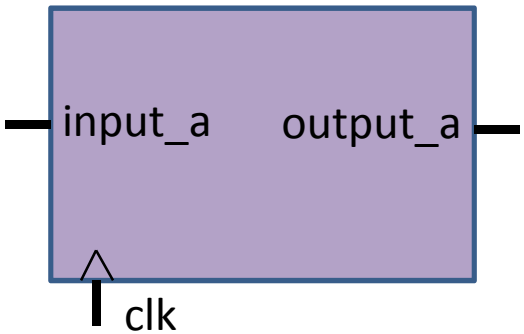
VHDL Portmapping Example

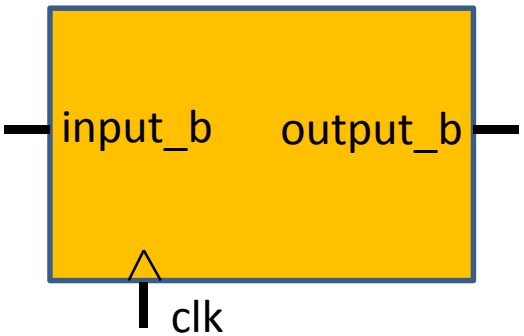
Given the following 2 modules:

module_a.vhd




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

entity module_a is
  port (
                clk:      in std_logic;
                input_a:  in std_logic;
                output_a: out std_logic
                );
end module_a;

architecture behav of module_a is
begin

output_a <= input_a;

end behav;
```

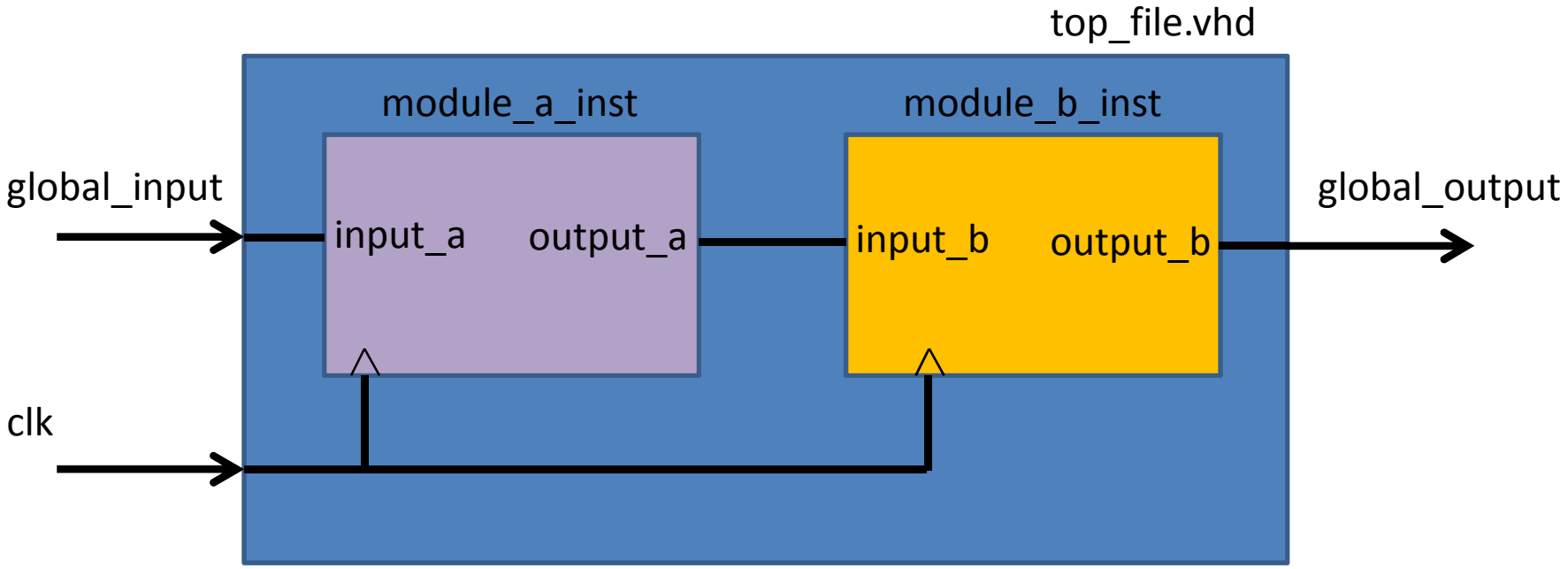
module_b.vhd




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

entity module_b is
  port (
                clk:      in std_logic;
                input_b:  in std_logic;
                output_b: out std_logic
                );
end module_b;

architecture behav of module_b is
begin

output_b <= not(input_b);

end behav;
```

