

## BÀI 6. MODELSIM VÀ TESTBENCH

### 1. Mục tiêu

Thông qua bài thực hành này, sinh viên sẽ hiểu rõ:

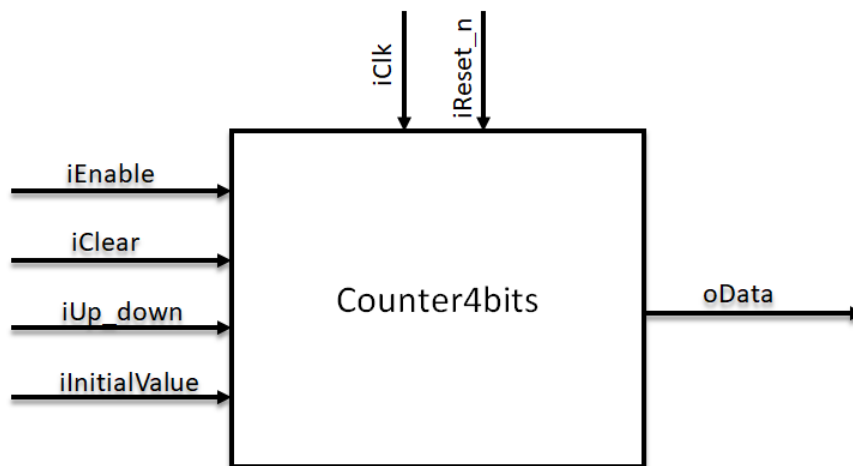
- Cách xây dựng testbench để khởi tạo dữ liệu đầu vào và kiểm tra kết quả đầu ra của các khối phần cứng.
- Cách sử dụng công cụ Modelsim để mô phỏng các khối phần cứng.

### 2. Phần lý thuyết

Trong bài thực hành này, sinh viên sẽ tìm hiểu cách mô phỏng bộ đếm 4 bits – counter4bits.

#### 2.1. Tổng quan mô đun thiết kế

Các tín hiệu của mô đun Counter4bits được mô tả ở hình 1 và bảng 1.



Hình 1. Tín hiệu của mô đun Counter4bits.

Bảng 1. Bảng mô tả các tín hiệu của mô đun Counter4bits.

STT	Tên tín hiệu	Độ rộng (bits)	Hướng	Mô tả
1	iClk	1	Input	Cấp xung clock cho mô đun hoạt động.
2	iReset_n	1	Input	Cấp tín hiệu reset mức thấp cho mô đun.
3	iEnable	1	Input	Nếu iEnable = 1, mô đun được phép hoạt động. Ngược lại, iEnable = 0, mô đun không hoạt động.
4	iClear	1	Input	Nếu iClear = 1, mô đun xóa giá trị đếm đang được lưu trữ.

5	iUp_down	1	Input	Nếu iUp_down = 1, mô đun đếm lên. Nếu iUp_down = 0, mô đun đếm xuống.
6	iInitialValue	4	Input	Giá trị khởi tạo ban đầu.
7	oData	4	Output	Giá trị đếm đầu ra.

## 2.2. Code verilog mô tả mô đun Counter4bits

```

module (
    input          iClk,
    input          iReset_n,
    input          iEnable,
    input          iClear,
    input          iUp_down,
    input [3:0]    iInitialValue,
    output reg [3:0] oData
);

always @(posedge iClk, negedge iReset_n)
begin
    if(~iReset_n)
    begin
        oData <= 3'd0;
    end
    else if(iClear)
    begin
        oData <= iInitialValue;
    end
    else if(iEnable & iUp_down)
    begin
        oData <= oData + 1'b1;
    end
    else if(iEnable & ~iUp_down)
    begin
        oData <= oData - 1'b1;
    end
end
endmodule

```

## 2.3. Xây dựng testbench cho Counter4bits

```

`timescale 1ns/1ps
module counter4bits_tb();
reg clk;
reg reset_n;
reg enable;

