Structural VHDL

Last updated 12/18/24

Structural VHDL

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

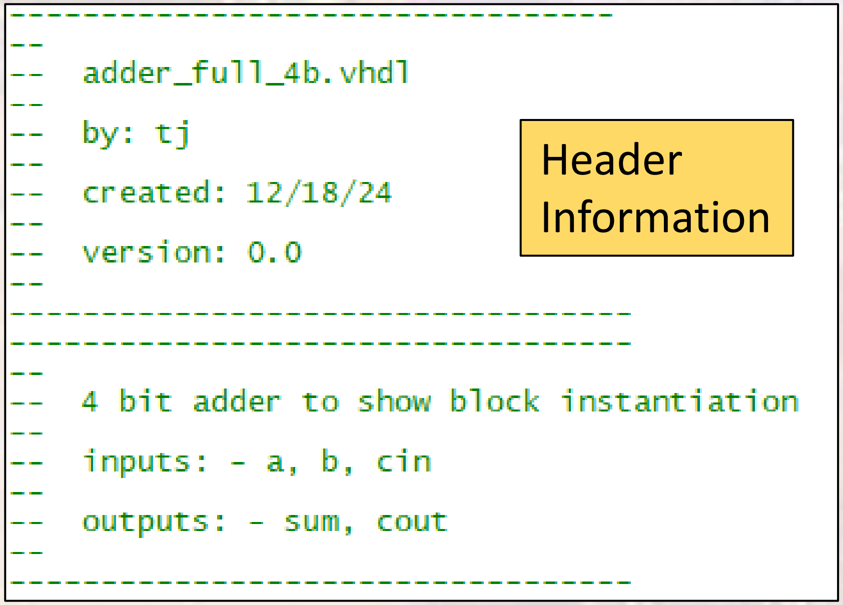
Structural VHDL

- 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
      - Including those 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

comments begin with 2 dashes
--
continue to the end of the current line



```
-------------------------------------
--
--   adder_full_4b.vhdl
--
--   by: tj
--
--   created: 12/18/24
--
--   version: 0.0
--
--------------------------------------
--------------------------------------
--
--   4 bit adder to show block instantiation
--
--   inputs: - a, b, cin
--
--   outputs: - sum, cout
--
--------------------------------------
```

Header Information

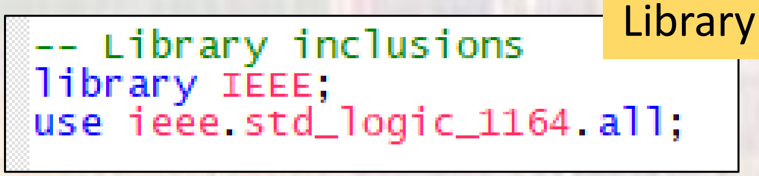
Document
who, when, what and how

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

Structural VHDL

- Structural VHDL file components - example

Library inclusions



```
-- Library inclusions
library IEEE;
use ieee.std_logic_1164.all;
```

