

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM

KHOA ĐIỆN TỬ - VIỄN THÔNG



## Báo Cáo Môn Học

### Thực hành Thiết Kế Logic Khả Trình

NĂM HỌC: 2020-2021

---

Đề Tài:

# THIẾT KẾ ĐỒNG HỒ SỐ SỬ DỤNG VERILOG HDL

---

## Nhóm 6

Mã số sinh viên	Email	Họ và Tên	Mã Lớp
18207006	<a href="mailto:18207006@student.hcmus.edu.vn">18207006@student.hcmus.edu.vn</a>	Lê Thành Nghị	18DTVT_CLC
18207019	<a href="mailto:18207019@student.hcmus.edu.vn">18207019@student.hcmus.edu.vn</a>	Chung Ngọc Lân	18DTVT_CLC
18207020	<a href="mailto:18207020@student.hcmus.edu.vn">18207020@student.hcmus.edu.vn</a>	Nguyễn Thanh Lộc	18DTVT_CLC

TP.HCM – 11/2021

## Mục lục

1. Mở đầu
  - a. Giới thiệu đề tài
  - b. Giới thiệu các thành phần sử dụng
2. Sơ đồ khối
  - a. Module DigitalCLK
  - b. Module TB\_CLK
3. Phân tích
  - a. Module DigitalCLK
  - b. Module TB\_CLK
4. Kết quả
  - a. Hình ảnh mô phỏng
  - b. Đánh giá kết quả
5. Ứng dụng

## 1. Mở đầu

### a. Giới thiệu đề tài

Đề tài là thiết kế đồng hồ số sử dụng code verilog HDL.

Đồ án sẽ thực hiện việc tạo ra các tín hiệu giờ phút giây từ xung clock có tần số là 1 Hz được cấp vào, đồng hồ cũng có thể thiết lập giá trị giờ phút giây ban đầu và bắt đầu đếm lên như một đồng hồ thời gian thực. Vì chỉ là đồ án mô phỏng, không nạp trên board thực tế, nên nhóm sẽ tạo ra output của đồng hồ chỉ là tín hiệu nhị phân và được trình bày trên Modelsim.

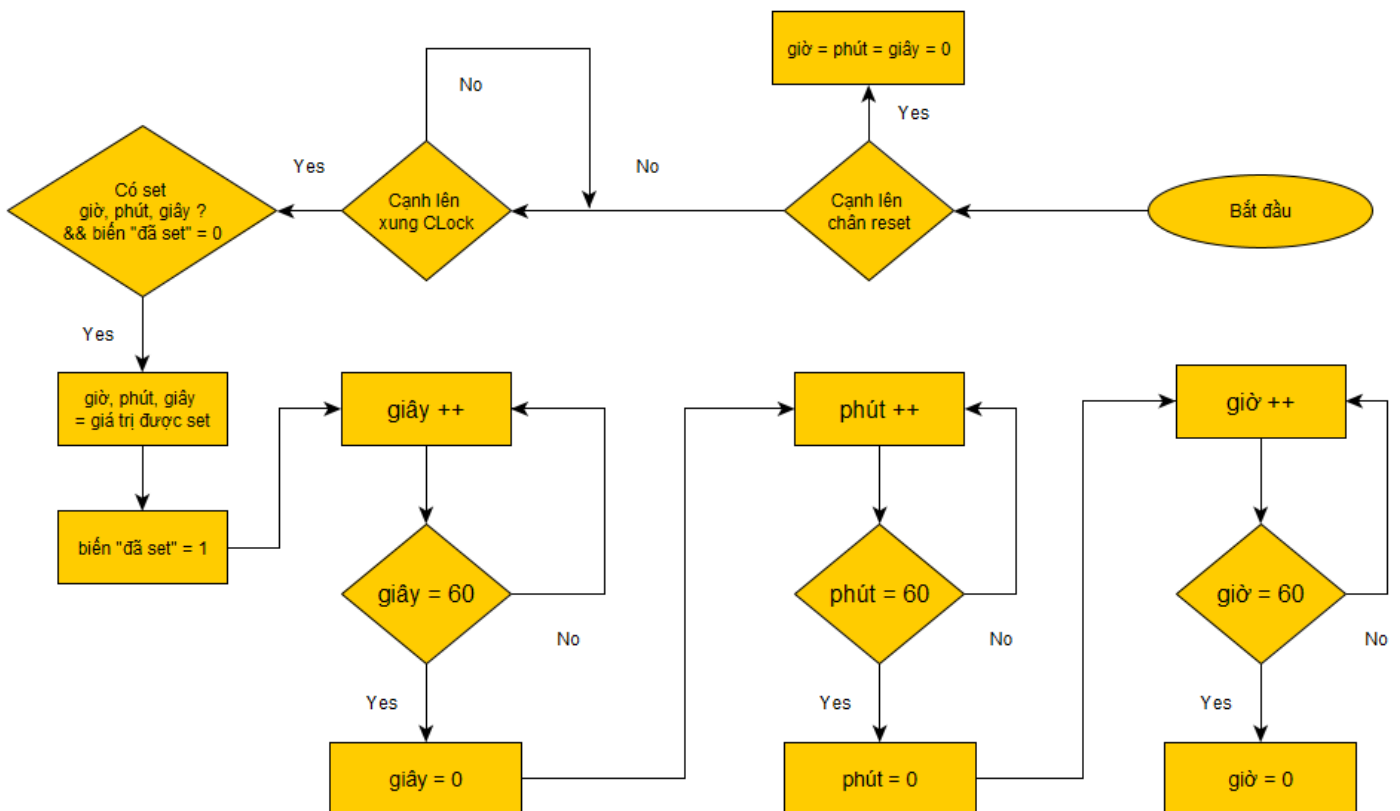
### b. Giới thiệu các thành phần sử dụng

