

数字逻辑实验报告 (2)

数字逻辑实验 2		
一、Verilog HDL 设计 设计数字逻辑电路 50%	二、Verilog HDL 设计 较复杂数字逻辑电路 50%	总成绩

评语：（包含：预习报告内容、实验过程、实验结果及分析）

教师签名

姓 名：刘本嵩
学 号：U201614531
班 级：CS1601
指 导 教 师：赵贻竹

计算机科学与技术学院
2018 年 6 月 25 日

华中科技大学

数字逻辑实验报告

Verilog HDL 设计数字逻辑电路预习报告

一、Verilog HDL 设计数字逻辑电路

1、实验名称

Verilog HDL 设计数字逻辑电路。

2、实验目的

要求同学用 Verilog HDL 设计数字逻辑电路，通过 3 个逻辑电路实验，并利用“Vivado 2015.2”软件进行“前、后”仿真检查电路设计，然后在“Xilinx NEXYS 4 开发板”上操作、记录实验结果，最后验证设计是否达到要求。

通过以上设计、仿真、验证 3 个训练过程使同学们掌握 Verilog HDL 设计数字逻辑电路的基本方法，同时掌握如何避免锁存器的产生以及电路设计中的一致性问题的处理方法。

3、实验所用设备

Xilinx NEXYS 4 开发板（芯片为 XC7A100TCSG324-1，封装为 CSG324，软件为 Vivado 2015.2）1 套。

4、实验内容

(1) 组合、时序逻辑电路的“always”设计

(A) 用“always 块”设计纯组合逻辑电路

组合电路的一个特性是它的输出永远受输入变化的影响。也就是说组合电路绝不会保持它们以前的值，即输出不会出现锁存。

在使用 always 块中的 case，if-else 等语句设计纯组合逻辑电路时，要保证所有输入条件，其输出均有输出值，否则有可能会产生锁存器，导致“综合”出错。

例如：某题目要求使用 Verilog 设计实现一个纯组合逻辑电路的选择器，某同学设计了一个带“flag”标识的 4 选 1 的多路选择器，参见程序 1-1 所示，但是在“综合”时，报 3 个错误，即：“Place 30-574、Place 30-99、Common 17-69”。

具体要求：

(a) 验证程序 1-1 在“综合”时，是否会出现上述问题；

《数字电路与逻辑设计》实验报告

(b) 如果存在上述问题，请更正程序 1-1，帮这位同学完成设计。

(a) 会

(b) 已改正

程序 1-1 带“flag”标识的 4 选 1 的多路选择器

```
module mux_latch(
    input  [3:0] data,
    input  [1:0] valid,
    input  flag,
    output reg valid_data);
initial begin
    valid_data=1'b0;
end
always @ (*)
begin
    case(valid)
        2'b00 : begin if(flag) valid_data = data[0];else
valid_data=0;end
        2'b01 : begin if(flag) valid_data = data[1];else
valid_data=0;end
        2'b10 : begin if(flag) valid_data = data[2];else
valid_data=0;end
        2'b11 : begin if(flag) valid_data = data[3];else
valid_data=0;end
    endcase
end
endmodule
```

//////////////////////////////////*.xdc 文件//////////////////////////////////

```
set_property PACKAGE_PIN T16 [get_ports flag]
set_property IOSTANDARD LVCMOS33 [get_ports flag]

set_property PACKAGE_PIN U8 [get_ports {valid[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {valid[0]}]
set_property PACKAGE_PIN R7 [get_ports {valid[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {valid[1]}]

set_property PACKAGE_PIN V7 [get_ports {data[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {data[0]}]
set_property PACKAGE_PIN V6 [get_ports {data[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {data[1]}]
set_property PACKAGE_PIN V5 [get_ports {data[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {data[2]}]
set_property PACKAGE_PIN U4 [get_ports {data[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {data[3]}]
```


后的仿真结果。

(这就是：所谓前仿真和后仿真是否一致的问题)

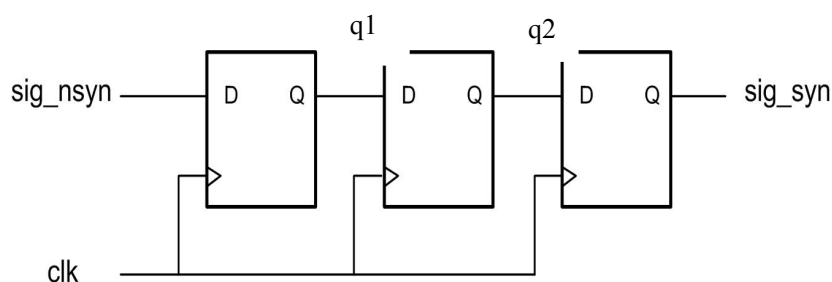


图 1-3 3 位移位寄存器电路

(B) NEXYS 4 开发板提供了一个 100Mhz 的同步时钟，引脚绑定为“E3”，在更正后的程序 1-2 中添加一个时钟分频部分，并将降频后的时钟信号接到图 1-3 中的“clk”，编译成功后再下载到开发板上测试它。

结论：在今后的设计中要保持：“前仿真和后仿真以及下载验证都正确”哟。

程序 1-2 3 位移位寄存器电路

```
module pipe3b(
    input sig_nsyn,
    input clk,
    output q1,
    output q2,
    output sig_syn
);
    reg q1;
    reg q2;
    reg sig_syn;
    always@(posedge clk) q2=q1;
    always@(posedge clk) sig_syn=q2;
    always@(posedge clk) q1=sig_nsyn;
endmodule
```

5、实验方案设计

(1) 组合、时序逻辑电路的“always”设计方案

(A) 用“always 块”设计纯组合逻辑电路

(更正程序 1-1 后的源程序)

```
module mux_latch(
```

《数字电路与逻辑设计》实验报告

```
    input [3:0] data,
    input [1:0] valid,
    input flag,
    output reg valid_data);
initial begin
    valid_data=1'b0;
end
always @ (*)
begin
    case(valid)
        2'b00 : begin if(flag) valid_data = data[0];else
valid_data=0;end
        2'b01 : begin if(flag) valid_data = data[1];else
valid_data=0;end
        2'b10 : begin if(flag) valid_data = data[2];else
valid_data=0;end
        2'b11 : begin if(flag) valid_data = data[3];else
valid_data=0;end
    endcase
end
endmodule
```

