FIAP

Sistemas Embarcados - SE

Prof. MSc. Fabio H. Pimentel

Linguagem VHDL

Parte 1

Projetos de Sistemas Embarcados

Linguagem VHDL

Very High Speed ASIC Description Language

Linguagem para descrição de hardware

Forma estruturada para a descrição de circuitos digitais

O circuito eletrônico é descrito por sentenças

Possibilita ao circuito ser simulado e sintetizado, isto é,

transformado em portas lógicas

Projetos de Sistemas Embarcados

Linguagem VHDL - Histórico

Very High Speed ASIC Description Language

1981: elaborada pelo Departamento de Defesa (EUA)
época da crise do ciclo de vida dos projetos eletrônicos

1983-1985: linguagem básica concluída pelas empresas
Intermetrics, IBM e Texas Instruments

1986: direitos transmitidos ao IEEE

1987: publicação do padrão IEEE VHDL 87

Várias revisões foram realizadas ao longo desses anos.

Última: IEEE STD 1076-2008

Projetos de Sistemas Embarcados

Linguagem VHDL

Very High Speed ASIC Description Language

Vantagens

VHDL é sinônimo de RAPIDEZ e PRODUTIVIDADE.

O mesmo código pode ser usado em diversas tecnologias.

Portabilidade e longevidade para um projeto.

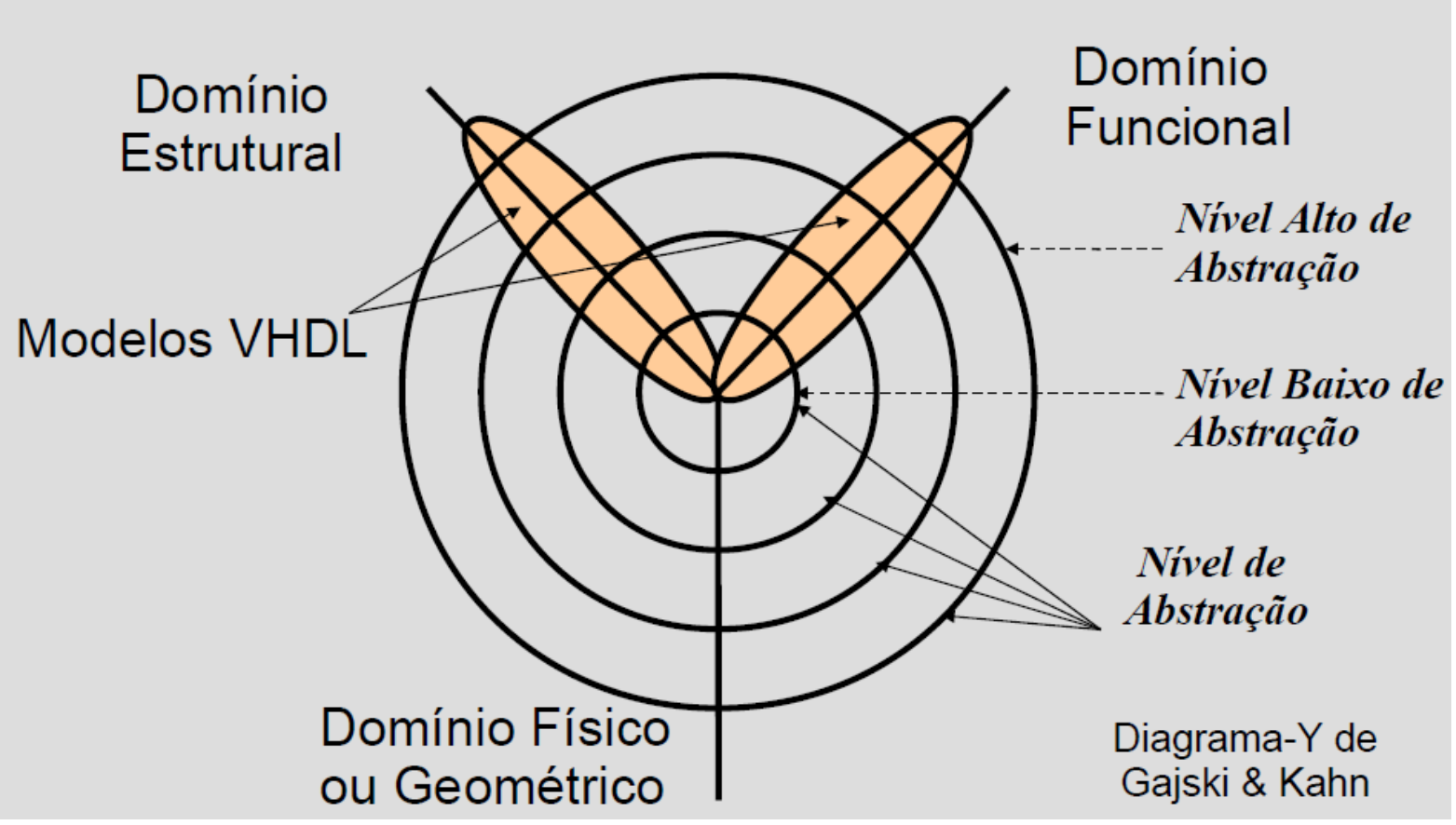
Diversos níveis de teste no código = confiabilidade nos resultados.