```

```

reg clear;
reg up_down;
reg [3:0] initial_value;
wire [3:0] count_value;
// =====
// Initialize clk signal
// =====
parameter PERIOD = 10; // 50 MHz clock
parameter PERIODx20 = (20 * PERIOD);
parameter PERIODx40 = (40 * PERIOD);
parameter PERIODx100 = (100 * PERIOD);

initial
begin
    clk <= 1'b0;
    forever # (PERIOD) clk <= ~clk;
end

// =====
// Initialize reset_n signal
// =====
initial
begin
    reset_n <= 1'b0;
    #(PERIODx20) // reset_n is asserted in 10 clocks
    @(negedge clk) reset_n <= 1'b1;
end

// =====
// Initialize clear, enable, up_down signal
// =====
initial
begin
    clear <= 1'b0;
    enable <= 1'b0;
    up_down <= 1'b0;
    #(PERIODx40) clear <= 1'b1;
    @(negedge clk) clear <= 1'b0;
    up_down <= 1'b1;
    #(PERIODx40) enable <= 1'b1;
    #(PERIODx100) up_down <= 1'b0;
    #(PERIODx100) enable <= 1'b0;
end

// =====
// Initialize initial_value

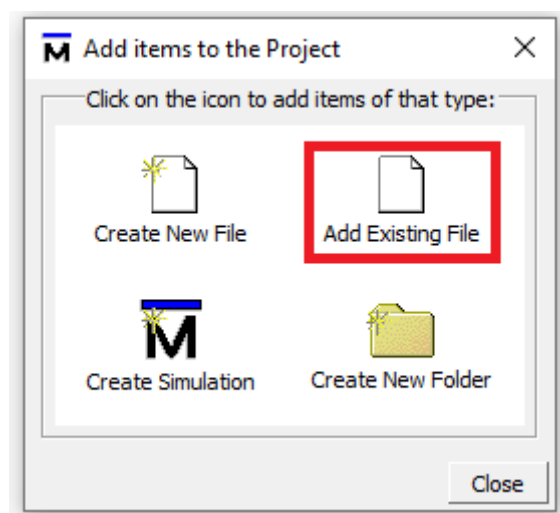
```

```
// =====
initial
begin
    initial_value <= 4'h3;
end
Counter4bits COUNTER4BITS (
    // Inputs
    .iClk (clk),
    .iReset_n (reset_n),
    .iEnable (enable),
    .iClear (clear),
    .iUp_down (up_down),
    .iInitialValue (initial_value),
    // Outputs
    .oData(count_value)
);
endmodule
```