(B) 用“always 块”设计实现一个同步时序逻辑电路

(a) 源程序

```
`timescale 1ns / 1ps
```

```
`include "macro.vh"
```

```
module mod1b(
    input clk,
    input sig_nsyn,
    output wire sig_syn
);
```

```
    wire q0, q1;
```

```
    wire one;
```

```
    assign one = 1'b1;
```

```
`ifdef USE_DIVIDER
```

```
    wire long_clk;
```

```
    divider div(clk, long_clk);
```

```
    tff tff1(one, long_clk, q0);
```

```
    tff tff2(sig_nsyn^q0, long_clk, q1);
```

```
`else
```

《数字电路与逻辑设计》实验报告

```
tff tff1(one, clk, q0);
tff tff2(sig_nsyn^q0, clk, q1);
`endif
assign sig_syn = sig_nsyn & ~q1;

endmodule // mod1b
```

(b) 仿真程序

```
`timescale 1ns / 1ps

module mod1b_tb(
);
    reg clk, _x;
    wire _y;
    mod1b_1(clk, _x, _y);

    initial forever #1 clk = ~clk;
    initial begin
        clk = 0;
        _x = 0;
        #10 _x = 1;
    end
endmodule // mod1b_tb
```

(2) 脉冲异步计数器的分析和设计方案

(A) 源程序

```
`timescale 1ns / 1ps
`include "macro.vh"

module mod2(
    input clk,
    output reg res
);

    integer cter = 0;
    `ifdef USE_DIVIDER
    wire long_clk;
    divider div(clk, long_clk);
```



```
always @(posedge long_clk) begin
`else
always @(posedge clk) begin
`endif
    if (cter == 7) begin
        cter <= 0;
        res <= 1'b1;
    end
    else begin
        cter <= cter + 1;
        res <= 1'b0;
    end
end
end
```

```
endmodule // mod2
```

(B) 仿真程序

```
`timescale 1ns / 1ps
```

```
module mod2_tb(
);
    reg clk;
    wire _res;
    mod2_1(clk, _res);
    initial begin
        clk = 0;
        forever #1 clk = ~clk;
    end
endmodule // mod2_tb
```

(C) 引脚约束 (绑定) 程序

```
## Clock signal
##Bank = 35, Pin name = IO_L12P_T1_MRCC_35, Sch name =
CLK100MHZ
set_property PACKAGE_PIN E3 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports clk]
```

《数字电路与逻辑设计》实验报告

```
## LEDs
##Bank = 34, Pin name = IO_L24N_T3_34, Sch name =
LED0
set_property PACKAGE_PIN T8 [get_ports {res}]
set_property IOSTANDARD LVCMOS33 [get_ports {res}]
```

(3) Verilog 设计中一致性问题的解决方案

(A) 源程序

(更正程序 1-2 后的源程序,且前仿真和后仿真以及下载验证都正确)

```
`timescale 1ns / 1ps
`include "macro.vh"

module mod3b(
    input sig_nsyn,
    input clk,
    output wire sig_syn
);

wire q0, q1;
`ifdef USE_DIVIDER
wire long_clk;
divider div(clk, long_clk);

dff dff0(sig_nsyn, long_clk, q0);
`else
dff dff0(sig_nsyn, clk, q0);
`endif
dff dff1(q0, clk, q1);
dff dff2(q1, clk, sig_syn);

endmodule // mod3b
```

(B) 仿真程序

```
`timescale 1ns / 1ps

module mod3b_tb(
);
    reg sig_nsyn, clk;
    wire sig_syn;
    mod3b _1(sig_nsyn, clk, sig_syn);
    initial forever #1 clk = ~clk;
```

《数字电路与逻辑设计》实验报告

```
initial begin
    sig_nsyn = 0;
    clk = 0;
    forever #20 sig_nsyn = ~ sig_nsyn;
end
endmodule // mod3b_tb
```

华中科技大学

数字逻辑实验报告

Verilog HDL 设计较复杂数字逻辑电路

二、Verilog HDL 设计较复杂数字逻辑电路

1、实验名称

Verilog HDL 设计较复杂数字逻辑电路。

2、实验目的

要求同学用 Verilog HDL 设计较复杂的数字逻辑电路，通过 3 个逻辑电路实验，并利用“Vivado 2015.2”软件进行“前、后”仿真检查电路设计，然后在“Xilinx NEXYS 4 开发板”上操作、记录实验结果，最后验证设计是否达到要求。

通过以上设计、仿真、验证 3 个训练过程使同学们掌握 Verilog HDL 设计较复杂数字逻辑电路的基本方法，同时掌握“电路例化”、“模块化”的使用、异步时序逻辑电路的同步化处理以及用状态机设计控制电路。

3、实验所用组件

Xilinx NEXYS 4 开发板（芯片为 XC7A100TCSG324-1，封装为 **CSG324**，软件为 Vivado 2015.2）1 套。

4、实验内容

（1）4 位二进制加法/减法计数器的设计

设计一个能清零、置数和进位/借位输出的加 1/减 1 的 4 位二进制计数器，其结构框图如图 2-1 所示。

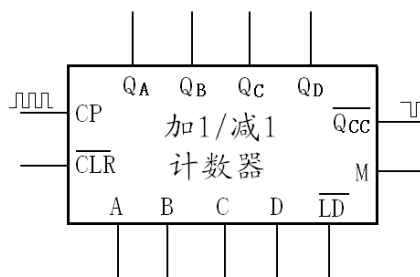


图 2-1 4 位二进制加法/减法计数器

电路输入为计数脉冲 CP、工作模式选择 M、预制初值 D，C，B，A（其中 D 为高位，A 为低位）和预制控制 \overline{LD} ，清零端 \overline{CLR} ；

《数字电路与逻辑设计》实验报告

输出为计数值 Q_D, Q_C, Q_B, Q_A (Q_D 为高位, Q_A 为低位) 和进位/借位输出 $\overline{Q_{cc}}$;

当 \overline{CLR} 为 0 时, 电路输出清零;

预制控制 $\overline{LD}=0$ 时, 将 D、C、B、A 的输入值送到计数器中, 并立即在 Q_D, Q_C, Q_B, Q_A 中输出;

模式选择端 $M=1$ 时加 1 计数;

当 $M=0$ 时减 1 计数;

当 CP 端输入一个上升沿信号时进行一次计数;

