# 实验8—同步时序电路典型设计实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名： | 赵冰骞 | | 学号： | 3090103420 | | | 专业： | 理科试验班 | | |
| 课程名称： | | 逻辑与计算机设计基础实验 | | | 同组学生姓名： | 朱里 | | | |
| 实验时间： | | 2010-11-26 | 实验地点： | | 紫金港东4-509 | | 指导老师： | | 王总辉 | |

# 一、实验目的和要求

1. 掌握典型同步时序电路的工作原理和设计方法
2. 掌握时序电路的激励函数、状态图、状态方程的运用
3. 掌握用Verilog 进行有限状态机的设计、调试、仿真
4. 掌握用FPGA实现时序电路功能

# 二、实验内容和原理

## 2.1 Verilog的结构化描述方法

* 调用内置门原语(在门级结构描述)
* 调用开关级原语(在晶体管开关级结构描述)
* 调用用户定义的原语(在门级结构描述)
* 模块实例(创建层次结构结构描述)

图表1 Verilog HDL 常用原语表

|  |  |  |  |
| --- | --- | --- | --- |
| 逻辑门 | 名称 | 接口 | 示例 |
| 非门 | INV | I,O | INV nQa\_L(.I(Qa), .O(nQa)); |
| 二输入或非门 | NOR2 | I0,I1,O | NOR2 t1(.I0(nQa), .I1(nQb), .O(t2)); |
| 三输入或非门 | NOR3 | I0,I1,I2,O |  |
| 四输入或非门 | NOR4 | I0,I1,I2,I3,O |  |
| 二输入异或非门 | XNOR2 | I0,I1,O | XNOR2 t1(.I0(nQa), .I1(nQb), .O(t2)); |
| 二输入与门 | AND2 | I0,I1,O |  |
| 二输入或门 | OR2 | I0,I1,O |  |
| 二输入与非门 | NAND2 | I0,I1,O |  |
| D触发器 | FD | C,clock  D,data in  Q dataout | FD FD\_A(.C(clk), .D(Da), .Q(Qa));  defparam FD\_A.INIT = 1'b0;  // 定义D触发器的初值为0 |

## 2.2 4位二进制同步计数器

根据D触发器原理，在clk作用下Q = D，4位计数器的Q和D关系如下表：

图表2 4位计数器的Q和D关系

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | QA` | QB | QC | QD | DA | DB | DC | DD |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 3 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 5 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 6 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 7 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 8 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 9 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 10 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 11 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 12 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 13 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 14 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 15 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

激励函数：



进位RC的输出函数



## 2.3 单时钟可逆二进制同步计数器

* 单时钟可逆二进制同步计数器通过控制端S选择正向或者反向计数
  + S = 1时，正向计数，各触发器逻辑表达式同前面
  + S = 0时，反向计数，各触发器逻辑表达式如下式



1. 采用行为描述的优缺点
   * 优点：Verilog代码描述非常直观
   * 缺点：生成的二进制代码效率低
2. 50MHz信号通过25,000,000次分频后，得到1Hz的秒脉冲方波，作为计数器的脉冲输入

