

大学毕业论文

# 基于 FPGA 的数字钟设计 (VHDL 语言实现)

## 摘要

本设计为一个多功能的数字钟，具有年、月、日、时、分、秒计数显示功能，以 24 小时循环计数；具有校对功能以及整点报时功能。

本设计采用 EDA 技术，以硬件描述语言 VHDL 为系统逻辑描述手段设计文件，在 MaxplusII 工具软件环境下，采用自顶向下的设计方法，由各个基本模块共同构建了一个基于 FPGA 的数字钟。

系统主芯片采用 EP1K100QC208-3，由时钟模块、控制模块、计时模块、数据译码模块、显示以及报时模块组成。经编译和仿真所设计的

程序，在可编程逻辑器件上下载验证，本系统能够完成年、月、日和时、分、秒的分别显示，由按键输入进行数字钟的校时、清零、启停功能。

关键词 数字钟；硬件描述语言；VHDL；FPGA；键盘接口

## Abstract

The design for a multi-functional digital clock, with a year, month, day, hours, minutes and seconds count display to a 24-hour cycle count; have proof functions and the whole point timekeeping function.

The use of EDA design technology, hardware-description language VHDL description logic means for the system design documents, in MaxplusII tools environment, a top-down design, by the various modules together build a FPGA-based digital clock.

The main system chips used EP1K100QC208-3, make up of the clock module, control module, time module, data decoding module, display and broadcast module. After compiling the design and simulation procedures, the programmable logic device to download verification, the system can complete the year, month, day and the hours, minutes and seconds respectively, using keys to modify, cleared , start and stop the digital clock.

Keywords digital clock; hardware description language; VHDL; FPGA; keyboard interface

## 目录

1	绪论 .....
	选题背景 .....
	课题相关技术的发展 .....
	课题研究的必要性 .....
	课题研究的内容 .....
2	FPGA 简介 .....
	FPGA 概述 .....
	FPGA 基本结构 .....

	FPGA 系统设计流程.....
	FPGA 开发编程原理.....
3	数字钟总体设计方案.....
	数字钟的构成.....
	数字钟的工作原理.....
4	单元电路设计 .....
	分频模块电路设计与实现 .....
	校时控制模块电路设计与实现 .....
	键盘接口电路原理.....
	键盘接口的 VHDL 描述 .....
	计数模块设计与实现.....
	秒计数模块.....
	日计数模块.....
	月计数和年计数模块 .....
	动态扫描及显示电路设计与实现 .....
	动态扫描模块.....
	显示模块 .....
5	实验结论与研究展望.....
	实验结论 .....
	研究展望 .....
	致谢.....
	附录.....
	参考文献 .....

# 基于 FPGA 的数字钟设计

## 1 绪论

现代社会的标志之一就是信息产品的广泛使用，而且是产品的性能越来越强，复杂程度越来越高，更新步伐越来越快。支撑信息电子产品高速发展的基础就是微电子制造工艺水平的提高和电子产品设计开发技术的发展。前者以微细加工技术为代表，而后者的代表就是电子设计自动化（**electronic design automatic,EDA**）技术。

本设计采用的VHDL是一种全方位的硬件描述语言，具有极强的描述能力，能支持系统行为级、寄存器传输级和逻辑门级三个不同层次的设计；支持结构、数据流、行为三种描述形式的混合描述、覆盖面广、抽象能力强，因此在实际应用中越来越广泛。ASIC是专用的系统集成电路，是一种带有逻辑处理的加速处理器。而FPGA是特殊的ASIC芯片，与其他的ASIC芯片相比，它具有设计开发周期短、设计制造成本低、开发工具先进、标准产品无需测试、质量稳定以及可实时在线检测等优点。

在控制系统中，键盘是常用的人机交换接口，当所设置的功能键或数字键按下的时候，系统应该完成该键所设置的功能。因此，键信息输入是与软件结构密切相关的过程。根据键盘的结构不同，采用不同的编码方法。但无论有无编码以及采用什么样的编码，最后都要转换成为相应的键值，以实现按键功能程序的转移。

钟表的数字化给人们生产生活带来了极大的方便，而且大大地扩展了钟表原先的报时功能。诸如定时自动报警、定时启闭电路、定时开关烘箱、通断动力设备，甚至各种定时电气的自动启用等，所有这些，都是以钟表数字化为基础的。因此，研究数字钟及扩大其应用，有着非常现实的意义。

## 选题背景

本节将从FPGA嵌入式应用开发技术与数字钟技术发展的客观实际出发，通过对该技术发展状况的了解，以及课题本身的需要，指出研究基于FPGA的芯片系统与设计——数字钟的设计与实现的必要性。

### 课题相关技术的发展

当今电子产品正向功能多元化,体积最小化,功耗最低化的方向发展。它与传统的电子产品在设计上的显著区别是大量使用大规模可编程逻辑器件，使产品的性能提高，体积缩小，提高产品的自动化程度和竞争力，缩短研发周期。EDA技术正是为了适应现代电子技术的要求，吸收众多学科最新科技成果而形成的一门新技术。

美国ALTERA公司的可编程逻辑器件采用全新的结构和先进的技术，加上MaxplusII(或最新的QUARTUS)开发环境，更具有高性能，开

发周期短等特点，十分方便进行电子产品的开发和设计。

EDA 技术，技术以大规模可编程逻辑器件为设计载体，以硬件描述语言为系统逻辑描述主要表达方式，以计算机、大规模可编程逻辑器件的开发软件及实验开发系统为设计工具，通过有关的开发软件，自动完成用软件的方式设计的电子系统到硬件系统的逻辑编译，逻辑化简，逻辑分割，逻辑映射，编程下载等工作。最终形成集成电子系统或专用集成芯片的一门新技术。

本设计利用 VHDL 硬件描述语言结合可编程逻辑器件进行的，并通过数码管动态显示计时结果。数字钟可以由各种技术实现，它具有易学，方便，新颖，有趣，直观，设计与实验项目成功率高，理论与实践结合紧密，体积小，容量大，I/O 口丰富，易编程和加密等特点，并且它还具有开放的界面，丰富的设计库，模块化的工具以及 LPM 定制等优良性能，应用非常方便。因此，本设计采用可编程逻辑器件实现。

