

基于 PCF8563 的数字钟 FPGA 设计与实现

付永乐, 王崇

(中国电子科技集团公司 第二十七研究所, 郑州 450047)

摘要: 针对基于时钟芯片的数字钟系统进行了研究, 采用 FPGA 作为主控芯片通过 I²C 总线实现对实时时钟芯片 PCF8563 的读写控制, 实现了时钟/日历切换显示, 对时钟/日历手动调校; 在调校状态下, 实现了数据移位时数码管闪烁指示功能; 给出了系统设计以及 FPGA 各模块详细设计, 通过 SignalTap 逻辑分析仪, 验证了系统功能实现的技术可行性, 并在开发板上完成调试, 对 FPGA 开发人员有一定的借鉴意义。

关键词: 数字钟; PCF8563; FPGA; I²C 总线; Verilog 硬件描述语言

FPGA Design and Implementation of Digital clock Based on PCF8563

Fu Yongle, Wang Chong

(The 27th Research Institute of China Electronic Technology Corporation, Zhengzhou 450047, China)

Abstract: The digital clock system based on clock chip is studied. FPGA was used as the control chip to realize the read-write control of PCF8563 based on I²C bus. At the same time, the switch display of clock/calendar and manual adjustment of clock/calendar were also realized. Under adjustment state, the flickering indication function of the digital tube was realized when the data-shift. The system design program and the detailed design of each module of FPGA were given. The technical feasibility of the system function was verified by the signal tap logic analyzer, and completed debugging on development board. This paper can provide certain reference for FPGA developers.

Keywords: digital clock; PCF8563; FPGA; I²C bus; Verilog HDL

0 引言

现场可编程门阵列 (field-programmable gate array, FPGA) 的飞速发展, 在工业控制、电子计量、便携仪器、数字通信、人工智能等各个领域得到广泛应用。Verilog HDL (Verilog hard design language) 作为目前主流的硬件描述语言得到了 FPGA 工程师的青睐。目前数字钟多应用于仪器仪表、电子计量、实时控制系统中^[1-2]。实时时钟芯片 PCF8563 在数字钟设计中应用也非常广泛, 实质是一个对标准频率计数的计数电路, 通常由晶体振荡电路、分频电路、时间计数电路、译码驱动电路等组成^[3]。本文提出了一种基于实时时钟芯片 PCF8563 的数字钟的 FPGA 设计实现思路, 采用 Altera 公司的 Cyclone IV E 系列 FPGA 芯片 EP4CE10F17C8 作为主控芯片, 通过 Inter-Integrated Circuit (I²C) 总线实现对实时时钟芯片 PCF8563 的读写控制, 通过 4 个按键分别完成时钟/日历切换、实时显示和手动调校切换、数据移位设置和数据累加设置功能, 完成实时和调校模式下的六位数码管显示, 并在调校模式下实现数码管闪烁指示功能。通过 SignalTap 逻辑分析仪, 验证了系统功能实现的技术可行性, 并在正点原子开拓者

EP4CE10 开发板上完成调试。

1 时钟芯片 PCF8563 简介

PCF8563 是飞利浦公司的一款具有极低功耗的工业级多功能 CMOS 实时时钟/日历芯片, 工作电流为 0.25 μ A, 具有时钟输出、报警、定时器以及中断输出功能, 能完成各种复杂的定时服务, 其内部功能如图 1 所示^[4]。

PCF8563 内部包括 16 个 8 位寄存器、内置 32.768 Hz 的振荡器、时钟监控、电源复位、分频器、可编程时钟输出、中断输出、看门狗电路以及 I²C 总线接口。分频器给实时时钟 RTC 提供源时钟, 内存地址 00 H、01 H 用作控制和状态寄存器, 地址 02 H~08 H 用作 TIME 计时器, 地址 09 H~0C H 用于报警寄存器, 地址 0D H 控制 CLKOUT 管脚的输出频率, 地址 0E H 和 0F H 分别用于定时器控制寄存器和定时器寄存器, 其中年月日时分秒寄存器均为 BCD 编码^[5]。

当一个 RTC 寄存器被写入或读取时, 所有时间计数器的内容将被冻结, 可避免在进位条件下错误地写或读时钟和日历, 因此所有访问必须在 1 s 内完成。开机复位 (POR) 覆盖, POR 持续时间与晶体振荡器的启动时间直接相关, 在上电复位后, 应持续大于 8 ms, 再进行总线操作^[6]。PCF8563 的所有数据和地址均通过 I²C 总线接口串行传递, 最大总线速度为 400 kbits/s, 每次读写数据后, 内嵌的地址寄存器会自动产生增量。

