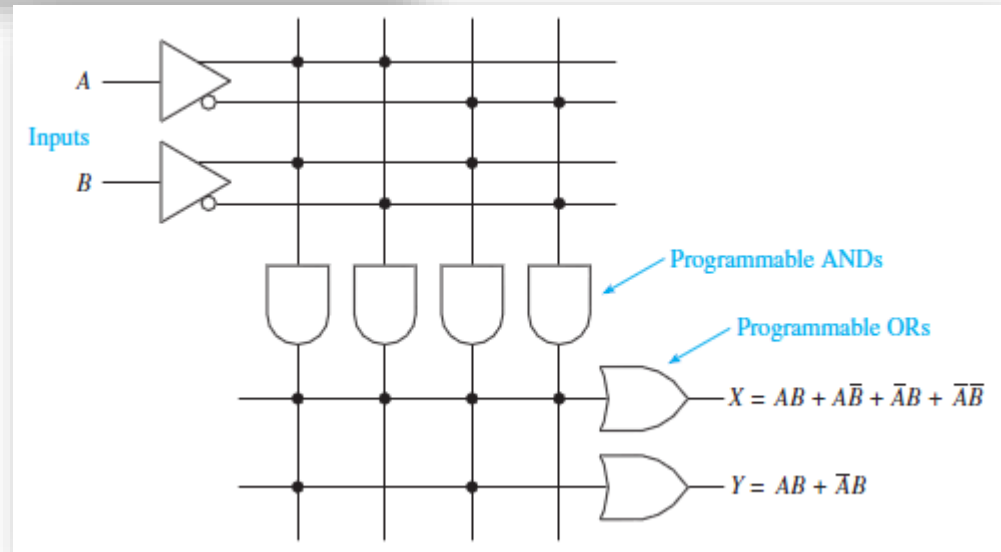
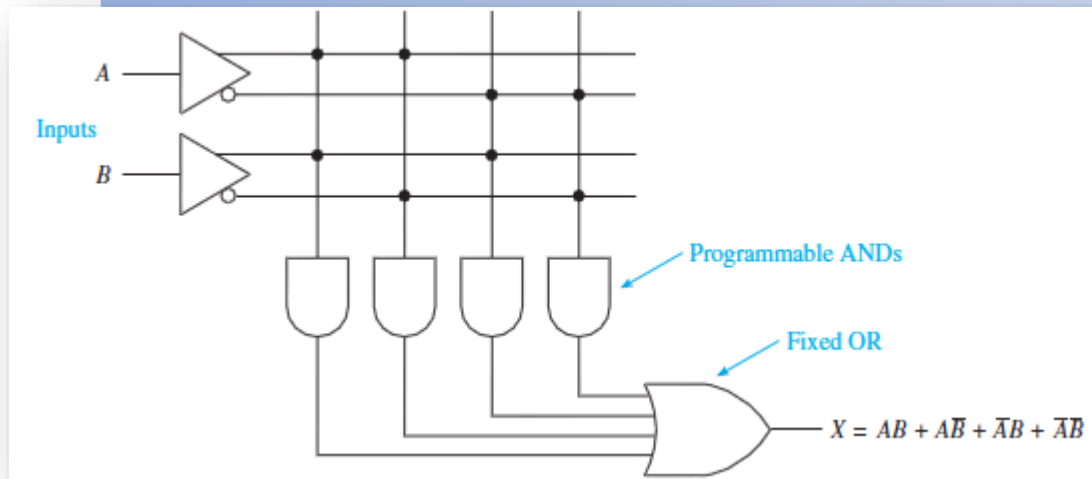
Exemplo DLPs