Về kiến thức, đồ án sẽ sử dụng các kiến thức từ môn thực hành logic khả trình, ngôn ngữ được sử dụng là verilog HDL. Mạch có dùng các chân input, output, các dây (wire) để xử lý, nhận, xuất dữ liệu và dùng các khối always, initial để so sánh các chân clock và chạy testbench.

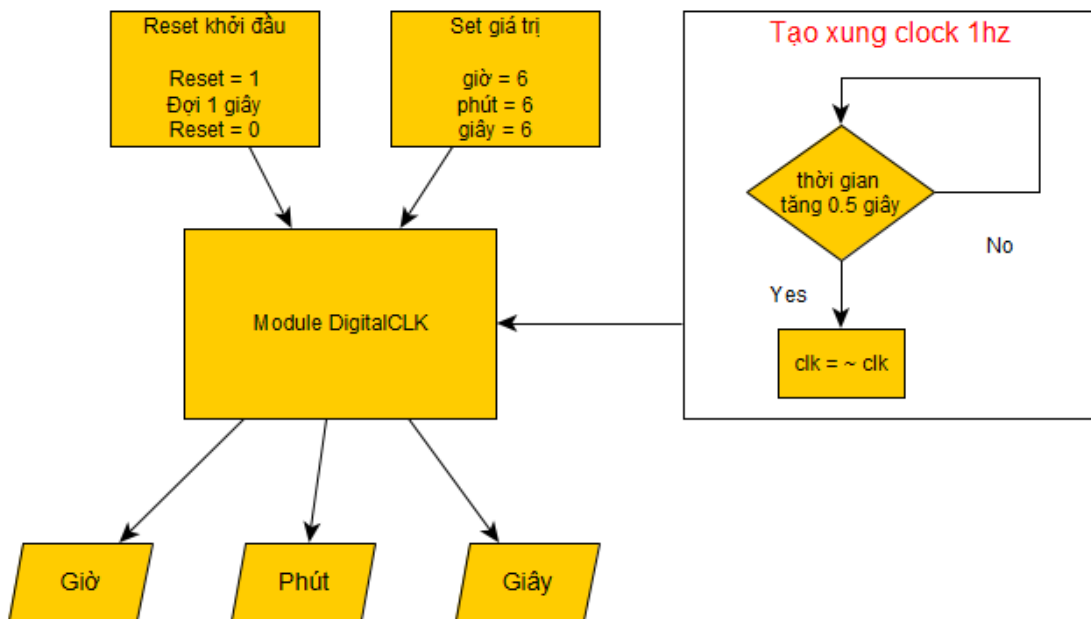
Về module, đồ án sử dụng module DigitalCLK là module chính để đếm giờ, phút, giây theo thời gian thực. Ngoài ra còn có module TB\_CLK để kiểm tra module DigitalCLK có thực hiện đúng tính logic của mạch đồng hồ hay không.