计数有进位/借位时 $\overline{Q_{cc}}$ 端输出一个负脉冲。

注意: 用 Verilog 设计电路时, 经常会遇到这样一些问题, 例如:

(A) 用两个 always 块对一个寄存器进行赋值, 无论其中经过了怎样的条件判断, 最终结果毫无疑问是将两个相独立的触发信号连在了寄存器的 CLK 端上, 一个端口接入两信号, 所以这样的语句是无法被综合成电路的。

(B) 某电路如果有多个输入都可能引起输出值的改变, 在使用“always”时, 如果其触发条件为电路的“多个输入”时, 如果语句的“并发性”处理不好, 会造成系统编译成功、“行为仿真”也成功, 但是系统生成不了“bit”文件。

解决方法: 采用所谓“异步时序逻辑电路的同步化处理”, 即: 减少“always”的触发条件。

具体要求:

(A) 用 Verilog HDL 实现该计数器, 将之下载到开发板中进行验证;

(B) 用已实现了的“4 位二进制计数器”, 采用“电路例化”或者“模块化”实现一个初值为 2 的模 8 计数器, 并下载到开发板中进行验证;

(C) 给出设计占用 FPGA 芯片的资源情况 (希望越少越好)。

(2) 采用有限状态机(FSM)实现序列检测器

设计一个简单的状态机, 其功能是检测一个串行的 5 位不可重叠的“10110”二进制序列检测器, 当输入值出现“10110”时, 给出输出标志。

具体要求如下:

(A) 给出不可重叠的“10110”二进制序列的状态转移图;

(B) 采用有限状态机“标准模板”来设计“10110”二进制序列检测器, 在仿真正确后再下载到开发板中进行验证;

(C) 给出设计占用 FPGA 芯片的资源情况 (希望越少越好)。

《数字电路与逻辑设计》实验报告

(所谓有限状态机“标准模板”请参考教材：夏雨闻。**Verilog** 数字系统设计教程第**3**版。北京:北京航空航天大学出版社,**2013**.)

(3) 3 位二进制数值比较器的设计

设计一个 3 位二进制数值比较器。当 $A > B$ 时, $F1=1$, $F2=F3=0$; 当 $A=B$ 时, $F2=1$, $F1=F3=0$; 当 $A < B$ 时, $F3=1$, $F1=F2=0$ 。

具体要求:

- (A) 用 Verilog HDL 设计一个一位二进制数值比较器;
- (B) 用已实现了一位二进制数值比较器, 采用“电路例化”或者“模块化”实现一个 3 位二进制数值比较器;
- (C) 将所设计的电路下载到开发板上进行验证;
- (D) 给出设计占用 FPGA 芯片的资源情况 (希望越少越好)。

5、实验方案设计

(1) 4 位二进制加法/减法计数器的设计方案

(A) 模 16 加 1/减 1 计数器

(a) 源程序

```
`include "macro.vh"
```

```
module mod1 (  
    input [3:0] abcd,  
    input clk,  
    input clr,  
    input mode,  
    input ld,  
    output reg qcc,  
    output reg [3:0] q  
);  
    initial begin  
        q <= abcd;  
    end  
    `ifdef USE_DIVIDER  
        wire long_clk;  
        divider __01(clk, long_clk);
```

```
always @(posedge long_clk) begin
`else
    always @(posedge clk) begin
`endif
    if(ld == 0) begin
        q <= abcd;
        qcc <= 1;
    end else if(clr == 0) begin
        q <= 0;
        qcc <= 1;
    end else if(mode == 1) begin
        if(q == 15 || q == 7) // Warning: fit mod1b
            qcc <= 0;
        else
            qcc <= 1;
            q <= q + 1;
    end else if(mode == 0) begin
        if(q == 0 || q == 8) // Warning: fit mod1b
            qcc <= 0;
        else
            qcc <= 1;
            q <= q - 1;
        end
    end
end
endmodule
```

(b) 仿真程序

```
`timescale 1ns / 1ps

module mod1_tb (

);
    reg clk = 0, clr = 1, mode = 1, ld = 1;
    wire qcc;
    reg [3:0] abcd = 2;
    wire [3:0] q;
    mod1_1(abcd, clk, clr, mode, ld, qcc, q);
```



```
initial forever #1 clk = ~clk;
initial begin
    #20 clr = 0;
    #2 clr = 1;
    #30 ld = 0;
    #2 ld = 1;
    #40 mode = 0;
    #80 mode = 1;
end
endmodule
```

(c) 引脚约束 (绑定) 程序

```
## This file is a general .xdc for the Nexys4 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to
the top level signal names in the project

## Clock signal
##Bank = 35, Pin name = IO_L12P_T1_MRCC_35,          Sch name =
CLK100MHZ
set_property PACKAGE_PIN E3 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports clk]
## Switches
##Bank = 34, Pin name = IO_L21P_T3_DQS_34,          Sch name =
SW0
set_property PACKAGE_PIN U9 [get_ports {abcd[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {abcd[0]}]
##Bank = 34, Pin name = IO_25_34,                    Sch name = SW1
set_property PACKAGE_PIN U8 [get_ports {abcd[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {abcd[1]}]
##Bank = 34, Pin name = IO_L23P_T3_34,                Sch name = SW2
set_property PACKAGE_PIN R7 [get_ports {abcd[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {abcd[2]}]
##Bank = 34, Pin name = IO_L19P_T3_34,                Sch name = SW3
set_property PACKAGE_PIN R6 [get_ports {abcd[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {abcd[3]}]
```

