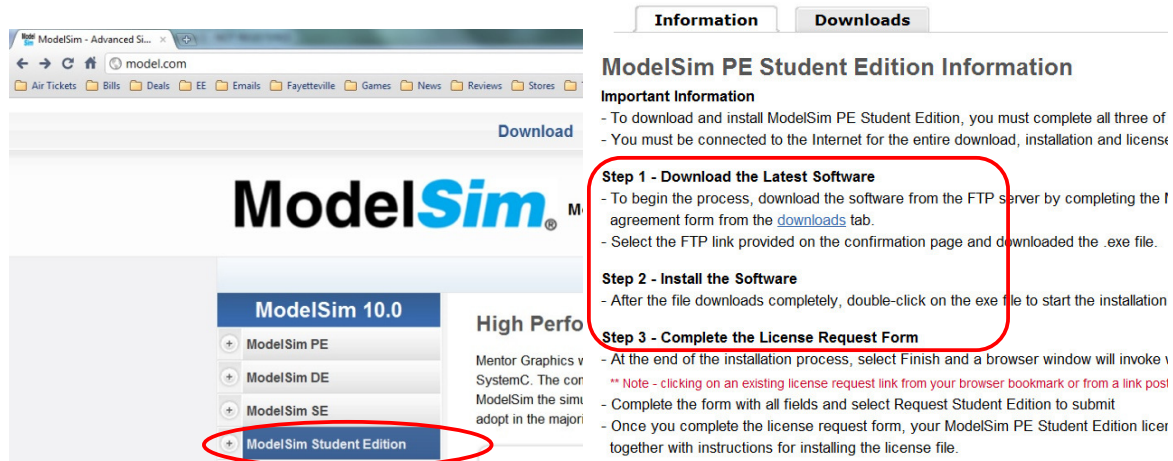


## HƯỚNG DẪN CƠ BẢN

### MÔI PHONG VHDL/VERILOG VỚI MODELSIM

#### Download và cài đặt chương trình

Download miễn phí chương trình ModelSim Student Edition (version 6.6d tại 12/2010) tại [www.model.com](http://www.model.com)



**ModelSim PE Student Edition Information**

**Important Information**

- To download and install ModelSim PE Student Edition, you must complete all three of the following steps.
- You must be connected to the Internet for the entire download, installation and license request process.

**Step 1 - Download the Latest Software**

- To begin the process, download the software from the FTP server by completing the license request form from the [downloads](#) tab.
- Select the FTP link provided on the confirmation page and download the .exe file.

**Step 2 - Install the Software**

- After the file downloads completely, double-click on the exe file to start the installation.

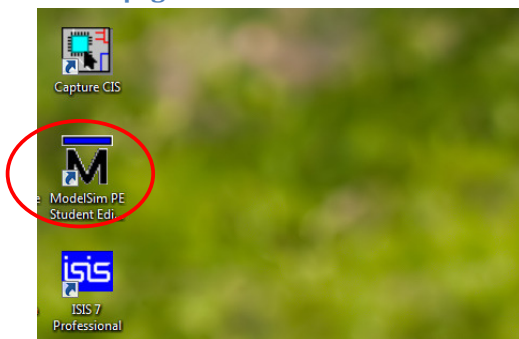
**Step 3 - Complete the License Request Form**

- At the end of the installation process, select Finish and a browser window will invoke the license request form.
- **Note** - clicking on an existing license request link from your browser bookmark or from a link post.
- Complete the form with all fields and select Request Student Edition to submit.
- Once you complete the license request form, your ModelSim PE Student Edition license file will be downloaded together with instructions for installing the license file.

Làm đúng theo 3 bước được hướng dẫn trên [www.model.com](http://www.model.com)

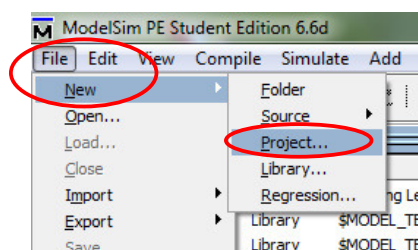
#### Hướng dẫn cơ bản

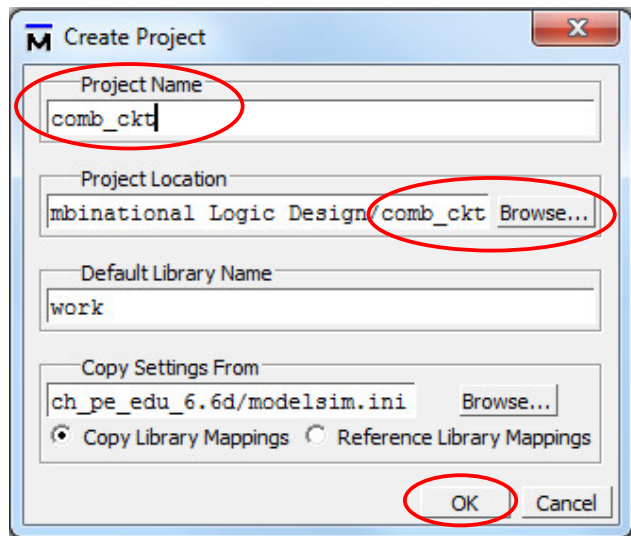
##### Khởi động ModelSim



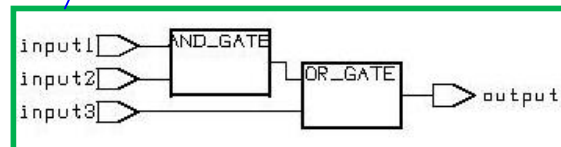
##### Tạo Project

Bấm **File -> New -> Project...**





Mạch tổ hợp cần thiết kế bao gồm một AND\_GATE và một OR\_GATE.

