

Introdução a VHDL

VHDL → VHSIC HDL (final de 80 – Nasa-Simulação)

Very Highspeed integrated circuit

HDL Hardware description language

Outras linguagens: Verilog

Característica: processos concorrência

Biblioteca: IEEE

Introdução a VHDL

Estrutura básica:

Entity → descreve E/S (não inclui variáveis internas)

Architecture → procedimento

Procedimento: a) comportamental
b) RTL (Register Transfer Level)
c) lógico (equações booleanas)

Introdução a VHDL

Estrutura:

Entity nome **IS**

[**Port** (lista);]

End [nome].

Exemplo:

Entity NOR is

Port (a,b: in bit; c:
out bit);

END NOR.

Introdução a VHDL

Estrutura:

Architecture Nome.arq **of** nome-entidade **is**

- - Declaração (variáveis locais)

Begin

- - Procedimento (comandos)

End Nome-arq.

Introdução a VHDL

Declaração → dados

Constante: **constant** nome: subtype:=valor;

Exemplos:

```
constant A: integer :=3;
```

```
constant B: standard bit_vector (4 downto 0):=00100;
```

Variable

Integer, Real, Array, Record, File

Exemplos:

```
Variable voltage: integer:=0;
```

```
variable sum: Real;
```

Introdução a VHDL

Operadores:

Lógicos: AND, OR, NAND, NOR, XOR, NOT

Relacionais: =, <, >, <=, >=, /= (diferente)

Atribuição: $c \leq A \text{ NAND } B$
 $c \leq A \text{ AND NOT } B$
 $d \leq A \text{ NAND } B \text{ NAND } C \rightarrow \text{illegal falta } ()\text{'s}$

Obs:

$:=$ para variável

\leq para sinal

Introdução a VHDL

Exemplos de declaração:

```
signal w,x: bit_vector (3 downto 0);
```

```
signal y: bit;
```

```
signal z: it_vector ( 4 downto 0);
```

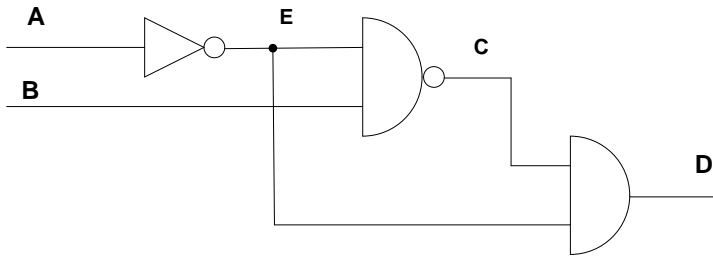
Atribuição:

```
w<= x or y → ilegal (x,y possuem dimensões diferentes)
```

```
Z <= w and x → ilegal ( z possui outra dimensão)
```

Introdução a VHDL

Circuito:



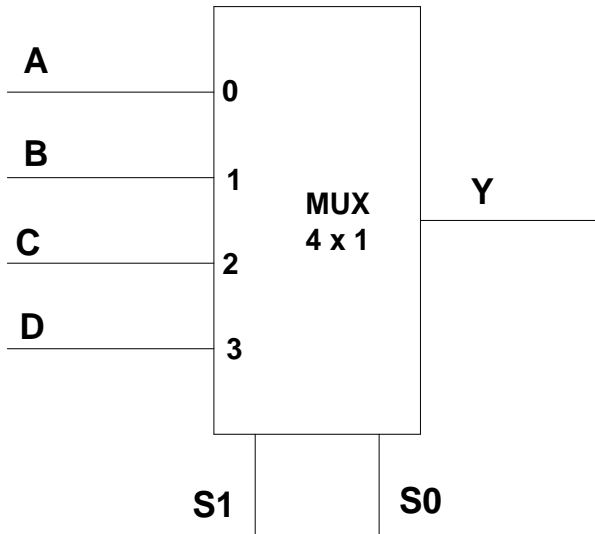
VHDL:

```
library IEEE;  
use IEEE.Std_logic_1164;  
entity C-comb is  
  Port (A,B: in bit; D: out bit);  
end C-comb;  
architecture Ex1 of C-comb is  
  signal C, E: bit;  
begin  
  E<= not(A);  
  C <= not(B and E);  
  D <= C and E;  
end Ex1;
```


Introdução a VHDL

VHDL

Exemplo:



```
entity MUX_4x1 is
```

```
Port (A,B,C,D,S0,S1: in bit; Y: out bit);
```

```
end MUX_4x1;
```

1) Architecture Ex1 of MUX_4x1 is

```
y<= ( A and not S1 and not S0) or
      (B and not S1 and S0) or
      (D and S1 and S0) or
      (C and S1 and not S0);
```

```
end Ex1;
```

Introdução a VHDL

Exemplo: 2) MUX_4x1

architecture Ex2 of MUX_4x1 is

```
Y<= A when (S1= '0' and  
           S0= '0') else
```

```
    B when (S1= '0' and  
           S0= '1') else
```

```
    C when (S1= '1' and  
           S0= '0') else D;
```

```
end Ex2;
```

Exemplo: 3) MUX_4x1

Architecture EX3 of MUX_4x1 is
signal sel: bit_vector (1 downto 0);
begin

```
Y <= A when '00';
```

```
    B when '01';
```

```
    C when '10';
```

```
    D when '11';
```

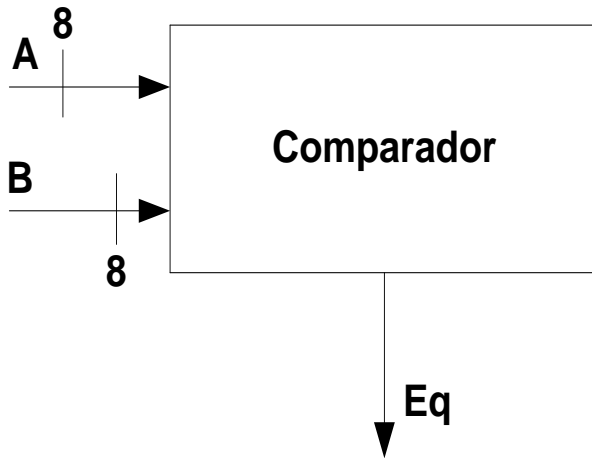
```
    sel<= S1&S0;
```

```
end EX3;
```

Introdução a VHDL

Exemplo:

Comparador
de 8bits

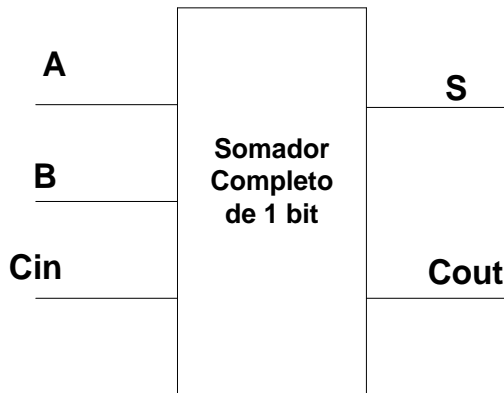


```
entity compara is  
  port (A,B: in bit_vector (0 to 7);  
        eq: out bit);  
end compara;  
architecture ex1 of compara is  
begin  
  eq <= '1' when (A=B) else '0';  
end ex1;
```

Introdução a VHDL

Exemplo-2:

Somador
completo de 1 bit



```
entity soma is
port (A,B,Cin: in bit; S,Cout: out
      bit);
end soma;
architecture ex2 of soma is
begin
  S<=A XOR B XOR Cin;
  Cout <= (A and B) OR (A and
           Cin) OR (B and Cin);
end ex2;
```