Digital Systems

Unresolved and resolved types

R. Pacalet

Telecom Paris

2020-03-19

Unresolved and resolved types

What are they?

- VHDL types can be *unresolved* or *resolved*
  - `std.standard.bit`: unresolved
  - `ieee.std_logic_1164.std_ulogic`: unresolved
  - `ieee.std_logic_1164.std_logic`: resolved
- Signal of unresolved type: only one driving process
- Signal of resolved type: one or more driving processes

When to use them?

Resolved types

- Model wires driven by more than one hardware circuit
    - Bi-directional memory data bus
    - CPU drives for writing
    - Memory drives for reading

Unresolved types

- In all other circumstances

Two processes driving the same signal of type bit

```
-- File md.vhd
entity md is
end entity md;

architecture arc of md is
  signal s: bit;
begin
  p1: process
  begin
    s <= '0';
    wait;
  end process p1;
  p2: process
  begin
    s <= '0';
    wait;
  end process p2;
end architecture arc;
```

Type bit unresolved

- Driving s from p1 and p2 forbidden
- This does not depend on the driven values (same value here)
- Compilation or elaboration raise error
- Designer's mistake
- VHDL equivalent of short-circuit
- This error protects designers
- It is beneficial!

GHDL simulation

Compile, elaborate, simulate

```
$ ghdl -a md.vhd
$ ghdl -e md
$ ./md
for signal: .md(arc).s
./md:error: several sources for unresolved signal
./md:error: error during elaboration
```

- Note that the error is raised even if, as in our example, all drivers agree on the driving value.

Resolution functions

What is it?

- Resolved type ⇒ *resolution function*
- Signal driven by more than one VHDL process ⇒ resolution function called to compute resulting value
- Resolution function
  - Pure function
  - One parameter: vector of values (e.g. `bit_vector` to resolve `bit`)
  - Returns value of type to resolve.

Example of resolution functions (1/2)

```
architecture arc of md is
  function and_resolve_bit(d: bit_vector) return bit is
    variable r: bit := '1';
  begin
    for i in d'range loop
      if d(i) = '0' then
        r := '0';
      end if;
    end loop;
    return r;
  end function and_resolve_bit;
  subtype res_bit is and_resolve_bit bit;
  signal s: res_bit;
begin
  ...
end architecture arc;
```

Example of resolution functions (2/2)

```
p1: process
begin
  s <= '0', '1' after 1 ns, '0' after 2 ns, '1' after 3 ns;
  wait;
end process p1;
p2: process
begin
  s <= '0', '1' after 2 ns;
  wait;
end process p2;
p3: process(s)
begin
  report bit'image(s); -- show value changes
end process p3;
```

GHDL simulation

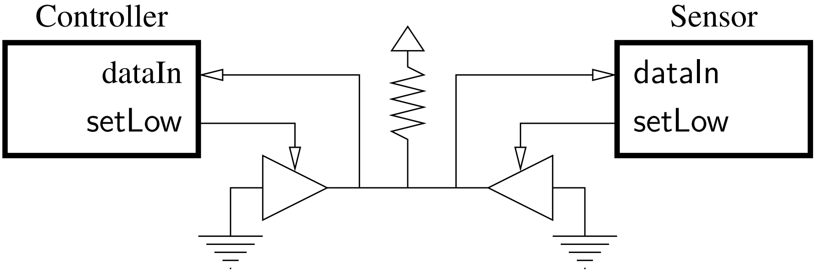
Compile, elaborate, simulate

```
$ ghdl -a md.vhd
$ ghdl -e md
$ ./md
md.vhd:39:5:@0ms:(report note): '0'
md.vhd:39:5:@3ns:(report note): '1'
```

A one-bit communication protocol

Single bi-directional data line

- Used in low-cost devices (e.g; sensors)
- Connects device to controller
- Frequently pulled up by pull-up resistor
- Communicating devices drive line low
- Duration of low state carries information



Example VHDL model of one-bit protocol

```
architecture arc of one_bit_protocol is
  signal data:           std_logic;  -- The bi-directional data line
  signal set_low_ctrl:     std_ulogic;
  signal set_low_sensor: std_ulogic;
begin
  u0: entity work.ctrl port map( -- Controller
    data_in => data,
    set_low => set_low_ctrl);
  u1: entity work.sensor port map( -- Sensor
    data_in => data,
    set_low => set_low_sensor);
  data <= 'H'; -- Pull-up resistor
  -- Controller tri-states buffer
  data <= '0' when set_low_ctrl = '1' else 'Z';
  -- Sensor tri-states buffer
  data <= '0' when set_low_sensor = '1' else 'Z';
end architecture arc;
```

The ieee.numeric_std package

- Declares signed and unsigned vector types
- Overloads arithmetic operators on them
- Frequently used when arithmetic and bit-wise operations needed on same data
- signed and unsigned are... resolved
- Using signed and unsigned ⇒
  - No error on accidental multiple drive situations
- VHDL2008 adds unresolved_signed and unresolved_unsigned (aliases u_signed and u_unsigned)
  - Should be preferred when multiple drive situations not desired...
  - ... and 2008 version of VHDL standard supported