《数字电路与逻辑设计》实验报告

```
##Bank = 34, Pin name = IO_L19N_T3_VREF_34,      Sch name = SW4
set_property PACKAGE_PIN R5 [get_ports {clr}]
set_property IOSTANDARD LVCMOS33 [get_ports {clr}]
##Bank = 34, Pin name = IO_L20P_T3_34,            Sch name = SW5
set_property PACKAGE_PIN V7 [get_ports {mode}]
set_property IOSTANDARD LVCMOS33 [get_ports {mode}]
##Bank = 34, Pin name = IO_L20N_T3_34,            Sch name = SW6
set_property PACKAGE_PIN V6 [get_ports {ld}]
set_property IOSTANDARD LVCMOS33 [get_ports {ld}]
##Bank = 34, Pin name = IO_L10P_T1_34,            Sch name = SW7
#set_property PACKAGE_PIN V5 [get_ports {sw[7]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
##Bank = 34, Pin name = IO_L8P_T1-34,             Sch name = SW8
#set_property PACKAGE_PIN U4 [get_ports {sw[8]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
##Bank = 34, Pin name = IO_L9N_T1_DQS_34,         Sch name = SW9
#set_property PACKAGE_PIN V2 [get_ports {sw[9]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
##Bank = 34, Pin name = IO_L9P_T1_DQS_34,         Sch name = SW10
#set_property PACKAGE_PIN U2 [get_ports {sw[10]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
##Bank = 34, Pin name = IO_L11N_T1_MRCC_34,       Sch name = SW11
#set_property PACKAGE_PIN T3 [get_ports {sw[11]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
##Bank = 34, Pin name = IO_L17N_T2_34,            Sch name = SW12
#set_property PACKAGE_PIN T1 [get_ports {sw[12]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
##Bank = 34, Pin name = IO_L11P_T1_SRCC_34,       Sch name = SW13
#set_property PACKAGE_PIN R3 [get_ports {sw[13]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
##Bank = 34, Pin name = IO_L14N_T2_SRCC_34,       Sch name = SW14
#set_property PACKAGE_PIN P3 [get_ports {sw[14]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
```

《数字电路与逻辑设计》实验报告

```
##Bank = 34, Pin name = IO_L14P_T2_SRCC_34, Sch name = SW15
```

```
#set_property PACKAGE_PIN P4 [get_ports {sw[15]}]
```

```
#set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]
```

```
## LEDs
```

```
##Bank = 34, Pin name = IO_L24N_T3_34, Sch name = LED0
```

```
set_property PACKAGE_PIN T8 [get_ports {qcc}]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports {qcc}]
```

```
###Bank = 34, Pin name = IO_L21N_T3_DQS_34, Sch name = LED1
```

```
#set_property PACKAGE_PIN V9 [get_ports {res}]
```

```
#set_property IOSTANDARD LVCMOS33 [get_ports {res}]
```

```
###Bank = 34, Pin name = IO_L24P_T3_34, Sch name = LED2
```

```
#set_property PACKAGE_PIN R8 [get_ports {out[2]}]
```

```
#set_property IOSTANDARD LVCMOS33 [get_ports {out[2]}]
```

```
##
```

```
#Bank = 34, Pin name = IO_L23N_T3_34, Sch name = LED3
```

```
set_property PACKAGE_PIN T6 [get_ports {q[0]}]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports {q[0]}]
```

```
##Bank = 34, Pin name = IO_L12P_T1_MRCC_34, Sch name = LED4
```

```
set_property PACKAGE_PIN T5 [get_ports {q[1]}]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports {q[1]}]
```

```
##Bank = 34, Pin name = IO_L12N_T1_MRCC_34, Sch name = LED5
```

```
set_property PACKAGE_PIN T4 [get_ports {q[2]}]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports {q[2]}]
```

```
##Bank = 34, Pin name = IO_L22P_T3_34, Sch name = LED6
```

```
set_property PACKAGE_PIN U7 [get_ports {q[3]}]
```

```
set_property IOSTANDARD LVCMOS33 [get_ports {q[3]}]
```

(B) 初值为 2 的模 8 计数器，要求：通过例化 A【调用 (A) 实现的计数器】来实现。

(a) 源程序

```
`include "macro.vh"
```

```
module mod1b (  
    input clk,  
    input mode,  
    output wire [3:0] q,  
    output wire qcc  
);  
    wire [3:0] _q;  
    wire [3:0] __q;  
  
    mod1_1(0, clk, 1, mode, 1, qcc, _q);  
    assign __q[2:0] = _q[2:0];  
    assign __q[3] = 0;  
    assign q = __q + 2;  
endmodule
```

(b) 仿真程序

```
`timescale 1ns / 1ps
```

```
module mod1b_tb (  
  
);  
    reg clk = 0;  
    reg mode = 1;  
    wire qcc;  
    wire [3:0] q;  
    mod1b_(clk, mode, q, qcc);  
    initial forever #1 clk = ~clk;  
    initial #40 mode = 0;  
endmodule
```

(c) 引脚约束 (绑定) 程序

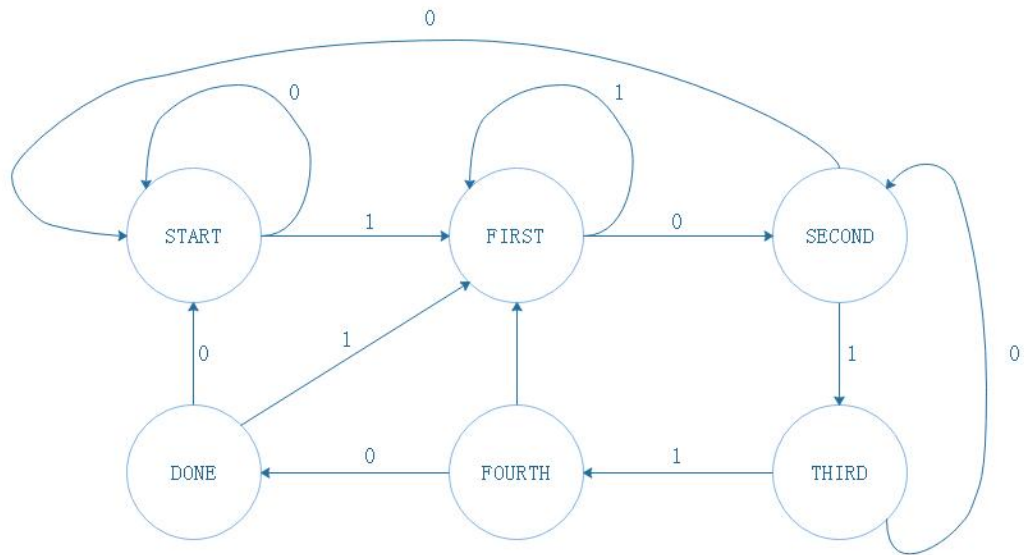
```
## Clock signal  
##Bank = 35, Pin name = IO_L12P_T1_MRCC_35, Sch  
name = CLK100MHZ  
set_property PACKAGE_PIN E3 [get_ports clk]  
set_property IOSTANDARD LVCMOS33 [get_ports clk]  
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}  
[get_ports clk]
```

