

Elektronika analogowa i cyfrowa

Dr Michał Tanaś
(<http://mtanas.home.amu.edu.pl/>)

VHDL – NUMERIC_STD

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.NUMERIC_STD.ALL;
```

Kolejna warstwa abstrakcji, umożliwiająca bezpośrednie programowanie działań arytmetycznych. VHDL sam skonstruuje odpowiedni układ bramek logicznych.

Żeby jej używać należy zadeklarować użycie biblioteki NUMERIC_STD

A next abstraction layer that allows to directly program arithmetic operations. VHDL itself construct the underlying logic gate circuit.

To use it the library NUMERIC_STD must be declared.

VHDL – NUMERIC_STD

New Source Wizard

Define Module
Specify ports for module.

Entity name

Architecture name

Port Name	Direction	Bus	MSB	LSB
A	in	☒	7	0
B	in	☒	7	0
S	out	☒	7	0
	in	☐		
	in	☐		
	in	☐		
	in	☐		
	in	☐		

[More Info](#) [< Back](#) [Next >](#) [Cancel](#)

Specyfikując nóżki we/wy zaznaczamy „bus” i podajemy zakres bitów (MSB – most significant bit, LSB – least significant bit).

When i/o pins are specified mark „bus” and set bit range (MSB – most significant bit, LSB – least significant bit)

VHDL – NUMERIC_STD

entity add8 is

Port (

**A : in STD_LOGIC_VECTOR(7
downto 0);**

**B : in STD_LOGIC_VECTOR(7
downto 0);**

**S : out STD_LOGIC_VECTOR(7
downto 0);**

);

end add8;

I w rezultacie zamiast ręcznie specyfikować piny A0, A1, A2, A3, A4, A5, A6, A7 itd. od razu wyspecyfikowaliśmy cały wektor.

And so instead of manual pin specification A0, A1, A2, A3, A4, A5, A6, A7 etc. a whole vector is specified at once.

VHDL – NUMERIC_STD

signal intA : integer;

signal intB : integer;

signal intS: integer;

Sygnal może być typu „integer”, czyli zmienną całkowitą znaną z innych języków programowania.

Uwaga! W VHDL oprócz „signal” istnieje również „variable” o identycznej składni i podobnym zastosowaniu. „variable” działa jednak w nieco inny sposób, proszę nie używać „variable” na zajęciach.

Signal can has type „integer”, which is an integer variable from other programming languages.

Notice, that in VHDL there is also „variable” statement with syntax identical of „signal”. However „variable” works slightly differently than „signal”. Please don’t use „variable” during classes.

VHDL – NUMERIC_STD

```
intA <= to_integer(unsigned(A));
```

Funkcje konwersji typów danych, znane również z innych języków programowania. Tutaj 8-bitowy wektor jest konwertowany na liczbę 8-bitową liczbę całkowitą bez znaku.

Data types conversion functions, known also from other programming languages. Here 8-bit vector is converted to 8-bit unsigned integer number

VHDL – NUMERIC_STD

```
intS <= intA + intB;
```

Standardowa arytmetyka liczb całkowitych, jak w innych językach programowania.

Standard integer arithmetic, like in other programming languages.

VHDL – NUMERIC_STD

```
S <= std_logic_vector(to_unsigned(intS, 8));
```

Na koniec wynik trzeba z powrotem skonwertować do wektora.

Uwaga, trzeba jawnie podać długość wektora.

Finally, the result must be converted back to vector. Notice that vector length must be explicitly stated.

VHDL – NUMERIC_STD

```
A<="00000010";
```

W testbench'u można ustawić stan wektora jednym przypisaniem, podając od razu stan wszystkich bitów.

In testbench the state of the whole vector can be set using a single assignment, where the state of the all bits are given.

VHDL – NUMERIC_STD

Name	Value	955 ns	960 ns	965 ns	970 ns	975 ns
a[7:0]	00000010	00001010	00000010	00001010	00000010	00001010
b[7:0]	00000001	00000101	00000001	00000101	00000001	00000101
s[7:0]	00000011	00001111	00000011	00001111	00000011	00001111
c_period	10000 ps					10000 ps

Zauważmy w jaki sposób pokazywane są stany wektorów w trakcie symulacji.

Notice how the vector states are shown during simulation.