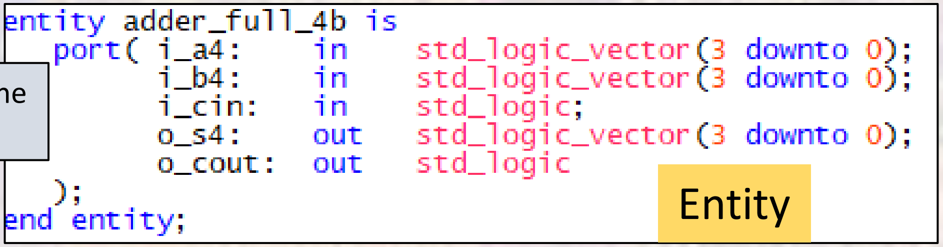
statements end with a semi-colon

Include the basic library definitions

Structural VHDL

- Structural VHDL file components - example

the entity name must match the file name



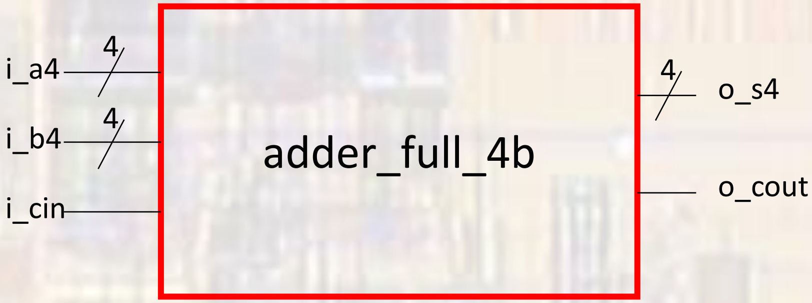
```
entity adder_full_4b is
   port( i_a4:    in    std_logic_vector(3 downto 0);
         i_b4:    in    std_logic_vector(3 downto 0);
         i_cin:   in    std_logic;
         o_s4:    out   std_logic_vector(3 downto 0);
         o_cout:  out   std_logic
   );
end entity;
```

Entity

port format:
name : direction signal_type ;

directions: in, out, bidir
signal types: std_logic
std_logic_vector(…)

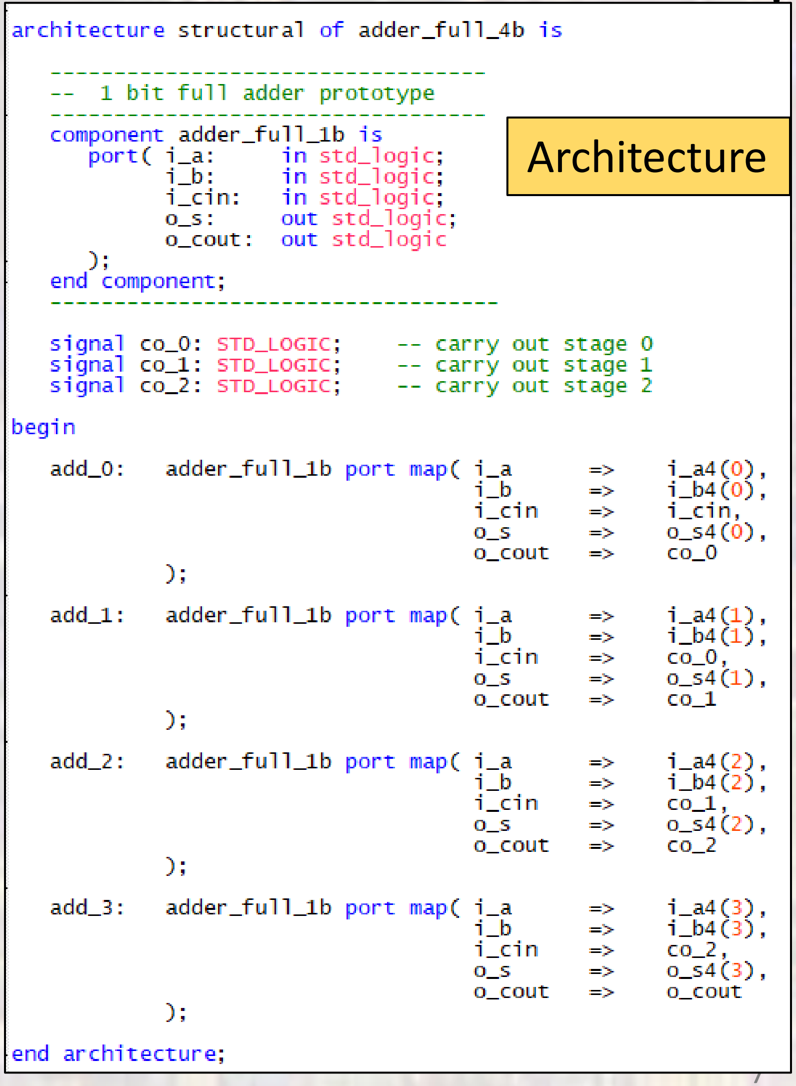
We will prepend ports with
i_ or o_



The Entity is the external view of the block
Ports are the i/o connections of the block
Ports have a direction but no interpretation
Wires or groups of wires

