
Báo cáo bài tập

# Ngôn ngữ mô tả phần cứng & FPGA

*Hồ Đức Vũ*

Ngày 14 tháng 5 năm 2024

**Sinh viên thực hiện**
106200284 - Hồ Đức Vũ - 20KTMT2

**Giáo viên hướng dẫn**
TS. Huỳnh Việt Thắng

**Môn học**
Ngôn ngữ mô tả phần cứng & FPGA

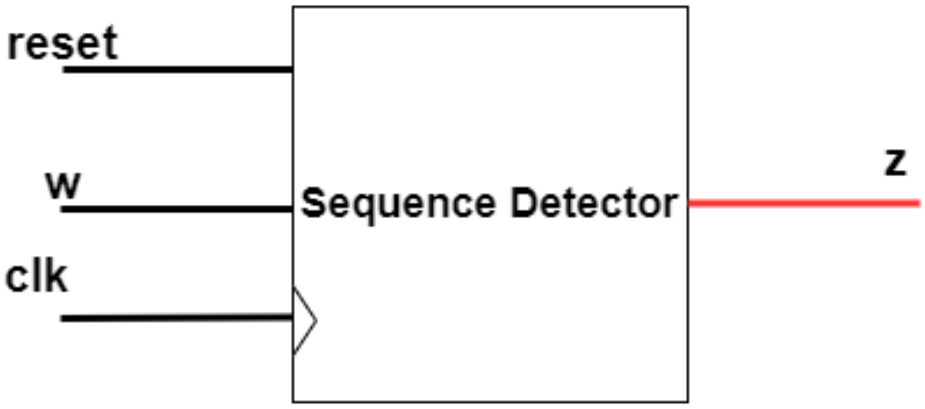
**Xuất bản**
Đà Nẵng, Ngày 14 tháng 5 năm 2024

**Số trang**
12

# Mục lục

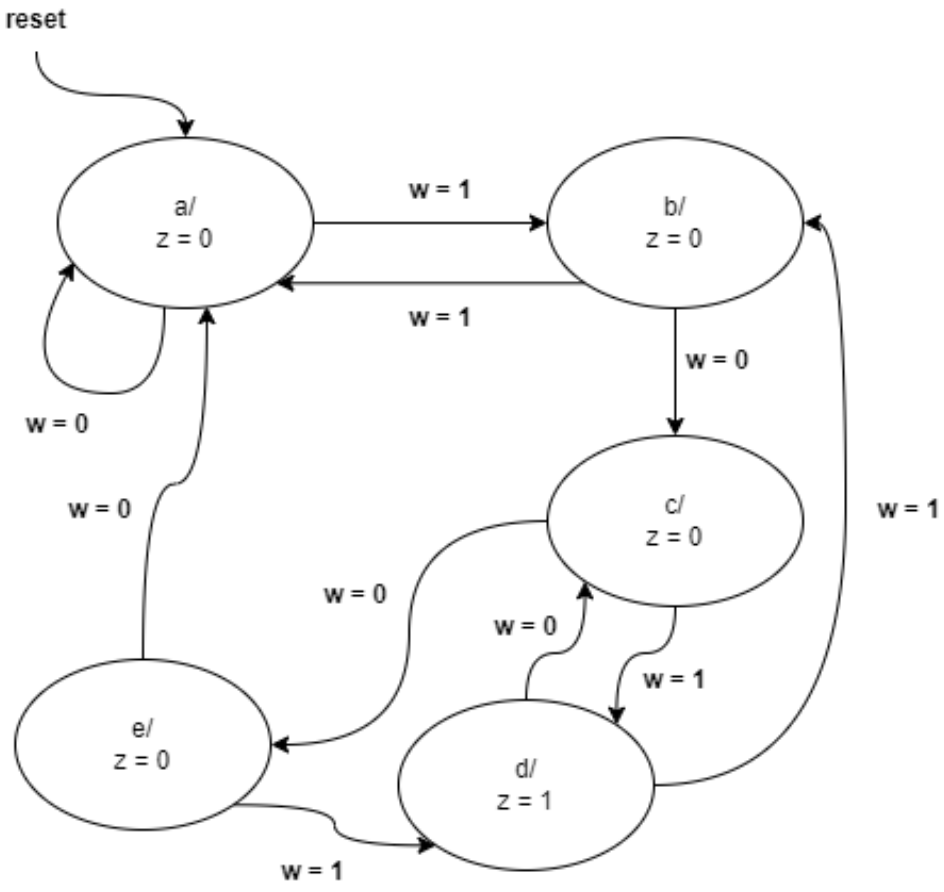
# 1 Sequence Detector

## 1.1 Sơ đồ khối của hệ thống




Hình 1.1: Sơ đồ khối của hệ thống.

