**Thực hành Thiết kế hệ thống số với HDL**

**CE213.L21.MTCL**

**Họ và tên sinh viên - Mã số sinh viên:**

**Phạm Quang Hải - 18520703**

**LAB03: THIẾT KẾ MẠCH TUẦN TỰ**

**Mô tả máy trạng thái kiểu Moore và Mealy**

Xác định Input, Output, chức năng của mạch?

* Mạch tuần tự này có chức năng phát hiện chuỗi số Input đầu vào liên tiếp khác nhau thì ngõ ra Output = 1.

1. **Kiểu Moore** 
   1. **Sơ đồ khối**

Q

D

Output

Input

CLK

Output Logic

Current State

Next state

* 1. **Mô tả chức năng**
* Nhập tuần tự chuỗi số bất kỳ từ bàn phím, nếu đúng chuỗi mã số sinh viên (0703) thì output = 1 tại trạng thái cuối.
* Bảng chuyển trạng thái:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| State/Input | =0 | =7 | =3 | Output |
| S0 | S1 |  |  | 0 |
| S1 | S1 | S2 |  | 0 |
| S2 | S3 |  |  | 0 |
| S3 | S1 | S2 | S4 | 0 |
| S4 | S1 |  |  | 1 |

* Từ bảng trên ta được:

D2 = I2’I1I0.Q2’Q1Q0

D1 = I2I1I0.Q2’Q0 + I2’I1’I0’.Q2’Q1Q0’

D0 = I2’I1’I0 (Q2’ + Q1’Q0’)

* 1. **Mô tả Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên port** | **Bit width** | **Loại** | **Mô tả** |
| 1 | out | 1 | Output |  |
| 2 | in | 3 | Input |  |
| 3 | clk | 1 | Input |  |
| 4 | clear | 1 | Input | Reset mức thấp sử dụng cho D Flipflop structural  để set giá trị về 0 trước khi sử dụng |

* 1. **Code**

**.** *// Mạch gồm module d\_ff theo kiểu behavior, 2 khối next state, outputlogic của máy trạng thái được chia nhỏ thành 2 module riêng*

*module d\_ff(q,d,clk,clear);*

*output reg q;*

*input wire d,clk,clear;*

*always@(posedge clk or negedge clear) begin*

*if(clear==1'b0) begin*

*q<=1'b0;*

*end else begin*

*q<=d;*

*end*

*end*

*endmodule*

*module nextstate(out,in,outq);*

*output wire [2:0] out;*

*input wire [2:0] in,outq;*

*wire [2:0]in\_bar,outq\_bar;*

*wire outand1,outand2,outand3,outor1;*

*not not1(in\_bar[2],in[2]);*

*not not3(in\_bar[1],in[1]);*

*not not4(in\_bar[0],in[0]);*

*not not2(outq\_bar[2],outq[2]);*

*not not5(outq\_bar[1],outq[1]);*

*not not6(outq\_bar[0],outq[0]);*

*and d2(out[2],in\_bar[2],in[1],in[0],outq\_bar[2],outq[1],outq[0]);*

*and and1(outand1,in[2],in[1],in[0],outq\_bar[2],outq[0]);*

*and and2(outand2,in\_bar[2],in\_bar[1],in\_bar[0],outq\_bar[2],outq[1],outq\_bar[0]);*