VHDL Portmapping Example

Portmap them as follows:



VHDL Portmapping Example

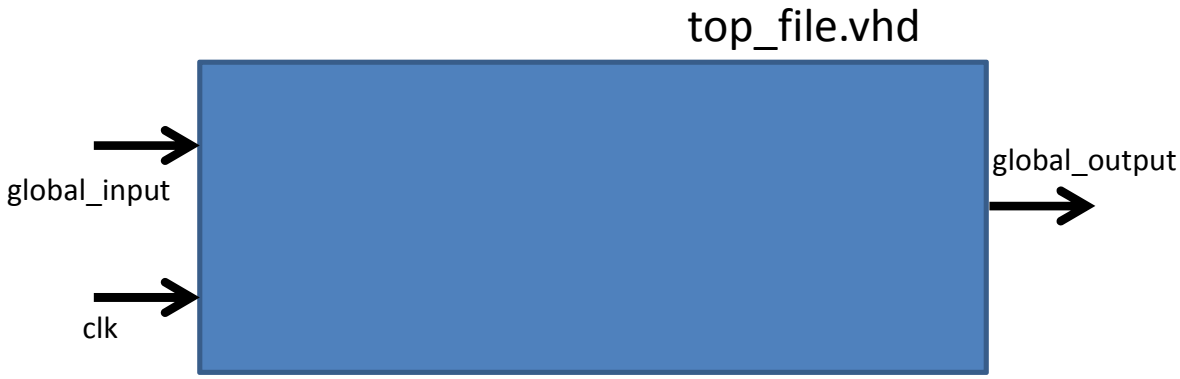
This is (usually) done in **four** steps:

1. Create empty top_file shell (only define ports)
2. Declare the modules
3. Define needed internal signals
4. Instantiate and portmap to connect the modules

VHDL Portmapping Example

1. Create empty top_file shell (only define ports)

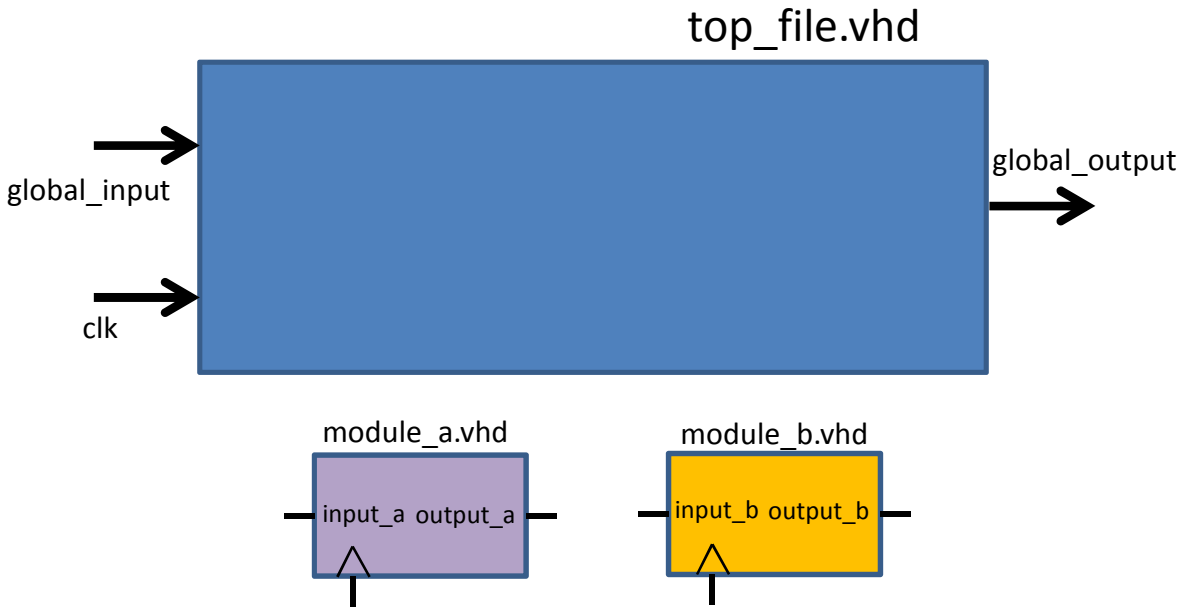
```
library IEEE;
use IEEE.std_logic_1164.all;

entity top_file is
  port (
                clk:          in std_logic;
                global_input:   in std_logic;
                global_output: out std_logic
                );
end top_file;

architecture behav of top_file is
begin









end behav;
```



