

成绩_____

基于FPGA的等精度频率计设计

摘要:数字频率计是一种能够测量被测信号频率的数字测量仪器。它被广泛用于航天、航空、电子、自动化测量、测控等领域。本文利用等精度测量原理,设计等精度频率计,主要硬件电路由Altera公司生产的复杂可编程逻辑(CPLD) EPM7128构成。复杂可编程逻辑器件CPLD芯片EPM7128SLC84-15完成各种时序逻辑控制、计数功能。在MAX+PLUSII平台上,用VHDL语言编程完成了CPLD的软件设计、编译、调试、仿真和下载。因为本系统采用了先进的EDA技术,不但大大缩短了开发研制周期,而且使本系统具有结构紧凑、体积小,可靠性高,测频范围宽、精度高等优点。

关键词: 等精度频率计; 可编程逻辑器件; VHDL

Abstract: Digital frequency meter is a digital measuring equipment which is capable of measuring the frequency of the measured signal. It has been widely used in aerospace, aviation, electronics, automation, measurement and control, and other fields. This paper introduces a method to design precision frequency meter based on equal precision measuring principle. The main circuit is composed of complex programmable logic (CPLD) EPM7128 which is a production of Altera company and AT89C51. The complex programmable logic device EPM7128SLC84-15 completes sequential logic control, and the counting function. In MAX + PLUSII platform, using VHDL completes the CPLD programming software design, compile, debugging, simulation and download. The system makes use of advanced EDA technology, not only greatly shortens the development cycle, but also makes that the system has so compact, small size, high reliability, wide frequency measurement range and high-precision.

Keywords: Equal precision frequency meters; CPLD; VHDL

前言

随着科学技术的发展,高精度集成电路的使用,生产力得到了大幅度的发展,以大规模集成电路为主的各种设备成了当今社会最常用的设备。频率计在电子项目,资源勘探,仪器仪表等设备有条不紊地工作着,高效率地支配着系统的运行,是项目技术人员必不可少的测量工具。频率计最重要的功能是根据

基准时钟信号实现对被测信号的频率进行检测。由此而延伸的频率测量是电子测量领域里的一项重要内容。因此频率计是项目技术工作者必须具有测量工具，在此基础上的等精度频率计具有相当重要的意义。基于传统测频原理的频率计将随被测信号频率的下降而降低，在实用中有较大的局限性，而等精度频率计不但具有较高的测频精度，不随所测信号的变化而变化，而且在整个测频区域能保持恒定的测频精度。本课题利用EDA技术，基于CPLD芯片EPM7128设计来实现等精度频率计测量，这使设计过程大大简化，缩短了开发周期，减小了电路系统的体积，同时也有利于保证频率计具有较高的精度和较好的可靠性。

EDA(Electronic Design Automation——电子设计自动化)代表了当今电子设计技术的最新发展方向，通过VHDL(Very High Speed Integrated Circuit Hardware Description Language)硬件描述语言的设计，用CPLD(Complex Programmable Logic Device——

复杂可编程逻辑器件)来实现小型电子设备的设计，是开发仪器仪表的主流。据统计，目前发达国家在电子产品开发中EDA工具的利用率已达50%，而大部分的ASIC和CPLD已采用HDL (Hardware Description Language——硬件描述语言)设计。因为VHDL已成为IEEE标准，目前的EDA工具可以使ASIC系统的行为、功能、算法用VHDL描述直接生成CPLD器件，使设计者将精力集中于设计构思，提高了设计效率，同时也利于设计的分解、交流和重用。

一、设计要求

- 1、对于频率测试功能，要求测频范围为1 Hz~10KHz。
- 2、对于测频精度，要求测频全域相对误差恒为百万分之一。

二、常用频率测量的基本方法

(1)

直接测频法：把被测频率信号经脉冲整形电路处理后加到闸门的一个输入端，只有在闸门开通时间T(以秒计)内，被计数的脉冲送到十进制计数器进行计数。

(2)

组合测频法：是指在高频时采用直接测频法，低频时采用直接测量周期法测信号的周期，然后换算成频率。

(3)

