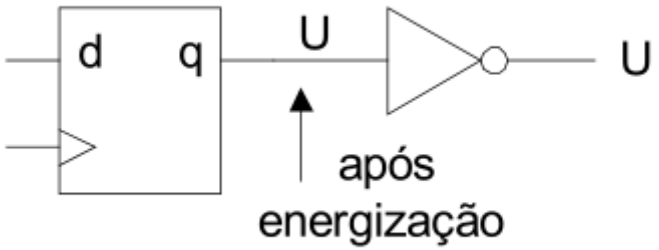
Pacote Padrão IEEE 1164

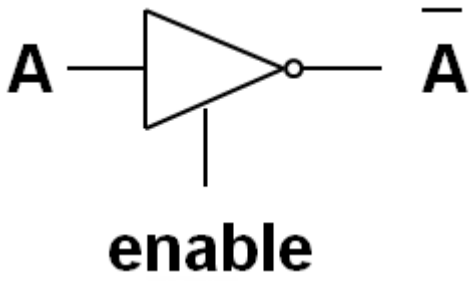
FGA - UnB

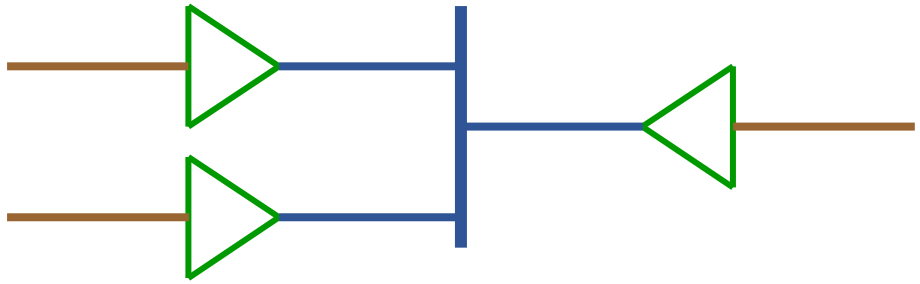
Prática de Eletrônica Digital 1

Prof. Henrique M. T. Menegaz

Insuficiência do Bit





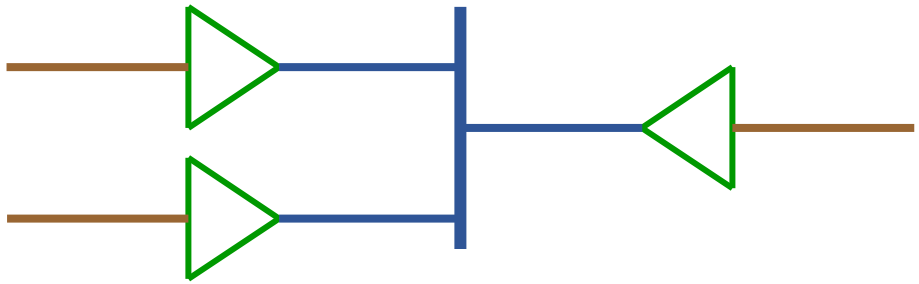


```
LIBRARY ieee
USE ieee.std_logic_1164.ALL
```

std_ulogic

valor do tipo std_ulogic		estado lógico corresponde			
U		uninitialized		não inicializado	
X		forcing unknown		impondo desconhecido	
0	**1**	forcing 0	forcing 1	impondo 0	impondo 1
W		weak unknown		desconhecido fraco	
L	**H**	weak 0	weak 1	0 fraco	1 fraco
Z		high impedance		alta impedância	
-		do not care		não importa	

std_ulogic Vs std_logic

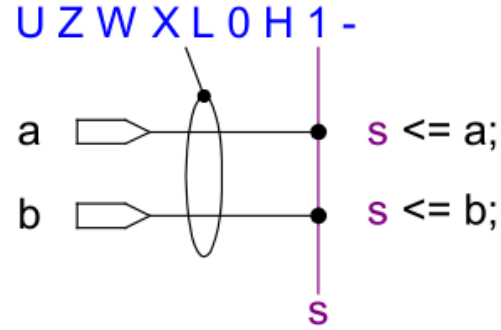


```
s <= a;

s <= b;

s <= c;
```

- **std_ulogic:** unresolved
- **std_logic:** resolved
  - Subtipo de std_ulogic

Função de resolução – tipos std_logic

- **2 controladores acionando o mesmo sinal**



U	U	U	U	U	U	U	U
X	X	X	X	X	X	X	U
0 / 1	X	0 / 1	0 / 1	0 / 1	*Nota A	X	U
W	X	W	W	W	0 / 1	X	U
L / H	X	L / H	*Nota B	W	0 / 1	X	U
Z	X	Z	L / H	W	0 / 1	X	U
-	X	X	X	X	X	X	U
	-	**Z**	**L / H**	**W**	**0 / 1**	**X**	**U**

*Nota A: controladores com níveis lógicos: iguais → resultam no mesmo valor;
diferentes → resultam no valor X

*Nota B: controladores com níveis lógicos: iguais → resultam no mesmo valor
diferentes → resultam no valor W

- **Subtipos de** std_**u**logic

  - Troca de valores possíveis

- **Todos definidos com função de resolução** (denominada **resolved**)

  - mais de um controlador pode acionar

```
SUBTYPE std_logic IS resolved std_ulogic;
SUBTYPE X01       IS resolved std_ulogic RANGE 'X' TO '1'; --('X','0','1')
SUBTYPE X01Z      IS resolved std_ulogic RANGE 'X' TO 'Z'; --('X','0','1','Z')
SUBTYPE UX01      IS resolved std_ulogic RANGE 'U' TO '1'; --('U','X','0','1')
SUBTYPE UX01Z     IS resolved std_ulogic RANGE 'U' TO 'Z'; --('U','X','0','1','Z')
```

