

# 2011年全国大学生电子设计竞赛

简易数字信号传输性能分析仪（ E 题）

【本科组】



2011 年 9 月

## 摘要

眼图对于展示数字信号传输系统的性能提供了很多有用的信息：可以从中看出码间串扰的大小和噪声的强弱，有助于直观地了解码间串扰和噪声的影响，评价一个基带系统的性能优劣。眼图的形成是由于示波器的余辉作用，扫描所得的每一个码元波形将重叠在一起，从而形成眼图。本设计中使用可编程器件和线性移位寄存器设计成一个数字信号发生器，用来产生数字信号和伪随机信号。用低通滤波器模拟传输信道，并且把伪随机信号模拟为信道噪声。用 FPGA 以及一定的外围电路作为数字信号处理系统，提取位同步信号。我们把经过模拟信道后的数字信号传输给数字信号分析模块处理，最后通过示波器观察其眼图，以便对信号进行进一步的分析和调理。

关键字：数字信号处理；伪随机信号；低通滤波；眼图

## **Abstract**

Eye diagram to display digital signal transmission system performance to provide much useful information: can also be seen from the crosstalk between the code size and noise intensity, contribute to the intuitive understanding of intersymbol interference and noise effects, evaluation of a baseband system performance. Eye formation is due to the persistence of oscilloscope, the scanned every symbol waveforms will overlap together, thereby forming the eye diagram. The design of using FPGA and linear shift register design into a digital signal generator, used to generate the digital signal and the pseudo random signal. With low pass filter is used as a transmission channel, and the pseudo random signal as channel noise. Using FPGA as well as some of the peripheral circuit as a digital signal processing system. We put after the analog channel after the digital signal is transmitted to the digital signal analysis circuit, finally through the oscilloscope to observe the eye, so that the signal for further analysis and conditioning.

Keywords: digital signal processing; pseudo random signal; low pass filter; eyediagram

# 目录

- 第一章系统方案 . . . . . 1
  - 1.1 信号发生模块的论证与选择 . . . . . 1
  - 1.2 低通滤波器模块的论证与选择 . . . . . 1
  - 1.3 数字信号分析方案论证与选择 . . . . . 2
- 第二章系统理论分析与计算 . . . . . 3
  - 2.1 低通滤波器的分析 . . . . . 3
    - 2.1.1 阶数的确定 . . . . . 3
    - 2.1.2 电阻、电容参数的确定。 . . . . 3
    - 2.1.3 通带增益 . . . . . 4
  - 2.2 m 序列数字信号的产生 . . . . . 4
    - 2.2.1 数字信号的产生 . . . . . 4
    - 2.2.2 伪随机信号的产生 . . . . . 5
  - 2.3 眼图显示方法 . . . . . 6
    - 2.3.1 眼图的概念 . . . . . 6
    - 2.3.2 眼图的意义 . . . . . 6
    - 2.3.3 眼图的显示 . . . . . 7
  - 2.4 同步信号提取 . . . . . 7
- 第三章电路与程序设计 . . . . . 8
  - 3.1 电路的设计 . . . . . 8
    - 3.1.1 系统总体框图 . . . . . 8
    - 3.1.2 电源 . . . . . 8
  - 3.2 程序的设计 . . . . . 8
    - 3.2.1 程序模块 . . . . . 8
    - 3.2.2 程序流程图 . . . . . 8
- 第四章测试方案与测试结果 . . . . . 10
  - 4.1 测试方案 . . . . . 10
    - 4.1.1 硬件测试 . . . . . 10
    - 4.1.2 软件仿真测试 . . . . . 10
    - 4.1.3 硬件软件联调 . . . . . 10
  - 4.2 测试条件与仪器 . . . . . 10
  - 4.3 测试结果及分析 . . . . . 10
    - 4.3.1 测试结果 (数据 ) . . . . . 10
    - 4.3.2 测试分析与结论 . . . . . 14
- 第五章结论 . . . . . 16
- 附录 1：电路原理图 . . . . . 17
- 附录 2：源程序 . . . . . 19

# 简易数字信号传输性能分析仪（ E 题）

## 【本科组】

### 第一章系统方案

本次设计主要由信号发生模块、低通滤波器、控制部分、数字信号分析部分和电源组成。

#### 1.1 信号发生模块的论证与选择

方案一：单片机控制线性移位寄存器产生数字信号和伪随机信号。由于普通单片机所能产生的时钟信号频率不是很高，而本设计中要产生数据率为 100Kbps 的数字信号和数据率为 10Mbps 的伪随机信号，所以必须要用 DDS 信号发生器来产生频率如此高的时钟信号，以控制线性移位寄存器的工作。一个 DDS 同一时间只能产生单一频率的信号，但是题目中数字信号的数据率与伪随机信号的数据率相差甚远，所以要控制这两个信号的发生，要使用两组 DDS，这不利于节约成本。

方案二：FPGA 控制线性移位寄存器产生数字信号和伪随机信号。要产生任意频率的时钟信号，用 FPGA 是相当容易实现的，因此原则上可以用 FPGA 同时控制两组移位寄存器同时产生数字信号和伪随机信号。由于考虑到题目要求数字信号发生器和数字信号分析电路各制作一块电路板，而且我们在设计中要使用 FPGA 来分析数字信号，这个矛盾是难以解决的。

方案三：分别用单片机控制移位寄存器产生数字信号，用 FPGA 控制移位寄存器产生伪随机信号。本方案很有效的解决了上面两种方案遇到的问题，这样能满足题目要求，又可节约成本。

综合以上三种方案，基于性价比和可行性的设计指标，我们选择方案三。

#### 1.2 低通滤波器模块的论证与选择

方案一：切比雪夫低通滤波。这种低通滤波电路采用切比雪夫传递函数，其带外衰减比较快，与巴特沃思滤波器相比所使用的阶数低，在设计中可以节约成本。但是这种滤波器在带内幅频特性很不稳定，通带内部分会带有纹波，因而会对原信号产生干扰，这对于小信号的滤波是相当不利的。

方案二：巴特沃思滤波器。这种滤波器采用的是巴特沃斯传递函数。巴特沃思滤波器的特点是通频带内的频率响应曲线最大限度平坦，没有起伏，而在阻频带则逐渐下降为零。在振幅的对数对角频率的波特图上，从某一边界角频率开始，振幅随着角频率的增加而逐步减少，趋向负无穷大。一阶巴特沃斯滤波器的衰减率为每倍频 6 分贝，每十倍频 20 分贝。二阶巴特沃斯滤波器的衰减率为每倍频 12 分贝、三阶巴特沃斯滤波器的衰减率为每倍频 18 分贝、如此类推。巴特沃斯滤波器的振幅对角频率单调下降，并且也是唯一的无论阶数，振幅对角频率曲线都保持同样的形状的滤波器。只不过滤波器阶数越高，在阻频带振幅衰减速度越快。

综合以上二种方案，巴特沃思滤波器只需两阶便可满足 40dB/十倍频程的要求，设计容易，幅频特性比较好，稳定性好，所以选择方案二。

### 1.3 数字信号分析方案论证与选择

本设计中数字信号的分析主要是提取数字信号的同步时钟，并将同步时钟进行位同步处理，最终使所提取的这个时钟信号与数字信号同频同相输入到示波器的外触发模式，从而观察眼图。

对于数字信号的分析可以使用单片机最为处理器进行数据分析，也可以使用 FPGA 最为处理器进行数据分析。

考虑到实际设计中提取同步时钟需要相当快的处理速度，这样才能达到比较高的精度，而且位同步处理也要进行复杂而快速的计算，在此单片机在这里显示出了它的不足，相反 FPGA 却能发挥出它同时高速处理大量数据的优势。因而我们这里优先选择 FPGA 进行数据处理。

## 第二章系统理论分析与计算

### 2.1 低通滤波器的分析

题目要求设计三个低通滤波器，截止频率分别为 100KHz、200 KHz、500 KHz，截止频率误差不大于 10%；带外衰减均不少于 40dB/十倍频程；滤波器通带增益  $A_F$  可调且范围为 0.2~4.0。

#### 2.1.1 阶数的确定

我们已经选定用巴特沃思滤波器来进行设计，那么首先得确定低通滤波器的阶数。有巴特沃斯低通滤波器的特性可知：一阶巴特沃斯滤波器的衰减率为每倍频 6 分贝，每十倍频 20 分贝。二阶巴特沃斯滤波器的衰减率为每倍频 12 分贝、每十倍频 40 分贝。三阶巴特沃斯滤波器的衰减率为每倍频 18 分贝、每十倍频 60 分贝，如此类推。根据题目要求，在误差允许范围内我们只需选用二阶低通滤波就能实现衰减 40dB/十倍频程的要求。低通滤波器的电路如图 2-1 所示。

