

Convert from Integer to Signed using Numeric_Std

The below example uses the `to_signed` conversion, which requires two input parameters. The first is the signal that you want to convert, the second is the length of the resulting vector.

```
1 signal input_3 : integer;
2 signal output_3 : signed(3 downto 0);
3
4 output_3 <= to_signed(input_3, output_3'length);
```

Convert from Integer to Std_Logic_Vector using Numeric_Std

First you need to think about the range of values stored in your integer. Can your integer be positive *and* negative? If so, you will need to use the `to_signed()` conversion. If your integer is only positive, you will need to use the `to_unsigned()` conversion.

Both of these conversion functions require two input parameters. The first is the signal that you want to convert, the second is the length of the resulting vector.

```
1 signal input_1 : integer;
2 signal output_1a : std_logic_vector(3 downto 0);
3 signal output_1b : std_logic_vector(3 downto 0);
4
5 -- This line demonstrates how to convert positive integers
6 output_1a <= std_logic_vector(to_unsigned(input_1, output_1a'length));
7
8 -- This line demonstrates how to convert positive or negative integers
9 output_1b <= std_logic_vector(to_signed(input_1, output_1b'length));
```

Convert from Integer to Unsigned using Numeric_Std

The below example uses the `to_unsigned` conversion, which requires two input parameters. The first is the signal that you want to convert, the second is the length of the resulting vector.

```
1 signal input_2 : integer;
2 signal output_2 : unsigned(3 downto 0);
3
4 output_2 <= to_unsigned(input_2, output_2'length);
```

Convert from Std_Logic_Vector to Integer using Numeric_Std

First you need to think about the data that is represented by your `std_logic_vector`. Is it signed data or is it unsigned data? Signed data means that your `std_logic_vector` can be a positive or negative number. Unsigned data means that your `std_logic_vector` is *only* a positive number. The example below uses the `unsigned()` typecast, but if your data can be negative you need to use the `signed()` typecast. Once you cast your input `std_logic_vector` as unsigned or signed, then you can convert it to integer as shown below:

```
1 signal input_4 : std_logic_vector(3 downto 0);
2 signal output_4a : integer;
3 signal output_4b : integer;
4
5 -- This line demonstrates the unsigned case
6 output_4a <= to_integer(unsigned(input_4));
7
8 -- This line demonstrates the signed case
9 output_4b <= to_integer(signed(input_4));
```

Convert from Std_Logic_Vector to Signed using Numeric_Std

This is an easy conversion, all you need to do is cast the `std_logic_vector` as signed as shown below:

```
1 signal input_6 : std_logic_vector(3 downto 0);
2 signal output_6 : signed(3 downto 0);
3
4 output_6 <= signed(input_6);
```

Convert from Std_Logic_Vector to Unsigned using Numeric_Std

This is an easy conversion, all you need to do is cast the `std_logic_vector` as unsigned as shown below:

```
1 signal input_5 : std_logic_vector(3 downto 0);
2 signal output_5 : unsigned(3 downto 0);
3
4 output_5 <= unsigned(input_5);
```

Convert from Signed to Integer using Numeric_Std

This is an easy conversion, all you need to do is use the `to_integer` function call from `numeric_std` as shown below:

```
1 signal input_10 : signed(3 downto 0);
2 signal output_10 : integer;
3
4 output_10 <= to_integer(input_10);
```

Convert from Signed to Std_Logic_Vector using Numeric_Std

This is an easy conversion, all you need to do is use the `std_logic_vector` cast as shown below:

```
1 signal input_11 : signed(3 downto 0);
2 signal output_11 : std_logic_vector(3 downto 0);
3
4 output_11 <= std_logic_vector(input_11);
```

Convert from Signed to Unsigned using Numeric_Std

This is an easy conversion, all you need to do is use the `unsigned` cast as shown below:

```
1 signal input_12 : signed(3 downto 0);
2 signal output_12 : unsigned(3 downto 0);
3
4 output_12 <= unsigned(input_12);
```

Convert from Unsigned to Integer using Numeric_Std

This is an easy conversion, all you need to do is use the `to_integer` function call from `numeric_std` as shown below:

```
1 signal input_7 : unsigned(3 downto 0);
2 signal output_7 : integer;
3
4 output_7 <= to_integer(input_7);
```

Convert from Unsigned to Signed using Numeric_Std

This is an easy conversion, all you need to do is use the `signed` cast as shown below:

```
1 signal input_9 : unsigned(3 downto 0);
2 signal output_9 : signed(3 downto 0);
3
4 output_9 <= signed(input_9);
```

Convert from Unsigned to Std_Logic_Vector using Numeric_Std

This is an easy conversion, all you need to do is use the `std_logic_vector` cast as shown below:

```
1 signal input_8 : unsigned(3 downto 0);
2 signal output_8 : std_logic_vector(3 downto 0);
3
4 output_8 <= std_logic_vector(input_8);
```