Operadores lógicos para tipos std_ulogic

- **Válidos para os subtipos**:
  - std_logic X01 X01Z UX01 UX01Z

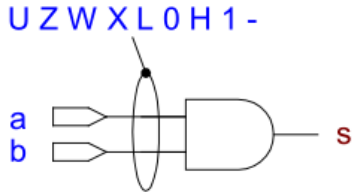
- **Valor de retorno tipo UX01**:
  - modela a porta que impõe estados com maior força: 1 0 U X

```
FUNCTION "and"  ( l : std_ulogic; r : std_ulogic ) RETURN UX01;
FUNCTION "nand" ( l : std_ulogic; r : std_ulogic ) RETURN UX01;
FUNCTION "or"   ( l : std_ulogic; r : std_ulogic ) RETURN UX01;
FUNCTION "nor"  ( l : std_ulogic; r : std_ulogic ) RETURN UX01;
FUNCTION "xor"  ( l : std_ulogic; r : std_ulogic ) RETURN UX01;
FUNCTION "xnor" ( l : std_ulogic; r : std_ulogic ) RETURN UX01;
FUNCTION "not"  ( l : std_ulogic                 ) RETURN UX01;
```

- Nota: a função xnor não é suportada no padrão VHDL-1987

Operadores lógicos para tipos std_ulogic

- **Exemplo**: valor de retorno para a função “and”



s <= a AND b;

```
FUNCTION "and"  ( l : std_ulogic; r : std_ulogic ) RETURN UX01;
```

- **Observe**: retorna tipo UX01 (modela saída com força elevada)

	U	X	0	1	Z	W	L	H	-
U	U	U	0	U	U	U	0	U	U
X	U	X	0	X	X	X	0	X	X
0	0	0	0	0	0	0	0	0	0
1	U	X	0	1	X	X	0	1	X
Z	U	X	0	X	X	X	0	X	X
W	U	X	0	X	X	X	0	X	X
L	0	0	0	0	0	0	0	0	0
H	U	X	0	1	X	X	0	1	X
-	U	X	0	X	X	X	0	X	X

Operadores lógicos para tipos std_logic_vector std_ulogic_vector

```
FUNCTION "and"  ( l, r : std_logic_vector ) RETURN std_logic_vector;
FUNCTION "and"  ( l, r : std_ulogic_vector) RETURN std_ulogic_vector;
FUNCTION "nand" ( l, r : std_logic_vector ) RETURN std_logic_vector;
FUNCTION "nand" ( l, r : std_ulogic_vector) RETURN std_ulogic_vector;
FUNCTION "or"   ( l, r : std_logic_vector ) RETURN std_logic_vector;
FUNCTION "or"   ( l, r : std_ulogic_vector) RETURN std_ulogic_vector;
FUNCTION "nor"  ( l, r : std_logic_vector ) RETURN std_logic_vector;
FUNCTION "nor"  ( l, r : std_ulogic_vector) RETURN std_ulogic_vector;
FUNCTION "xor"  ( l, r : std_logic_vector ) RETURN std_logic_vector;
FUNCTION "xor"  ( l, r : std_ulogic_vector) RETURN std_ulogic_vector;
FUNCTION "xnor" ( l, r : std_logic_vector ) RETURN std_logic_vector;   -- VHDL-1993
FUNCTION "xnor" ( l, r : std_ulogic_vector) RETURN std_ulogic_vector;  -- VHDL-1993
FUNCTION "not"  ( l    : std_logic_vector ) RETURN std_logic_vector;
FUNCTION "not"  ( l    : std_ulogic_vector) RETURN std_ulogic_vector;
```

- Nota: **std_logic_vector** não é um subtipo de **std_ulogic_vector**:
  - é necessário definir funções para cada tipo

Conversão para std_logic_vector std_ulogic_vector

- **Conversões para std_logic std_ulogic std_logic_vector std_ulogic_vector:**
  - mudam o tipo mantendo o valor

```
FUNCTION To_StdULogic       (b : BIT              ) RETURN std_ulogic;
FUNCTION To_StdLogicVector  (b : BIT_VECTOR       ) RETURN std_logic_vector;
FUNCTION To_StdLogicVector  (s : std_ulogic_vector) RETURN std_logic_vector;
FUNCTION To_StdULogicVector (b : BIT_VECTOR       ) RETURN std_ulogic_vector;
FUNCTION To_StdULogicVector (s : std_logic_vector ) RETURN std_ulogic_vector;
```

- Nota: **std_logic_vector** não é um subtipo de **std_ulogic_vector**:
  - é necessário definir funções para cada tipo

Detecção de bordas de subidas ou descidas

- **Úteis para a descrição de circuitos sensíveis a bordas de um sinal de relógio**

```
FUNCTION rising_edge  (SIGNAL s : std_ulogic) RETURN BOOLEAN;
FUNCTION falling_edge (SIGNAL s : std_ulogic) RETURN BOOLEAN;
```

- **Detecção de valores sem correspondência com nível lógico alto ou baixo**
  - Retornam verdadeiro para: U X Z W -

```
FUNCTION Is_X ( s : std_ulogic_vector ) RETURN  BOOLEAN;
FUNCTION Is_X ( s : std_logic_vector  ) RETURN  BOOLEAN;
FUNCTION Is_X ( s : std_ulogic        ) RETURN  BOOLEAN;
```