VHDL Architecture

Last updated 8/10/24

VHDL Architecture

- The VHDL architecture is the description of the block's functionality
  - Format: architecture type of blockname is
    - type is user defined and include:
      - behavioral, structural, testbench, …
  - 2 sections – separated by begin
  - Declarations
    - Internal signal declarations
    - Included hierarchical block descriptions
  - Functional Description
    - Logic operations
    - Instantiated blocks with connections

VHDL Architecture

- The VHDL architecture is the description of the block's functionality
  - Behavioral

```
-- Behavioral Architecture Definition
architecture behavioral of basic_vhdl is
    signal e:   std_logic;
    signal f:   std_logic;
    signal g:   std_logic;
    signal t:   unsigned(N-1 downto 0);
    signal u:   signed(N-1 downto 0);

    Begin

        e <= i_a and i_b;
        f <= i_c nor i_a;
        g <= e xor f;
        o_x <= g and not i_p(3);

        t <= unsigned(i_p xor i_q);

end architecture;
```

Declarations

Functional Definition

VHDL Architecture

- The VHDL architecture is the description of the block's functionality
  - Structural

Declarations

```
architecture structural of adder_4bit is
-----------------------------------
-- 1 bit full adder prototype
-----------------------------------
component full_adder is
    port(  i_a:      in  std_logic;
           i_b:      in  std_logic;
           i_cin:    in std_logic;
           o_s:      out std_logic;
           o_cout:   out std_logic
    );
end component;

signal co: STD_LOGIC_VECTOR(3 downto 0);
```

Functional Definition

```
begin

    add_0:  full_adder port map( i_a => i_A(0),
                                 i_b => i_B(0),
                                 i_cin => i_CIN,
                                 o_s => o_SUM(0),
                                 o_cout => co(0)
                                 );
    add_1:  full_adder port map( i_a => i_A(1),
                                 i_b => i_B(1),
                                 i_cin => co(0),
                                 o_s => o_SUM(1),
                                 o_cout => co(1)
                                 );
    add_2:  full_adder port map( i_a => i_A(2),
                                 i_b => i_B(2),
                                 i_cin => co(1),
                                 o_s => o_SUM(2),
                                 o_cout => co(2)
                                 );
    add_3:  full_adder port map( i_a => i_A(3),
                                 i_b => i_B(3),
                                 i_cin => co(2),
                                 o_s => o_SUM(3),
                                 o_cout => co(3)
                                 );


    o_COUT <= co(3);

end architecture;
```

Instantiations