

Signal Types (1A)

Copyright (c) 2025 - 2012 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

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

https://faculty-web.msoe.edu/johnsontimof/ELE3510/files3510/vhdl_signal_types.pdf

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

https://faculty-web.msoe.edu/johnsontimof/ELE3510/files3510/vhdl_signal_types.pdf

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

https://faculty-web.msoe.edu/johnsontimmoj/ELE3510/files3510/vhdl_signal_types.pdf

VHDL Signal Types

Acceptable Signal Types for Synthesis – numeric_std

- Functions

- 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

https://faculty-web.msoe.edu/johnsontimmoj/ELE3510/files3510/vhdl_signal_types.pdf

Numeric Std Conversions

Acceptable Signal Types for Synthesis – numeric_std

- Functions

- 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

https://faculty-web.msoe.edu/johnsontimmoj/ELE3510/files3510/vhdl_signal_types.pdf

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));
```

- 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
```

https://faculty-web.msoe.edu/johnsontimoj/ELE3510/files3510/vhdl_signal_types.pdf

References

- [1] <http://en.wikipedia.org/>
- [2] J. V. Spiegel, VHDL Tutorial,
http://www.seas.upenn.edu/~ese171/vhdl/vhdl_primer.html
- [3] J. R. Armstrong, F. G. Gray, Structured Logic Design with VHDL
- [4] Z. Navabi, VHDL Analysis and Modeling of Digital Systems
- [5] D. Smith, HDL Chip Design
- [6] <http://www.csee.umbc.edu/portal/help/VHDL/stdpkg.html>
- [7] VHDL Tutorial - VHDL online www.vhdl-online.de/tutorial/