SL1Calo VHDL Guidelines

Version 0.4 July 17, 2009

There are two levels of advice in this document:

- **Rules**, which must be obeyed,
- **Recommendations**, which designers are encouraged to follow for their own benefit.

General requirements

Rule: a VHDL file must contain no more than one component (entity and architecture) or one library**.**

Rule: the code must contain a maximum of one statement per line.
Recommendation: lines of code should be no more than 80 characters long.

```
-- EXAMPLE:
--------------------------------------------------------------------------------
  data     <= moddata when modden = '1' else (others => 'Z');
--------------------------------------------------------------------------------
```

Rule: code must be properly indented to indicate the hierarchy of entities, architectures, processes, loops, etc. The depth of the indentation must be consistent throughout the code
Recommendation: space characters should be used rather than tabs to indent the code.
Recommendation: inline comments should be indented and aligned to enhance readability.

```
-- EXAMPLE:
--------------------------------------------------------------------------------
entity vmeif is
  port (
    data:     inout  std_logic_vector(31 downto 0);  -- VME data bus
    clk:      in     std_logic                       -- FPGA clock
  );
end vmeif;
--------------------------------------------------------------------------------
```

Recommendation: related constructs should be grouped together, and these groups should be separated using blank lines or lines made of dashes where this increases the readability.

```
-- EXAMPLE:
--------------------------------------------------------------------------------
  signal   vmeadh:   std_logic;                      -- VME base address high
  signal   vmeadl:   std_logic;                      -- VME base address low
  signal   vmeam:    std_logic_vector(5 downto 0);   -- VME address modifier
--------------------------------------------------------------------------------
```

Naming conventions

Rule: all design objects (entities, ports, signals, components etc.) must be given meaningful, non-cryptic names, consistent throughout the design.

Example 1

Recommendation: port names should begin with an Uppercase letter.
Recommendation: signal names should be written in lowercase letter
Recommendation: active low ports and signals should have names ending with '_n'.

```
-- EXAMPLE:
-------------------------------------------------------------------------------
entity vmeif is
  port (
    Data:     inout  std_logic_vector(31 downto 0);  -- VME data bus
    Reset_n:  in     std_logic;                      -- FPGA reset
    Clk:      in     std_logic                       -- FPGA clock
  );
end vmeif;
-------------------------------------------------------------------------------
architecture rtl of vmeif is
  -- VME bus signals
  signal   vmeadh_n: std_logic;                      -- base address high
  signal   vmeadl_n: std_logic;                      -- base address low
  signal   vmeam:    std_logic_vector(5 DOWNTO 0);   -- address modifier
begin
end rtl;
-------------------------------------------------------------------------------
```

Example 2

Recommendation: all invented design objects names - entity, architecture, ports, signals, constants, variables, labels (components, processes, etc.), types, states, etc. – should be written in UPPERCASE letters.
Recommendation: only ports of the top entity (connected to the FPGA pins) and ports of the components, directly connected to these pins during instantiation, can be active low signals and should have names ending with '_L' or '_N'.
Recommendation: all signals in the design should be active high signals. Any active low port should be converted to the signal with the same name without '_L' or '_N'.

```
-- EXAMPLE:
-------------------------------------------------------------------------------
entity VMEIF is
  port (
    DATA:     inout  std_logic_vector(31 downto 0);  -- VME data bus
    RESET_N:  in     std_logic;                      -- FPGA reset pin
    CLK:      in     std_logic                       -- FPGA clock pin
  );
end VMEIF;
-------------------------------------------------------------------------------
architecture RTL of VMEIF is
  -- VME bus signals
  signal   RESET:    std_logic;                      -- internal FPGA reset
begin
  RESET <= not RESET_N;                              -- pin to signal
end RTL;
-------------------------------------------------------------------------------
```

Header

Rule: at the beginning of a VHDL file there must be a header with the following information:

- The name of the VHDL entity (name of design);
- Name of the original author;
- The date of creation and last modification;
- A brief but clear description of the design's function (comments);
- A list of modifications (date of modification, the change made, the author of the change).

Recommendation: an email address of the original author may be included in the header.

```
-- EXAMPLE:
--------------------------------------------------------------------------------
-- Design     : VMEIF
-- Author     : Name FAMILYNAME
-- Created    : 01.01.2001 Last Modified: 01.03.2001
-- Comments   : VME interface A24:D32; AM:3E,3D,3A,39; no BLT;
-- 01.02.2001 : first modification details; Name FAMILYNAME
-- 01.03.2001 : second modification details; Name FAMILYNAME
--------------------------------------------------------------------------------
```

Library declaration

```
-- EXAMPLE:
--------------------------------------------------------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
--------------------------------------------------------------------------------
```

Entity declaration

Rule: the top-level entity must have the same name as the device or hardware modeled.