《数字电路与逻辑设计》实验报告

```
## Switches
##Bank = 34, Pin name = IO_L21P_T3_DQS_34,          Sch
name = SW0
set_property PACKAGE_PIN U9 [get_ports {mode}]
set_property IOSTANDARD LVCMOS33 [get_ports {mode}]
## LEDs
##Bank = 34, Pin name = IO_L24N_T3_34,              Sch name
= LED0
set_property PACKAGE_PIN T8 [get_ports {q[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[0]}]
##Bank = 34, Pin name = IO_L21N_T3_DQS_34,          Sch
name = LED1
set_property PACKAGE_PIN V9 [get_ports {q[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[1]}]
##Bank = 34, Pin name = IO_L24P_T3_34,              Sch name
= LED2
set_property PACKAGE_PIN R8 [get_ports {q[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[2]}]
##Bank = 34, Pin name = IO_L23N_T3_34,              Sch name
= LED3
set_property PACKAGE_PIN T6 [get_ports {q[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {q[3]}]
###Bank = 34, Pin name = IO_L12P_T1_MRCC_34,        Sch
name = LED4
set_property PACKAGE_PIN T5 [get_ports {qcc}]
set_property IOSTANDARD LVCMOS33 [get_ports {qcc}]
```

(2) 采用有限状态机(**FSM**)实现序列检测器的设计方案

(A) 串行 5 位不可重叠的“10110”二进制序列检测器的状态图



(B) 源程序

```
`define zero 1'b0
```

```
`define one 1'b1
```

```
`include "macro.vh"
```

```
module mod2 (
```

```
    input fclk,
```

```
    input val,
```

```
    output wire [7:0] status_view,
```

```
    output reg flag
```

```
);
```

```
    reg [4:0] seq = 5'b01101; // reversed 10110
```

```
    integer curr_status = 0;
```

```
    assign status_view = curr_status;
```

```
    always @(posedge fclk) begin
```

```
        if(val == seq[curr_status]) begin
```

```
            if(curr_status == 4) begin
```

```
                flag <= 1;
```

```
                curr_status <= 0;
```

```
            end else begin
```

```
                curr_status <= curr_status + 1;
```

```
                flag <= 0;
```

```
            end
```

```
end else begin
    flag <= 0;
    case(curr_status)
        0:
            curr_status <= 0;
        1:
            curr_status <= 1;
        2:
            curr_status <= 0;
        3:
            curr_status <= 2;
        4:
            curr_status <= 1;
    endcase
end
end
```

```
endmodule
```

(C) 仿真程序

```
`timescale 1ns / 1ps
```

```
module mod2_tb (
```

```
);
```

```
    reg clk=0, val=0;
```

```
    wire flag;
```

```
    mod2_1(clk, val, flag);
```

```
    initial forever #1 clk = ~clk;
```

```
    initial begin
```

```
        #2 val = 1;
```

```
        #2 val = 0;
```

```
        #2 val = 1;
```

```
        #2 val = 0;
```

```
        #2 val = 1;
```

```
        #2 val = 1;
```

```
#2 val = 0;
#2 val = 1;
#2 val = 1;
#2 val = 0;
end
endmodule
```

(D) 引脚约束 (绑定) 程序

```
## Switches
#Bank = 34, Pin name = IO_L21P_T3_DQS_34,          Sch
name = SW0
set_property PACKAGE_PIN U9 [get_ports {fclk}]
set_property IOSTANDARD LVCMOS33 [get_ports {fclk}]
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets fclk_IBUF]
#Bank = 34, Pin name = IO_25_34,                    Sch name =
SW1
set_property PACKAGE_PIN U8 [get_ports {val}]
set_property IOSTANDARD LVCMOS33 [get_ports {val}]
## LEDs
#Bank = 34, Pin name = IO_L24N_T3_34,                Sch name
= LED0
set_property PACKAGE_PIN T8 [get_ports {status_view[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {status_view[0]}]
##Bank = 34, Pin name = IO_L21N_T3_DQS_34,          Sch
name = LED1
set_property PACKAGE_PIN V9 [get_ports {status_view[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {status_view[1]}]
##Bank = 34, Pin name = IO_L24P_T3_34,                Sch name
= LED2
set_property PACKAGE_PIN R8 [get_ports {status_view[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {status_view[2]}]
#Bank = 34, Pin name = IO_L23N_T3_34,                Sch name
= LED3
set_property PACKAGE_PIN T6 [get_ports {status_view[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {status_view[3]}]
##Bank = 34, Pin name = IO_L12P_T1_MRCC_34,          Sch
name = LED4
```


《数字电路与逻辑设计》实验报告

```
set_property PACKAGE_PIN T5 [get_ports {status_view[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {status_view[4]}]
##Bank = 34, Pin name = IO_L12N_T1_MRCC_34, Sch
name = LED5
set_property PACKAGE_PIN T4 [get_ports {status_view[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {status_view[5]}]
##Bank = 34, Pin name = IO_L22P_T3_34, Sch name
= LED6
set_property PACKAGE_PIN U7 [get_ports {status_view[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {status_view[6]}]
##Bank = 34, Pin name = IO_L22N_T3_34, Sch name
= LED7
set_property PACKAGE_PIN U6 [get_ports {status_view[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {status_view[7]}]
##Bank = 34, Pin name = IO_L10N_T1_34, Sch name
= LED8
set_property PACKAGE_PIN V4 [get_ports {flag}]
set_property IOSTANDARD LVCMOS33 [get_ports {flag}]
```

(3) 3 位二进制数值比较器的设计方案

(A) 一位二进制数值比较器

(a) 源程序

```
module mod3s (
    input left,
    input right,
    output wire left_greater,
    output wire equal,
    output wire left_smaller
);
    assign left_greater = (left > right) ? 1 : 0;
    assign equal = (left == right) ? 1 : 0;
    assign left_smaller = (left < right) ? 1 : 0;
endmodule
```

(b) 仿真程序

```
`timescale 1ns / 1ps
```

```
module mod3_tb (
```

```
);  
    reg l, r;  
    wire greater, even, smaller;  
    mod3s_1(l,r,greater,even,smaller);  
    initial begin  
        l = 7;  
        r = 4;  
        #2 l = 1;  
        #4 r = 1;  
    end  
endmodule
```