VHDL Portmapping Example

2. Declare the modules

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

entity top_file is
  port (
                clk:          in std_logic;
                global_input:   in std_logic;
                global_output: out std_logic
                );
end top_file;

architecture behav of top_file is

component  module_a
   port(
                clk, input_a: in std_logic;
                output_a: out std_logic);
end component;

component  module_b
   port(
                clk, input_b: in std_logic;
                output_b: out std_logic);
end component;

begin




end behav;
```



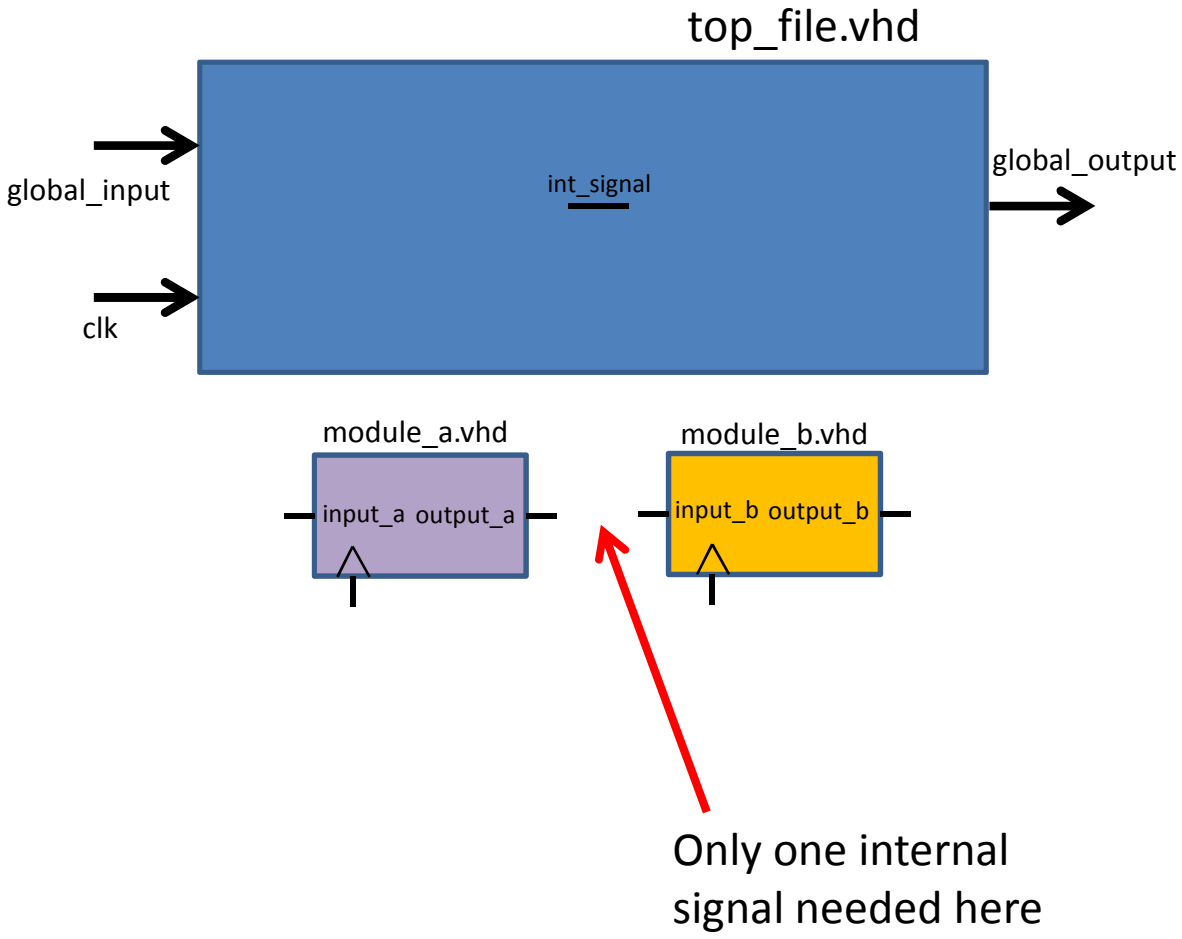
Same as entity in vhd file, only named component

Note: no instantiation of the modules exists yet!

VHDL Portmapping Example

3. Define needed internal signals

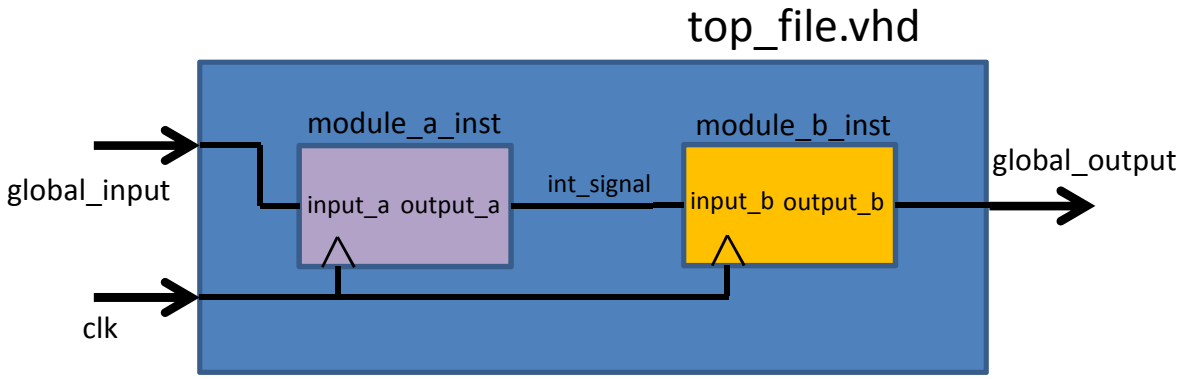
```
library IEEE;
use IEEE.std_logic_1164.all;