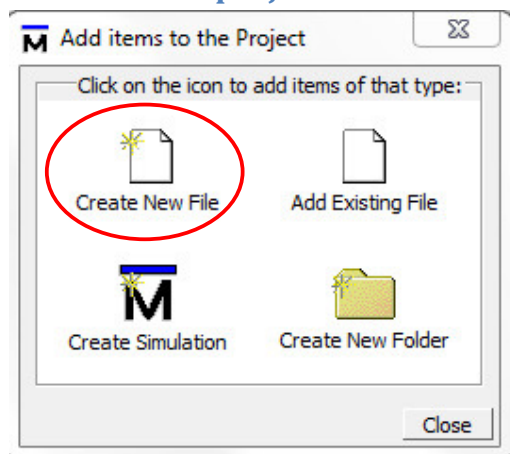


Nhập tên project là **comb\_ckt**, thường chọn tên project giống với tên file chứa mã VHDL (.vhd) và thông thường cũng giống với tên của entity của mã VHDL.

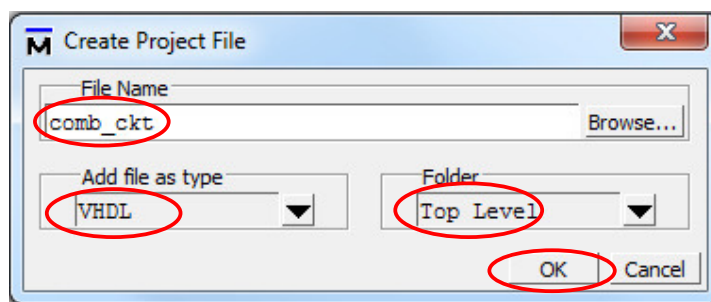
Bấm **Browse** để chọn thư mục chứa project. Nên tạo một thư mục riêng để chứa các file của một project.

Bấm **OK**

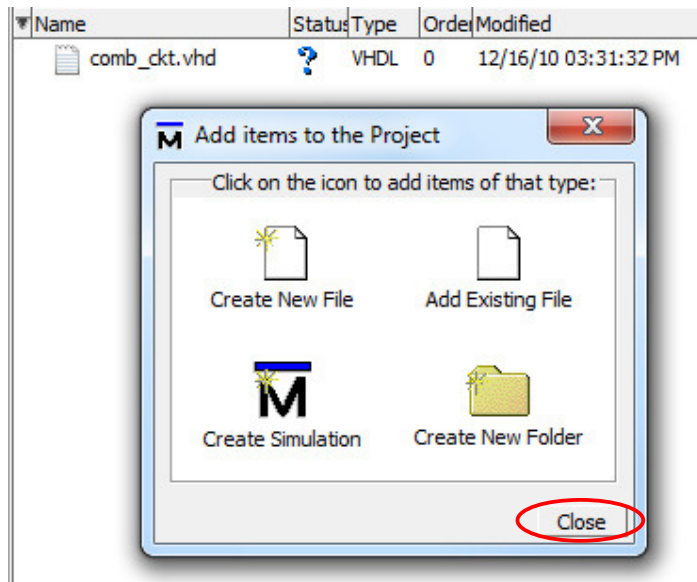
### Thêm file vào project



Bấm **Create New File** để tạo một file mới. (Nếu lười gõ đoạn mã ví dụ sau đây thì chọn **Add Existing File**, rồi chọn file **comb\_ckt.vhd** kèm theo hướng dẫn này 😊)

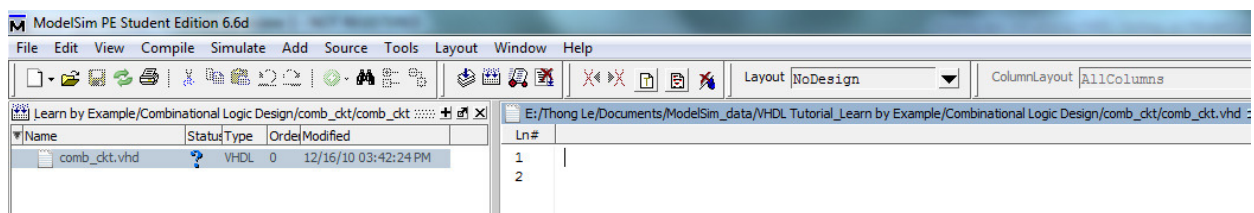


Nhập tên file là **comb\_ckt**, chọn loại file là **VHDL**, giữ nguyên Folder là **Top Level**, rồi bấm **OK**.



Bấm **Close**.

Bấm đôi vào tên file **comb\_ckt.vhd** để mở cửa sổ soạn thảo chương trình



Nhập vào đoạn mã VHDL sau

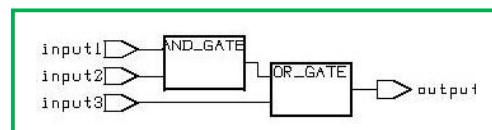
```
-----
-- Combinational Logic Design
-- (ESD book figure 2.4)
-- by Weijun Zhang, 04/2001
--
-- A simple example of VHDL Structure Modeling
-- we might define two components in two separate files,
-- in main file, we use port map statement to instantiate
-- the mapping relationship between each components
-- and the entire circuit.
-----
```

```
library ieee;                                -- component #1
use ieee.std_logic_1164.all;

entity OR_GATE is
port(   X:    in std_logic;
        Y:    in std_logic;
        F2:   out std_logic);
end OR_GATE;

architecture behv of OR_GATE is
begin
process(X,Y)
begin
    F2 <= X or Y;                                -- behavior des.
end process;
end behv;
```

```
-----
library ieee;                                -- component #2
use ieee.std_logic_1164.all;
```