## 1.2 Giản đồ máy trạng thái




Hình 1.2: Giản đồ máy trạng thái mạch Sequence Detector.

- Mạch có 1 tín hiệu đầu vào w, 1 tín hiệu đầu ra z, 1 tín hiệu clock clk sườn lên và 1 tín hiệu reset mức cao.
- Tín hiệu đầu ra z = 1 khi tín hiệu đầu vào có chuỗi w = "1001"hoặc w = "101". Nếu trường hợp khác thì z = 0.

## 1.3 Chương trình thiết kế (sequence_detector_2.v)

```
module sequence_detector_2(w, z, clk, reset);

  input wire w, clk, reset;
  output z;

  localparam [2:0] a = 0, //00
                   b = 1, // 01
                   c = 2,// 10
                   d = 3, // 11
                   e = 4;// 100

  reg [2:0] state_reg, state_next;

  always @(posedge clk)
    if (reset) begin
      state_reg <= a;
    end
    else
      state_reg <= state_next;

  assign z = (state_reg == d) ? 1 : 0;

  always @(state_reg or w)
    case(state_reg)
      a: begin
       if(w)
         state_next = b;
       else
         state_next = state_reg;
      end
      b:
      begin
      if(w)
        state_next = a;
      else
        state_next = c;
      end
      c:
      begin
      if (w) begin
        state_next = d;
        end
      else begin
        state_next = e;
        end
      end
      d:
      begin
        if (w)
```

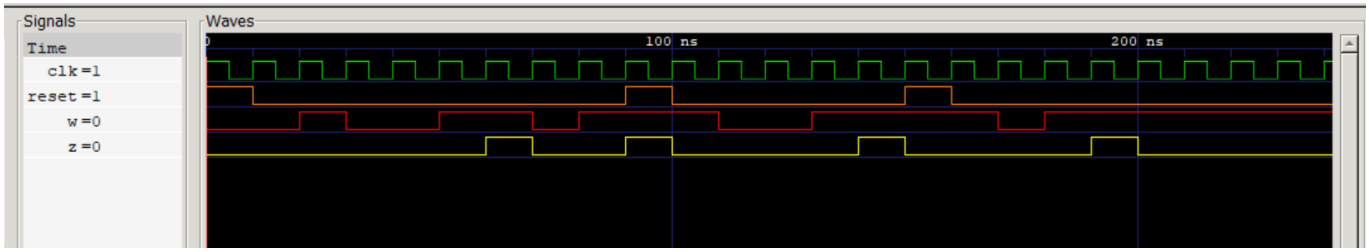
```
            state_next = b;
          else
            state_next = c;
        end
        e:
        begin
          if(w) begin
            state_next = d;
          end
          else begin
            state_next = a;
          end
        end
      endcase
endmodule
```

## 1.4 Chương trình Test Bench (sequence_detector_2_tb.v)

```
`timescale 1ns/ 10ps
`include "sequence_detector_2.v"

module sequence_detector_2_tb;

  reg w, clk, reset;
  wire z;
  localparam T = 20;
  sequence_detector_2 detector(.w(w), .z(z), .clk(clk),.reset(reset));

  always
  begin
  clk = 1'b1;
  #(T/4);
  clk = 1'b0;
  #(T/4);
  end

  initial
  begin
    $dumpfile("sequence_detector_2_tb.vcd");
    $dumpvars(0, sequence_detector_2_tb);
  end

  initial
  begin
    reset = 1; w = 0;

    #(T/2); reset <= 0; w <= 0;
    #(T/2); w <= 1;
    #(T/2); w <= 0;
    #(T/2); w <= 0;
    #(T/2); w <= 1;

    #(T/2); w <= 1;
    #(T/2); w <= 0;
    #(T/2); w <= 1;

    #(T/2); reset <= 1;
```

```
40      #(T/2); reset <= 0;
41      #(T/2); w <= 0;
42      #(T/2); w <= 0;
43      #(T/2); w <= 1;
44      #(T/2); w <= 1;
45
46      #(T/2); reset <= 1;
47      #(T/2); reset <= 0;
48      #(T/2); w <= 0;
49      #(T/2); w <= 1;
50      #(T/2); w <= 1;
51      #(T*4); $finish;
52    end
53  endmodule
```

## 1.5 Kết quả mô phỏng trên phần mềm Icarus




Hình 1.3: Dạng sóng tín hiệu mô phỏng mạch Sequence Detector.