### 课题研究的必要性

现在是一个知识爆炸的新时代。新产品、新技术层出不穷，电子技术的发展更是日新月异。可以毫不夸张的说，电子技术的应用无处不在，电子技术正在不断地改变我们的生活，改变着我们的世界。在这快速发展的年代，时间对人们来说是越来越宝贵，在快节奏的生活时，人们往往忘记了时间，一旦遇到重要的事情而忘记了时间，这将会带来很大的损失。因此我们需要一个定时系统来提醒这些忙碌的人。数字化的钟表给人们带来了极大的方便。近些年，随着科技的发展和社会的进步，人们对数字钟的要求也越来越高，传统的时钟已不能满足人们的需求。多功能数字钟不管在性能还是在样式上都发生了质的变化，有电子闹钟、数字闹钟等等。

### 课题研究的内容

本设计主要研究基于 FPGA 的数字钟，要求时间以 24 小时为一个周期，显示年、月、日、时、分、秒。具有校时以及报时功能，可以对年、月、日、时、分及秒进行单独校对，使其校正到标准时间。校对时间由 4×4 矩形键盘进行控制，为了保证计时的稳定及准确须由晶体振荡器提

供时间基准信号。

本设计小组成员共有三人：其他两人分别采用原理图设计和 Verilog HDL 语言设计。

## 2 FPGA 简介

### FPGA 概述

FPGA是现场可编程门阵列（Field Programmable Gate Array）的简称，与之相应的CPLD是复杂可编程逻辑器件（Complex Programmable Logic Device）的简称，两者的功能基本相同，只是实现原理略有不同，所以有时可以忽略这两者的区别，统称为可编程逻辑器件或CPLD/PGFA。CPLD/PGFA几乎能完成任何数字器件的功能，上至高性能CPU，下至简单的74电路。它如同一张白纸或是一堆积木，工程师可以通过传统的原理图输入或硬件描述语言自由的设计一个数字系统。通



过软件仿真可以事先验证设计的正确性，在PCB完成以后，利用CPLD/FPGA的在线修改功能，随时修改设计而不必改动硬件电路。使用CPLA/FPGA开发数字电路，可以大大缩短设计时间，减少PCB面积，提高系统的可靠性。这些优点使得CPLA/FPGA技术在20世纪90年代以后得到飞速的发展，同时也大大推动了EDA软件和硬件描述语言HDL的进步。

## FPGA 基本结构

FPGA具有掩膜可编程门阵列的通用结构，它由逻辑功能块排成阵列，并由可编程的互连资源连接这些逻辑功能块来实现不同的设计。

FPGA一般由3种可编程电路和一个用于存放编程数据的静态存储器SRAM组成。这3种可编程电路是：可编程逻辑模块（CLB--Configurable Logic Block）、输入/输出模块（IOB--I/O Block）和互连资源（IR—Interconnect Resource）。可编程逻辑模块CLB是实现逻辑功能的基本单元，它们通常规则的排列成一个阵列，散布于整个芯片；可编程输入/输出模块（IOB）主要完成芯片上的逻辑与外部封装脚的接口，它通常排列在芯片的四周；可编程互连资源包括各种长度的连接线段和一些可编程连接开关，它们将各个CLB之间或CLB、IOB之间以及IOB之间连接起来，构成特定功能的电路。

。图2-1是CLB基本结构框图，它主要由逻辑函数发生器、触发器、数据选择器等电路组成。CLB中3个逻辑函数发生器分别是G、F和H，相应的输出是G'、F'和H'。G有4个输入变量G1、G2、G3和G4；F也有4个输入变量F1、F2、F3和F4。这两个函数发生器是完全独立的，均可以实现4输入变量的任意组合逻辑函数。逻辑函数发生器H有3个输入信号；前两个是函数发生器的输出G'和F'，而另一个输入信号是来自信号变换电路的输出H1。这个函数发生器能实现3输入变量的各种组合函数。这3个函数发生器结合起来，可实现多达9变量的逻辑函数。

CLB中有许多不同规格的数据选择器（四选一、二选一等），通过对CLB内部数据选择器的编程，逻辑函数发生器G、F和H的输出可以连接到CLB输出端X或Y，并用来选择触发器的激励输入信号、时钟有效边沿、时钟使能信号以及输出信号。这些数据选择器的地址控制信号均

CLB中的逻辑函数发生器F和G均为查找表结构，其工作原理类似于ROM。F和G的输入等效于ROM的地址码，通过查找ROM中的地址表可以得到相应的组合逻辑函数输出。另一方面，逻辑函数发生器F和G还可以作为器件内高速RAM或小的可读写存储器使用，它由信号变换电路控制。

。IOB提供了器件引脚和内部逻辑阵列之间的连接。它主要由输入触发器、输入缓冲器和输出触发/锁存器、输出缓冲器组成。

每个IOB控制一个引脚，它们可被配置为输入、输出或双向I/O功能。当IOB控制的引脚被定义为输入时，通过该引脚的输入信号先送入输入缓冲器。缓冲器的输出分成两路：一路可以直接送到MUX，另一路经延