Mô tả linh kiện thứ 1:  
**OR\_GATE**

Mô tả linh kiện thứ 2:  
**AND\_GATE**

```

entity AND_GATE is
port(   A:      in std_logic;
        B:      in std_logic;
        F1:     out std_logic
);
end AND_GATE;

architecture behv of AND_GATE is
begin
process(A,B)
begin
    F1 <= A and B;          -- behavior des.
end process;
end behv;

-----

library ieee;                -- top level circuit
use ieee.std_logic_1164.all;
use work.all;

entity comb_ckt is
port(   input1: in std_logic;
        input2: in std_logic;
        input3: in std_logic;
        output: out std_logic
);
end comb_ckt;

architecture struct of comb_ckt is

    component AND_GATE is      -- as entity of AND_GATE
    port(   A: in std_logic;
            B: in std_logic;
            F1: out std_logic
    );
    end component;

    component OR_GATE is      -- as entity of OR_GATE
    port(   X: in std_logic;
            Y: in std_logic;
            F2: out std_logic
    );
    end component;

    signal wire: std_logic;    -- signal just like wire

begin

    -- use sign ">=" to clarify the pin mapping


    Gate1: AND_GATE port map (A=>input1, B=>input2, F1=>wire);
    Gate2: OR_GATE port map (X=>wire, Y=>input3, F2=>output);

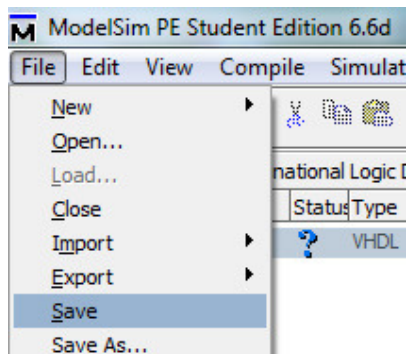
end struct;
-----

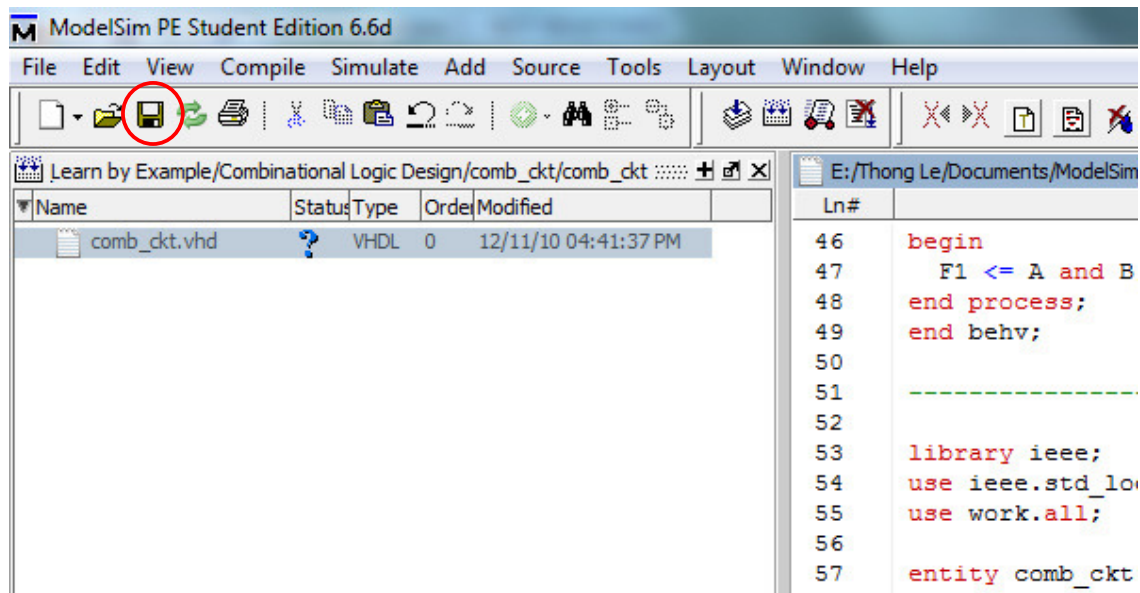
```

Mô tả mạch tổ hợp:

**comb\_ckt**

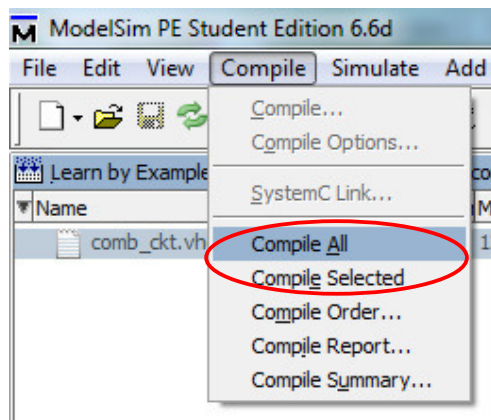
Bấm **File** -> **Save** hoặc bấm biểu tượng  để lưu chương trình.