Structural VHDL

- Structural VHDL file components - example



```
architecture structural of adder_full_4b is

   ------------------------------------
   --  1 bit full adder prototype
   ------------------------------------
   component adder_full_1b 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;
   ------------------------------------

   signal co_0: STD_LOGIC;    -- carry out stage 0
   signal co_1: STD_LOGIC;    -- carry out stage 1
   signal co_2: STD_LOGIC;    -- carry out stage 2

begin
   add_0:   adder_full_1b port map( i_a      =>     i_a4(0),
                                    i_b      =>     i_b4(0),
                                    i_cin    =>     i_cin,
                                    o_s      =>     o_s4(0),
                                    o_cout   =>     co_0
            );

   add_1:   adder_full_1b port map( i_a      =>     i_a4(1),
                                    i_b      =>     i_b4(1),
                                    i_cin    =>     co_0,
                                    o_s      =>     o_s4(1),
                                    o_cout   =>     co_1
            );

   add_2:   adder_full_1b port map( i_a      =>     i_a4(2),
                                    i_b      =>     i_b4(2),
                                    i_cin    =>     co_1,
                                    o_s      =>     o_s4(2),
                                    o_cout   =>     co_2
            );

   add_3:   adder_full_1b port map( i_a      =>     i_a4(3),
                                    i_b      =>     i_b4(3),
                                    i_cin    =>     co_2,
                                    o_s      =>     o_s4(3),
                                    o_cout   =>     o_cout
            );

end architecture;
```