(c) 引脚约束 (绑定) 程序

```
## Switches  
#Bank = 34, Pin name = IO_L21P_T3_DQS_34,          Sch name =  
SW0  
set_property PACKAGE_PIN U9 [get_ports {right[0]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {right[0]}]  
#Bank = 34, Pin name = IO_25_34,                    Sch name = SW1  
set_property PACKAGE_PIN U8 [get_ports {right[1]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {right[1]}]  
#Bank = 34, Pin name = IO_L23P_T3_34,                Sch name =  
SW2  
set_property PACKAGE_PIN R7 [get_ports {right[2]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {right[2]}]  
#Bank = 34, Pin name = IO_L19P_T3_34,                Sch name =  
SW3  
set_property PACKAGE_PIN R6 [get_ports {left[0]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {left[0]}]  
#Bank = 34, Pin name = IO_L19N_T3_VREF_34,           Sch name =  
SW4  
set_property PACKAGE_PIN R5 [get_ports {left[1]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {left[1]}]  
#Bank = 34, Pin name = IO_L20P_T3_34,                Sch name =  
SW5  
set_property PACKAGE_PIN V7 [get_ports {left[2]}]
```

《数字电路与逻辑设计》实验报告

```
set_property IOSTANDARD LVCMOS33 [get_ports {left[2]}]
## LEDs
#Bank = 34, Pin name = IO_L24N_T3_34,          Sch name =
LED0
set_property PACKAGE_PIN T8 [get_ports {left_smaller}]
set_property IOSTANDARD LVCMOS33 [get_ports {left_smaller}]
##Bank = 34, Pin name = IO_L21N_T3_DQS_34,      Sch name
= LED1
set_property PACKAGE_PIN V9 [get_ports {equal}]
set_property IOSTANDARD LVCMOS33 [get_ports {equal}]
##Bank = 34, Pin name = IO_L24P_T3_34,          Sch name =
LED2
set_property PACKAGE_PIN R8 [get_ports {left_greater}]
set_property IOSTANDARD LVCMOS33 [get_ports {left_greater}]
```

(B) 3 位二进制数值比较器，要求：通过例化 A【调用 (A) 实现的一位二进制数值比较器】来实现。

(a) 源程序

```
module mod3p (
    input [2:0] left,
    input [2:0] right,
    output wire left_greater,
    output wire equal,
    output wire left_smaller
);
    wire _1, _2, _3;
    reg [2:0] _4, _5;
    mod3s_0(_4, _5, _1, _2, _3);
    assign left_greater = (left > right) ? 1 : 0;
    assign equal = (left == right) ? 1 : 0;
    assign left_smaller = (left < right) ? 1 : 0;
endmodule
```

(b) 仿真程序

```
`timescale 1ns / 1ps
```

```
module mod3_tb (
```

```
);
```

《数字电路与逻辑设计》实验报告

```
reg [2:0] l, r;
wire greater, even, smaller;
mod3p_1(l,r,greater,even,smaller);
initial begin
    l = 7;
    r = 4;
    #2 l = 1;
    #4 r = 1;
end
endmodule
```

(c) 引脚约束 (绑定) 程序

Switches

#Bank = 34, Pin name = IO_L21P_T3_DQS_34, Sch name = SW0

set_property PACKAGE_PIN U9 [get_ports {right[0]}]

set_property IOSTANDARD LVCMOS33 [get_ports {right[0]}]

#Bank = 34, Pin name = IO_25_34, Sch name = SW1

set_property PACKAGE_PIN U8 [get_ports {right[1]}]

set_property IOSTANDARD LVCMOS33 [get_ports {right[1]}]

#Bank = 34, Pin name = IO_L23P_T3_34, Sch name = SW2

set_property PACKAGE_PIN R7 [get_ports {right[2]}]

set_property IOSTANDARD LVCMOS33 [get_ports {right[2]}]

#Bank = 34, Pin name = IO_L19P_T3_34, Sch name = SW3

set_property PACKAGE_PIN R6 [get_ports {left[0]}]

set_property IOSTANDARD LVCMOS33 [get_ports {left[0]}]

#Bank = 34, Pin name = IO_L19N_T3_VREF_34, Sch name = SW4

set_property PACKAGE_PIN R5 [get_ports {left[1]}]

set_property IOSTANDARD LVCMOS33 [get_ports {left[1]}]

#Bank = 34, Pin name = IO_L20P_T3_34, Sch name = SW5

set_property PACKAGE_PIN V7 [get_ports {left[2]}]

set_property IOSTANDARD LVCMOS33 [get_ports {left[2]}]

LEDs

#Bank = 34, Pin name = IO_L24N_T3_34, Sch name =

《数字电路与逻辑设计》实验报告

LED0

```
set_property PACKAGE_PIN T8 [get_ports {left_smaller}]
set_property IOSTANDARD LVCMOS33 [get_ports {left_smaller}]
##Bank = 34, Pin name = IO_L21N_T3_DQS_34,          Sch name
= LED1
```

```
set_property PACKAGE_PIN V9 [get_ports {equal}]
set_property IOSTANDARD LVCMOS33 [get_ports {equal}]
##Bank = 34, Pin name = IO_L24P_T3_34,          Sch name =
LED2
```

```
set_property PACKAGE_PIN R8 [get_ports {left_greater}]
set_property IOSTANDARD LVCMOS33 [get_ports {left_greater}]
```

6、实验结果记录

(1) 4 位二进制加法/减法计数器的实验结果记录

(A) 给出 **Verilog** 设计的模 16 加 1/减 1 计数器的电路图 (RTL Analysis 下“Schematic”截图)

