实验 17 多功能等精度频率计

参考 VHDL 实用教程 298 页

基于传统测频原理的频率计的测量精度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量精度，而且在整个频率区域保持恒定的测试精度。本项设计的基本指标为：

(1) 频率测试功能，测频范围 0.1Hz～70MHz，测频精度：测频全域相对误差恒为百万分之一。

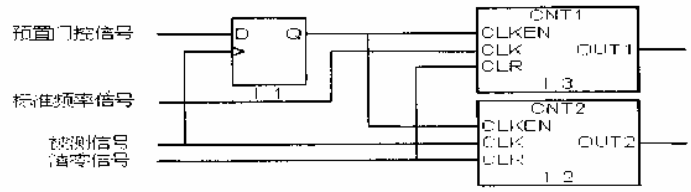
(2) 周期测试功能、信号测试范围与精度要求与测频功能相同。

(3) 脉宽测试功能，测试范围 0.1μs～1s，测试精度 0.01μs 。

(4) 占空比测试功能，测试精度 1%～99%。

一、测频原理

等精度测频的实现方式可以简化为图 14-1 来说明。图中预置门控信号是宽度为 Tpr 的一个脉冲，CNT1 和 CNT2 是两个可控计数器，标准频率信号从 CNT1 的时钟输入端 CLK 输入，其频率为 Fs；经整形后的被测信号从 CNT2 的时钟输入端 CLK 输入，设其实际频率为 Fxe，测量频率为 Fx。



当预置门控信号为高电平时，经整形后的被测信号的上沿通过 D 触发器的 Q 端同时启动计数器 CNT1 和 CNT2。CNT1、CNT2 分别对被测信号（频率为 Fx）和标准频率信号（频率为 Fs）同时计数。当预置门信号为低电平时，随后而至的被测信号的上沿将使这两个计数器同时关闭。设在一次预置门时间 Tpr 中对被测信号计数值为 Nx，对标准频率信号的计数值为 Ns，则下式成立：

$$Fx / Nx = Fs / Ns \qquad (14\text{-}1)$$

由此可推得：

$$Fx = (Fs / Ns) \cdot Nx \qquad (14\text{-}2)$$

其误差分析如下：

若设所测频率值为 Fx，其真实值为 Fxc，标准频率为 Fs。在一次测量中，由于 Fx 计数的起停时间都是由该信号的上跳沿触发的，在 Tpr 时间内对 Fx 的计数 Nx 无误差；在此时间内 Fs 的计数 Ns 最多相差一个脉冲，即 $|\Delta e t| \leq 1$，则下式成立：

$$Fx/Nx = Fs / Ns \tag{14-3}$$

$$Fxe / Nx = Fs / (Ns+\Delta et) \tag{14-4}$$

由此可分别推得：

$$Fx = (Fs / Ns) \cdot Nx \tag{14-5}$$

$$Fxe = [Fs / (Ns+\Delta et)] \cdot Nx \tag{14-6}$$

根据相对误差公式有：

$$\frac{\Delta Fxe}{Fxe} = \frac{|Fxe-Fx|}{Fxe} \tag{14-7}$$

将式(14-5)、(14-6)代入式（14-7）并整理得：

$$\frac{\Delta Fxe}{Fxe} = \frac{|\Delta et|}{Ns} \tag{14-8}$$

$$\because |\Delta et| \leqslant 1 \qquad \therefore \frac{|\Delta et|}{Ns} \leqslant \frac{1}{Ns} \tag{14-9}$$

即

$$|\delta| = \frac{\Delta Fxe}{Fxe} \leqslant \frac{1}{Ns} \tag{14-10}$$

$$Ns = Tpr \cdot Fs \tag{14-11}$$

由上式可以得出以下结论：

(1) 相对测量误差与频率无关；

(2) 增大 Tpr 或提高 Fs，可以增大 Ns，减少测量误差，提高测量精度。

(3) 标准频率误差为 $\Delta Fs/Fs$，由于晶体的稳定度很高，标准频率误差可以进行校准。

(4) 等精度测频方法测量精度与预置门宽度和标准频率有关，与被测信号的频率无关。在预置门时间和常规测频闸门时间相同而被测信号频率不同的情况下，等精度测量法的测量精度不变，而常规的直接测频法精度随着被测信号频率的下降而下降。测试电路可采用高频率稳定度和高精度的恒温可微调的晶体振荡器作标准频率发生电路。