PCF8563 寄存器写时序过程如下: 先发送器件地址十写操作 0, 随后发 8 位寄存器地址, 最后发 8 位寄存器值,

收稿日期: 2020-03-21; 修回日期: 2020-04-03。

作者简介: 付永乐 (1985-), 男, 山西忻州人, 硕士, 工程师, 主要从事软件测试、FPGA 开发与验证方向的研究。

王崇 (1972-), 男, 河南兰考人, 硕士, 高级工程师, 主要从事计量测试技术方向的研究。

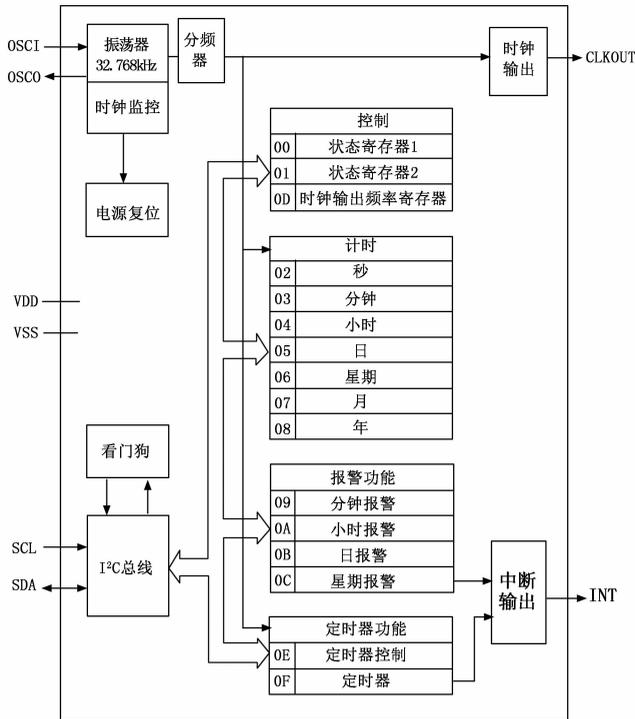


图 1 PCF8563 内部功能框图

S 为起始信号，A 为应答信号，P 为停止信号。读时序过程如下：先发 7 位地址+写操作 0，然后发送寄存器地址，重新发送起始信号 Sr，再次发送 7 位地址+读操作 1，读取寄存器值。

2 数字钟系统设计

数字钟系统由晶振、电源模块、实时时钟芯片 PCF8563、FPGA 电路模块、按键模块以及数码管显示模块 6 部分组成。系统采用 3.3 V 供电，晶振选择 32.768 kHz 和 50 MHz 两种，分别用于实时时钟芯片 PCF8563 工作的外部晶振和 FPGA 工作的基准时钟，FPGA 采用 Cyclone IV E 系列 EP4CE10F17C8 芯片^[7]，时钟芯片 PCF8563 与 FPGA 通过 I²C 总线实现双向通信，系统框图如图 2 所示。

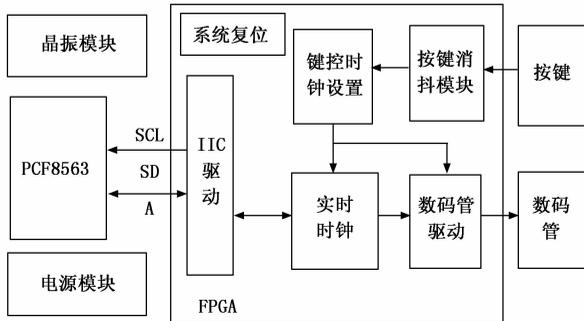


图 2 数字时钟系统框图

数字钟系统具体实现功能如下：1) 实时时钟功能，能够完成常规时钟/日历的实时计数；2) 手动调校功能，能够手动设置年月日时分秒，数据按照相应约束条件进行循

环累加和移位操作，完成时钟/日历的手动调校；3) 数码管动态显示功能，能够在实时和调校模式下完成时钟/日历的数码管切换显示，并在调校模式下实现数码管在当前调整位置的闪烁指示功能。

3 FPGA 设计实现

FPGA 作为本设计的核心部件，按照自顶向下的设计流程，各模块功能通过 Verilog 硬件描述语言实现，主要电路由按键消抖模块、键控时钟设置模块、实时时钟模块、I²C 驱动模块、数码管驱动模块 5 部分构成。