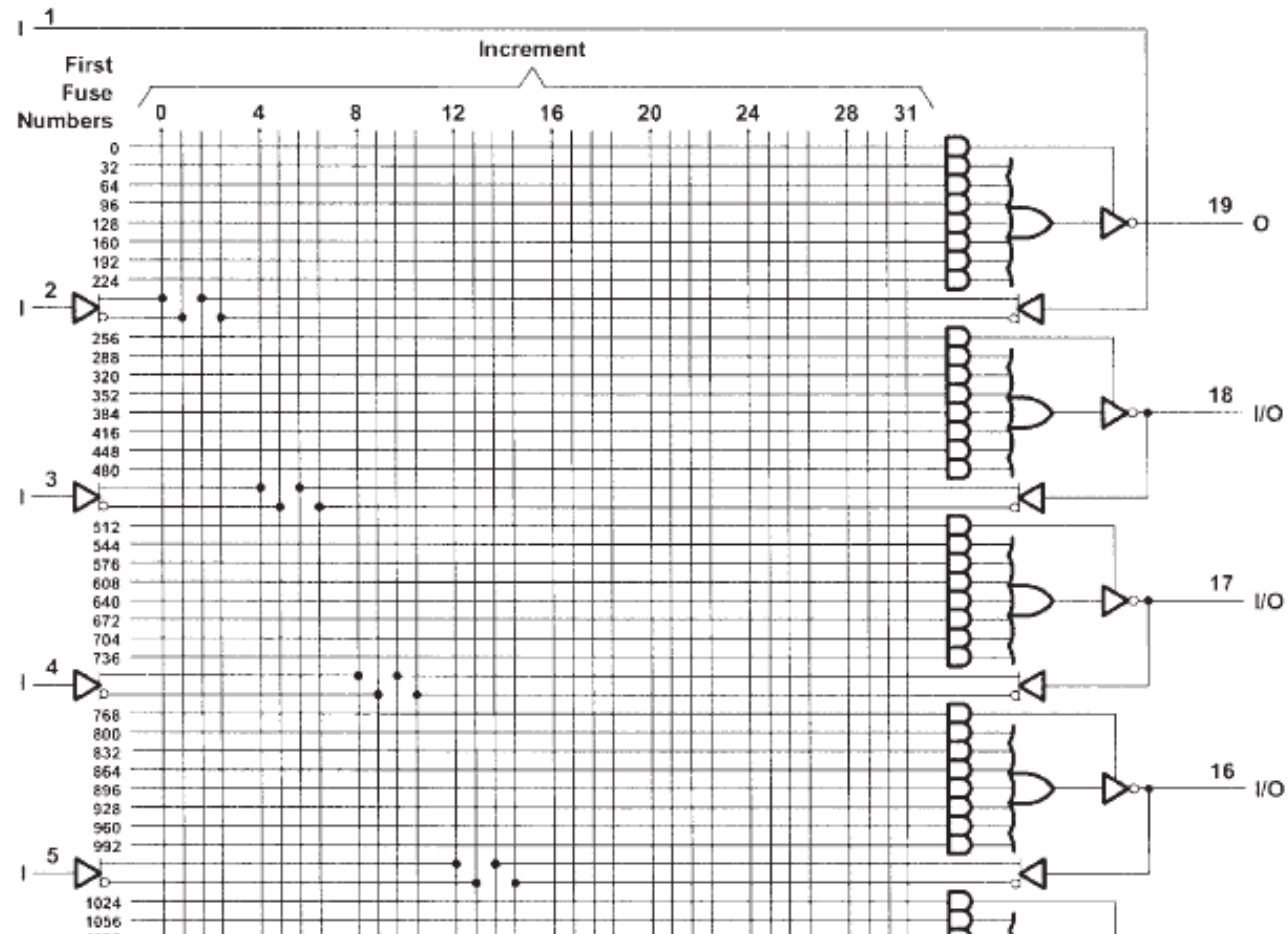
Programação: Fusíveis



Exemplo DLPs



logic diagram (positive logic)