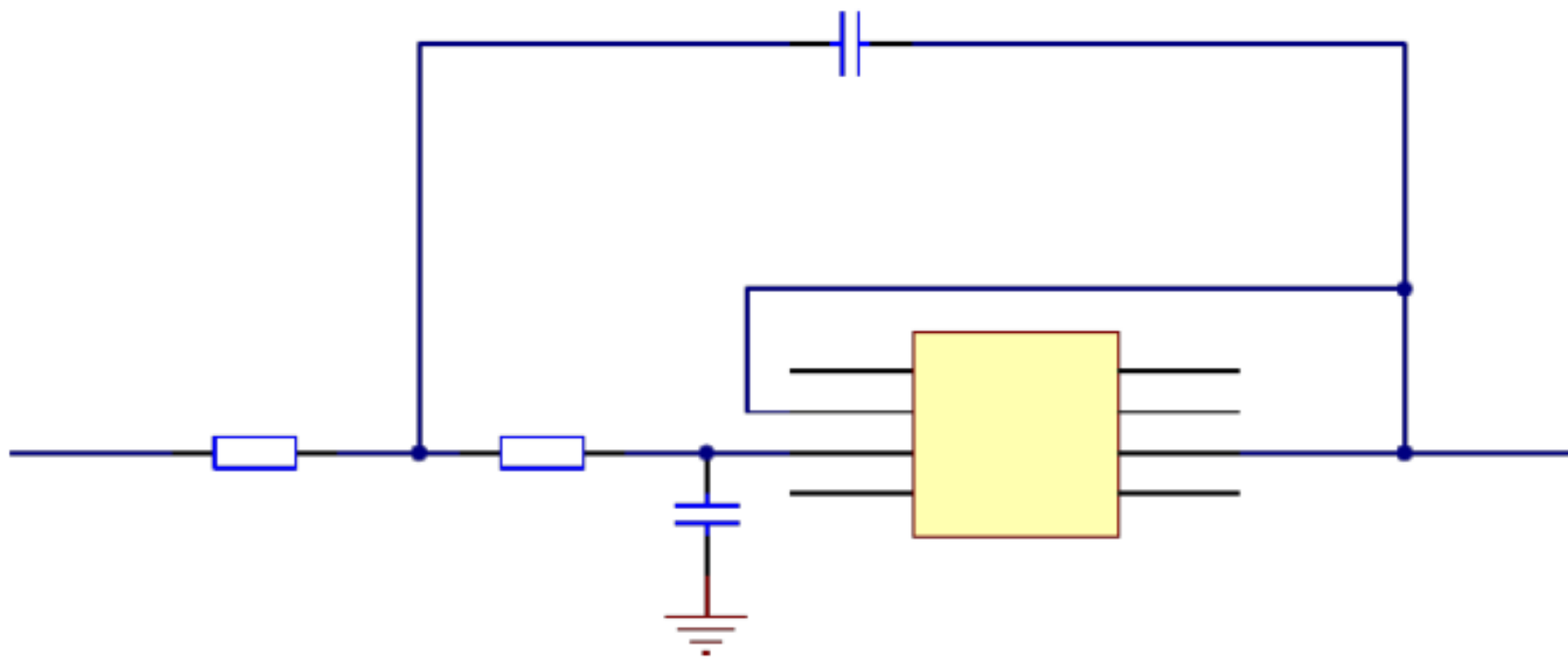


图 2-1 低通滤波电路原理图

#### 2.1.2 电阻、电容参数的确定。

由于市面上常用的电容标称值之间的差距较大，所以我们应优先确定电容的标准值。然后再借助截止频率  $f_c$  与 RC 之间的关系便可大致确定阻值。用 Filter Wiz PRO 软件可以大概确定电容、电阻的值。经过调整和计算最终确定的 R、C 参数分别为：

截止频率为 100KHz 时， $R_1=R_2=2.4K$ ， $C_1=330pF$ ,  $C_2=1.32nF$ 。

截止频率为 200KHz 时， $R_1=R_2=1.2K$ ， $C_1=330pF$ ,  $C_2=1.32nF$ 。

截止频率为 500KHz 时， $R_1=R_2=480\ \Omega$ ， $C_1=330\text{pF}$ , $C_2=1.32\text{nF}$

### 2.1.3 通带增益

由于低通滤波的运放是同相接入的，而同相运放的放大倍数至少为 1 倍，不能实现增益 AF 在 0.2~4.0 可调的要求，所以要把低通滤波器中的运放设计为一个电压跟随器，再在其后串接一个反相比例放大器，这样便能实现 AF 在 0.2~4.0 可调的要求。所加反相比例放大器如图 2-2 所示。其增益为

$$A_{vf} = - \frac{R_f}{R_1} \quad (1)$$

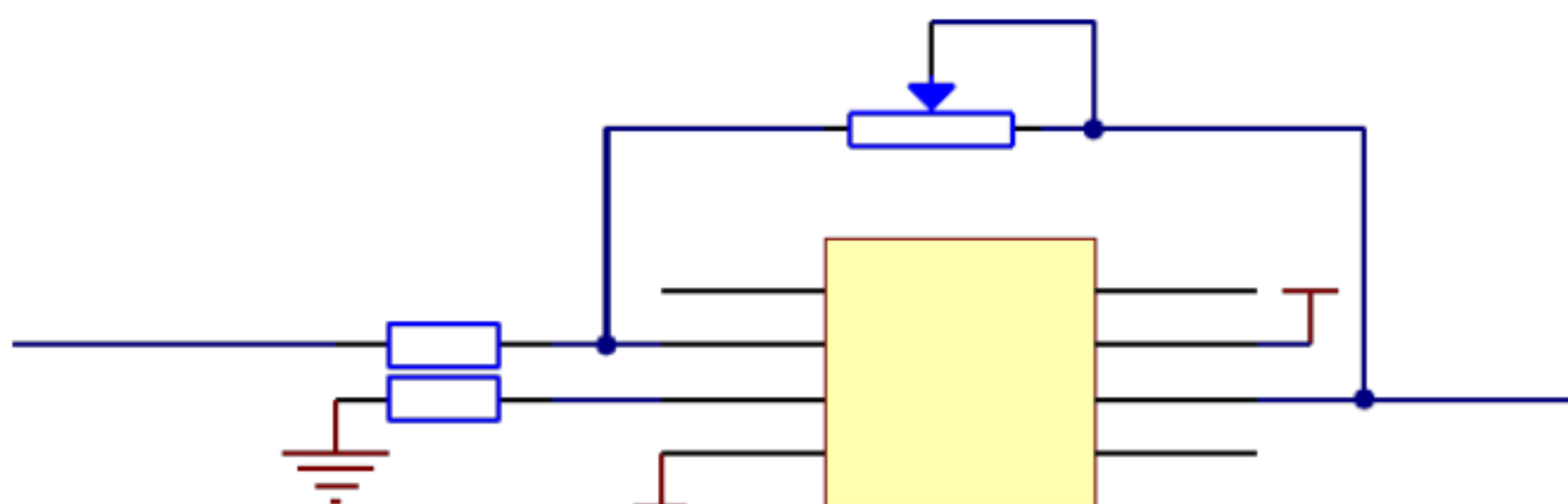


图 2-2 反相比例放大器电路图

## 2.2 m 序列数字信号的产生

设计要求产生  $f_1(x)=1+x^2+x^3+x^4+x^8$  的 m 序列数字信号和  $f_2(x)=1+x+x^4+x^5+x^{12}$  的 m 序列伪随机信号。

### 2.2.1 数字信号的产生

m 序列是一种由线性移位寄存器产生的周期最长的序列。设计时，我们先通过程序设定一个初始状态值，值得注意的是这个初始值是不能全为 0 的，因为当这些值全为 0 时，序列经过移位、相加后的值会一直为 0，即这样的序列恒等于 0，所以也就没意义了。m 序列  $f_1(x)=1+x^2+x^3+x^4+x^8$  的产生原理图如图 2-3 所示。

这里我们对数字信号设定的初始值为 11111111，这部分因为频率不高，所以用单片机进行时序控制，当上电复位后给 8 位线性移位寄存器置入初始值 11111111，然后通过 DDS 产生的时钟信号 V1-clock 控制逐位移位， $x^n$  (n = m) 系数为 1 的相应的位上接入反馈，并且该反馈与对应为进行异或运算，这样我们只需在最高位上便可以取出所要的数字信号。其电路如图 2-4 所示。