14.1.2 测频专用模块工作原理和设计

根据以上给出的等精度测频原理，利用 VHDL 设计的测频模块逻辑结构如图 14-2 所示，各模块功能和工作步骤如下。

1. 测频/测周期实现

被测信号脉冲从 CONTRL 模块的 FIN 端输入，标准频率信号从 CONTRL 的 FSD 端输入，CONTRL 的 CLR 是此模块电路的工作初始化信号输入端。在进行频率或周期测量时，完成如下步骤：

(1) 令 TF=0，选择等精度测频，然后在 CONTRL 的 CLR 端加一正脉冲信号以完成测试电路状态的初始化。

(2) 由预置门控信号将 CONTRL 的 START 端置高电平，预置门开始定时，此时由被测信号的上沿打开计数器 CONT1，进行计数，同时使标准频率信号进入计数器 CONT2。

(3) 预置门定时结束信号把 CONTRL 的 START 端置为低电平（由单片机来完成），在被测信号的下一个脉冲的上沿到来时，CONT1 停止计数，同时关断 CONT2 对 Fs 的计数。

(4) 计数结束后，CONTRL 的 EEND 端将输出低电平来指示测量计数结束，单片机得到此信号后，即可利用 ADRB、ADRA 分别读回 CONT1 和 CONT2 的计数值，并根据等精度测量公式进行运算，计算出被测信号的频率或周期值。

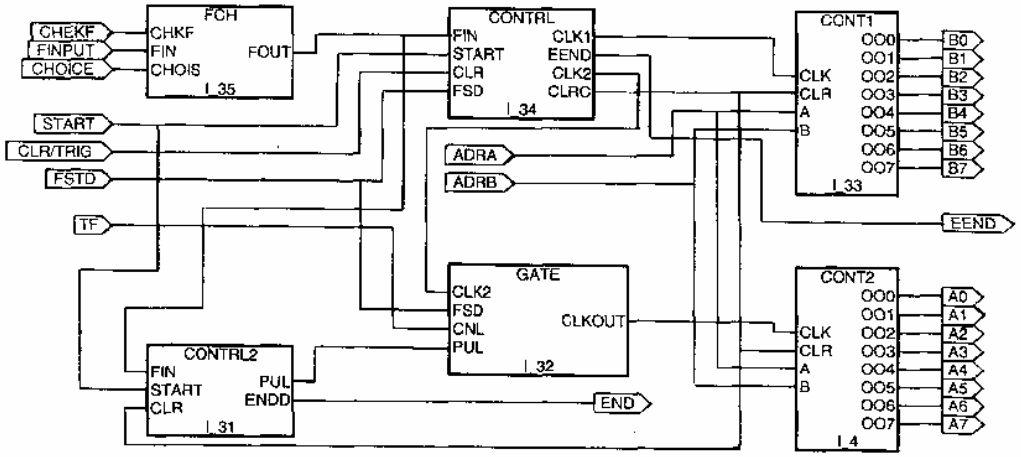


图 14-2 测频模块逻辑图

2. 控制部件设计

如图 14-3 所示，当 D 触发器的输入端 START 为高电平时，若 FIN 端来一个上沿，则 Q 端变为高电平，导通 FIN→CLK1 和 FSD→CLK2，同时 EEND 被置为高电平作为状态标志；在 D 触发器的输入端 START 为低电平时，当 FIN 端输入一个脉冲上沿，FIN→CLK1

与 FSD→CLK2 的信号通道被切断。

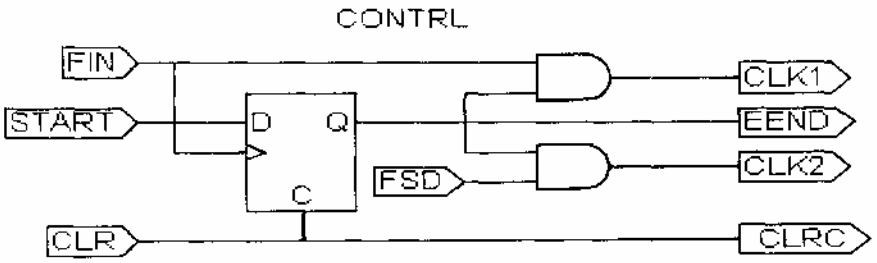


图 14-3　测频与测周期控制部分电路

3．计数部件设计

图 14-2 中的计数器 CONT1/CONT2 是 32 位二进制计数器，输出 8 位数据总线，单片机可分 4 次将 32 位数据全部读出。

4．脉冲宽度测量和占空比测量模块设计

根据上述脉宽测量原理，设计如图 14-4 (CONTRL2)的电路原理示意图。该信号的上沿和下沿信号对应于未经处理时的被测信号的 50% 幅度时上沿和下沿信号。被测信号从 FIN 端输入，CLR 为初始化信号，START 为工作使能信号，图 14-4 中 CONTRL2 的 PUL 端与 GATE 的输入端 PUL 相连。其测量脉冲宽度的工作步骤是：