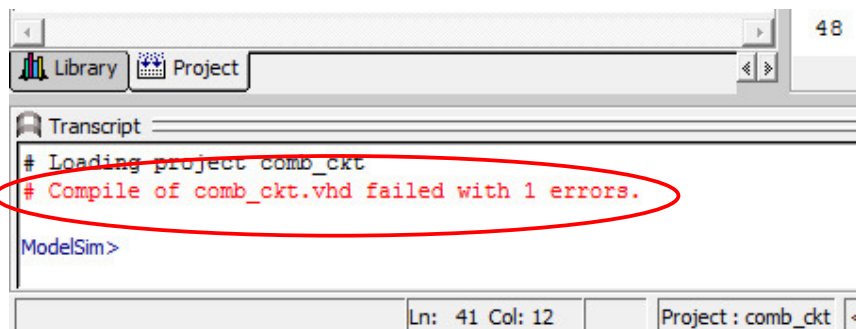


### Biên dịch (compile)

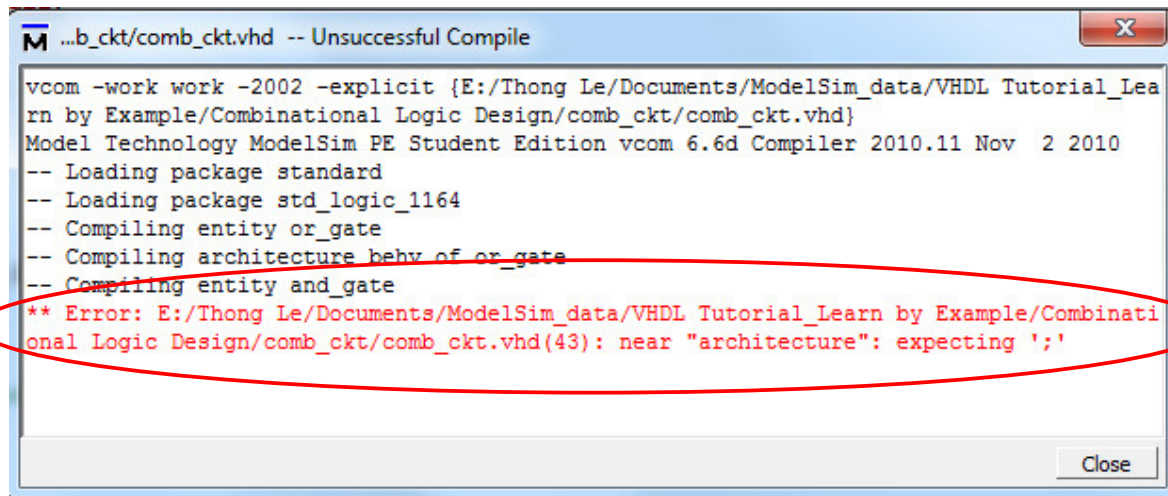
Bấm **Compile** -> **Compile All** (hoặc **Compile Selected** nếu chỉ muốn biên dịch 1 file)



Nếu có thông báo lỗi thì bấm đôi vào dòng thông báo lỗi để mở ra cửa sổ báo lỗi.







```

M ...b_ckt/comb_ckt.vhd -- Unsuccessful Compile

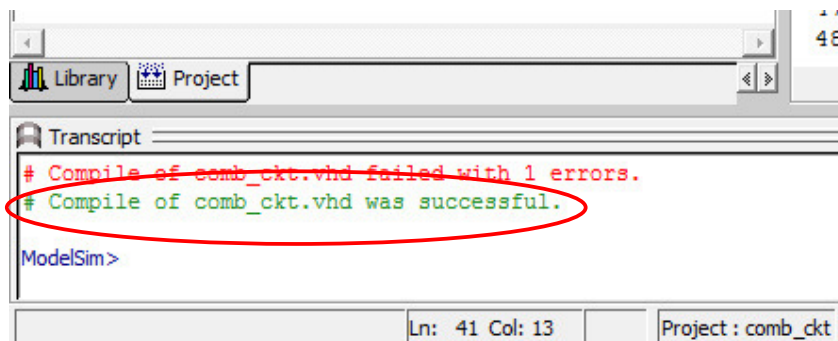
vcom -work work -2002 -explicit {E:/Thong Le/Documents/ModelSim_data/VHDL Tutorial_Learn by Example/Combinational Logic Design/comb_ckt/comb_ckt.vhd}
Model Technology ModelSim PE Student Edition vcom 6.6d Compiler 2010.11 Nov 2 2010
-- Loading package standard
-- Loading package std_logic_1164
-- Compiling entity or_gate
-- Compiling architecture behv of or_gate
-- Compiling entity and_gate
** Error: E:/Thong Le/Documents/ModelSim_data/VHDL Tutorial_Learn by Example/Combinational Logic Design/comb_ckt/comb_ckt.vhd(43): near "architecture": expecting ';'

```

Xem báo lỗi gì và tại dòng lệnh nào. (Trong ví dụ trên, lỗi thiếu dấu ; tại dòng lệnh 43)

Bấm **Close** để đóng cửa sổ báo lỗi.

Sửa lỗi, lưu và biên dịch lại cho đến khi không còn lỗi.



```

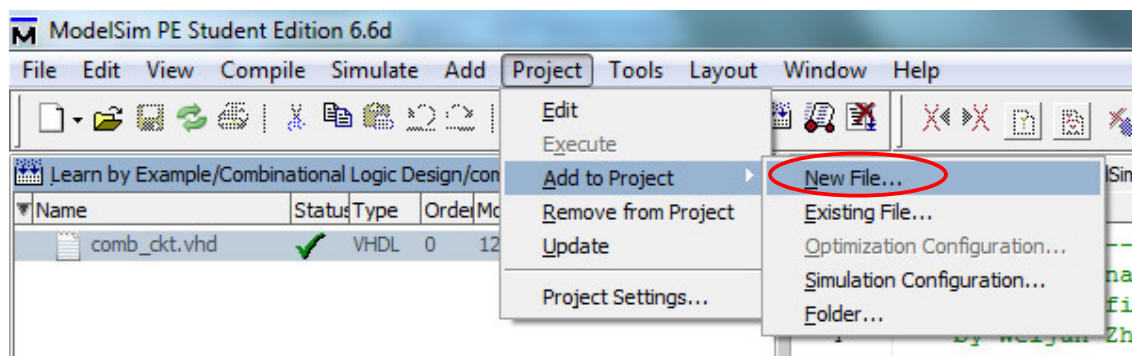
# Compile of comb_ckt.vhd failed with 1 errors.
# Compile of comb_ckt.vhd was successful.

ModelSim>

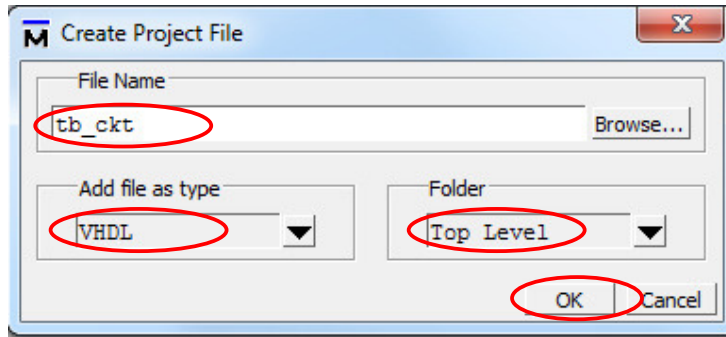
```

## Tạo Testbench

Bấm **Project -> Add to Project -> New File...** để tạo một file testbench. (Nếu lười gõ đoạn mã ví dụ sau đây thì chọn **Existing File**, rồi chọn file **tb\_ckt.vhd** kèm theo hướng dẫn này ☺)



Nhập vào tên file testbench là **tb\_ckt**, chọn loại file là **VHDL**, giữ nguyên Folder là **Top Level**, rồi bấm **OK**.



