

Data Objects

Copyright (c) 2011-2012 Young W. Lim.

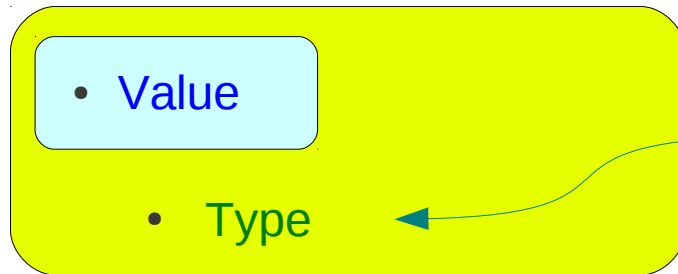
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is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

Data Object

Data Object



*Set of values that the object can have +
Set of operations that are allowed*

Classes of Data Object

- Constant – no change
- Variable – change without any delay
- Signal – change with a certain / delta delay

Classes of Data Object

To model the behavior of a circuit

- **Constant** – no change

*static : local to process, be held until the next process call
can be declared in processes, procedures, functions, architectures*

- **Variable** – change without any delay

*dynamic : not be held from one call to the next
must be declared inside a process*

Represents wires in the schematic of a circuit

- **Signal** – change with a certain / delta delay

must be declared outside a process

Variable Example

```
architecture varch of vent is
  signal trigger, result : integer := 0;
begin
  process
    variable var1: integer := 1;
    variable var2: integer := 2;
    variable var3: integer := 3;
  begin
    wait on trigger;
    var1 := var2;
    var2 := va1 + var3;
    var3 := var2;
    result <= var1 + var2 + var3;
  end process
end varch
```

The diagram illustrates the dependencies between the assignment statements in the process. Green arrows show the flow of data dependencies:

- A green arrow points from `var2` in the second statement to `var1` in the third statement.
- A green arrow points from `var1` and `var3` in the second statement to `var2` in the third statement.
- A green arrow points from `var2` in the third statement to `var3` in the fourth statement.
- A green arrow points from `var2` in the third statement to `var2` in the third statement, forming a loop.

Signal Example

```
architecture sarch of sent is
    signal trigger, result : integer := 0;
begin
    process
        signal sig1: integer := 1;
        signal sig2: integer := 2;
        signal sig3: integer := 3;
    begin
        wait on trigger;
        sig1 <= sig2;
        sig2 <= sig1 + sig3;
        sig3 <= sig2;
        result <= sig1 + sig2 + sig3;
    end process
end sarch
```

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/