### 3. Phần thực hành

#### 3.1. Mô phỏng Counter4bits dùng công cụ ModelSim dưới dạng waveform

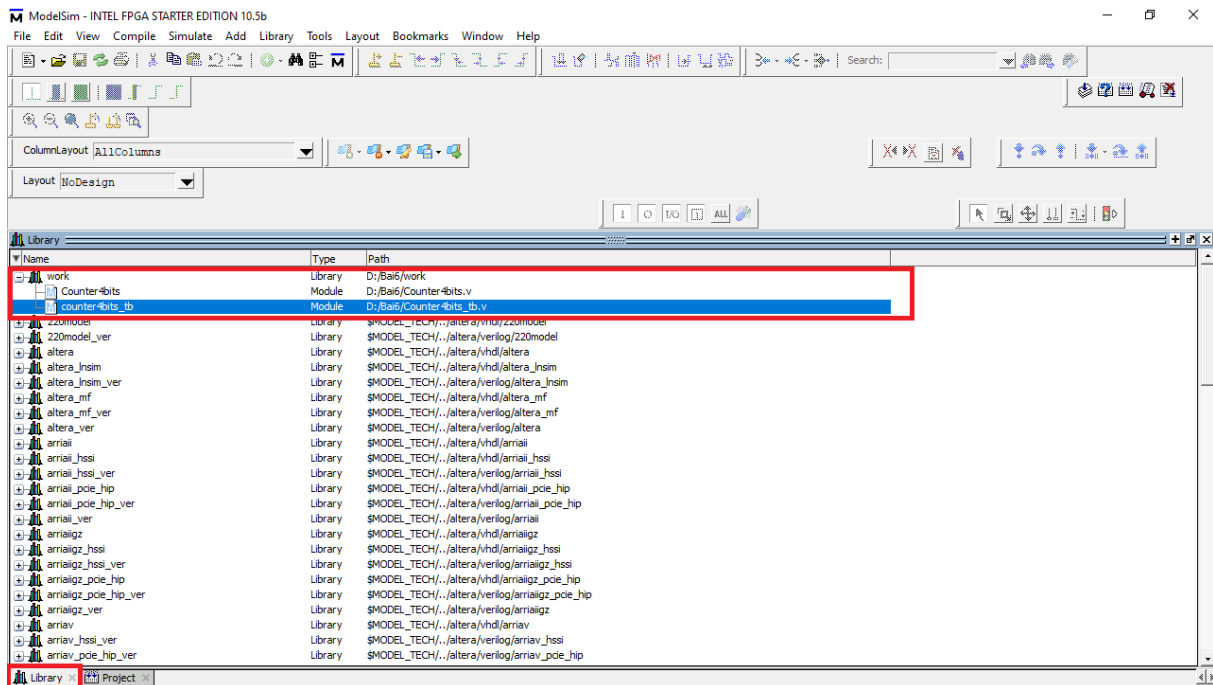
- Trong công cụ Modelsim, chọn **File → New → Project ...** → đặt tên project là **Bai6** và đường dẫn lưu project theo: **D:/Bai6**. Lưu ý: đường dẫn lưu project phải không được chứa khoảng trắng → **OK**.
- Trong cửa sổ Add items to the Project → chọn **Add Existing File** (hình 2) → add 2 files **Counter4bits.v** và **Couter4bits\_tb.v** → **OK** → **Close**.



Hình 2. Cửa sổ Add items to Project.

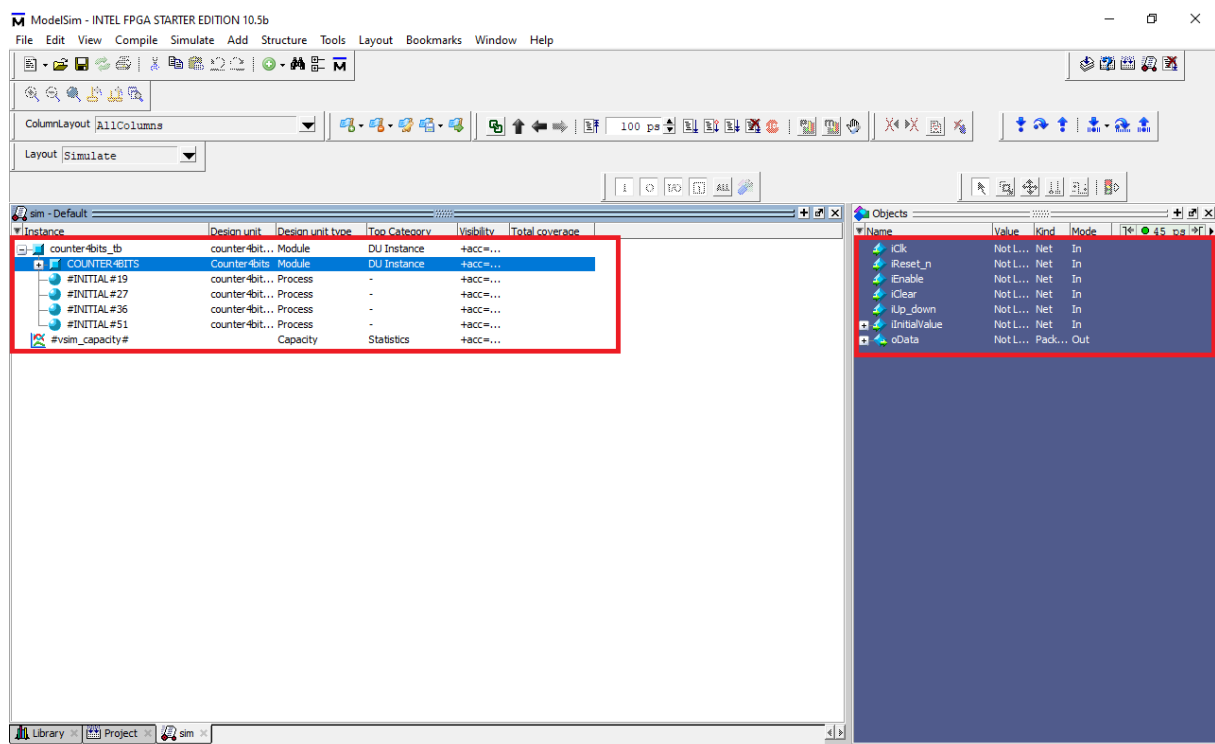
- Chọn **Compile** → **Compile All**. Quan sát cửa sổ Transcript để kiểm tra biên dịch có lỗi hay không.
- Chuyển sang tab **Library**, mở mục work và nhấn đúp chuột vào **counter\_4bits\_tb**