## 2. Sơ đồ khối

### a. Module DigitalCLK



## b. Module TB\_CLK



## 3. Phân tích

### a. Module DigitalCLK

#### Biến:

Các biến output là giờ, phút, giây: seconds, minutes, hours. Biến giờ có độ dài là 6 bit, biến phút và giây có độ dài là 7 bit vì modelsim định nghĩa bit 1 ở MSB là giá trị âm nên nếu chỉ dùng 5 bit và 6 bit thì sẽ không chạy đúng.

Các biến input là set\_minutes, set\_hours, set\_seconds để gán giá trị khởi tạo cho giờ, phút, giây. Biến input clk\_1sec tạo xung clock 1 giây giúp các biến giờ, phút, giây dựa vào để đếm đúng giá trị thời gian thực. Biến input reset để reset đồng hồ về giá trị 00:00:00. Ngoài ra còn có các biến khác để xử lý trong khi chạy chương trình.

#### Chương trình chính:

Khi biến reset = 1 giá trị giờ, phút, giây được gán về 0.

Khi biến set\_mode = 1 các giá trị giờ, phút, giây được gán theo yêu cầu.

Mỗi lần có một cạnh lên của xung clk\_1sec, giá trị giây sẽ tăng 1 đơn vị. Khi giây bằng 60 thì sẽ tăng giá trị phút lên 1 đơn vị và gán giá trị giây về 0. Khi phút bằng 60 thì sẽ tăng giá trị giờ lên 1 đơn vị và gán giá trị phút về 0. Khi giờ bằng 24 sẽ gán giá trị giờ về 0.

```

module DigitalCLK(
    Clk_1sec,      // Clock with 1 Hz frequency
    reset,
    seconds,
    minutes,
    hours,
    setmode,      // setmode = 1 : confirm that time will be set at beginning
    set_seconds,  // set value seconds for clock
    set_minutes,  // set value minutes for clock
    set_hours);   // set value hours for clock

// Input variables
input Clk_1sec;
input reset;
input setmode;
input [6:0] set_seconds;
input [6:0] set_minutes;
input [5:0] set_hours;

// Output variables
output [6:0] seconds;
output [6:0] minutes;
output [5:0] hours;

// Internal variables
reg [6:0] seconds;
reg [6:0] minutes;
reg [5:0] hours;
reg setmode_state = 1'b0; // setmode_state = 0: not ever set time
                          // setmode_state = 1: already set time -> not set time anymore

always @(posedge(Clk_1sec) or posedge(reset))
begin
    if(reset == 1'b1) // Check for active high reset
    begin
        seconds = 6'd0;
        minutes = 6'd0;
        hours = 5'd0;
    end

    else if(Clk_1sec == 1'b1) // At the beginning of each second
    begin
        if ((setmode == 1'b1) && (setmode_state == 1'b0))
        begin
            seconds = set_seconds;
            minutes = set_minutes;
            hours = set_hours;
            setmode_state = 1'b1;
        end

        seconds = seconds + 6'd1;
        if(seconds == 6'd60)
        begin
            seconds = 6'd0;
            minutes = minutes + 6'd1;
            if(minutes == 6'd60)
            begin
                minutes = 6'd0;
                hours = hours + 5'd1;
                if(hours == 5'd24)
                begin
                    hours = 5'd0;
                end
            end
        end
    end
end
end
end

```

## b. Module TB\_CLK

`timescale 100ms / 100ms: đơn vị thời gian sử dụng là 100ms, khoảng hiển thị cũng là 100ms. Nên #5 nghĩa là 500ms = 0.5 giây, #10 nghĩa là 1 giây.