3.1 按键消抖模块设计

设计采用 4 个机械弹性按键，按下为低电平，弹起为高电平。由于机械按键开关在闭合和断开时均不会马上稳定，存在瞬间抖动，为解决不利影响，故采用按键消抖的方法，即当检测到按键状态改变且按键状态稳定后，计数器从 20 ms 开始倒计时，当计数器递减到 1 时，说明按键稳定状态维持了 20 ms，此时按键消抖过程结束，通过寄存器寄存 4 个按键的状态值 key_value [3: 0]，分别表示按键 0、1、2、3 状态，并送入键控时钟设置模块。

3.2 键控时钟设置模块设计

键控时钟设置模块，主要完成边沿检测、模式切换和调校控制功能。

1) 边沿检测功能是一个按键按下时，按键状态值 key_value [3: 0] 中的某一位为低电平，经过下降沿的边沿检测，得到对应的一个周期 CLK 的高电平信号，边沿检测后的信号表示为 neg_sgn [3: 0]，即按键标志信号。

2) 模式切换功能是完成按键标志信号 neg_sgn [3: 0] 产生的相应动作。

当按键 0 按下，按键标志信号 neg_sgn [0] 检测到一个周期的高电平，则通过状态模式切换信号 switch_0 [1: 0] 完成状态循环跳转，其中 0 表示实时时钟模式，1 表示手动调校模式，2 表示调校完成并跳转至实时时钟模式；

当按键 1 按下，按键标志信号 neg_sgn [1] 检测到一个周期的高电平，则通过显示模式切换信号 switch_1 完成状态翻转，其中 0 表示时钟显示，1 表示日历显示；

3) 调校控制功能完成时钟和日历设置。当按键 2 按下，按键标志信号 neg_sgn [2] 检测到一个周期高电平，当前调整数据值移位一次；当按键 3 按下，按键标志信号 neg_sgn [3] 检测到一个周期高电平，当前调整数据值累加一次。时钟设置代码如下，日历设置代码类似，不再赘述。

//检测到移位按键 2 标志信号有效，且在手动调校模式、时钟显示模式

```
else if(neg_sgn[2]=1'b1 &&.switch_0=2'd1 &&.switch_1=1'b1)
```

```
begin
if(displ_drive_t< 3'b101)
displ_drive_t<= displ_drive_t + 1'b1;
else
```

```

disp_drive_t<= 3'd0;
end
//检测到移位按键 3 标志信号有效,当前调整数据值累加
一次。
elseif(neg_sgn[2]=1 &&.swtich_0=2'd 1&&.switch_1=1'
b1)begin
//时间驱动信号
case(disp_drive_t)
3'd0;begin
//时计数
if(hour_set< 4'b0010)
hour_set<= hour_set + 1'b1;
else
hour_set<= 4'd0;
3'd1;//分计数
...
default:...
end
else
//数据保持
begin
disp_drive_t<= disp_drive_t;
hour_set<= hour_set;
...
end

```

最后,将寄存的时钟和日历数据分别拼接为调校时间输出值 num_timeset 和调校日期输出值 num_dataset,通过显示模式切换信号 switch_1 实现时钟和日历的切换输出,具体代码如下: assignnum_set_xs = switch_1 ? num_dataset : num_timeset; 将得到的调校输出值 num_set_xs [23: 0] 送入到数码管显示模块进行调校模式的数据显示,同时将调校时间输出值 num_timeset 和调校日期输出值 num_dataset 进行拼接,得到设置值 num_set [47: 0] 送入实时时钟模块作为实时时钟的初值,即通过 I²C 总线要写入 PCF8563 实时时钟芯片中的数据,并将状态模式切换信号 switch_0、显示模式切换信号 switch_1、时间驱动信号 disp_drive_t、日期驱动信号 disp_drive_d 一并送出至后级模块。

3.3 实时时钟模块设计

实时时钟模块通过调用 I²C 驱动模块来实现对 PCF8563 实时时钟数据的读写,实时时钟模块与 I²C 驱动模块的操作时钟一致,频率选为 250 kHz^[8]。送至 I²C 驱动模块的信号包括 I²C 读写控制信号 I2C_r_w、I²C 触发执行信号 I2C_exe、I²C 器件内地址 I2C_addr、I²C 要写的的数据 I2C_data_w,以及从 I²C 驱动模块得到的信号包括 I²C 一次操作完成标识信号 I2C_done、从 I²C 读的数据 I2C_data_r。通过状态机实现 I²C 数据的读写操作,实时时钟模块状态流程如图 3 所示,主体循环状态为 12 个。写状态时,将设置初值 num_set [47: 0] 数据分段写入 I2C_data_w 寄存器;读状态是将 I²C 读的数据 I2C_data_r,分别存入秒寄