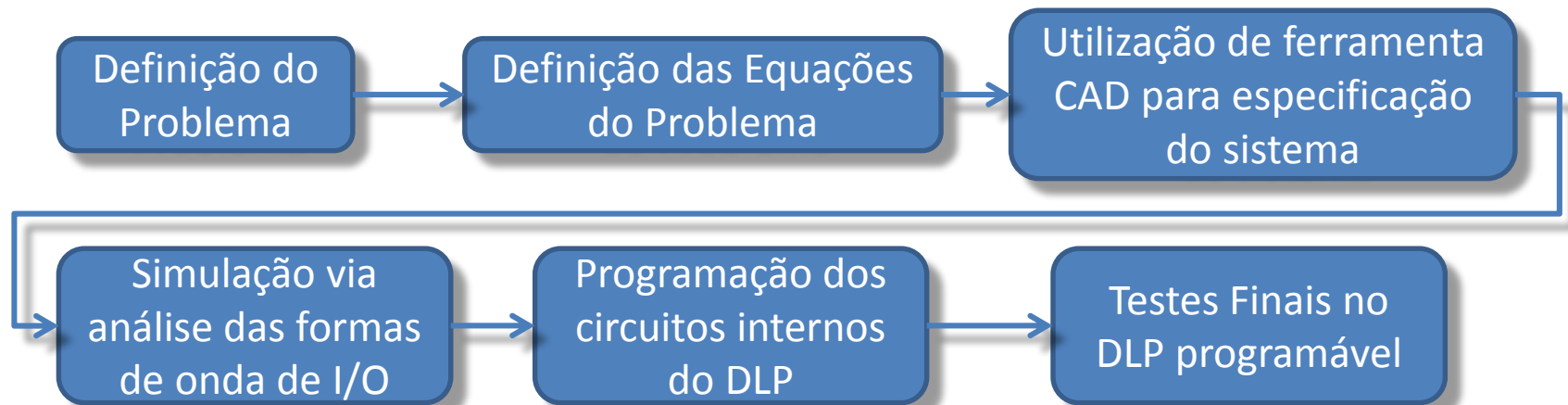


DLPs – Dispositivos Lógicos Programáveis

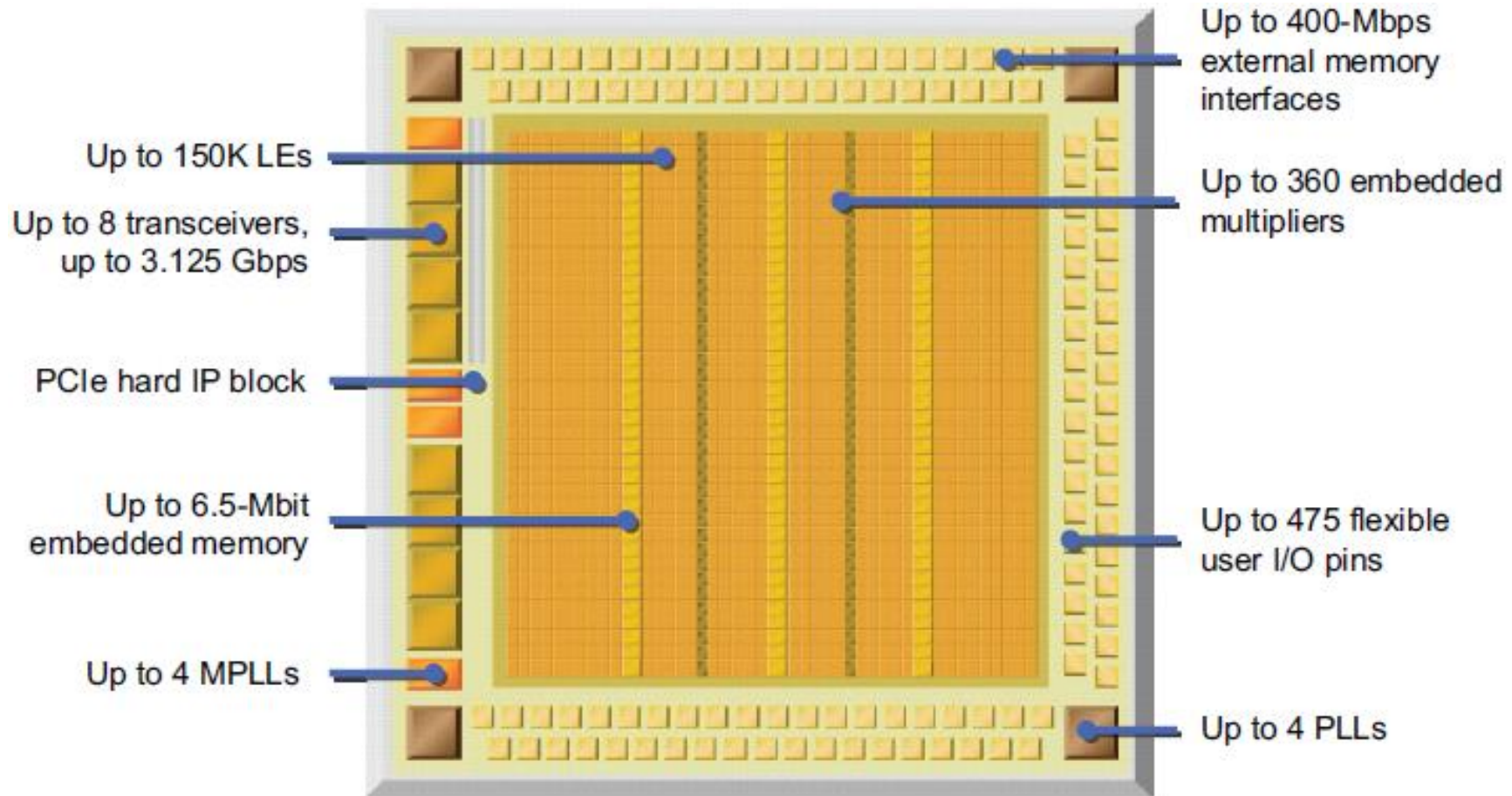
- Milhares de unidades lógicas;
- Funções lógicas sequenciais;
- Programação/Especificação via ferramentas CAD;
- Possibilidade de testar o sistema antes de efetivamente programá-lo.



Fluxo de Projeto em DLPs



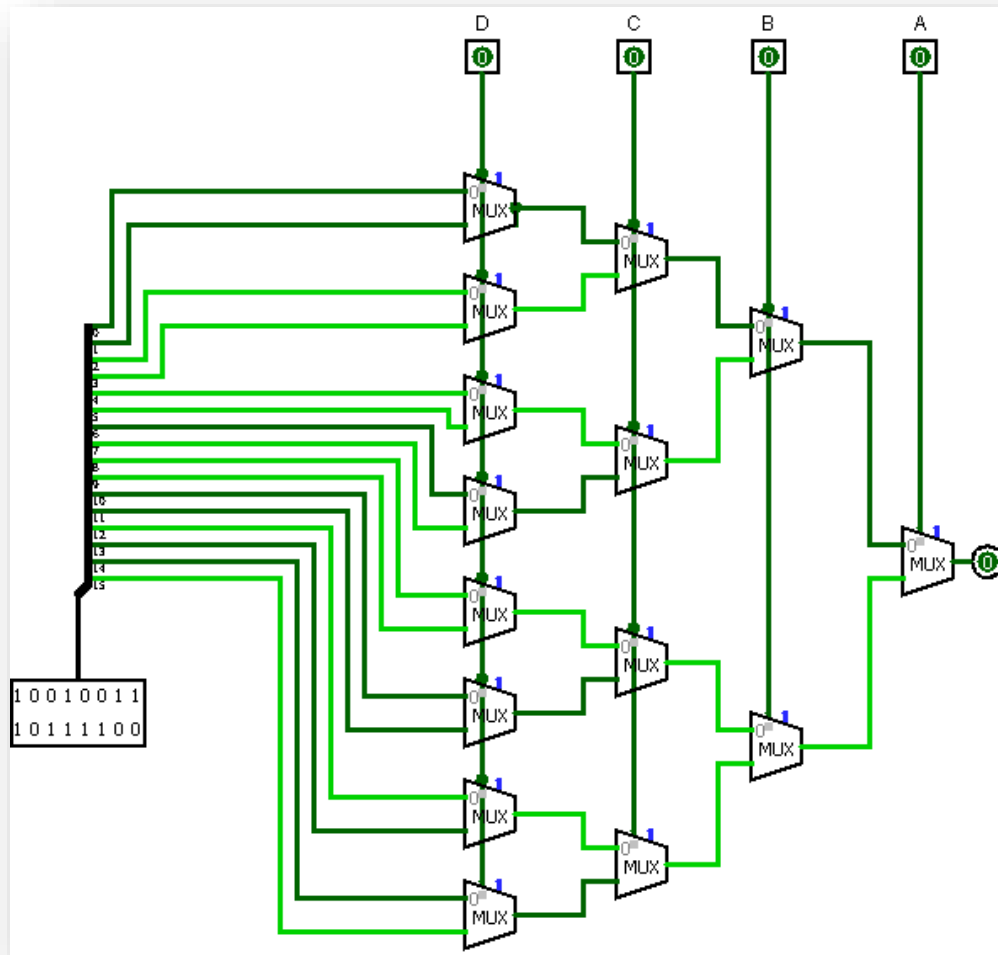
Visão Interna FPGA



FPGA – Field-Programmable Gate Arrays

- Array de portas interconectadas em uma matriz linha x coluna no campo a partir de um computador via conexão USB;
- Utiliza uma LUT – Look-up table;
- Possui diversas vantagens se comparado a outras tecnologias DLPs;
- Ex:
 - Altera Cyclone IV, Xilinx, etc...

Exemplo LUT/FPGA



Kits de Desenvolvimento FPGA



Introdução a VHDL

- Linguagem de especificação/descrição de hardware;
- Não é linguagem de programação;
- Existem outras linguagens (e.g. AHDL, Verilog);
- Comportamento paralelo;
- Comportamento sequencial (processos).

Estrutura de uma Descrição em VHDL

```
<libs to be imported>
ENTITY <ID> IS
    PORT ( <var_in> : IN BIT;
          <var_out>: OUT BIT) ;
END <ID>;
ARCHITECTURE <IDA> OF <ID> IS
BEGIN
    <architecture implementation>
END <IDA>;
```

Entity

- Descreve a interface (entradas e saídas);

```
ENTITY e_comb IS
    PORT ( A, B, : IN BIT;
           S, : OUT BIT) ;
END e_comb;
```

Architecture

- Descreve o funcionamento, especificação interna de uma entidade;

```
ARCHITECTURE dataflow OF e_comb IS
    SIGNAL temp,temp2 : Bit;
BEGIN
    temp    <= A and not B;
    temp2   <= not A and B;
    S       <= temp or temp2;