Giá trị khởi tạo để gán cho giờ, phút giây là 6,6,6. Nếu không gán thì giá trị giờ, phút, giây sẽ đều = 0. Sẽ có một khoảng thời gian reset là 1 giây, sau đó đồng hồ sẽ đếm từ thời gian được khởi tạo.

```
`timescale 100ms / 100ms
module TB_CLK;

    // Input variables
    reg Clk_1sec;
    reg reset;
    reg setmode = 1'b1;
    reg [6:0] set_seconds = 6'd6;
    reg [6:0] set_minutes = 6'd6;
    reg [5:0] set_hours = 6'd6;

    // Output variables
    wire [6:0] seconds;
    wire [6:0] minutes;
    wire [5:0] hours;

    // Block test for testing values of Digital_Clock
    DigitalCLK test (
        .Clk_1sec(Clk_1sec),
        .reset(reset),
        .seconds(seconds),
        .minutes(minutes),
        .hours(hours),
        .setmode(setmode),
        .set_seconds(set_seconds),
        .set_minutes(set_minutes),
        .set_hours(set_hours)
    );

    // Generating the Clock with 1 Hz frequency
    initial begin
        Clk_1sec = 1'b1;
        forever #5 Clk_1sec=~Clk_1sec;
    end

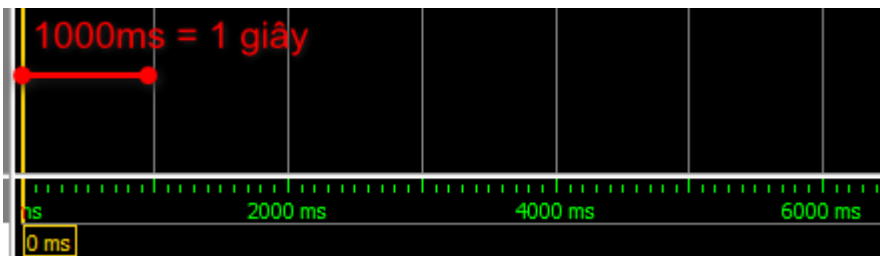
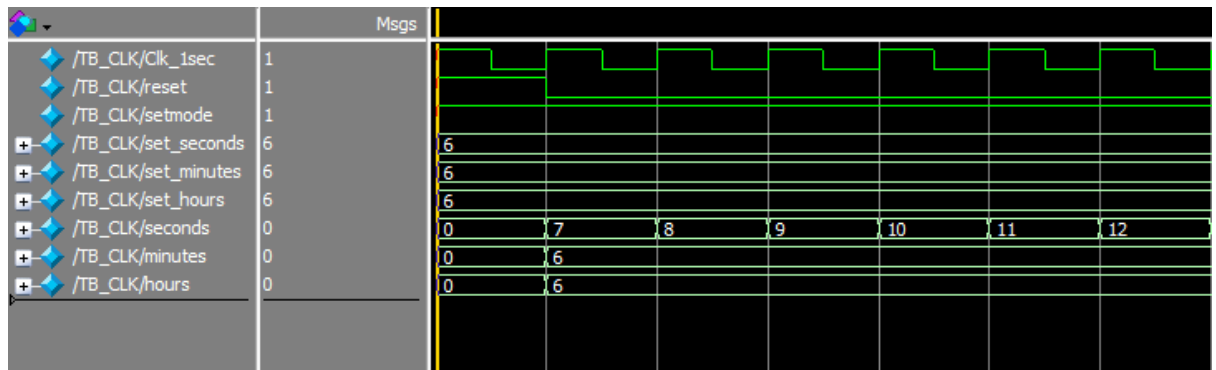
    initial begin
        reset = 1'b1;
        #10 // Wait 1 second to reset
        reset = 1'b0;
    end

endmodule
```