Tương tự như phần tạo file **comb\_ckt.vhd**, nhập đoạn mã VHDL sau cho file **tb\_ckt.vhd**.

```

-----
-- Test Bench for comb_ckt.vhd
-- (ESD figure 2.4)
-- by Weijun Zhang, 04/2001
--
-- Testbench is used to ensure the design is working properly
-- according to the specification.
-- assert statements are used to test the wrong value against
-- our desired one. we should test as many cases as possible,
-- particularly, we should include upper and lower limits
-- of the operations.
-----

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;

entity CKT_TB is                                -- empty entity
end CKT_TB;

-----

architecture TB of CKT_TB is

-- declare the whole circuit(entity of comb_ckt.vhd) as a component

component comb_ckt is
port(  input1: in std_logic;
       input2: in std_logic;
       input3: in std_logic;
       output: out std_logic
);
end component;

-- declare all I/O ports from unit under test as signals.
-- signals are usually declared within architecture

signal T_input1, T_input2, T_input3, T_output: std_logic;

begin

    U_UT: comb_ckt port map (T_input1,T_input2,T_input3,T_output);

    process

        variable err_cnt: integer := 0;

    begin

        -- Test case 1
        T_input1 <= '0';
        T_input2 <= '0';
        T_input3 <= '0';
        wait for 10 ns;
        assert (T_output=((T_input1 or T_input2) and T_input3))
        report "Failed Case1!" severity error;
        if (T_output/=((T_input1 or T_input2) and T_input3)) then
            err_cnt := err_cnt +1;
        end if;

        -- Test case 2
    
```

```

T_input1 <= '1';
T_input2 <= '1';
T_input3 <= '1';
wait for 10 ns;
assert (T_output=((T_input1 or T_input2) and T_input3))
report "Failed Case1!" severity error;
if (T_output/=((T_input1 or T_input2) and T_input3)) then
    err_cnt := err_cnt +1;
end if;

-- Test case 3
T_input1 <= '1';
T_input2 <= '0';
T_input3 <= '1';
wait for 10 ns;
assert (T_output=((T_input1 or T_input2) and T_input3))
report "Failed Case1!" severity error;
if (T_output/=((T_input1 or T_input2) and T_input3)) then
    err_cnt := err_cnt +1;
end if;

-- Test case 4
T_input1 <= '0';
T_input2 <= '1';
T_input3 <= '0';
wait for 10 ns;
assert (T_output=((T_input1 or T_input2) and T_input3))
report "Failed Case1!" severity error;
if (T_output/=((T_input1 or T_input2) and T_input3)) then
    err_cnt := err_cnt +1;
end if;

-- summary of all the tests to see if any errors

if (err_cnt=0) then
    assert false report "Testbench completed successfully!"
    severity note;
else
    assert true
    report "Something wrong, try again pls!"
    severity error;
end if;

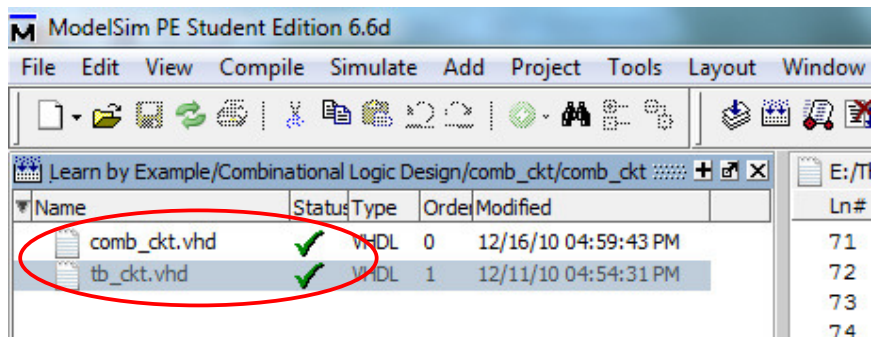
wait;                                -- stop running

end process;
end TB;

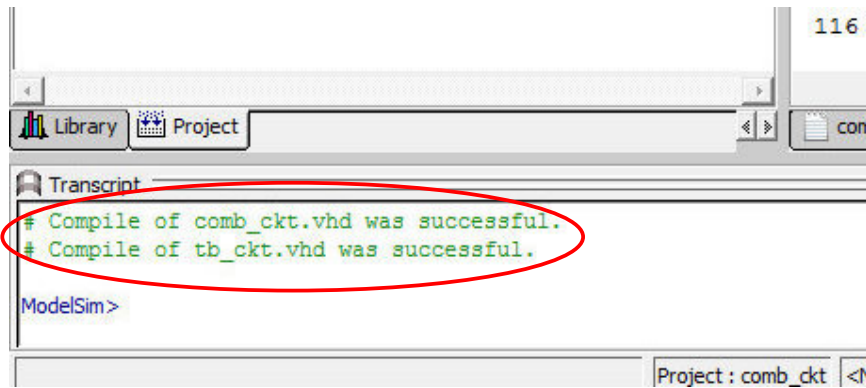
-----
configuration CFG_TB of CKT_TB is
    for TB
        end for;
end CFG_TB;
-----

```

Lưu và biên dịch file **tb\_ckt.vhd** này cho đến khi không còn lỗi.

