

VHDL: Funciones de conversión

Microarquitecturas y Softcores



Laboratorio de
Sistemas Embebidos



**FACULTAD
DE INGENIERIA**
Universidad de Buenos Aires

Funciones de conversión

ieee.numeric_std:

Es un paquete de la librería estándar de la IEEE

ieee.std_logic_arith, ieee.std_logic_unsigned/signed:

Paquetes de Synopsys. Eran usados casi por defecto por ser una de las empresas cuyo software es uno de los más usados

Funciones de conversión

Nunca utilizar los siguientes paquetes en el mismo proyecto:

- `ieee.numeric_std.all`
- `ieee.std_logic_arith.all`, `ieee.std_logic_unsigned`,
`ieee.std_logic_unsigned`

Funciones de conversión

ieee.numeric_std:

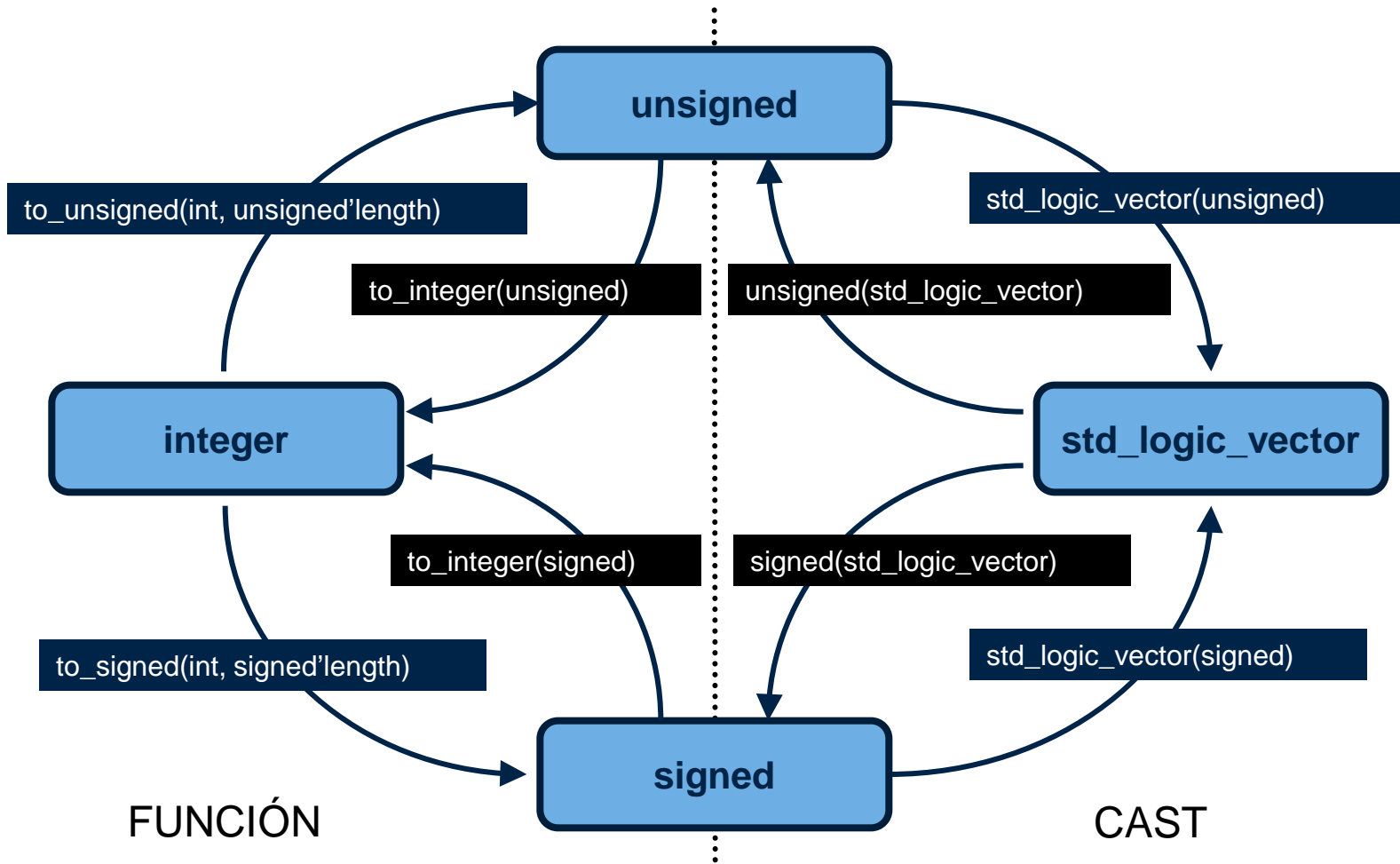
- No tiene definidas las operaciones matemáticas para `std_logic`, `std_logic_vector`
- Tiene definidas las operaciones matemáticas para los tipos `signed`, `unsigned` e `integer`

ieee.std_logic_arith, ieee.std_logic_unsigned/signed:

- Tiene definidas las operaciones matemáticas para los tipos `std_logic`, `std_logic_vector` e `integer`

Funciones de conversión

- Diagrama de conversiones y casteos



Funciones de conversión: Ejemplo 1

-- Declaración de librerías (incluir numeric_std)

entity contador **is**

port (

clk, rst, ld: **in** std_logic;

initial_value: **in** std_logic_vector(3 **downto** 0);

count: **out** std_logic_vector(3 **downto** 0)

);

end contador;

architecture beh **of** contador **is**

begin

count_proc: **process**(clk, rst)

variable count_i: integer range 0 to 16;

begin

if (rst='1') **then**

count_i := 0;

elsif (rising_edge(clk)) **then**

if (ld = '1') **then**

count_i := to_integer(unsigned(initial_value));

else

count_i := count_i + 1;

if count_i = 16 **then**

count_i := 0;

end if;

end if;

end if;

count <= std_logic_vector(to_unsigned(count_i,4));

end process count_proc;

end architecture;

Funciones de conversión: Ejemplo 2

-- Declaración de librerías (incluir numeric_std)

```
entity contador is
  port (
    clk, rst, ld: in std_logic;
    initial_value: in std_logic_vector(3 downto 0);
    count: out std_logic_vector(3 downto 0)
  );
end contador;

architecture beh of contador is
begin
  count_proc: process(clk, reset)
    variable count_i: unsigned(3 downto 0);
  begin
    if (rst='1') then
      count_i <= (others => '0');
    elsif (rising_edge(clk)) then
      if (ld = '1') then
        count_i := unsigned(initial_value);
      else
        count_i <= count_i + 1;
      end if;
    end if;
    count <= std_logic_vector(count_i);
  end process count_proc;
end architecture;
```

Funciones de conversión

FIN