VHDL Signal Types

Last updated 8/20/24

VHDL Signal Types

- See the VHDL Types notes for a complete description of the ‘types’ available in VHDL

- These slides review the acceptable VHDL signal types for synthesis

VHDL Signal Types

- Acceptable Signal Types for Synthesis – std_logic_1164
  - Used for wires
  - Used for busses that are treated as a collection of wires
  - 2 types
    - std_logic
    - std_logic_vector
  - Operators
    - Comparison*: =, /=
    - Shifting: srl, sll, rol, ror
    - Boolean: not, and, or, nand, nor, xor, xnor
    - Concatenation: &

Value	Definition	Synthesizable
'0'	Forcing 0	Y
'1'	Forcing 1	Y
'Z'	High Impedance	Y

* Comparison of std_logic_vectors can return unexpected results
  Limited to = and /=

VHDL Signal Types

- Acceptable Signal Types for Synthesis – numeric_std
  - Used for busses that are treated as a signal collectively
  - 2 types
    - signed array of std_logic (analogous to a std_logic_vector)
    - unsigned array of std_logic (analogous to a std_logic_vector)
  - Values
    - signed is interpreted as 2's complement (positive and negative)
    - unsigned is interpreted as unsigned magnitude (always positive)
  - Physical implementation
    - Signed and unsigned signals are implemented identically – as a group of wires
  - Synthesis
    - Different logic solutions may be created for signed and unsigned signals due to their interpretation as signed or unsigned values

VHDL Signal Types

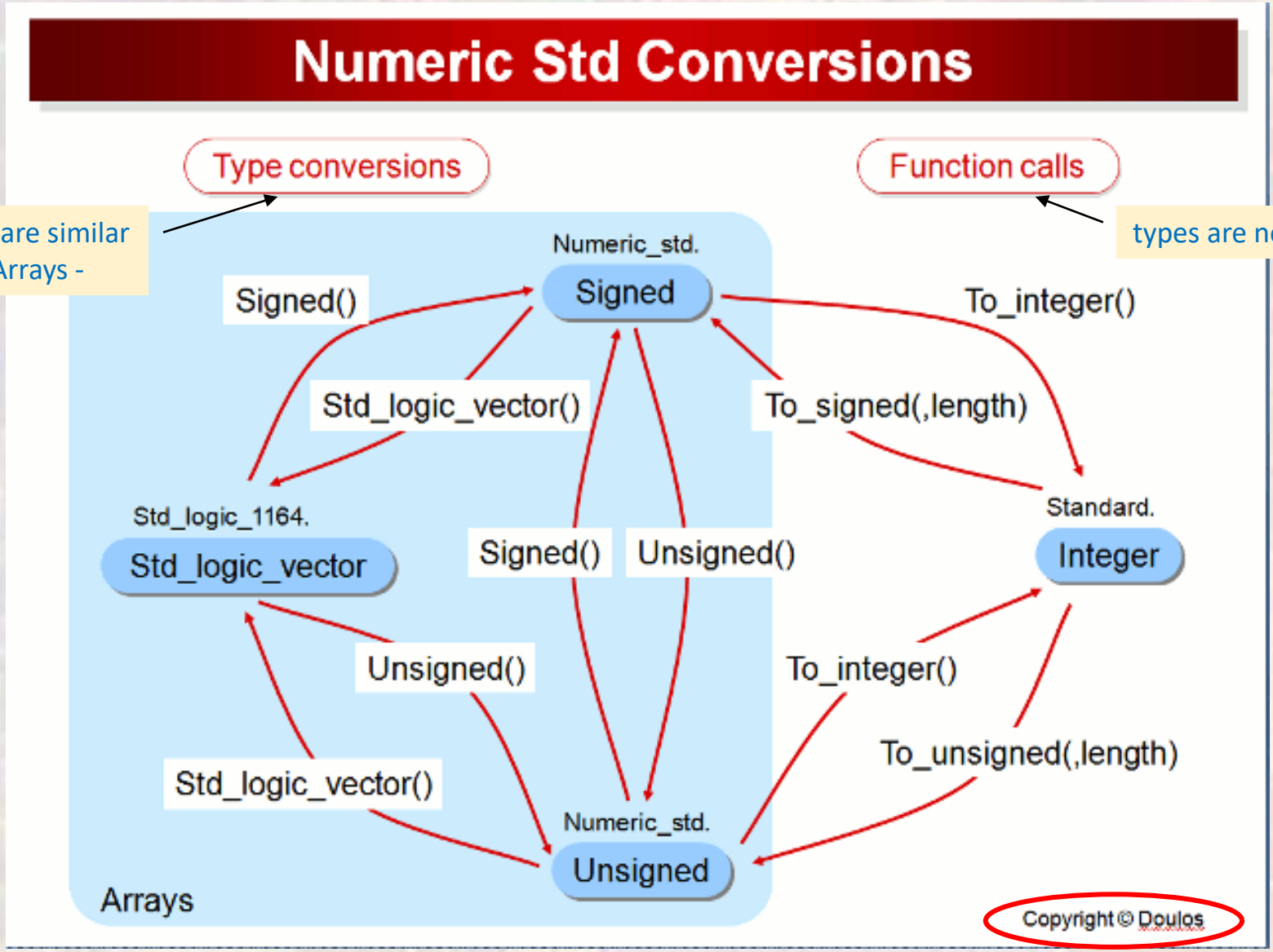
- Acceptable Signal Types for Synthesis – numeric_std
  - Operators
    - Comparison: =, /=, <, <=, >, >=
    - Shifting: srl, sll, rol, ror
    - Boolean: not†, and, or, nand, nor, xor, xnor
    - Concatenation: &
    - Arithmetic††:
      - - †††
      - abs †††
      - +, -
      - *, / ††††
      - mod, rem
      - ** †††††

†	negation of 2's complement most negative value will return the most negative value
††	arithmetic operators other than multiplication preserve the length of the result vector i.e. wrap
†††	signed only
††††	* and / will create large logical solutions
†††††	only use with a base of 2

VHDL Signal Types

- Acceptable Signal Types for Synthesis – numeric_std
  - Functions
    - resize — resize unsigned using zero extension; resize signed using sign extension
    - shift_right() — unsigned using zero extension; signed using sign extension
    - shift_left() — uses zero extension for signed and unsigned
    - rotate_right() — more robust solution for signed and unsigned signals
    - rotate_left() — more robust solution for signed and unsigned signals

VHDL Signal Types



VHDL Signal Types

- Signal type conversion – examples
  - std_logic_vector to signed

    signed_sig <= signed(slv_sig);

    sum_sig <= (signed(slv_sig1) + signed(slv_sig2));

    type conversion

    function call

  - unsigned to std_logic_vector

    slv_sig <= std_logic_vector(unsigned_sig);

    slv_sig <= std_logic_vector(unsigned_sig1 – unsigned_sig2);

  - integer to signed

    signed_sig <= to_signed(int_val, num_bits); -- set signed_sig to int_val using numbits

  - unsigned to integer

    wire_val <= slv_sig(to_integer(addr_sig)); -- choose addr_sig wire in slv_sig vector

  - std_logic_vector to integer

    wire_val <= slv_sig(to_integer(signed(slv_sig2))); -- must covert to signed or unsigned

  - integer to std_logic_vector

    slv_sig <= std_logic_vector(to_signed(int_val, num_bits)); -- must covert to signed or unsigned