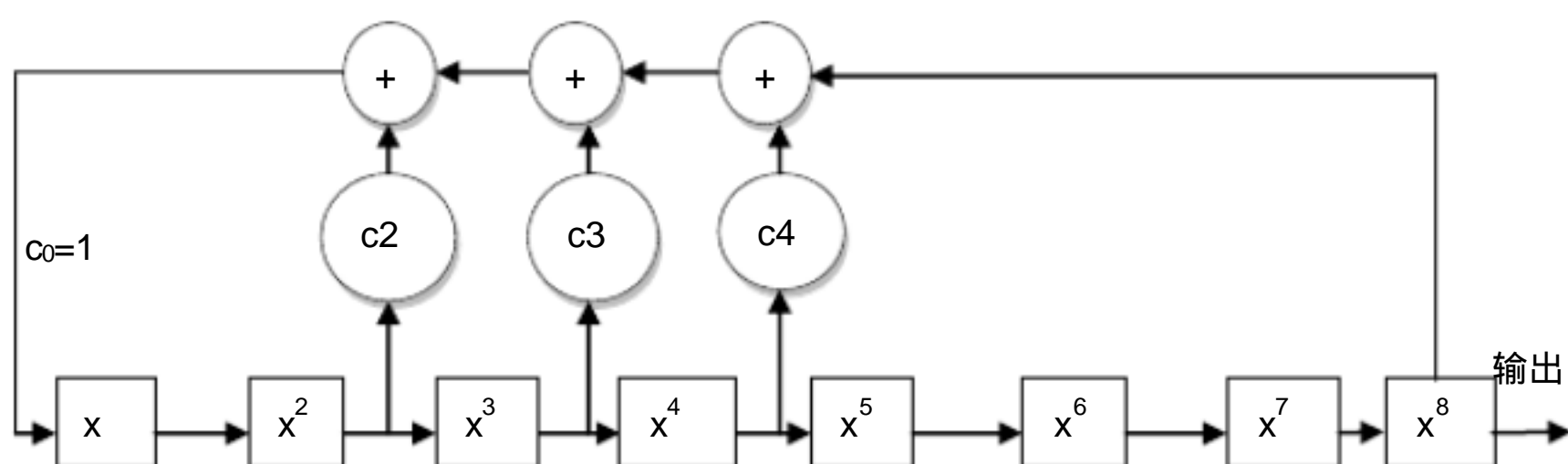


图 2-3 m 序列  $f_1(x)=1+x^2+x^3+x^4+x^8$  的产生原理图

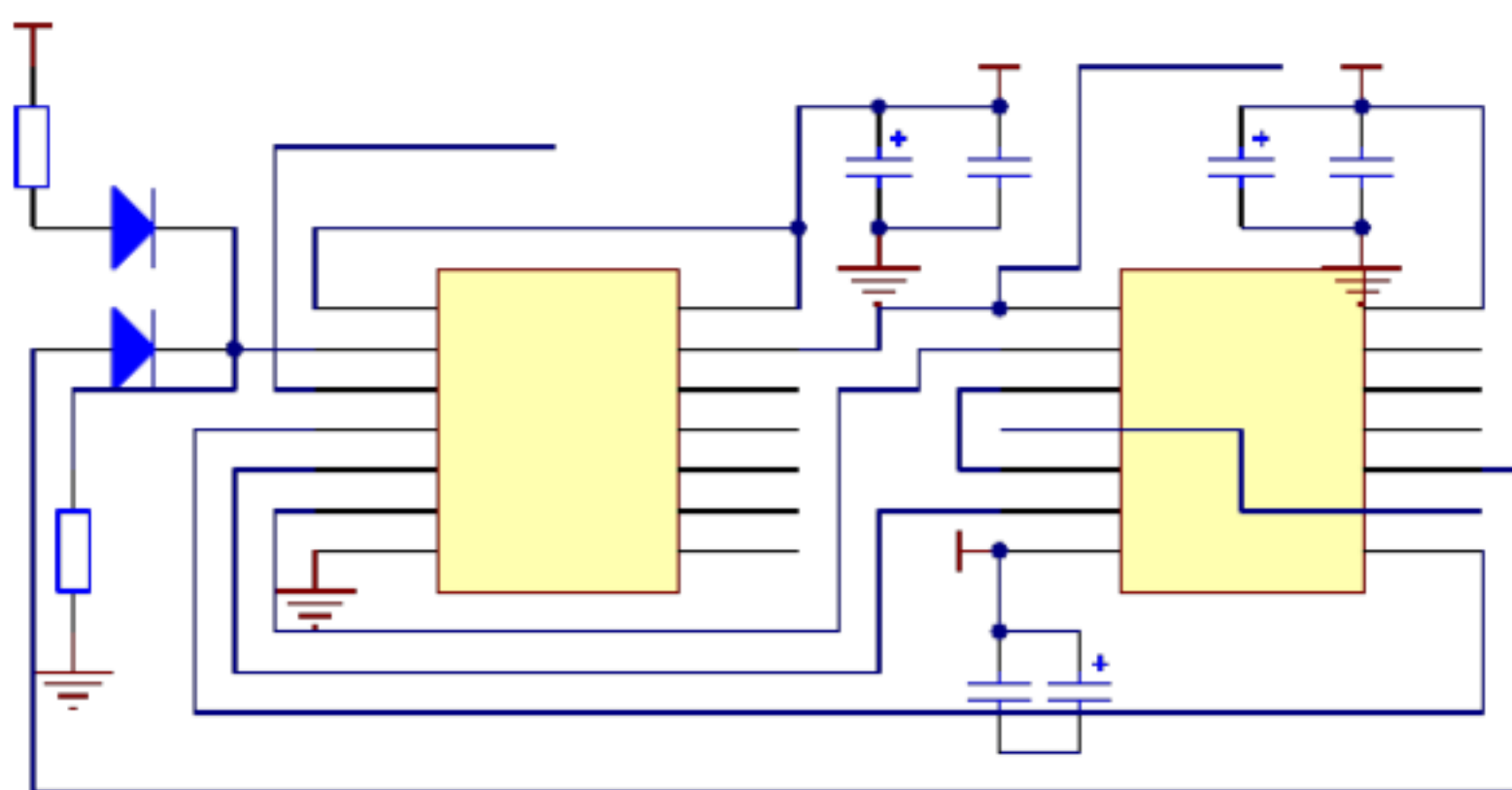


图 2-4 伪随机信号发生电路图

### 2.2.2 伪随机信号的产生

伪随机信号的产生与数字信号的产生原理完全相同，只是位数和频率上有所不同而已。这里因为时钟频率为 10MHz，用单片机比较难以实现，所以我们可以使用 FPGA 轻易的产生这个时钟信号。其电路如图 2-5 所示。

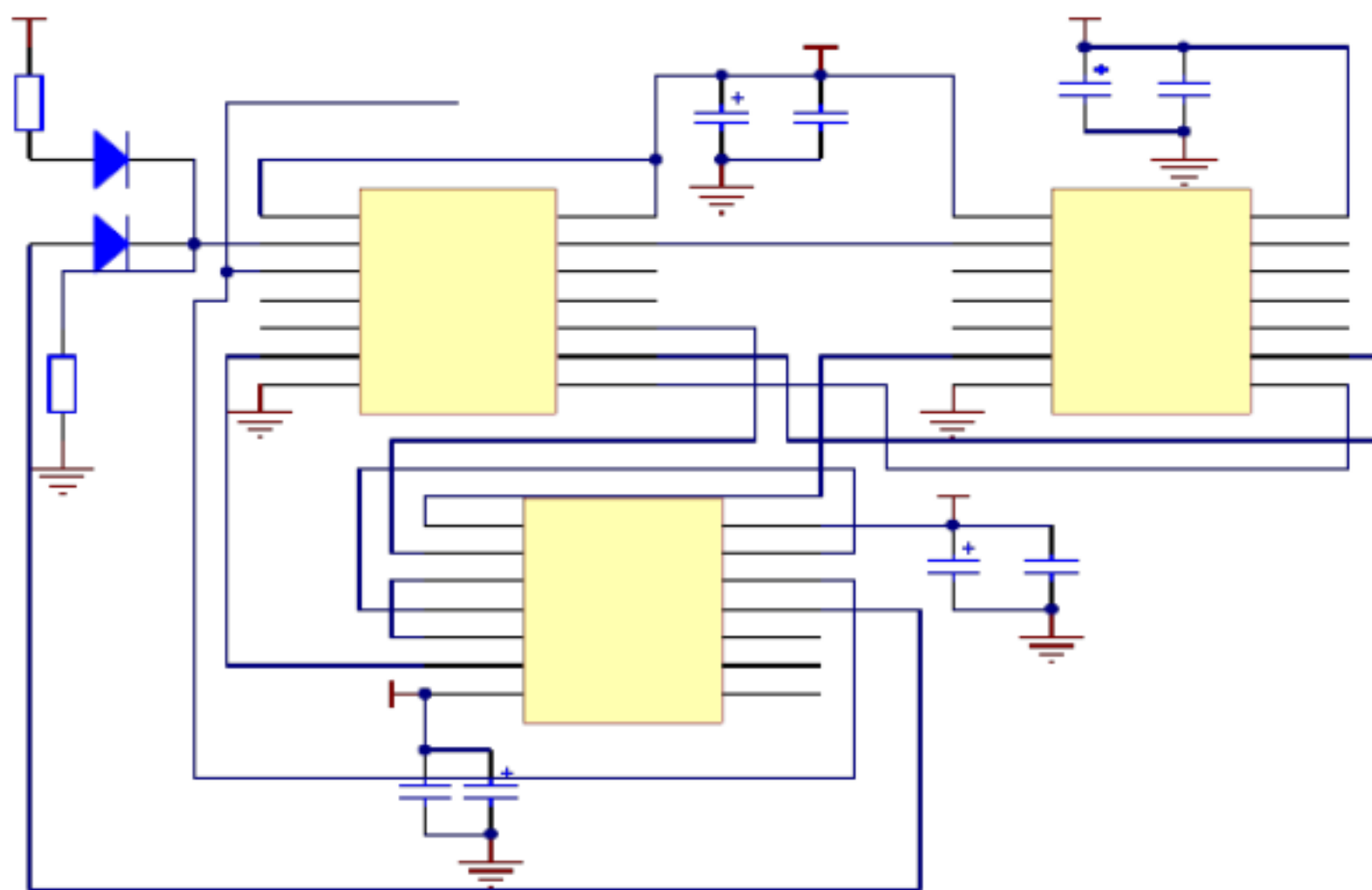


图 2-5 伪随机信号发生电路图

## 2.3 眼图显示方法

### 2.3.1 眼图的概念

在示波器上显示数字信号时，由于示波器的余辉作用，扫描所得的每一个码元波形将重叠在一起，从而形成眼图。通过观察眼图的形状可以得知信道中，信号传输的码间串扰的强弱。当存在噪声时，噪声将叠加在信号上，观察到的眼图的线迹会变得模糊不清。若同时存在码间串扰，‘眼睛’将张开得更小。与无码间串扰时的眼图相比，原来清晰端正的细线迹，变成了比较模糊的带状线，而且不很端正。噪声越大，线迹越宽，越模糊；码间串扰越大，眼图越不端正。

### 2.3.2 眼图的意义

为更形象的说明眼图的意义，将眼图的模型画在图 2-3 中。

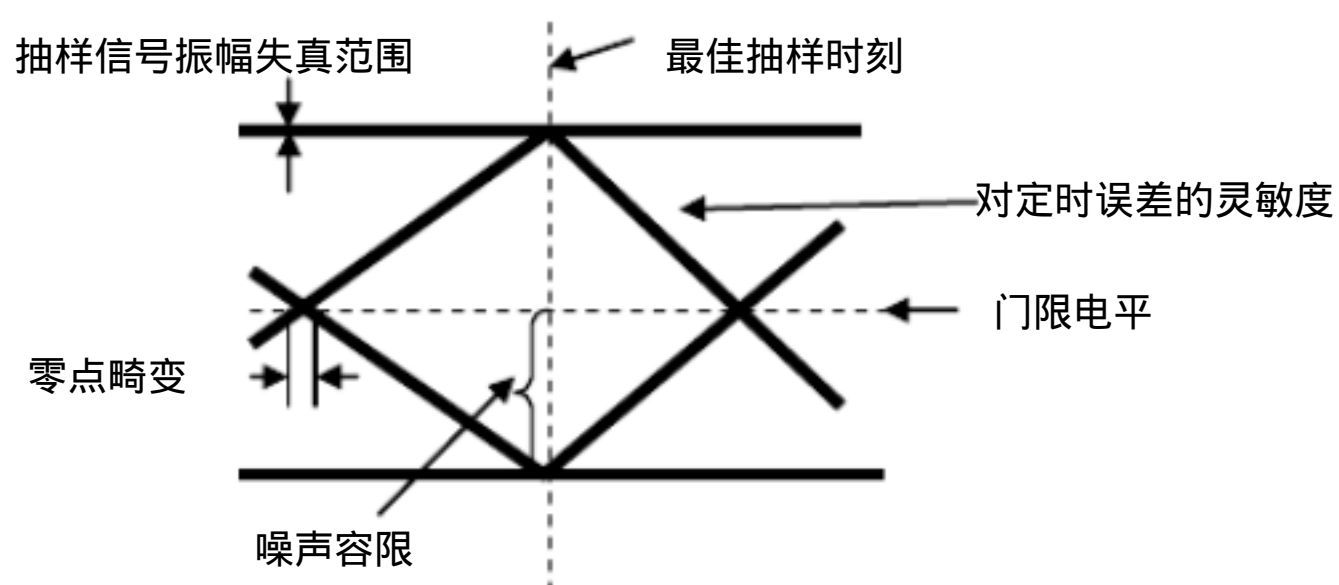


图 2-3 眼图模型

在此图中我们可以得到以下信息：

a.最佳抽样时刻应在“眼睛”张开最大的时刻。

b.对定时误差的灵敏度可由眼图斜边的斜率决定。斜率越大，对定时误差就越灵敏。

c.在抽样时刻上，眼图上下两分支阴影区的垂直高度，表示最大信号畸变。

e.眼图中央的横轴位置应对应判决门限电平。

f.在抽样时刻上，上下两分支离门限最近的一根线迹至门限的距离表示各相应电平的噪声容限，噪声瞬时值超过它就可能发生错误判决。

g.对于利用信号过零点取平均来得到定时信息的接收系统，眼图倾斜分支与横轴相交的区域的大小，表示零点位置的变动范围，这个变动范围的大小对提取定时信息有重要的影响。

### 2.3.3 眼图的显示

观察眼图的方法是：用示波器的一个通道接在接收滤波器的输出端，将与数字信号同频率的时钟信号接入到外触发端，这时示波器屏幕上便能观察到眼图。

## 2.4 同步信号提取

同步信号提取主要是靠程序来处理的。首先我们用 50MHz 的时钟作为计数的触发时钟对数字信号的高电平脉宽进行计数，并将所计得的数据  $cnt$  保存在寄存器中，然后将数字信号各个连续高电平时间内计得的数进行比较，最终选出最小的计数值  $cnt_{min}$ 。然后对 50MHz 进行  $2cnt_{min}$  分频，这样我们就得到了数字信号的同步时钟信号  $V_{clock}$ 。由于数字信号使用的是 8 位的 m 序列，出去全 0 序列，总共有  $2^8-1$  种组态，因此在这里我们只需计  $2^8-1$  个  $cnt$  便可以比较得出  $cnt_{min}$ 。

