# 实验9—计数器定时器设计与应用实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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： | | 2010-12-3 | 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. 掌握二进制计数器/定时器的工作原理与设计方法
2. 掌握用计数器进行分频的概念和方法

# 二、实验内容和原理

计数器是复杂数字系统和计算机硬件系统中的一个基本部件，是计数、分频、定时、同步和时基等电路的核心，在计算机、网络、通信等设备中经常使用到。

本实验以计数器为例，采用行为级描述的方法设计时序电路，实现各种常用的计数器和定时器。

## 2.1 六十进制的实现

* 60进制计数器用10进制与6进制计数器的组合来实现



图表 1 60进制计数器原理图

## 2.2 二十四进制的实现

* 24进制计数器用4进制与6进制计数器的组合来实现



图表 1 24进制计数器原理图

## 生成分钟、小时

* 分钟的生成以秒脉冲为时钟信号，将10进制和6进制组合成60进制，并产生小时脉冲信号。
* 小时的生成和分钟的生成类似，不同的是少了一个计数器

# 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

设计一个数字钟，使用60进制和24进制计数器，实现24小时内时间的实时显示。60进制计数器用10进制与6进制计数器的组合来实现；24进制可用类似方法实现。

采用4个计数器分别实现分钟的个位、分钟的十位、小时的个位、小时的十位计数。

数字钟的初值通过初始化语句来实现，用数码管前两位显示小时的十位和个位，后两位显示分钟的十位和个位。

当sw[1:0]=11时，btn[1]按下时，显示时钟。

**4.1分别编写并测试分钟、小时生成、时钟生成模块，并检查正确性。**

1. 分钟生成模块

|  |
| --- |
| `timescale 1ns / 1ps  module m\_gen\_min(clk\_sec, clk\_hour, min\_low, min\_high);  input wire clk\_sec;  output reg clk\_hour;  output reg[3:0] min\_low, min\_high;  reg [15:0] cnt;    initial min\_low<=0;  initial min\_high<=0;    always @(posedge clk\_sec) begin  if (clk\_hour == 1)  clk\_hour = 0;  if (cnt == 59) begin  cnt = 0;  if (min\_low == 9) begin  min\_low = 0; /\* base 10 \*/  if (min\_high == 5) begin  min\_high = 0; /\* base 6 \*/  clk\_hour = 1;  end else  min\_high = min\_high + 1;  end else  min\_low = min\_low + 1;  end else  cnt = cnt + 1;  end    endmodule |

1. 小时生成模块

|  |
| --- |
| `timescale 1ns / 1ps  module m\_gen\_hour(clk\_hour, clk\_day, hour\_low, hour\_high);  input wire clk\_hour;  output reg clk\_day;  output reg[3:0] hour\_low, hour\_high;    initial hour\_low<=0;  initial hour\_high<=0;    always @(posedge clk\_hour)  if (hour\_low == 3 && hour\_high == 2) begin  hour\_low = 0;  hour\_high = 0;  clk\_day = ~clk\_day;  end else if (hour\_low == 9) begin  hour\_low = 0;  hour\_high = hour\_high + 1;  end else  hour\_low = hour\_low + 1;  endmodule |

1. 时钟生成模块

|  |
| --- |
| `timescale 1ns / 1ps  module clock\_24h(clk\_sec,digit);  input wire clk\_sec;  output wire[15:0] digit;  wire clk\_hour,clk\_day;  wire [3:0] hour\_high,hour\_low,min\_high,min\_low;    m\_gen\_min MIN (clk\_sec, clk\_hour, min\_low, min\_high);  m\_gen\_hour HOUR(clk\_hour, clk\_day, hour\_low, hour\_high);  assign digit[15:12] = hour\_high[3:0] ;  assign digit[11:8] = hour\_low[3:0] ;  assign digit[7:4] = min\_high[3:0] ;  assign digit[3:0] = min\_low[3:0] ;  endmodule |

**4.2在上次实验基础上，编写顶层模块，将上述模块组装在一起**

Top模块

|  |
| --- |
| `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,disp\_clock;  wire [3:0] btn\_out;  wire [15:0] result;  wire clk\_1s;  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 m3(clk,clk\_1s);  counter\_16bit\_rev m4((clk\_1s&btn\_out[0]),switch[2],disp\_counter);  clock\_24h m5((clk\_1s & btn\_out[1]), disp\_clock);    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  if(btn\_out[1:0] == 2'b01)  display\_num = disp\_counter;  else if(btn\_out[1:0] == 2'b10)  display\_num = disp\_clock;  end  endcase  end  endmodule |

* 1. **修改UCF引脚定义代码**

|  |
| --- |
| 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" ; |

**4.4生成FPGA代码，下载到实验板并调试**

# 五、实验结果与分析

数字钟计时正常，其他功能模块工作正常。

# 六、讨论、心得

这次实验是在之前实验所作的计算器、计数器上再增加时钟模块。

通过上几次实验的操作，我已基本掌握模块编写、调用，了解Verilog的变量定义。所以这次实验进行得特别顺利。

实验中，我想对时钟赋当前时间值，但从老师那了解到实验版上没有当前时间，只有时钟信号。这样，我只好手动对时钟赋初始值。