# 三、主要仪器设备

1. 装有ISE的计算机系统 1台
2. Spartan-III 开发板 1套

# 四、操作方法与实验步骤

## 4.1 结构化描述方法设计一个4位二进制同步计数器

1. 用Verilog HDL结构化描述方法，编写4位二进制同步计数器代码模块counter\_4bit，并检查语法

|  |
| --- |
| `timescale 1ns / 1ps  module counter\_4bit(clk, Qa, Qb, Qc, Qd, Rc);  input wire clk;  output wire Qa, Qb, Qc, Qd, Rc;  wire Nor\_nQa\_nQb, Nor\_nQa\_nQb\_nQc;    FD FD\_A(.C(clk), .D(Da), .Q(Qa)),  FD\_B(.C(clk), .D(Db), .Q(Qb)),  FD\_C(.C(clk), .D(Dc), .Q(Qc)),  FD\_D(.C(clk), .D(Dd), .Q(Qd));  defparam FD\_A.INIT = 1'b0, FD\_B.INIT = 1'b0;  defparam FD\_C.INIT = 1'b0, FD\_D.INIT = 1'b0;    INV nQa\_L(.I(Qa), .O(nQa)), nQb\_L(.I(Qb), .O(nQb)),  nQc\_L(.I(Qc), .O(nQc)), nQd\_L(.I(Qd), .O(nQd));  assign Da = nQa;    XNOR2 Db\_L(.I0(Qa), .I1(nQb), .O(Db)),  Dc\_L(.I0(Nor\_nQa\_nQb), .I1(nQc), .O(Dc)),  Dd\_L(.I0(Nor\_nQa\_nQb\_nQc), .I1(nQd), .O(Dd));  NOR4 Rc\_L(.I0(nQa), .I1(nQb), .I2(nQc), .I3(nQd), .O(Rc));  NOR2 Nor\_nQa\_nQb\_L (.I0(nQa), .I1(nQb), .O(Nor\_nQa\_nQb));  NOR3 Nor\_nQa\_nQb\_nQc\_L(.I0(nQa), .I1(nQb), .I2(nQc), .O(Nor\_nQa\_nQb\_nQc));  endmodule |

1. 针对上面的计数器模块，编写仿真测试输入代码记录并分析仿真结果，若有错误应返回修改代码

|  |
| --- |
| `timescale 1ns / 1ps  module test;  // Inputs  reg clk = 1'b0, rst = 1'b0;  // Outputs  wire Qa;  wire Qb;  wire Qc;  wire Qd;  wire Rc;  // Instantiate the Unit Under Test (UUT)  counter\_4bit INST(.clk(clk), .Qa(Qa), .Qb(Qb), .Qc(Qc), .Qd(Qd), .Rc(Rc));  parameter PERIOD = 200;  parameter real DUTY\_CYCLE = 0.5;    initial forever begin  // Initialize Inputs  clk = 1'b0;  // Wait 100 ns for global reset to finish  #100;    // Add stimulus here  #(PERIOD-(PERIOD\*DUTY\_CYCLE)) clk = 1'b1;  #(PERIOD\*DUTY\_CYCLE);  end    endmodule |

1. 对计数器模块进行仿真，并分析其行为，比较是否和预期结果一致

## 4.2 在计算器上增加计数器功能

1. 在前一个实验基础上，用Verilog HDL行为描述方法，编写16位二进制可逆同步计数器代码模块counter\_16bit\_rev。

|  |
| --- |
| `timescale 1ns / 1ps  module create\_oprands(  input wire [1:0] switch,  input wire [3:0] btn,  output reg [15:0] op1,op2  );  initial op1 <= 16'b1010\_1011\_1100\_1101; //op1,op2初始化  initial op2 <= 16'b1010\_1011\_1100\_1101; |

|  |
| --- |
| `timescale 1ns / 1ps  module counter\_16bit\_rev(clk, s, cnt);  input wire clk, s;  output reg [15:0] cnt;  always @ (posedge clk) begin  if (s)  cnt <= cnt + 1;  else  cnt <= cnt - 1;  end  endmodule |