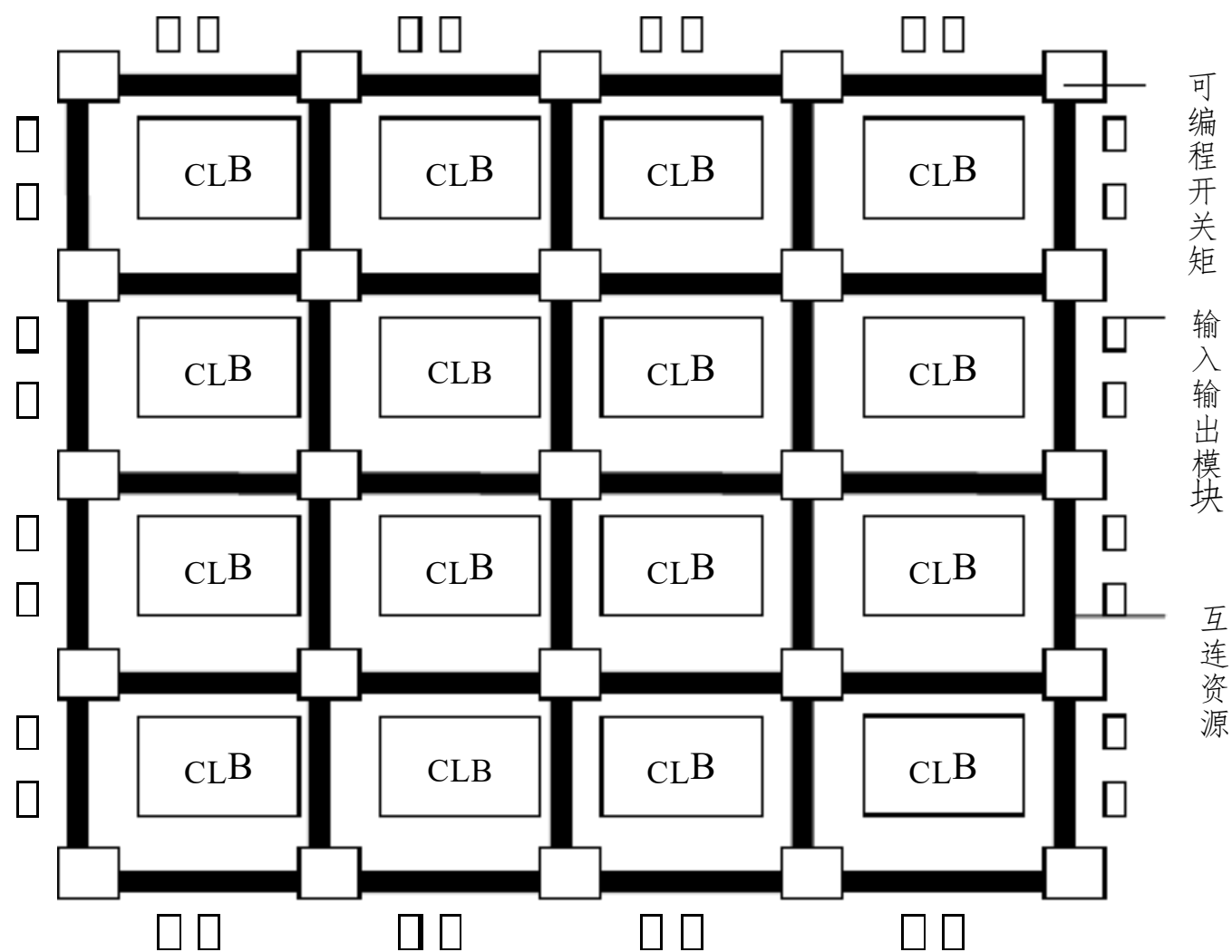


图2-1 CLB基本结构

时几纳秒（或者不延时）送到输入通路D触发器，再送到数据选择器。通过编程给数据选择器不同的控制信息，确定送至CLB阵列的I1和I2是

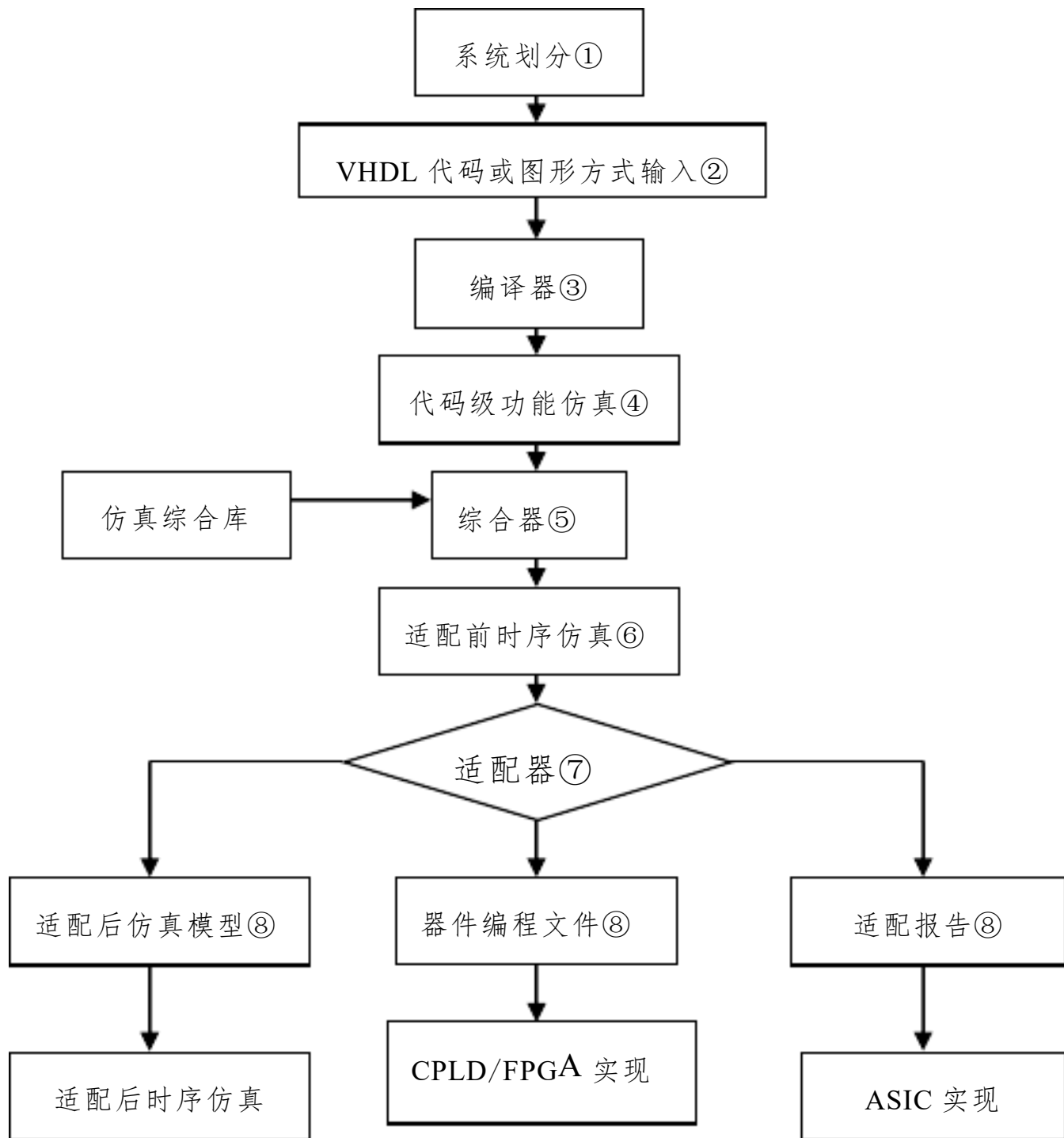
当IOB控制的引脚被定义为输出时，CLB阵列的输出信号OUT也可以有两条传输途径：一条是直接经MUX送至输出缓冲器，另一条是先存入输出通路D触发器，再送至输出缓冲器。

IOB输出端配有两只MOS管，它们的栅极均可编程，使MOS管导通或截止，分别经上拉电阻接通Vcc、地线或者不接通，用以改善输出波形和负载能力。

。可编程互连资源IR可以将FPGA内部的CLB和CLB之间、CLB和IOB之间连接起来，构成各种具有复杂功能的系统。IR主要由许多金属线段构成，这些金属线段带有可编程开关，通过自动布线实现各种电路的连接。