## 1.6 Kết luận

Mạch hoạt động với đầu ra tương ứng so với đầu vào theo nguyên lý hoạt động của mạch. Đối với đầu vào w có giá trị liên tục 1001 thì đầu ra z = 1, với đầu vào w có giá trị liên tục 101 thì đầu ra z cũng là 1, các trường hợp còn lại thì z = 0. Giá trị w có 2 chuỗi 1001 và 101 liên tục thì đầu ra z vẫn hoạt động đúng theo nguyên lý mạch.

Mạch có thể được ứng dụng trong việc xử lý chuỗi tín hiện, đông bộ hóa trong truyền tín hiệu số và cũng có thể được ứng dụng trong việc kiểm tra và xác minh thông tin trong mạch số.

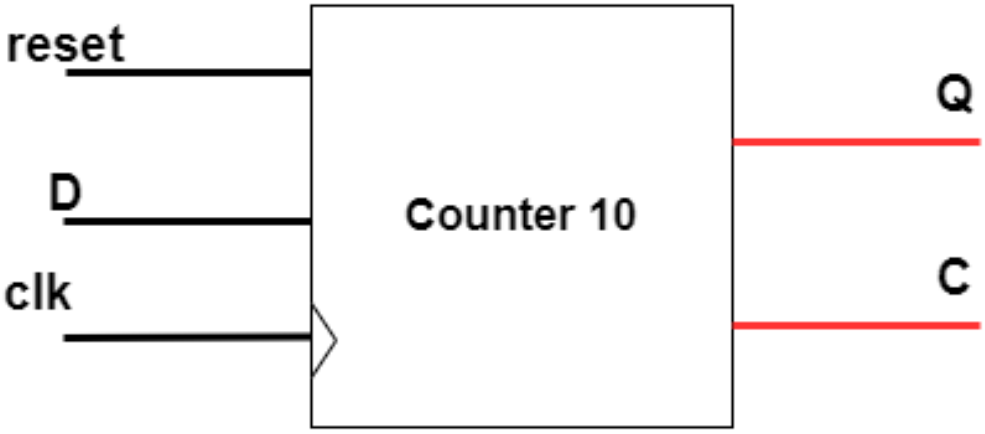
# 2 Counter to 10

## 2.1 Sơ đồ khối của hệ thống

Thiết kế module đếm 10 thực hiện mạch đếm đồng bộ đếm 10 (từ 0 đến 9) cho phép đếm lên/xuống với các tín hiệu vào như sau:

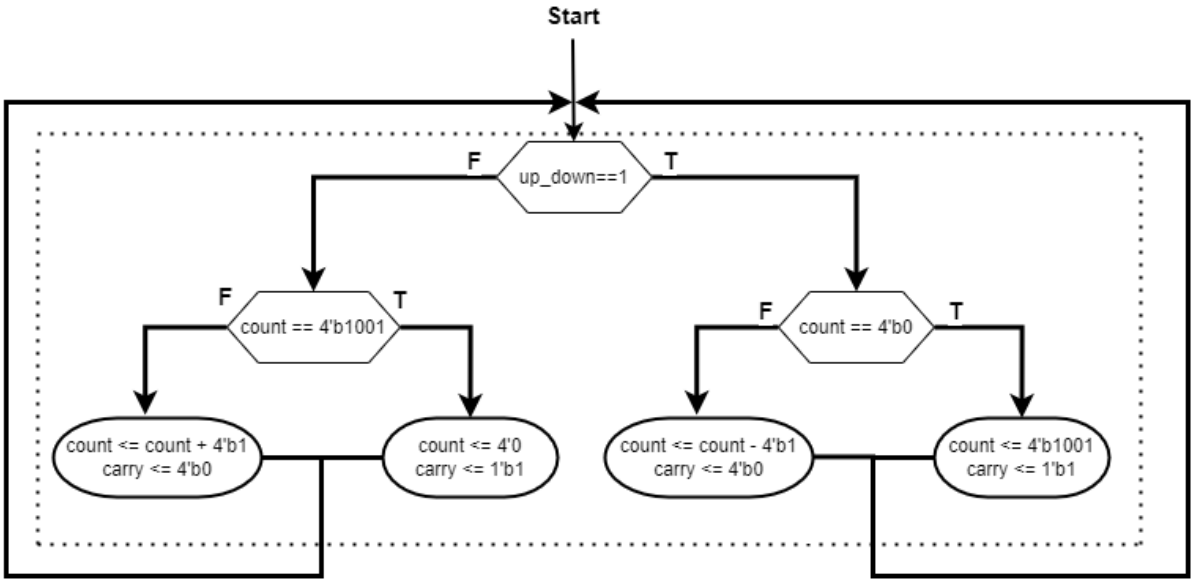
- Clk: xong clock tác động theo sườn lên.
- reset: tín hiệu xóa bộ đếm về 0, mức tích cự là 1.
- D: tín hiệu điều khiển chiều đếm, D = 0 đếm lên, D = 1 đếm xuống.
- Q: tín hiệu ra, lưu giữ giá trị đếm.

- C: tín hiệu ra báo cho biết bộ đếm tràn khi đếm lên hoặc xuống, có 2 trường hợp:
  - Đếm lên: C = 1 khi bộ đếm đang đếm lên và giá trị bộ đếm chuyển từ 9 sang 0, ngược lại C = 0.
  - Đếm xuống: C = 1 khi bộ đếm đang đếm xuống với giá trị bộ đếm chuyển từ 0 sang 9, ngược lại C = 0.




Hình 2.1: Sơ đồ khối của hệ thống.

## 2.2 ASM chart




Hình 2.2: ASM Chart thể hiện nguyên lý hoạt động của mạch counter to 10.

## 2.3 Chương trình thiết kế (counter_to_10.v)

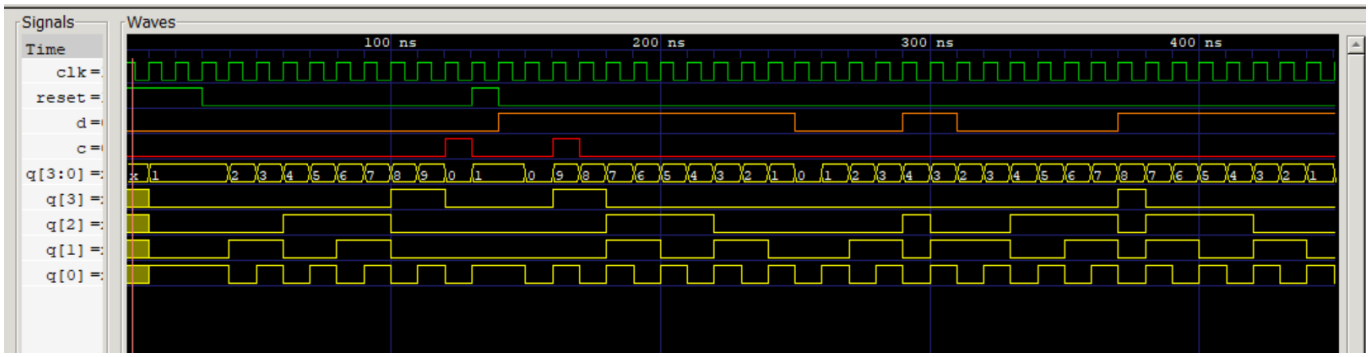
```
module counter_to_10(clk, reset, up_down, count, carry);

  input wire clk, reset, up_down;
  output reg carry;
  output reg[3:0] count;

  always @(posedge clk)
  begin
    if(reset) begin
      count = 4'b0;
      carry = 0;
    end
    if(up_down) begin
      if(count == 4'b0) begin
        count <= 4'b1001;
        carry <= 1'b1;
      end
      else begin
        count <= count - 4'b1;
        carry <= 0;
      end
    end
    else begin
      if(count == 4'b1001) begin
        count <= 4'b0;
        carry <= 1'b1;
      end
      else begin
        count <= count + 4'b1;
        carry <= 0;
      end
    end
  end
endmodule
```