要想让  $V_{clock}$  与数字信号同步，则必须要做到二者的同相。在这里首先用  $V_{clock}$  与数字信号进行异或运算，计算出相位差，并判断这个差值得正负，然后进行相位补偿，便可得到与数字信号同频率、同相位的时钟信号  $V_{4-syn}$ 。

### 第三章电路与程序设计

#### 3.1 电路的设计

##### 3.1.1 系统总体框图

系统总体框图如图 3-1 所示。

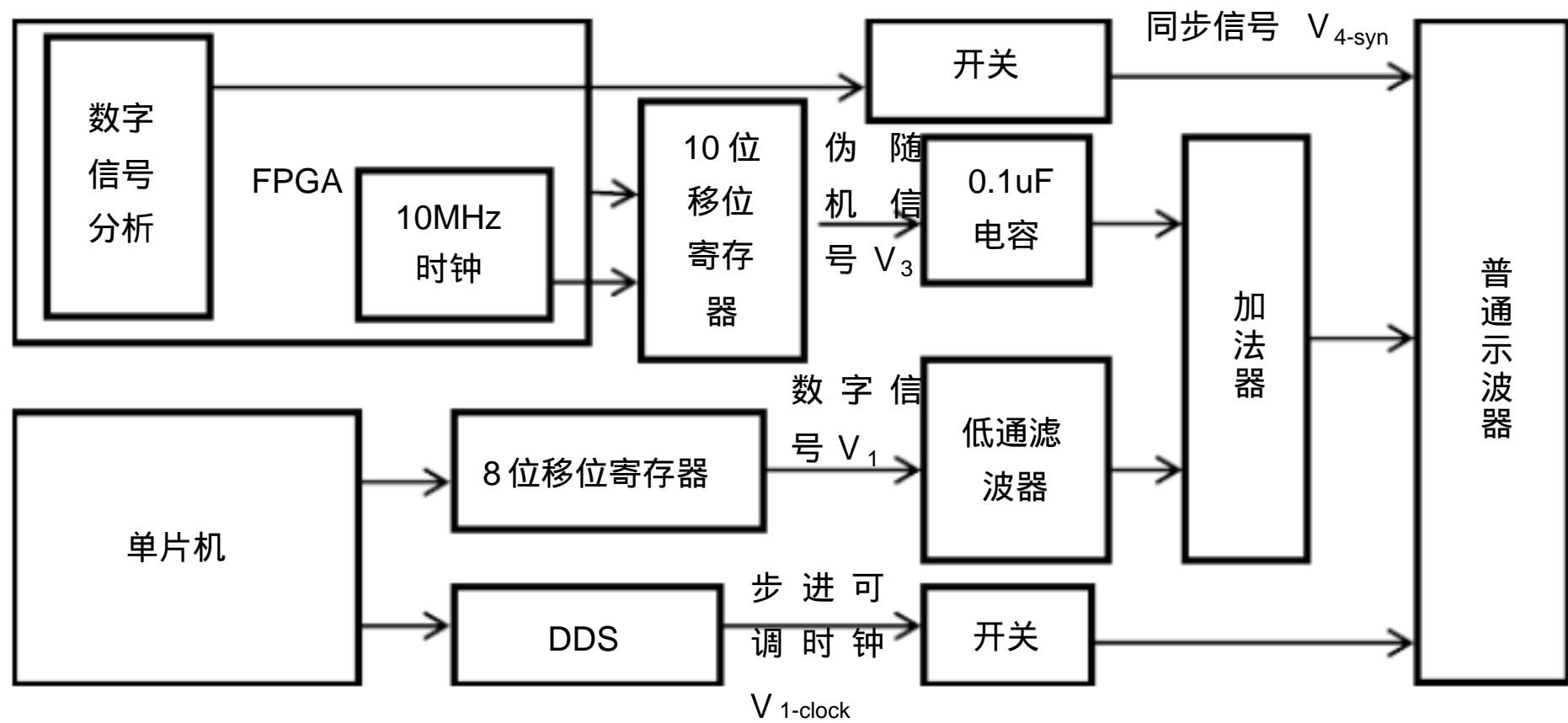


图 3-1 系统总体框图

##### 3.1.2 电源

电源由变压部分、滤波部分、稳压部分组成。为整个系统提供  $\pm 5V$  或者  $\pm 12V$  电压，确保电路的正常稳定工作。这部分电路比较简单，都采用三端稳压管实现，故不作详述。

#### 3.2 程序的设计

##### 3.2.1 程序模块

本设计的程序分为以下几块：伪随机信号数据率 10KHz 步进控制，同步信号提取。

##### 3.2.2 程序流程图

伪随机信号数据率 10KHz 步进控制流程如图 3-2 所示。其工作流程为，单片机上电复位初始化后便立即将初始频率值传输给 DDS，DDS 便输出一定频率的时钟信号，当检测到有键按下是便对当前频率值进行步进 10K 运算，然后用这个新频率值代替旧频率值，再传输给 DDS，输出新频率的时钟信号。

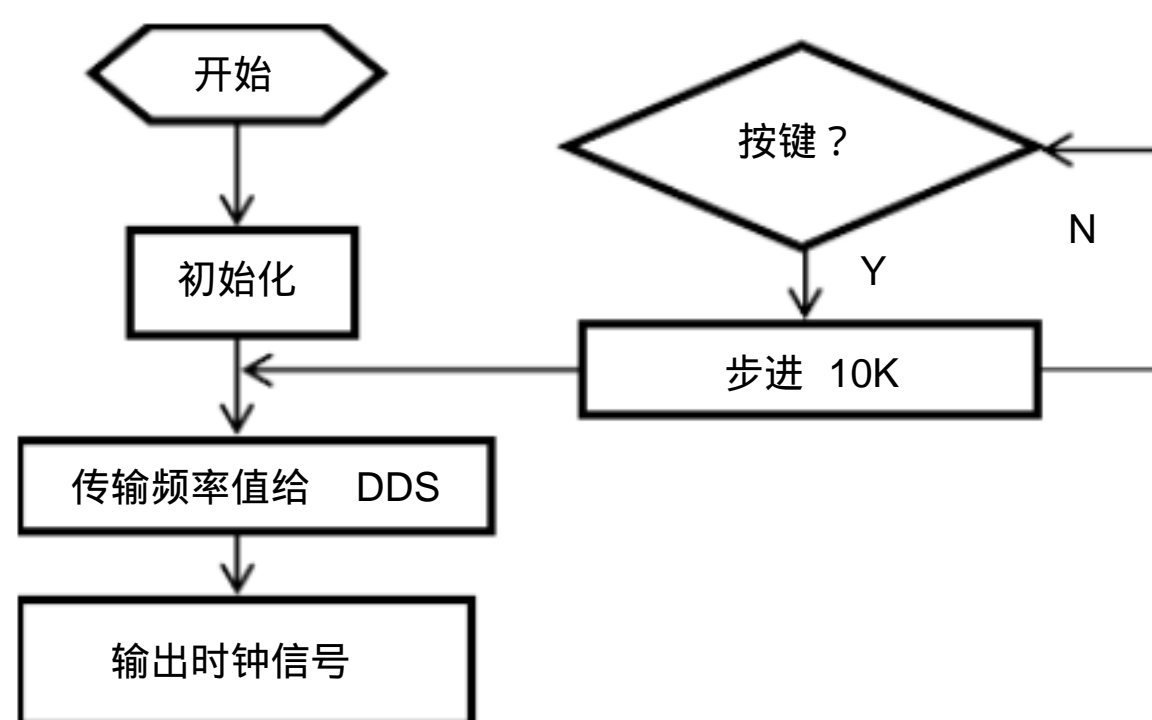


图 3-2 10K 步进控制流程图

同步时钟信号提取流程如图 3-3 所示。首先用 FPGA 产生频率 50MHz 的时钟信号（这个频率越高，同步时钟的提取越精确），当检测到数字信号 V1 为高电平是就对 50MHz 进行计数，计数值给寄存器  $cnt_{min}$ ，紧接着对下一个数字信号脉宽进行计数，计数值为  $cnt$ ，把  $cnt$  与  $cnt_{min}$  进行比较，将较小的那个数据存数到  $cnt_{min}$  寄存器中，如此循环  $2^8$  次便能测出同步信号  $V_{clock}$  频率。程序流程如图 3-3 所示。