## FPGA

一般说来，一个比较大的完整的项目应该采用层次化的描述方法：分为几个较大的模块，定义好各功能模块之间的接口，然后各个模块再细分去具体实现，这就是TOP DOWN（自顶向下）的设计方法。目前这种高层次的设计方法已被广泛采用。高层次设计只是定义系统的行为特征，可以不涉及实现工艺，因此还可以在厂家综合库的支持下，利用综合优化工具将高层次描述转换成针对某种工艺优化的网络表，使工艺转化变得轻而易举。CPLD/FPGA系统设计的工作流程如图2-2所示。



2-2 CPLD/FPGA 系统设计流程

流程说明：

“自顶向下”的设计方法进行系统划分。

，这是设计中最普遍的输入方式。此外，还可以采用图形输入方式（框图、状态图等），这种输入方式具有直观、容易理解的优点。

。

，主要是检验系统功能设计的正确性。这一步骤适用于大型设计，因为对于大型设计来说，在综合前对源代码仿真，就可以大大减少设计重复的次数和时间。一般情况下，这一仿真步骤可略去。

，生成门级描述的网络表文件，这是将高层次描述转化为硬件电路的关键步骤。综合优化是针对ASIC芯片供应商的某一产品系列进行的，

，仿真过程不涉及具体器件的硬件特性，是较为粗略的。一般的设计，也可略去这一步骤。

，包括底层器件配置、逻辑分割、逻辑优化和布局布线。

，产生多项设计结果：（a）适配报告，包括芯片内部资源利用情况，设计的布尔方程描述情况等；（b）适配后的仿真模型；（c）器件编程文件。根据适配后的仿真模型，可以进行适配后时序仿真，因为已经得到器件的实际硬件特性（如时延特性），所以仿真结果能比较精确的预期未来芯片的实际性能。如果仿真结果达不到设计要求，就修改VHDL源代码或选择不同速度和品质的器件，直至满足设计要求。

最后将适配器产生的器件编程文件通过编程器或下载电缆载入到目标芯片CPLD/FPGA中。

## FPGA

硬件设计需要根据各种性能指标、成本、开发周期等因素，确定最佳的实现方案，画出系统框图，选择芯片，设计PCB并最终形成样机。

CPLD/FPGA软件设计可分为两大块：编程语言和编程工具。编程语言主要有VHDL和Verilog两种硬件描述语言；编程工具主要是两大厂家Altera和Xilinx的集成综合EDA软件（如MAX+plusII、QuartusII、Foundation、ISE）以及第三方工具（如FPGA Express、Modelsim、Synposys SVS等）。具体的设计输入方式有以下几种：

。HDL既可以描述底层设计，也可以描述顶层的设计，但它不容易做到较高的工作速度和芯片利用率。用这种方式描述的项目最后所能达到的性能与设计人员的水平、经验以及综合软件有很大的关系。

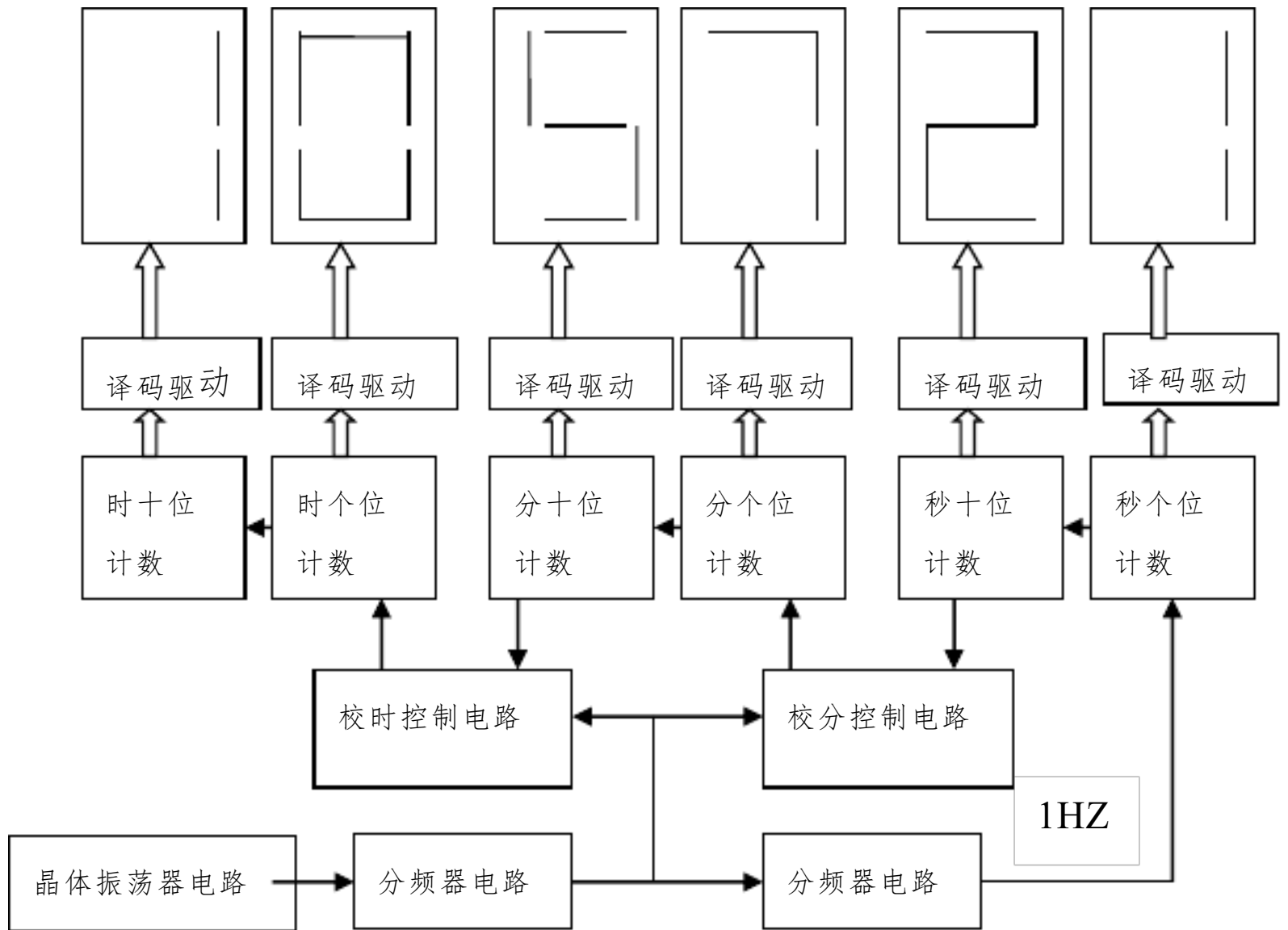
。可以分为电路原理图描述，状态机描述和波形描述3种形式。有的软件3种输入方法都支持，如Active-HDL。MAX+plusII 图形输入方式只支持电路原理图描述和波形描述两种。电路原理图方式描述比较直观和高效，对综合软件的要求不高。一般大都使用成熟的IP核和中小规模集成电路所搭成的现成电路，整体放到一片可编程逻辑器件的内部去，所以硬件工作速度和芯片利用率很高，但是但项目很大的时候，该方法就显得有些繁琐；状态机描述主要用来设计基于状态机思想的时序电