倍频法：是指把频率测量范围分成多个频段，使用倍频技术，根据频段设置倍频系数，将经整形的低频信号进行倍频后再进行测量，对高频段则直接进行测

原创力文档
max.book118.com
预览与源文档一致,下载高清无水印

量。倍频法较难实现。

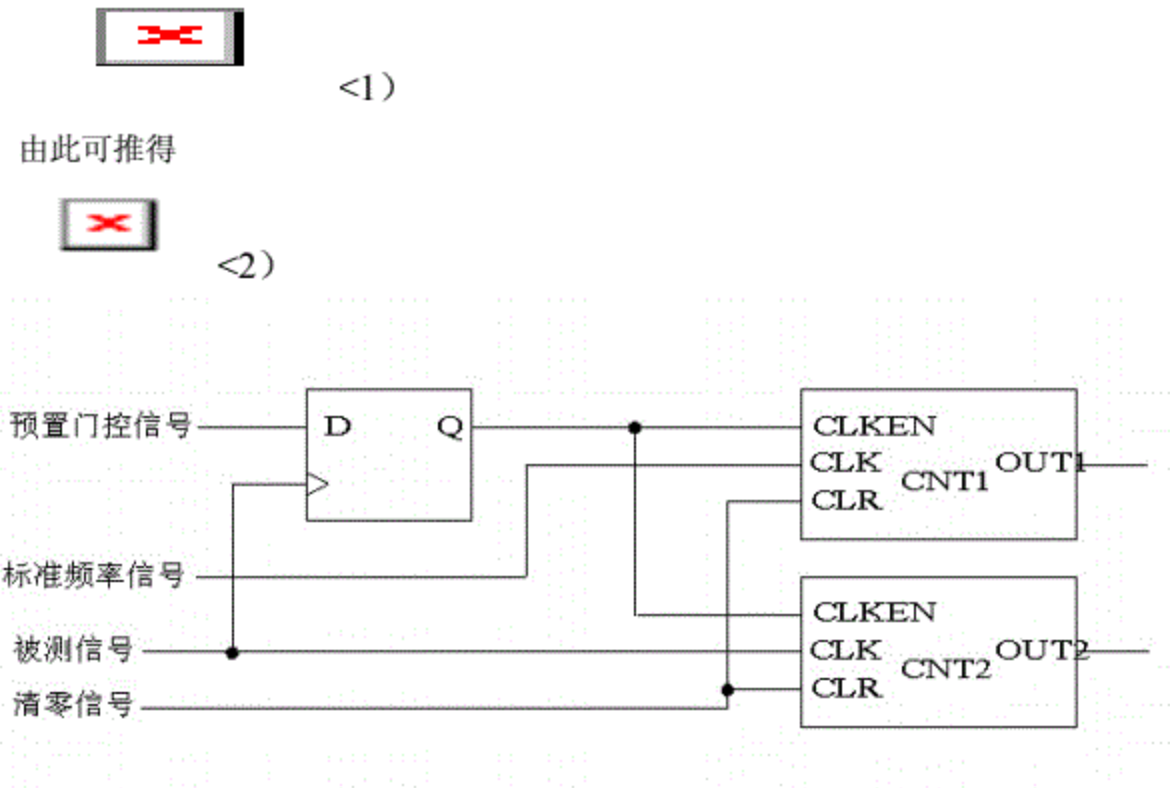
(4>等精度法：等精度测频法。其精确门限由被测信号和预制门控制共同控制，测量精度与被测信号的频率无关，只与基准信号的频率和稳定度有关，因此可以保证在整个测量频段内测量精度不变。

因为等精度法测量频率准确性和稳定性较高所以我们采用等精度法测量频率。

三、实验原理

1.等精度测频法

(4> 等精度测频法：其实现方式可用图1来说明。



图<1>

若所测频率值为 f_x ，其真实值为 f_{xe} ，标准频率为 f_s ，一次测量中，因为 f_x 计数的起停时间都是由该信号的上跳沿触发的，因此在 T_{pr} 时间内对 f_x 的计数 N_x 无误差，在此时间内的计数 N_s 最多相差一个脉冲，即 $\Delta et \leq 1$ ，则下式成立：

 (3)