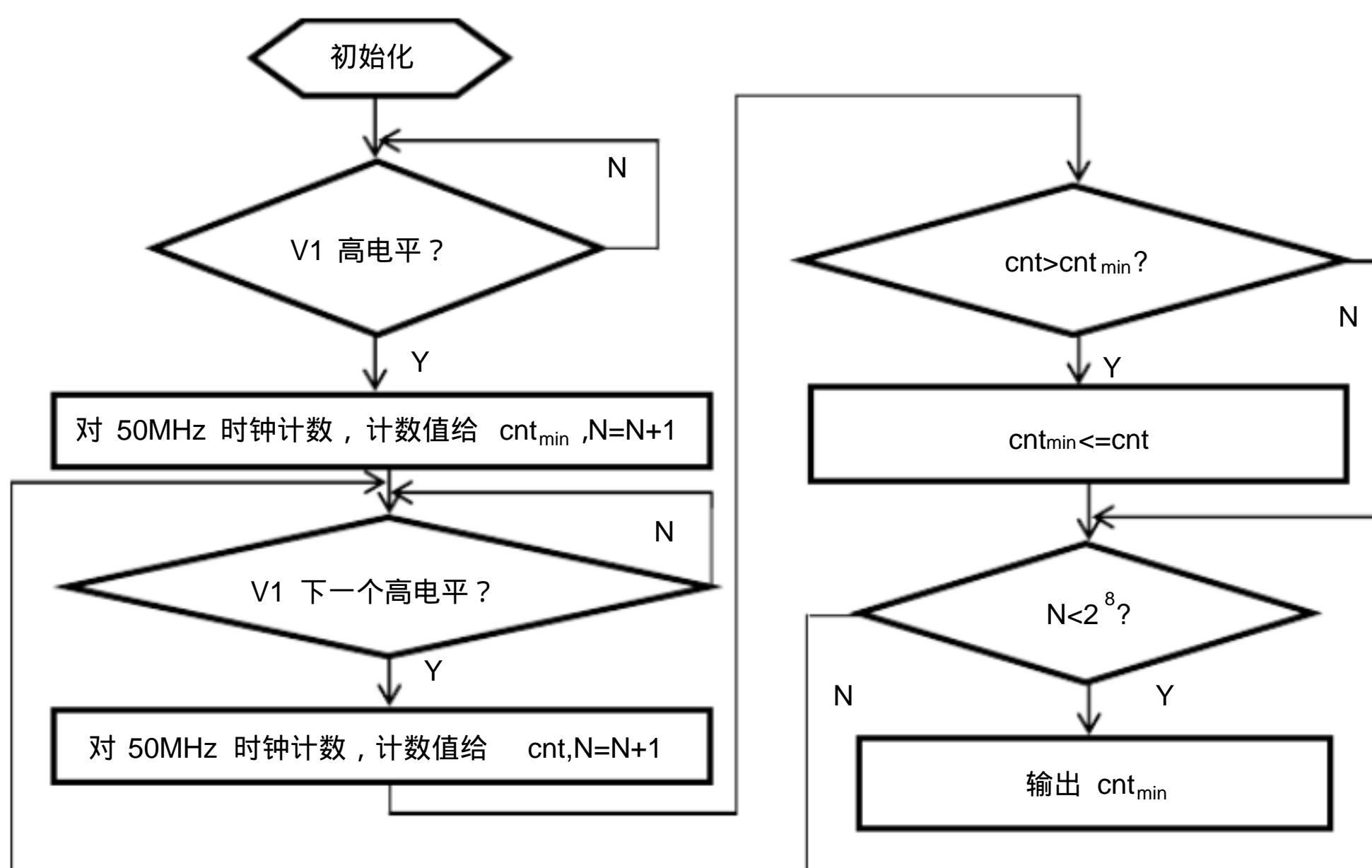


图 3-3 同步时钟信号提取

## 第四章测试方案与测试结果

### 4.1 测试方案

#### 4.1.1 硬件测试

a.低通滤波电路的测试。分别对截止频率为 100KHz、200KHz 和 500KHz 三个通道接入  $V_{P-P}=2V$ ，频率可调的标准正弦波信号。通过调节频率来测试低通滤波器的性能。

b.加法器电路的测试。将数字信号和通过 0.1uF 电容耦合过来的伪随机信号输入到加法电路的两个输入端，加法电路的输出端接到普通示波器上，调节伪随机信号的幅值观察输出波形。

#### 4.1.2 软件仿真测试

$V_{1-clock}$  时钟信号步进 10KHz 的测试，测试时给定一个初始频率值，然后通过按一下增步进按钮增加 10KHz 和按一下减步进按钮减少 10KHz 的规律变化，把这个  $V_{1-clock}$  信号接入到示波器的 CH1 端，观察其电平值和频率的变化规律。

#### 4.1.3 硬件软件联调

将加法器电路输出的混合信号输入到普通示波器的 CH1 通道，将  $V_{1-clock}$  信号输入到普通示波器的外触发端。调节伪随机信号的幅值，观察眼图的变化。

### 4.2 测试条件与仪器

测试条件：检查多次，仿真电路和硬件电路必须与系统原理图完全相同，并且检查无误，硬件电路保证无虚焊。

测试仪器：普通示波器，信号源。

### 4.3 测试结果及分析

#### 4.3.1 测试结果 (数据)

低通滤波器性能测试结果如下表 4-1、4-2、4-3 所示

表 4-1 100KHz 低通滤波器性能测试

输入信号频率 /Hz	10K	50K	100K	150K	500K	1M	1.2M
输出峰 -峰值 /V	2.0	2.2	1.72	0.84	0.12	0.01	0.002

表 4-2 200KHz 低通滤波器性能测试

输入信号频率 /Hz	50K	100K	200K	500K	1M	1.5M	2M
输出峰 -峰值 /V	2.0	2.0	1.85	0.28	0.08	0.04	0.008

表 4-3 500KHz 低通滤波器性能测试

输入信号频率 /Hz	100K	300K	400K	500K	2MK	4M	5M
输出峰 -峰值 /V	2.0	2.0	1.86	1.48	0.12	0.01	0.015

10KHz 步进可调测试结果如下表 4-4 所示。

表 4-4 10KHz 步进测试

初始值	增	增	增	增	增	增	增	增	增
10K	20K	30K	40K	50K	60K	70K	80K	90K	100K
初始值	减	减	减	减	减	减	减	减	减
100K	90K	80K	70K	60K	50K	40K	30K	20K	10K