条件以及相应的输入输出，最后生成 语言描述，送去综合软件综合到可编程逻辑器件的内部。由于状态机到HDL语言有一种标准的对应描述方式，所以这种输入方式最后所能达到的工作速度和芯片利用率主要取决于综合软件；波形描述方式是基于真值表的一种图形输入方式，直接描述输入与输出的波形关系。这种输入方式最后所能达到的工作速度和芯片利用率也是主要取决于综合软件。

### 3

#### 数字钟的构成

数字钟实际上是一个对标准频率（1HZ）进行计数的计数电路。由于计数的起始时间不可能与标准时间（如北京时间）一致，故需要在电路上加一个校时电路，同时标准的 1HZ 时间信号必须做到准确稳定。通常使用石英晶体振荡器电路构成数字钟。图 3-1 所示为数字钟的一般构成框图。主要包括时间基准电路、计数器电路、控制电路、译码和显示电路。其中的控制逻辑电路是比较灵活多样的，不断完善它可以增强数字钟的功能。



3-1 数字钟的一般组成框图

本设计在上面数字钟结构的基础上还加入了显示年、月、日的功能，其显示范围为2000年1月1日至2099年12月31日。其总体框架如图3-2。

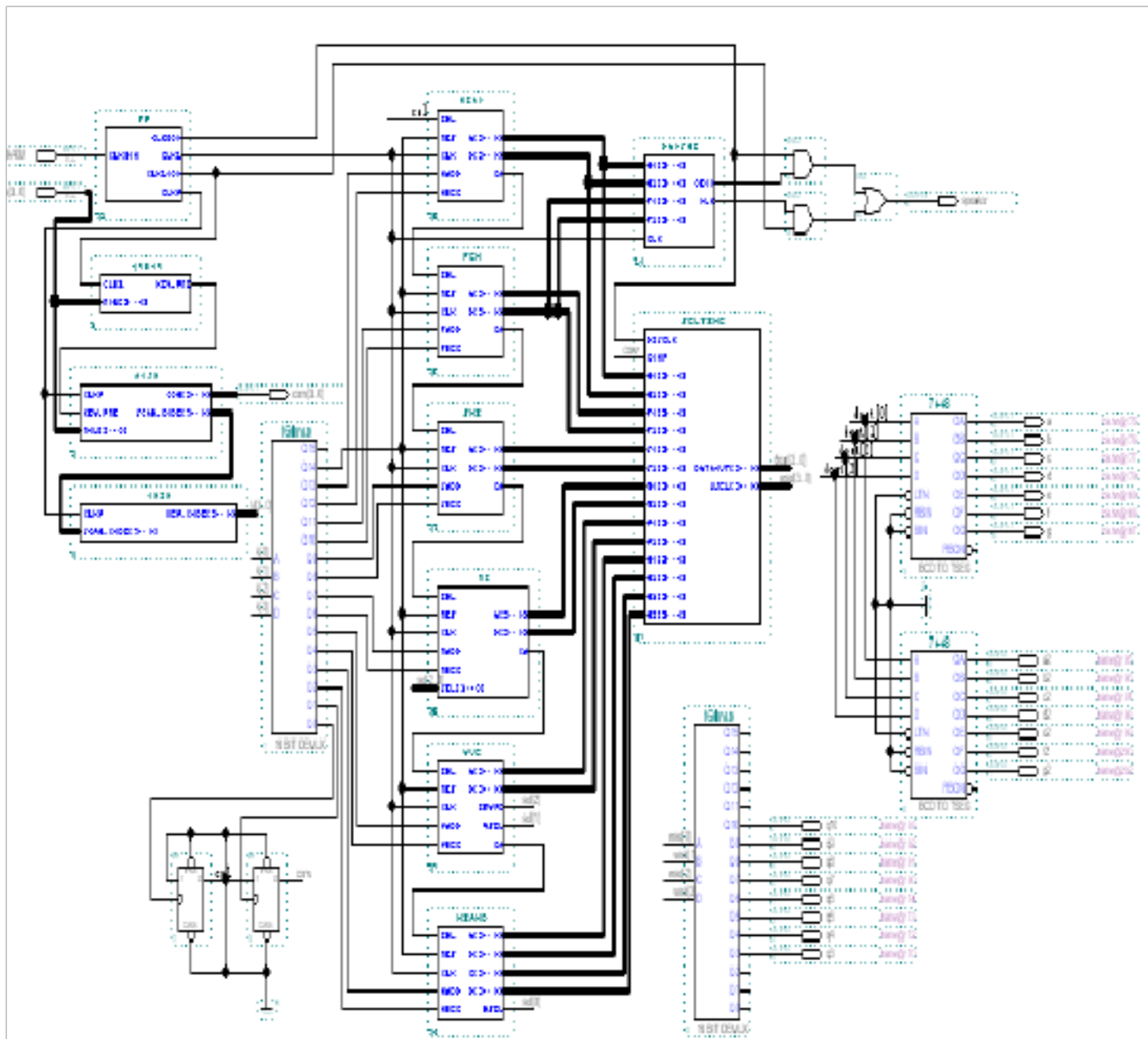


图3-2 总体框架

## 数字钟的工作原理

振荡器产生稳定的高频脉冲信号，作为数字钟的时间基准，然后经过分频器输出标准秒脉冲。秒计数器满60后向分计数器进位，分计数器满60后向小时计数器进位，小时计数器按照“24翻1”规律计数。计满后各计数器清零，重新计数。日期部分由于日有28天、29天、30天、31天4种情况，故日由年和月共同判断其天数，日计满后向月进位，月满后向年进位。计数器的输出分别经译码器送数码管显示。计时出现误差时，可以用校时电路校时、校分、校秒和校年、校月和校日。在控制信号中除了一般的校时信号外，还有时钟使能信号、时钟清零信号。控制信号由4×4矩形键盘输入。时基电路可以由石英晶体振荡电路构成，如果晶振频率为1MHz，经过6次十分频就可以得到秒脉冲信号。译码显示