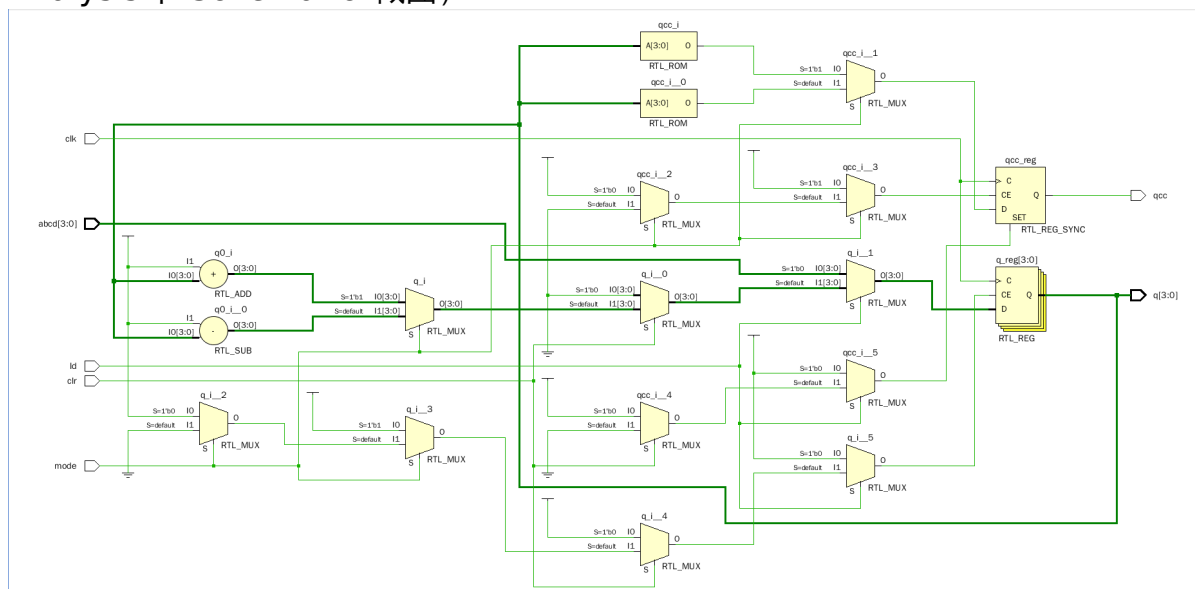


图 2-2 4 位二进制加法/减法计数器

(B) 初值为 2 的模 8 计数器仿真结果截图 (波形含 CP, M, Qa, Qb, Qc, Qd, Q_{模8}等)

仿真对象 cter8, 输入参数 clk, 输出参数 q[3:0], clk 周期 1, q 周期 16.

《数字电路与逻辑设计》实验报告

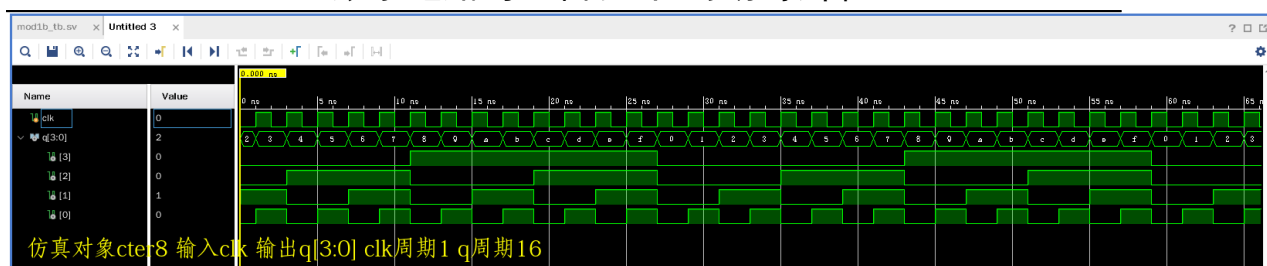


图 2-3 初值为 2 的模 8 计数器仿真

(C) 开发板上的验证情况 (主要记录：验证过程和结论)

验证过程：将 divider 加到模块的 clk 输入和子模块的 long_clk 输入之间，divider 将频率延长为 1Hz，输入绑定到不同的 switch，输出绑定到不同的 led。

结论：观察之后即可发现，结果与仿真相符，与预期相符，为正确的。

(2) 采用有限状态机(FSM) 实现序列检测器的实验结果记录

(A) 给出 Verilog 设计的时序逻辑电路图 (RTL Analysis 下“Schematic”截图)

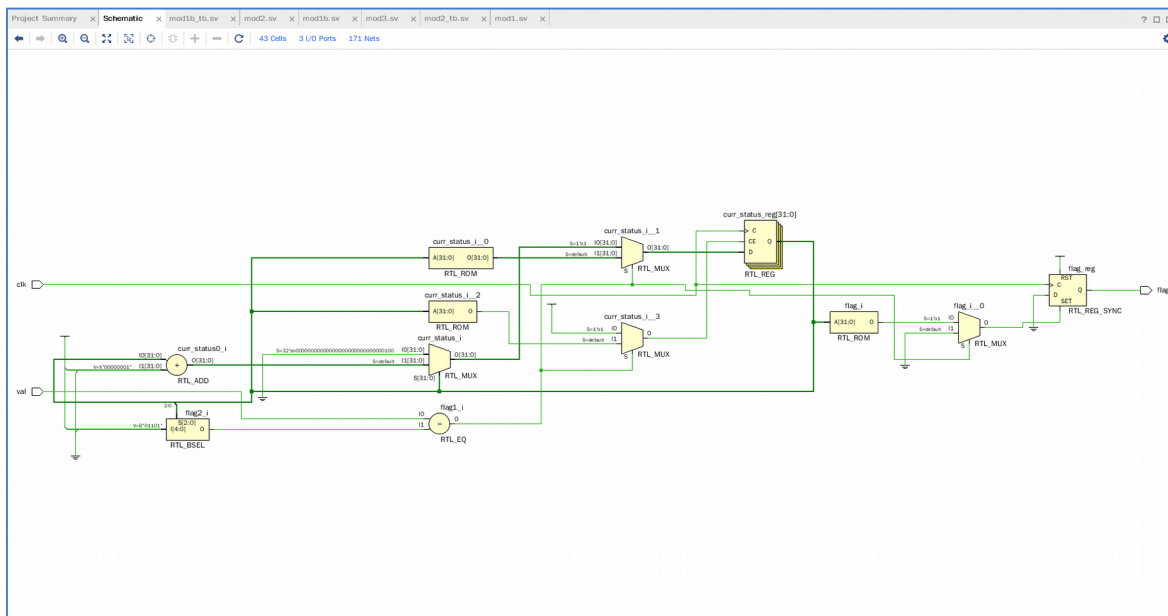


图 2-4 10110 序列检测器

(B) 仿真结果截图 (波形含 clk，输入值，输出值等)

(要求对仿真进行如下说明：仿真对象、输入参数、输出参数、如果具有周期性的要在仿真图上将其“周期”加以标注和说明)

仿真对象 mod2, 输入参数 clk 和 val，输出参数 flag，clk 周期 1.