可分别推得

 (4)



$$\frac{\Delta f_{xe}}{f_{xe}} \leq \frac{1}{N_s}$$

根据相对误差的公式有

$$N_s = T_{pr} f_s$$

经整理可得到



因 $\Delta et \leq 1$ ，故 $\Delta et/N_s \leq 1/N_s$ ，即



2. 周期测量模块

(1>

直接周期测量法：用被测信号经放大整形后形成的方波信号直接控制计数门控电路，使主门开放时间等于信号周期 T_x ，时标为 T_s 的脉冲在主门开放时间进入计数器。设在 T_x 期间计数值为 N ，可以根据以下公式来算得被测信号周期：

$$T_x = NT_s \quad (13.10)$$

经误差分析，可得结论：用该测量法测量时，被测信号的频率越高，测量误差越大。

(2>

等精度周期测量法：该方法在测量电路和测量精度上与等精度频率测量完全相同，只是在进行计算时公式不同，用周期 $1/T$ 代换频率 f 即可，其计算公式为

$$T_x = \frac{T_s \times N_s}{N_x} \quad (13.11)$$

3. 脉宽测量模块

在进行脉冲宽度测量时，首先经信号处理电路进行处理，限制只有信号的50%幅度及其以上部分才能输入数字测量部分。脉冲边沿被处理得非常陡峭，然

后送入测量计数器进行测量。测量电路在检测到脉冲信号的上升沿时打开计数器，在下降沿时关闭计数器，设脉冲宽度为 T_{wx} ，计算公式为

$$T_{wx} = \frac{N_x}{f_s} \quad (13.12)$$

原创力文档
max.book118.com
预览与源文档一致, 下载高清无水印

4. 占空比测量模块

测一次脉冲信号的脉宽，记录其值为 T_{wx1} ，然后将信号反相，再测一次脉宽并记录其值为 T_{wx2} ，通过下式计算占空比：

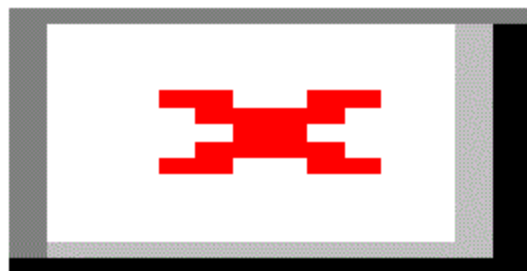
$$\text{占空比} = \frac{T_{wx1}}{T_{wx1} + T_{wx2}} 100\%$$



4、设计分析

1.系统组成

等精度数字频率计涉及到的计算包括加、减、乘、除，耗用的资源比较大，用一般中小规模CPLD/FPGA芯片难以实现。因此，我们选择单片机和CPLD/FPGA的结合来实现。电路系统原理框图如图所示，其中单片机完成整个测量电路的测试控制、数据处理和显示输出；CPLD/FPGA完成各种测试功能；键盘信号由AT89C51单片机进行处理，它从CPLD/FPGA读回计数数据并进行运算，向显示电路输出测量结果；显示器电路采用七段LED动态显示，由8个芯片74LS164分别驱动数码管。



图<2>

2.FPGA测频专用模块的VHDL程序设计

利用VHDL设计的测频模块逻辑结构如图3所示，其中有关的接口信号规定如下：

(1> TF(P2.7>: TF=0时等精度测频；TF=1时测脉宽。

(2>

CLR/TRIG(P2.6>: 当TF=0时系统全清零功能；当TF=1时CLRTRIG的上跳沿将启动CNT2，进行脉宽测试计数。

(3> ENDD(P2.4>: 脉宽计数结束状态信号，ENDD=1计数结束。

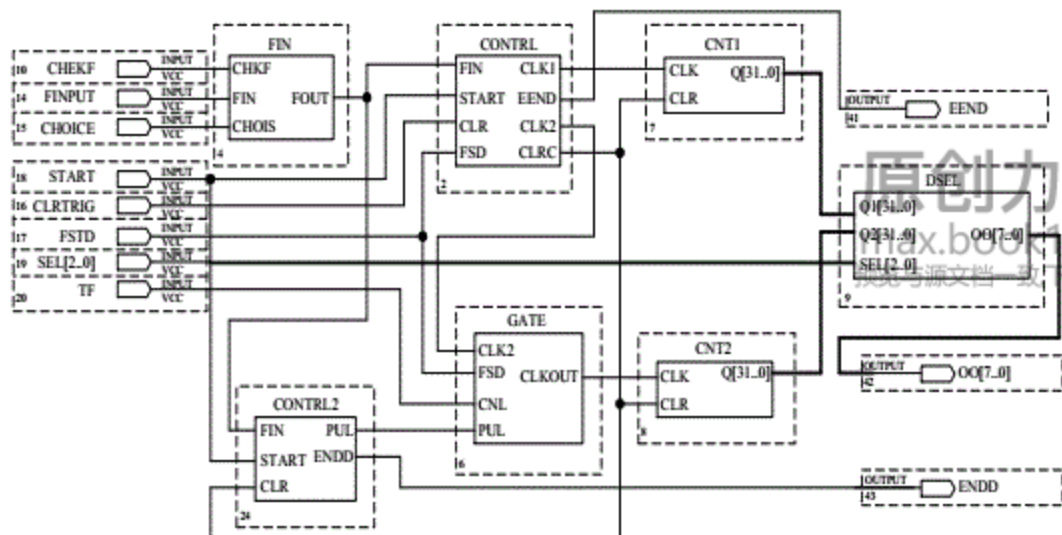
(4> CHOICE(P3.2>: 自校/测频选择，CHOICE=1测频；CHOICE=0自校。

