Structural VHDL

Last updated 8/19/24

Structural VHDL

- Structural VHDL implements hierarchical designs
  - Instantiates blocks into a more complex design
  - Complex Designs
  - Test Benches
  - Can include behavioral VHDL content

Structural VHDL

- Structural VHDL file components
  - Header
    - Description of who created the file
    - Description of the purpose of the block
    - Description of inputs and outputs
  - Inclusions
    - Any libraries that will be referenced in the design
  - Entity
    - Formal definition of the inputs and outputs
    - Any generic parameters are defined here also
  - Architecture
    - Internal signal declarations
      - Required for connecting structural blocks
    - Component declarations for hierarchical blocks used in the design
    - Instantiation (hook-up) of any hierarchical blocks used in the design
    - HDL description of the desired additional functionality
      - Concurrent, sequential, structural logic

Structural VHDL

- Structural VHDL file components - example

```
----------------------------------
--
-- adder_4bit.vhdl
--
-- by: tj
--
-- created: 7/5/2017
--
-- version: 0.0
--
----------------------------------
--
-- 4 bit adder to show cell instantiation
--
-- inputs: - a, b, cin
--
-- outputs: - sum, cout
--
----------------------------------
```

Header Information

Document
who, when, what and how

Should be sufficient to allow a co-worker to use/modify the design

Structural VHDL

- Structural VHDL file components - example

```
----------------------------------
--
-- adder_4bit.vhdl
--
-- by: tj
--
-- created: 7/5/2017
--
-- version: 0.0
--
----------------------------------
--
-- 4 bit adder to show cell instantiation
--
-- inputs: - a, b, cin
--
-- outputs: - sum, cout
--
----------------------------------
```

Header Information

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

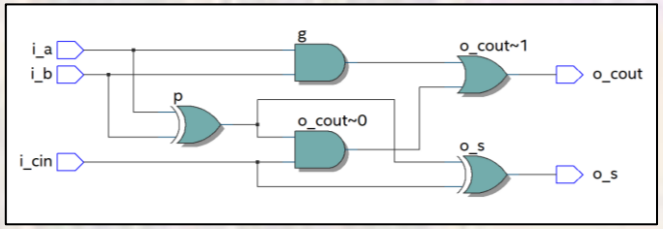
Library Inclusions

Companies may have their own libraries
- we will use the IEEE standard library

Only include the portions of the library you need
- ieee.std_logic_1164.all provides:
  - std_logic
  - std_logic_vector
  - 9 logic levels
- ieee.numeric_std.all provides
  - Signed vectors – signed, unsigned

Structural VHDL

- Structural VHDL file components - example
  - Explicitly build the logic from hierarchical blocks
  - Example
    - 1-bit Full Adder block




```
--------------------------------
--
-- full_adder.vhdl
--
-- by: johnsontimoj
--
-- created: 7/5/2017
--
-- version: 0.0
--
--------------------------------
--------------------------------
--
-- 1 bit full adder
--
-- inputs: a, b, cin
--
-- outputs: s, cout
--
--------------------------------

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

```
entity full_adder is
    port( i_a:      in  std_logic;
          i_b:      in  std_logic;
          i_cin:    in std_logic;
          o_s:      out std_logic;
          o_cout:   out std_logic
    );
end entity;

architecture behavioral of full_adder is

    signal p: std_logic;
    signal g: std_logic;

begin

    p <= i_a xor i_b;
    g <= i_a and i_b;

    o_s <= p xor i_cin;
    o_cout <= g or (p and i_cin);

end;
```

Structural VHDL

- Structural VHDL file comp
  - 4 bit adder example

```
----------------------------------
--
--  adder_4bit.vhdl
--
--  by: tj
--
--  created: 7/5/2017
--
--  version: 0.0
--
----------------------------------
--
--  4 bit adder to show cell instantiation
--
--  inputs: - a, b, cin
--
--  outputs: - sum, cout
--
----------------------------------
library IEEE;
use ieee.std_logic_1164.all;

entity adder_4bit is
    port(   i_A:    in  std_logic_vector(3 downto 0);
            i_B:    in std_logic_vector(3 downto 0);
            i_CIN:      in std_logic;
            o_SUM:      out std_logic_vector(3 downto 0);
            o_COUT:     out std_logic
    );
end entity;

architecture structural of adder_4bit is
```

Top Level Entity

```
----------------------------------
--  1 bit full adder prototype
----------------------------------
component full_adder is
    port(   i_a:        in  std_logic;
            i_b:        in  std_logic;
            i_cin:      in std_logic;
            o_s:        out std_logic;
            o_cout:     out std_logic
    );
end component;
----------------------------------
-- intermediate carrys mapped to co
-- with 1st stage Cout mapped to co(0) and 4th stage cout mapped to co(3)
----------------------------------
signal co: STD_LOGIC_VECTOR(3 downto 0); -- intermediate carrys
                                         -- no vector interpretation
```

Component Prototype

```
begin

    add_0:  full_adder port map(  i_a => i_A(0),
                                  i_b => i_B(0),
                                  i_cin => i_CIN,
                                  o_s => o_SUM(0),
                                  o_cout => co(0)
                                  );
    add_1:  full_adder port map(  i_a => i_A(1),
                                  i_b => i_B(1),
                                  i_cin => co(0),
                                  o_s => o_SUM(1),
                                  o_cout => co(1)
                                  );
    add_2:  full_adder port map(  i_a => i_A(2),
                                  i_b => i_B(2),
                                  i_cin => co(1),
                                  o_s => o_SUM(2),
                                  o_cout => co(2)
                                  );
    add_3:  full_adder port map(  i_a => i_A(3),
                                  i_b => i_B(3),
                                  i_cin => co(2),
                                  o_s => o_SUM(3),
                                  o_cout => co(3)
                                  );


    o_COUT <= co(3);

end architecture;
```

Instantiations

Explicit Port Mapping

Structural VHDL

- Structural VHDL file components - example
  - 4 bit adder example