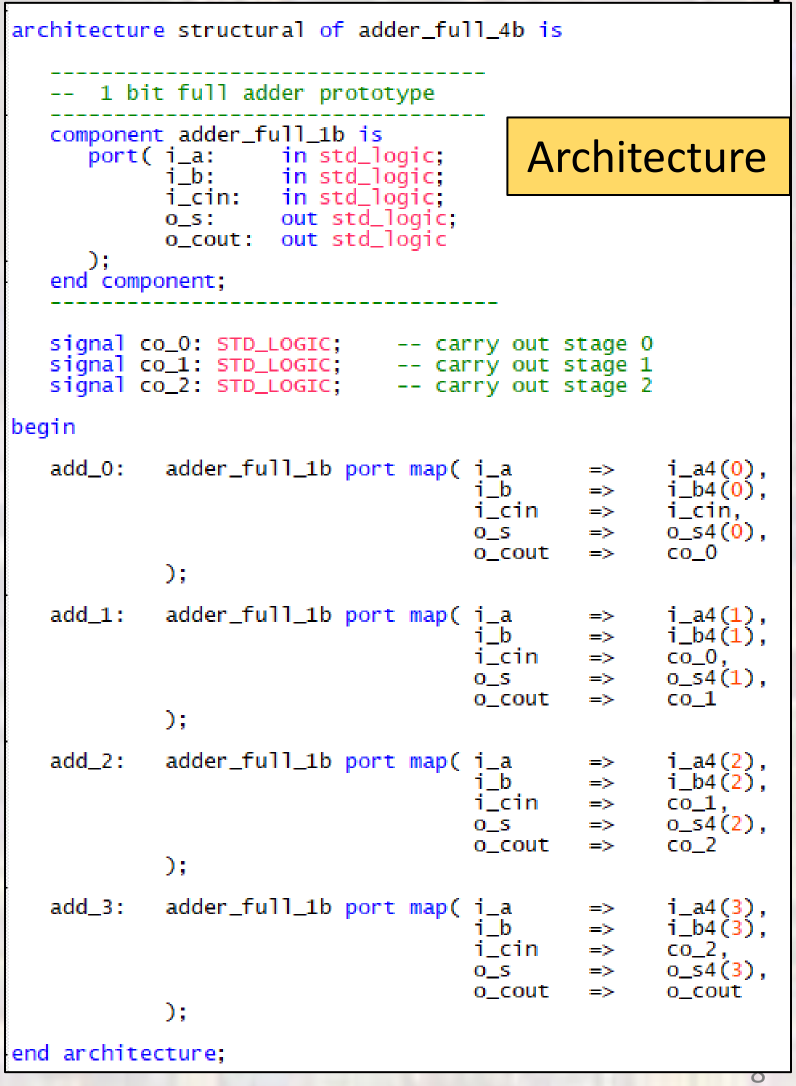
Architecture

Before the begin
    Block prototypes
    Internal signals
    Constants used for readability

After the begin
    Instantiation of blocks
        Use EXPLICITE port mapping
    HDL for the specific application

Structural VHDL

- Structural VHDL file components - example



```
architecture structural of adder_full_4b is

   -----------------------------------
   --  1 bit full adder prototype
   -----------------------------------
   component adder_full_1b 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;
   ------------------------------------

   signal co_0: STD_LOGIC;    -- carry out stage 0
   signal co_1: STD_LOGIC;    -- carry out stage 1
   signal co_2: STD_LOGIC;    -- carry out stage 2

begin

   add_0:   adder_full_1b port map( i_a      =>    i_a4(0),
                                    i_b      =>    i_b4(0),
                                    i_cin    =>    i_cin,
                                    o_s      =>    o_s4(0),
                                    o_cout   =>    co_0
            );

   add_1:   adder_full_1b port map( i_a      =>    i_a4(1),