Projetos de Sistemas Embarcados

Linguagem VHDL

Very High Speed ASIC Description Language

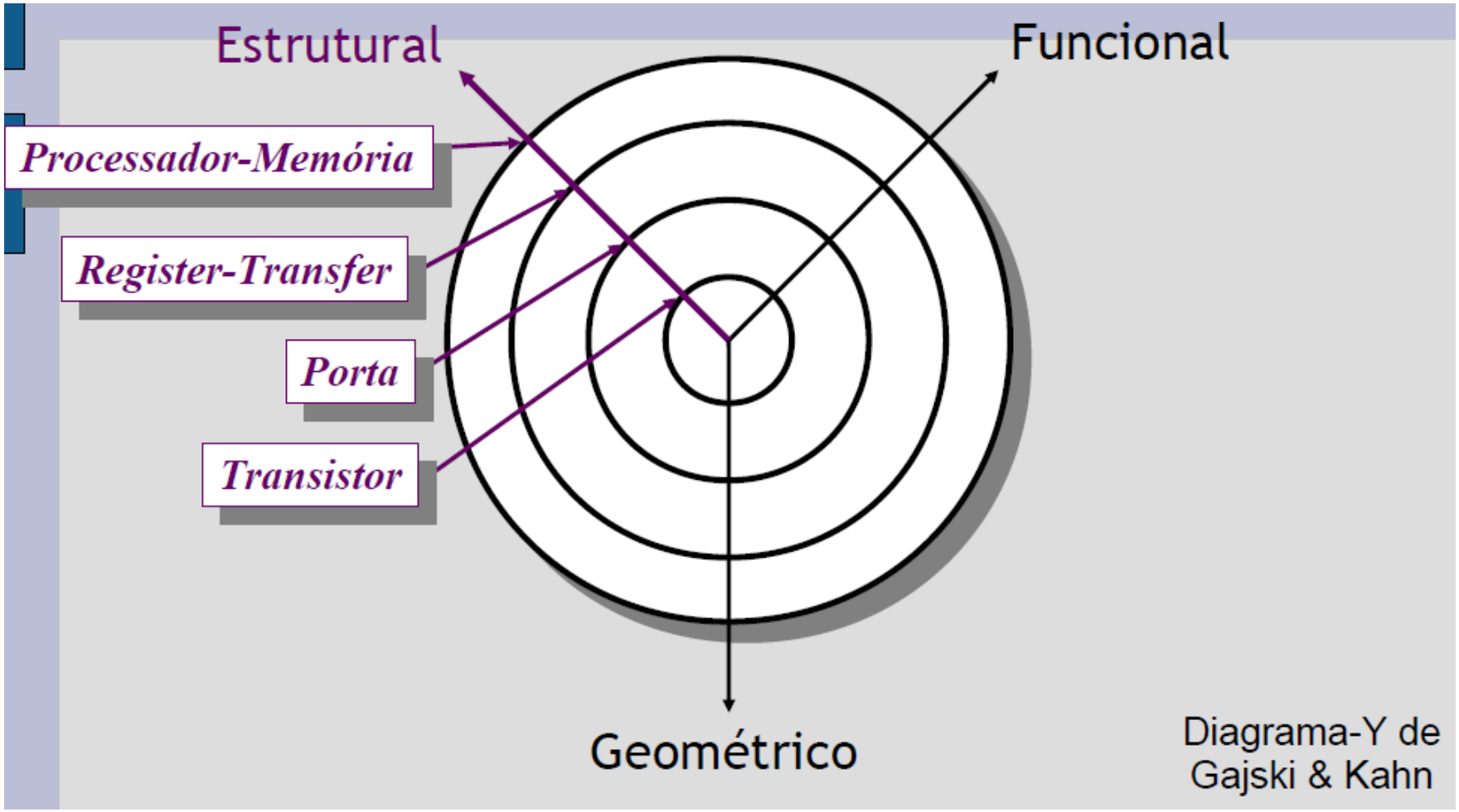
Permite descrição do circuito funcionalmente e estruturalmente.



Projetos de Sistemas Embarcados

Linguagem VHDL

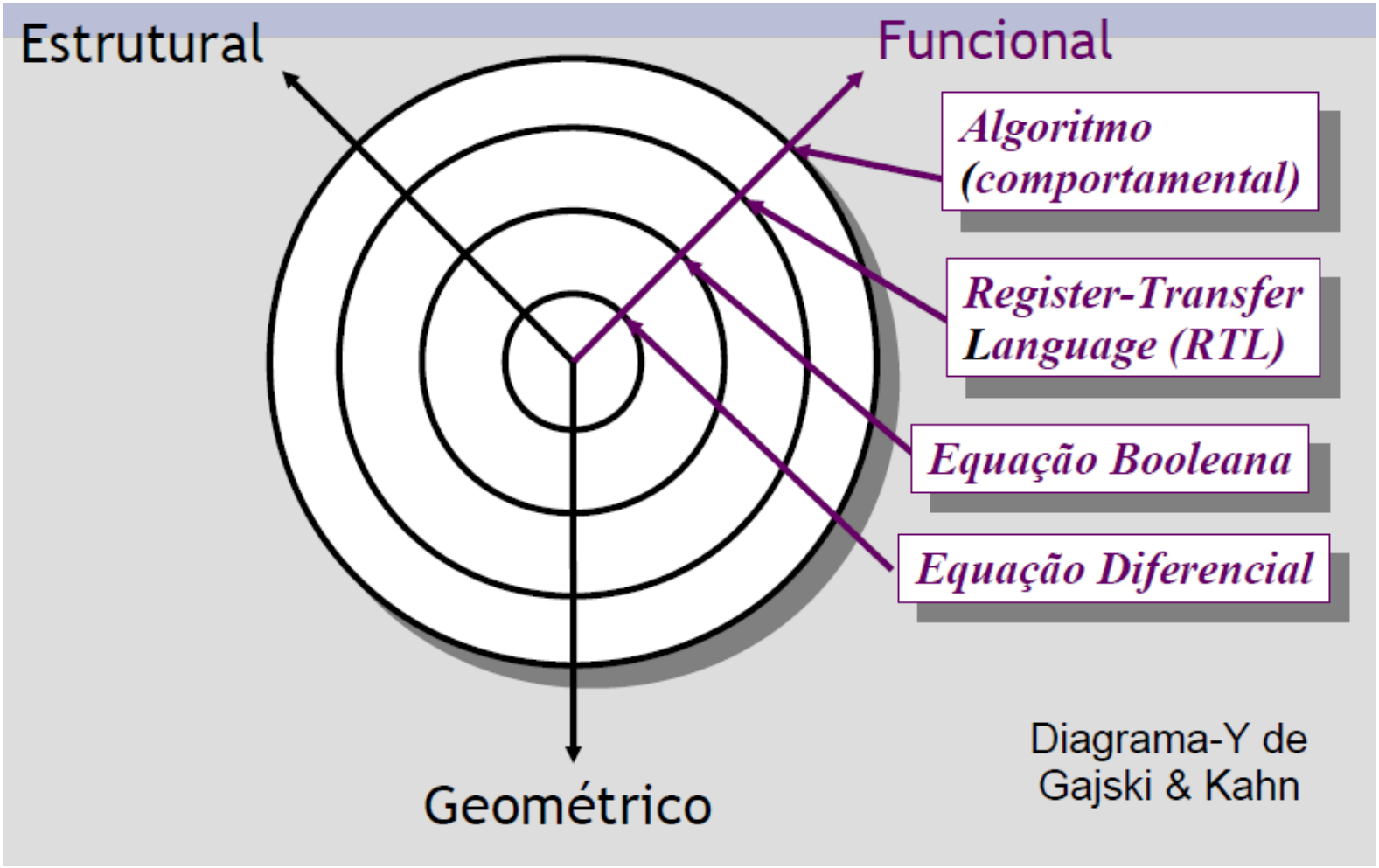
Very High Speed ASIC Description Language



