3. VHDL Modularity

3.1 VHDL Modularity Explanation

Example: **4-bit Ripple Carry Adder** using 4 Full Adders. A Ripple Carry Adder adds binary numbers with a chained carry.

3.1.1 Stage 1: Full Adder

```
entity Full_Adder is
  port (
    A, B, Cin: in std_logic;
    Sum, Cout: out std_logic
  );
end entity Full_Adder;

architecture RTL of Full_Adder is
begin
  Sum <= (A xor B) xor Cin;
  Cout <= (A and B) or ((A xor B) and Cin);
end architecture;
```

3.1.2 Stage 2: 4-bit Ripple Carry Adder

```
entity Four_Bit_RCA is
  port (
    A, B: in std_logic_vector(3 downto 0);
    Sum: out std_logic_vector(3 downto 0);
    Cout: out std_logic
  );
end entity Four_Bit_RCA;

architecture RTL of Four_Bit_RCA is
  signal Carry: std_logic_vector(3 downto 0);
```

```
begin
  FA0: Full_Adder port map (A(0), B(0), '0', Sum(0), Carry(0));
  FA1: Full_Adder port map (A(1), B(1), Carry(0), Sum(1), Carry(1));
  FA2: Full_Adder port map (A(2), B(2), Carry(1), Sum(2), Carry(2));
  FA3: Full_Adder port map (A(3), B(3), Carry(2), Sum(3), Cout);
end architecture;
```

3.1.3 Stage 3: Testbench

```
entity RCA_tb is
end entity RCA_tb;

architecture RTL of RCA_tb is
  signal A, B, Sum: std_logic_vector(3 downto 0);
  signal Cout: std_logic;
  signal Clock: std_logic := '0';
  constant Clock_Period: time := 10 ns;

  component Four_Bit_RCA
    port (
      A, B: in std_logic_vector(3 downto 0);
      Sum: out std_logic_vector(3 downto 0);
      Cout: out std_logic
    );
  end component;
begin
  Clock_Process: process
  begin
    while now < 1000 ns loop
      Clock <= not Clock;
      wait for Clock_Period / 2;
    end loop;
    wait;
  end process;

  A <= "1101";
  B <= "0011";

  RCA: Four_Bit_RCA port map (A, B, Sum, Cout);
```

```
  process
  begin
    wait until rising_edge(Clock);
    if Cout = '1' then
      report "The addition result is overflowing";
    end if;
    report "Addition result: " & to_string(Sum);
    wait;
  end process;
end architecture RTL; -- Corrected from Main_Architecture to RTL
```

Revision #2
Created 2025-09-24 12:59:34 UTC by AX
Updated 2025-09-29 01:50:13 UTC by AX