## 2.4 Chương trình Test Bench (counter_to_10_tb.v)

```
`timescale 1ns/1ns
`include "counter_to_10.v"

module counter_to_10_tb();
  reg clk, reset, d;
  wire c;
  wire[3:0] q;

  localparam T = 20;
  //module counter_to_10(clk, reset, up_down, count, carry);
  counter_to_10 counter (clk, reset, d, q, c);
  always
  begin
    clk = 1'b1;
    #(T/4);
    clk = 1'b0;
    #(T/4);
  end

```

```
  initial
  begin
  $dumpfile("counter_to_10_tb.vcd");
  $dumpvars(0, counter_to_10_tb);
  end

  initial
  begin
    reset <= 1; d <= 0;
    #(T)
    #(T/2); reset <= 0; d <=0;
    #(T*5);
    reset <= 1;
    #(T/2); reset <= 0; d <=1;
    #(T*5);
    #(T/2); d <=0;
    #(T*2); d<=1;
    #(T); d <=0;
    #(T*3); d<=1;

    #(T*4);$finish;
  end
endmodule
```

## 2.5 Kết quả mô phỏng trên phần mềm Icarus




Hình 2.3: Dạng sóng tín hiệu mô phỏng mạch counter to 10.

## 2.6 Kết luận

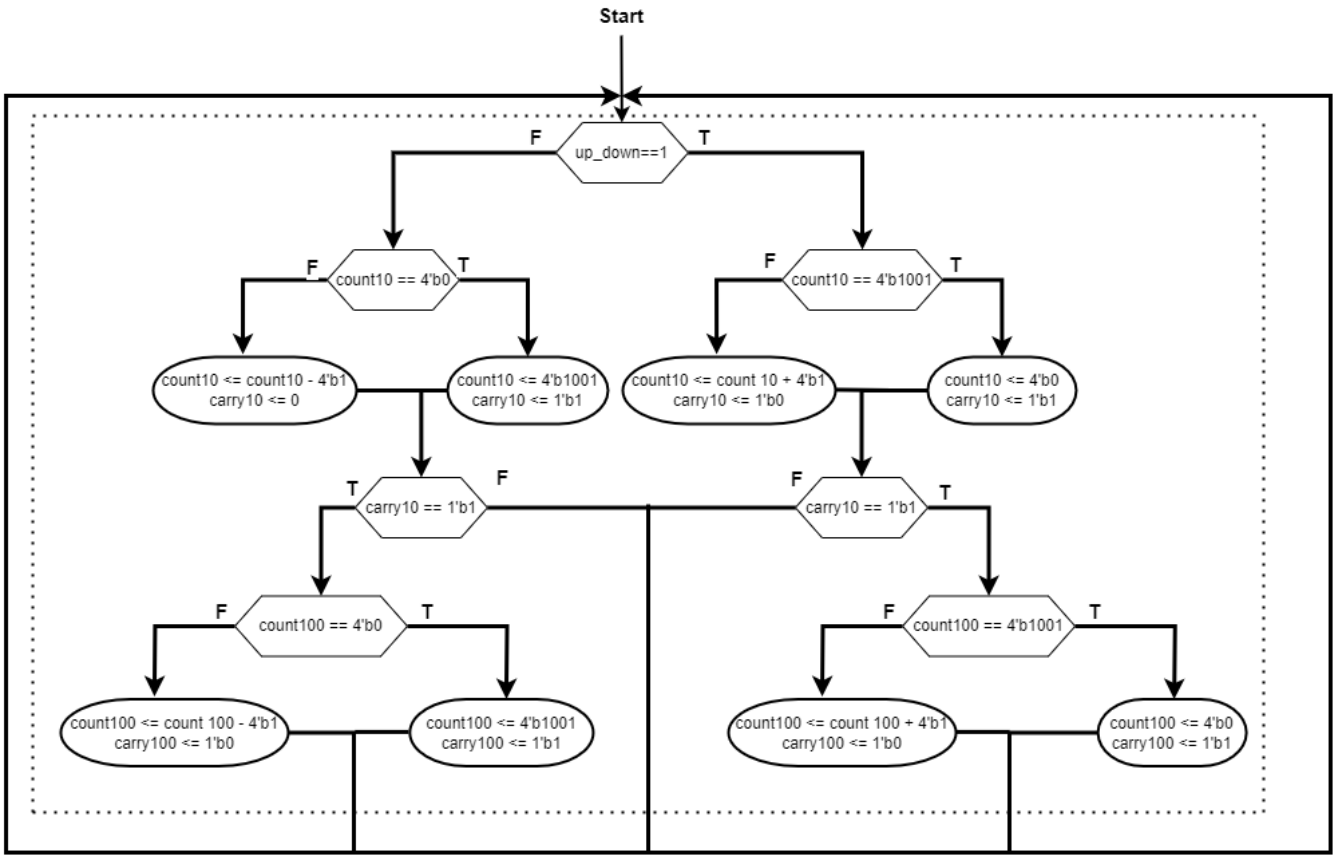
Mạch hoạt động với đầu ra tương ứng so với đầu vào theo nguyên lý hoạt động của mạch. Khi tín hiệu reset ở mức cao, giá trị của q quay về lại giá trị ban đầu là 0 và khi reset ở mức thấp thì nó không tác động đến đầu ra q. Khi tín hiệu đầu vào d ở mức thấp, giá trị q được đếm tăng dần đều, khi q = 9 và vẫn đếm lên thì q được chuyển sang giá trị 0, tín hiệu đầu ra c lúc này = 1. Khi tín hiệu đầu vào d ở mức cao, giá trị q được đếm giảm dần đều, khi q = 0 và vẫn đếm xuống thì q lúc này có giá trị là 9 và tín hiệu c = 1. Thử thay đổi giá trị d liên tự từ 0 lên 1 và 1 về 0 thì giá trị đầu ra q thay đổi theo giá trị đầu vào d đúng theo nguyên lý hoạt động của mạch.