Projetos de Sistemas Embarcados

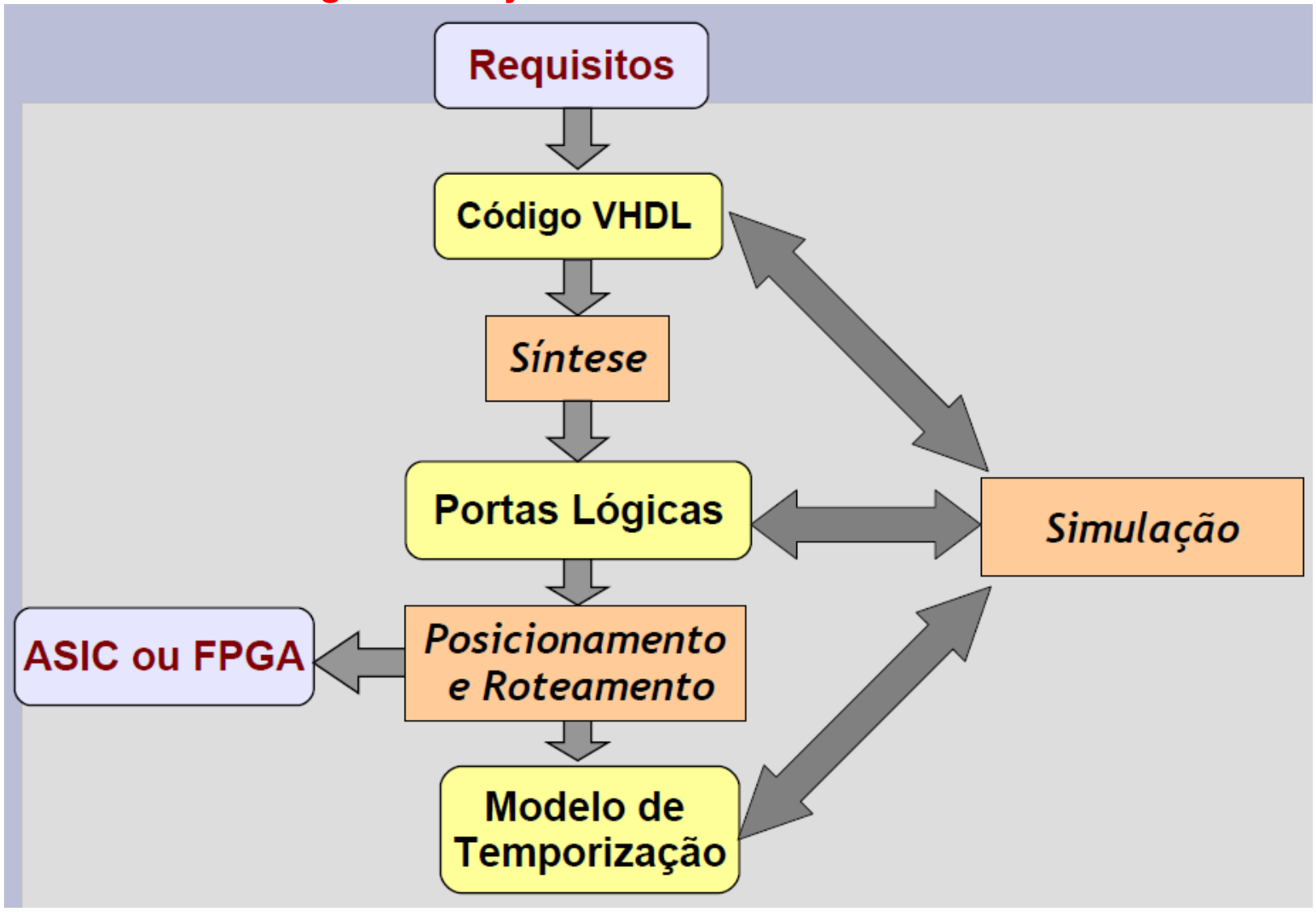
Linguagem VHDL

Very High Speed ASIC Description Language



Projetos de Sistemas Embarcados

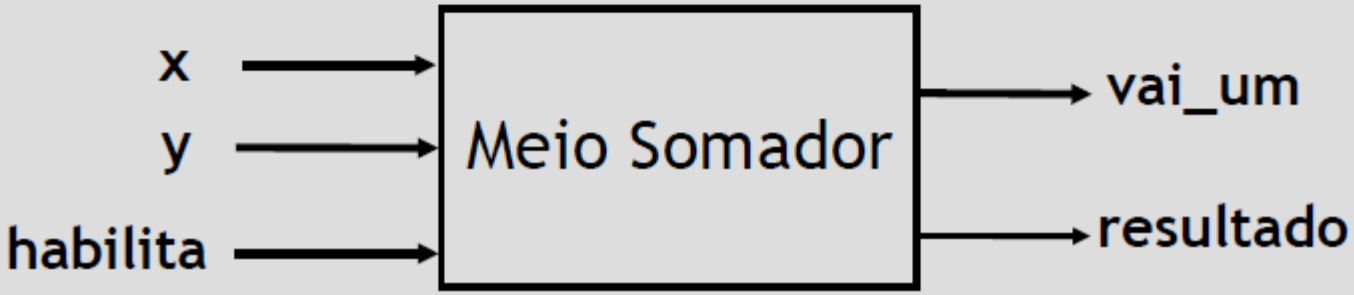
Linguagem VHDL - Metodologia de Projeto



Projetos de Sistemas Embarcados

Linguagem VHDL – Exemplo Processo

* Problema:
  Projetar um meio somador de um bit com vai_um e habilita.
* Especificações:
  - Passa o resultado apenas se habilita for igual a '1'.
  - Resultado é zero se habilita for igual a '0'.
  - Resultado recebe x + y
  - Vai_um recebe o vai_um, se houver, de x + y



Projetos de Sistemas Embarcados

Linguagem VHDL – Exemplo Projeto Comportamental