《数字电路与逻辑设计》实验报告

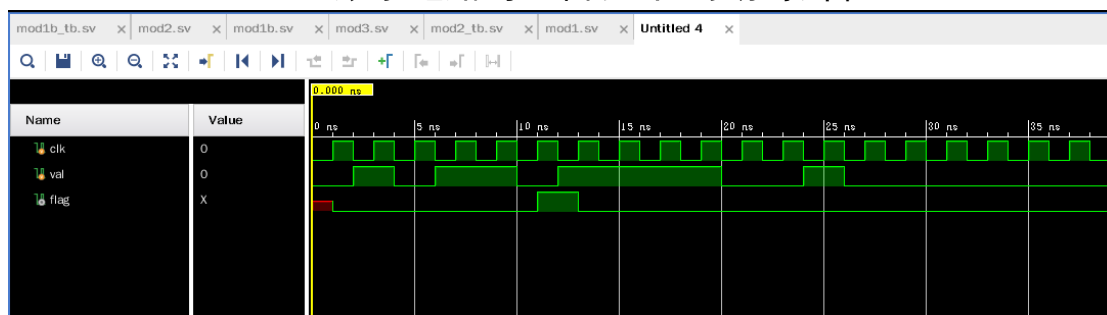


图 2-5 10110 序列检测器仿真

(C) 开发板上的验证情况 (主要记录：验证过程和结论)

验证过程：将 divider 加到模块的 clk 输入和子模块的 long_clk 输入之间，divider 将频率延长为 1Hz，输入绑定到不同的 switch，输出绑定到不同的 led。

结论：观察之后即可发现，结果与仿真相符，与预期相符，为正确的。

(3) 3 位二进制数值比较器的实验结果记录

(A) 给出 Verilog 设计的 3 位二进制数值比较器的电路图 (RTL Analysis 下“Schematic”截图)

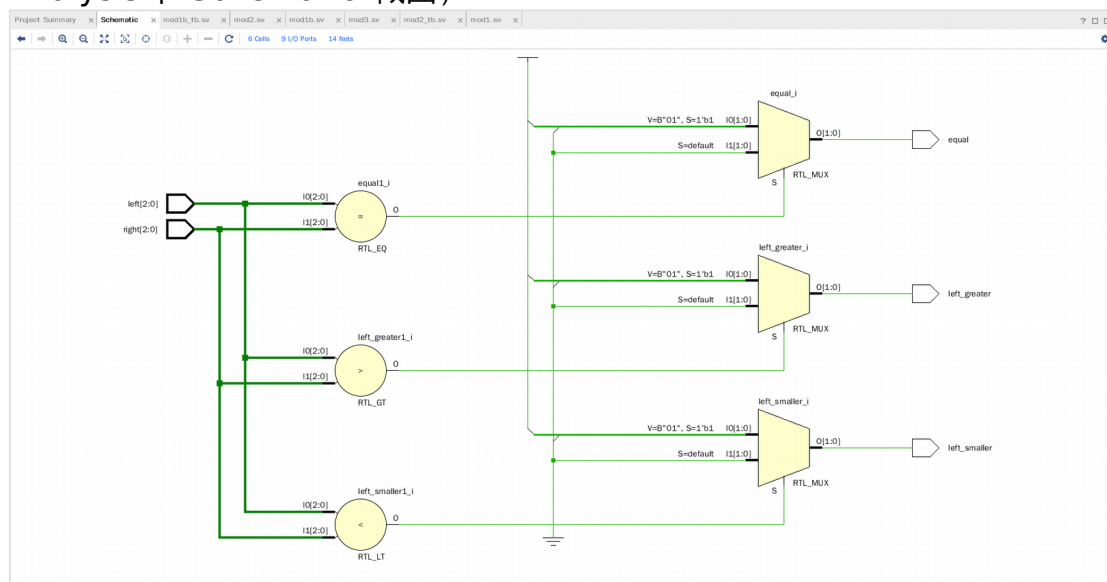


图 2-6 3 位二进制数值比较器

(B) 仿真结果截图 (含输入值，输出值等)

(要求对仿真进行如下说明：仿真对象、输入参数、输出参数、如果具有周期性的要在仿真图上将其“周期”加以标注和说明)

《数字电路与逻辑设计》实验报告

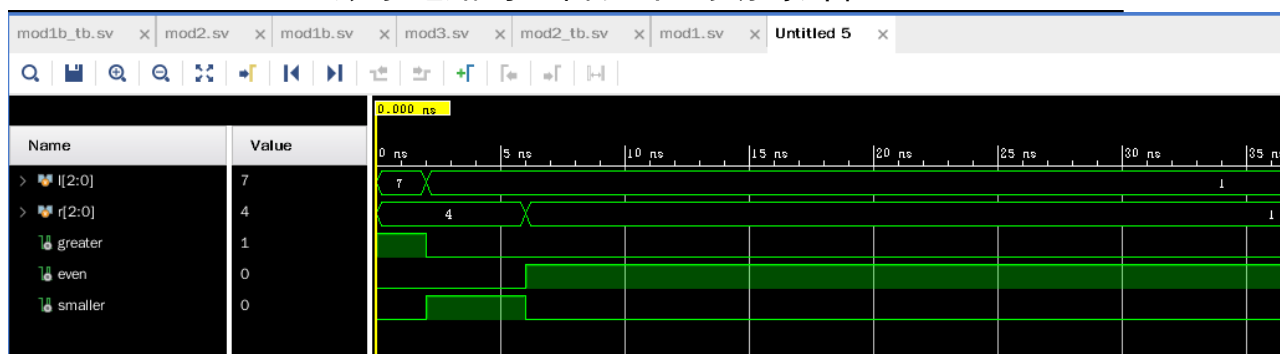


图 2-7 3 位二进制数值比较器仿真

(C) 开发板上的验证情况（主要记录：验证过程和结论）

验证过程：将 divider 加到模块的 clk 输入和子模块的 long_clk 输入之间，divider 将频率延长为 1Hz，输入绑定到不同的 switch，输出绑定到不同的 led。

结论：观察之后即可发现，结果与仿真相符，与预期相符，为正确的。

7、实验后的思考

1. 请通过一个具体的实例来说明你是如何用仿真来验证你电路设计的正确性。

比如，根据图 2-7，可以得到在任何时刻对应的 **a** 和 **b** 的值，由此可以得到理论上 **greater, even, smaller** 应该得到的值，再分别与仿真图中对应时刻的相应变量的真实值进行对比与比较，若理论值与变量的实际值不同，则设计不正确，若完全一致，则说明电路设计正确。

2. 意见和建议

无。