寄存器 sec [7: 0]、分寄存器 min [7: 0]、时寄存器 hour [7: 0]、日寄存器 day [7: 0]、月寄存器 month [7: 0]、年寄存器 year [7: 0],最后将年月日时分秒寄存器值拼接为实时日期显示值 num_data [23: 0] 和实时时间显示值 num_time [23: 0] 的数据,通过显示模式切换信号 switch_1 实现实时显示下的时钟/日历切换,具体代码如下: assignnum = switch_1 ? num_data : num_time,最后将得到的实时显示输出值 num [23: 0] 传递至数码管驱动模块进行显示。

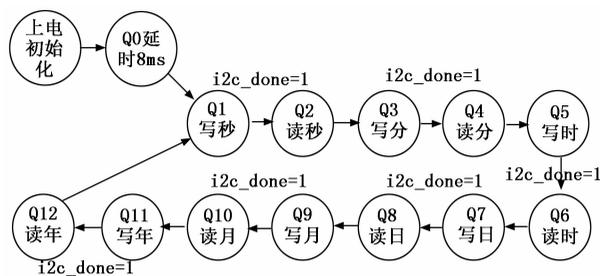


图 3 实时时钟模块状态流程图

状态机流程描述如下:上电初始化后,延时 8 ms,按照顺序执行 Q1~Q12 状态。进入 Q1 写秒状态时,I²C 触发执行信号 I2C_exe 设为 1,器件内地址 I2C_addr 设为 0x02,将设置初值中的低 8 位 num_set [8: 0] 存入 I²C 要写的的数据 I2C_data_w 寄存器;进入 Q2 读秒状态时,当 I²C 一次操作完成标识信号 I2C_done=1,将从 I²C 读的数据 I2C_data_r 存入秒寄存器 sec [7: 0] 寄存器;其他读写过程类似,不同之处是器件内地址、写入和读出的数据值不同。顺序执行到 Q12 状态时,I²C 读写控制信号 I2C_r_w 拉高。

特别指出的是,当进入循环状态后,写入数据完成,I²C 一次操作完成标识信号 I2C_done 拉高后,一直在执行读操作。设计要求完成调校设置后,需要重新写入调校后的值。这时我们在循环体外,加入判断条件。当状态模式切换信号 switch_0 设置为 2 时,即调校设置成功后,模块中状态和数据寄存器清零,否则进入 Q0 状态重新开始执行循环体,实现了调校设置后的数据值成功写入 I²C 总线,完成了时钟/日历初值调整。

3.4 I²C 驱动模块设计

I²C 驱动模块实现底层的 I²C 读写时序控制^[9],读写状态转移如图 4 所示。

当 I²C 触发执行信号 I2C_exe=1 时,从空闲 W0 跳转至 W1 状态,Bit_ctrl=1 表示 16 位字地址,Bit_ctrl=0 表示 8 位字地址,本系统采用 8 位字地址,故跳转至 W3 状态。当 I²C 读写控制标识信号 rw_flag=0 时,表示写数据,跳转至 W6 状态,当状态结束信号 done=1 时,结束 I²C 操作 W7 状态,表示一次 I²C 写操作完成,然后转至空闲;当 I²C 读写控制标识信号 rw_flag=1 时,表示读数据,跳转至 W4 状态,写地址读后,进入 W5 完成读数据,检测到 done=1 时结束 I²C 操作 W7 状态,表示一次 I²C 读