                                    i_b      =>    i_b4(1),
                                    i_cin    =>    co_0,
                                    o_s      =>    o_s4(1),
                                    o_cout   =>    co_1
            );

   add_2:   adder_full_1b port map( i_a      =>    i_a4(2),
                                    i_b      =>    i_b4(2),
                                    i_cin    =>    co_1,
                                    o_s      =>    o_s4(2),
                                    o_cout   =>    co_2
            );

   add_3:   adder_full_1b port map( i_a      =>    i_a4(3),
                                    i_b      =>    i_b4(3),
                                    i_cin    =>    co_2,
                                    o_s      =>    o_s4(3),
                                    o_cout   =>    o_cout
            );

end architecture;
```

Architecture

Component prototype
The entity code for the desired block with entity replaced with component

signal format:
signal name : signal_type ;

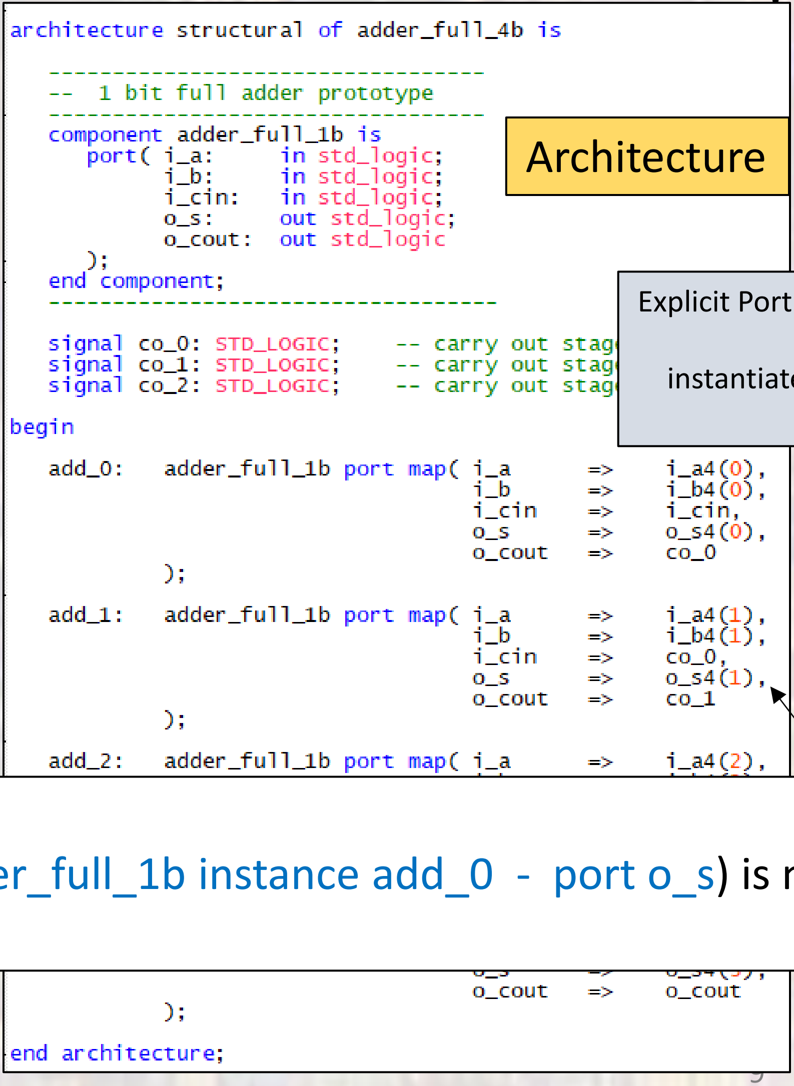
signal types: std_logic
std_logic_vector(…)