加法电路测试效果。

图 4-1 为原数字信号的实际图，图 4-2 为伪随机信号的实际图，图 4-3 为二者经过加法电路后的叠加图。

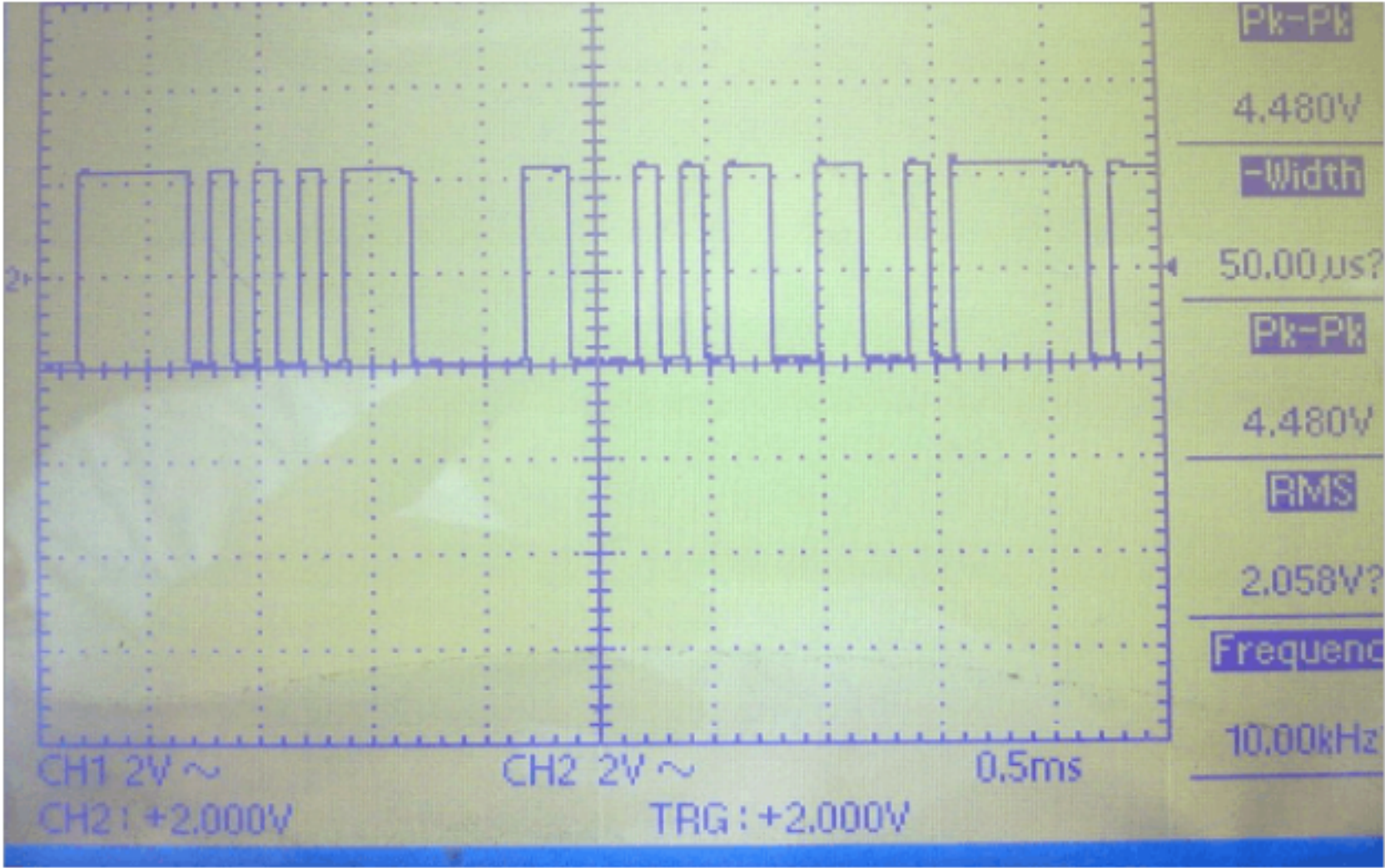


图 4-1 数字信号测试图





图 4-2 伪随机信号测试图

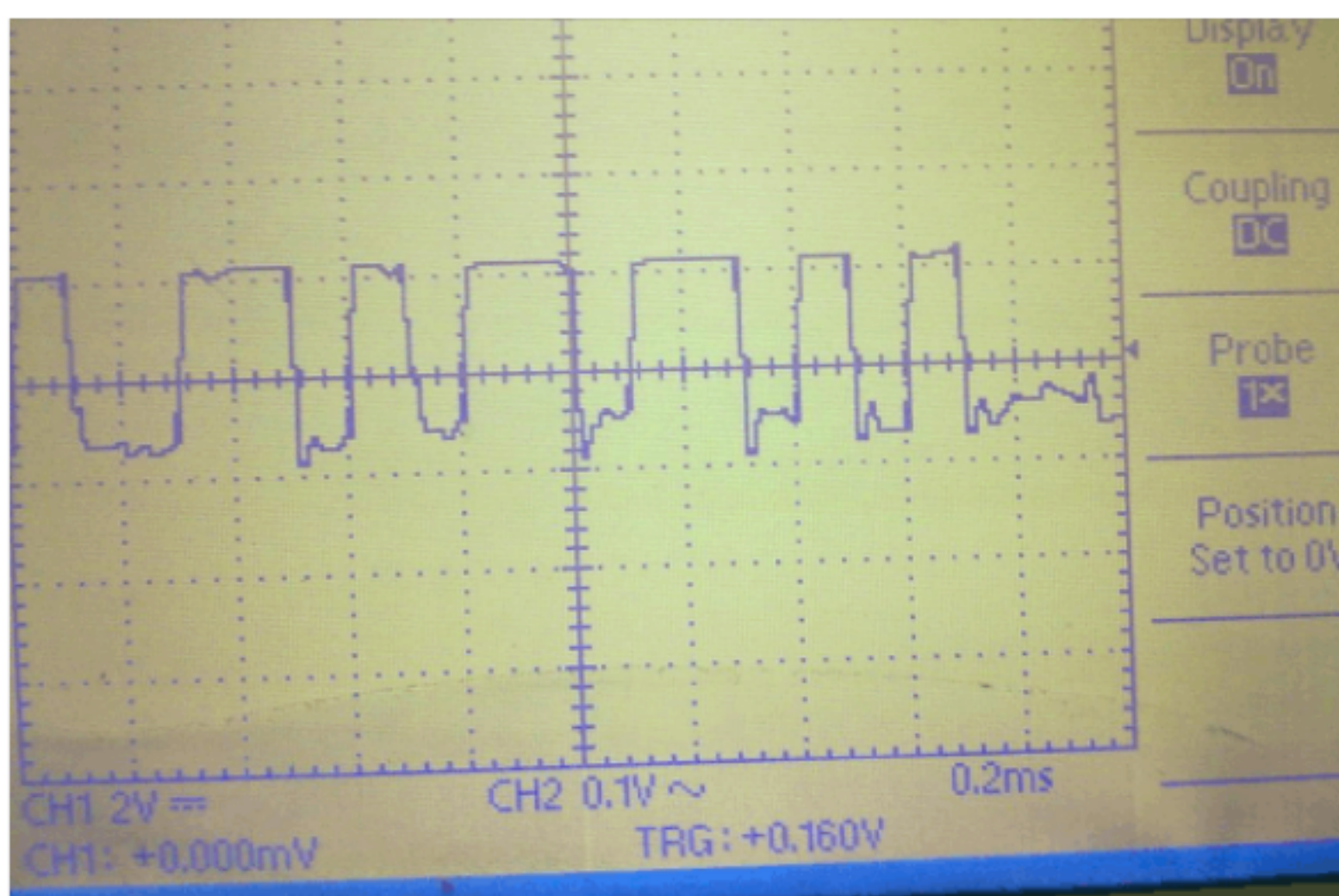


图 4-3 叠加测试图

眼图测试效果

图 4-4 表示未叠加伪随机信号时的眼图效果。图 4-5 表示叠加伪随机信号后的效果。



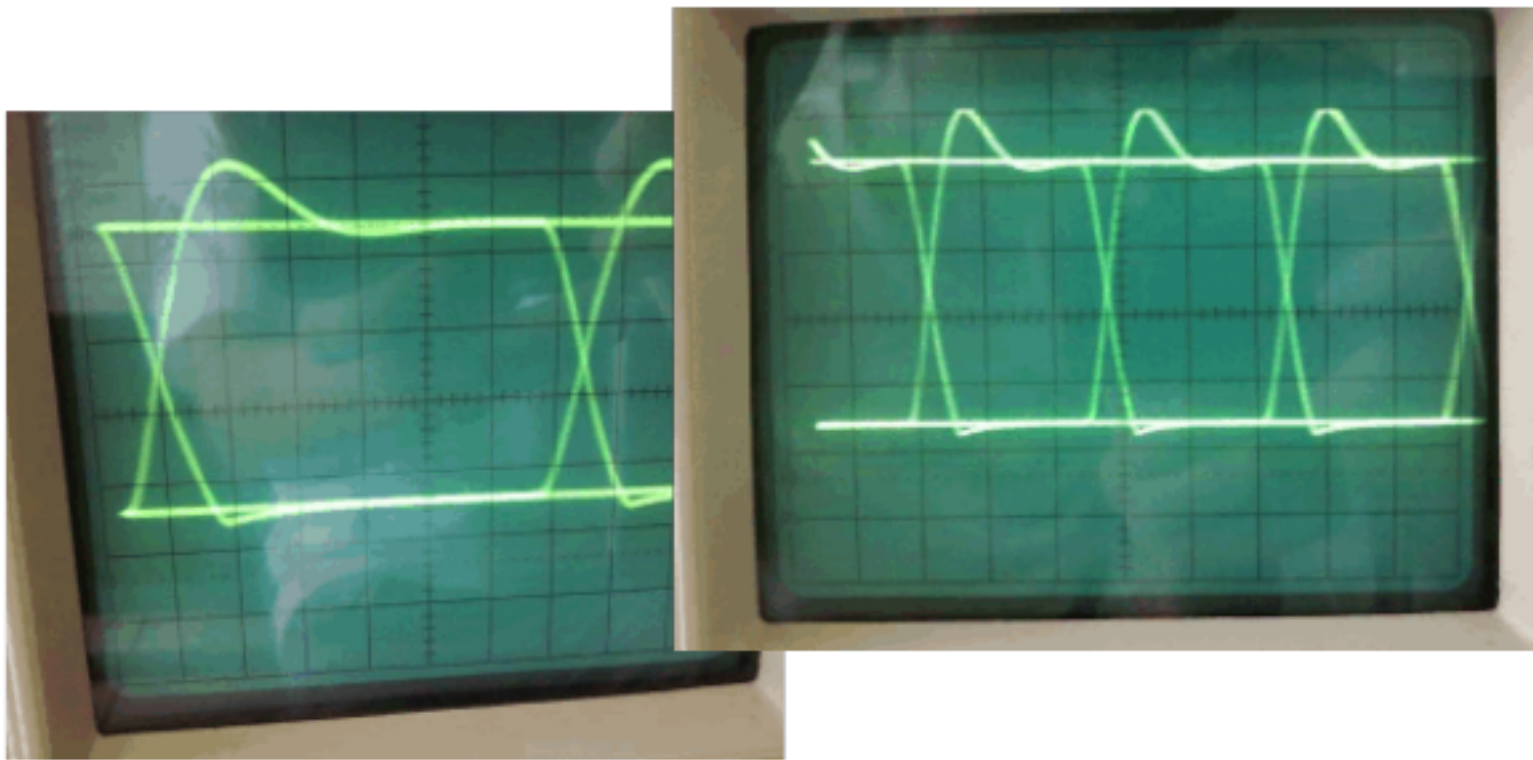


图 4-4 数字信号眼图

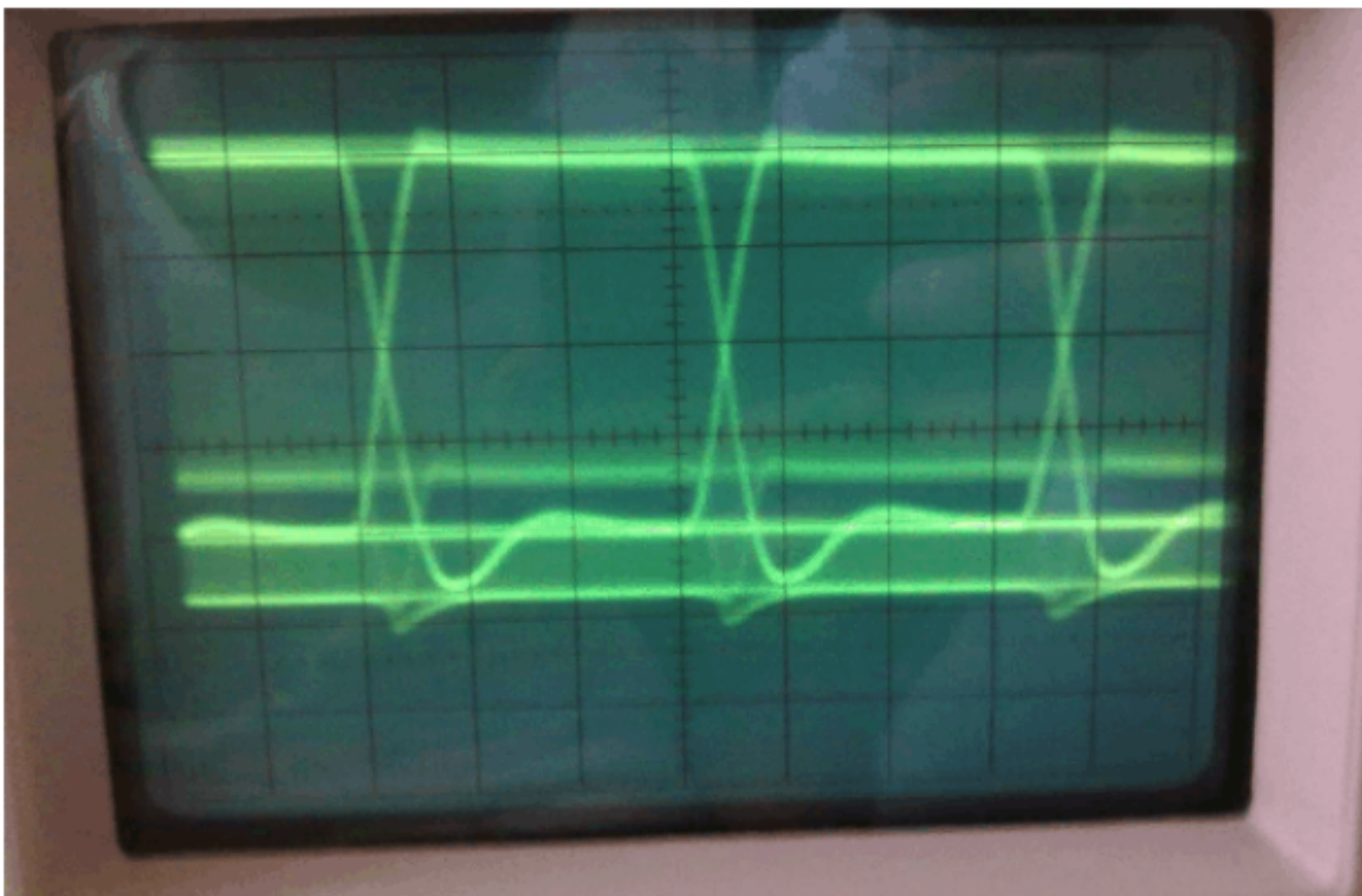


图 4-5 叠加伪随机信号后的眼图

#### 4.3.2 测试分析与结论

a.对于 100KHz 低通滤波，当输入为 100KHz 时，衰减

$$A_{dB} = -20 \lg \frac{1.74}{2} = 1.21 \text{ dB}$$

1.21dB<3dB,所以截止频率满足要求。

当频率从 100KHz 增大到 1MHz 时，衰减

$$A_{dB} = -20 \lg \frac{0.01}{2} = 46 \text{ dB}$$

完全可以满足 40dB/十倍频程的衰减要求。

b. 对于 200KHz 低通滤波，当输入为 200KHz 时，衰减

$$A_{dB} = -20 \lg \frac{1.85}{2} = 0.6 \text{ dB}$$

0.6dB<3dB,所以截止频率满足要求。

当频率从 200KHz 增大到 2MHz 时，衰减

$$A_{dB} = -20 \lg \frac{0.008}{2} = 47.9 \text{ dB}$$

完全可以满足 40dB/十倍频程的衰减要求。

c. 对于 500KHz 低通滤波，当输入为 500KHz 时，衰减

$$A_{dB} = -20 \lg \frac{1.48}{2} = 2.6 \text{ dB}$$