* **Iniciando com um algoritmo, uma descrição de alto nível do somador é criada:**

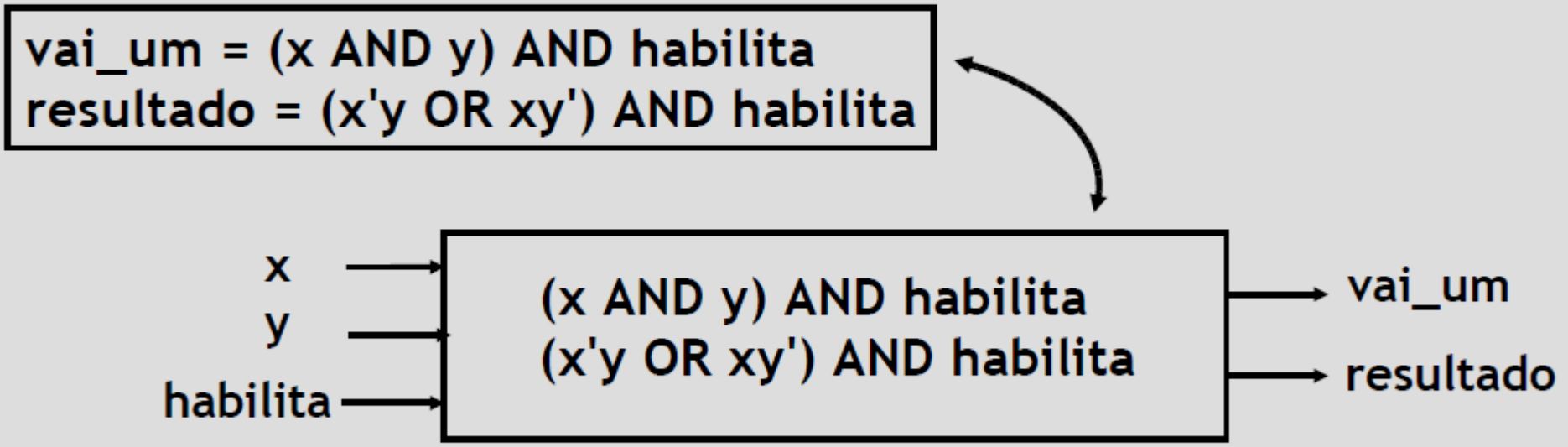
```
IF habilita = 1 THEN
        resultado = x XOR y
        vai_um = x AND y
ELSE
        vai_um = 0
        resultado = 0
```

* **O modelo pode ser agora simulado nesse nível de descrição para verificar o correto entendimento do problema.**

Projetos de Sistemas Embarcados

Linguagem VHDL – Exemplo Projeto Fluxo de Dados

* Com a descrição de alto nível confirmada, equações lógicas descrevendo o fluxo de dados são então criadas.

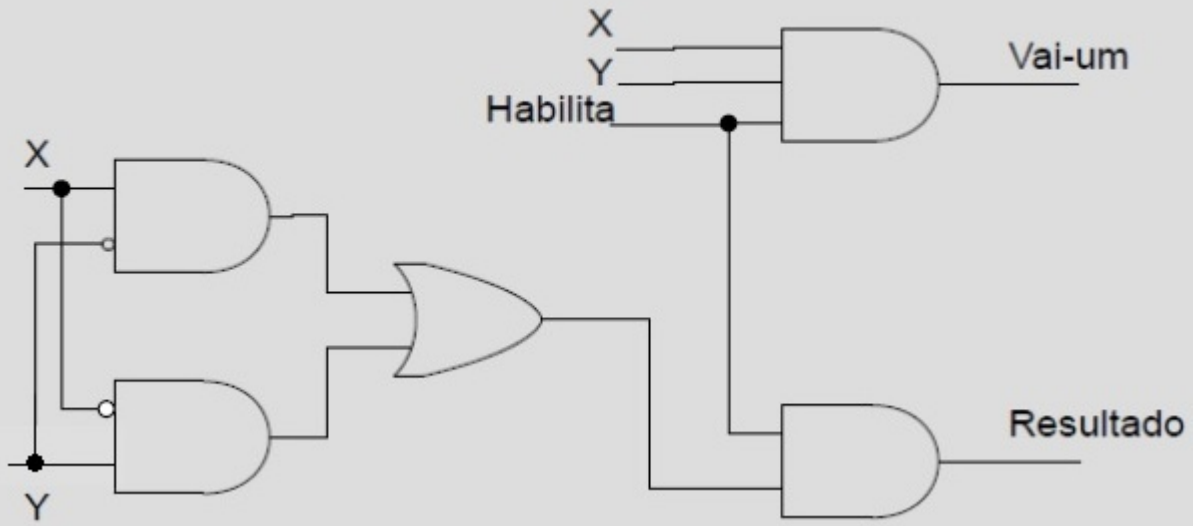


* Novamente, o modelo pode ser simulado neste nível para confirmar as equações lógicas.