电路由七段译码器完成，显示由数码管构成。

## 4 单元电路设计

### 分频模块电路设计与实现

晶体振荡器是构成数字式时钟的核心，振荡器的稳定度及频率的精度决定了数字钟计时的准确程度，它保证了时钟的走时准确及稳定。

石英晶体的选频特性非常好，只有某一频率点的信号可以通过它，其它频率段的信号均会被它所衰减，而且，振荡信号的频率与振荡电路中的R、C元件的数值无关。因此，这种振荡电路输出的是准确度极高的信号。然后再利用分频电路，将其输出信号转变为秒信号，其组成框图如图4-1。

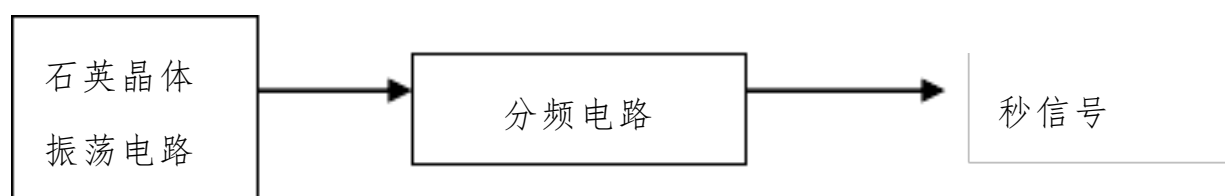


图 4-1 秒信号产生电路框图

本系统使用的晶体振荡器电路给数字钟提供一个频率稳定准确的40MHz的方波信号，其输出至分频电路。分频电路的逻辑框图如图4-2所示。

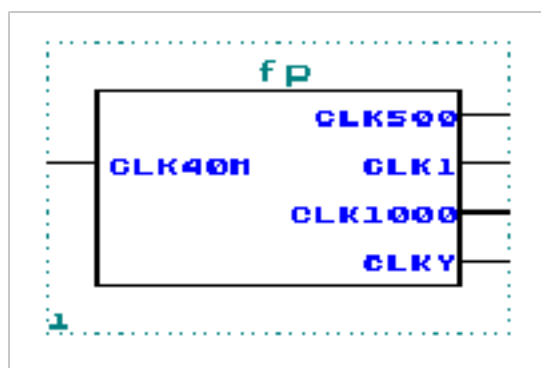


图 4-2 分频电路模块

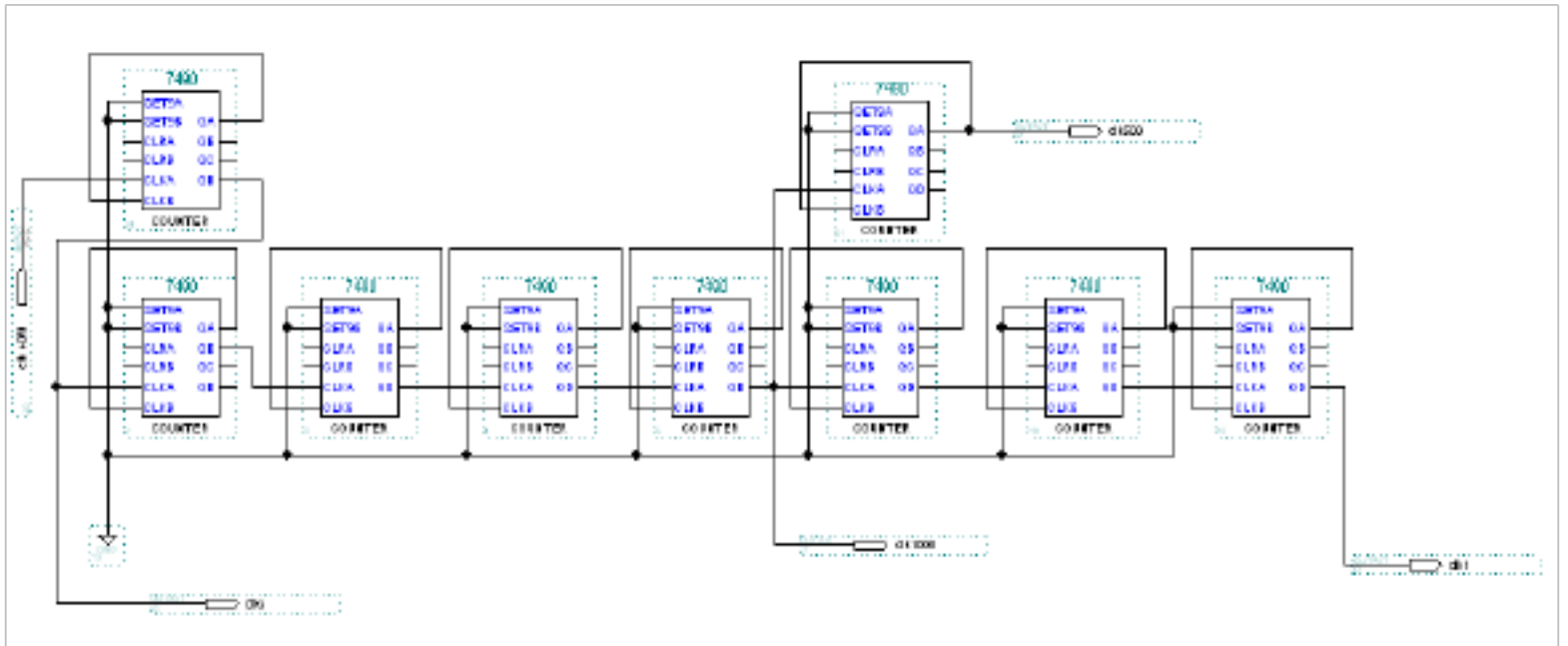


图 4-3 分频模块电路设计

其内部结构图见图 4-3。电路中采用 Max+plusII 元器件库中的计数器 7490 进行硬件分频。经分频后输出 1HZ 的标准秒信号 CLK1、4MHZ 的按键扫描信号、1KHZ 的按键去抖信号和 500HZ 用于报时模块的输入信号。该模块的时序仿真图如图 4-4 所示，满足设计要求。