2.6dB<3dB,所以截止频率满足要求。

当频率从 500KHz 增大到 5MHz 时，衰减

$$A_{dB} = -20 \lg \frac{0.015}{2} = 42.3 \text{ dB}$$

完全可以满足 40dB/十倍频程的衰减要求。

对于 10K 步进的测试,由表中数据可知，每按一次步进增按钮，频率会增加 10KHz，连续按 9 次，频率可以从 10KHz 可增大到 100KHz。每按一次减按钮，频率会减少 10KHz，连续按 9 次，频率可以从 100KHz 减小到 10KHz。

通过观察示波器显示得知，所产生的数字信号与理论相符。并且可以按 10Kbps 可调。误差满足要求。

从图 4-4 可以观察到，为叠加伪随机信号时，我们随观察到的眼图的线迹比较细，且非常清晰，这说明信道中对数字信号的干扰不叫小，信道（低通滤波器模拟信道）的传输特性良好。当叠加伪随机信号后，眼图的线迹就显得很粗，

且会出现“双眼皮”现象。这说明，数字信号在信道中传输过程中收到了干扰，或者产生了码元串绕。现实生活中我们可以通过观察眼图的好坏修复和改善信道。

综上所述，本设计达到设计要求。

## 第五章结论

经过本次设计，我们认识到基础知识的重要性。很多原理都是课本上原有的，比如本设计中的数字信号  $m$  序列的产生和眼图的显示，都可以在通信原理课本中找到理论根据。因此，我们应该加强对课本理论知识的学习和应用。在设计中，我们获得很多技能，比如查阅资料、电路设计、系统调试、分工协作等。

最后感谢主办单位给我们大学生提供这一难得的展现自我的机会，也感谢各位评委老师的辛勤工作。

## 参考文献

- [1] 黄智伟.全国大学生电子设计大赛系统设计 .北京：北京航空航天大学出版社，  
2011
- [2] 樊昌信.通信原理教程第 2 版.北京：电子工业出版社， 2010
- [3] 王冠，于一鸣.面向 CPLD/FPGA 的 Verilog 设计.北京：机械工业出版社， 2007
- [4] 康华光.电子技术基础模拟部分 .北京：高等教育出版社， 2006