(5>

START(P2.5>: 当TF=0时，作为预置门闸，门宽可通过键盘由单片机控制，START=1时预置门开；当TF=1时，START有第二功能，此时，当START=0时测负脉宽，当START=1时测正脉宽。利用此功能可分别获得脉宽和占空比数据。

(6> EEND(P2.3>: 等精度测频计数结束状态信号，EEND=0时计数结束。

(7> SEL[2..0](P2.2, P2.1, P2.0>: 计数值读出选通控制。



图<3>

3.系统的基本工作方式如下

(1>

P0口是单片机与FPGA的数据传送通信口，P1口用于键盘扫描，实现各测试功能的转换；P2口为双向控制口。P3口为LED的串行显示控制口。系统设置5个功能键：占空比、脉宽、周期、频率和复位。

(2>

7个LED数码管组成测量数据显示器，另一个独立的数码管用于状态显示。

(3> BCLK为测频标准频率50 MHz信号输入端，由晶体振荡源电路提供。

(4> 待测信号经放大整形后输入CPLD/FPGA的TCLK。

CPLD/FPGA测频专用模块的VHDL程序设计

利用VHDL设计的测频模块逻辑结构如图3所示，其中有关的接口信号规定如下：

(1> TF(P2.7>: TF=0时等精度测频；TF=1时测脉宽。

(2>

CLR/TRIG(P2.6>: 当TF=0时系统全清零功能；当TF=1时CLRTRIG的上跳沿将启动CNT2，进行脉宽测试计数。

(3> ENDD(P2.4>: 脉宽计数结束状态信号，ENDD=1计数结束。

(4> CHOICE(P3.2>: 自校/测频选择，CHOICE=1测频；CHOICE=0自校。

(5>

START(P2.5>: 当TF=0时，作为预置门限，门宽可通过键盘由单片机控制，START=1时预置门开；当TF=1时，START有第二功能，此时，当START=0时测