# 3 Counter to 100

## 3.1 Sơ đồ khối của hệ thống


Hình 3.1: Sơ đồ khối của hệ thống

## 3.2 ASM chart




Hình 3.2: ASM chart biểu diễn nguyên lý hoạt động của mạch counter to 100.

## 3.3 Chương trình thiết kế (counter_to_100.v)

```
module counter_to_100(clk, reset, up_down, count10, count100, carry10, carry100);

  input wire clk, reset, up_down;
  output reg carry10, carry100;
  output reg[3:0] count10, count100;

  always @(posedge clk or posedge reset)
  begin
    if(reset) begin
      count10 = 4'b0; count100 = 4'b0;
      carry10 = 1'b0; carry100 = 1'b0;
    end
    if(up_down) begin
      if(count10 == 4'b1001) begin
        carry10 = 1'b1;
        count10 <= 4'b0;
      end
      else begin
        carry10 = 1'b0;
        count10 <= count10 + 4'b1;
      end
      if(carry10 == 1'b1) begin
        if (count100 == 4'b1001) begin
        count100 <= 4'b0;
        carry100 = 1'b1;
        end
        else begin
          count100 <= count100 + 4'b1;
          carry100 = 1'b0;
        end
      end
    end
    else begin
      if(count10 == 4'b0) begin
        carry10 = 1'b1;
        count10 <= 4'b1001;
      end
      else begin
        count10 <= count10 - 4'b1;
        carry10 = 0;
      end
      if(carry10 == 1) begin
        if (count100 == 4'b0) begin
          count100 <= 4'b1001;
          carry100 = 1;
        end
        else begin
          count100 <= count100 - 4'b1;
          carry100 = 0;
        end
      end
    end
  end
endmodule
```

