Digital Systems - 0

Pere Palà - Alexis López

iTIC http://itic.cat

February 2016

Introduction

- ▶ VHDL: VHSIC Hardware Description Language
  - ▶ VHSIC: Very High Speed Integrated Circuit
- ▶ IEEE Standard (Institute of Electrical and Electronic Engineers)
- ▶ VHDL-87, **VHDL-93**, VHDL-2002

std_logic_1164

- Standardized package : `std_logic_1164`

```vhdl
type std_ulogic is ( 'U',-- Uninitialized
                     'X',-- Forcing Unknown
                     '0',-- Forcing zero
                     '1',-- Forcing one
                     'Z',-- High Impedance
                     'W',-- Weak Unknown
                     'L',-- Weak zero
                     'H',-- Weak one
                     '-' );-- Don't care
```

- This is used in almost any VHDL file

```vhdl
library ieee;
use ieee.std_logic_1164.all;
```

Example: AND gate

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity and_gate is
   port( a, b : in  std_logic;
         y    : out std_logic);
end and_gate;

architecture logic_and of and_gate is
begin
   y <= a and b;
end;
```

- ▶ entity: connections to the outside world
- ▶ architecture: what it does

Identifiers

- Case insensitive: `AND` is the same as `aNd`
- Reserverd words
    - `entity`, `or`, `and`, `register`, `begin`, ... The editor usually highlights them!
- **Only** alphabetic letters ('Aa' to 'Zz'), decimal digits ('0' to '9') and the underscore character ('_')
- **Must** start with an alphabetic letter
- May **not end** with an underscore character
- May not include **two** successive underscore characters

Example: Full Adder

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity full_adder is
   port( a, b, c_in : in  std_logic;
         s, c_out   : out std_logic);
end full_adder;

architecture arch_1 of full_adder is
   signal temp : std_logic;
begin
   temp  <= a xor b;
   s     <= temp xor c_in;
   c_out <= (a and b) or (c_in and temp);
end;
```

All assignments are concurrent! This is exactly the same:

```vhdl
   s     <= temp xor c_in;
   c_out <= (a and b) or (c_in and temp);
   temp  <= a xor b;
```

Testing

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity full_adder_tb is
end full_adder_tb;

architecture behav of
              full_adder_tb is
  component my_adder
  port(a,
       b,
       c_in : in std_logic;
       s,
       c_out: out std_logic);
    end component ;
    for dut : my_adder use
       entity work.full_adder;

signal t_a,t_b,t_c_in,
       t_s,t_c_out:std_logic;
```

```vhdl
begin
dut : my_adder port map
          (a       => t_a,
           b       => t_b,
           c_in    => t_c_in,
           s       => t_s,
           c_out => t_c_out);
process
   begin
   t_a     <= '0';
   t_b     <= '0';
   t_c_in <= '0';
   wait for 1 sec;
   t_a     <= '0';
   t_b     <= '1';
   t_c_in <= '0';
   wait for 1 sec;
   wait;
end process ;
end behav ;
```

Testing/2

- ▶ `$ ghdl -a full_adder.vhd`
- ▶ `$ ghdl -a full_adder_tb.vhd`
- ▶ `$ ghdl -e full_adder_tb`
- ▶ `$ ghdl -r full_adder_tb --vcd=full_adder_tb.vcd`
- ▶ `$ gtkwave full_adder_tb.vcd &`