Projetos de Sistemas Embarcados

Linguagem VHDL – Exemplo Projeto Lógico

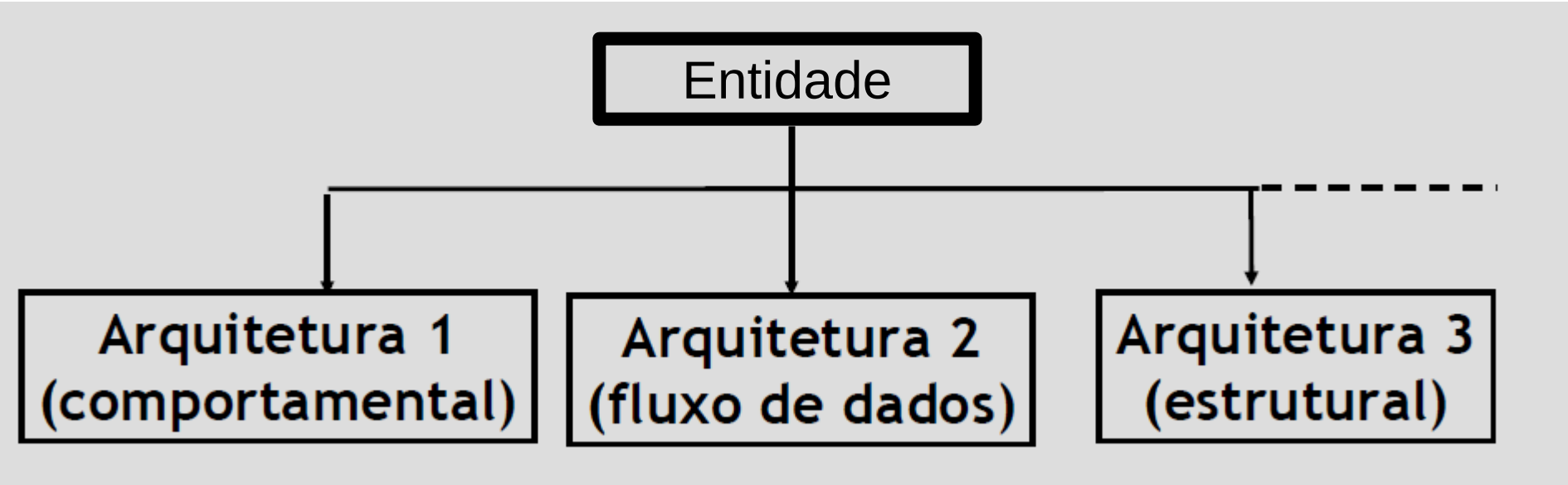
* Finalmente, uma descrição estruturada é criada no nível de portas.



* Essas portas podem ser obtidas de uma biblioteca de componentes.

Projetos de Sistemas Embarcados

Linguagem VHDL – Processo de Projeto VHDL



Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Entidade

* Uma declaração de entidade (ENTITY) descreve a interface do componente.

* Uma cláusula PORT indica as portas de entrada e saída.

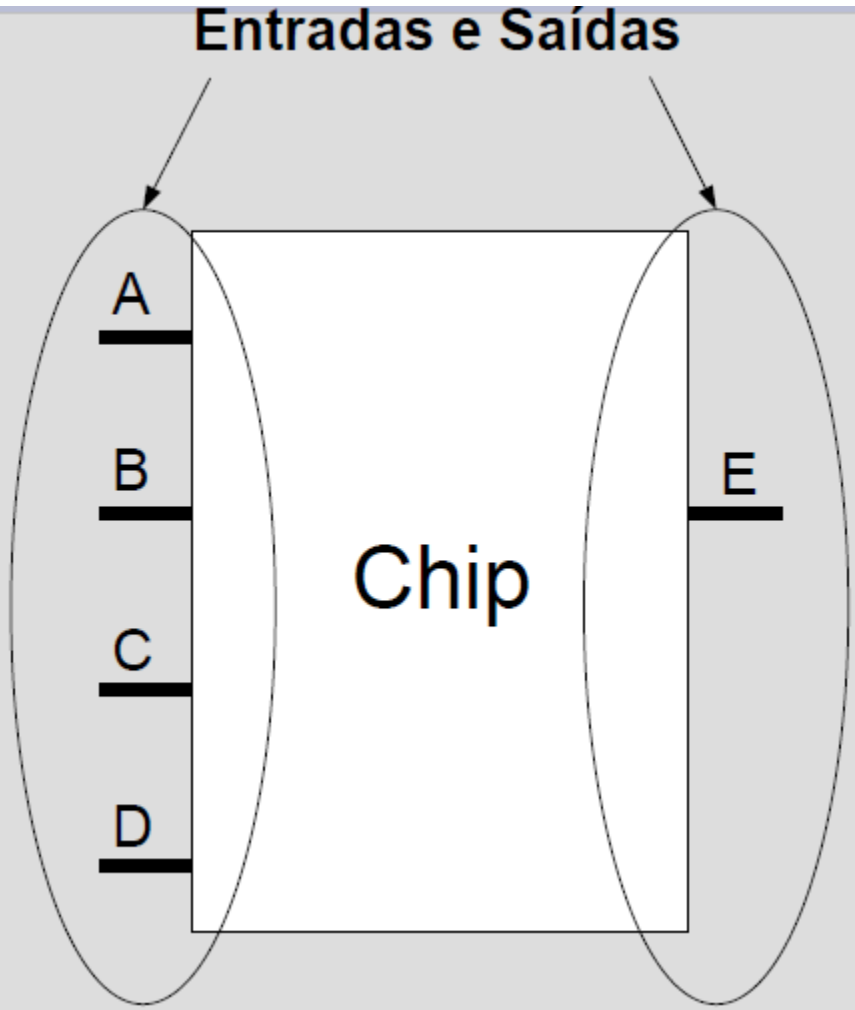
* Uma entidade pode ser pensada como um símbolo para um componente.

