VHDL Signals

Last updated 12/17/24

VHDL Signals

This is not a programming language
It is a hardware description language

VHDL Signals

- VHDL elements fall into 2 categories
  - Synthesizable – types that can be used to create hardware
  - Un-synthesizable – types that cannot be used to create hardware
    - These exist for simulation, test and documentation

- The basic VHDL library is invoked in 2 steps

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

VHDL Signals

- A VHDL signal value represents a logic value
  - There are 3 values for a VHDL signal that can be synthesized

Value	Definition	Synthesizable
'0'	Forcing 0	Y
'1'	Forcing 1	Y
'Z'	High Impedance	Y

  - There are 2 additional values for a VHDL signal used in simulations

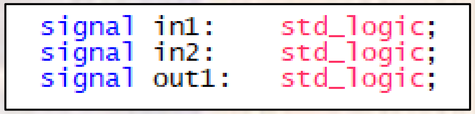
Value	Definition	Synthesizable
'U'	Un-initialized	N
'X'	Unknown	N

additional values exist – but are out of the scope of this class

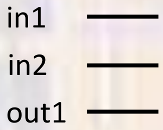
VHDL Signals

- A VHDL signal represents a wire
  - A single wire is identified as a std_logic signal

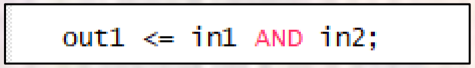
Declaration



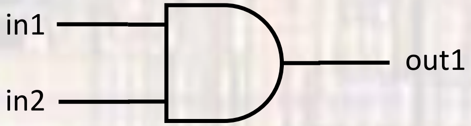
```
signal in1:     std_logic;
signal in2:     std_logic;
signal out1:    std_logic;
```



Behavior



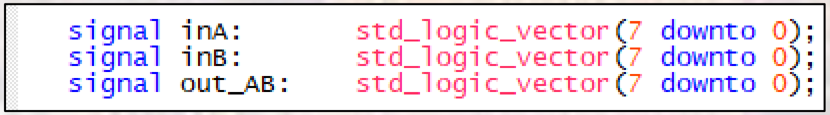
```
out1 <= in1 AND in2;
```



VHDL Signals

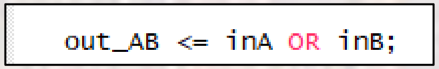
- A VHDL signal represents a wire
  - A bus (collection of wires) is identified as a std_logic_vector signal
    - Includes the dimension of the bus – in wire order – 7 6 5 4 3 2 1 0
      - Format: std_logic_vector((n-1) downto 0)

Declaration

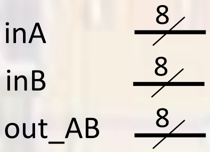


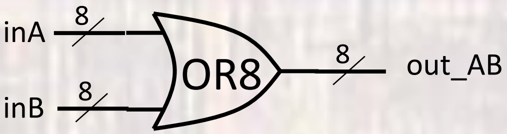
```
   signal inA:      std_logic_vector(7 downto 0);
   signal inB:      std_logic_vector(7 downto 0);
   signal out_AB:   std_logic_vector(7 downto 0);
```

Behavior



```
   out_AB <= inA OR inB;
```



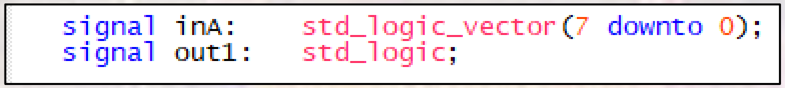


After a SLV signal has been declared – we refer to the entire signal by its name

VHDL Signals

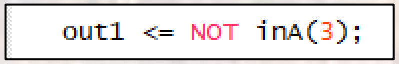
- Individual wires in a bus can be identified
  - signal_name(wire number)

Declaration

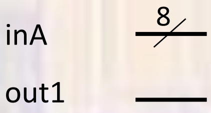


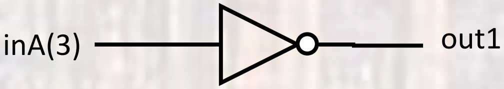
```
signal inA:    std_logic_vector(7 downto 0);
signal out1:   std_logic;
```

Behavior



```
out1 <= NOT inA(3);
```

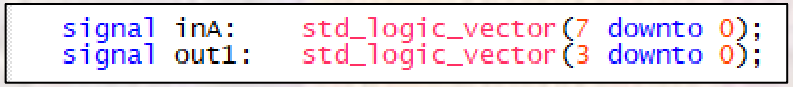




VHDL Signals

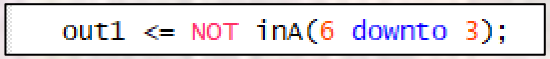
- Contiguous wires in a bus can be identified
  - signal_name(wire range)
    - Format: signal_name(n downto m)

Declaration

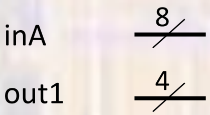


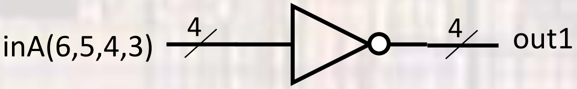
```
signal inA:    std_logic_vector(7 downto 0);
signal out1:   std_logic_vector(3 downto 0);
```

Behavior



```
out1 <= NOT inA(6 downto 3);
```

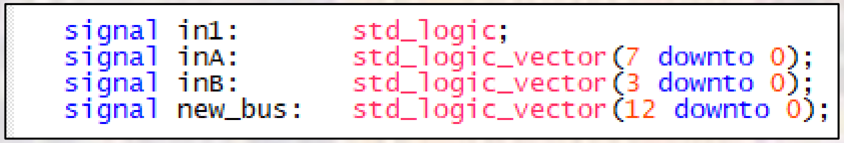




VHDL Signals

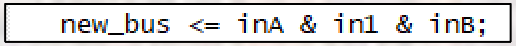
- Wires and buses can be combined to create a new signal (bus) name
  - No new wires are created – just connected to an additional name
  - Concatenation operator &

Declaration

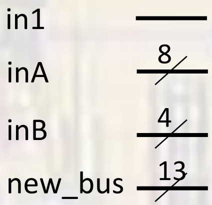


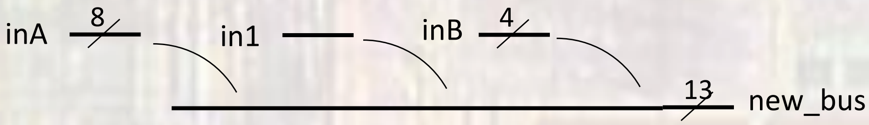
```
signal in1:       std_logic;
signal inA:       std_logic_vector(7 downto 0);
signal inB:       std_logic_vector(3 downto 0);
signal new_bus:   std_logic_vector(12 downto 0);
```

Behavior



```
new_bus <= inA & in1 & inB;
```





new_bus: inA(7), inA(6), … inA(0), in1, inB(3), …inB(0)