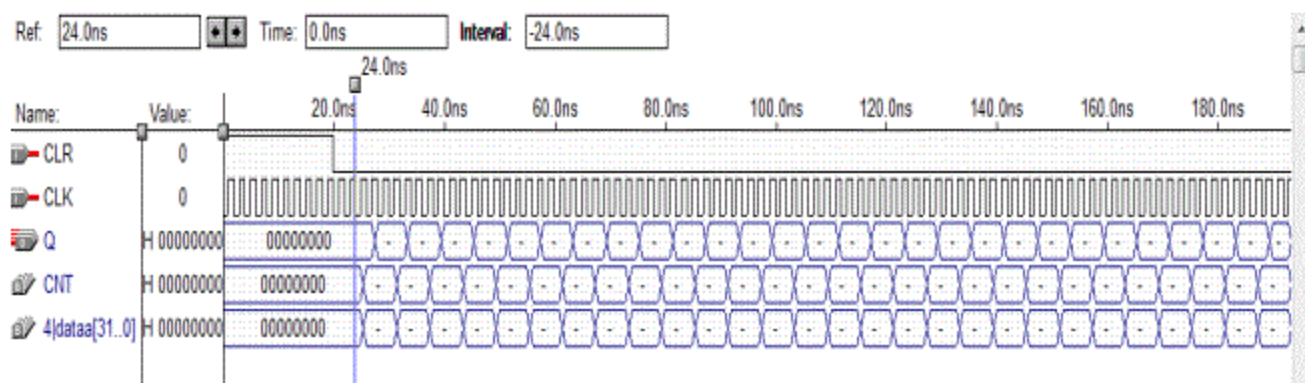
负脉宽，当START=1时测正脉宽。利用此功能可分别获得脉宽和占空比数据。

6> EEND(P2.3>: 等精度测频计数结束状态信号，EEND=0时计数结束。

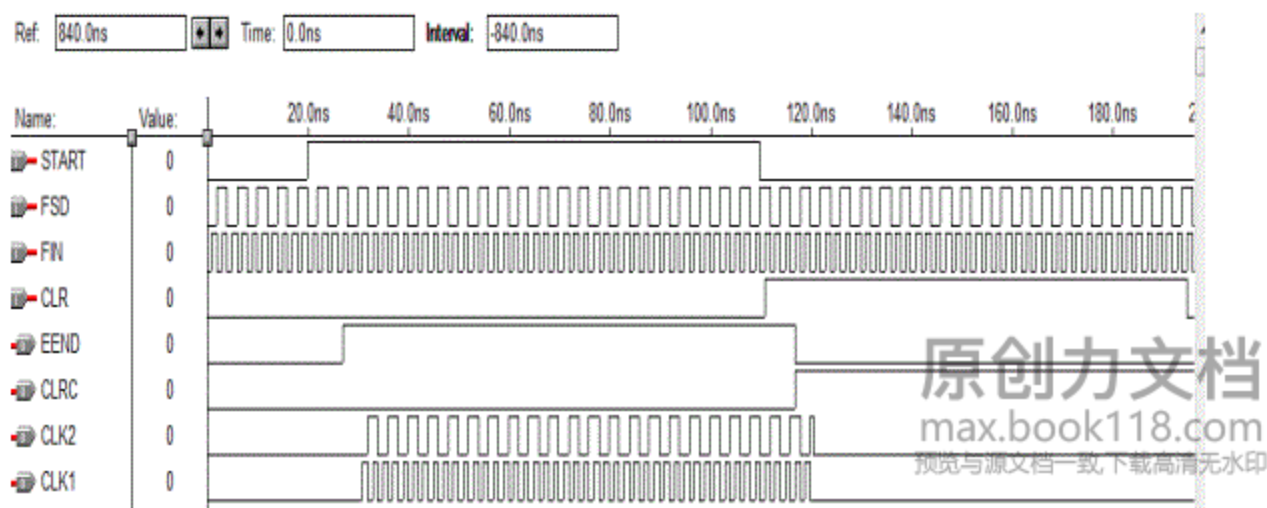
(7> SEL[2..0](P2.2, P2.1, P2.0>: 计数值读出选通控制。