1. 编写其他代码。

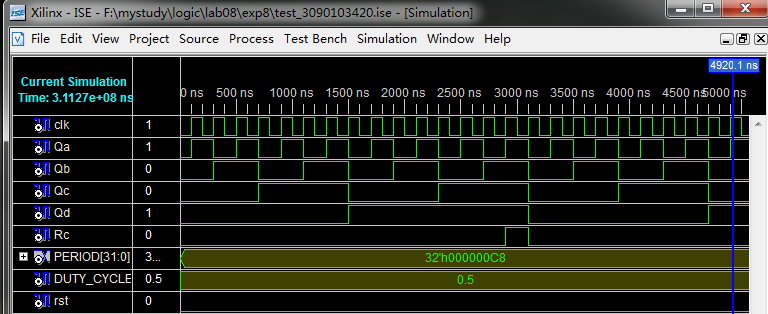
|  |
| --- |
| `timescale 1ns / 1ps  module top(  input wire clk,  input wire [3:0]btn\_in,  input wire[2:0]switch,  output wire [3:0]anode,  output wire [7:0]segment  );    //variable definition: op1, op2,disp\_num…  reg [15:0] display\_num;  wire [15:0] op1,op2,disp\_counter;  wire [3:0] btn\_out;  wire [15:0] result;  pbdebounce b0(clk, btn\_in[0], btn\_out[0]);  pbdebounce b1(clk, btn\_in[1], btn\_out[1]);  pbdebounce b2(clk, btn\_in[2], btn\_out[2]);  pbdebounce b3(clk, btn\_in[3], btn\_out[3]);  display m0(clk, 0, display\_num, 4'b1111, anode, segment); //display module  calculate\_result m1(btn\_out[3:0], op1, op2, result); //calculate result  create\_oprands m2(switch[1:0], btn\_out[3:0], op1, op2); //generate op1/op2    counter\_1s m4(clk,clk\_1s);  counter\_16bit\_rev m5((clk\_1s&btn\_out[0]),switch[2],disp\_counter);  always @\* begin  case (switch[1:0])  2'b01:begin //operand 1  display\_num = op1;  end  2'b10:begin //operand 2  display\_num = op2;  end  2'b00:begin //result  display\_num = result;  end  2'b11:begin //counter  display\_num = disp\_counter;  end  endcase  end  endmodule  module counter\_1s(clk, clk\_1s);  input wire clk;  output reg clk\_1s;  reg [31:0] cnt;  always @ (posedge clk) begin  if (cnt < 25\_000\_000) begin  cnt <= cnt + 1;  end else begin  cnt <= 0;  clk\_1s <= ~clk\_1s;  end  end  endmodule |

1. 编写UCF代码，生成FPGA文件。

|  |
| --- |
| NET "clk" LOC = "t9" ;  NET "btn\_in[0]" LOC = "m13" ;  NET "btn\_in[1]" LOC = "m14" ;  NET "btn\_in[2]" LOC = "l13" ;  NET "btn\_in[3]" LOC = "l14" ;  NET "switch[0]" LOC = "f12" ;  NET "switch[1]" LOC = "g12" ;  NET "switch[2]" LOC = "h14" ;  NET "segment[0]" LOC = "e14" ;  NET "segment[1]" LOC = "g13" ;  NET "segment[2]" LOC = "n15" ;  NET "segment[3]" LOC = "p15" ;  NET "segment[4]" LOC = "r16" ;  NET "segment[5]" LOC = "f13" ;  NET "segment[6]" LOC = "n16" ;  NET "segment[7]" LOC = "p16" ;  NET "anode[0]" LOC = "D14" ;  NET "anode[1]" LOC = "G14" ;  NET "anode[2]" LOC = "F14" ;  NET "anode[3]" LOC = "E13" ; |

1. 将生成的FPGA代码下载到Spartan 3 实验板并调试。
2. 记录、分析实验过程和实验数据，得出实验结论。

# 五、实验结果与分析



图表3 4位二进制同步计数器仿真结果

4位二进制同步计数器调试仿真结果正常

# 六、讨论、心得

本次实验，在上次实验的基础上增加计数器功能，有了上次实验的经验，做得还是比较顺利的。

这次实验的一个不同之处是仿真模拟，需要编写仿真带来来做，而且代码参数要相对应。我在实际操作中，由于没有注意代码添加顺序，造成仿真结果不正常，耽误了不少时间。

这次实验教训我，对于自己不懂的新内容，应该尽快向老师请教，以免不必要的麻烦。