Projetos de Sistemas Embarcados

Linguagem VHDL – Entidade

- Define entradas e saídas
- Exemplo:

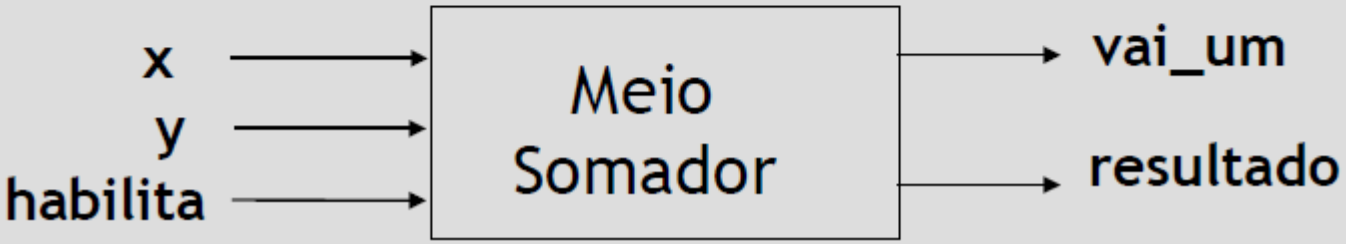
```
Entity test is
Port ( A,B,C,D: in std_logic;
       E: out std_logic);
End test;
```



Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Entidade

```
ENTITY meio_somador IS
    PORT (x, y, habilita: IN bit;
          vai_um, resultado: OUT bit);
END meio_somador;
```



Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta

* Uma declaração de porta (PORT) estabelece a interface entre o componente e o mundo externo

* Há três partes na declaração PORT
  - Nome
  - Modo
  - Tipos de Dados

```
ENTITY test  IS
        PORT (<nome> : <modo>  <tipos_dados>);
END test;
```

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta - Nome

* Apenas letras, dígitos e sublinhados podem ser usados;
* O primeiro caractere deve ser uma letra;
* O último caractere não pode ser um sublinhado;
* Não são permitidos dois sublinhados consecutivos.

Nomes Legais	Nomes Ilegais
rs_clk	_rs_clk
ab08B	sinal#1
A_1023	A__1023
	rs_clk_

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta - Nome

- Não é sensível à "Caixa Alta ou Baixa"
  - inputa, INPUTA e InputA se referem à mesma variável.
- Comentários
  - '--' marca um comentário até o final da linha atual
  - Se você deseja comentar múltiplas linhas, um '--' precisa ser colocado no início de cada linha.
- As sentenças são terminadas por ';'
- Atribuição de valores aos sinais: '<='
- Atribuição de valores às variáveis: ':='

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta - Modo

* O modo da porta de interface descreve o sentido do fluxo de dados tomando como referência o componente.
* Os cinco tipos de fluxo de dados são:
  - IN: os dados entram nesta porta e podem apenas ser lidos (é o padrão).
  - OUT: os dados saem por essa porta e podem apenas serem escritos.
  - BUFFER: similar a Out, mas permite realimentação interna.
  - INOUT: o fluxo de dados pode ser em qualquer sentido, com qualquer número de fontes permitidas (barramento)
  - LINKAGE: o sentido do fluxo de dados é deconhecido

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Tipos de Dados

* Os tipos de dados que passam através de uma porta devem ser especificados para completar a interface.

* Os dados podem ser de diferentes tipos, dependendo do pacote e bibliotecas utilizados.

* Alguns tipos de dados definidos no padrão IEEE são:

  - Bit, Bit_vector
  - Boolean
  - Integer
  - std_ulogic, std_logic

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Tipos de Dados

- **bit values: '0', '1'**
- **boolean** values: TRUE, FALSE
- **integer** values: -(231) to +(231 – 1)
- **std_logic** values: 'U','X','1','0','Z','W','H','L','-'
  - U' = uninitialized
  - 'X' = unknown
  - 'W' = weak 'X'
  - 'Z' = floating
  - 'H'/'L' = weak '1'/'0'
  - '-' = don't care
- **std_logic_vector** (n downto 0);
- **std_logic_vector** (0 upto n);

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Tipos de Dados - Arquitetura

* Declarações do tipo Architecture descrevem a operação do componente.

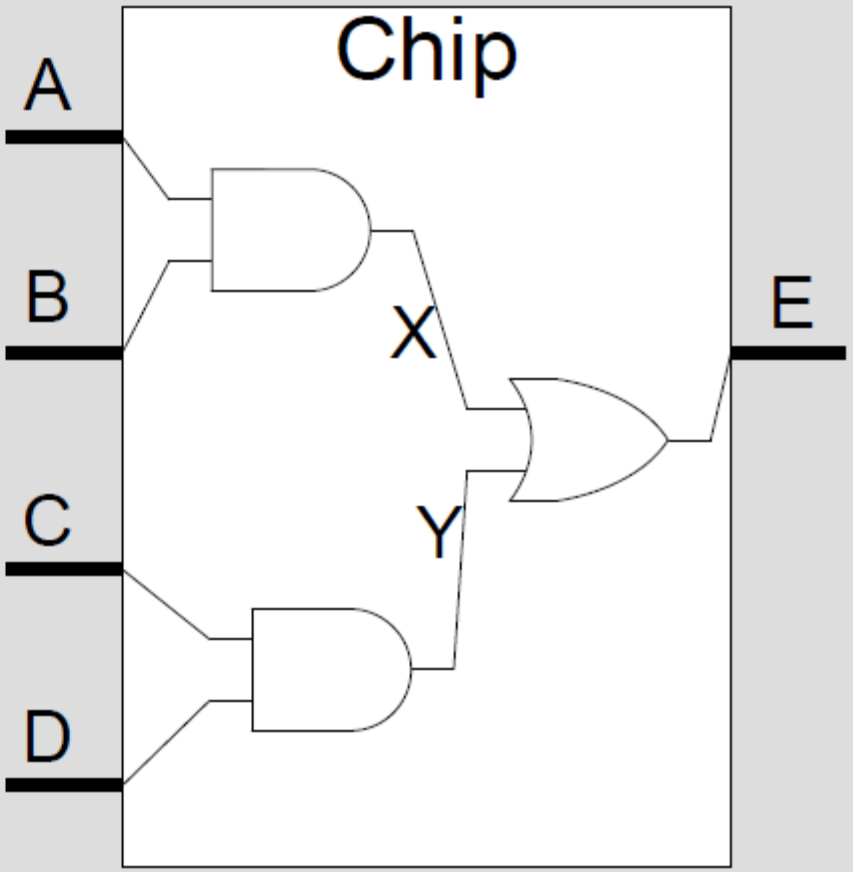
* Muitas arquiteturas podem existir para uma mesma entidade, mas apenas pode haver uma delas ativa por vez.

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Tipos de Dados - Arquitetura

- **Define a funcionalidade do circuito**

```
X <= A AND B;
Y <= C AND D;
E <=  X OR Y;
```



Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Tipos de Dados – Arquitetura Body 1

```
ARCHITECTURE behavior1 OF meio_somador IS
BEGIN
        PROCESS (habilita, x, y)
        BEGIN
            IF (habilita = '1') THEN
                resultado <= x XOR y;
                vai_um <= x AND y;
            ELSE
                vai_um <= '0';
                resultado <= '0';
            END IF;
        END PROCESS;
END behavior1;
```

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Tipos de Dados – Arquitetura Body 2

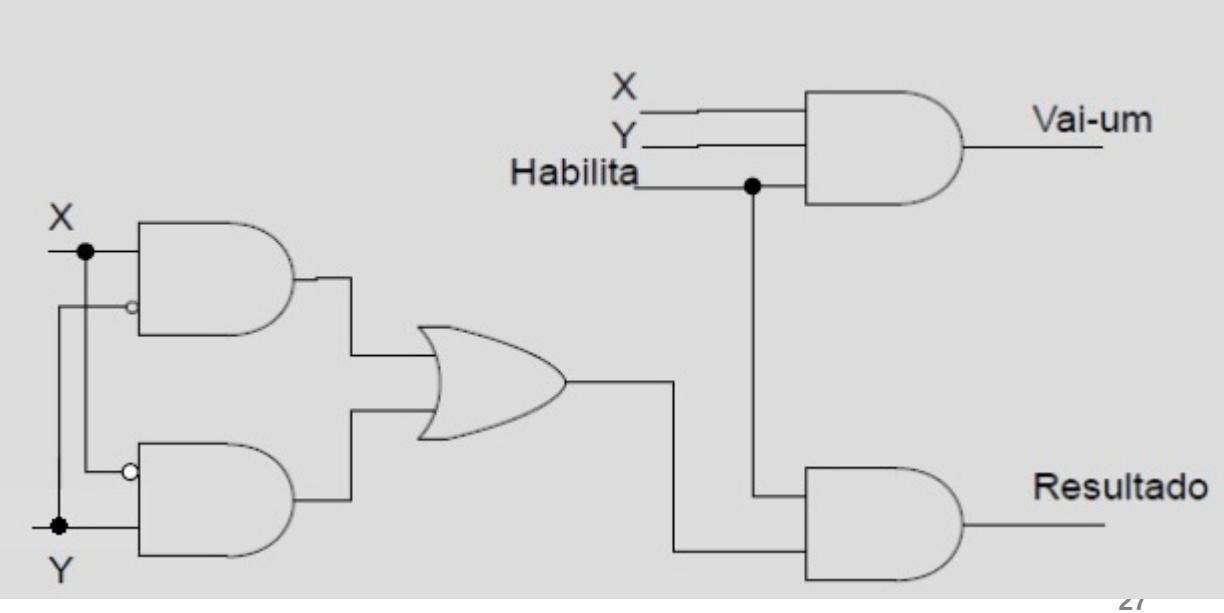
```
ARCHITECTURE data_flow OF meio_somador IS
BEGIN
        vai_um <= (x AND y) AND habilita;
        resultado <= (x XOR y) AND habilita;
END data_flow;
```

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Tipos de Dados – Arquitetura Body 3

* **Para fazer a arquitetura estrutural, nós precisamos primeiro definir as portas a serem utilizadas.**

* **No exemplo a seguir, nós precisamos definir as portas NOT, AND, e OR.**



Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Tipos de Dados – Arquitetura Body 3

```
ENTITY not_1 IS
        PORT (a: IN bit; output: OUT bit);
END not_1;

ARCHITECTURE data_flow OF not_1 IS
BEGIN
        output <= NOT(a);
END data_flow;
```

```
ENTITY and_2 IS
        PORT (a,b: IN bit; output: OUT bit);
END and_2;

ARCHITECTURE data_flow OF and_2 IS
BEGIN
        output <= a AND b;
END data_flow;
```

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Tipos de Dados – Arquitetura Body 3

```
ENTITY or_2 IS
        PORT (a,b: IN bit; output: OUT bit);
END or_2;

ARCHITECTURE data_flow OF or_2 IS
BEGIN
        output <= a OR b;
END data_flow;
```

```
ENTITY and_3 IS
        PORT (a,b,c: IN bit; output: OUT bit);
END and_3;

ARCHITECTURE data_flow OF  and_3 IS
BEGIN
        output <= a AND b AND c;
END data_flow;
```