## 附录 1：电路原理图

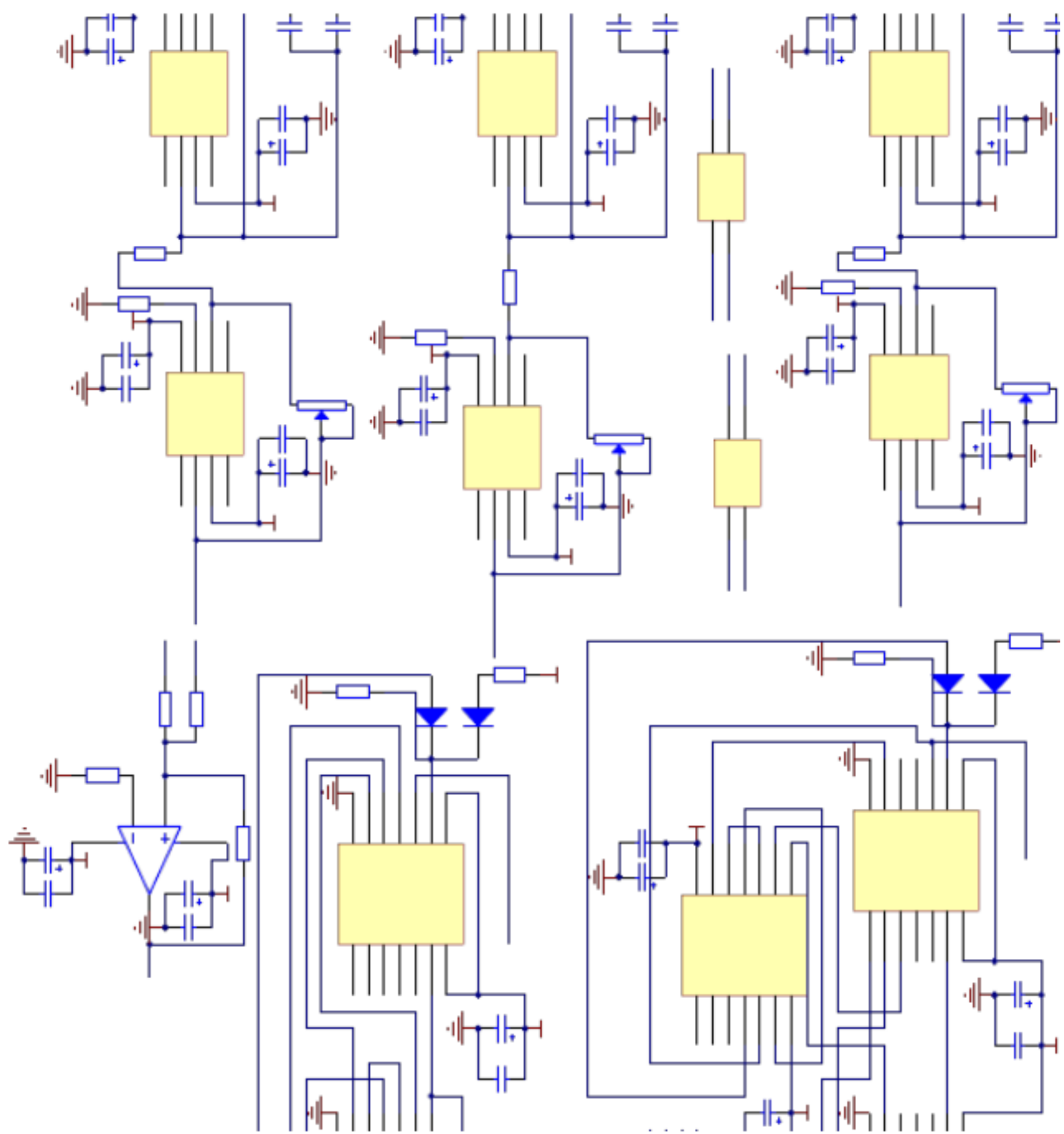


图 1 总体硬件图



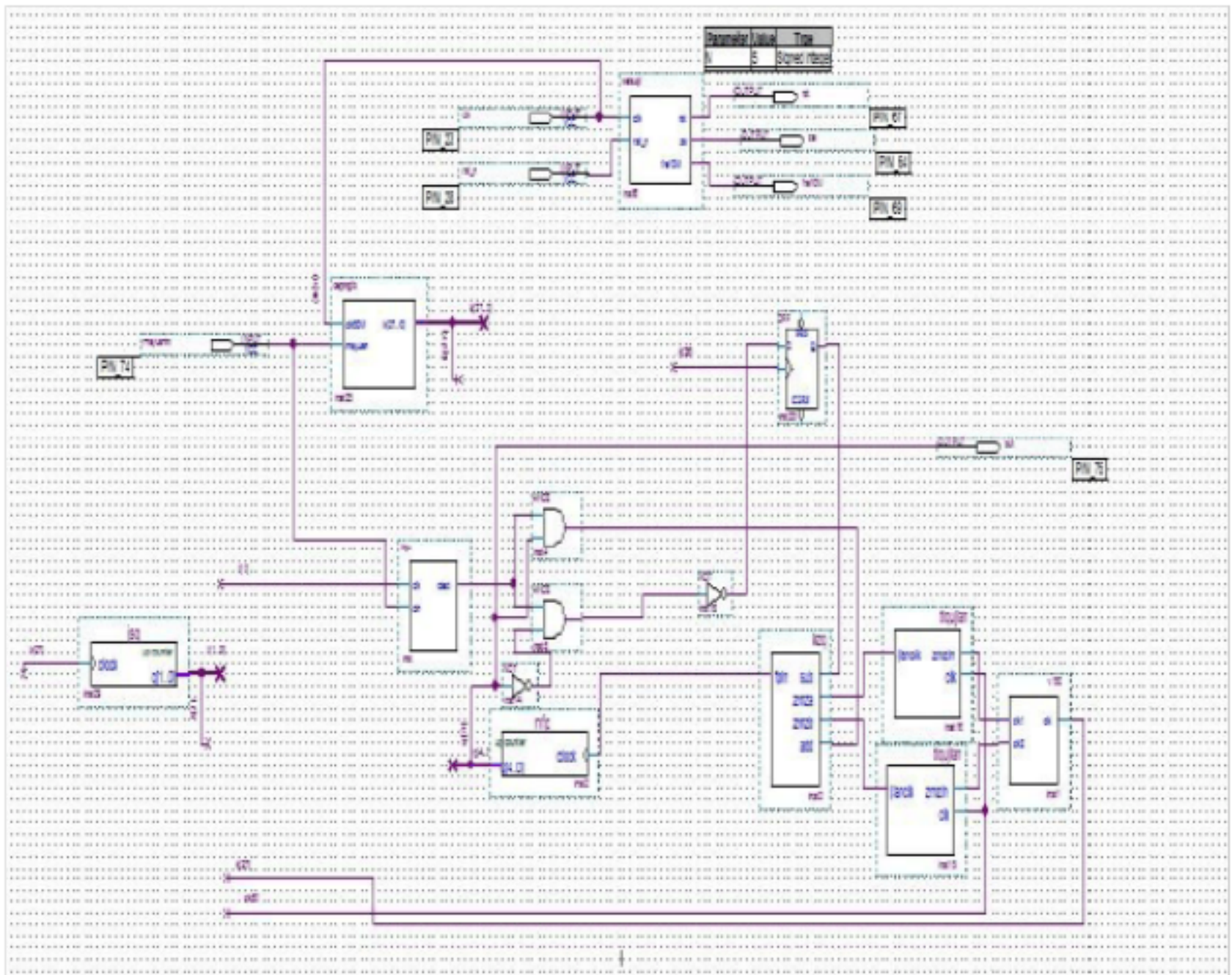


图 2 FPGA 位同步信号提取模块

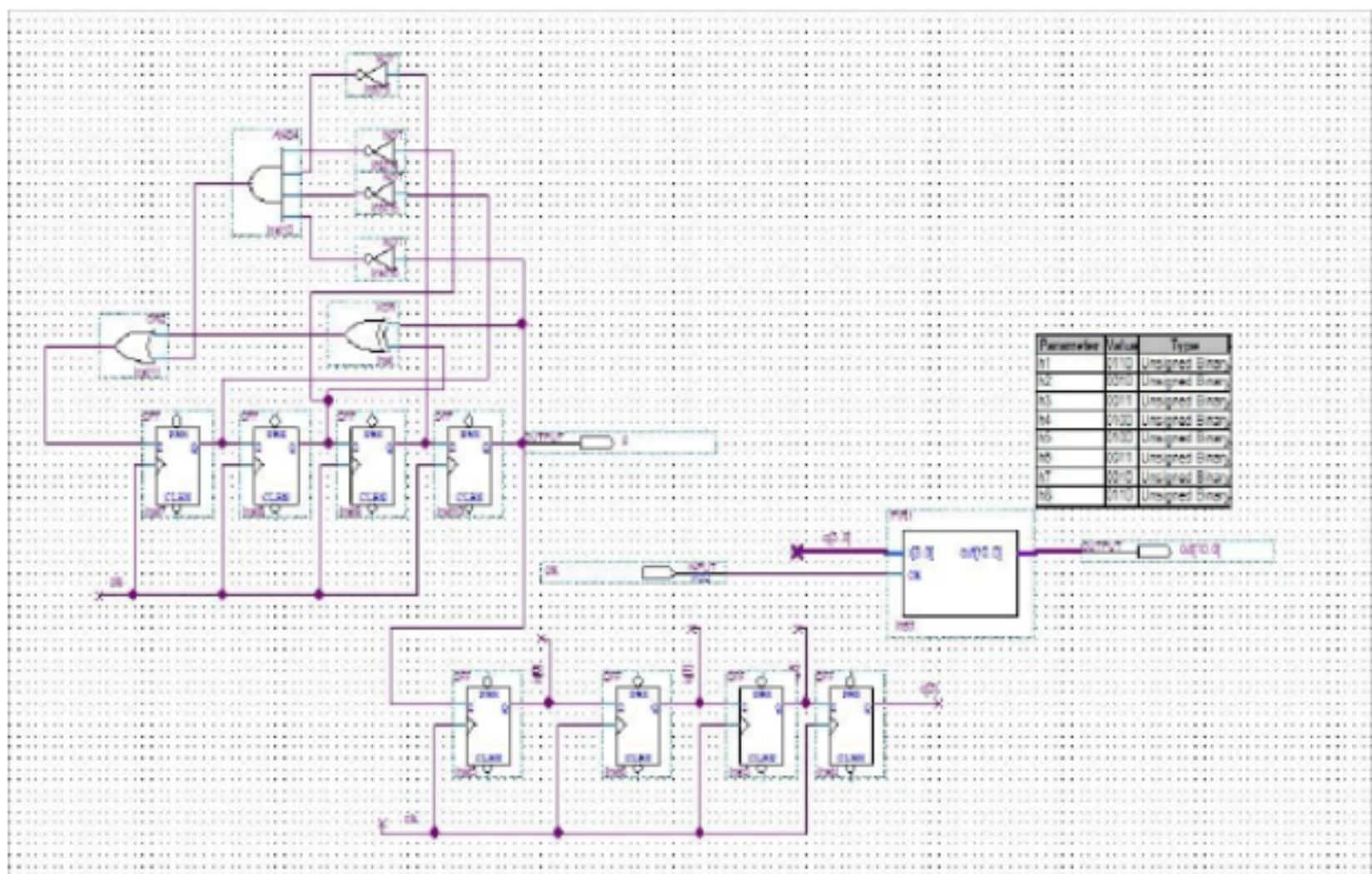


图 3 FPGA 生成的 FIR 模块

## 附录 2：源程序

### DDS 控制源程序

```
#include <reg51.h>
#include <stdio.h>
#include <intrins.h>
#define uchar unsigned char
#define uint unsigned int
sbit ad9850_w_clk    =P2^0;    //P2.0 口接 ad9850 的 w_clk 脚/PIN7
sbit ad9850_fq_up    =P2^1;    //P2.1 口接 ad9850 的 fq_up 脚/PIN8
sbit ad9850_rest     =P2^2;    //P2.2 口接 ad9850 的 rest 脚/PIN12
sbit ad9850_bit_data =P2^7;    //P2.7 口接 ad9850 的 D7 脚/PIN25
//P1 为 8 位数据
sbit key1=P1^0;
sbit key2=P1^1;
sbitclk=P1^2;    //指示灯 +/-
sbitzs=P0^7;
sbitrst=P1^6;
void ad9850_reset()
{
    ad9850_w_clk=0;
    ad9850_fq_up=0;
    //rest 信号
    ad9850_rest=0;
    ad9850_rest=1;
    ad9850_rest=0;
}
void ad9850_reset_serial()
{
    ad9850_w_clk=0;
    ad9850_fq_up=0;
    //rest 信号
    ad9850_rest=0;
    ad9850_rest=1;
    ad9850_rest=0;
    //w_clk 信号
    ad9850_w_clk=0;
    ad9850_w_clk=1;
    ad9850_w_clk=0;
    //fq_up 信号
    ad9850_fq_up=0;
    ad9850_fq_up=1;
    ad9850_fq_up=0;
}
```

```

//*****//
//          向 ad9850 中写命令与数据 （并口）          //
void ad9850_wr_parrel(unsigned char w0,double frequency)
{
    unsigned char w;
    longint y;
    double x;// 计算频率的 HEX 值
    x=4294967295/125;// 适合 125M 晶振 / , 如果时钟频率不为 180MHZ , 修改该处的频率值 , 单
位 MHz !!!
    frequency=frequency/1000000;
    frequency=frequency*x;
    y=frequency;// 写 w0 数据
    w=w0;
    P1=w;          //w0
    ad9850_w_clk=1;
    ad9850_w_clk=0;// 写 w1 数据
    w=(y>>24);
    P1=w;          //w1
    ad9850_w_clk=1;
    ad9850_w_clk=0;// 写 w2 数据
    w=(y>>16);
    P1=w;          //w2
    ad9850_w_clk=1;
    ad9850_w_clk=0;// 写 w3 数据
    w=(y>>8);
    P1=w;          //w3
    ad9850_w_clk=1;
    ad9850_w_clk=0;// 写 w4 数据
    w=(y>>0);
    P1=w;          //w4
    ad9850_w_clk=1;
    ad9850_w_clk=0;// 移入始能
    ad9850_fq_up=1;
    ad9850_fq_up=0;
}
//*****//
//          向 ad9850 中写命令与数据 （串口）          //
void ad9850_wr_serial(unsigned char w0,double frequency)
{
    unsigned char i,w;
    longint y;
    double x;// 计算频率的 HEX 值
    x=4294967295/125;// 适合 125M 晶振 //如果时钟频率不为 180MHZ , 修改该处的频率值 , 单
位 MHz !!!

```



```

frequency=frequency/1000000;
frequency=frequency*x;
y=frequency;// 写 w4 数据
w=(y>=0);
for(i=0;i<8;i++)
{
ad9850_bit_data=(w>>i)&0x01;
ad9850_w_clk=1;
ad9850_w_clk=0;
} // 写 w3 数据
w=(y>>8);
for(i=0;i<8;i++)
{
ad9850_bit_data=(w>>i)&0x01;
ad9850_w_clk=1;
ad9850_w_clk=0;
} // 写 w2 数据
w=(y>>16);
for(i=0;i<8;i++)
{
ad9850_bit_data=(w>>i)&0x01;
ad9850_w_clk=1;
ad9850_w_clk=0;
} // 写 w1 数据
w=(y>>24);
for(i=0;i<8;i++)
{
ad9850_bit_data=(w>>i)&0x01;
ad9850_w_clk=1;
ad9850_w_clk=0;
} // 写 w0 数据
w=w0;
for(i=0;i<8;i++)
{
ad9850_bit_data=(w>>i)&0x01;
ad9850_w_clk=1;
ad9850_w_clk=0;
} // 移入始能
ad9850_fq_up=1;
ad9850_fq_up=0;
}
//                               测试程序 1000Hz                               //
ucharkey_scan(void);
voidkey_scan_delay(uchar y);

```

```

void display();
main()
{
    unsigned long intfre,i;
    P0=0x00;
    //    P1=0x00;
    P2=0x00;
    P3=0x00;
    fre =10000;    //用户需要写入的频率，单位    Hz
    zs=1;
    rst=0;
    i=0;
    //串行写 1000Hz 程序
    ad9850_reset_serial();
    while(1)
    {
        key1=1;key2=1;
        switch(key_scan())
        {
            case 1:display();  fre+=10000;  if(fre>100500)  fre=100000;  break;
            case 2:display();    if(fre>11500)  fre-=10000;    /*  ad9850_wr_serial(0x00,fre);    */
break ;
            default: break;
        }
        ad9850_wr_serial(0x00,fre);
        rst=1;
        if(i<100) { zs=1; i++; }
        elsezs=0;
    }
}
ucharkey_scan(void)
{
    key1=1;  //扫描第一行
    if(key1==0)
        key_scan_delay(10);
    if(key1==0)
    {
        while(0==key1);
        return 1;
    }
    key2=1;
    if(key2==0)
        key_scan_delay(10);
    if(key2==0)
    {

```

```

        while(0==key2);
        return 2;
    }}

voidkey_scan_delay(uchar y)
{
    uchar i,j;
    for (i=0;i<y;i++)
        for(j=0;j<120;j++);
}

void display()
{
    clk=0;
    key_scan_delay(1000);
    clk=1;
}

```