END dataflow;
```

Bibliotecas

- VHDL permite que sistemas já criados sejam importados;
- Também permite a importação de bibliotecas para execução de diversos outros recursos;
- Ex:

library ieee;

library ieee.std_logic_1164.all;

library ieee.numeric_std.all;

Tipos de Dados

- STD_LOGIC;
- ARRAY;
- INTEGER;
- REAL;
- RECORD;
- NATURAL;
- UNSIGNED;
- ETC...

Processos

- Parte da arquitetura que é executada apenas quando um ou mais sinais específicos sofrem alteração de valor:

```
LIBRARY IEEE; USE IEEE.std_logic_1164.ALL;
ENTITY nand2 IS
    PORT( a, b : IN std_logic;
          c : OUT std_logic);
END nand2;
ARCHITECTURE nand2 OF nand2 IS
BEGIN
    PROCESS( a, b )
        VARIABLE temp : std_logic;
    BEGIN
        temp := NOT (a and b);
        IF (temp = '1') THEN
            c <= temp AFTER 6 ns;
        ELSIF (temp = '0') THEN
            c <= temp AFTER 5 ns;
        ELSE
            c <= temp AFTER 6 ns;
        END IF;
    END PROCESS;
END nand2;
```

Bibliografia Comentada

- TOCCI, R. J., WIDMER, N. S., MOSS, G. L. **Sistemas Digitais – Princípios e Aplicações.** 11ª Ed. Pearson Prentice Hall, São Paulo, S.P., 2011, Brasil.
- CAPUANO, F. G., IDOETA, I. V. **Elementos de Eletrônica Digital.** 40ª Ed. Editora Érica. São Paulo. S.P. 2008. Brasil.