Projetos de Sistemas Embarcados

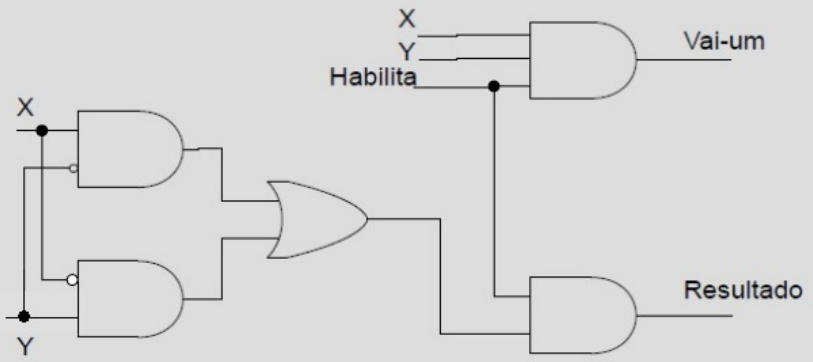
Linguagem VHDL – Declaração de Porta – Tipos de Dados – Arquitetura Body 3

```
ARCHITECTURE structural OF meio_somador IS
  COMPONENT and2 PORT(a,b: IN bit; output: OUT bit); END COMPONENT;
  COMPONENT and3 PORT(a,b,c: IN bit; output: OUT bit); END COMPONENT;
  COMPONENT or2 PORT(a,b: IN bit; output: OUT bit); END COMPONENT;
  COMPONENT not1 PORT(a: IN bit; output: OUT bit); END COMPONENT;

  FOR ALL: and2 USE ENTITY work.and_2 (data_flow);
  FOR ALL: and3 USE ENTITY work.and_3 (data_flow);
  FOR ALL: or2 USE ENTITY work.or_2 (data_flow);
  FOR ALL: not1 USE ENTITY work.not _1 (data_flow);

  SIGNAL v,w,z,nx ,ny: BIT;

BEGIN
  c1: not1 PORT MAP (x,nx);
  c2: not1 PORT MAP (y,ny);
  c3: and2 PORT MAP (nx,y,v);
  c4: and2 PORT MAP (x,ny,w);
  c5: or2 PORT MAP (v,w,z);
  c6: and2 PORT MAP (habilita,z ,resultado);
  c7: and3 PORT MAP (x,y,habilita,vai_um);
END structural;
```

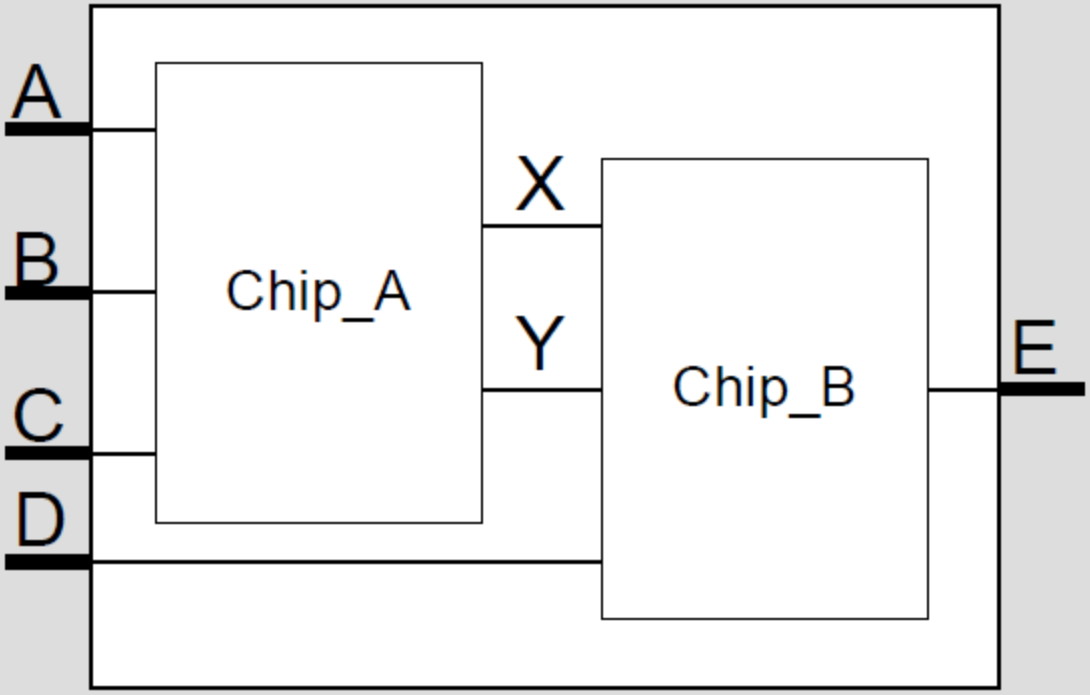


Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Port Map

```
Chip1 : Chip_A
Port map (A,B,C,X,Y);

Chip2 : Chip_B
Port map (X,Y,D,E);
```



Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Port Map Exemplo

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY TEST IS
PORT (A,B,C,D  :  IN STD_LOGIC;
          E          : OUT STD_LOGIC);
END TEST;

ARCHITECTURE BEHAVIOR OF TEST IS

SIGNAL X,Y : STD_LOGIC;

COMPONENT Chip_A
PORT (L,M,N   :  IN STD_LOGIC;
          O,P    : OUT STD_LOGIC);
END COMPONENT;
```

```
COMPONENT Chip_B
PORT (Q,R,S :  IN STD_LOGIC;
             T  : OUT STD_LOGIC);
END COMPONENT;

BEGIN

Chip1 : Chip_A
PORT MAP (A,B,C,X,Y);

Chip2 : Chip_B
PORT MAP (X,Y,D,E);

END BEHAVIOR;
```

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Port Map Exemplo AND

```
entity and2 is
    port ( a, b : in bit;  y : out bit );
end entity and2;

architecture basic of and2 is
begin
    and2_behavior : process is
    begin
        y <= a and b after 2 ns;
        wait on a, b;
    end process and2_behavior;
end architecture basic;
```

Projetos de Sistemas Embarcados

Linguagem VHDL – Declaração de Porta – Port Map Exemplo Somador de 1 bit

```
1   ENTITY add1 IS
2   PORT(
3         cin   :IN BIT;
4         a     :IN BIT;
5         b     :IN BIT;
6         s     :OUT BIT;
7         cout  :OUT BIT);
8   END add1;
9
10  ARCHITECTURE a OF add1 IS
11  BEGIN
12
13       s      <= a XOR b XOR cin;
14       cout   <= (a AND b) OR (a AND cin) OR (b AND cin);
15  END a;
```

Continua...