Component instantiation
One instantiation for each block to be placed in the design

format:
instance_name : block_name port map(
port maps
);

Structural VHDL

- Structural VHDL file components - example



```
architecture structural of adder_full_4b is

   ----------------------------------
   --  1 bit full adder prototype
   ----------------------------------
   component adder_full_1b 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;
   -----------------------------------

   signal co_0: STD_LOGIC;    -- carry out stag
   signal co_1: STD_LOGIC;    -- carry out stag
   signal co_2: STD_LOGIC;    -- carry out stag

begin

   add_0:   adder_full_1b port map( i_a      =>     i_a4(0),
                                    i_b      =>     i_b4(0),
                                    i_cin    =>     i_cin,
                                    o_s      =>     o_s4(0),
                                    o_cout   =>     co_0
            );

   add_1:   adder_full_1b port map( i_a      =>     i_a4(1),
                                    i_b      =>     i_b4(1),
                                    i_cin    =>     co_0,
                                    o_s      =>     o_s4(1),
                                    o_cout   =>     co_1
            );

   add_2:   adder_full_1b port map( i_a      =>     i_a4(2),

                                    o_s      =>     o_s4(3),
                                    o_cout   =>     o_cout
            );

end architecture;
```

Architecture

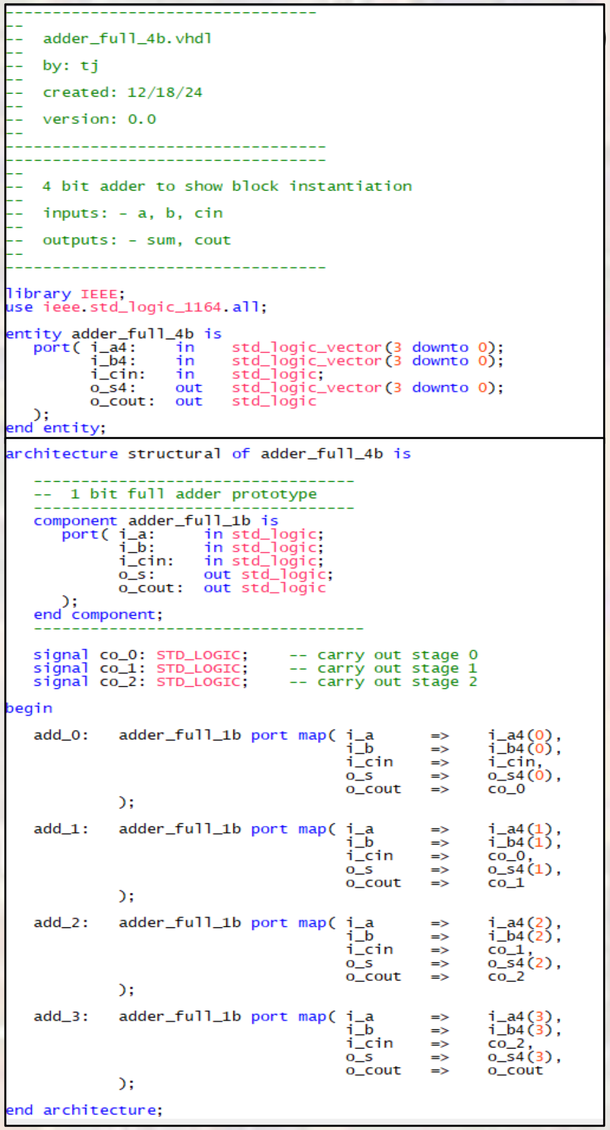
Explicit Port Mapping

instantiated block port name => structural design signal or port name ,

(adder_full_1b instance add_0 - port o_s) is mapped to (adder_full_4b - port o_s4 bit 1)

