A Matrix is just a way of looking at a group of associated numbers.

- Functions similar to those in "Matlab".
- Based on the built in types "real", "integer"
- Compatible with VHDL-93, VHDL-2001 and VHDL-2008
- Extended to work with complex numbers
- Extendable to cover synthesizable data types.

This is a proposed package for the VHDL-201X standard.

Packages can be downloaded from: http://www.vhdl.org/fphdl

---

**Type Definitions**

- Two new types are defined.
- The real_matrix is an array of type COMPLEX_POLAR.
- The integer_matrix is an array of type COMPLEX.

And in the Complex_Matix package:
- The complex_matrix is an array of type COMPLEX_POLAR.
- The complex_polar_matrix is an array of type COMPLEX.

Dependancies:
- ieee.math_real.all;
- ieee.math_complex.all;

---

**Matrix Math package for VHDL**

- A basic matrix math function is defined.
  - use ieee_proposed.real_matrix_pkg.all;
    - signal a : real_matrix (1 to 2, 1 to 4);
    - signal b : real_matrix (2 to 3, 1 to 2);
    - signal c : real_matrix (1 to 3, 1 to 4);
    - begin
      - c <= a * b; -- Matrix Multiply

- The Matrix Inversion function is defined.
  - use ieee_proposed.real_matrix_pkg.all;
    - signal a : real_matrix (0 to 2, 0 to 2);
    - signal b : real_matrix (0 to 2, 0 to 2);
    - ... b <= inv(a);
    - For a 2x2 Matrix:
      - $A^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
    - For a 3x3 Matrix:
      - $A^{-1} = \begin{bmatrix} a & b & c \\ d & a & b \\ c & d \end{bmatrix}$

- The Inverse matrix function is defined.
  - use ieee_proposed.real_matrix_pkg.all;
    - signal a : real_matrix (0 to 3, 0 to 3);
    - signal b : real_matrix (0 to 3, 0 to 3);
    - ... b <= inv(a);

---

**Operators**

- Comparison operators: =, <, <=, >, >=, !=
- Arithmetic operators: +, -, *, /, **
- Logical operators: and, or, not
- Bitwise operators: &amp;, |, ~
- Relational operators: <, <=, >, >=, ==
- Assignment operators: =, +=, -=, *=, /=, **=

---

**Functions**

- abs(): Returns the absolute value of a number.
- sqrt(): Returns the square root of a number.
- pow(): Returns the base to the power of the exponent.
- round(): Returns the nearest integer to a number.
- floor(): Returns the largest integer less than or equal to a number.
- ceil(): Returns the smallest integer greater than or equal to a number.
- trunc(): Returns the truncated integer.
- min(): Returns the minimum of two numbers.
- max(): Returns the maximum of two numbers.
- pow(): Returns the base to the power of the exponent.
- rem(): Returns the remainder of a division.
- fmod(): Returns the floating-point remainder of a division.
- int(): Returns the integer part of a number.
- int32(): Returns the integer value of a number.
- int64(): Returns the integer value of a number.
- uint32(): Returns the unsigned integer value of a number.
- uint64(): Returns the unsigned integer value of a number.
- float32(): Returns the floating-point value of a number.
- float64(): Returns the floating-point value of a number.
- complex(): Returns a complex number.
- real(): Returns the real part of a complex number.
- imag(): Returns the imaginary part of a complex number.
- conj(): Returns the complex conjugate of a complex number.
- abs(): Returns the absolute value of a complex number.
- arg(): Returns the argument of a complex number.
- abs(): Returns the absolute value of a complex number.
- conj(): Returns the complex conjugate of a complex number.
- imag(): Returns the imaginary part of a complex number.
- real(): Returns the real part of a complex number.
- pow(): Returns the base to the power of the exponent.
- rem(): Returns the remainder of a division.
- fmod(): Returns the floating-point remainder of a division.
- int(): Returns the integer part of a number.
- int32(): Returns the integer value of a number.
- int64(): Returns the integer value of a number.
- uint32(): Returns the unsigned integer value of a number.
- uint64(): Returns the unsigned integer value of a number.
- float32(): Returns the floating-point value of a number.
- float64(): Returns the floating-point value of a number.
- complex(): Returns a complex number.
- real(): Returns the real part of a complex number.
- imag(): Returns the imaginary part of a complex number.
- conj(): Returns the complex conjugate of a complex number.
- abs(): Returns the absolute value of a complex number.
- arg(): Returns the argument of a complex number.
- abs(): Returns the absolute value of a complex number.
- conj(): Returns the complex conjugate of a complex number.
- imag(): Returns the imaginary part of a complex number.
- real(): Returns the real part of a complex number.
- pow(): Returns the base to the power of the exponent.
- rem(): Returns the remainder of a division.
- fmod(): Returns the floating-point remainder of a division.
- int(): Returns the integer part of a number.
- int32(): Returns the integer value of a number.
- int64(): Returns the integer value of a number.
- uint32(): Returns the unsigned integer value of a number.
- uint64(): Returns the unsigned integer value of a number.
- float32(): Returns the floating-point value of a number.
- float64(): Returns the floating-point value of a number.
- complex(): Returns a complex number.
- real(): Returns the real part of a complex number.
- imag(): Returns the imaginary part of a complex number.
- conj(): Returns the complex conjugate of a complex number.
- abs(): Returns the absolute value of a complex number.
- arg(): Returns the argument of a complex number.
- abs(): Returns the absolute value of a complex number.
- conj(): Returns the complex conjugate of a complex number.
- imag(): Returns the imaginary part of a complex number.
- real(): Returns the real part of a complex number.
- pow(): Returns the base to the power of the exponent.
- rem(): Returns the remainder of a division.
- fmod(): Returns the floating-point remainder of a division.
- int(): Returns the integer part of a number.
- int32(): Returns the integer value of a number.