5、系统波形仿真

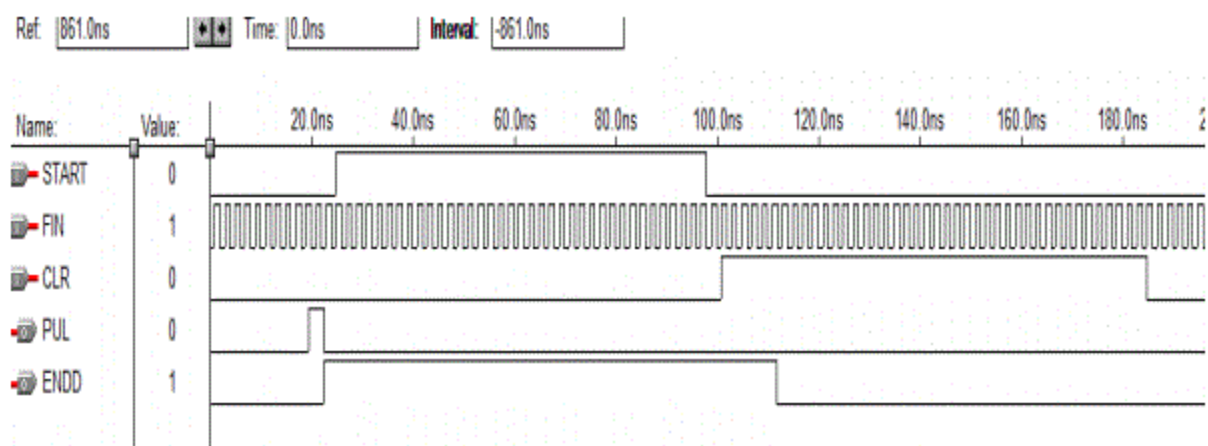
1. 计数模块仿真



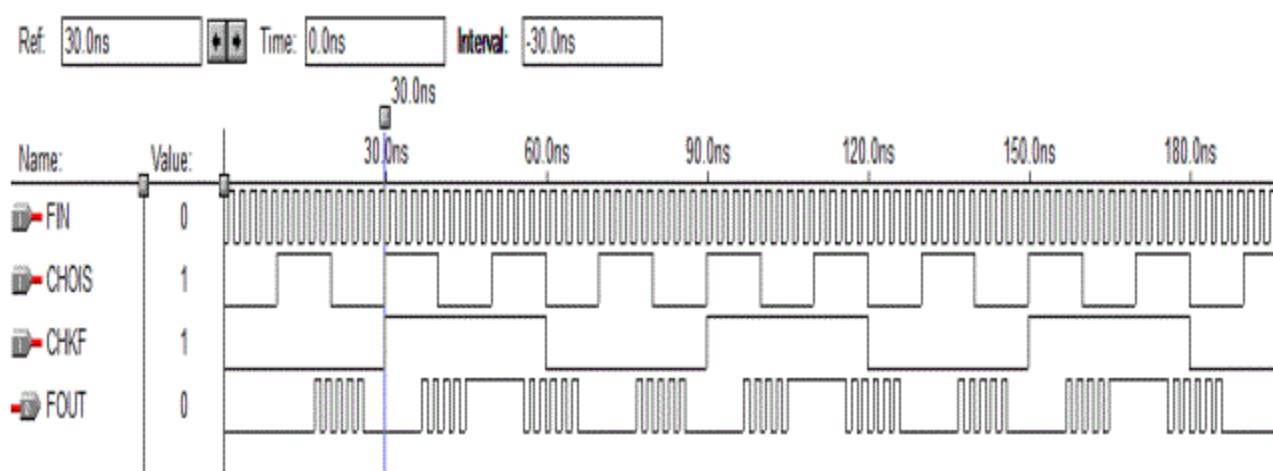
2. 测频、周期控制模块



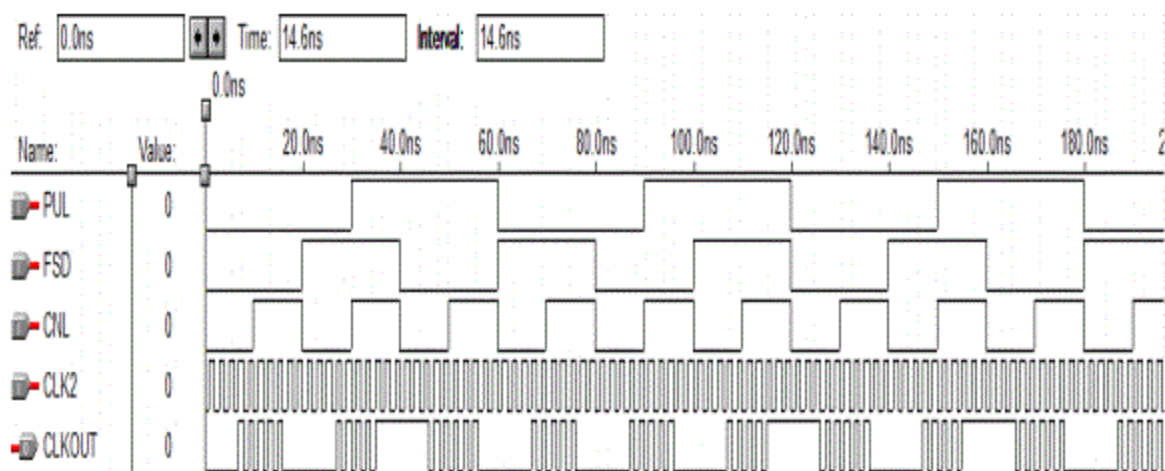
3. 测脉宽、占空比控制模块



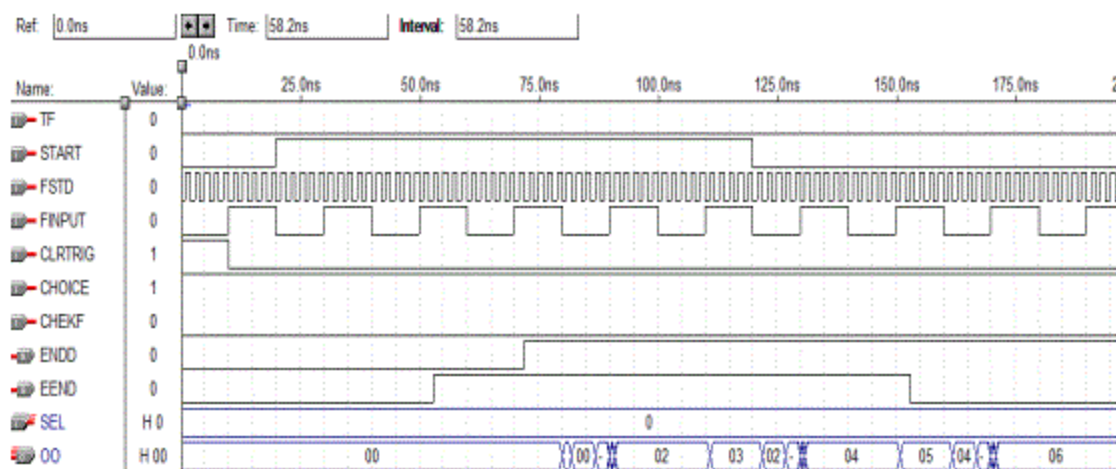
4. 自校/测试频率选择模块



5. 计数器二频率切换模块



6. 系统总体仿真图



六、VHDL主要程序

1.测量计数模块

```

library ieee.
use ieee.std_logic_1164.all.
use ieee.std_logic_unsigned.all.
use ieee.std_logic_arith.all.
entity celiang is
    port (clk:in std_logic.           --基准时钟 (25m>
          tclk:in std_logic.          --被测信号
          reset:in std_logic.         --复位信号
          start:in std_logic.
          --预置闸门控制信号，为 '1' 则可开始计数
          start0:out std_logic.
          s3,s2,s1,s0:out std_logic_vector(3 downto 0)>.      --
    )
    分别对应4个数码管的段选
end celiang.

architecture behav of celiang is
    signal q2:std_logic_vector(24 downto 0>.      --基准信号计数器
    signal en:std_logic.      --预置闸门
    signal en1:std_logic.
    signal s_3,s_2,s_1,s_0:std_logic_vector(3 downto 0>.
begin
    process(tclk,reset>
    此进程计被测信号脉冲数，和得到一个实际闸门信号
    begin
        if reset='1' then

```

```

s_3<="0000"。 s_2<="0000"。 s_1<="0000"。 s_0<="0000"。 --
q1<=(others=>'0')。
    elsif tclk'event and tclk='1' then
        if en='1' and en1='0' then
            s_0<=s_0+1。
                if s_0="1001" then s_0<="0000"。
                    s_1<=s_1+1。
                        if s_1="1001" then s_1<="0000"。
                            s_2<=s_2+1。
                                if s_2="1001" then s_2<="0000"。
                                    s_3<=s_3+1。
                                        if s_3="1001" then