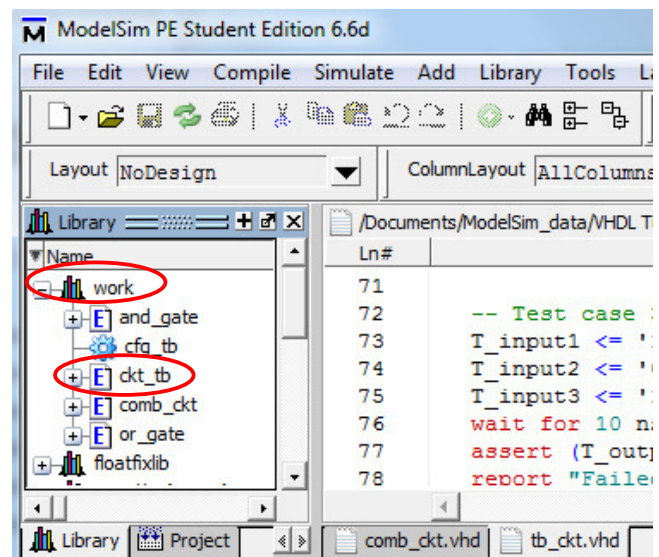
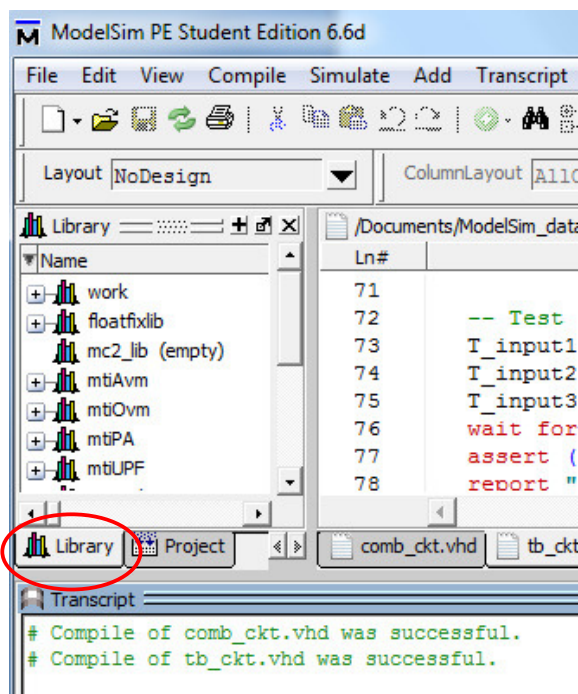






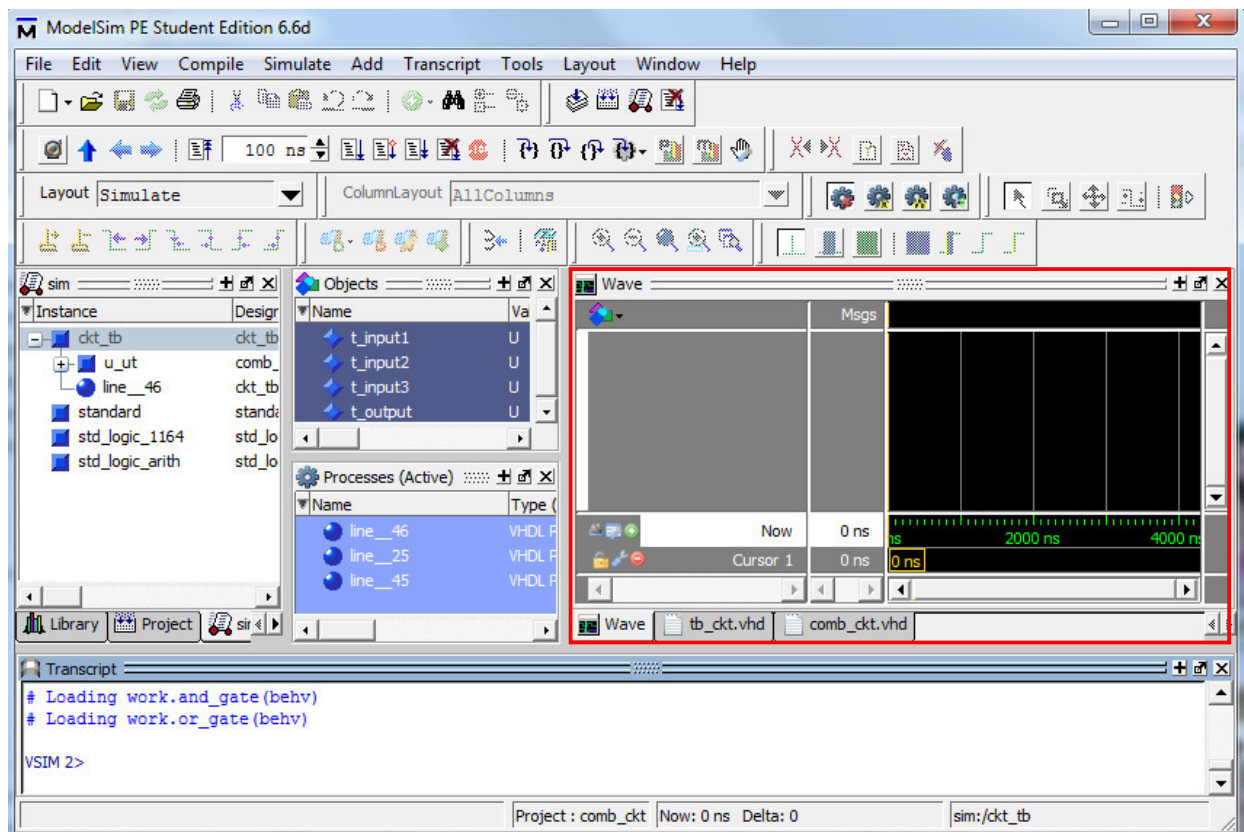
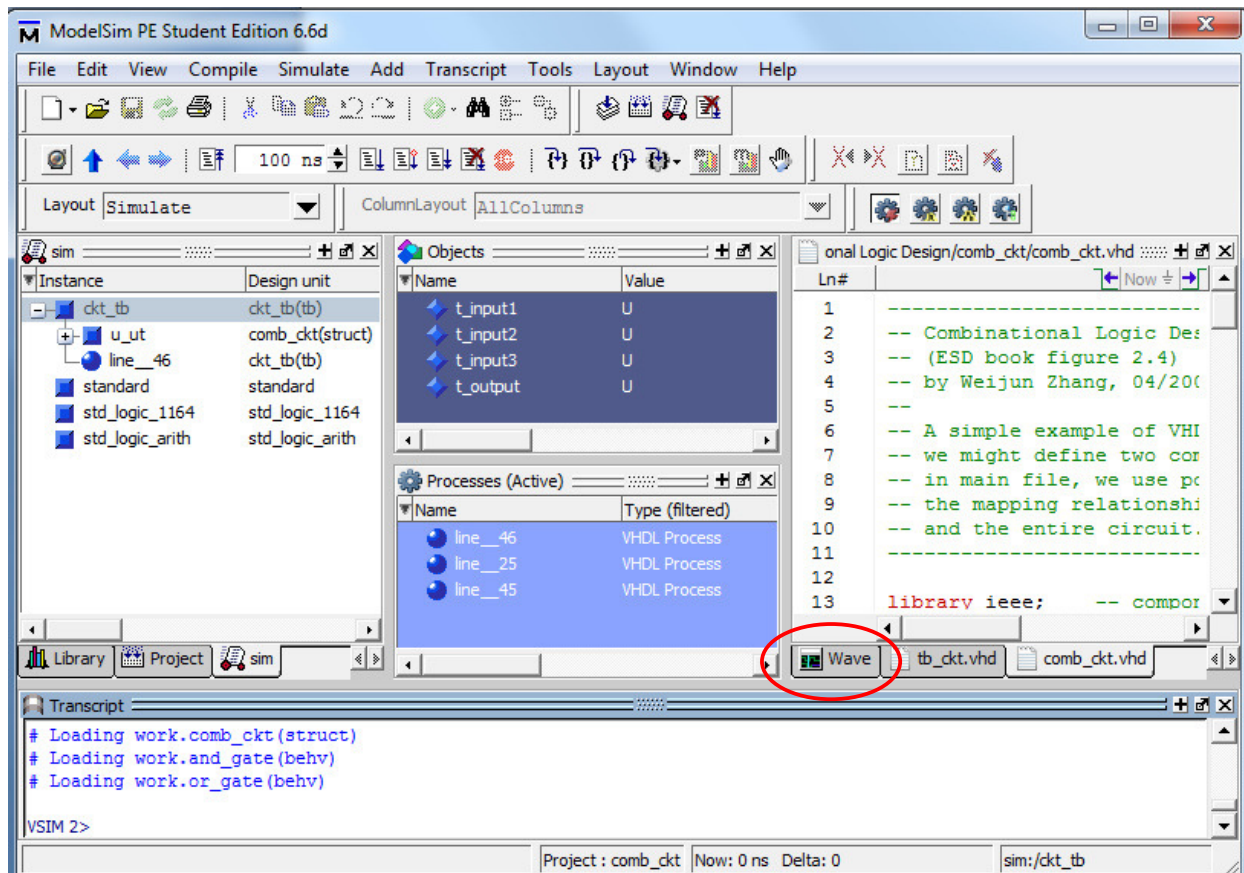
## Chạy mô phỏng

