Buổi 2

**GIỚI THIỆU VỀ RTL DESIGN VÀ MẠCH SỐ**

|  |  |  |
| --- | --- | --- |
| **TT** | **Họ và tên** | **Lớp** |
| 1 | Phạm Nguyễn Cao Triều | RDBL12 |

**LAB\_1:**

|  |
| --- |
| **File mô phỏng** |
| module count4bit\_DFF\_st (input clk, input reset, output reg [3:0] count);  always @(posedge clk or posedge reset)  begin  if (reset)  count <= 4'b0000;  else  count <= count + 1;  end  endmodule |
| **File testbench** |
| module count4bit\_DFF\_tb;  reg clk, reset;  wire [3:0] count;  count4bit\_DFF\_st uut ( .clk(clk), .reset(reset), .count(count) );  initial begin  clk = 0;  forever #5 clk = ~clk;  end  initial begin  reset = 1;  #10 reset = 0;  #110 reset = 1;  #10 reset = 0;  end  endmodule |

|  |
| --- |
| **Kết quả mô phỏng** |
|  |

**LAB\_2:**

|  |
| --- |
| **File mô phỏng** |
| module shift4bit\_DFF\_st (input clk, input reset, input D, output reg [3:0] Q);  always @(posedge clk or posedge reset) begin  if (reset)  Q <= 4'b0000;  else  Q <= {Q[2:0], D};  end  endmodule |
| **File testbench** |
| module shift4bit\_DFF\_tb;  reg D, clk, reset;  wire [3:0] Q;  shift4bit\_DFF\_st uut ( .clk(clk),.reset(reset), .D(D), .Q(Q) );  initial begin  D = 0;  forever #7.5 D = ~D;  end  initial begin  clk = 0;  forever #5 clk = ~clk;  end  initial begin  reset = 1;  forever #70 reset = ~reset;  end  endmodule |
| **Kết quả mô phỏng** |
|  |

**LAB 3:**

|  |
| --- |
| **File mô phỏng** |
| module shift8bit\_SAVE\_st (input clk, input reset, input D, output reg [7:0] Q);  always @(posedge clk or posedge reset) begin  if (reset)  Q <= 8'h00;  else  Q <= {Q[6:0], D};  end  endmodule |
| **File testbench** |
| module shift8bit\_SAVE\_tb;  reg D, clk, reset;  wire [7:0] Q;  shift8bit\_SAVE\_st uut ( .clk(clk),.reset(reset), .D(D), .Q(Q) );  initial begin  D = 0;  forever #6 D = ~D;  end  initial begin  clk = 0;  forever #5 clk = ~clk;  end  initial begin  reset = 1;  forever #250 reset = ~reset;  end  endmodule |
| **Kết quả mô phỏng** |
|  |

**HW 1:**

|  |
| --- |
| **File mô phỏng** |
| module and\_gate\_st(input A, input B, output Y);  assign Y = A&B;  endmodule |
| **File testbench** |
| module and\_gate\_tb;  reg A,B;  wire Y;  and\_gate\_st uut (.A(A), .B(B), .Y(Y));  initial begin  A = 0; B = 0;  #10 A = 0; B = 1;  #10 A = 1; B = 0;  #10 A = 1; B = 1;  //#10 $finish;  end  endmodule |
| **Kết quả mô phỏng** |
|  |

**HW 2:**

|  |
| --- |
| **File mô phỏng** |
| module count\_4bit\_DOWN\_st (input clk, input reset, input D, output reg [3:0] Q);  always @(posedge clk or posedge reset) begin  if (reset)  count <= 4'b1111;  else  count <= count - 1;  end  endmodule |
| **File testbench** |
| module count\_4bit\_DOWN\_tb;  reg clk, reset;  wire [3:0] count;  count\_4bit\_DOWN\_st uut ( .clk(clk), .reset(reset), .count(count) );  initial begin  clk = 0;  forever #5 clk = ~clk;  end  initial begin  reset = 1;  forever begin  #10 reset = 0;  #110 reset = 1;  #10 reset = 0;  end  end  endmodule |
| **Kết quả mô phỏng** |
|  |