

# Structural Style, Port Mapping, and Generic Map

## Structural Style Programming

Structural Style Programming in VHDL allows designers to build digital circuits using basic components connected to form a more complex system. In this approach, circuits are represented as collections of entities connected in a specific way to achieve the desired function.

## Port Mapping

Port mapping is the process of associating the ports of a VHDL component (entity) with signals in the architecture. This allows entities to be connected with the actual circuit in the design.

## Important Points

- **Entity Definition:** An entity must be defined first, which includes its ports and data types.
- **Port-Map List:** Ports of the entity are mapped to corresponding signals in the architecture.
- **Port Mapping Order:** Mapping must follow the order defined in the entity.
- **Signal Declaration:** Signals used in port mapping must be declared in the architecture.

## Example

```
entity AND2 is
  port (
    A, B: in std_logic;
    Y: out std_logic
  );
end entity;

-- Port mapping
architecture RTL of AND2 is
```

```

begin
    Y <= A and B;
end architecture;

-- Using the entity with port mapping
D1: AND2 port map (
    A => input_signal_A,
    B => input_signal_B,
    Y => output_signal
);

```

## Generic Map

Generic map is the process of mapping generic values in an entity to corresponding values in the architecture. Generics are parameters that can be set for an entity to configure the behavior or characteristics of a component.

## Important Points

- **Generic:** Parameters used to modify characteristics of an entity.
- **Generic Map:** Defines the values for generics when instantiating an entity.
- **Default Value:** Generics often have default values, but they can be overwritten during instantiation.

## Example

```

entity Counter is
    generic (
        WIDTH: positive := 8; -- Default value for WIDTH is 8
        ENABLED: boolean := true -- Default value for ENABLED is true
    );
    port (
        clk: in std_logic;
        reset: in std_logic;
        count: out std_logic_vector(WIDTH-1 downto 0)
    );
end entity;

-- Generic map when instantiating the entity
architecture RTL of MyDesign is
    signal my_counter_output: std_logic_vector(7 downto 0);

```

```
begin
  my_counter_inst: Counter
    generic map (
      WIDTH => 8, -- Generic value WIDTH is reset to 8
      ENABLED => true -- Generic value ENABLED is reset to true
    )
    port map (
      clk => system_clock,
      reset => reset_signal,
      count => my_counter_output
    );
end architecture;
```

---

Revision #2

Created 23 September 2024 16:02:15 by GI

Updated 23 September 2024 16:03:15 by GI