Bấm chọn tab **Library**

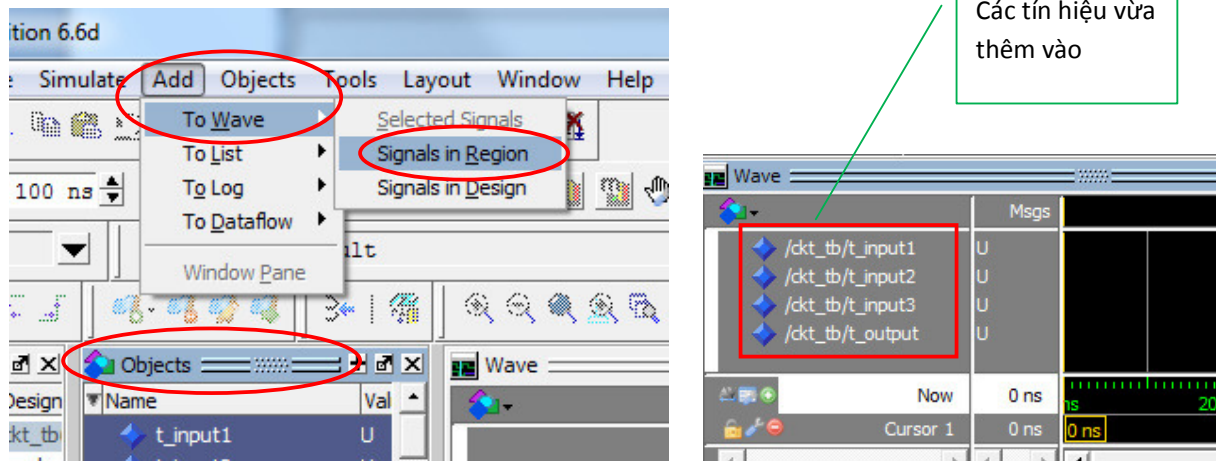


Bấm vào dấu cộng trước thư mục **work** trong cửa sổ **Library**, rồi bấm đôi vào dòng **ckt\_tb** để bắt đầu chạy mô phỏng

Bấm chọn tab **Wave** để thấy cửa sổ Wave

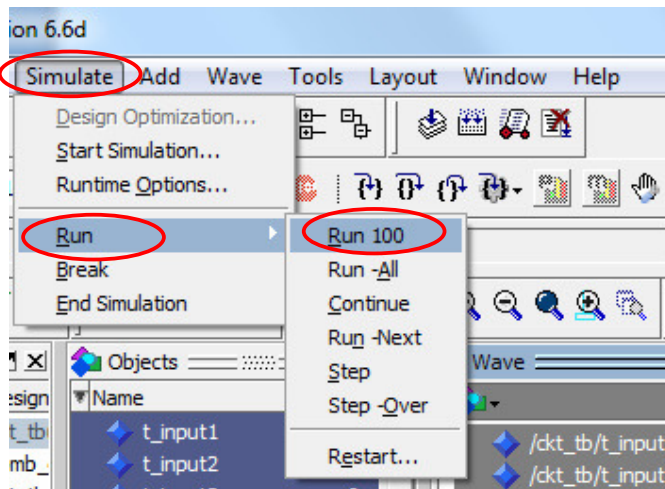



Bấm chọn cửa sổ **Objects**, rồi bấm **Add -> To Wave -> Signals in Region** để thêm các tín hiệu cần quan sát vào cửa sổ Wave