*or d1(out[1],outand1,outand2);*

*and and3(outand3,outq\_bar[1],outq\_bar[0]);*

*or or1(outor1,outand3,outq\_bar[2]);*

*and d0(out[0],outor1,in\_bar[2],in\_bar[1],in\_bar[0]);*

*endmodule*

*module outputlogic(out,in);*

*output wire out;*

*input wire [2:0] in;*

*wire [1:0]in\_bar;*

*not not1(in\_bar[1],in[1]);*

*not not2(in\_bar[0],in[0]);*

*and and1(out,in[2],in\_bar[1],in\_bar[0]);*

*endmodule*

*module moore(out,in,clear,clk);*

*output wire out;*

*input wire [2:0] in;*

*input wire clk,clear;*

*wire [2:0] outq,ind;*

*d\_ff d\_ff\_inst[2:0](*

*.q(outq),*

*.d(ind),*

*.clk(clk),*

*.clear(clear)*

*);*

*nextstate nextstate\_inst(*

*.out(ind),*

*.in(in),*

*.outq(outq)*

*);*

*outputlogic outputlogic\_inst(*

*.out(out),*

*.in(outq)*

*);*

*endmodule*

* 1. **Testbench**

*.*

*//Module golden check máy moore:*

*module moore\_golden(out,state,in,clk,clear);*

*input [2:0] in;*

*input clk;*

*input clear;*

*output out;*

*output reg [2:0] state;*

*reg [2:0] next\_state;*

*parameter s0=3'b000;*

*parameter s1=3'b001;*

*parameter s2=3'b010;*

*parameter s3=3'b011;*

*parameter s4=3'b100;*

*always @(state or in) begin*

*case (state)*

*s0:*

*if (in==3'd0)*

*next\_state = s1;*

*else*

*next\_state = s0;*

*s1:*

*if (in==3'd7)*

*next\_state = s2;*

*else if (in==3'd0)*

*next\_state = s1;*

*else*

*next\_state = s0;*

*s2:*

*if (in==3'd0)*

*next\_state = s3;*

*else*

*next\_state = s0;*

*s3:*

*if (in == 3'd3)*

*next\_state = s4;*

*else if (in==3'd0)*

*next\_state = s1;*

*else if (in==3'd7)*

*next\_state = s2;*

*else*

*next\_state = s0;*

*s4:*

*if (in==3'd0)*

*next\_state = s1;*

*else*

*next\_state = s0;*

*default:*

*next\_state = s0;*

*endcase // case(state)*

*end*

*always @(posedge clk) begin*

*if (clear == 0)*

*state <= s0;*

*else*

*state <= next\_state;*

*end*

*assign out = (state == s4) ? 1: 0;*

*endmodule*

*//testbench*

*`timescale 1ns/1ps*

*module testbench();*

*wire out;*

*wire out\_t;*

*wire [2:0] state\_DUT,state\_golden;*

*reg [2:0] in;*

*reg clk,clear;*

*moore\_golden moore\_golden\_inst(*

*.out(out\_t),*

*.state(state\_golden),*

*.in(in),*

*.clk(clk),*

*.clear(clear)*

*);*

*moore moore\_DUT(*

*.out(out),*

*.outq(state\_DUT),*

*.in(in),*

*.clk(clk),*

*.clear(clear)*

*);*

*task check;*

*input [2:0] in\_c;*

*input out\_c,out\_t\_c;*

*input [2:0] state\_DUT\_c,state\_golden\_c;*

*begin*

*if(state\_DUT\_c==3'b100 && state\_golden\_c ==3'b100) begin*

*if (out\_c != out\_t\_c) begin*

*$display("fail");*

*end*

*else begin*

*$display("da phat hien 0703");*

*$display("state moore\_DUT = ",state\_DUT\_c);*

*$display("state moore\_golden = ",state\_golden\_c);*

*$display("out\_DUT = ",out\_c);*

*$display("expected out = ",out\_t\_c);*

*$display("=> ----Correct Function!----");*

*end*

*end*

*end*

*endtask*

*initial begin*

*clear = 1'b1;*

*clk = 1'b0;*

*$display("-------may trang thai moore phat hien chuoi ma so sinh vien: 0703");*

*$display("chuoi so nhap tu ban phim: ");*

*#30*

*in = 3'd6;*

*#60*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd0;*

*#60*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd7;*

*#60*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd0;*

*#60*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd7;*

*#60*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd0;*

*#60*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd3;*

*$display(in);*

*#1500 $finish;*

*end*

*always @(clk) begin*

*#30 clk <= ~clk;*

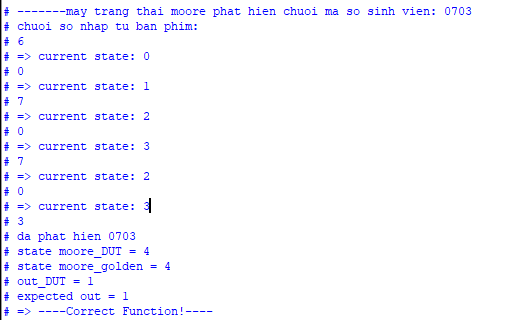
*end*

*always@(in or out) begin*

*check(in,out,out\_t,state\_DUT,state\_golden);*

*end*

*endmodule*

* Kết quả:

1. **Kiểu Mealy** 
   1. **Sơ đồ khối**

Input

Q

D

Output

CLK

Output Logic

Current State

Next state

* 1. **Mô tả chức năng**
* Nhập tuần tự chuỗi số bất kỳ từ bàn phím, nếu đúng chuỗi mã số sinh viên (0703) thì output = 1 ngay khi có tín hiệu Input = 3.
* Bảng chuyển trạng thái:

|  |  |  |  |
| --- | --- | --- | --- |
| State/Input | =0 | =7 | =3 |
| S0 | S1/0 |  |  |
| S1 | S1/0 | S2/0 |  |
| S2 | S3/0 |  |  |
| S3 | S1/0 | S2/0 | S0/1 |

* Từ bảng trên ta được:

D1 = I2I1I0.(Q1 xor Q0) + I2’I1’I0’Q1Q0

D0 = I2’I1’I0’Q1’Q0’ + I2’I1’I0’(Q1 xor Q0) + I2I1I0(Q1 xor Q0)

* 1. **Mô tả Input/Output**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên port** | **Bit width** | **Loại** | **Mô tả** |
| 1 | out | 1 | Output |  |
| 2 | in | 3 | Input |  |
| 3 | clk | 1 | Input |  |
| 4 | clear | 1 | Input | Reset mức thấp sử dụng cho D Flipflop structural  để set giá trị về 0 trước khi sử dụng |

* 1. **Code**

**.** *// Mạch gồm module d\_ff theo kiểu behavior, 2 khối next state, outputlogic của máy trạng thái được chia nhỏ thành 2 module riêng*

