VHDL Architecture

Last updated 1/1/25

VHDL Architecture

- The VHDL architecture is the description of the block's functionality
  - Type
    - Identifies the kind of architecture described
    - User defined: Common types:
      - behavioral
      - structural
      - testbench
  - Declarations
    - Internal signal declarations
    - Included hierarchical block descriptions
  - Functional Description
    - Logic operations
    - Instantiated blocks with connections

VHDL Architecture

- Format - Block

```
-- Architecture definition
architecture behavioral of block_name is

begin

end architecture;
```

block_name must match the file name and entity

common types are
- behavioral
- structural
- testbench

VHDL Architecture

- Format - Declarations

```
-- Architecture definition
architecture behavioral of block_name is
  signal sig_name1 std_logic;
  signal sig_name2 std_logic_vector(7 downto 0);

  component component_name is
      generic(…
              )
      port(…
           );
  end component;

begin

end architecture;
```

declare any internal signals

component description for any structural elements – matches the entity for that component

VHDL Architecture

- Format – Definitions / Instantiations

```
-- Architecture definition
architecture behavioral of block_name is
   …
begin
      sigZ <= sigX OR sigY;
      sigC <= sigA when sigD = '0' else
                  sigB when sigD = '1';

      inst_0:   component_name
          generic map(  …
                              )
          port map(…
                             );

end architecture;
```

behavioral / logic descriptions

instantiation of any structural blocks

VHDL Architecture

- Example

Declarations

```
architecture logic of oddeven_16bit_logic_structural is

   -- internal signals
   signal sig_15_12: std_logic;
   signal sig_11_8:  std_logic;
   signal sig_7_4:   std_logic;
   signal sig_3_0:   std_logic;

   component oddeven_4bit_logic is
   port( i_a:           in std_logic;
         i_b:           in std_logic;
         i_c:           in std_logic;
         i_d:           in std_logic;
         o_oddeven_out: out std_logic
      );
   end component;
```

Definitions / Instantiations

```
begin

   oe_1512: oddeven_4bit_logic
   port map(   i_a            => i_in(15),
               i_b            => i_in(14),
               i_c            => i_in(13),
               i_d            => i_in(12),
               o_oddeven_out  => sig_15_12
            );

   oe_1108: oddeven_4bit_logic
   port map(   i_a            => i_in(11),
               i_b            => i_in(10),
               i_c            => i_in(9),
               i_d            => i_in(8),
               o_oddeven_out  => sig_11_8
            );

   oe_0704: oddeven_4bit_logic
   port map(   i_a            => i_in(7),
               i_b            => i_in(6),
               i_c            => i_in(5),
               i_d            => i_in(4),
               o_oddeven_out  => sig_7_4
            );

   oe_0300: oddeven_4bit_logic
   port map(   i_a            => i_in(3),
               i_b            => i_in(2),
               i_c            => i_in(1),
               i_d            => i_in(0),
               o_oddeven_out  => sig_3_0
            );

   oe_final: oddeven_4bit_logic
   port map(   i_a            => NOT sig_15_12,
               i_b            => NOT sig_11_8,
               i_c            => NOT sig_7_4,
               i_d            => NOT sig_3_0,
               o_oddeven_out  => o_oddeven_out
            );

end architecture;
```