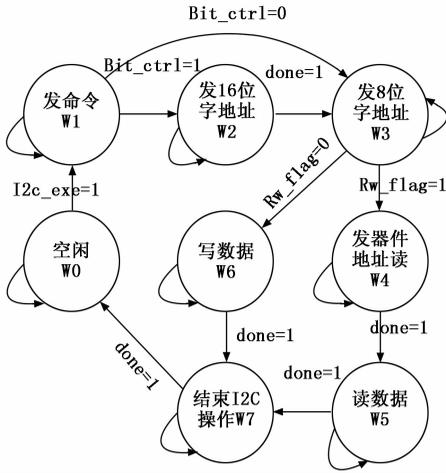


图 4 I²C 读写状态转移图

操作完成，转至空闲。

具体各状态的输出，按照 I²C 总线协议要求设计^[10-11]，SDA 和 SCL 两条信号线同时为高电平时，总线处于空闲状态，即释放总线。当 SCL 为高电平且 SDA 信号由高电平跳变为低电平，即为开始信号 S；当 SCL 为高电平且 SDA 信号由低电平跳变为高电平，即为停止信号 P，开始信号和停止信号均有主控制器（FPGA）产生。SDA 数据在 SCL 为高电平期间必须保持稳定，只有在 SCL 为低电平期间，SDA 数据线上的电平状态才可以改变。输出在 SDA 线上的每个字节均为 8 位，每个字节后需要有一个应答信号 ACK，即 FPGA 在时钟 9 期间释放数据线 SDA，由从器件（PCF8563）反馈应答信号，确保在 SCL 时钟 9 高电平期间为稳定的低电平，则表示接收器（PCF8563）成功接收到该字节。当 FPGA 为接收器时，即从 PCF8563 中读取数据，当 FPGA 接收完最后一个字节后，需要发送一个非应答信号 NACK，通知 PCF8563 停止数据发送，并释放数据线 SDA，以便 FPGA 发送停止信号 P。

3.5 数码管驱动模块

数码管采用动态扫描显示，通过段选和位选信号进行分时点亮控制。为实现在手动调校模式下，进行数据移位设置时，具有当前数码管闪烁指示功能，首先通过显示模式切换信号 switch_1 对时间驱动信号 disp_drive_t 和日期驱动信号 disp_drive_d 进行选择，得到位选闪烁标志信号 disp_drive，在 0~5 范围内变化；通过一个延时计数器得到一个 0.5 s 电平翻转一次的信号 sel_reg，当进入调校模式后，case 语句进行 0~5 状态之间的常规动态扫描，在 0 状态情况下，位选信号 sel [5: 1] <= 5' b11111；如果位选闪烁标志信号 disp_drive = 0，则 sel [0] <= sel_reg，即数码管 0 闪烁显示；否则 sel [0] <= 0，即数码管位选 0 点亮；在 1 状态情况下，位选信号 sel [5: 2] <= 4' b1111；sel [0] <= 1' b1；如果位选闪烁标志信号 disp_drive = 1，则 sel [1] <= sel_reg，即数码管 1 闪烁显示；否则 sel [1] <= 0，即数码管位选 1 点亮；同理实现

其他位选闪烁显示。进入显示模式后，进行常规动态扫描，从而完成数码管数据实时显示和调校模式的闪烁指示。

具体实现代码如下：

```

//得到位选闪烁标志信号 disp_drive
assign disp_drive = switch_1 ? disp_drive_t : disp_drive_d;
//得到 0.5 s 电平翻转一次的信号 sel_reg
if(! rst_n)
begin
sel_reg <= 1'b0;
cnt_ss <= 26'd0;
end
else if(cnt_ss >= 26'd2499999)
begin
sel_reg <= ~sel_reg;
cnt_ss <= cnt_ss + 1'b1;
end
else if(cnt_ss >= 26'd4999999)
begin
sel_reg <= ~sel_reg;
cnt_ss <= 26'd0;
end
else
begin
sel_reg <= sel_reg;
cnt_ss <= cnt_ss + 1'b1;
end
//当进入调校模式，数码管闪烁指示
else if(switch_0 >= 2'd1)
begin
//动态扫描
case(cnt)
//秒低位或日低位显示
3'd0: begin
sel[5:1] <= 5'b11111; //数码管 1~5 灭
num1 <= num[3:0]; //传入秒低位或日低位
point1 <= point[0]; //小数点显示
if(disp_drive == 0)
sel[0] <= sel_reg; //0.5 s 翻转，数码管 0 闪烁
else
sel[0] <= 1'b0; //数码管 0 点亮
end
//秒高位或日高位显示
3'd1: begin
sel[5:2] <= 4'b1111; //数码管 2~5 灭
sel[0] <= 1'b1; //数码管 0 灭
num1 <= num[7:4]; //传入秒高位或日高位
point1 <= point[1]; //小数点显示
if(disp_drive == 1)
sel[1] <= sel_reg; //0.5 s 翻转，数码管 1 闪烁
else
sel[1] <= 1'b0; //数码管 1 点亮
end
endcase
end
end