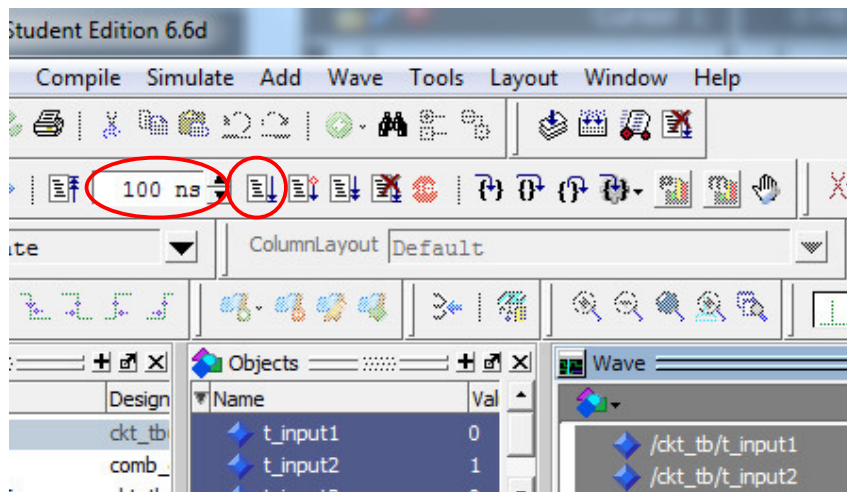


## Bắt đầu chạy mô phỏng

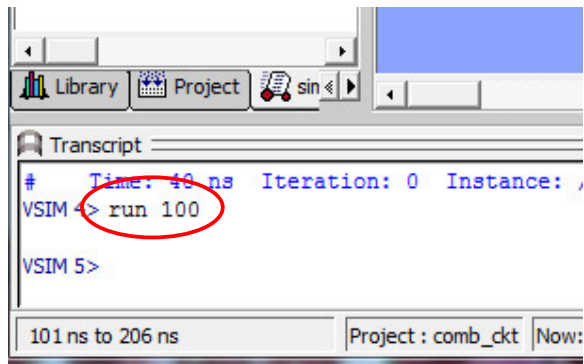
**Cách 1:** Bấm **Simulate -> Run -> Run 100** để chạy mô phỏng trong 100 ns



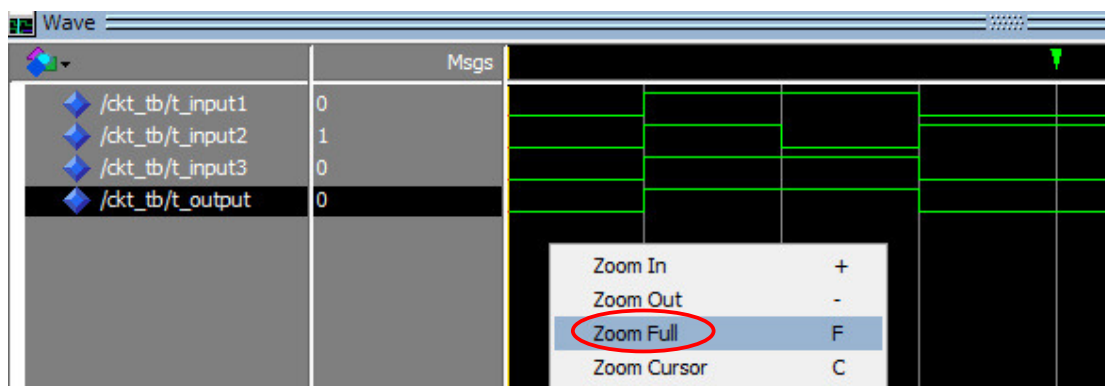
**Cách 2:** Nhập thời gian cần chạy mô phỏng, rồi bấm nút  (Run)



**Cách 3:** Nhập lệnh **run 100** vào dấu nhắc lệnh trong cửa sổ Transcript rồi **Enter**.



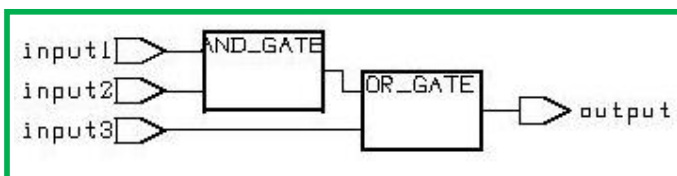
Quan sát dạng sóng thu được trên cửa sổ Wave. Để dễ quan sát, bấm nút phải chuột trên cửa sổ Wave, rồi bấm **Zoom Full**.



Kết quả từ dạng sóng:

Thời gian	t_input1	t_input2	t_input3	t_output
0 – 10 ns	0	0	0	0
10 – 20 ns	1	1	1	1
20 – 30 ns	1	0	1	1
30 – 40 ns	0	1	0	0
...				

Kết quả này chứng tỏ mạch tổ hợp đã thiết kế đúng ☺ !



Trong ví dụ trên, ta đã viết mã VHDL của **AND\_GATE** và **OR\_GATE** chung với **comb\_ckt** trong cùng một file **comb\_ckt.vhd**. Ta cũng có thể viết mã VHDL của **AND\_GATE** và **OR\_GATE** trong hai file riêng rẽ là **AND\_GATE.vhd** và **OR\_GATE.vhd**. Khi đó, ta phải thêm vào project tổng cộng là 4 file (**comb\_ckt.vhd**, **AND\_GATE.vhd**, **OR\_GATE.vhd** và **tb\_ckt.vhd**).

Xem thêm và thử chạy mô phỏng các mạch trong thư mục VHDL codes kèm theo bài hướng dẫn này.

Các mã VHDL dùng trong bài hướng dẫn này và trong thư mục VHDL codes là của tác giả **Weijun Zhang**, University of California Riverside.