                                            s_3<="1001"。
                                                end if。
                                            end if。
                                        end if。
                                    end if。
                                end if。
                            end if。
                        end if。
                    end if。
                end if。
            end if。
        end if。
    end process。
process(clk,reset)
    此进程完成在实际闸门时间内，计基准脉冲数
begin
    if reset='1' then en1<='0'。 q2<=(others=>'0')。 start0<='0'。
    elsif clk'event and clk='1' then
        if en='1' then
            q2<=q2+1。
                if q2="111" then en1<='1'。 start0<='1'。
                    end if。
                end if。
            end if。
        end if。
    end process。
process(tclk,reset)
begin
    if reset='1' then en<='0'。
    elsif tclk'event and tclk='1' then
        en<=start。
    end if。

```

```

end process。
    s3<=s_3。 s2<=s_2。 s1<=s_1。 s0<=s_0。
end behav。

```

2. 计数模块CNT.VHD

```

LIBRARY IEEE。
USE IEEE.STD_LOGIC_1164.ALL。
USE IEEE.STD_LOGIC_UNSIGNED.ALL。
ENTITY CNT IS
    PORT(CLK, CLR: IN STD_LOGIC。
          Q: OUT STD_LOGIC_VECTOR(31 DOWNTO 0)>)。
END ENTITY CNT。
ARCHITECTURE ART OF CNT IS
    SIGNAL CNT: STD_LOGIC_VECTOR(31 DOWNTO 0)。
    BEGIN
        PROCESS(CLK, CLR) IS
            BEGIN
                IF CLR='1' THEN CNT<="000000000000"。
                ELSIF CLK'EVENT AND CLK='1' THEN CNT<=CNT+1。
                END IF。
            END PROCESS。
            Q<=CNT。
        END ARCHITECTURE ART。