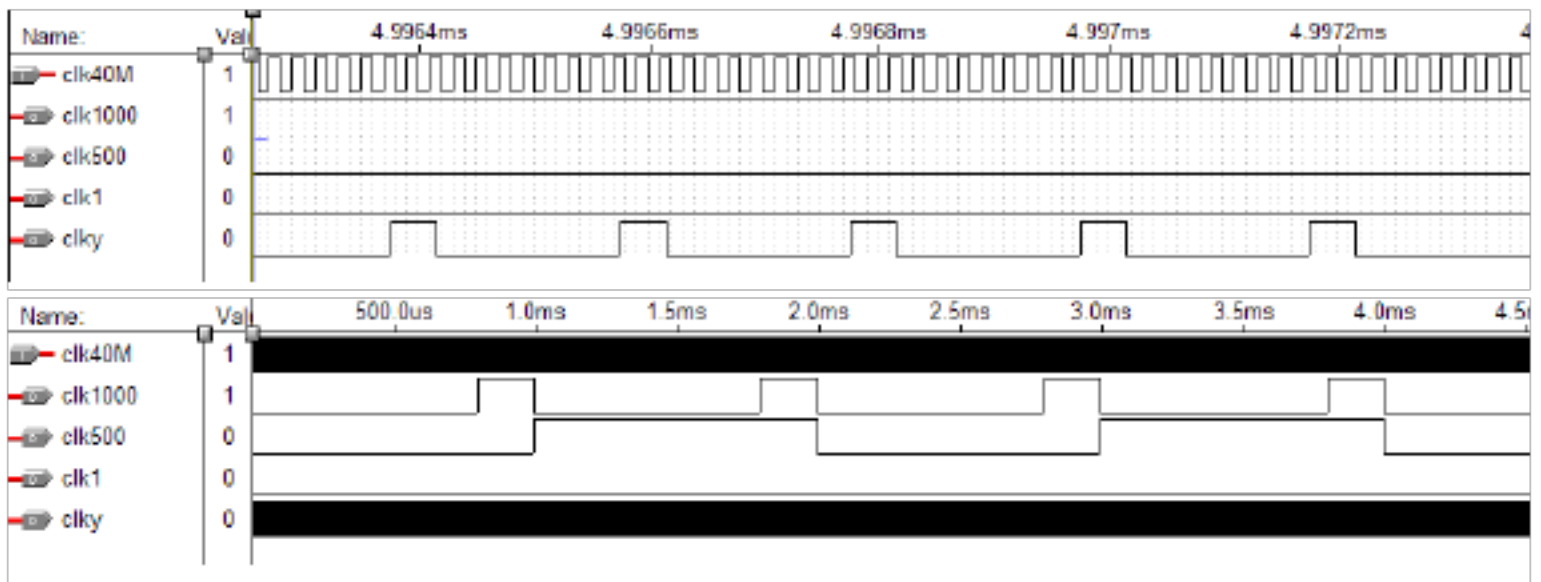


图 4-4 分频模块仿真图

## 校时控制模块电路设计与实现

### 键盘接口电路原理

校时控制模块在本系统中也就是键盘接口电路部分。下面先介绍键盘接口电路的工作原理，如图 4-5。本系统采用的就是这种行列式键盘接口，相对个按键的键盘接口来说节省了 I/O 接口。如图所示，行线通

过一个电阻被上拉到+5V 电压。行线与按键的一个引脚相连，列线与按键的另一个引脚相连。平时列线被置成低电平，没有按键被按下的时候，行线保持高电平，而有按键被按下的时候，行线被拉成低电平，这时候控制器就知道有按键被按下，但只能判断出在哪一行，不能判断出在哪一列，因此接下来就要进行键盘扫描，以确定具体是哪个按键被按下。键盘扫描的过程是将列线逐列置成低电平，然后读取行线状态，直到行线中出现低电平，可知这时哪一列是低电平，然后将行线与列线的状态装入键码寄存器，进行按键译码，得到按下的按键的相应编码，这样就完成了按键扫描的过程。当然，一个完整的按键扫描过程还需要配合相应的键盘去抖手段才能正确的识别按键，不会发生重键和错误判断等情况。