(1) 向 CONTRL2 的 CLR 端送一个脉冲以便进行电路的工作状态初始化。

(2) 将 GATE 的 CNL 端置高电平，表示开始脉冲宽度测量，这时 CONT2 的输入信号为 FSD。

(3) 在被测脉冲的上沿到来时，CONTRL2 的 PUL 端输出高电平，标准频率信号进入计数器 CONT2。

(4) 在被测脉冲的下沿到来时，CONTRL2 的 PUL 端输出低电平，计数器 CONT2 被关断。

(5) 由单片机读出计数器 CONT2 的结果，并通过上述测量原理公式计算出脉冲宽度。

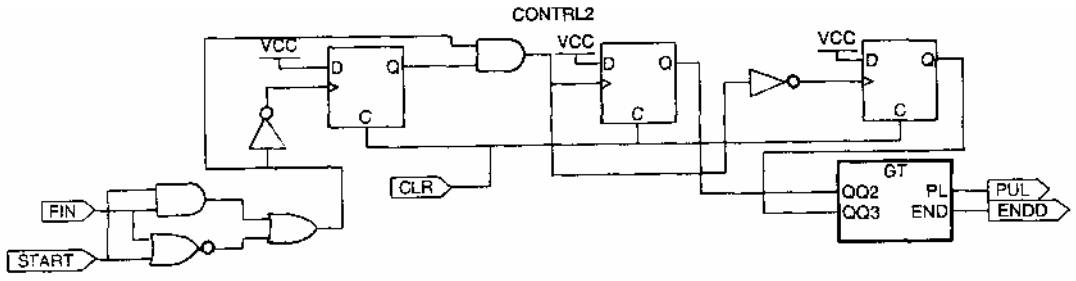


图 14-4　脉冲宽度测量原理图

CONTRL2 子模块的主要特点是：电路的设计保证了只有 CONTRL2 被初始化过后才能工作，否则 PUL 输出始终为零。只有在先检测到上沿后 PUL 才为高电平，然后在检测到下

沿时，PUL 输出为低电平；ENDD 输出高电平以便通知单片机测量计数已经结束；如果先检测到下沿，PUL 并无变化；在检测到上沿并紧接一个下沿后，CONTRL2 不再发生变化直到下一个初始化信号到来。占空比的测量方法是通过测量脉冲宽度记录 CONT2 的计数值 N1，然后将输入信号反相，再测量其脉冲宽度，测得 CONT2 计数值 N2，则可以计算出：

$$\text{占空比} = \frac{N1}{N1+N2} \times 100\%$$

14.1.3 频率计功能模块的 VHDL 描述

基于以上的测试原理与各模块的功能描述，以下给出相应的 VHDL 逻辑描述。

1. CNT1/CNT2

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY CNT IS
    PORT (A, B, CLK, CLR: IN STD_LOGIC;
                    OO: OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
                     Q: OUT STD_LOGIC_VECTOR(31 DOWNTO 0) );
END CNT;
ARCHITECTURE behav OF CNT IS
    SIGNAL CNT : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL SEL : STD_LOGIC_VECTOR(1 DOWNTO 0);
BEGIN
    PROCESS(CLK, CLR)
    BEGIN
      IF CLR = '1' THEN     CNT <= "00000000000000000000000000000000";
        ELSIF CLK'EVENT AND CLK = '1' THEN CNT <= CNT + 1;
      END IF;
    END PROCESS;
    PROCESS(A, B)
    BEGIN
        SEL(0) <= A;        SEL(1) <= B;
          IF SEL = "00" THEN    OO <= CNT(7 DOWNTO 0);
        ELSIF SEL = "01" THEN   OO <= CNT(15 DOWNTO 8);
        ELSIF SEL = "10" THEN   OO <= CNT(23 DOWNTO 16);
        ELSIF SEL = "11" THEN   OO <= CNT(31 DOWNTO 24);
            ELSE                OO <= "00000000";       END IF;
```