như hình 4.



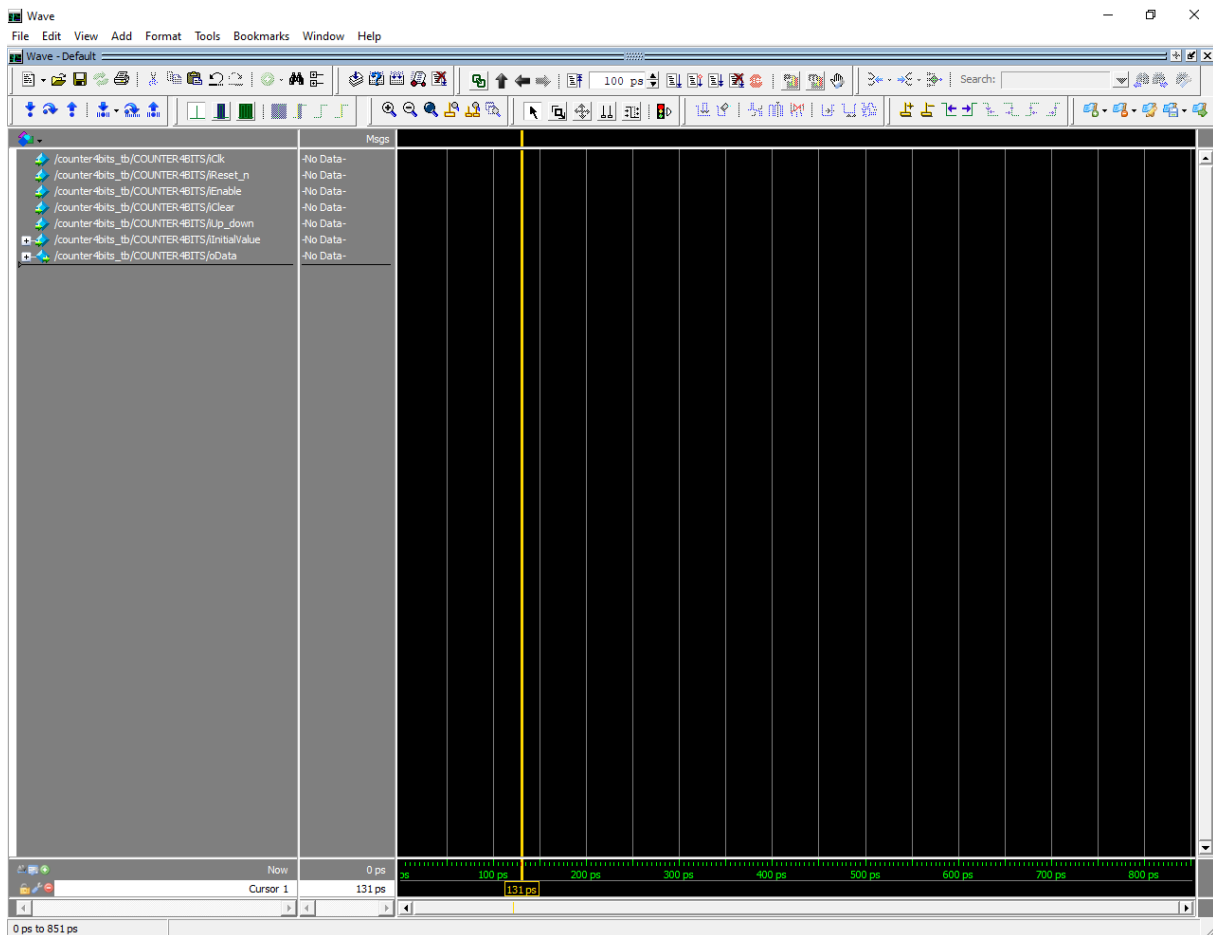
Hình 3. Chuyển sang tab Library

- Trong cửa sổ **sim**, chọn **COUNTER4BITS**. Ở cửa sổ objects, chọn tín hiệu như hình 4 rồi bấm tổ hợp phím **Ctrl + W** để thêm tín hiệu vào cửa sổ Wave.



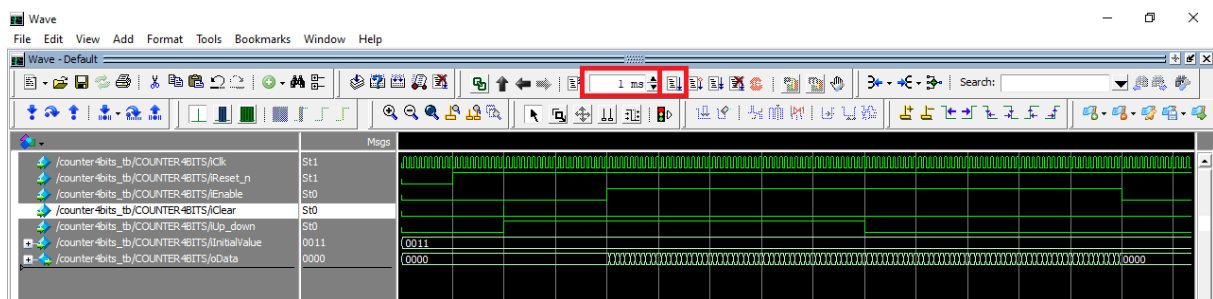
Hình 4. Cửa sổ sim và objects.

- Sau khi thêm tín hiệu, ta có kết quả ở cửa sổ Wave như hình 5.



Hình 5. Cửa sổ Wave.

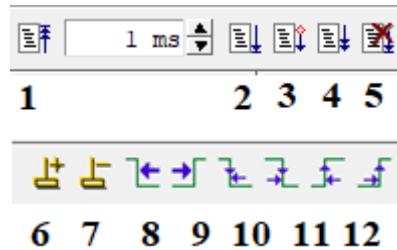
- Trong cửa sổ Wave, gõ **1 ms** vào ô thời gian và click **Run** để chạy chương trình mô phỏng, hình 6. Quan sát dạng sóng xuất hiện trên cửa sổ Wave.



Hình 6. Kết quả mô phỏng.

## BÀI TẬP CHUẨN BỊ Ở NHÀ

Bài. Dựa vào tài liệu tham khảo [1] và [2] , hãy cho biết ý nghĩa các biểu tượng sau trong Modelsim.





## **BÁO CÁO TRÊN LỚP**

Bài 1. Tiến hành mô phỏng như bài hướng dẫn thực hành.

Bài 2. Tiến hành mô phỏng quá trình đọc ghi 3 thanh ghi của mô đun Compute (Bài 4).

## **TÀI LIỆU THAM KHẢO**

[1] Modelsim Advanced Verification and Debugging Altera Tutorial.

[2] Modelsim Tutorial.