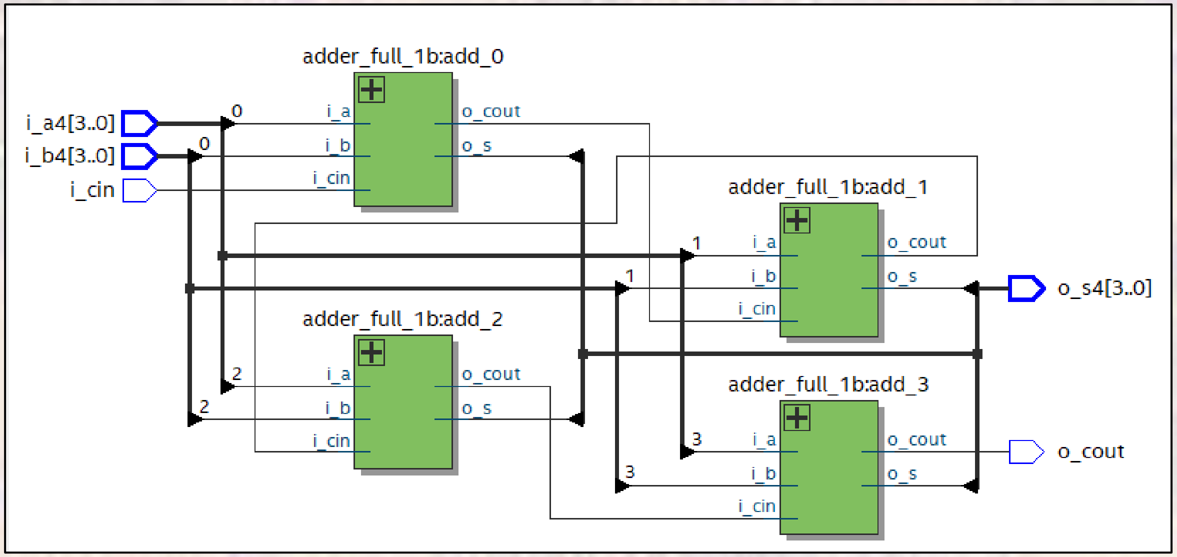
Structural VHDL

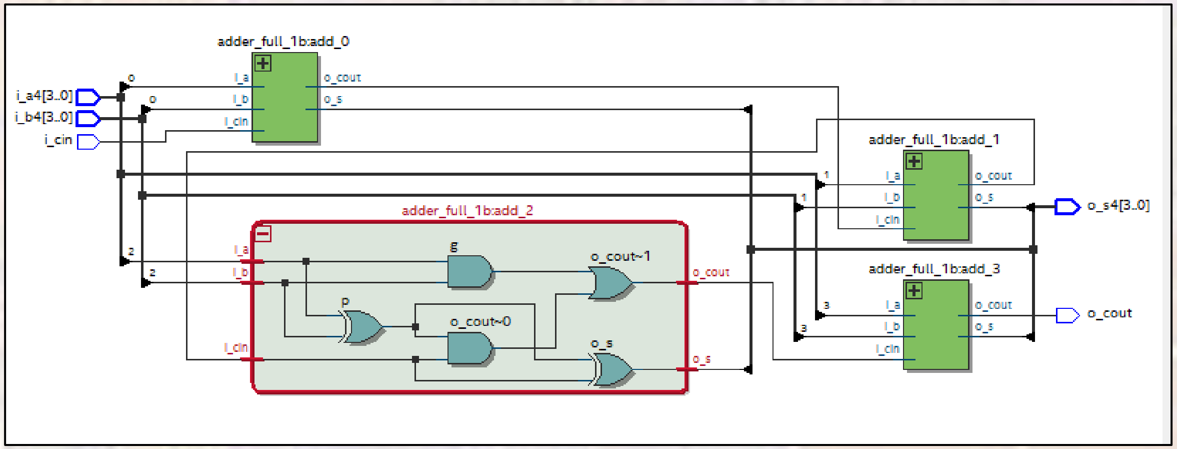
L file components - example



```
-----------------------------------
--
--  adder_full_4b.vhdl
--
--  by: tj
--
--  created: 12/18/24
--
--  version: 0.0
--
-------------------------------------
-------------------------------------
--
--  4 bit adder to show block instantiation
--
--  inputs: - a, b, cin
--
--  outputs: - sum, cout
--
-------------------------------------

library IEEE;
use ieee.std_logic_1164.all;

entity adder_full_4b is
   port( i_a4:    in    std_logic_vector(3 downto 0);
         i_b4:    in    std_logic_vector(3 downto 0);
         i_cin:   in    std_logic;
         o_s4:    out   std_logic_vector(3 downto 0);
         o_cout:  out   std_logic
   );
end entity;
```

```
architecture structural of adder_full_4b is

   -------------------------------------
   --  1 bit full adder prototype
   -------------------------------------
   component adder_full_1b 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;
   --------------------------------------

   signal co_0: STD_LOGIC;     -- carry out stage 0
   signal co_1: STD_LOGIC;     -- carry out stage 1
   signal co_2: STD_LOGIC;     -- carry out stage 2

begin

   add_0:   adder_full_1b port map( i_a      =>    i_a4(0),
                                    i_b      =>    i_b4(0),
                                    i_cin    =>    i_cin,
                                    o_s      =>    o_s4(0),
                                    o_cout   =>    co_0
            );

   add_1:   adder_full_1b port map( i_a      =>    i_a4(1),
                                    i_b      =>    i_b4(1),
                                    i_cin    =>    co_0,
                                    o_s      =>    o_s4(1),
                                    o_cout   =>    co_1
            );

   add_2:   adder_full_1b port map( i_a      =>    i_a4(2),
                                    i_b      =>    i_b4(2),
                                    i_cin    =>    co_1,
                                    o_s      =>    o_s4(2),
                                    o_cout   =>    co_2
            );

   add_3:   adder_full_1b port map( i_a      =>    i_a4(3),
                                    i_b      =>    i_b4(3),
                                    i_cin    =>    co_2,
                                    o_s      =>    o_s4(3),
                                    o_cout   =>    o_cout
            );

end architecture;
```


adder_full_1b:add_0
adder_full_1b:add_1
adder_full_1b:add_2
adder_full_1b:add_3
i_a4[3..0]
i_b4[3..0]
i_cin
i_a
i_b
i_cin
o_cout
o_s
0
1
2
3
o_s4[3..0]
o_cout


adder_full_1b:add_0
adder_full_1b:add_1
adder_full_1b:add_2
adder_full_1b:add_3
i_a4[3..0]
i_b4[3..0]
i_cin
i_a
i_b
i_cin
o_cout
o_s
g
p
o_cout~1
o_cout~0
o_s
0
1
2
3
o_s4[3..0]
o_cout