|  |  |  |  |
| --- | --- | --- | --- |
| Họ tên sinh viên | MSSV | Lớp (thứ - tiết) |  |
|  |  |  |

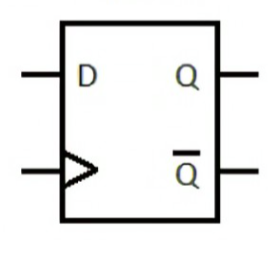
Chú ý: Sinh viên thay bằng QR code của mã số sinh viên (ví dụ: 23119012), có thể tham khảo tại https://barcode.tec-it.com)

Quick question : chapter 5

*Lưu ý: Trong mỗi thiết kế yêu cầu sinh viên thực hiện*

* *Sơ đồ khối (nguyên lý, cấu trúc)*
* *Bảng trạng thái*
* *Mô tả bằng ngôn ngữ Verilog cho module cần thiết kế,*
* *Mô tả Verilog cho module dùng để kiểm tra thiết kế*
* *Kết quả mô phỏng quá trình kiểm tra, có phân tích*
* *Module test được đặt tên theo cú pháp: tensv\_testbench\_tenmodule, ví dụ để test module encoder, sinh viên Nguyen Van An phải đặt tên module test như sau: An\_testbench\_encoder. Các kết quả mô phỏng phải được chụp màn hình bao gồm cả tên của module test trong đó có tên sinh viên thì mới hợp lệ*

1. Thiết kế và mô phỏng kiểm chứng mạch Flip Flop D



Bảng trạng thái Q(n+1) = D(n)

|  |  |  |
| --- | --- | --- |
| Clk | D | Q(n+1) |
| 0 | x | Q(n) |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

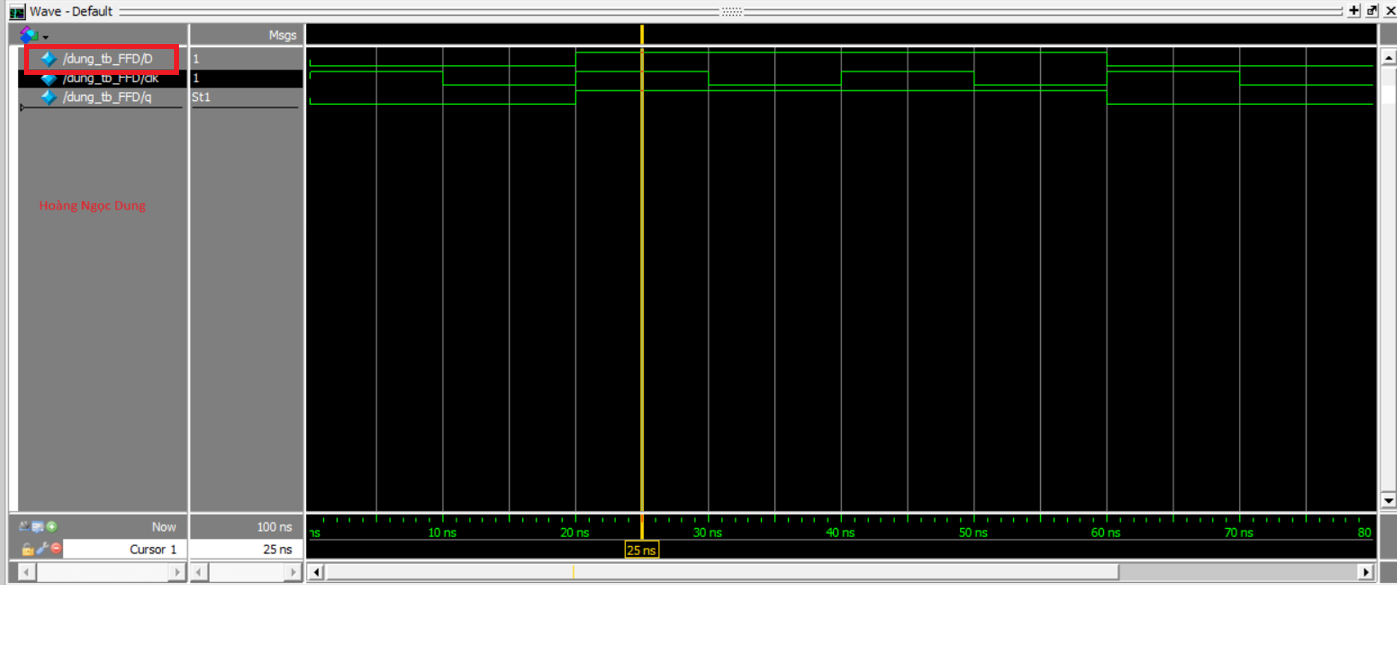
Mô tả bằng ngôn ngữ Verilog

|  |
| --- |
| module FF\_D(input wire d,clk, output reg q);  always @(posedge clk) // positive edge  begin      q <= d // Non-Blocking Assignment  end  endmodule |