## 3.4 Chương trình Test Bench (counter_to_100_tb.v)

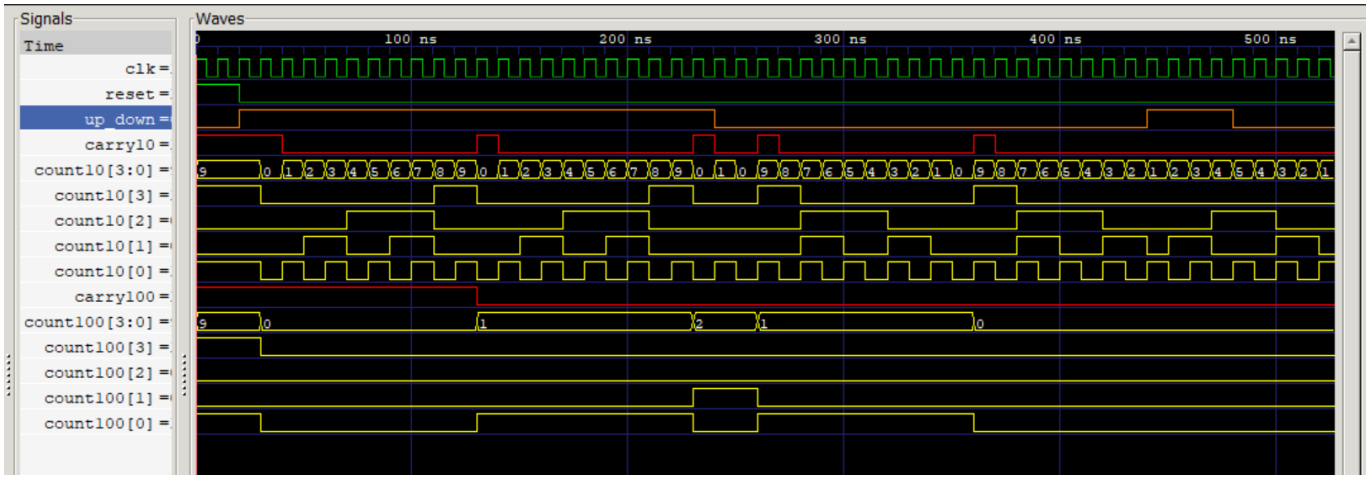
```
1  `timescale 1ns/1ns
2  `include "counter_to_100.v"
3
4  module counter_to_100_tb();
5    reg clk, reset, up_down;
6    wire carry10, carry100;
7    wire[3:0] count10, count100;
8
9    localparam T = 20;
10   //module counter_to_100(clk, reset, up_down, count10, count100, carry10,
       carry100);
11   counter_to_100 countBlock (clk, reset, up_down, count10, count100, carry10,
       carry100);
12
13   always
14   begin
15     clk = 1'b1;
16     #(T/4);
17     clk = 1'b0;
18     #(T/4);
19   end
20
21   initial
22   begin
23   $dumpfile("counter_to_100_tb.vcd");
24   $dumpvars(0, counter_to_100_tb);
25   end
26
27   initial
28   begin
29     #(T/2) reset <= 1;
30     #(T/2); reset <= 0; up_down <=1;
31     #(T*10);
32     //reset <= 1;
33     #(T);up_down <=0;
34     #(T*10);
35     up_down <= 1;
36     #(T*2) up_down <= 0;
37     #(T*4);$finish;
38   end
39 endmodule
```

## 3.5 Kết quả mô phỏng trên phần mềm Icarus

Dạng sóng mô phỏng được biểu diện tại hình 3.3 tại trang 12.

## 3.6 Kết luận

Mạch hoạt động với đầu ra tương ứng so với đầu vào theo nguyên lý hoạt động của mạch. Khi tín hiệu reset ở mức cao, giá trị của count10 và count100 quay về lại giá trị ban đầu là 0 và khi reset ở mức thấp thì nó không tác động đến giá trị các đầu ra. Khi tín hiệu đầu vào up_down ở mức thấp, giá trị count10 và count100 được đếm giảm dần đều, khi count10 = 0 và vẫn đếm xuống thì count10 được chuyển sang giá trị 9, tín hiệu đầu ra carry10 lúc này = 1, count100 sẽ tăng lên 1 giá




Hình 3.3: Dạng sóng tín hiệu mô phòng mạch counter to 100.

trị, tương tự như count10, nếu count100 = 0 thì nó sẽ chuyển về 9 và carry100 = 1. Khi tín hiệu đầu vào up_down ở mức cao, giá trị count10 và count100 được đếm tăng dần đều, khi count10 = 9 và vẫn đếm lên thì count10 được chuyển sang giá trị 0, tín hiệu đầu ra carry10 lúc này = 1, count100 sẽ tăng lên 1 giá trị, nếu count100 =9 thì nó sẽ chuyển về 0 và carry100 = 1. Thử thay đổi giá trị d liên tự từ 0 lên 1 và 1 về 0 thì giá trị các đầu ra thay đổi theo giá trị đầu vào đúng theo nguyên lý hoạt động của mạch.