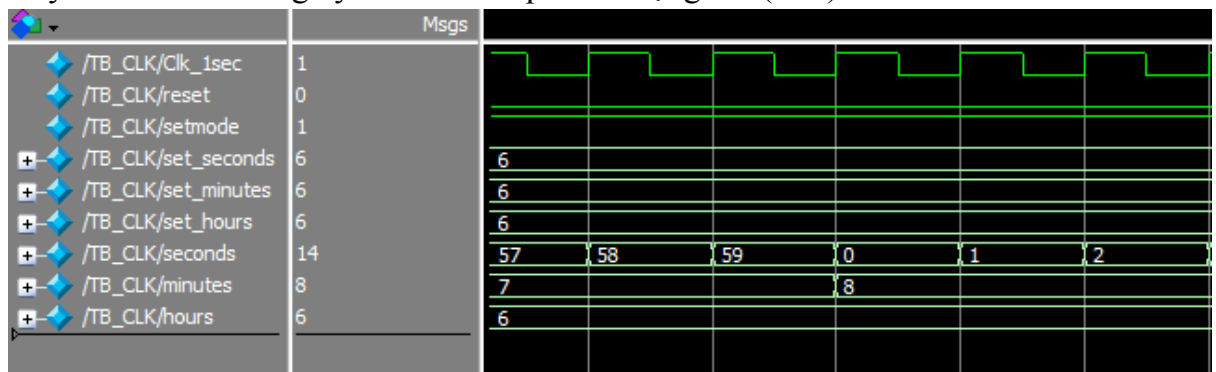
## 4. Kết quả

### a. Hình ảnh khi chạy chương trình

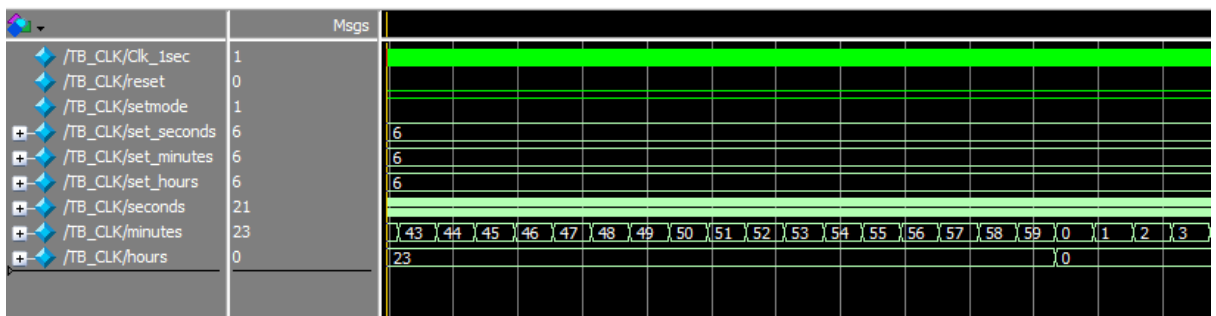
Đây là hình ảnh giai đoạn đầu tiên, vì thiết lập giá trị giây là 6 nên giây đầu tiên sẽ tăng 1 đơn vị là 7, phút và giây = 6 vì cũng là giá trị thiết lập ban đầu.



Đây là hình ảnh sau giây 59 lần 2 thì phút sẽ cộng lên (7+1) thành 8.



Đây là hình ảnh từ giờ 23 lên giờ 00.



## **b. Đánh giá kết quả**

Bài làm đã đạt được đầy đủ yêu cầu đặt ra. Tất cả các trường hợp đã được kiểm tra và đều đúng tính logic của một đồng hồ số. Ngôn ngữ verilog HDL đã được kiểm tra và không có lỗi.

Bài làm có sử dụng các kiến thức đã được học trong môn thực hành logic khả trình bằng ngôn ngữ verilog HDL.

Bài làm được trình bày mạch lạc, rõ ràng, theo đúng cấu trúc module và có chú thích mục đích dùng của các câu lệnh và biến.

## **5. Ứng dụng**

Về học thuật, đồ án đồng hồ số là một đồ án cơ bản để có thể kiểm tra và tổng hợp các kiến thức đã học. Từ việc phát triển sơ đồ, xây dựng module, xây dựng testbench, đồ án giúp tăng kỹ năng phân tích và xây dựng một hệ thống điện tử đơn giản theo verilog HDL.

Về thực tế, đồ án là một bước nền tảng để đặt trong các hệ thống lớn hơn, với chức năng chính là đếm giờ, phút, giây để thực hiện một sự kiện nào đó. Việc tạo tần số 1hz là một bước phổ biến để giúp người kỹ sư hiểu và tạo ra nhiều tín hiệu với tần số khác trong tất cả các hệ thống.