Mô tả Verilog cho module dùng để kiểm tra thiết kế

|  |
| --- |
| `timescale 1ns/1ns // define timescale  module dung\_tb\_FFD();  reg D;  reg clk;  wire q;  initial begin  clk = 1;  D = 0;  end  always forever #20 clk = ~clk;  always forever #40 D = ~D;  FF\_D m0(D,clk,q);  endmodule |

Kết quả mô phỏng



1. Thiết kế và mô phỏng kiểm chứng thanh ghi dịch 8 bit sử dụng FlipFlop D

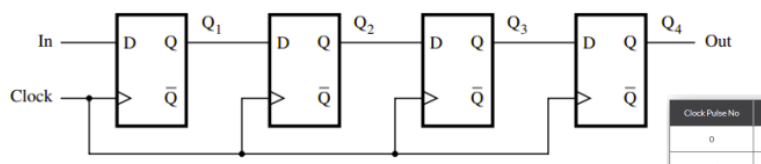


Figure 1: 4-bit SISO shift register(tương tự cho 8 – bit)

Bảng trạng thái

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Clk Pulse | D | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 |
| 0 | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 5 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

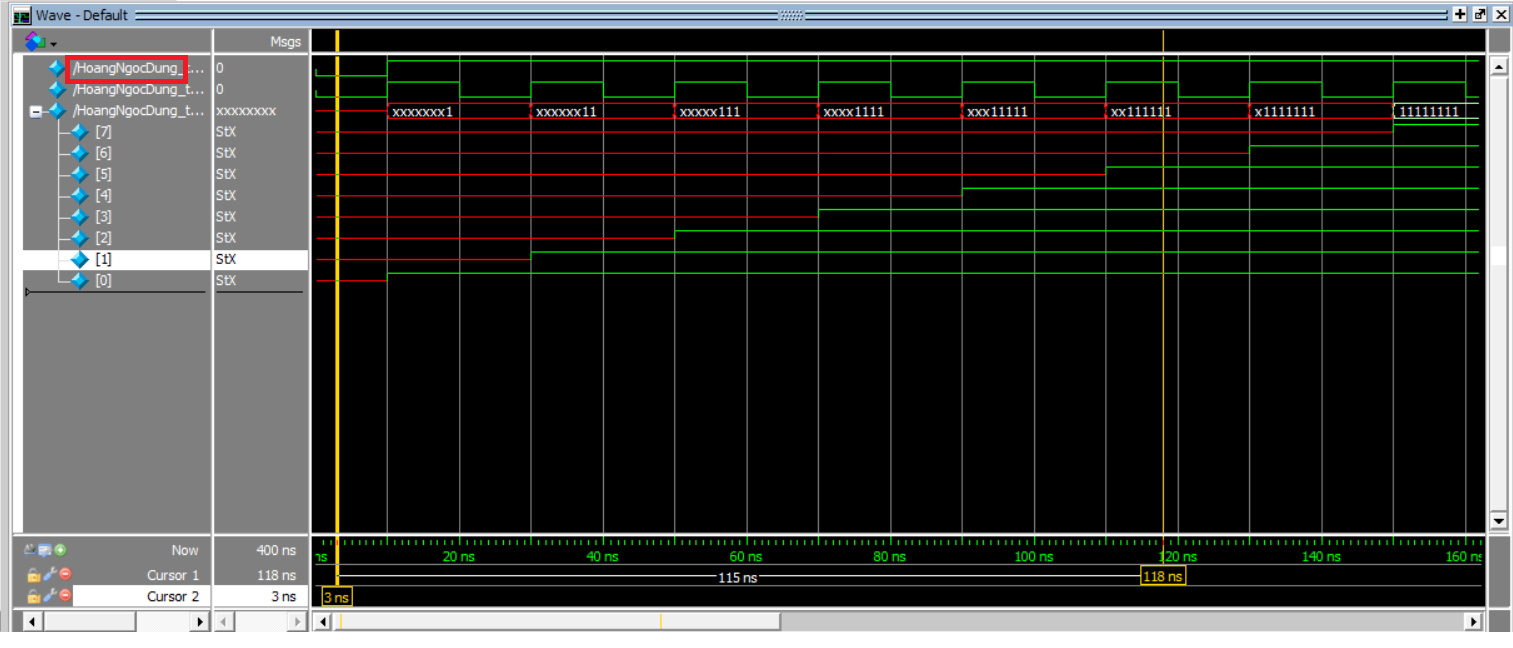
Mô tả bằng ngôn ngữ Verilog

|  |
| --- |
| module FF\_D(input wire d, clk, output reg q);    always @(posedge clk)      q <= d;  endmodule  module SR\_8bit(input wire in, clk, output wire[7:0] out);      FF\_D F0 (in, clk, out[0]);    FF\_D F1 (out[0], clk, out[1]);    FF\_D F2 (out[1], clk, out[2]);    FF\_D F3 (out[2], clk, out[3]);    FF\_D F4 (out[3], clk, out[4]);    FF\_D F5 (out[4], clk, out[5]);    FF\_D F6 (out[5], clk, out[6]);    FF\_D F7 (out[6], clk, out[7]);  endmodule |