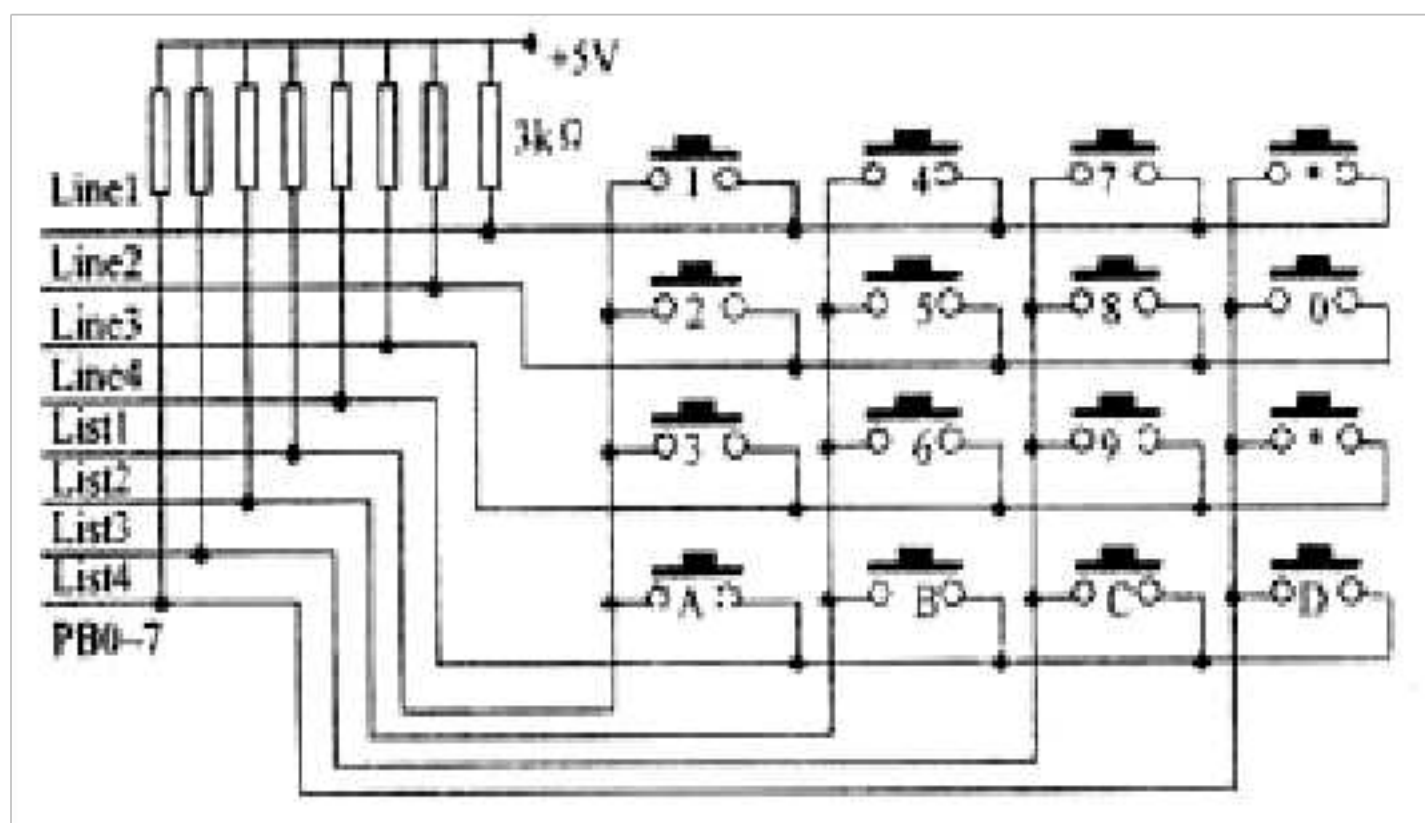


图 4-5 键盘接口电路

### 键盘接口的 VHDL 描述

本模块用于当有按键按下时，采用软件消抖的办法去除按键抖动。模块的实现方法是先判断是否有按键按下，如有按键按下则延时一段时间，待抖动过去之后再读行线状态，如果仍有低电平行线，则确定有按

键按下，然后产生一个有按键按下的信号。该模块有一个时钟输入端口，输入时钟信号是分频出来的 1KHZ 的时钟；有一个输入端口与行线相连，用于输入行线状态；一个输出端口，用于输出有按键按下的信号。该模块的逻辑框图如图 4-6 所示。

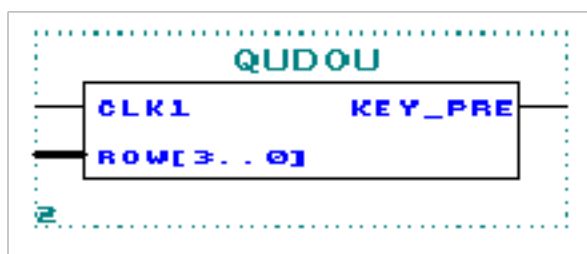


图 4-6 去抖逻辑框图

该电路的 VHDL 程序如下：

```

library ieee;
use ;
use ;
use ;
entity qudou is
port(clk1:in std_logic;
      row:in std_logic_vector(3 downto 0);
      key_pre:out std_logic);
end qudou;
architecture behav of qudou is
    signal sig1,counter:std_logic_vector(3 downto 0);
    signal tmp1,sig2:std_logic;
begin
    sig1<=row;
    tmp1<=sig1(0)and sig1(1)and sig1(2)and sig1(3);
    key_pre<=counter(0)and counter(1)and counter(2)and counter(3);
process(clk1)
begin
if(clk1'event and clk1='1')then
    if(tmp1='0')then
        if(sig2='0')then

```

```

        sig2<='1';
    end if;
end if;
if(sig2='1')then

    sig2<='0';

    else
        counter<=counter+'1';
    end if;
end if;
end if;
end process;
end behav;

```

程序说明：

这段程序是实现按键消抖，在这里实现的比较简单，原理是当有按键按下的时候，temp1 会变成低电平，这时把 sig2 变成高电平，如果此时 counter 不为“1111”时，内部计数器计数，从“0000”直到“1111”，当计数到“1111”时，key\_pre 输出高电平，同时计数器清零。由于计数脉冲为 1KHZ，故从有按键按下到输入信号产生大概需要 15ms。

如果有按键抖动的话，tem1 会在“0”和“1”之间变动多次，但是计数过程不会停止，原因是计数由 sig2 的电平决定，而 sig2 一旦变成高电平，再要变成低电平，需要计数完成即 counter 等于“1111”时。所以计数过程不会受抖动影响。一旦计数完成，抖动已经过去，不会发生重键现象了，这样就去除了抖动。

键扫描模块的框图如图 4-7 所示。

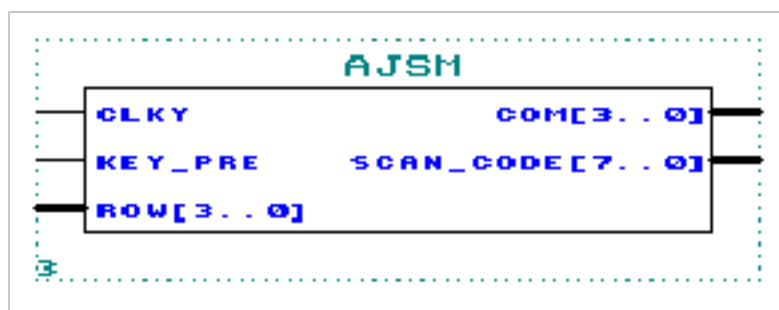


图 4-7 按键扫描模块

CLKY 为输入时钟，该时钟是分频模块分出的 4MHZ 的时钟信号，之所以在这里采用频率高的时钟信号就是因为键扫描是一个快过程，不需要太慢的时钟。Key\_pre 是由去抖模块输出的有键按下的信号，这个信号引发按键扫描模块内部信号的变化，在该状态信号的作用下，模块可以键盘扫描。

ROW[3..0]是行输入信号，与键盘的行线相连，COM[3..0]是列输出信号，与键盘的列线相连。

SCAN\_CODE[7..0]是扫描的键码输出端口。

键扫描的基本方法是将列线逐一置成低电平，然后读行线输入端口，如果行线中有低电平出现，说明按键已经确定，将行向量和列向量读入键码中即可。键盘扫描程序如下：

```
library ieee;
use ;
use ;
use ;
entity ajasm is
port(clky,key_pre:in std_logic;
      row:in std_logic_vector(3 downto 0);
      com:out std_logic_vector(3 downto 0);
      scan_code:out std_logic_vector(7 downto 0));
end ajasm;
architecture behav of ajasm is
signal sig_com:std_logic_vector(3 downto 0);
signal counter:std_logic_vector(1 downto 0);
signal tmp,sig1,sig2:std_logic;
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/288062077101006035>