Rule: where possible, ports must be given the same name as appears in the device specification.
Recommendation: legal VHDL names should be used in the device specification document.

Rule: signal types other than std_logic or std_logic_vector must not be used for component ports.
Recommendation: do not declare more than one generic and port per line, each declaration should be followed by a comment clarifying the nature and use of the port, unless this is obvious.

```
-- EXAMPLE:
--------------------------------------------------------------------------------
entity VMEIF is
  generic (
    WIDTH:    integer:=32                             -- generic parameter
  );
  port (
    DATA:     inout  std_logic_vector(31 downto 0);   -- VME data bus
    RESET_N:  in     std_logic;                       -- FPGA reset pin
    CLK:      in     std_logic                        -- FPGA clock pin
  );
end VMEIF;
--------------------------------------------------------------------------------
```

Architecture

Recommendation: each VHDL architecture should begin with a header comment that contains the following:

- a description of the function of the architecture;
- a description of any limitations that the architecture might have;
- a note of any assumptions made during the design process;
- descriptions of any generic parameters.

```
-- EXAMPLE:
--------------------------------------------------------------------------------
architecture RTL of VMEIF is
--------------------------------------------------------------------------------
-- header (functions, limitations, assumptions, generics)
--------------------------------------------------------------------------------
-- constant declarations, component declarations, signal declarations
begin
-- component instances, processes, combinatorial expressions
end RTL;
--------------------------------------------------------------------------------
```

Constant declarations

Rule: constant parameters with meaningful names must be used to avoid burying ‘magic numbers’ within VHDL architecture.

```
-- EXAMPLE:
--------------------------------------------------------------------------------
  constant VERSION:  std_logic_vector(1 downto 0):="10";  -- ver.2
  constant REVISION: std_logic_vector(1 downto 0):="11";  -- rev.3
--------------------------------------------------------------------------------
```

Component declarations

Rule: the identifier, port clause and generic clause of a component declaration must be identical to the declarations in the corresponding entity declaration (i.e. the same identifiers in the same order).

```
-- EXAMPLE:
--------------------------------------------------------------------------------
  component VMEIF
  generic (
    WIDTH:    integer:=32                             -- generic parameter
  );
  port (
    DATA:     inout  std_logic_vector(31 downto 0);   -- VME data bus
    CLK:      in     std_logic                        -- FPGA clock
  );
  end component;
--------------------------------------------------------------------------------
```

Recommendation: after all components have been declared, a configuration statement should explicitly specify from what library and file each component should be instantiated

Signal declarations

Recommendation: do not end signal names with an integer.
Recommendation: signals should be grouped together**;** one signal per line, each declaration should be followed by a comment clarifying the nature and use of the signal, unless this is obvious.

```
-- EXAMPLE:
--------------------------------------------------------------------------------
  signal   VMEADH:   std_logic;                         -- VME base address high
  signal   VMEADL:   std_logic;                         -- VME base address low
  signal   VMEAM:    std_logic_vector(5 downto 0);      -- VME address modifier
  signal   RESET:    std_logic;                         -- internal FPGA reset
--------------------------------------------------------------------------------
```

Recommendation: where possible, the same name for a signal throughout all levels of a design should be used. In cases where exactly the same name cannot be used, for example when two identical sub-components have been instantiated, use names derived from the same root.

Component instances port maps

Rule: named association (rather than positional association) must be used to define the port maps of component instances.
Recommendation: when instantiating a component, the instance name should be descriptive.

```
-- EXAMPLE:
--------------------------------------------------------------------------------
  -- VME interface port map
  VME_INTERFACE: VMEIF
    port map (
    DATA      => VMEDATA,
    RESET_N   => RESET_N,                               -- FPGA reset pin
    CLK       => FPGA_CLK
    );
--------------------------------------------------------------------------------
```

Procceses

Rule: all processes must be named.
Recommendation: there should be some description for each "process" block in the architecture

```
-- EXAMPLE:
--------------------------------------------------------------------------------
  ------------------------------------------------------------------------------
  -- VME signals register
  ------------------------------------------------------------------------------
  VME_SIGNALS_REGISTERS: process (CLK)
  begin
    if rising_edge(CLK) then
    -- ...
    end if;
  end process VME_SIGNALS_REGISTERS;
--------------------------------------------------------------------------------
```

Combinatorial expressions

Rule: brackets must be used to enhance the readability of algebraic and Boolean equations.

```
-- EXAMPLE:
--------------------------------------------------------------------------------
  e := (a * b) + (c * d)                                -- variable calculation
--------------------------------------------------------------------------------
```

Packages

Rule: a package declaration must contain full documentation (in comments) about the declared types, constants, subprograms etc..
Recommendation: do not use names such as my_package — everybody will have a different version of my_package, which will create problems when sharing code