```

3. 测频、周期控制模块CONTRL.VHD

```

LIBRARY IEEE。
USE IEEE.STD_LOGIC_1164.ALL。
ENTITY CONTRL IS
    PORT(FIN, START, CLR, FSD: IN STD_LOGIC。
          CLK1, EEND, CLK2, CLRC: OUT STD_LOGIC>)。
END ENTITY CONTRL。
ARCHITECTURE ART OF CONTRL IS
    SIGNAL QQ1: STD_LOGIC。
    BEGIN
        PROCESS(FIN, CLR, START) IS
            BEGIN
                IF CLR='1' THEN QQ1<='0'。
                ELSIF FIN'EVENT AND FIN='1' THEN QQ1<=START。
                END IF。
            END PROCESS。

```

```

    CLRC<=CLR。      EEND<=QQ1。
    CLK1<=FIN AND QQ1。
    CLK2<=FSD AND QQ1。
END ARCHITECTURE ART。

```

6、测脉宽、占空比控制模块CONTRL2.VHD

```

LIBRARY IEEE。
USE IEEE.STD_LOGIC_1164.ALL。
USE IEEE.STD_LOGIC_UNSIGNED.ALL。
ENTITY CONTRL2 IS
    PORT (FIN, START, CLR: IN STD_LOGIC。
          ENDD, PUL: OUT STD_LOGIC)。
END ENTITY CONTRL2。
ARCHITECTURE ART OF CONTRL2 IS
    SIGNAL QQ: STD_LOGIC_VECTOR(3 DOWNTO 1)。
    SIGNAL A0, B0, C0, F2: STD_LOGIC。
    SIGNAL S: STD_LOGIC_VECTOR(1 DOWNTO 0)。
BEGIN
    S(0)<=QQ(3)。      S(1)<=QQ(2)。
    PROCESS(START, S) IS
        BEGIN
            IF START='1' THEN F2<=FIN。
            ELSE F2<=NOT FIN。
            END IF。
            IF S=2 THEN PUL<='1'。
            ELSE PUL<='0'。
            END IF。
            IF S=3 THEN ENDD<='1'。
            ELSE ENDD<='0'。
        END IF。
    END PROCESS。
    A0<=F2 AND QQ(1)。
    B0<=NOT A0。
    C0<=NOT F2。
    PROCESS(C0, CLR) IS
        BEGIN
            IF CLR='1' THEN QQ(1)<='0'。
            ELSIF C0'EVENT AND C0='1' THEN QQ(1)<='1'。

```

```

        END IF。
    END PROCESS。
PROCESS (A0, CLR) IS
    BEGIN
        IF CLR='1' THEN QQ(2)<='0'。
        ELSIF A0'EVENT AND A0='1' THEN QQ(2)<='1'。
        END IF。
    END PROCESS。
PROCESS (B0, CLR) IS
    BEGIN
        IF CLR='1' THEN QQ(3)<='0'。
        ELSIF B0'EVENT AND B0='1' THEN QQ(3)<='1'。
        END IF。
    END PROCESS。
END ARCHITECTURE ART。

```

七、小结

通过这次设计是我和我的同伴们熟练的掌握了max plusII软件的基本使用方法，对VHDL语言的编程方法和编程技巧有了更深层次的理解，对FPGA的了解及使用有了更深层次的认识，也了解了用FPGA开发产品的基本过程。在完成此次设计的过程中，遇到了很多困难，但最终都一一攻克，这全是因为组员间的团结互助。让我们懂得了组员之间协作的重要性。

参考文献

【1】EDA技术实用教程—VHDL版<第四版> 作者：潘松、黄继业

【2】EDA技术与使用<第二版> 清华大学出版社 作者：潘松、黄继业

【3】基于FPGA的嵌入式开发与使用 电子工业出版社 作者：徐光辉、程东旭

、黄如

【4】EDA技术与实践 清华大学出版社 作者：赵明富

【5】Altera FPGA/CPLD设计<基础篇> 人民邮电出版社 作者：吴继华、

王诚

【6】Max Plus ii handbook