```

Type	Alias	Name	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50
in		key2[0]																		
in		key2[1]																		
in		key2[2]																		
in		key2[3]																		
C		key_control_u_key_control_num_set																		00000211110h
C		pcf8563_u_pcf8563_num_set																		00000211110h
C		pcf8563_u_pcf8563_num_xs																		211110h
C		pcf8563_u_pcf8563_i2c_data_w																		00h
C		pcf8563_u_pcf8563_i2c_data_r																		21h
R		pcf8563_u_pcf8563_hour																		21h
R		pcf8563_u_pcf8563_min																		11h
R		pcf8563_u_pcf8563_sec																		10h
C		seg_bcd_driv_u_seg_bcd_driv_disp_drive_01																		4h
C		seg_bcd_driv_u_seg_bcd_driv_disp_drive_02																		0h

图 5 SignalTap 逻辑分析结果

```

end
//分低位或月低位显示
3'd2;...
...
default;...
end
//当进入显示模式
else
begin
//动态扫描数码管 0~5
case(cnt)
3'd0;begin
sel<= 6'b111110; //数码管 1~5 灭,0 亮
num1 <= num[3:0]; //传入秒低位或日低位
point1 <= point[0]; //小数点显示
... ; ...
end
3'd5;begin
sel<= 6'b011111; //数码管 0~4 灭,5 亮
num1 <= num[23:20]; //传入时高位或年高位
point1 <= point[0]; //小数点显示
end
default;...
end

```

4 功能验证

通过 SignalTap 逻辑分析仪进行功能验证, 模式切换按键 1 设置为边沿触发, 在开发板上将模式切换为调校模式, 设置时间为 21 : 11 : 10, 移位按键调整至位选信号 sel [3] 控制的数码管上, 点击按键 1 后, 模式切换至实时显示, SignalTap 输出逻辑分析结果如图 5 所示。

实测结果为: 键控时钟设置模块输出 num _ set [47 : 0]=0x000000211110h, 实时时钟模块显示输出 num _ xs [23 : 0]=0x211110h, 时钟显示驱动信号 disp _ drive _ t=0x4h, 结果符合期望要求, 验证了该设计的时钟数据写入和读取以及位选闪烁指示功能的正确性, 日历数据不再赘述。

5 结束语

通过 FPGA 对实时芯片 PCF8563 进行读写控制, 实现了时钟/日历切换显示, 对时钟/日历手动调校。在调校状态下, 实现了数据移位时数码管闪烁指示功能。通过 SignalTap 逻辑分析仪, 验证了系统功能实现的技术可行性, 并在开发板上完成调试, 可供 FPGA 开发和数字钟感兴趣的人员参考, 为后续实现报警、秒表、星期的显示设置提供借鉴。PCF8563 是一款功能完备的实时时钟芯片, 其可靠性高、功耗和成本低, 可用于便携式仪器、电子计量等场合或实际产品中^[12]。

参考文献:

- [1] 雷 琼. 基于 CPLD 的数字钟设计 [J]. 自动化与仪器仪表, 2017 (01): 35 - 37.
- [2] 车 玮. 基于 CPLD 的多功能数字钟的设计 [J]. 仪器仪表用户, 2016 (3): 16 - 17.
- [3] 陈 莹, 郭鹏飞. 基于 FPGA 数字钟的设计 [J]. 科技创新导报, 2017, 14 (05): 74, 76.
- [4] 王 瑜. PCF8563 在电子时钟设计中的应用 [J]. 电子设计工程, 2009 (6): 115 - 116.
- [5] 王君勤, 马孝义, 等. PCF8563 在智能仪表中应用 [J]. 电子测量技术, 2005 (1): 81 - 82.
- [6] 高 健. 外围器件 PCF8563 I²C 实时时钟/日历芯片在 8051 系统中的应用 [J]. 国外电子元器件, 2001 (9): 51 - 54.
- [7] 崔 刚, 陈文楷. 基于 FPGA 的数字钟设计 [J]. 现代电子技术, 2004 (22): 102 - 103.
- [8] 张 颖. 基于 VHDL 语言的数字钟设计 [J]. 科技创新与应用, 2017 (12): 80 - 81.
- [9] 王维斌. 基于 CPLD 与 VHDL 语言的数字钟设计 [J]. 电子技术, 2017, 46 (09): 71 - 72.
- [10] 何立民. I²C 总线应用系统设计 [M]. 北京: 北京航空航天大学出版社, 1995.
- [11] 正点原子. 开拓者 FPGA 开发指南 V1.2 [EB/OL]. <http://www.csdn.net>, 2019.
- [12] 吴小红. 基于 FPGA 的多功能数字钟的设计与实现 [J]. 电子测试, 2018 (22): 19 - 20.