entity top_file is
  port (
                clk:          in std_logic;
                global_input:   in std_logic;
                global_output: out std_logic
                );
end top_file;

architecture behav of top_file is

component  module_a
   port(
                clk, input_a: in std_logic;
                output_a: out std_logic);
end component;

component  module_b
   port(
                clk, input_b: in std_logic;
                output_b: out std_logic);
end component;

signal int_signal: std_logic;

begin




end behav;
```



VHDL Portmapping Example

4. Instantiate and portmap to connect the modules

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

entity top_file is
  port (
                clk:          in std_logic;
                global_input:   in std_logic;
                global_output: out std_logic
                );
end top_file;

architecture behav of top_file is

component  module_a
   port(
                clk, input_a: in std_logic;
                output_a: out std_logic);
end component;

component  module_b
   port(
                clk, input_b: in std_logic;
                output_b: out std_logic);
end component;

signal int_signal: std_logic;

begin

end behav;
```



Note: multiple instantiations of same module possible!

```
module_a_inst:  module_a
port map(
                input_a => global_input,
                output_b => int_signal,
                clk => clk
);

module_b_inst:  module_b
port map(
                input_b => int_signal,
                output_b => global_output,
                clk => clk
);
```

Left-hand side:
component signals

Right-hand side:
top_file signals

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

entity top_file is
  port (
                clk:          in std_logic;
                global_input:   in std_logic;
                global_output: out std_logic
                );
end top_file;

architecture behav of top_file is

component  module_a
   port(
                clk, input_a: in std_logic;
                output_a: out std_logic);
end component;

component  module_b
   port(
                clk, input_b: in std_logic;
                output_b: out std_logic);
end component;

signal int_signal: std_logic;

begin

module_a_inst:  module_a
port map(
                input_a => global_input,
                output_b => int_signal,
                clk => clk
);

module_b_inst:  module_b
port map(
                input_b => int_signal,
                output_b => global_output,
                clk => clk
);
end behav;
```

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

entity module_a is
  port (
                clk:          in std_logic;
                input_a:   in std_logic;
                output_a: out std_logic
                );
end module_a;

architecture behav of module_a is
begin

output_a <= input_a;

end behav;
```

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

entity module_b is
  port (
                clk:          in std_logic;
                input_b:   in std_logic;
                output_b: out std_logic
                );
end module_b;

architecture behav of module_b is
begin

output_b <= not(input_b);

end behav;
```