Mô tả Verilog cho module dùng để kiểm tra thiết kế

|  |
| --- |
| `timescale 1ns/1ns // define timescale  module HoangNgocDung\_tb\_FFD\_SR\_8bit();  reg in;  reg clk;  wire [7:0] out;  initial begin  clk = 0;  in = 0;  end  always #10 clk = ~clk;  always #10 in = 1;  SR\_8bit m0 (in,clk,out);  endmodule |

Kết quả mô phỏng



1. Thiết kế và mô phỏng kiểm chứng mạch đếm đồng bộ 8 bit sử dụng FlipFlop T

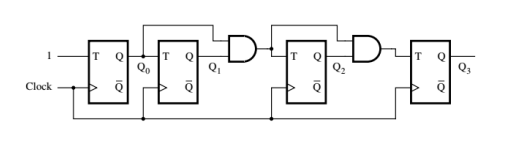


Figure 2: Mạch đếm đồng bộ 8 bit này sẽ sử dụng T-Flip-Flop (T-FF) và sẽ đếm từ 00000000 đến 11111111 (từ 0 đến 255 trong hệ thập phân). Mỗi xung đồng hồ sẽ làm tăng giá trị của bộ đếm lên 1.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Clock Pulse No | Q7 | Q6 | Q5 | Q4 | Q3 | Q2 | Q1 | Q0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 6 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 8 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 10 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 11 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 12 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 14 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 15 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 16 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 255 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Mô tả bằng ngôn ngữ Verilog

|  |
| --- |
| module t\_flip\_flop (    input t, clk,    output reg q,qb  );  initial  begin  q = 0;  qb = 1;  end  always @(posedge clk)      if (t) begin        q = ~q;        qb = !qb;      end  endmodule  module up\_down\_counter\_8bit (    input clk, up,    output [7:0] count  );    wire t0, t1, t2, t3, t4, t5, t6, t7;    wire q0, q1, q2, q3, q4, q5, q6, q7;    // T flip-flop logic for the counter    assign t0 = 1;    assign t1 = (up) ? q0 : ~q0;    assign t2 = (up) ? (q0 & q1) : (~q0 & ~q1);    assign t3 = (up) ? (q0 & q1 & q2) : (~q0 & ~q1 & ~q2);    assign t4 = (up) ? (q3 & q2 & q1 & q0) : (~q0 & ~q1 & ~q2 & ~q3);    assign t5 = (up) ? (q4 & q3 & q2 & q1 & q0) : (~q0 & ~q1 & ~q2 & ~q3 & ~q4);    assign t6 = (up) ? (q5& q4 & q3 & q2 & q1 & q0) : (~q0 & ~q1 & ~q2 & ~q3 & ~q4 & ~q5);    assign t7 = (up) ? (q6 & q5& q4 & q3 & q2 & q1 & q0) : (~q0 & ~q1 & ~q2 & ~q3 & ~q4 & ~q5 & ~q6);    t\_flip\_flop ff0  (t0, clk, q0);    t\_flip\_flop ff1  (t1, clk, q1);    t\_flip\_flop ff2  (t2, clk, q2);    t\_flip\_flop ff3  (t3, clk, q3);    t\_flip\_flop ff4  (t4, clk, q4);    t\_flip\_flop ff5  (t5, clk, q5);    t\_flip\_flop ff6  (t6, clk, q6);    t\_flip\_flop ff7  (t7, clk, q7);    assign count = {q7, q6, q5, q4, q3, q2, q1, q0};  endmodule |

Mô tả Verilog cho module dùng để kiểm tra thiết kế

|  |
| --- |
| `timescale 1ns/1ns // define timescale  module HoangNgocDung\_tb\_up\_down\_FFT();  reg clk;  reg up;  wire [7:0] count;  initial begin    clk = 0;    up = 0;  end  always forever #20 clk = ~clk;  always forever #50 up = 1;  up\_down\_counter\_8bit counter (    .clk(clk),    .up(up),    .count(count)  );  endmodule |

Kết quả mô phỏng