*module d\_ff(q,d,clk,clear);*

*output reg q;*

*input wire d,clk,clear;*

*always@(posedge clk or negedge clear) begin*

*if(clear==1'b0) begin*

*q<=1'b0;*

*end else begin*

*q<=d;*

*end*

*end*

*endmodule*

*module nextstate\_mealy(out,in,outq);*

*output wire [1:0] out;*

*input wire [2:0] in; //in = input*

*input wire [1:0] outq; //outq = out của khối current state*

*wire [2:0] in\_bar;*

*wire outand1,outand2,outand3,outand4,outand6,outand7,outand8,outxor1;*

*wire [1:0] outq\_bar;*

*not not11(in\_bar[2],in[2]);*

*not not22(in\_bar[1],in[1]);*

*not not33(in\_bar[0],in[0]);*

*not not44(outq\_bar[1],outq[1]);*

*not not55(outq\_bar[0],outq[0]);*

*and and11(outand1,in[2],in[1],in[0]);*

*and and22(outand2,in\_bar[2],in\_bar[1],in\_bar[0]);*

*xor xor1(outxor1,outq[1],outq[0]);*

*and and33(outand3,outand1,outxor1);*

*and and44(outand4,outand2,outq[1],outq[0]);*

*or d1\_mealy(out[1],outand4,outand3);*

*and and66(outand6,outand2,outq\_bar[1],outq\_bar[0]);*

*and and77(outand7,outand2,outxor1);*

*and and88(outand8,outand1,outxor1);*

*or d0\_mealy(out[0],outand6,outand7,outand8);*

*endmodule*

*module outputlogic\_mealy(out,in,outq);*

*output wire out;*

*input wire [1:0] outq;*

*input wire [2:0] in;*

*wire [2:0]in\_bar;*

*wire [1:0]outq\_bar;*

*not notq(outq\_bar[0],outq[0]);*

*not notin(in\_bar[2],in[2]);*

*and and1(out,in\_bar[2],in[1],in[0],outq[1],outq\_bar[0]);*

*endmodule*

*module mealy(out,in,clear,clk);*

*output wire out;*

*input wire [2:0] in;*

*input wire clk,clear;*

*wire [1:0] outq,ind;*

*d\_ff d\_ff\_inst[1:0](*

*.q(outq),*

*.d(ind),*

*.clk(clk),*

*.clear(clear)*

*);*

*nextstate\_mealy nextstate\_ml\_inst(*

*.out(ind),*

*.in(in),*

*.outq(outq)*

*);*

*outputlogic\_mealy outputlogic\_ml\_inst(*

*.out(out),*

*.in(in),*

*.outq(outq)*

*);*

*endmodule*

* 1. **Testbench**

*//module golden check máy mealy:*

*module mealy\_golden(out,state,in,clk,clear);*

*input [2:0] in;*

*input clk;*

*input clear;*

*output out;*

*output reg [1:0] state;*

*reg [1:0] next\_state;*

*parameter s0=2'b00;*

*parameter s1=2'b01;*

*parameter s2=2'b11;*

*parameter s3=2'b10;*

*always @(state or in) begin*

*case (state)*

*s0:*

*if (in==3'd0)*

*next\_state = s1;*

*else*

*next\_state = s0;*

*s1:*

*if (in==3'd7)*

*next\_state = s2;*

*else if (in==3'd0)*

*next\_state = s1;*

*else*

*next\_state = s0;*

*s2:*

*if (in==3'd0)*

*next\_state = s3;*

*else*

*next\_state = s0;*

*s3:*

*if (in==3'd0)*

*next\_state = s1;*

*else if (in==3'd7)*

*next\_state = s2;*

*else*

*next\_state = s0;*

*default:*

*next\_state = s0;*

*endcase*

*end*

*always @(posedge clk) begin*

*if (clear == 0)*

*state <= s0;*

*else*

*state <= next\_state;*

*end*

*assign out = (state == s3 && in == 3’d3) ? 1: 0;*

*endmodule*

*//testbench*

*`timescale 1ns/1ps*

*module testbench\_mealy();*

*wire out;*

*wire out\_t;*

*wire [1:0] state\_DUT,state\_golden;*

*reg [2:0] in;*

*reg clk,clear;*

*mealy\_golden golden\_inst2(*

*.out(out\_t),*

*.state(state\_golden),*

*.in(in),*

*.clk(clk),*

*.clear(clear)*

*);*

*mealy mealy\_DUT(*

*.out(out),*

*.outq(state\_DUT),*

*.in(in),*

*.clk(clk),*

*.clear(clear)*

*);*

*task check;*

*input [2:0] in\_c;*

*input out\_c,out\_t\_c;*

*input [1:0] state\_DUT\_c,state\_golden\_c;*

*begin*

*if(state\_DUT\_c==2'b10 && state\_golden\_c ==2'b10 && in\_c == 3'd3) begin*

*if (out\_c != out\_t\_c) begin*

*$display("fail");*

*end*

*else begin*

*#60*

*$display("da phat hien 0703");*

*$display("state mealy\_DUT = ",state\_DUT\_c);*

*$display("state mealt\_golden = ",state\_golden\_c);*

*$display("out\_DUT = ",out\_c);*

*$display("expected out = ",out\_t\_c);*

*$display("=> ----Correct Function!----");*

*end*

*end*

*end*

*endtask*

*initial begin*

*clear = 1'b1;*

*clk = 1'b0;*

*$display("-------may trang thai mealy phat hien chuoi ma so sinh vien: 0703");*

*$display("chuoi so nhap tu ban phim: ");*

*#30*

*in = 3'd6;*

*#45*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd0;*

*#45*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd7;*

*#45*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd0;*

*#60*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd7;*

*#75*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd0;*

*#45*

*$display(in);*

*$display("=> current state: ",state\_DUT);*

*in = 3'd3;*

*$display(in);*

*#1500 $finish;*

*end*

*always @(clk) begin*

*#30 clk <= ~clk;*

*end*

*always@(in or out) begin*

*check(in,out,out\_t,state\_DUT,state\_golden);*

*end*

*endmodule*

* Graphical user interface, text, application

  Description automatically generatedKết quả: