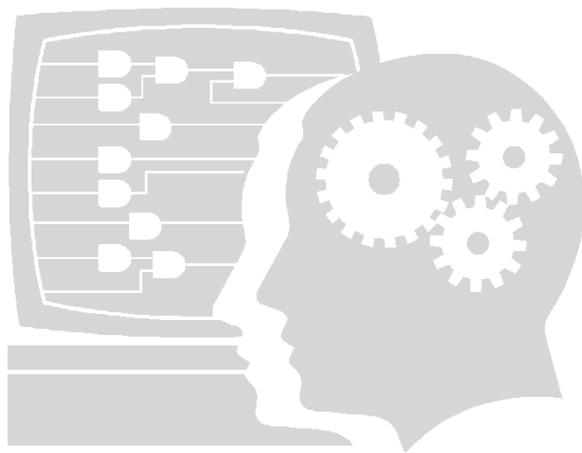


KX-CDS

# 综合实验开发系统 说明书及实验讲义

第二版



修改日期：2019年8月

杭州康芯电子有限公司

[www.kx-soc.com](http://www.kx-soc.com)

# 目 录

前言.....	6
第一章 KX-CDS 创新设计综合实验开发系统.....	8
1.1 KX-CDS5 系统的主要功能与结构特色.....	8-9
1.2 配套教材.....	10-11
第二章 KX-CDS 主系统平台.....	12
2.1 主系统介绍.....	13
6. 实验电路结构特点与实用范围简述.....	14-17
2.2 FPGA 核心板扩展至康芯主系统引脚对照表（模式类）.....	18-19
2.3 FPGA 核心板及扩展板介绍.....	19
2.3.1 核心板 FPGA 板模块.....	20
2.3.2 可重构型 DDS 全数字函数信号发生器.....	21
2.3.3 单片机模块.....	22
2.3.4 4X4 十六键键盘.....	22
2.3.5 4X4+8 综合键盘模块.....	22
2.3.6 交通灯显示模块.....	23
2.3.7 八位动态扫描模块.....	23
2.2.8 点阵式 128X64 液晶显示模块.....	23
2.3.9 4 行 X20 字字符型液晶显示模块.....	23
2.3.10 800X480 数字 TFT 彩屏模块.....	24
2.3.11 普通 DAC 和 ADC 标准模块.....	24
2.3.12 高速 A/D 和双通道 DA 模块.....	25
2.3.13 高速 12 位 SPI 串行双 ADC.....	25
2.3.14 8 位+16 位高分辨率 ADC 等模块.....	25
2.3.15 SPI 串行接口高速 ADC+DAC 模块.....	26
2.3.16 USB 接口模块.....	26
2.3.17 电机接口模块.....	26
2.3.18 双串行存储器/逻辑笔设计模块.....	26
2.3.19 SRAM/EPPROM 模块.....	26
2.3.20 继电器/CAN/RS485 总线模块.....	27
2.3.21 GPS 实验开发模块.....	27
2.3.22 看门狗定时器+时钟日历模块.....	27
2.3.23 无线编码收发+数字温度传感器模块.....	27
2.3.24 摄像头模块.....	27
2.3.25 传统 74 系列实验模块.....	28
2.3.26 基于 5200 控制的网口模块.....	28
2.3.27 WIFI 无线网卡+超声波.....	28
2.3.28 8X16/16X16 点阵模块.....	28
2.3.29 高速 12 位 AD/DA 模块.....	29
2.3.30 语音处理模块.....	29
2.3.31 SD+PS/2+RS232+VGA 模块.....	29
2.3.32 无需配置 HDMI 输入模块（新）.....	29
2.3.33 无需配置 HDMI 输输出模块（新）.....	29
2.4 FPGA 核心板扩展至康芯主系统引脚对照表(扩展模块类).....	30-31
第三章 软硬件使用流程.....	32
3.1 十六进制计时器使用流程.....	32
3.1.1 建立工作库文件夹和编辑文件.....	32
3.1.2 创建工程.....	33-34

3.1.3 约束项目设置.....	35
3.1.4 全程综合与编译.....	36
3.1.5 RTL 图观察器应用.....	37
3.2 时序仿真.....	38
3.3 硬件测试.....	39
3.3.1 引脚锁定.....	40
3.3.2 编译文件下载.....	41
3.3.3 通过 AS 口对配置芯片直接编程.....	42
3.3.4 通过 JTAG 口对配置芯片间接编程.....	43
第四章 EDA 基础及综合实验.....	43
实验 4-1 十六进制计数器.....	43-44
实验 4-2 七段译码器.....	45
实验 4-3 计数器+七段译码器.....	46
实验 4-4 多路选择器设计.....	46
实验 4-5 逻辑门设计.....	47
实验 4-6 触发器实验.....	48
实验 4-7 八位加法器设计.....	49
实验 4-8 数控分频器设计.....	50
实验 4-9 原理图 8 位全加器设计.....	51
实验 4-10 移位运算器设计.....	52
实验 4-11 序列检测器设计.....	53
实验 4-12 16 进制频率计设计.....	54-55
实验 4-13 10 进制频率计设计.....	56
实验 4-14 交通灯设计.....	56
实验 4-15 抢答器设计.....	57
实验 4-16 出租车计费器设计.....	58-59
实验 4-17 数字钟设计.....	60-61
实验 4-18 硬件电子琴设计.....	62
实验 4-19 乐曲硬件演奏电路设计.....	63-64
实验 4-20 五首乐曲演奏电路设计.....	65-66
实验 4-21 乒乓球游戏电路设计.....	67
实验 4-22 八位移位相加硬件乘法器设计.....	68-69
实验 4-23 八位动态扫描设计.....	70-72
实验 4-24 4X4 阵列键盘键信号设计.....	73-74
实验 4-25 硬件消抖动电路设计.....	75
实验 4-26 ADC 采样控制电路设计.....	76
实验 4-27 直流电机综合测控系统设计.....	77-79
实验 4-28 步进电机控制电路设计.....	80
实验 4-29 步进电机细分控制电路设计.....	81-82
实验 4-30 8*16/16*16 点阵电路设计.....	83-84
实验 4-31 数字彩屏显示控制电路.....	85
实验 4-32 模块化交通灯设计.....	86
实验 4-33 正弦信号发生器设计.....	87-88
实验 4-34 DDS 正弦信号发生器设计.....	89
实验 4-35 移相信号发生器设计.....	89-90
实验 4-36 高速 ADC_DAC 存储示波器设计.....	91
实验 4-37 PS2 键盘控制模型电子琴电路设计.....	92-93
实验 4-38 VGA 显示实验.....	94-95
实验 4-39 基于 FT245BM 的 USB 通信控制模块设计.....	96
实验 4-40 五功能智能逻辑笔设计.....	97-98
实验 4-41 无线编码收发电路设计.....	99

实验 4-42 SDRAM 测试.....	100
实验 4-43 串口收发实验.....	100
实验 4-44 DS1302 数码管显示 RTC 时间实验.....	100
实验 4-45 SD 卡读写实验.....	100
实验 4-46 SD 卡音乐播放例程.....	100
实验 4-47 SD 卡读取 BMP 图片显示.....	100
实验 4-48 RS485 通信实验.....	100
实验 4-49 18B20 温度测试.....	100
实验 4-50 HDMI 输入输出测试.....	100
第五章 单片机系统综合实验.....	101
5.1 单片机基本实验.....	101
实验 5-1. 存储器块清零程序设计.....	101
实验 5-2 二进制到 BCD 转换程序设计.....	101
实验 5-3 十六进制到 ASCII 码转换程序设计.....	102
实验 5-4 存储块移动程序设计.....	103
实验 5-5 多分支程序.....	104
实验 5-6 数据排序程序设计.....	105
实验 5-7 P1 口输入、输出实验.....	106
实验 5-8 交通灯控制（软件延时法）.....	107
实验 5-9 交通灯控制（定时器延时法）.....	108
实验 5-10 计数器应用实验.....	109
实验 5-11 外部中断实验.....	109
实验 5-12 定时器实验 1（P1 口状态取反）.....	109
实验 5-13 定时器输出 PWM 实验.....	110
实验 5-14 外部中断实验.....	111
5.2 单片机扩展和接口实验与设计.....	112
实验 5-15 单片机串口扩展.....	112
实验 5-16 键盘与液晶显示控制.....	112
实验 5-17 单片机串行通信和红外双向通信.....	112-113
实验 5-18 单片机扩展 X5045 看门狗器件.....	114
实验 5-19 单片机扩展 DS1302 时钟/日历器件.....	115
实验 5-20 SPI 串行 DAC TLV5637 与单片机的接口.....	116
实验 5-21 串行精密 ADC 器件 ADS1100 与单片机的接口.....	117
实验 5-22 串行高速 ADC 器件 ADS7816 与单片机的接口.....	118
实验 5-23 高速低功耗串行 ADC 器件 TLV2541 与单片机的接口.....	118
实验 5-24 双通道 A/D 转换芯片 ADC0832 与单片机的接口.....	119
实验 5-25 高速同步 10 位串行 A/D 转换器与单片机的接口.....	119
第六章 基于单片机 IP 核的 FPGA 片上系统 SOC 设计.....	120
实验 6-1. FPGA 片上系统 8051 核测试电路.....	120
实验 6-2. 串进并出\并进串出 8051 核测试电路.....	121-122
实验 6-3. 超声波 8051 核测距设计.....	123
实验 6-4. 扩展存储器的 FPGA 单片系统设计.....	124
实验 6-5. 直流电机测控 FPGA 单片系统设计.....	125-126
实验 6-6. 等精度频率计 FPGA 单片系统设计.....	127
实验 6-7. 8051 控制 WIFI 和实验系统的实验.....	128
实验 6-8. 8051 控制网口和实验系统的实验.....	128
实验 6-9. 8051 控制 GPS 的实验.....	128
实验 6-10. 8051 控制 AD0809 实验.....	128
实验 6-11. 8051 控制点阵液晶实验.....	128
实验 6-12. 8051 控制 DDS 移相信号发生器实.....	128

实验 6-13. 8051控制李萨如图输出实验.....	128
实验 6-14. 8051控制占空比可调方波输出实验.....	128
实验 6-15. 805118B20数字温度测控实验.....	128
MIF 文件生成器使用方法.....	129
STC89C 单片机编程方法.....	130-131

# 前 言

## ● 设备型号说明:

◆ 此讲义是针对康芯 KX-4C/10C 系列及友晶 DE 系列设计的核心板加装实验开发系统,可作为 EDA、组成原理、微机原理、单片机实验系统讲义,主要区分型号标准是,DE 系列分别是 DE0/DE0-CV/DE1-SOC/康芯的 KX-4C6/10/55/10CL55,其他扩展模块根据用户需求可任意配置。

## ● 每套设备基本配件

◆ DE 系列可根据友晶提供的光盘来查看,分别是电源、USB 下载线等;

◆ 康芯提供, ● 电源线一根, ● RS232 串口一根, ● 十芯线数根(根据用户模块数量配置) ● 十四芯线一根,单根线数根。如康芯设计核心板,如核心板是康芯生产,提供双功能 USB 接口下载器一台及 USB 线一根。

◆ 模块板根据供货清单。

◆ 光盘内容说明:(无论哪个型号,光盘的文件夹名都是统一)

● 友晶系列的 DE0 提供光盘,DE0-CV/DE1-SOC 根据板上提供的网址下载资料,或康芯提供。

● 基本软件:QUARTUSIIX,NIOSIIX。

● 光盘文件夹:“友晶原版资料”“DE0/DE0-CV/DE1-SOC”是友晶提供的资料,KX 是康芯提供的资料。

● “DEMO”是康芯提供的例程,1、“DEMO/EDA\_DEMO”是 EDA 实验项目,其中“LED\_DEMO”是主系统关于模式控制的实验“MODU\_DEMO”是主系统和扩展模块的实验;2“DEMO/NIOS\_SOC”是 NIOSII 调试实验项目、3、“DEMO/8051Core\_DEMO”是 51 核实验项目,“COMT\_DEMO”是现代计算机组成原理实验。大多数实验给出源码,并用图片 PPT 或 PDF 形式给出实验操作方法;“MCU\_TO\_FPGA”是 FPGA 和单片机通信实验,此实验项目无论哪个型号都是给的 3C10 板和单片机的实验及说明,用户根据提供的实验说明及源码可自行修改。

● “FPGA\_单片机\_编程”包括 USB 下载器功能、FPGA 的编程方法及单片机编程方法等。

● “MCU\_Tech\_DEMO”是单片机对串行 A/D,D/A,等开发板的实验,具体在“MCU 控制 ADC\_DAC\_等器件实验指导”提供说明。

● “VIVI\_FILE”是在实验中用到的通信上位机及工具软件、液晶资料、开发板专用芯片手册、USB 驱动软件等。

● “重要文件”包括实验系统说明、DDS 模块使用方法开发板用到的专用芯片使用手册等。

● “原理图”是此系统的模块的原理图,用户可作参考。

## 声 明

凡本公司提供的实验源程序及资料仅用作教学研究之用,不得用于商业开发,否则将追究法律责任。

## 版本历史

版本	修改日期	修改内容
第二版	2019-8	修改

## 使用注意事项：

1. 为了安全实验 **KX-CDS** 系列实验设备，请您在使用前仔细阅读此讲义使用说明，充分了解本讲义内容，并保管好此讲义，以备随时查阅。
2. 设备不用时，请关闭电源，并把箱盖盖好，防止灰尘落入线路板上，导致使用故障。
3. 实验设备上的裸露的金属针、孔等，不要随意用手触摸，以免静电导致芯片烧坏和污迹粘上氧化。
4. 实验设备上的器件不要随意卸装，以免装配失误导致设备故障；
5. 下载器十芯口连接核心板的 **JTAG** 口，不要随意插拔，特别是核心板的 **JTAG** 口金属针不要随意用手触摸，以免静电致 **JTAG** 口损坏。
6. 实验设备箱体部分，不要随意卸装，以免操作失误导致人身安全及设备损坏。
7. 设备在运输过程中，不要剧烈晃动。
8. 设备发生故障、异常或停电时，请立即切断本机电源。
9. 为防止触电、漏电、保持桌面清洁干燥，要接地线。

# 第一章 KX-CDS 创新设计综合实验开发系统

KX-CDS5 系统的组成部分包括：(1)主系统板；(2)基于大规模 FPGA 核心板；(3) Multi-task Reconfiguration (多功能重配置结构) 实验控制系统；(4) 包括液晶等各类显示模块；(5) 可自由组合的模块化综合创新实验系统；(6) 可二次开发型全数字 DDS 信号发生器；(7) 适应教学实验与开发目标的不同微处理器 IP 核。

## 1.1 KX-CDS 系统的主要功能与结构特色：

### 一、学科课程适用面宽。

具体课程包括（根据不同的核心板）：(1) 面向现代数字系统的数字电路基础课；(2) EDA 技术（包括硬件语言）；(3) SOPC/SOC 技术；(4) 现代计算机组成原理；(5) 基于片上系统的单片机技术；(6) 微机原理与接口技术（包括基于 FPGA 的 SOC 技术）；(7) ARM 开发与应用技术（基于嵌入 FPGA 的 ARM 硬核）；(8) 现代 DSP 技术（基于 FPGA 中的各类 IP 核构建的 DSP 系统）。

配套教材主要有：1. 科学出版社的《EDA 技术实用教程-Verilog 版》第 6 版；2. 科学出版社的《EDA 技术实用教程-VHDL 版》第 5 版；3. 清华大学出版社的《EDA 技术与 VHDL》第 5 版；4. 《EDA 技术与 Verilog HDL》第 3 版，其中详细介绍了硬件描述语言，QuartusII、DSP-Builder 应用开发等；5. 清华大学出版社的《单片机原理与应用技术》。介绍了 FPGA 与单片机接口技术及 8051 片上系统 SOC 应用；6. 科学出版社的《数字电子技术基础》。基于全新的数字电子技术教学理念；7. 科学出版社的《现代计算机组成原理——结构，原理，设计与 SOC 实现》。

### 二、实验开发内容完整全面。

对于配置不同的 FPGA 核心板，可完成各课程从基础知识实验到综合性自主创新开发型实验，乃至科研创新项目。

### 三、多功能重配置型高效率实验控制电路。

KX-CDS5S 主系统板上含有 Multi-task Reconfiguration（多功能重配置结构）控制电路。该电路结构能仅通过一个键的控制，实现纯电子方式切换，选择十余种面向不同实验需要的针对 FPGA 目标芯片的硬件电路连接结构，并且毫不影响系统工作速度，大大提高了实验系统的连线灵活性，免除了传统情况下由于大量实验连接线导致的低效率，电路低可靠性，以及实验目标系统的低速性。

通常，传统的手工插线方式虽然灵活，由于插线长、多、乱，会严重影响系统速度、系统可靠性和电磁兼容性能，不适合以高速见长的 FPGA/SOPC 等电子系统的实验与设计。其实所采用的 Multi-task Reconfiguration 技术已被广泛应用，如虚拟仪器、通用编程器等。使系统的灵活性和高速特性两方面都得到了充分的满足。

### 四、模块化自主创新设计结构。

KX-CDS5 主系统板上含有模块化实验系统，已成为高校目前十分流行的实践平台，其主要优势是：

- ◆ 由于系统的各实验功能模块可自由组合、增减，故不仅可实现的实验项目多，类型广，更重要的是很容易实现形式多样的创新设计；
- ◆ 由于各类实验模块功能集中，结构经典，接口灵活，对于任何一项具体实验设计都能给学生独立系统设计的体验，甚至可以脱离系统平台；
- ◆ 面对不同的专业特点，不同的实践要求和不同的教学对象，教师，甚至学生自己可以动手为此平台开发增加新的实验和创新设计模块；
- ◆ 由于系统上的各接口，以及插件模块的接口都是统一标准的，康芯提供所有接口电路，因此此系统可以通过增加相应的模块而随时升级。

### 五、允许 FPGA 中运行多种经典实用处理器 IP 核。

KX-CDS5S 系统允许在核心板的大规模 FPGA 中运行多种经典实用软硬处理器 IP 核，包括：(1) 基于微指令的 8 位模型处理器、(2) 基于状态机指令控制结构的 16 位实用处理器、(3) 8051 软核处理器(基于商业级

全兼容 MCS-51 单片机 IP 核。利用此核,实验者可以实现传统单片机实验系统无法达到的 SOC(片上系统)设计。即将单片机 CPU、RAM、ROM 以及其它各类接口电路模块设计在同一片 FPGA 中。此类技术对于面向高新技术企业的就业十分必要)、(4) 32 位软核 NiosII Qsys 嵌入式系统处理器、(5) SOC FPGA 的嵌入式硬核 ARM 处理器 (Cortex-A9 单核或双核)。

这许多课程的实验和创新设计中,这些核具有不可或缺的地位,例如现代计算机组成原理的实验需要以上的软核 (1)、(2)、(3) 等;基于 SOC 技术的微机原理与接口技术的实验和综合性自主实验所需要的软核有 (4) 和 (5);面向单片机技术,单片机与 FPGA 接口以及单片机的 SOC 技术,则需要软核处理器 (3);基于嵌入于 FPGA 的 ARM 硬核的开发应用则需处理器 (5);SOPC 技术的学习和应用,包括机器人控制则涉及处理器 (4) 的使用;而在综合性要求更高的 EDA 技术实验和创新开发中,处理器 (1)、(2)、(3)、(4)、(5) 等都可能涉及。

## 六、可二次开发型 DDS 函数信号发生器。

KX-CDS55 主系统板上包含一个完整的可重构(可二次开发)的全数字型 DDS 函数信号发生器。采用了数字频率直接合成技术、频率精度高、无量程限制、信号过渡时间短、波形精度高、全程扫描特性好、稳定可靠等等;而且还实现了许多独特的功能,如宽频率等精度测频率、测脉宽/占空比功能、信号采集显示、逻辑分析仪、宽频域李萨如图形信号输出、任意波编辑输出及其频谱显示等。该信号发生器基于 EDA/SOPC 设计技术及数控制振荡器 NCO、AM 纯数字发生器、数字锁相环等 IP 核。它解决了普通 DDS 信号发生器的传统缺陷(如灵活性差,功能有限、无定制功能等),特别适合于 DDS 信号发生器二次开发、电子设计竞赛、通信系统开发等场合中应用。同时也为学习者自主创新能力的培养和施展提供了优良的平台

与传统方式不同,此系统没有使用 DDS 专用 IC 及模拟乘法器,而是利用 IP 核,数字乘法器、NCO 核、AllPLL 锁相环等,以纯数字的方式构建在一片 FPGA 内,因此具有可二次开发性。即可在此 FPGA 中从新构建系统或增加新的硬件功能。当用户需要用到某种特定功能及信号输出时,即可选择使 DDS 系统进入用户自定义/设计功能项,此时该系统将交出所有控制权、系统构建权。此时将由用户自行设计所需要的功能模块。

## 七、康芯的 KX-CDS 型模块化综合创新实验开发系统的课程/实验设计类型有以下 6 类:

本开发系统支持的课程/实验设计类型主要包括以下 9 类:

- ☆ EDA 技术系列实验。配套教材 2、4、6,含多个经典和创新实验与设计项目,多数含源程序演示示例。
- ☆ 单片机技术系列实验。配套教材。可完成许多传统和现代的单片机实验,特别包括与 FPGA 接口的实用系统设计实验项目。
- ☆ SOPC 技术系列实验。含基于 CycloneIV 等 FPGA 的 SOPC 实验,包括源程序演示示例。
- ☆ 基于单片机 IP 核的 SOC 片上系统设计系列实验。配套教材 3。基于 FPGA 硬件平台的 8051 核系统设计,含源程序演示示例。
- ☆ 计算机组成与设计系列实验。配套教材 1。基于配备的大规模 FPGA 和各类接口设备,成为计算机组成与设计创新实验的最好选择。
- ☆ 电子设计竞赛系列项目开发训练。作为电子设计训练平台,以其极大的灵活性和实用性,比传统固定结构的实验系统更能胜任此项任务。

注. 图 2-1 的 KX-CDS 平台上可以插图 2.4 所示所有模块,构成不同配置的创新开发系统,平台每一模块接口基本相同,因此多数模块可以安插在系统上的任一插座上,十分灵活。插座上也可插教师或学生自主开发的升级模块或特定设计项目的模块。由教材[2]给出的大量的实验和设计项目涉及许多不同类型的扩展模块,主系统平台上有许多标准接口,以其为核心,对于不同的实验设计项目,可接插上对应的接口模块,如 GPS 模块、彩色液晶模块、USB 模块、各类 ADC/DAC 模块等。这些模块可以是现成的,也可以根据主系统平台的标准接口和创新要求由读者(教师或学生)自行开发。

## 1.2 配套教材 (主要作者: 潘松, 黄继业, 潘明, 陈龙)

### 1、《现代计算机组成原理》第二版 教育部推荐之精品教材



杭州康芯的现代计算机组成与设计实验开发系统的配套教材是科学出版社的《现代计算机组成原理》作者潘松等。该书于 2007 年评为“十一五”国家级规划教材; 又于同年被国家教育部评为国家级精品教材。

从教育部的网上可以了解到, 教育部向高校推荐的 218 种精品教材的品种中, 涉及计算机组成原理教学的, 唯《现代计算机组成原理》一种。

事实上, 多年来在高校中被广泛选用的有关计算机组成原理的教科书不下数十种, 其中不乏名师名著; 然而《现代计算机组成原理》一书却何以能先后被教育部组织的专家评审并推荐为“十一五”国家级规划教材和国家级精品教材呢? 显然, 在高等教育领域, 除传播专业知识外, 培养学生的自主创新能力将成为一个教学领域中永恒的主题!

《现代计算机组成原理—原理、结构与设计技术》第二版对第一版的《现代计算机组成原理》作了加大的改进, 其中包括: 1、将原来的基于 VHDL 语言换成了更流行实用的 Verilog 语言; 2、对于原来许多重要且精彩的内容作了保留和进一步改进, 包括 8 位模型机的结构与设计与设计、16 位计算机的设计、基于计算机体系结构探讨的流水线和精简指令 CPU 设计等; 3、删除了 SOPC 的内容, 增加了 8051 单片机核的应用系统构建, 以及基于 8088/8086 微机系统 IP 核的 SOC 应用系统构建、调试和应用; 另外还增加了 32 位 OPEN RISC 软核嵌入式系统构建、设计与应用; 4、软件平台升级为 QuartusII9. x, 硬件实验平台升级为 CycloneIII EP3C55F484。

### 2、《EDA 技术实用教程》——(第六版, Verilog HDL/VHDL 两个版本)



本书根据课堂教学和实验操作的要求, 以提高实际工程设计能力为目的, 深入浅出地对 EDA 技术、VerilogHDL 硬件描述语言、FPGA 开发应用及相关知识做了系统和完整的介绍, 使读者通过本书的学习并完成推荐的实验, 能初步了解和掌握 EDA 的基本内容及实用技术。

全书包括 EDA 的基本知识、常用 EDA 工具的使用方法和目标器件的结构原理、以向导形式和实例为主的方法介绍的多种不同的设计输入方法对 Verilog 的设计优化以及基于 EDA 技术的典型设计项目。各章都安排了习题和针对性较强的实验与设计。书中列举的大部分 Verilog 设计实例和实验示例实现的 EDA 工具平台是 Quartus II, 硬件平台是 Cyclone III/IV 系列 FPGA,

并在 EDA 实验系统上通过了硬件测试。并加入了 16 位 CPU 设计, 在第 14 章节对 Test Bench 仿真, 和重点介绍了 Modelsim-Altera 的使用方法。

该教材从一面世, 就受到广大高校教师的欢迎, 使用范围不断扩大, 从 2002 年开始的第一版, 到目前的第 5 版, 经出版社的统计, 印数已近 50 万册。

### 3、《单片机原理及实用技术》



本书主要介绍单片机原理及实用技术, 讲解深入浅出, 内容新颖实用。主要章节保留了经典 51 系列单片机的基本教学内容, 并沿袭了传统的教学流程, 但其中的许多内容突破了传统的单片机应用理念, 对此课程的教学目标有了全新的拓展和延伸, 例如, 删除了许多在传统情况下必讲而已无任何实用价值的内容, 增加了实用的 ADC/DAC 内容以及单片机与 FPGA 扩展和基于单片机 IP 软核的片上系统构建及应用的知识, 从而开拓了一个将普通单片机技术、FPGA 开发技术、EDA 技术、片上系统应用技术有机融合、综合运用和培养自主创新能力的平台。全书共 9 章, 主要内容包括 51 系列单片机结构、汇编指令及程序设计、定时/计数器与中断应用、串行通信接口技术、单片机普通扩展技术、高速串行 ADC/DAC 扩展、单片机 C 语言编程设计、单片机与 FPGA 接口及软硬件联合设计技术、基于单片机 IP 软核的 FPGA 片上系统应用技术。

与传统的单片机教材不同, 由于安排了单片机与 FPGA 的接口技术及基于单片机 IP 核的 SOC 片上系统内容的介绍和应用实例的推荐, 该教材成为 EDA 技术课程后续教学、实践和创新培养必不可少的内容。

### 4、《EDA 技术及其应用》第二版 (图 1-4) 高职高专专用教材: 高等职业教育“十一五”规划教材

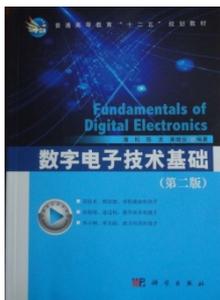


本书采用教、学、做相结合的教学模式, 以提高实际工程应用能力为目的, 通过实例引入, 深入浅出地介绍 EDA 技术、Verilog 硬件描述语言、FPGA 开发应用及相关知识, 并给出了丰富的 EDA 设计实例, 使读者通过本书的学习并完成推荐的实验, 能初步了解和掌握 EDA 的基本内容及实用技术。

全书内容分四部分, 第一部分简要介绍了 EDA 技术背景知识、常用的 EDA 基本工具使用方法和目标器件的结构原理; 第二部分通过实例, 以向导的形式介绍了三种不同的设计输入方法; 第三部分对 Verilog HDL 的设计作了介绍; 第四部详细讲述了基于 EDA 技术的典型的设计项目。各章都安排了相应的习题和有较强针对性的实验、设计实践要求。书中给出的绝大部分 Verilog 设计实例和实验示例都在 EDA 实验系统上通过了

硬件测试。

## 5、《数字电子技术基础》第二版



电子工业出版社出版。该书将传统的数字电路教学实验的内容与现代电子技术的理念和方法有机融合，同时革除传统教材中陈旧过时，甚至有误导性的内容，优化结构，在满足原理认知的基础上激发自主学习能力和自主设计能力，降低与其它前期课程知识的依存度，但加大有效知识梯度，从而使此教材能分别独立实现作者设定的2个重要目标：a) 学生通过基于该教材的数字电路课后，自主学习能力和自主创新能力可大幅提高；b) 不改变原课程设置的条件下可将数字电路课程大为提前，如移到大一年级下，使学生尽早接触与与现代社会技术相关的所有课程成为可能，并有充分的时间对所学的知识进行消化以及有更多的实践机会。从而使学生在本科阶段就积累了面向社会竞争所需的工作和学习能力，成为一个合格的有自主创新能力的工程师。

该教材可作为高校本科数字电路课程教学改革教材的优先选择。现已有不少学校依据该教材对数字电路课程作出了成功的教学改革，为培养卓越工程师和创新型人才作了坚实的铺垫。

## 6、《EDA技术与VHDL》（Verilog HDL 第二版/VHDL 第四版两个版本）



清华大学出版社，该书系统的介绍了 EDA 技术和 VHDL/verilong 硬件描述语言，将硬件描述语言的基础知识，编程技巧和实用方法与实际工程开发技术在先进的 EDA 设计平台——QuartusII 上很好的结合起来，使读者能通过本书的学习迅速的了解并掌握 EDA 技术的基本理论和工程开发实用技术，并为后续的深入学习和发展打下坚实的理论与实践基础。作者依据高校课堂教学和实验操作的规律与要求，并以提高学生的实际工程设计能力和自主创新能力为目的，对全书内容作了恰当的编排。全书共分为7个部分其中有基于 ModelSim 的 Test Bench 仿真技术，基于 MATLAB 和 DSP Builder 的平台的 EDA 设计及大量的实用系统设计示例，并安排针对性习题及设计项目。

## 7、《微机原理、接口技术及其片上系统实现》



科学出版社将于 2014 年初出版。在传统教材《微机原理与接口技术》经典内容的基础上，将这些内容与现代 EDA 技术、硬件描述语言、SOC 片上系统技术等有机结合，为这门经典的老课程赋予了全新的生命。例如在介绍了传统的相关内容后，又详细介绍了在结构和功能上完全对应的 8088/8086 CPU 核、8253 定时器核、8237 DMA 核、8259 中断控制核、8255 可编程 I/O 核、8250 UART 串行通信核等，以及它们与 8088/8086 CPU 核间的 SOC 系统构建，并给出大量 SOC 软硬件设计实例。于是使读者从原来仅仅学习一些经典原理和接口技术的被动过程，转变为能够能动地设计和构建整个硬件片上系统，将传统的微机原理探讨和接口技术与现代 EDA 技术及片上系统设计技术有机结合，将学习目标调整为培养学习者的自主创新能力，从而极大地提升了这门课程的重要性和必要性。

## 第二章 KX-CDS 主系统平台

本章主要介绍 KX-CDS 系列主系统平台的结构和性能，更形象的说明可以参考以下内容及光盘“KX-CDS 系统及模块说明”。

为了使实验系统能更好地实现实验者自主创新能力和工程实践能力的提高，KX-CDS 系统采用多任务重配置和模块化相结合结构，以下对主系统做说明，说明采用标注形式表达。

### 2.1 主系统介绍（主板为：KX-EDA5）

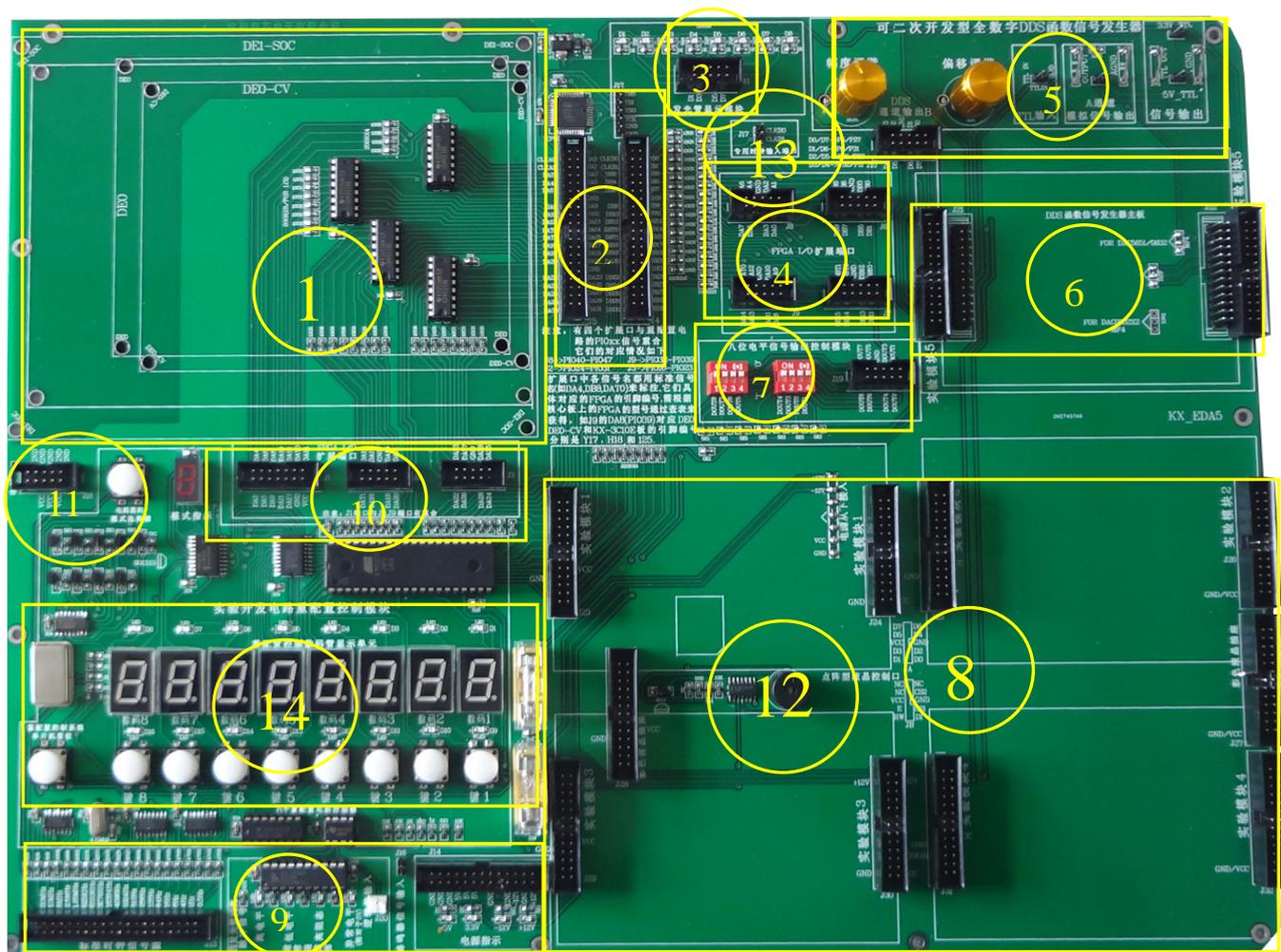


图 2-1 KX-CDS 主系统平台

#### ● 标注“1”

区域是加装核心板，DE 系列和 KX 系列 KX4CE6(10\55)、DE0\DE0-CV\DE1-SOC

#### ● 标注“2”

两组 40 芯座和核心板的把预留的 2 个 40 芯座通过排线向主系统引出，并且每个 IO 口都标上通用的标识号，比如 DB0 或 DA0,再根据标识号和 FPGA 具体型号，可对照本讲义 2.4 查出相应引脚号。

#### ● 标注“3”

是上方发光管控制端口，可根据标识和每个发光管一一对应。

#### ● 标注“4”

是四组 10 芯座 FPGA 的 IO 口的引脚，中间上端是 GND 脚，可利用十芯线连接扩展板，根据每个边标号查表查出对应 FPGA 的引脚号，“J8、J9”和多任务重配置引脚冲突，不可同时使用，“J6、J7”独立 IO 口没冲突；

#### ● 标注“5”

是用于调谐输出模拟信号和偏移电平的幅度；DDS 函数信号发生器的 TTL 信号输输入口；DDS 函数信号

发生器模拟信号输出通道的 A 通道（此信号发生器可以输出双通道模拟信号），如正弦波信号等，幅度最大 $\pm 10V$ ，可通过电位器调谐，TTL 信号输出口。

### ● 标注“6”

J21/J22 此座用作专门插 DDS 模块，具体 DDS 功能请参考以下” DDS 模块说明，J35/J36 是接插另外一个 DDS 模块，或插扩展板。

### ● 标注“7”

十芯座是可外引 8 个 IO 口，可通过查表获得引脚号，左边是控制 IO 口的拨码开关，拨上为‘0’，下为‘1’。

### ● 标注“8”

分别扩展板座，每个含两个 26 针。它们的尺寸大小，结构布置和信号安排大致相同。所以以下所述的多数实验功能模块可以随意插在这四组插座中任何一个位置上，这为实验系统的灵活构建奠定了基础。

其中 J28/J27 是插彩色液晶专用座，左右插座分别标“彩色液晶插座”，每组座上提供含有地 GND 和工作电源 VCC (+5V)；“JP29/JP30(实验模块 3) 提供 $\pm 12V$  电压的设置，适合插带有 DA 运放实验的板。

### ● 标注“9”

左边是 0.5HZ 到 20MHZ 时钟源，用户可通过单引线引出，或链接“J17” CLKB0/1 FPGA 专用时钟脚；中间是逻辑笔功能模块，用户可以通过“J20”测试逻辑功能；左边模块是系统提供的电压源，边标有电压值，用户可根据自己需要外引。板的下端有两个针端口“J16”是蜂鸣器的输入口，如果使用主系统蜂鸣器时，需要单线和 IO 相互连接；“J14”是电源输出口，各电压值在边上已标出，可用单线外引，注意：有 $\pm 12V$ 。

### ● 标注“10”

IO 口外引十芯座，和标注“4”功能相似，FPGA 的 IO 口的引脚在边上已经标出，根据每个边标号查表查出对应 FPGA 的引脚号，中间上端是 GND 脚，可利用十芯线连接扩展板，其中两组十芯座“J2、J3”和多任务重配置有关引脚冲突，十四芯座是用来连接彩色液晶屏的口用，不仅和多任务重配置 IO 口有冲突，还和标注“4”上的“J8、J9”兼用，所以使用此口时，“J8、J7”和多任务重配置功能相关 IO 口不能同时使用。

### ● 标注“11”

“J18”是电源 VCC 和 GND 十芯座，可用十芯线连接康芯设计的核心板对应的端口，以达到给核心板供电的目的，当然友晶 DE 系列板不需要；

“J17”是专业时钟源”边上标出引脚名“CLKB0/CLKB1”可通过查表获得引脚号,使用时通过标注 9 上的时钟源，通过单线连接。

### ● 标注“12”

主系统上扬声器，输入端口在标注“9”位置，通过单线连接 IO 输入给此端口。

### ● 标注“13”

“J17”是专业时钟源”边上标出引脚名“CLKB0/CLKB1”可通过查表获得引脚号。

### ● 标注“14” “多任务重配置”功能区域：

#### 1、选择键及模式数码显示”

按动按键，数码显示“1-B”，该电路结构能仅通过一个键，完成纯电子切换，（有的产品只能通过许多机械开关手动切换）的方式，Multi-task Reconfiguration 电路结构（多功能重配置结构）选择十余种不同的实验系统硬件电路连接结构，大大提高了实验系统的连线灵活性，但又不影响系统的工作速度（手工插线方式虽然灵活，但会影响系统速度和电磁兼容性能，不适合高速 FPGA/SOPC 等电子系统实验设计）。该系统的实验电路结构是可控的。即可通过控制接口键，使之改变连接方式以适应不同的实验需要。因而，从物理结构上看，实验板的电路结构是固定的，但其内部的信息流在主控器的控制下，电路结构将发生变化重配置。这种“多任务重配置”设计方案的目的是有 3 个：1、适应更多的实验与开发项目；2、适应更多的 PLD 公司的器件；3、适应更多的不同封装的 FPGA 和 CPLD 器件。关于电路模式选择用法详见第 2.3 节；**模式切换使用举例：**若模式键选中了“实验电路结构图 NO.1”，这时的主系统板所具有的接口方式变为：FPGA/CPLD 端口 PI/031~28（即 PI/031、PI/030、PI/029、PI/028）、PI/027~24、PI/023~20 和 PI/019~16，共 4 组 4 位二进制 I/O 端口分别通过一个全译码型 7 段译码器输向系统板的 7 段数码管。这样，如果有数据从上述任一组四位输出，就能在数码管上显示出相应的数值，其数值对应范围为：

FPGA/CPLD 输出	0000	0001	0010	...	1100	1101	1110	1111
数码管显示	0	1	2	...	C	D	E	F

端口 I/O32~39 分别与 8 个发光二极管 D8~D1 相连，可作输出显示，高电平亮。还可分别通过键 8 和键 7，发出高低电平输出信号进入端口 I/O49 和 48；键控输出的高低电平由键前方的发光二极管 D16 和 D15 显示，高电平输出为亮。此外，可通过按动键 4 至键 1，分别向 FPGA/CPLD 的 PI00~PI015 输入 4 位 16 进制码。每按一次键将递增 1，其序列为 1, 2, ..., 9, A, ...F。注意，对于不同的目标芯片，其引脚的 I/O 标号一般是同 KX-CDS 系统接口电路的“PIO”标号是一致的（这就是引脚标准化），但具体引脚号是不同的，而在逻辑设计中引脚的锁定数必须是该芯片的具体的引脚号。具体对应情况需要参考第 2.2 节的引脚对照表。

## 2、“重配置控制系统复位”

在对 FPGA 下载以后，按动此键，起到稳定系统作用；在实验中，当选中某种模式后，要按一下左侧的**复位键**，以使系统进入该结构模式工作。注意此复位键仅对实验系统的监控模块复位，而对目标器件 FPGA 没有影响，FPGA 本身没有复位的概念，上电后即工作，在没有配置前，FPGA 的 I/O 口是随机的，故可以从数码管上看到随机闪动，配置后的 I/O 口才会有确定的输出电平。

## 3、键 1~键 8

为实验信号控制键，此 8 个键受“多任务重配置”电路控制，它在每一张电路图的功能及其与主系统的连接方式随模式选择键的选定的模式而变，使用中需参照“6”中的电路图。

4、发光管 D1~D16：受“多任务重配置”电路控制，它们的连线形式也需参照“6”的电路图。

## 5、实验电路信号资源符号图说明

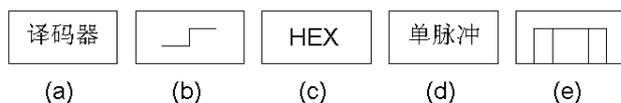


图 2-2 实验电路信号资源符号图

结合附图 2-2，以下对实验电路结构图中出现的信号资源符号功能作出一些说明：

(1) 附图 2-2a 是 16 进制 7 段全译码器，它有 7 位输出，分别接 7 段数码管的 7 个显示输入端：a、b、c、d、e、f 和 g；已有系统接入，用户不必接七段，用户只需接入输入端为 D、C、B、A，直接译码，D 为最高位，A 为最低位。例如，若所标输入的口线为 PI019~16，表示 PI019 接 D、18 接 C、17 接 B、16 接 A，对于刚接触语言学习者来说，不必先学译码器。

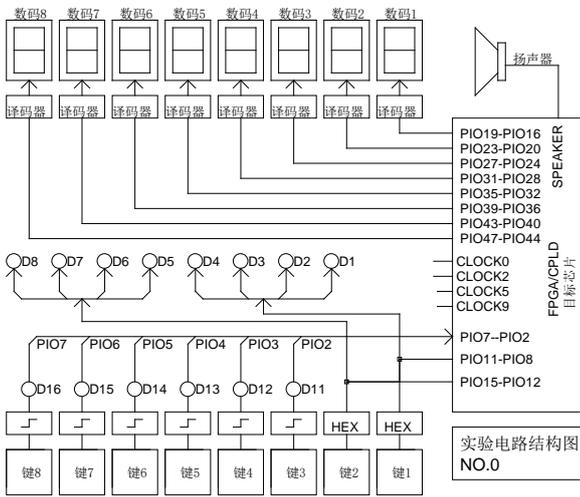
(2) 附图 2-2b 是高低电平发生器，每按键一次，输出电平由高到低、或由低到高变化一次，且输出为高电平时，所按键对应的发光管变亮，反之不亮。

(3) 附图 2-2c 是 16 进制码（8421 码）发生器，由对应的键控制输出 4 位 2 进制构成的 1 位 16 进制码，数的范围是 0000~1111，即 $\hat{H}0$  至 $\hat{H}F$ 。每按键一次，输出递增 1，输出进入目标芯片的 4 位 2 进制数将显示在该键对应的数码管上。

(5) 附图 2-2d 是单次脉冲发生器。每按一次键，输出一个脉冲，与此键对应的发光管也会闪亮一次，时间 20ms。

(6) 附图 2-2e 是琴键式信号发生器，当按下键时，输出为高电平，对应的发光管发亮；当松开键时，输出为高电平，此键的功能可用于手动控制脉冲的宽度。具有琴键式信号发生器的实验结构图是 NO. 3。

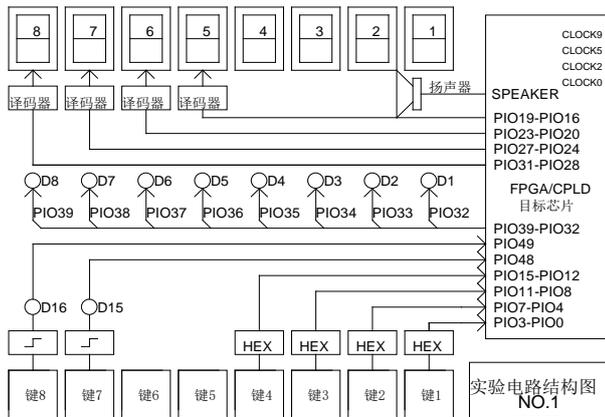
## 6. 各实验电路结构图特点与适用范围简述



(1) 结构图 NO. 0:

目标芯片的PIO16至PIO47共8组4位2进制码输出，经外部的7段译码器可显示于实验系统上的8个数码管。键1和键2可分别输出2个四位2进制码。一方面这四位码输入目标芯片的PIO11~PIO8和PIO15~PIO12，另一方面，可以观察发光管D1至D8来了解输入二进制的数值。例如，当键1控制输入PIO11~PIO8的数为“B”时，则发光管D4和D2亮，D3和D1灭。电路的键8至键3分别控制一个高低电平信号发生器向目标芯片的PIO7至PIO2输入高电平或低电平。

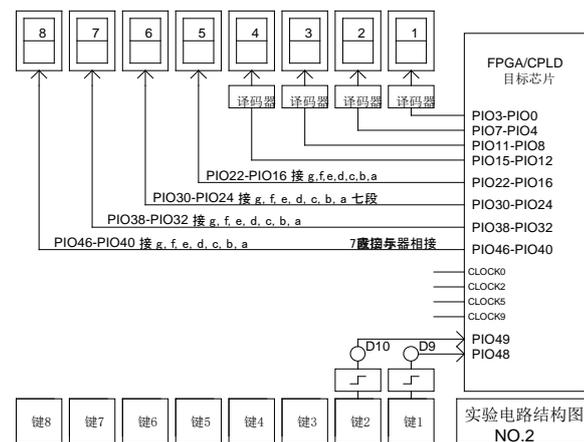
图 2-3 实验电路结构图 NO.0



(2) 结构图 NO. 1:

适用于作加法器、减法器、比较器或乘法器等。例如，加法器设计，可利用键4和键3输入8位加数；键2和键1输入8位被加数，输入的加数和被加数将显示于键对应的数码管4-1，相加的和显示于数码管6和5；可令键8/7控制此加法器的最低位进位。

图 2-4 实验电路结构图 NO.1



(3) 结构图 NO. 2:

直接与7段数码管相连的连接方式的设置是为了便于对7段显示译码器的设计学习。以图NO.2为例，如图所标“PIO46-PIO40接g、f、e、d、c、b、a”表示PIO46、PIO45..PIO40分别与数码管的7段输入g、f、e、d、c、b、a相接。可用于作VGA视频接口逻辑设计，或使用数码管8至数码管5共4个数码管作7段显示译码方面的实验；而数码管4至数码管1，4个数码管可作译码后显示，键1和键2可输入高低电平。

图 2-5 实验电路结构图 NO. 2

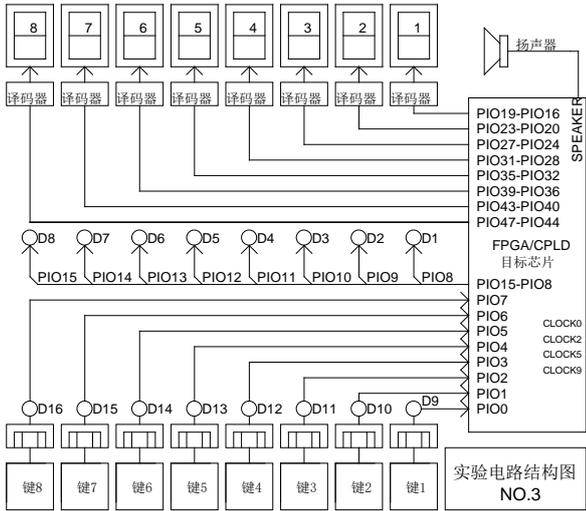


图 2-6 实验电路结构图 NO. 3

(4)结构图 NO.3:

特点是有 8 个琴键式键控发生器，可用于设计八音琴等电路系统。也可以产生时间长度可控的单次脉冲。该电路结构同结构图 NO.0 一样，有 8 个译码输出显示的数码管，以显示目标芯片的 32 位输出信号，且 8 个发光管也能显示目标器件的 8 位输出信号。

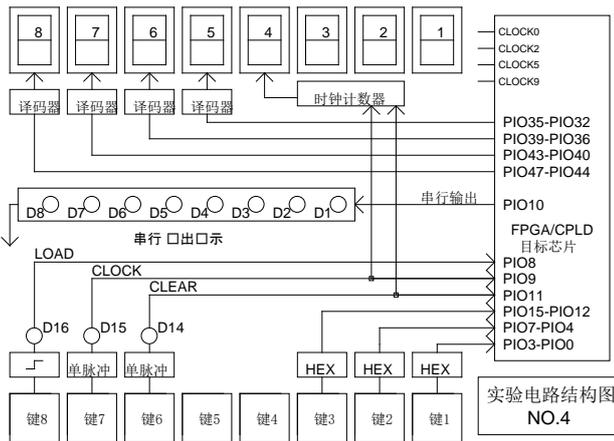


图 2-7 实验电路结构图 NO. 4

(5)结构图 NO. 4:

适合于设计移位寄存器、环形计数器等。电路特点是，当在所设计的逻辑中有串行 2 进制数从 PIO10 输出时，若利用键 7 作为串行输出时钟信号，则 PIO10 的串行输出数码可以在发光管 D8 至 D1 上逐位显示出来，这能很直观地看到串出的数值。

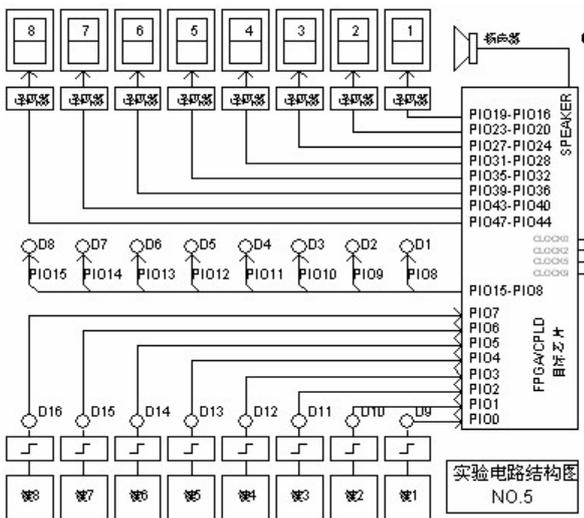


图 2-8 实验电路结构图 NO.5

(6)结构图 NO. 5:

8 键输入高低电平功能，目标芯片的 PIO19 至 PIO44 共 8 组 4 位 2 进制码输出 32 位，经外部的 7 段译码器可显示于实验系统上的 8 个数码管。

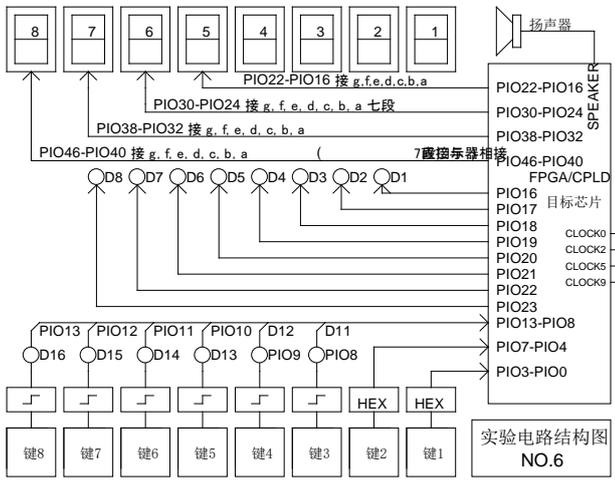


图 2-9 实验电路结构图 NO. 6

(7) 结构图 NO. 6:

此电路与 NO.2 相似,但增加了两个 4 位 2 进制数发生器,数值分别输入目标芯片的 PIO7~PIO4 和 PIO3~PIO0。例如,当按键 2 时,输入 PIO7~PIO4 的数值将显示于对应的数码管 2,以便了解输入的数值。

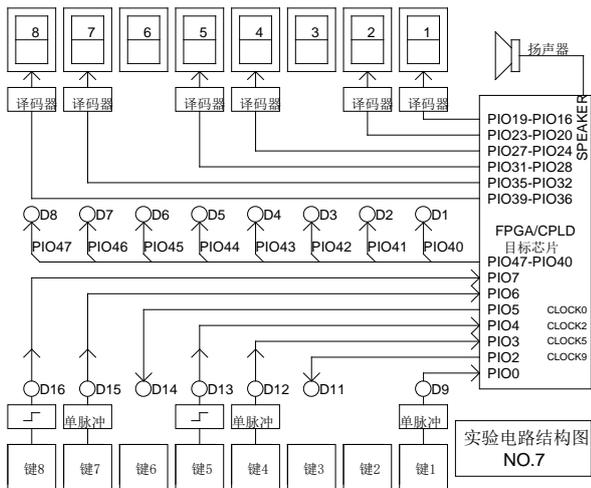
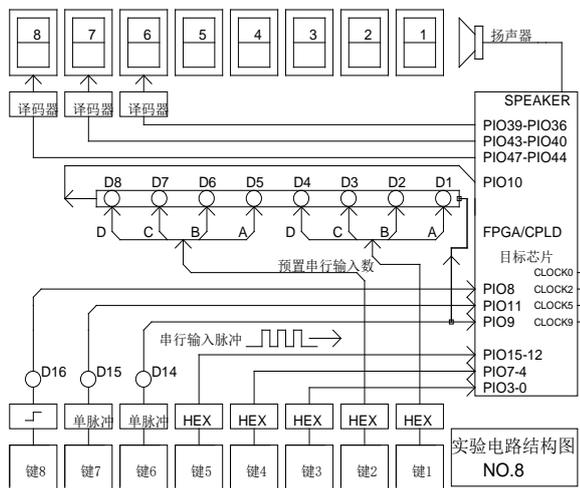


图 10 实验电路结构图 NO. 7

(8) 结构图 NO. 7:

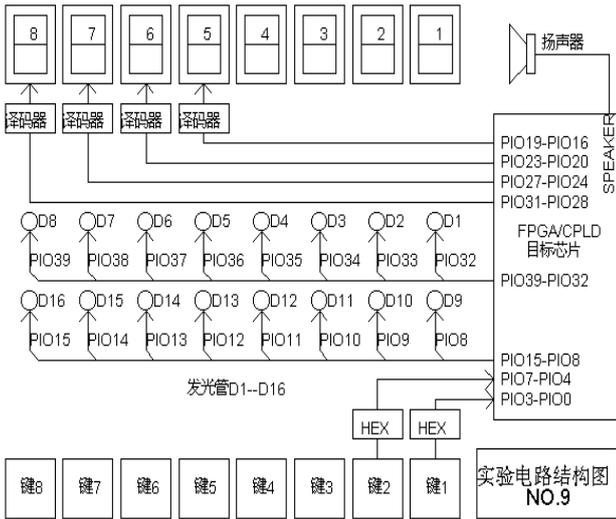
此电路适合于设计时钟、定时器、秒表等。因为可利用键 8 和键 5 分别控制时钟的清零和设置时间的使能;利用键 7、4 和 1 进行时、分、秒的设置。



(9) 结构图 NO. 8:

此电路适用于作并进/串出或串进/并出等工作方式的寄存器、序列检测器、密码锁等逻辑设计。它的特点是利用键 2、键 1 能序置 8 位 2 进制数,而键 6 能发出串行输入脉冲,每按键一次,即发一个单脉冲,则此 8 位序置数的高位在前,向 PIO10 串行输入一位,同时能从 D8 至 D1 的发光管上看到串形左移的数据,十分形象直观。

图 11 实验电路结构图 NO. 8



(10) 结构图 NO. 9:

若欲验证交通灯控制等类似的逻辑电路，可选此电路结构。

图 12 实验电路结构图 NO. 9

## 2.2、FPGA 核心板扩展至康芯主系统引脚对照表（模式类）

结构图上的信号名	KX-4CE6/10EP4 CE6/10 E22C8 CycloneIV 引脚号	<b>KX-10CE55</b> <b>10CL55YF484</b> <b>Cyclone10</b> 引脚号	KX-4CE55 EP4CE55F484 CycloneIV 引脚号	DE0 E3C16F484C6N CycloneIII 引脚号	DE0-CV 5CEBA4F23C7N CycloneV 引脚号	DE1-SOC 5CSEMA5F31 C6N CycloneV 引脚号	2 组 40 芯 端口名 扩展口名	DE10- nane5CSEBA6 U23II7 CycloneV 引脚号
PI00	52	N1	N1	U7	T15	AJ21	DB31	AA11
PI01	55	R1	R1	W6	T18	AG20	DB29	AA26
PI02	64	V1	V1	V8	T20	AG21	DB27	AB26
PI03	66	Y1	Y1	W10	R17	AF21	DB25	Y17
PI04	67	AB3	AB3	V11	P18	AE19	DB23	W14
PI05	75	AA6	AA6	V12	K17	AD20	DB21	W11
PI06	34	Y7	Y7	W13	L17	AK21	DB19	AB23
PI07	84	AB6	AB6	U14	M18	AJ20	DB17	AC22
PI08	60	U2	U2	T8	R16	AF19	DB26	AB25
PI09	65	W2	W2	Y10	R15	AF20	DB24	Y18
PI010	70	AA3	AA3	R10	K19	AE18	DB22	AA18
PI011	74	AB5	AB5	U13	L19	AD19	DB20	AA19
PI012	77	W6	W6	Y13	L18	AH20	DB18	Y19
PI013	83	W8	W8	V14	P16	AH19	DB16	AC23
PI014	42	P1	P1	V7	K16	AC22	DA31	Y5
PI015	39	N2	N2	V6	J17	AA20	DA30	AC24
PI016	44	U1	U1	U8	G12	AD21	DA29	AA15
PI017	43	R2	R2	Y7	G13	AE22	DA28	AD26
PI018	49	W1	W1	T9	G15	AF23	DA27	AG28
PI019	46	V2	V2	U9	G16	AF24	DA26	AF28

KONXIN

PI020	51	Y14	AA1	T10	F12	AG22	DA25	AE25
PI021	50	Y2	Y2	U10	F13	AH22	DA24	AF27
PI022	59	AA5	AA5	R12	F15	AJ22	DA23	AG26
PI023	54	AA4	AA4	R11	F14	AK22	DA22	AH27
PI024	69	Y6	Y6	T12	E16	AH23	DA21	AG25
PI025	68	V6	V6	U12	E15	AK23	DA20	AH26
PI026	72	Y8	Y8	R14	E14	AG23	DA19	AH24
PI027	71	W7	W7	T14	C15	AK24	DA18	AF25
PI028	76	AB7	AB7	AB7	B15	AJ24	DA17	AG23
PI029	73	AA7	AA7	AA7	A14	AJ25	DA16	AF23
PI030	85	AB9	AB9	AA9	L8	AH25	DA15	AG24
PI031	80	AA9	AA9	T16	J13	AK26	DAT1	AH22
PI032	101	V11	V11	AB9	A15	AJ26	DA14	AH21
PI033	100	Y10	Y10	R16	H14	AK27	DAT0	AG21
PI034	113	AB14	AB14	V15	J11	AK28	DA13	AH23
PI035	105	AA13	AA13	W15	H10	AK29	DA12	AA20
PI036	120	T16	T16	T15	G11	AJ27	DA11	AF22
PI037	114	AA15	AA15	U15	J19	AH27	DA10	AE22
PI038	128	W17	W17	W17	J18	AH24	DA9	AG20
PI039	125	Y17	Y17	Y17	H18	AG26	DA8	AF21
PI040	135	AB16	AB16	AB17	G17	AG25	DA7	AG19
PI041	136	AA16	AA16	AA17	G18	AF26	DA6	AH19
PI042	137	U20	U20	AA18	D13	AF25	DA5	AG18
PI043	138	AB18	AB18	AB18	C13	AE24	DA4	AH18
PI044	141	AA19	AA19	AB19	B13	AE23	DA3	AF18
PI045	142	AB19	AB19	AA19	A13	AD24	DA2	AF20
PI046	143	U21	U21	AB20	B12	AC23	DA1	AG15
PI047	144	U22	U22	AA20	A12	AA21	DA0	AE19
PI048	58	P2	P2	W7	T19	AF18	DB28	Y11
PI049	53	M2	M2	V5	T17	AG18	DB30	AA13
CLKB0	90	W22	W22	AB12	N16	AC18	CLKB0	V12
CLKB1	91	W21	W21	AA12	M16	AD17	CLKB1	W12
<b>8051/ 88</b>	EP4CE6/10 E22C8	10CL55YF484 Cyclone10	EP4CE55F484 CycloneIV	E3C16F484C6N CycloneIII	5CEBA4F23C7N CycloneV	5CSEMA5F31 C6N	2组40芯 端口名	5CSEBA6U23I 17
MT	88	V22	V22	AB11	H16	AB17	CLKA0	AE12
NO	89	V21	V21	AA11	H15	AB21	CLKA1	AE20
PE0	127	AA21	AA21	AA15	D17	AK16	DB2	D8
PE2	124	W20	W20	AA14	K21	AK19	DB4	AF7
	133	Y22	Y22	AB16	B16	Y17	DB0	E8
	129	Y21	Y21	AA16	C16	Y18	DB1	D11
	126	AA20	AA20	AB15	K20	AK18	DB3	AH13
	121	AB20	AB20	AB14	K22	AJ19	DB5	AH15
	119	AA18	AA17	AB13	M20	AJ17	DB6	AF4
	115	AB4	AB17	AA13	M21	AJ16	DB7	AH3
	111	V16	V16	AB10	N21	AH18	DB8	AD5
	112	U16	U16	AA10	R22	AH17	DB9	AG14
	106	AA14	AA14	AB8	R21	AG16	DB10	AE23

110	AB15	AB15	AA8	T22	AE16	DB11	AE6
103	Y13	Y13	AB5	N20	AF16	DB12	AD23
104	AB13	AB13	AA5	N19	AG17	DB13	AE24
98	AA10	AA10	AB3	M22	AA18	DBT0	D12
99	AB10	AB10	AB4	P19	AA19	DB14	AD20
86	AA8	AA8	AA3	L22	AE17	DBT1	C12
87	AB8	AB8	AA4	P17	AC20	DB15	AD17

## 2.3 核心板及扩展模块介绍

本章主要介绍 KX-CDS 系统主要配套实验功能模块。这些模块可以是系统的配套模块，也可以是定购模块，或是根据此系统的接插口，以及开发项目的需要，自己设计出的模块。因此在 KX-CDS 系统上用于完成不同类型的实验和设计的模块数量和种类没有任何限制。这里仅将一些主要和核心的功能模块的结构特点和使用注意作一些介绍。至于对于这些模块更加详细的了解和熟练的应用必须通过实际使用后才能实现。

好在本讲义推荐的几乎所有实验与设计都配有示例源文件和对应的实验指导 ppt 课件，在这些课件的照片上有所有相关实验模块的详细使用指导，通过这些实验指导课件就能完全掌握实验模块的使用方法。

此外，还应该注意，这些模块的一个共同特点，即他们可以插于 KX-CDS 系统上组合成设计系统进行实验，也可脱离实验平台构成独立的模块和模块组合进行更加实际的系统，这是 KX-CDS 系统的主要特点。

各模块的具体情况可以参见相关实验指导 ppt 文件以及相关说明。

**实验操作方法：**模块与模块之间的连接方式，KX-CDS 实验平台上的实验模块之间的连接方式主要采用十芯线连接，为了用户使用简单方便，每个模块的控制及数据端口全部外引，大多数是十芯座为一组，所有模块都标准化，每个十芯座有 10 根针，中间的两个针分别上面是“GND”，其他 8 根针全部是用来引脚号，全部在旁边标出，用户在使用时，用十芯线连接，根据每根针所在的位置一一对应锁定引脚号即可。

为了用户快捷了解核心板及扩展板的接口功能，采用注解的方式进行介绍。

### 2.3.1 核心板 FPGA 模块

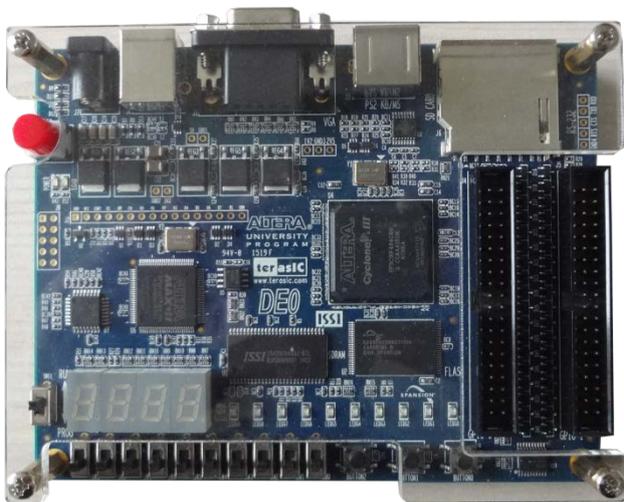


图 2-1 友晶 DE0 核心板

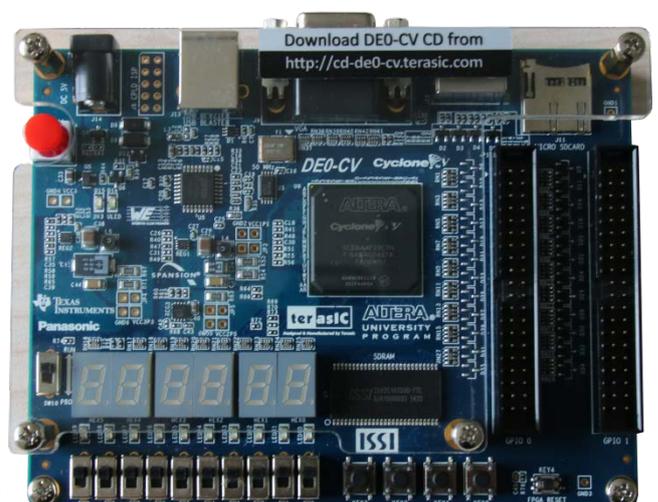


图 2-2 友晶 DE0-CV 核心板

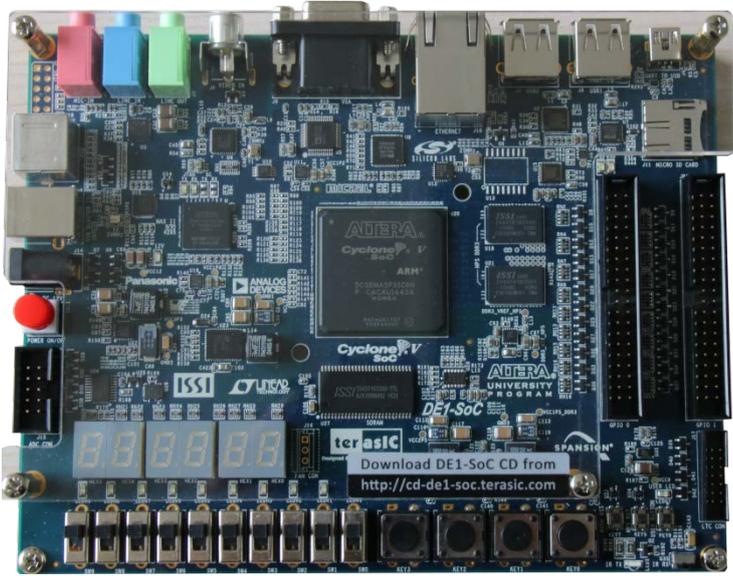


图 2-3 友晶 DE1-SOC 核心板

以上是 DE 系列核心板，有关资料请文件夹“友晶原版资料”，其中包括的原理图、板的使用说明，实验介绍等。

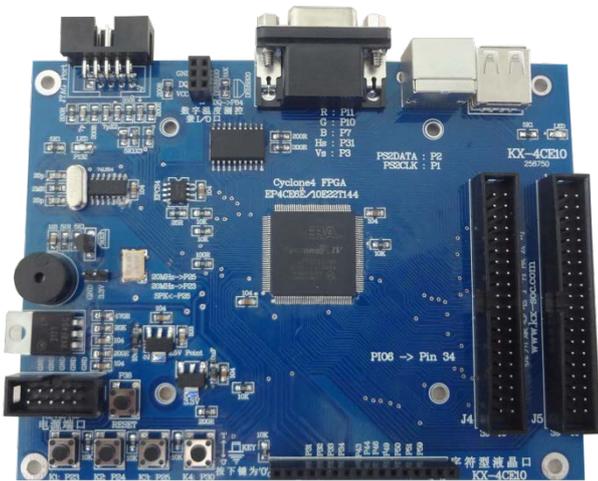


图 2-4 康芯设计的 KX-4CE6/10 板

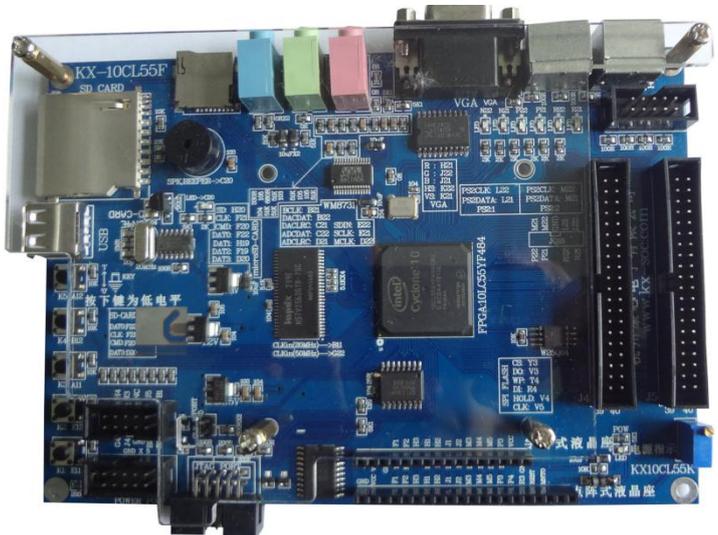
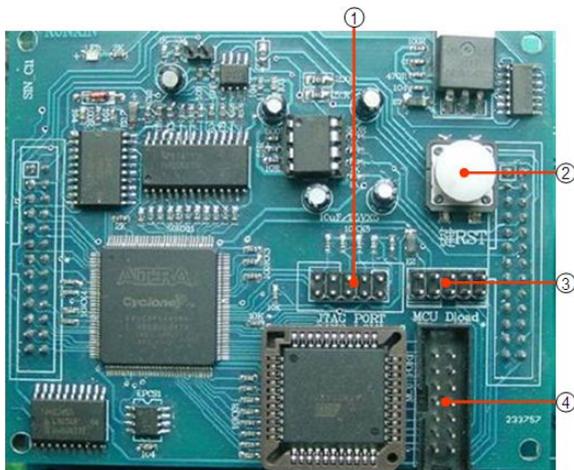


图 2-5 康芯设计的 KX-4CE55/10LC55 板

图 2-5 核心 FPGA 分别是 Cyclone IV EP4CE55F23C8 和 Cyclone 10 10CL55WF48，详细说明请参阅“CDS4/4S/10S”文件夹核心板说明及引脚对照表”

### 2.3.2 可重构型 DDS 全数字函数信号发生器



DDS 函数信号发生器主板

系统配套的全数字型 DDS 函数信号发生器模块含 FPGA、单片机、超高速 DAC、高速运放等。既可用于全数字型 DDS 函数信号发生器，同时也可作为 EDA/DSP 系统及专业级 DDS 函数信号发生器设计开发平台。作为 DDS 函数发生器的功能主要包括：高精度频率计，全程扫描频率源（扫速、步进频宽、扫描方式等可数控），移相信号发生，里萨如图信号发生，方波/三角波/锯齿波和任意波形发生器，以及 AM、PM、FM、FSK、ASK、FPK 等各类调制信号发生器。

系统上配有一个功能强大的 DDS 函数信号发生 此信号发生器的主模块插于平台的器。

左上图方。它必须结合插座“6”和“8”上插的 20 字 X4 行字符型液晶和插座 4X4 键盘，联合使用，这是实验的辅助测试和信号系统。由于此系统设计的操作较多，功能也较丰富，所以必须参考 ppt/PDF 文件，以便更详细地了解使用方法。对应文件是：重要 PDF 文件\系统标准 DDS 函数发生器用法.ppt/

可重构 DDS 函数信号发生器使用了有别于传统模拟信号发生器和普通 DDS 函数信号发生器的更新换代理念。尽管普通 DDS 函数信号发生器同样采用了数字频率直接合成技术，有许多模拟信号发生器无法比拟的优点：频率精度高、无量程限制、信号过度时间极短、波形精度高、不同方式和全程扫描特性好、调整功能强、全数字化控制、稳定可靠等等，但由于采用 DDS 专用器件，缺乏灵活性，功能受限于专用芯片的即定功能，不仅无法适应用户许多特定功能的要求，就是不少专用功能也无法实现，从而在不少应用场合使用户面对许多尴尬局面。这是因为任何一台功能强大的 DDS 信号发生器都不可能总是满足用户，特别是通信系统或一些电子系统设计领域的用户的需求，如一些特定编码方式或调制方式的信号发生功能和解调功能等。

可重构 DDS 函数信号发生器基于 EDA/SOPC 设计技术及数控制振荡器 NCO/DDS、AM 纯数字发生器（注意，目前绝大多数 DDS 信号发生器的 AM 信号是靠模数结合，如使用模拟乘法器等方式生成的，因此在数字通信中没有实用价值）、数字锁相环等 IP 核，是 EDA/SOPC 技术高度发展的产物，它彻底解决了普通 DDS 信号发生器的传统缺陷，而且整体功能和性能都有了质的飞跃。

1、作为普通函数信号发生器，从技术的先进性、功能的完备性、使用的便利性及性能指标的优越方面看，此系统无论作为普通信号发生器，还是高档函数信号发生器，都可谓当之无愧。

2、作为应用电路模块的开发系统。由于该系统是基于 EDA 技术和大规模高速 FPGA，具有良好的重构功能，以及端口完善的驱动与保护特性，开发者大量的硬件模型和实用系统（特别是通信领域中的各类功能模块）可以借助该系统，以及 QuartusII、硬件描述语言等迅速开发出来。

3、作为大学生电子设计竞赛的实验系统和开发系统。由于该系统中的许多功能都曾出现在历届大学生电子设计竞赛的赛题中，且该系统的性能指标都超越了相关赛题中发挥部分要求的技术指标，而实现方法又十分类似（技术类型和软硬件方面），所以无论作为培训工具还是实战开发系统，都可以帮助竞赛者高效对付许多类型的赛题。因此，该系统同样可作为毕业设计、学位论文、课余科技活动高效有力的开发工具。

4、作为自主创新型实验开发系统。创新就是原创，就是独创，在电子领域就是设计出全新而又性能优良适用面宽阔的系统或功能模块。创新是需要适当平台（可行性环境）的，几个 74 系列器件构成的平台，显然不如单片机系统，而单片机系统又不及嵌入式系统，但嵌入式系统在自主设计方面又不及 SOPC/EDA 技术。因为嵌入式系统中几乎所有硬件模块，从 CPU 到各种接口功能模块都是现成的，开发者主要工作是在软件方面，虽说可以在此平台上有许多创新之作，但最重要的自主知识产权却无法拥有，因为硬件的产权是属于别人的。显然，创新不等于自主，只有创新而没有自主，则很可能失去创新的价值和意义，从本质上看，便不能属于真正的创新。而 EDA/SOPC 技术则解决了硬件设计、软件设计和综合设计的根本问题，从而也解决了创新和自主这一对矛盾，不言而喻，基于 EDA 技术的平台将为设计者提供了最大可能的自主创新的平台。

DDS 函数信号发生器主要模块和电路结构在实验系统的左上侧，除了左侧的 DDS 主模块、液晶显示屏和 4X4 键盘外，在右上侧还有许多功能模块和信号通道：

**(1) A 通道。**这里 DDS 函数信号发生器模拟信号输出通道的 A 通道（此信号发生器可以输出双通道模拟信号），如正弦波信号等，幅度最大 $\pm 10V$ ，可通过电位器调谐。

**(2) TTL 信号输出。**此是 DDS 函数信号发生器的 TTL 信号输出口。

**(3) B 通道。**在主系统标注“4”是 DDS 函数信号发生器模拟信号输出通道的 B 通道之信号口。如果需要得到 B 通道的模拟信号输出，必须将此 B 通道口线与某一 DAC 的输入接口，然后得到输出信号，此通道在平台的左上方 J2 口。

**(4) 信号测试输入口。**即“TTL 输入”口。可以通过 DDS 函数信号发生器测试此口输入信号的频率、脉宽、占空比等。数字调制信号和扫频信号外部控制时钟也可通过此口进入。

**(5) 调谐电位器。**有两个电位器，一个用于调谐输出模拟信号的幅度，另一个调谐信号的偏移电平。

标注“1”是此板上 Cyclone1c3 FPGA JTAG 下载口，此口可对 FPGA 二次开发，用户可根据自己需要来开发。

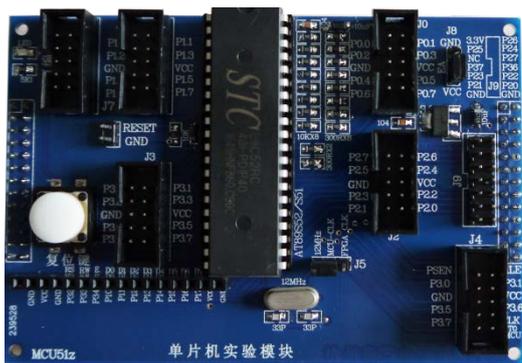
标注“2”是系统复位键，可对系统初始化。

标注“3”对单片机 8253 编程口。

标注“4”是 14 芯接 2004 液晶及 4X4 键盘的接插口，是用来 DDS 显示和操作的接口，对应的是 2004 液晶模块的标注“1”。

注意，一般情况下请不要清除和覆盖 FPGA 和单片机的程序，否则将无法运行 DDS 功能。

此 FPGA 由含 100 万门的 CycloneIII 新型大规模 FPGA EP3C10 构成，2 锁相环，44 万 RAM 位，EPCS4 4M Flash。超宽超高锁相环输出频率 1300MHz 至 2kHz！



### 2.3.3 单片机模块说明

单片机模块是 KX\_DN 系统配套的核心模块之一，原理图请参考文件夹“原理图\MCU8051\_SCH.pdf”，例程参考:\MCU\_TECH,法注意如下：标注“1”如果是单片机 AT89S51，则可通过康芯提供的下载器-USB to Serial Comm 口对其编程，即 USB 转串口对单片机（P3.0 和 P3.1）编程，这种方式更方便,编程方法参考附录 2。

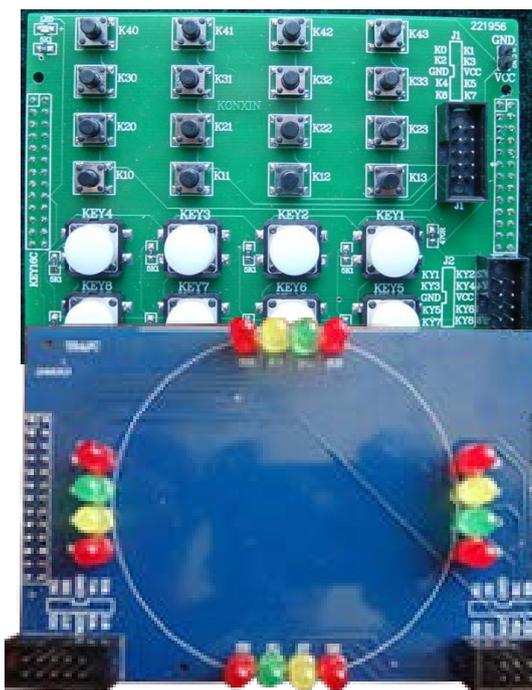
注：此单片机编程方法:\FPGA\_单片机\_编程，编程软件：VIVI\stc89c51 编程下载.rar



### 2.3.4 4X4 十六键键盘

此模块可作单片机实验键盘、FPGA 控制的键盘，也可兼做 KX\_DN 系统上的 DDS 函数信号发生器的控制键盘，因为此键盘上已标注每一键的功能。此键盘输出端口每一端口都含上拉电阻，做为 DDS 模块应插在 A6 座上。

标注“1”是此 16 键的 8 根线扫描控制端口，原理图请参考文件夹“原理图\KEYs4X4\_SCH.pdf”。如此模块做为 DDS 操作使用时连接方式可参考“DDS 使用说明”及 2004 字符液晶使用说明。



### 2.3.5 4X4+8 个单脉冲综合键盘模块

此键盘是综合使用键盘，上面的黑色键盘采用 8 个线扫描方式接 16 个键，可参考上面单独 4X4 键盘原理一样，下面 8 个白色键盘是独立的单脉冲键盘，原理图请参考文件夹“原理图\4X4KEYs\_8KEYs\_SCH.pdf”，实验参考：

/DEMO/EDA\_DEMO/MODU\_DEMO/DEMO3\_SCAN\_4X4KEY

### 2.3.6 交通灯显示模块

从右向左依次提供四组红、黄、绿、蓝交通模式，下方二十芯口分别是对应的灯的引脚端口。此板的例程路径：

/DEMO/EDA\_DEMO/MODU\_DEMO\DEMO1\_jiaotong



### 2.3.7 动态扫描模块

提供 8 组七段译码动态扫描数码，左边十芯口是数码段的接口端，右边是 8 数码的位控制端。参考示例：

/DEMO/EDA\_DEMO/MODU\_DEMO\DEMO2\_SCAN8\_LED

### 2.3.8 点阵式 128X64 液晶显示模块

此模块为点阵式 128X64 液晶显示模式，其使用手册在文件夹“VIVI\_FILE\LCD\_FILE”里。原理图请参考文件夹“原理图\LCD128X64\_SCH.pdf”。例程参考：



/DEMO/EDA\_DEMO/MODU\_DEMO/DEMO24\_PS2\_64X128LCD

标注“1”液晶的 7 位数据控制端口，对应的脚名在背面。

标注“2”是调节液晶背光的强弱的电位器。

标注“3”是此液晶的功能控制端口。

### 2.3.9 字符式 20X4 液晶显示模块

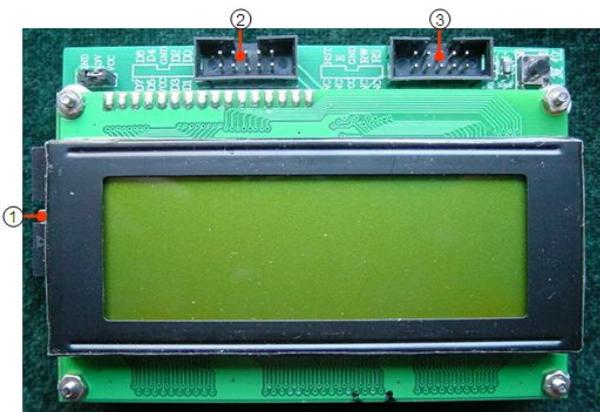
此模块做为 DDS 显示模块时，应插在 A3 做上。其使用手册在文件夹“VIVI\_FILE\LCD\_FILE”里。原理图请参考文件夹“原理图\LCD4X20\_SCH.pdf”。参考示例：

/DEMO/EDA\_DEMO/8051Core\_DEMO

标注“1”做为 DDS 模块显示时，此接口是通过 14 芯线连接到 DDS 模块的标注“4”的接口上，在把标注“2”端口利用 10 芯线连接到 4X4 键盘的标注“1”接口上。这样就构成了 DDS 的硬件操作系统。

标注“2”是此液晶的 8 位数据控制端口。

标注“3”是此液晶的功能控制端口。



### 2.3.10 800X480 数字 TFT 彩屏液晶显示模块

作为实验模块，此显示屏幕只能用 FPGA 驱动。此液晶的手册请浏览文件夹“VIVI\_FILE AT070TN83V.1.pdf”。原理图请参考文件夹“原理图\LCD\_CL1.pdf”。参考示例：

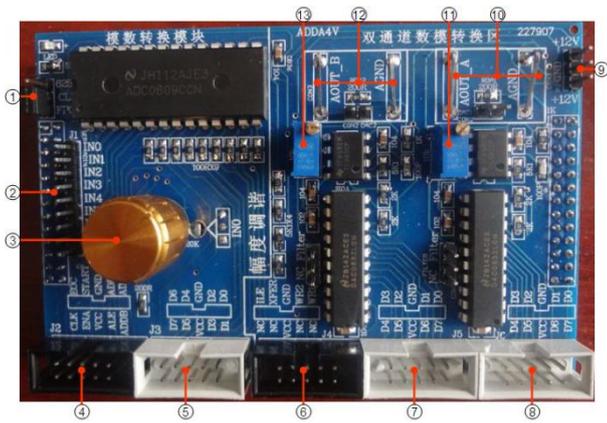
/DEMO/EDA\_DEMO/MODU\_DEMO

/DEMO9\_COLOR\_LCD；液晶右边上方两口是液晶驱动及数据口，如采用触摸形式，下方一端口是触摸数据端口，采用 ads7846 控制方式，彩色液晶显示屏上有 5 个跳线选择：

- 1、控制模式 MODE 跳线选择：选择“H”，即选择普通 LCD 扫描控制方法，“L”选择 VGA 方式
- 2、DCLK 跳线选择：选择“HS”即选择 VGA 方式扫描控制；“DCLK”：选择普通 LCD 控制方式。
- 3、VS/DE 跳线选择：选择“VS”即选择 VGA 方式扫描控制；“DE”：选择普通 LCD 控制方式。
- 4、L/R 跳线选择：选择“H”即选择从右至左方式扫描；选择“L”，即选择从左至右方式扫描。



5、U/D 跳线选择：选择“H”即选择从上至下式扫描；选择“L”，即选择从下至上方式扫描。



### 2.3.11 8 位双通道 DAC 和 ADC 标准模块

此板一个是 ADC0809，双通道的 DAC0832。原理图请参考文件夹“原理图\ADC\_DAC\_SCH08090832.pdf”。参考示例：/DEMO/EDA\_DEMO/MODU\_DEMO

/DEMO5A\_ADC0809\_VHDL  
&/DEMO10\_DDS\_SIN\_WAVE

标注“1”是 AD0809 需提供时钟的方式，如跳线帽跳上，0809 的工作时钟需通过主系统提供，注意，主系统 A8 座提供 625KHZ 的时钟，此板必须插在 A8 座上，如跳线帽跳下，此板可插其他座上，但时钟需通过 FPGAIO 口提供，标注“4”

是外围时钟输入口，FPGA 可对应于其口输入时钟。

标注“2”是 AD0809 的模拟输入通道。

标注“3”是 AD0809 通道“IN0”输入的选择钮，可通过此按钮输入电压信号。

标注“4”是 0809 的控制端口，其中有一个“CLK”端，是 FPGA 向 0809 的输入时钟端。

标注“5”是 0809 的数据输出端。

标注“6”是 DA0832 的控制端，用户可根据 0832 的使用手册进行控制。

标注“7”是 0832 的 B 通道的数据输入端。

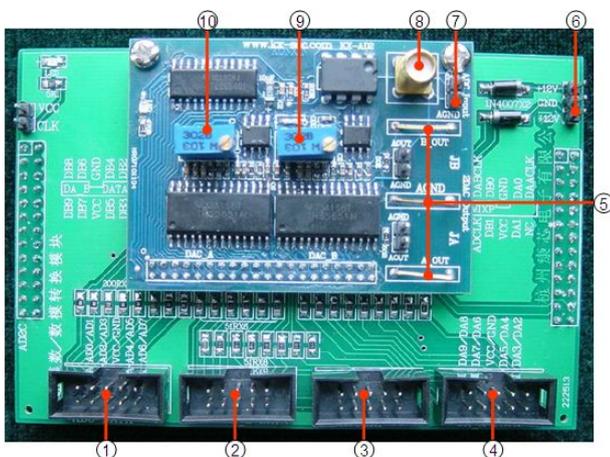
标注“8”是 0832 的 A 通道的数据输入端。

标注“9”是+12V 输入端，注意，上为-12V，这里板上标错，一般此板脱离主系统才用到。如果插在主系统上使用，要选择主系统的带+12V 的的插座。

标注“10、12”分别是 A/B 通道的输出接示波器端口。

标注“11、13”是调节 A/B 通道的幅度的点位器。

另外 0832 左边上是分别有个跳线帽，是滤波选择，如跳下是无滤波，跳上是有滤波。



### 2.3.12 高速 A/D 和双通道 DA 模块

双通道高速并行 DAC/ADC 模块。最高 180MHz 转换时钟率双路超高速 10 位 DAC (DAC900)、50MHz 单通道超高速 8 位 ADC (5540)、300MHz 高速单运放 2 个。由于速度很高，通常只适用于 FPGA 来接口控制，不适合单片机接口。原理图请参考文件夹“原理图\AD2C.pdf 和 ADDA\_SCH.pdf”。参考示例：

/DEMO/EDA\_DEMO/MODU\_DEMO/DEMO12\_Hi\_Seed\_DAC

标注“1”是 AD5540 数据数据输出端口，共 8 位。

标注“2”是 DA5651B 通道的输入端口，共 10 位，其中 DB2-DA9 数据脚号在板的左上方标出，

标注“3”是 AD\DA 的时钟输入端口及 DA5651A 通道，在右上方标注的端口名 DA0/1 是 DA 低两位输入端，DA5651B 通道 DB0/1 输入口，“ADCLK”表示 AD5540 时钟输入端，DABCLK 和 DAACLK 分别是 DA5651 时钟输入端，（注意：此 A/D 和 D/A 的时钟是通过 FPGA 的 IO 口输入）。

标注“4”是 DA5651A 通道高 8 位输入端。

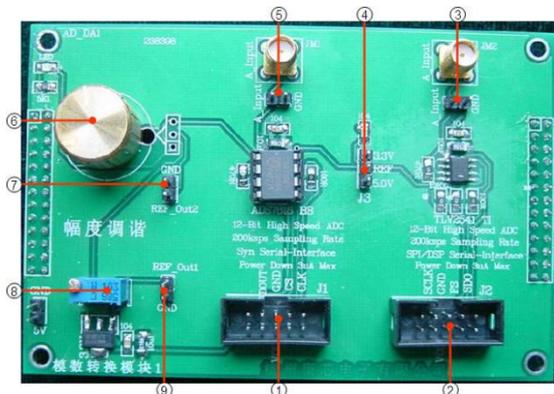
标注“5”是 DA 模拟信号输出接示波器探头端口。

标注“6”是 DA 运放的 $\pm 12V$ 的输入端，如果此板独立用，需从此端口输入 $\pm 12V$ ，注意：上端为 $-12V$ ，下方为 $+12V$ ，在此板上标注有错误。

标注“7”是 AD 以针形式的模拟信号输入端。

标注“8”是 AD 专用输入端，用此端口，可减少干扰信号。

标注 9、10 “分别是 DA 模拟信号幅度调谐点位器。

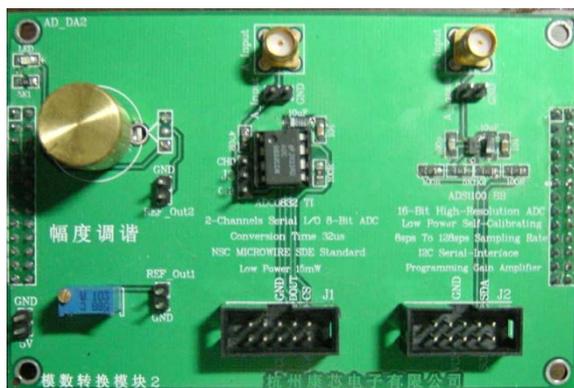


### 2.3.13 高速 12 位 SPI 串行双 ADC

左边器件高速 12 位同步串行 ADC ADS7816 模块，200ksps；体积小功耗低，无需高压电源。FPGA 和单片机都能将其作为接口扩展器件。ADS7816 详细资料在 MCU\_TECH\ADC\_DAC\ADS7816\_12B\_200K.pdf，

图右边器件是高速 12 位 SPI 串行 ADC TLV2541 模块，200ksps，SPI 接口；体积小功耗低，无需高压电源。FPGA 和单片机都能将其作为接口扩展器件。ADC TLV2541 详细资料在 MCU\_TECH\ADC\_DAC\TLV2541\_12B\_200K.pdf。

标注“1”是 ADS7816 控制端口。标注“2”是 ADC TLV2541 控制端口。标注“3、5”分别是此两器件的模拟输入专用及针形式端口，根据需要来选择。标注“4”是外围提供的模拟信号输出端。标注“6、7”分别是此板提供旋钮式模拟信号调谐及输出端之一。标注“8、9”分别是此板提供电位器式模拟信号调谐及输出端之一。此板参考示例程序及用法：\MCU\_TECH\_DEMO\MCU 控制 ADC\_DAC\_等器件实验指导。原理图请参考文件夹“原理图\AD\_DA1.pdf”。



### 2.3.14 8 位+16 位高分辨率 ADC 模块

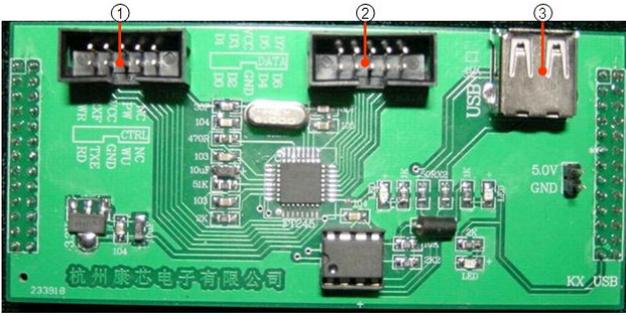
左边器件 ADC0832 二通道八位 ADC，SDE 标准串行通信接口，此器件详细资料在 MCU\_TECH\ADC\_DAC\ADC0832.pdf。右边器件 ADS1100 16 位高分辨率 ADC，低功耗、自动校正功能，I2C 串行接口。此器件详细资料在 MCU\_TECH\ADC\_DAC\ADS1100\_.pdf。此板上各端口功能请参照标注说明，基本功能一样。此板参考示例程序及用法：\MCU\_TECH\_DEMO\MCU 控制图 ADC\_DAC\_等器件实验指导。原理图请参考文件夹“原理图\AD\_DA2.pdf”。



### 2.3.15 SPI 串行接口高速 ADC+DAC 模块

左边器件高速串行 ADC TLV1572 10 位 QSPI/SPI/DSP 串行接口，1.25MSPS，自动功率控制；FPGA 和单片机都能将其作为接口扩展器件。

右边器件 DAC TLV5637，10 位 QSPI/SPI/DSP 串行接口高速 DAC，片内可编程参考电压，可编程转换速率控制。此模块更适用于基于 FPGA 的 DSP 模块设计开发。此器件详细资料在 \MCU\_TECH\ADC\_DAC\TLV5637.pdf。此板上各端口功能请参照板标注说明，基本功能一样。此板参考示例程序及用法：MCU\_TECH\_DEMO\MCU 控制 ADC\_DAC\_等器件实验指导。原理图请参考文件夹“原理图\AD\_DA3.pdf”。



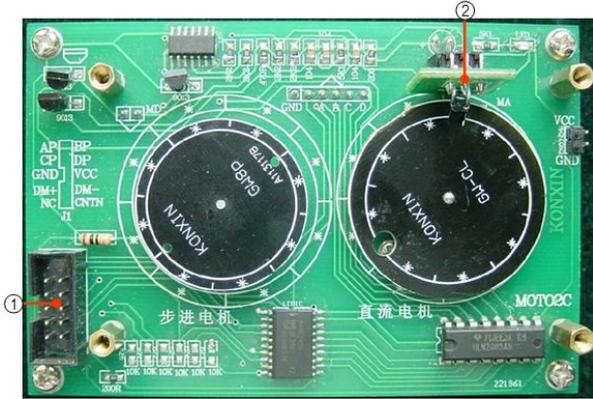
### 2.3.16 USB 接口模块

基于 FT245BM 的 USB 通信实验开发模块原理图请参考文件夹“原理图\ KX\_USB.pdf”。基于 FT245BM 的 USB 通信实验开发模块。示例如：

/DEMO/EDA\_DEMO/MODU\_DEMO/DEMO16\_USB\_FT245。软件及资料：文件夹\FT245BM\_USB 驱动。

标注“1”是 FT245 控制端口。标注“2”是 FT245

的数据口标注“3”是 USB 通信接口。



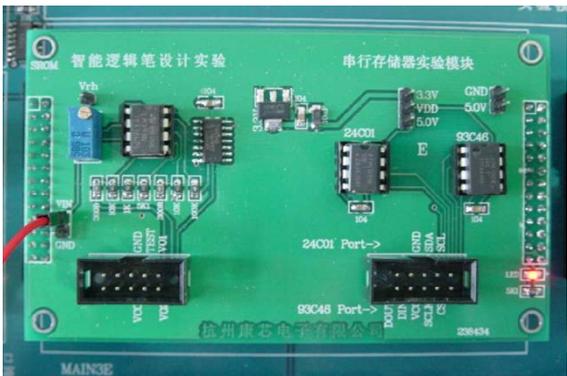
### 2.3.17 电机接口模块

电机模块包括步进电机和直流电机扩展模块。原理图请参考文件夹“原理图\ MOTO\_SCH.pdf”。示例如：

“/DEMO/EDA\_DEMO/MODU\_DEMO/DEMO6\_DC\_MOTO& DEMO7\_STEP\_MOTO&DEMO8\_FDIV\_STEP\_MOTO。

标注“1”是直流电机 DM+和 DM1 接口,步进电机 AP,BP,CP,DP 控制端口, CNTN 是直流电机计数端。

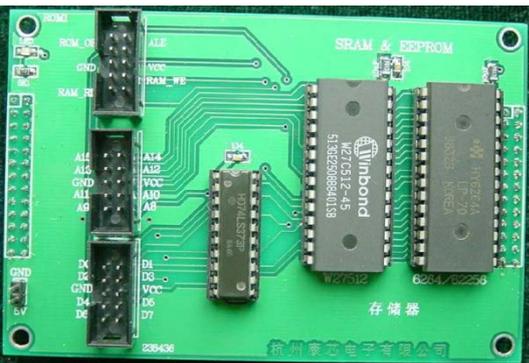
标注“2”是直流电机转动圈数红外接收模块。



### 2.3.18 双串行存储器/逻辑笔设计模块

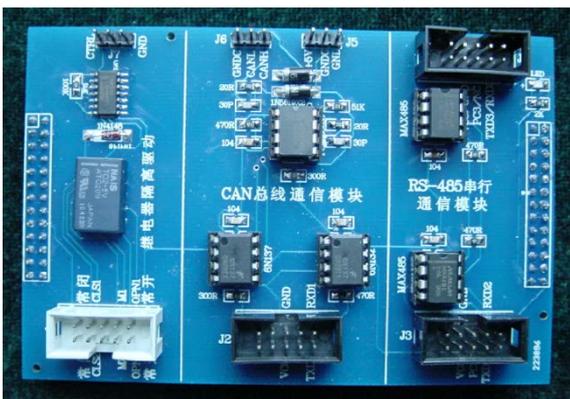
左边是智能逻辑笔设计实验用的逻辑笔实验模块。详细用法可参考相关示例。示例如：

/DEMO/EDA\_DEMO/MODU\_DEMO/DEMO18\_LOGIC\_PEN。右边是 93C46 和 24C01 串行存储器,下方是控制端口标注。此模块的原理图参考：“原理图\ SROM.pdf”。



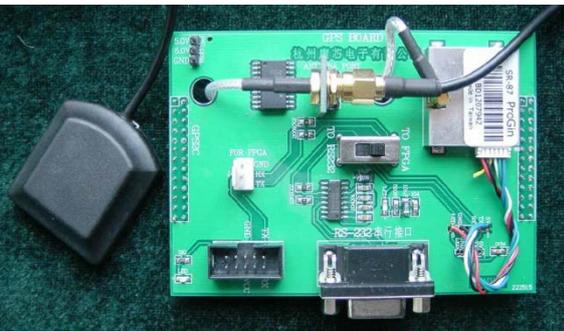
### 2.3.19 SRAM/EEPROM 模块

此模块含 SRAM/EPROM 扩展模块,左边是控制和数据端口原理图参考：“原理图\ ROM1.pdf”。



### 2.3.20 继电器/CAN/RS485 总线模块

此模块分三部分,最左边是继电器模块,中间是 CAN 总线模块,最右边 RS485 串口模块。原理图参考：“原理图\ RS485\_CAN\_SCH.pdf”。



### 2.3.21 GPS 实验开发模块

含 ProGin SR87 GPS 模块，最高 9600 波特率的 GPS 模块的串行接口 GPS 实验开发模块。ProGin SR87 资料参考：\VIVI\_FILE\GPS 文件，原理图参考：“原理图\GPS2C.pdf”。参考示例：

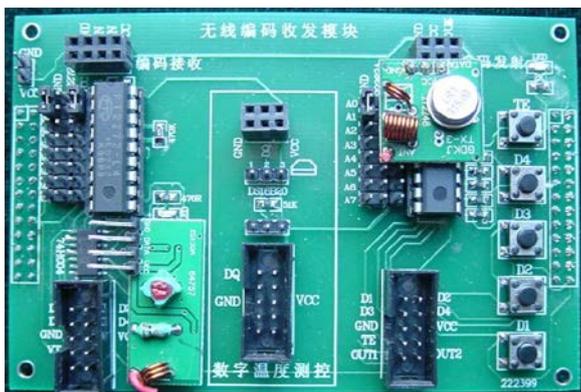
/DEMO/8051Core\_DEMO/DEMO7\_8051Core\_GPS



### 2.3.22 看门狗定时器+时钟日历模块

左边是看门狗定时器芯片是 X5040(含上电复位控制、看门狗定时器、降压管理和块保护功能串行 EEPROM 四模块；X5040 资料在\MCU\_TECH\ADC\_DAC\x5045cn.pdf。

右边时钟日历芯片是 DS1302, 实时年月周日时分秒计时功能, 串口数据通信, 掉电保护模块等。DS1302 资料在\MCU\_TECH\ADC\_DAC\DS1302 原理及程序说明.doc。此二模块都可由基于 FPGA 的状态机控制也可用单片机控制。此板参考示例程序及用法：\MCU\_TECH\_DEMO\MCU 控制 ADC\_DAC\_ 等器件实验指导。原理图参考：“原理图\WDG1.pdf”。



### 2.3.23 无线编码收发+数字温度传感器模块

此模块是基于 PT2272 和 PT2262 的数字编码无线通信收发模块, 和基于 DS18B20 的数字温度传感器模块两部分。其中最左边是接收模块, 最右边是发送模块, 测试方法, 可同时按动 TE 键和 D1-D4 的任意键, 如果对应的接收模块的对应的小灯亮, 表示接收成功。中间部分是数字温度传感器模块。原理图参考：“原理图/RADIO\_TRANS\_RESV\_SCH.pdf”。参考示例：

DEMO\_MODU/DEMO19\_WRIELESS\_CORD&DEMO/8051Core\_DEMO/DEMO14\_8051Core\_DS18B20

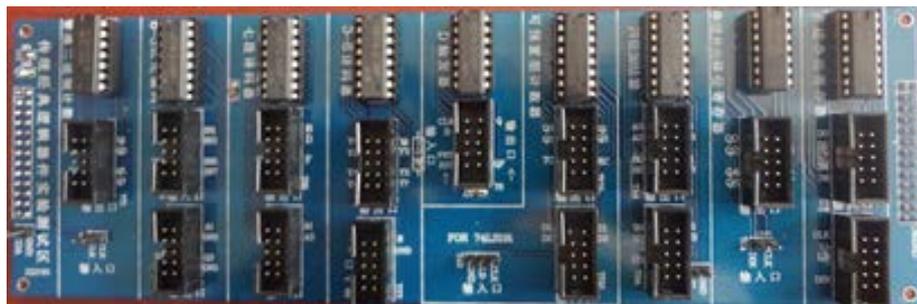


### 2.3.24、摄像头模块

OV7670 是 30 万像素 CMOS 摄像头传感器, 可以输出 640×480@30fps 视频图像, 即 VGA@30fps, 输出颜色支持 RGB565 和 YUV 等格式, 接插形式用一个转接板直接插在 FPGA 板上端口, 具体提供电子文档说明和实验。

参考示例：DEMO\_MODU/DEMO\_CAM\_LCD

### 2.2.25、传统数字电路模块, 提供部分 74 系列实验器件及接口



提供部分传统 74 系列器件, 具体在每个器件下方已标出型号, 接插方式在器右方利用十芯口把每个器件的引脚都已针形式接出, 引脚名称也在其边上标出, 一一对应就是, 可以用十芯线或单线外接。

### 2.3.26、基于 5200 网口模块

基于 W5200 控制的以太网卡，可调制 10-100M

资料及例程路径在：/DEMO/8051Core\_DEMO

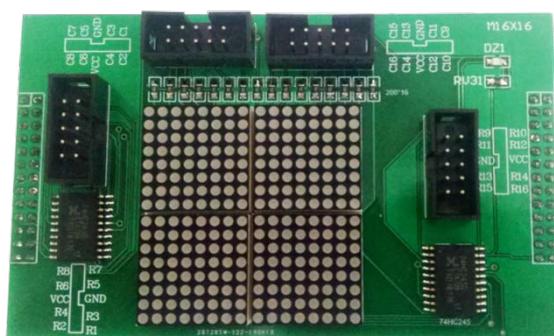
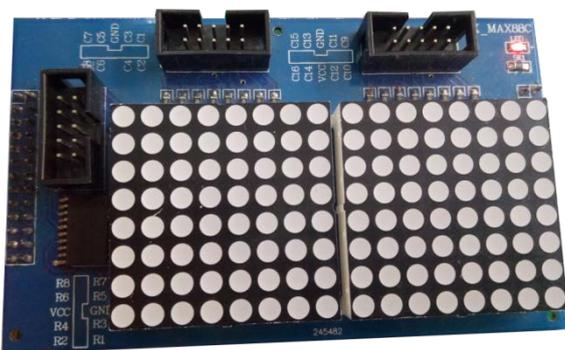
\DEMO11\_Basic8051\_W5200



### 2.3.27、WIFI+超声波模块

WIFI 由 XLW210A 控制的无线网卡，超声波由 US100 控制测距模块。资料及例程路径在：

/DEMO/8051Core\_DEMO/DEMO13\_Basic8051\_R2WiFi\_XLW210A



### 2.3.28、点阵模块

2 组 8\*8 点阵，构成 8\*16 和 16\*16 点阵模块。示例：DEMO\_MODU/DEMO14\_seg70



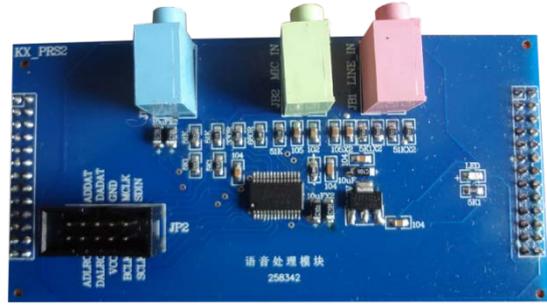
### 2.3.29 、12 位 AD/DA 模块

并行采样速率 50MHz 12 位 AD807+转换速率 165MHz 12 位模块

示例：DEMO\_MODU/12 bit\_adda

### 2.3.30 、语音处理模块

LINE OUT/LIN IN/MIC IN 接口，采用 WM8731S 语音处理芯片模块



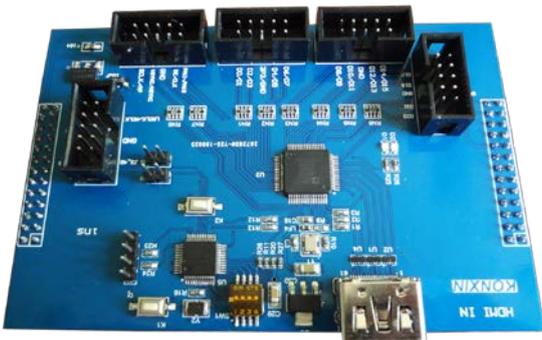
### 2.3.31 、SD+2 个 PS/2+RS232+VGA 模块

含三色 VGA、PS/2 键盘和鼠标接口、RS232 接口、SD 卡接口。



### 2.3.32 、 HDMI 输入模块

ADV7611 控制模式输入，分辨率无需配置，减低开发难度，详见配套例程资料。



### 2.3.33 、 HDMI 输出模块

Si19022 控制模式输出，分辨率无需配置，减低开发难度，详见配套例程资料。



## 2.4、FPGA 核心板扩展至康芯主系统引脚对照表（模块类）

（此表是针对扩展板提供的 IO 口查找表，根据主系统上扩展的十芯或十四芯口的标号对应查找引脚号）

	主系统 扩展标号	KX-10CE55 板	KX-4CE55 板	KX- 4C6/10E 板	DE0 板	DE0-CV 板	DE1-SOC 板	DE10-nane 板	模式 查找号
1	CLKB0	W22	W22	90	AB12	N16	AC18	V12	
2	CLKB1	W21	W21	91	AA12	M16	AD17	W12	
3	DBT0	AA10	AA10	98	AB3	M22	AA18	D12	
4	DBT1	AA8	AA8	86	AA3	L22	AE17	C12	
5	DB0	Y22	Y22	133	AB16	B16	Y17	E8	

## KONXIN

6	DB1	Y21	Y21	129	AA16	C16	Y18	D11	
7	DB2	AA21	AA21	127	AA15	D17	AK16	D8	PE0(51核)
8	DB3	AA20	AA20	126	AB15	K20	AK18	AH13	
9	DB4	W20	W20	124	AA14	K21	AK19	AF7	PE2(51核)
10	DB5	AB20	AB20	121	AB14	K22	AJ19	AH15	
11	DB6	AA18	AA17	119	AB13	M20	AJ17	AF4	
12	DB7	AB4	AB17	115	AA13	M21	AJ16	AH3	
13	DB8	V16	V16	111	AB10	N21	AH18	AD5	
14	DB9	U16	U16	112	AA10	R22	AH17	AG14	
15	DB10	AA14	AA14	106	AB8	R21	AG16	AE23	
16	DB11	AB15	AB15	110	AA8	T22	AE16	AE6	
17	DB12	Y13	Y13	103	AB5	N20	AF16	AD23	
18	DB13	AB13	AB13	104	AA5	N19	AG17	AE24	
19	DB14	AB10	AB10	99	AB4	P19	AA19	AD20	
20	DB15	AB8	AB8	87	AA4	P17	AC20	AD17	
21	DB16	W8	W8	83	V14	P16	AH19	AC23	PIO13
22	DB17	AB6	AB6	84	U14	M18	AJ20	AC22	PIO7
23	DB18	W6	W6	77	Y13	L18	AH20	Y19	PIO12
24	DB19	Y7	Y7	79	W13	L17	AK21	AB23	PIO6
25	DB20	AB5	AB5	74	U13	L19	AD19	AA19	PIO11
26	DB21	AA6	AA6	75	V12	K17	AD20	W11	PIO5
27	DB22	AA3	AA3	70	R10	K19	AE18	AA18	PIO10
28	DB23	AB3	AB3	67	V11	P18	AE19	W14	PIO4
29	DB24	W2	W2	65	Y10	R15	AF20	Y18	PIO9
30	DB25	Y1	Y1	66	W10	R17	AF21	Y17	PIO3
31	DB26	U2	U2	60	T8	R16	AF19	AB25	PIO8
32	DB27	V1	V1	64	V8	T20	AG21	AB26	PIO2
33	DB28	P2	P2	58	W7	T19	AF18	Y11	PIO48
34	DB29	R1	R1	55	W6	T18	AG20	AA26	PIO1
35	DB30	M2	M2	53	V5	T17	AG18	AA13	PIO49
36	DB31	N1	N1	52	U7	T15	AJ21	AA11	PIO0
37	CLKA0	V22	V22	88	AB11	H16	AB17	AE12	MT(51核)
38	CLKA1	V21	V21	89	AA11	H15	AB21	AE20	NO(51核)
39	DAT0	Y10	Y10	100	R16	H14	AK27	AG21	PIO33
40	DAT1	AA9	AA9	80	T16	J13	AK26	AH22	PIO31
41	DA0	U22	U22	144	AA20	A12	AA21	AE19	PIO47
42	DA1	U21	U21	143	AB20	B12	AC23	AG15	PIO46
43	DA2	AB19	AB19	142	AA19	A13	AD24	AF20	PIO45
	主系统 扩展标号	KX-10CE55 板	KX-4CE55 板	KX- 4C6/10E 板	DE0 板	DE0-CV 板	DE1-SOC 板	DE10-nane 板	模式 查找号
44	DA3	AA19	AA19	141	AB19	B13	AE23	AF18	PIO44
45	DA4	AB18	AB18	138	AB18	C13	AE24	AH18	PIO43
46	DA5	U20	U20	137	AA18	D13	AF25	AG18	PIO42
47	DA6	AA16	AA16	136	AA17	G18	AF26	AH19	PIO41
48	DA7	AB16	AB16	135	AB17	G17	AG25	AG19	PIO40
49	DA8	Y17	Y17	125	Y17	H18	AG26	AF21	PIO39
50	DA9	W17	W17	128	W17	J18	AH24	AG20	PIO38

## KONXIN

51	DA10	AA15	AA15	114	U15	J19	AH27	AE22	PIO37
52	DA11	T16	T16	120	T15	G11	AJ27	AF22	PIO36
53	DA12	AA13	AA13	105	W15	H10	AK29	AA20	PIO35
54	DA13	AB14	AB14	113	V15	J11	AK28	AH23	PIO34
55	DA14	V11	V11	101	AB9	A15	AJ26	AH21	PIO32
56	DA15	AB9	AB9	85	AA9	L8	AH25	AG24	PIO30
57	DA16	AA7	AA7	73	AA7	A14	AJ25	AF23	PIO29
58	DA17	AB7	AB7	76	AB7	B15	AJ24	AG23	PIO28
59	DA18	W7	W7	71	T14	C15	AK24	AF25	PIO27
60	DA19	Y8	Y8	72	R14	E14	AG23	AH24	PIO26
61	DA20	V6	V6	68	U12	E15	AK23	AH26	PIO25
62	DA21	Y6	Y6	69	T12	E16	AH23	AG25	PIO24
63	DA22	AA4	AA4	54	R11	F14	AK22	AH27	PIO23
64	DA23	AA5	AA5	59	R12	F15	AJ22	AG26	PIO22
65	DA24	Y2	Y2	50	U10	F13	AH22	AF27	PIO21
66	DA25	Y14	AA1	51	T10	F12	AG22	AE25	PIO20
67	DA26	V2	V2	46	U9	G16	AF24	AF28	PIO19
68	DA27	W1	W1	49	T9	G15	AF23	AG28	PIO18
69	DA28	R2	R2	43	Y7	G13	AE22	AD26	PIO17
70	DA29	U1	U1	44	U8	G12	AD21	AA15	PIO16
71	DA30	N2	N2	39	V6	J17	AA20	AC24	PIO15
72	DA31	P1	P1	42	V7	K16	AC22	Y5	PIO14

## 第三章 软硬件操作流程

### 3.1 十进制计数器流程（参照《EDA 技术实用教程》第六版）

#### 3.1.1 建立工作库文件夹和编辑设计文件

任何一项设计都是一项工程（Project），都必须首先为此工程建立一个文件夹用来放置与此工程相关的所有设计文件。此文件夹将被 EDA 软件默认为工作库（Work Library）。一般地，不同的设计项目最好放在不同的文件夹中，而同一工程的所有文件都放在同一文件夹中。注意不要将工程文件夹设在已有的安装目录中，也

不要建立在桌面上。在建立了文件夹后就可以将设计文件通过 Quartus II 的文本编辑器编辑并存盘了，步骤如下：

(1) 设本项设计的文件夹名为 MY\_PROJECT，在 D 盘中，路径为 D:\MY\_PROJECT。

(2) 输入源程序，初学者需直接复制文本，请参考光盘文件夹“MY\_PROJECT”，打开后可直接复制。打开 Quartus II，选择 File→New 命令。在 New 窗口中的 Design Files 栏选择编译文件的语言类型，这里选择 VHDL File 选项，如图 3-1 所示或 Verilog HDL File 选项 如图 3-2。然后在 VHDL/Verilog HDL 文本编译窗口中输入例 3-1 程序。

例 3-1

VHDL:

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY CNT10 IS
    PORT (CLK,RST,EN,LOAD : IN STD_LOGIC;
          DATA : IN STD_LOGIC_VECTOR(3 DOWNTO 0); --4 位预置数
          DOUT : OUT STD_LOGIC_VECTOR(3 DOWNTO 0); --计数值输出
          COUT : OUT STD_LOGIC); --计数进位输出
END CNT10;
ARCHITECTURE behav OF CNT10 IS
BEGIN
    PROCESS(CLK, RST, EN, LOAD)
        VARIABLE Q : STD_LOGIC_VECTOR(3 DOWNTO 0);
    BEGIN
        IF RST='0' THEN Q := (OTHERS=>'0'); --复位低电平时，计数寄存器清 0
        ELSIF CLK'EVENT AND CLK='1' THEN --测试时钟上升沿
            IF EN='1' THEN --计数使能高电平，允许计数
                IF (LOAD='0') THEN Q := DATA; ELSE --预置控制低电平，允许加载
                    IF Q<9 THEN Q := Q + 1; --计数小于 9，继续累加
                    ELSE Q := (OTHERS=>'0'); --否则计数清 0
                END IF;
            END IF;
        END IF;
    END IF;
    IF Q="1001" THEN COUT<='1'; --当计数为 9 时，进位输出 1
    ELSE COUT<='0'; END IF; --否则进位输出 0
    DOUT <= Q; --计数寄存器的值输出端口
END PROCESS;
END behav;
```

Verilog HDL::

```
module CNT10 (CLK,RST,EN,LOAD,COUT,DOUT,DATA);
    input CLK,EN,RST,LOAD ; //时钟，时钟使能，复位，数据加载控制信号
    input [3:0] DATA ; //4 位并行加载数据
    output [3:0] DOUT ; //4 位计数输出
    output COUT ; //计数进位输出
    reg [3:0] Q1 ; reg COUT ;
    assign DOUT = Q1; //将内部寄存器的计数结果输出至 DOUT
    always @(posedge CLK or negedge RST) //时序过程
    begin
        if (!RST) Q1 <= 0; //RST=0 时，对内部寄存器单元异步清零
        else if (EN) begin //同步使能 EN=1，则允许加载或计数
            if (!LOAD) Q1<=DATA; //当 LOAD=0，向内部寄存器加载数据
            else if (Q1<9) Q1 <= Q1+1; //当 Q1 小于 9 时，允许累加
        end
    end
endmodule
```

(3) 文件存盘。选择 File→Save As 命令，找到已设立的文件夹 D:\MY\_PROJECT，存盘文件名应该与实体名一致，即 CNT10.vhd 或 CNT10.v。点击“保存”后将出现问句“Do you want to create a new project with this file?”，若单击“Yes”按钮，则直接进入创建工程的流程；若单击“No”按钮，可按以下方法进入创建工程流程。这里不妨选“No”。

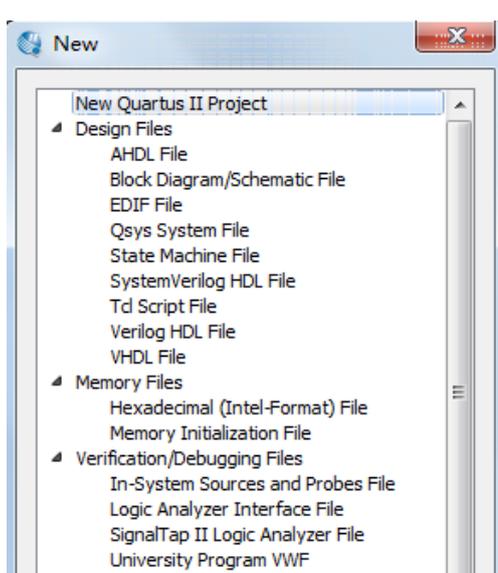


图 4-1 选择编辑文件类型 VHDL

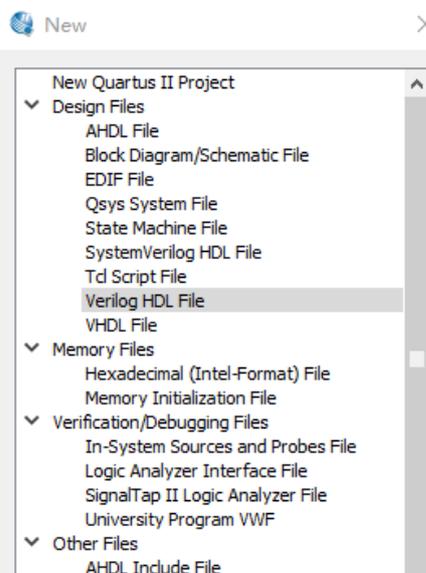


图 4-2 选择编辑文件类型 Verilog HDL

### 3.1.2 创建工程

在此要利用 New Project Wizard 工具选项创建此设计工程，即令 CNT10.vhd/v 为工程，并设定此工程的一些相关信息，如工程名、目标器件、综合器、仿真器等。步骤如下：

(1) 打开并建立新工程管理窗口。选择 File→New Project Wizard 命令，即弹出设置窗口，如图 3-3 所示。单击此对话框第二栏右侧的“...”按钮，找到文件夹 D:\MY\_PROJECT，选中已存盘的文件 CNT10.vhd/v，再单击“打开”按钮，即出现如图 4-3 所示的设置情况。其中第一行的 D:\MY\_PROJECT/表示工程所在的工作库文件夹；第二行的 CNT10 表示此项工程的工程名，工程名可以取任何其他的名，也可直接用顶层文件的实体名作为工程名。在此就是按这种方式取的名；第三栏是当前工程顶层文件的实体名，这里即为 CNT10。

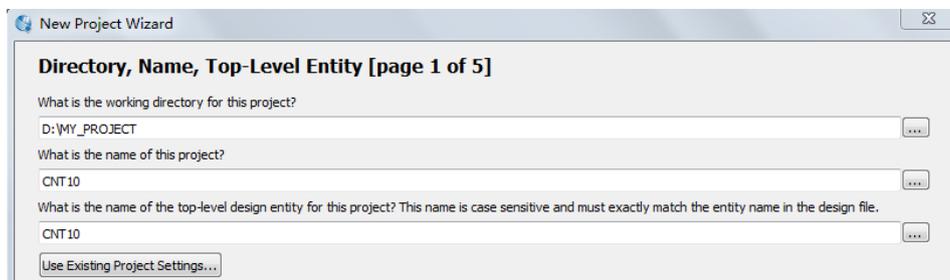


图 3-3 利用 New Project Wizard 创建工程 CNT10

(2) 将设计文件加入工程中。单击 Next 按钮，在弹出的对话框中单击 File 栏后的按钮，将与工程相关的所有 VHDL 文件都加入此工程。

(3) 选择目标芯片。单击 Next 按钮，选择目标器件。首先在 Device Family 下拉列表框中选择芯片系列，在此选 Cyclone IV E 系列（即 Cyclone 4E 系列）。设选择此系列的具体芯片名是 EP4CE55F23C8（系统：KX-CDS4S）。这里 EP4CE55 表示 Cyclone IV E 系列及此器件的逻辑规模，F 表示芯片是 FBGA 封装，C8 表示速度级别。便捷的方法是通过如图 3-4 所示的窗口右边的 3 个下拉列表框选择过滤条件，分别选择 Package 为 FBGA、

Pin Count 为 484（即此芯片的引脚数量是 484 只）和 Speed grade 为 8。我们提供多种不同系列型号 FPGA，根据目标芯片不同，在这里选择对应型号，如：

KX-CD4， 对应 FPGA 型号为：Cyclone IV EP4CE6/10E23C8N(两款不同资源芯片兼容)；

KX-CD3ES， 对应 FPGA 型号为：Cyclone III EP3CE16F484C6N(核心板 DE0)；

KX-CD5V， 对应 FPGA 型号为：Cyclone V EP5CEBA4F23C7N(核心板 DE0-CV)；

KX-CD5S， 对应 FPGA 型号为：Cyclone V EP5CSEMA5F31C6N(核心板 DE1-SOC)；

KX-CD10S， 对应 FPGA 型号为：Cyclone 10 10CL55WF484C8(核心板 KX-10C55)；

在这里我们以 KX-CDS4S 系统，FPGA 对应型号 EP4CE55F23C8 为例。

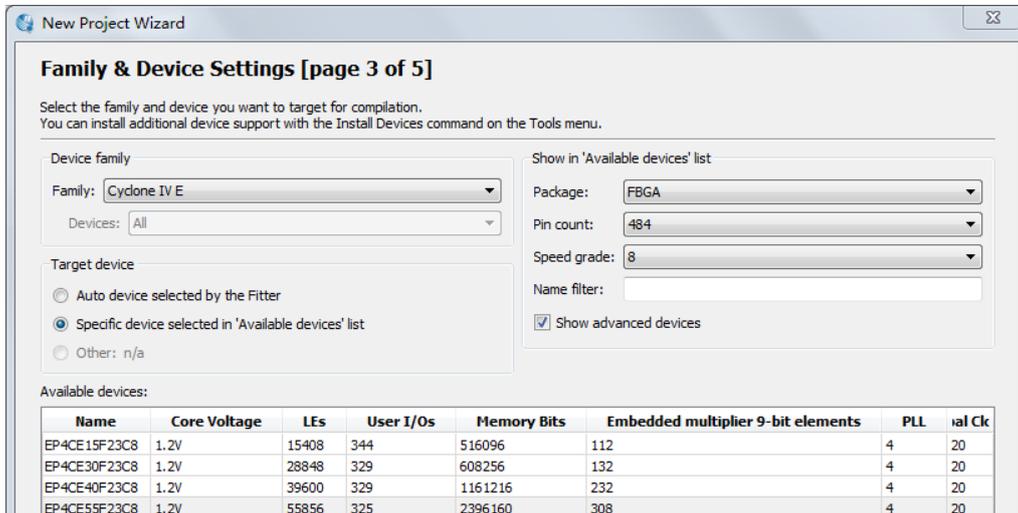


图 3-4 选择目标器件 EP4CE55F23C8

(4) 工具设置。单击 Next 按钮后，弹出的下一个窗口是 EDA 工具设置窗口——EDA Tool Settings（图 3-5）。此窗口有 3 项选择，其它选择默认，即选择自带的工具，而 Simulation 栏，选择仿真工具：ModelSim-Altera，Format（格式）栏选择 VHDL。

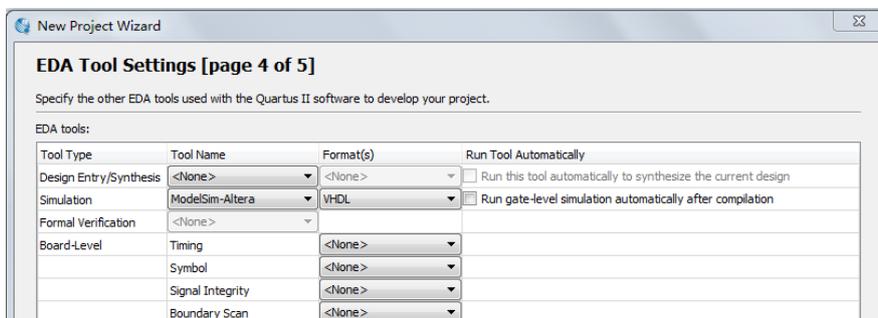


图 3-5 设计与验证工具软件选择

(5) 结束设置。单击 Next 按钮后即弹出工程设置统计窗口，上面列出了此项工程相关设置情况。最后单击 Finish 按钮，即已设定好此工程，并出现 CNT10 的工程管理窗口，或称 Compilation Hierarchies 窗口，主要显示本工程项目的层次结构和实体名。

Quartus II 将工程信息存储在工程配置文件（quartus）中。它包含有关 Quartus II 工程的所有信息，包括设计文件、波形文件、内部存储器初始化文件等，以及构成工程的编译器、仿真器和软件构建设置。

### 3.1.3 约束项目设置

在对工程进行编译处理前，必须给予必要的设置和约束条件，以便使设计结果满足工程要求。主要步骤如下。

(1) 选择编译约束条件。选择 Assignments→Settings，进入如图 3-6 所示对话框。在 Category 栏可以进行多项选择，如确认 VHDL 版本（默认 VHDL 1993）、布线布局方式、适配努力程度、内嵌逻辑分析仪使能、仿真文件和仿真方式确定；或在此对话框选择 Analysis & Synthesis Settings 项，并点击 More Settings 按钮，进入更多可供综合与适配控制的选项栏（图 4-6 右侧）。

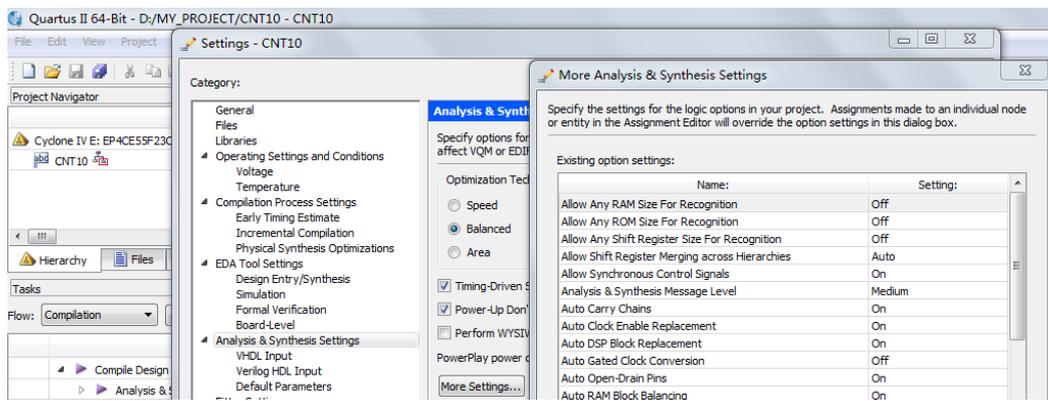


图 3-6 选择编译综合的工作方式

(2) 选择目标芯片的其它控制项。选择 Assignments→Device (如图 3-7 左侧所示), 然后选择目标芯片为 EP4CE55F23C8 (此芯片已在以上建立工程时选定了)。

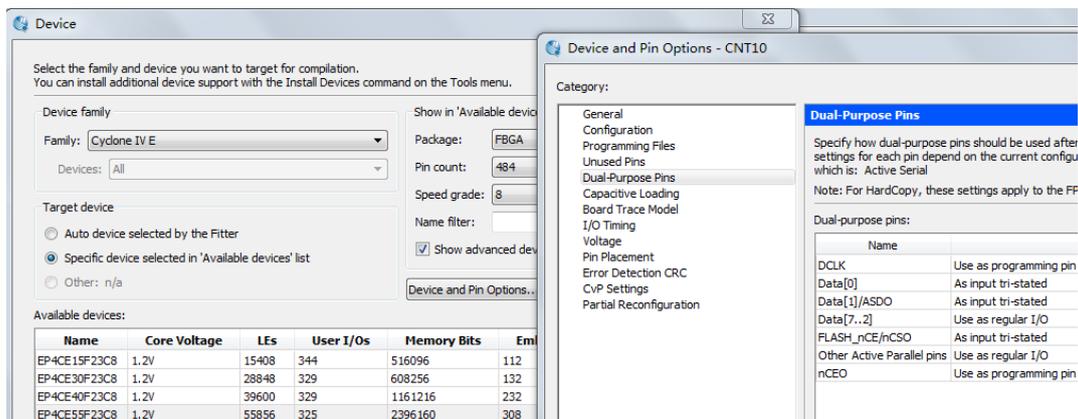


图 3-7 选择目标器件和工作方式

(3) 选择配置器件的工作方式。于如图 3-7 的 Device 对话框中, 单击 Device and Pin Options 按钮后, 在弹出的窗口中, 选择配置器件、编程方式和工作方式等。如果希望对编程配置文件能在压缩后下载进配置器件中, 可在编译前做好设置。这对 FPGA 的专用 Flash 配置存储器的编程设置很重要, 它将确保基于 FPGA 的数字系统在脱离计算机后能稳定独立地工作。注意, 窗口下方将随项目名而显示对应的帮助说明文字, 用户可随时参考。

(3) 选择目标器件引脚端口状态。例如, 选择图 3-7 所示窗口中的无用引脚 Unused Pins 选项, 可根据实际需要选择目标器件闲置引脚的状态, 如可选择为输入状态呈高阻态 (推荐此项选择), 或输出状态 (呈低电平), 或输出不定状态, 或不作任何选择。在其他选项中也可作一些选择, 各选项的功能可参考窗口下的 Description 说明。

例如对双功能引脚进行设置。选择图 3-7 所示窗口中的 Dual-Purpose Pins 选项, 对必要的引脚进行选择, 如选择 nCEO 为 Use as regular I/O。即选择 nCEO 脚当做普通 I/O 脚来使用 (对于引脚不够用时可以这样选, 通常可以不动它)。

### 3.1.4 全程综合与编译

Quartus II 编译器是由一系列处理工具模块构成的, 这些模块负责对设计项目的检错、逻辑综合、结构综合、输出结果的编辑配置, 以及时序分析等。在这一过程中, 将设计项目适配到 FPGA 目标器件中, 同时产生多种用途的输出文件, 如功能和时序信息文件、器件编程的目标文件等。编译器首先检查出工程设计文件中可能的错误信息, 以供设计者排除, 然后产生一个结构化的以网表文件表达的文件。

在编译前, 设计者可以通过各种不同的设置和约束选择, 指导编译器使用各种不同的综合和适配技术 (如时序驱动技术、增量编译技术、逻辑锁定技术等), 以便提高设计项目的工作速度, 优化器件的资源利用率。而且在编译过程中及编译完成后, 可以从编译报告窗口中获得所有相关的详细编译信息, 以利于设计者及时调整设计方案。

在设计文件的编辑输入、创建工程和约束设置后, 就要在 Quartus 平台上进行编译了。开始编译前首先选择 Processing→Start Compilation 命令, 启动全程编译 (如图 3-8 所示)。这里所谓的全程编译 (Compilation) 包括

以上提到的 Quartus II 对设计输入的多项处理操作，其中包括输入文件的排错、数据网表文件提取、逻辑综合、适配、装配文件（仿真文件与编程配置文件）生成，以及基于目标器件的工程时序分析等。

编译过程中要注意工程管理窗下方的 Processing 处理栏中的编译信息。如果工程中的文件有错误，启动编译后，在下方的 Processing 栏中会显示出来。对于 Processing 栏显示出的语句格式错误，可双击此条文，即弹出对应的 VHDL 文件，在深色标记条附近有文件错误所在，改错后再次进行编译直至排除所有错误。

如果发现报出多条错误信息，每次只需要检查和纠正最上面报出的错误即可。因为许多情况下，是由于某一种错误导致了多条错误信息报告。

若编译成功，可见到图 3-8 所示的工程管理窗口的左上角显示了工程 CNT10 的层次结构和其中结构模块耗用的逻辑宏单元数；在此栏下是编译处理流程，包括数据网表建立、逻辑综合、适配、配置文件装配和时序分析等。最下是编译处理信息；中栏（Compilation Report 栏）是编译报告项目选择菜单，单击其中各项可以详细了解编译与分析结果。例如选择 Flow Summary 选项，将在右栏显示硬件耗用统计报告，其中报告了当前工程耗用了 9 个逻辑宏单元（Logic Cell）、4 个专用寄存器（Register）、0 个内部 RAM 位等。

如果单击 TimeQuest Timing Analyzer 选项的“+”图标，则能通过单击其下列出的各项目，看到当前工程所有相关时序特性报告。

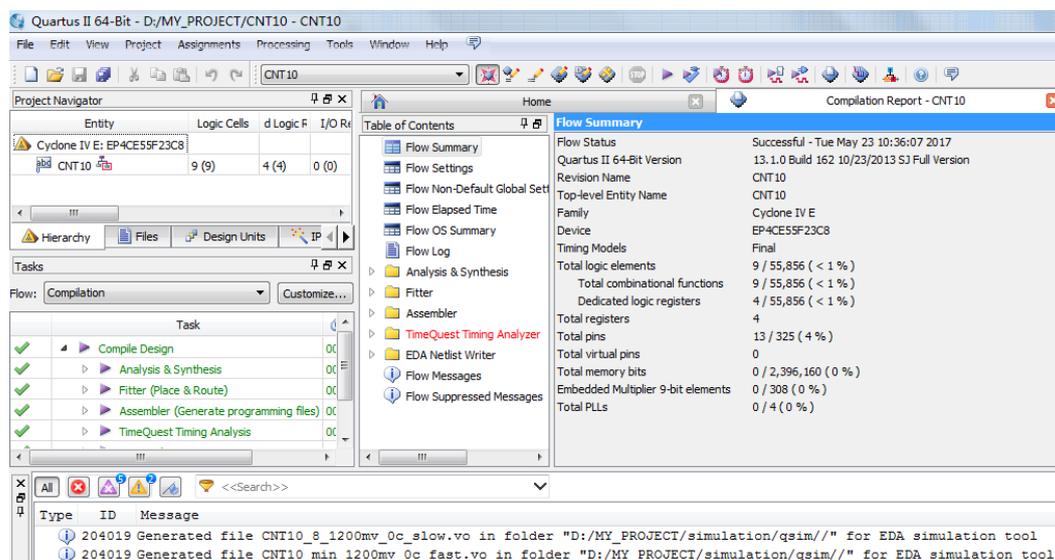


图 3-8 全程编译无错后的报告信息

如果单击 Fitter 选项的“+”图标，则能通过单击其下列出的各项标题，看到当前工程所有相关硬件特性适配报告，如其中的 Floorplan View，可观察此项工程在 FPGA 器件中逻辑单元的分布情况和使用情况。需要特别提醒两点。

(1) 应该时刻关注图 3-8 所示的工程管理窗口左上角的路径指示和工程名。它指示的是当前处理的一切内容皆为此路径文件夹中的工程，而不是其他任何文件。

(2) 若编译能无错通过，甚至也有 RTL 电路产生，但仿真波形就是不对，硬件功能也出不来。对此，不能一味地靠软件排错，必须仔细检查 Quartus II 的各项设置的正确性；此外对 Processing 栏中显示的编译处理信息中的 Warning 和 Critical Warning 警告信息要仔细阅读，不要放过，问题可能就在此处。

### 3.1.5 RTL 图观察器应用

Quartus II 可实现硬件描述语言或网表文件（Quartus II 网表文件格式包括：VHDL、Verilog、BDF、TDF、EDIF、VQM）对应的 RTL 电路图的生成。方法如下：

选择 Tools→Netlist Viewers 命令，在出现的下拉菜单中有 3 个选项：RTL Viewer，即 HDL 的 RTL 级图形观察器；Technology Map Viewer，即 HDL 对应的 FPGA 底层门级布局观察器；State Machine Viewer，即 HDL 对应状态机的状态图观察器。

选择第一项，可以打开 CNT10 工程的 RTL 电路图（图 3-9）。再双击图形中有关模块或选择左侧各项，还可逐层了解各层次的电路结构。

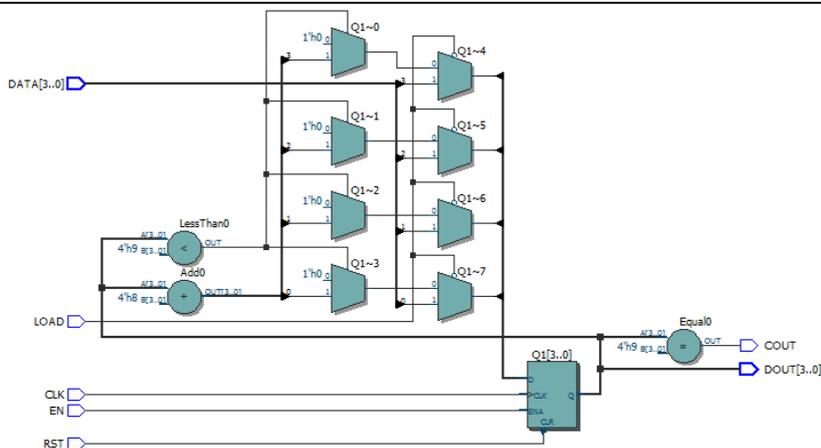


图 3-9 CNT10 工程的 RTL 图

### 3.2 时序仿真

当前的工程编译通过后，必须对其功能和时序性质进行仿真测试，以了解设计结果是否满足原设计要求。这可以用针对逻辑电路的仿真软件来完成。仿真软件主要有两类：第一类是由 FPGA 供应商自己推出的仿真软件，如 Altera 公司的 Quartus II 中自带的门级波形仿真软件，此类软件针对性强，易学易用，缺点是只能适合于小规模设计。Quartus II 10.0 版本后都已撤除了此类软件，此后只能直接使用第三方仿真软件 Modelsim 了；另一类是 EDA 专业仿真软件商提供的，所谓第三方仿真工具软件，如上提到的 Modelsim。在 Quartus II 的平台上使用第三方仿真软件 Modelsim 有多种方式，一种是直接使用，其详细用法《EDA 技术与 VHDL/verilog》第六版第 10 章介绍；另一种是面向大学计划的以间接形式的使用，即在新版 Quartus II 中将 Modelsim 整合成旧版门级波形仿真器那样，保持了原有的直观易用的特性，使得用户几乎可以使用原来早已熟悉的操作流程进行便利的仿真。

以下即给出基于 Modelsim 的波形仿真器的仿真流程详细步骤：

(1) 确认 Quartus II 中的仿真工具是否指向 Modelsim 所在路径。选择 Tools→Options，在 General 选择 EDA Tool Options，即出现如图 3-10 所示窗口。在此窗口最下的 Modelsim-Altera 栏，可以见到指向了安装软件的 Modelsim-ase 的路径（图 3-51）。此路径是安装 Quartus II 13.1 或 16.1 时自动加上去的：  
E:\altera\13.1\modelsim\_ase\win32aloem

(2) 打开波形编辑器。选择 File→New 命令，在 New 窗口（如图 3-1 所示）中选择 University Program VWF(即 Vector Waveform File)选项。单击 OK 按钮，即出现空白的 VWF 波形编辑窗，如图 3-11 所示，注意选择 View 中的 Full Screen 将窗口扩大，以利观察。但在启动编译时必须把窗口还原。

(3) 设置仿真时间区域。对于时序仿真来说，将仿真时间轴设置在一个合理的时间区域上十分重要。通常设置的时间范围可在数十微秒间。选择 Edit→Set End Time 命令，在弹出的窗口中的 End Time 栏可输入仿真时间，例如输入 55，单位为微秒（ $\mu\text{s}$ ，窗口中用 us 代替）。于是整个仿真域的时间即设定为 55 $\mu\text{s}$ ，如图 3-12 所示。

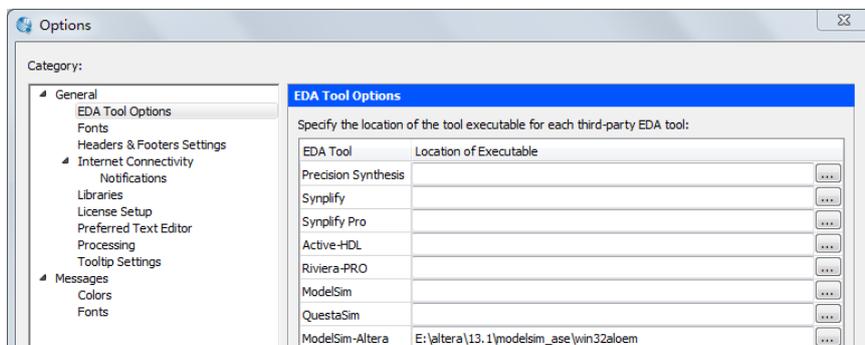


图 3-10 查看 Quartus 仿真工具指向 Modelsim 仿真软件的路径

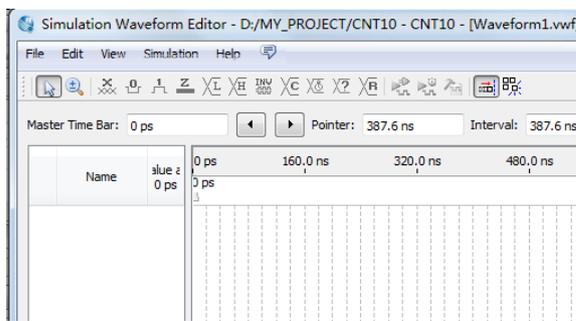


图 3-11 Vector Waveform File 文件编辑窗

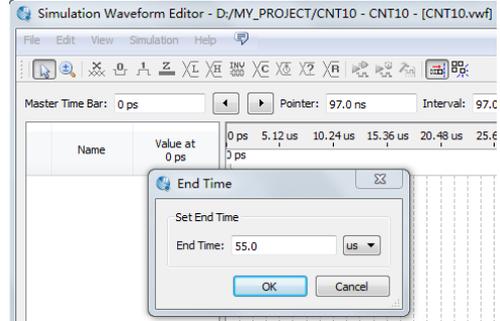


图 3-12 设置仿真时间长度

(4) 波形文件存盘。选择 File→Save As 命令，将波形文件存盘于 D:\MY\_PROJECT 中。例如，文件名可取为：CNT10.vwf。（我测试的是保持原名称不变）

(5) 将工程 CNT10 的端口信号节点选入波形编辑器中。方法是首先选择 Edit→Insert，将弹出 Insert Node or Bus 窗口（如图 3-13 所示）。在此窗口单击 Node Finder 按钮，进入 Node Finder 窗（如图 3-13 的右图所示）。在 Filter 下拉列表框中选“Pins: all”，然后单击 List 按钮，于是在左侧的 Nodes Found 窗口中出现设计中的 CNT10 工程的所有端口引脚名。注意，如果此对话框中的 List 不显示工程的端口引脚名，需重新编译一次，即选择 Processing→Start Compilation 命令，然后再重复以上操作过程。

最后选中仿真所需的重要的端口节点 CLK、EN、RST、DATA 于右侧 Selected Nodes 栏。单击 OK 按钮后，所有选中的信号被送到图 3-11 所示的波形编辑窗中。单击波形窗口左侧的“全屏显示”按钮，使全屏显示，并单击“放大缩小”按钮后，再于波形编辑区域右击，使仿真坐标处于适当位置。这时仿真时间横坐标设定在数十微秒数量级。

(6) 设置激励信号波形。获得选中的信号节点的波形编辑窗如图 3-14 所示。可以首先选择总线数据格式。例如，DOUT 的数据格式设置是这样的：若单击如图 3-14 所示的输入数据信号 DOUT 边的小三角，则能展开此总线中的所有信号；如果双击左边引脚符号，将弹出对该信号数据格式设置的 Node Properties 对话框（如图 3-14 所示）。在该对话框的 Radix 下拉列表框中有 4 种选择，这里可选择十六进制 Hexadecimal 表达方式。

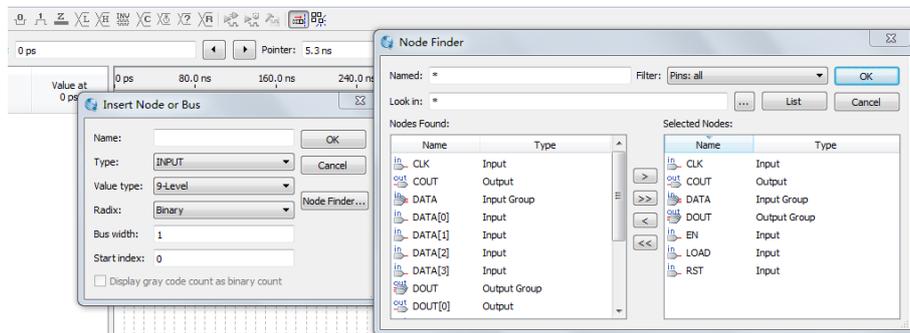


图 3-13 加入仿真需要的信号节点

时钟参数的设置方法是：单击图 3-15 所示窗口的时钟信号名 CLK，使之变成蓝色条，再单击选择左上行中的时钟设置按钮（小钟形按钮），在 Clock 对话框中设置 CLK 的时钟周期为 900ns（如图 3-15 所示）；Clock 窗口中的 Duty cycle 是占空比，默认为 50，即 50% 占空比。然后还要设置一系列输入信号（如 EN，LOAD，RST）的电平。

然后是编辑输入数据。由于 DATA 是 4 位待加载的输入数据，需要预先进行设置。用鼠标在图 3-16 所示信号名 DATA 的某一数据区拖拉出来一块蓝色区域，然后单击左侧工具栏的“？”按钮，在弹出的窗口输入数据，如 5。继而在不同区域设置不同数据。



图 3-14 设置总线数据格式

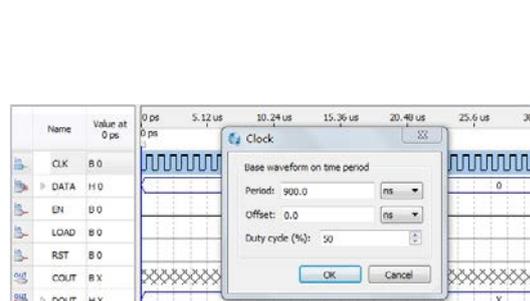


图 3-15 设置时钟参数

(7) 图 3-16 是最后设置好的.vwf 仿真激励波形文件图。最后对波形文件再次存盘。

(8) 启动仿真器。现在所有设置进行完毕,选择图 3-16 所示窗口上方的 Simulation→Run Timing Simulation,即启动仿真运算。

(9) 观察仿真结果。仿真波形文件 Simulation Report 通常会自动弹出,如图 3-17 所示。Quartus II 的仿真波形文件中,波形编辑文件(.vwf)与波形仿真报告文件(Simulation Report)是分开的,故有利于 Quartus II 从外部获得独立的仿真激励文件。

分析图 3-17 所示文件可以看出,其输入输出波形完全符合设计要求。波形图显示,当 EN 为高电平时允许计数;而 RST 有一个低电平脉冲后计数器被清零;初始值加载控制信号 LOAD 为高电平时允许计数,而当其为低电平时,在第一个时钟上升沿后,初始值 5 被加载于计数器,而当 LOAD 高电平后,计数器即以加载进的 5 开始计数。当计数到 9 时,进位输出信号 COUT 输出一个高电平信号,脉宽等于一个 CLK 周期。

需要特别指出的是,对于每次修改后的仿真波形文件,在再次启动时序仿真前必须先关闭修改好的文件,然后再打开这个文件后才能进行仿真,每次都必须这样做!

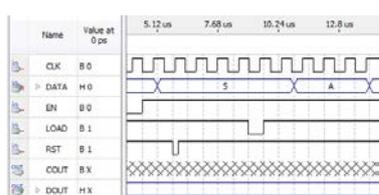


图 3-16 编辑好激励波形

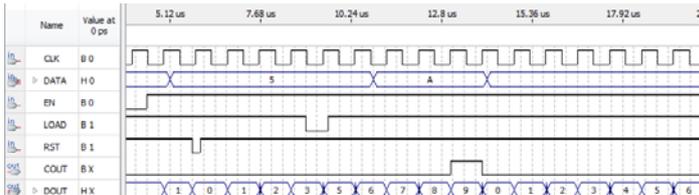


图 3-17 仿真输出的波形文件

### 3.3 硬件测试

为了能对此计数器进行硬件验证,应将其输入输出信号锁定在芯片确定的引脚上,编译下载。当硬件测试完成后,还必须对 FPGA 的配置芯片进行编程。

#### 3.3.1 引脚锁定

在 KX-CDS 系统完成此项示例实验,于是可以选择系统上的多功能重配置电路系统(具体在本讲义第二章-实验电路模式选择介绍)。当按动系统左侧的模式键选择电路模式时,将出现一系列实验电路,可以根据当前设计电路的具体情况选择一个实验电路。在此选择模式 0,对应的电路如图 3-18 所示,电路中的键 3 至键 8 的功能是电平控制键(第二章.“标注 13”)。设本次实验的核心板(插在 KX-CDS 系统上)是如 2-5 所示的 KX-4CE55 板,它上面的 FPGA 是 Cyclone 4E 型的 EP4CE55F23C8。所以可以用键 8、7、6、5 分别控制信号 CLK、EN、LOAD,和 RST 的输入。对于 CLK,每按键 8 两次可以输入一个时钟脉冲。

计数器的 4 位预制输入数据 DATA[3..0]可以利用键 1(键 2 也有相同功能,参考第二章)来输入。此键 1 控制一个输入 FPGA 的 4 位二进制数,图中显示,从高位到低位分别是 PIO11~PIO8。每按一次键 1,输出的 4 位二进制数加 1,具体数字由对应的数码管 D4~D1 显示。而 FPGA 的输出可以选择 8 个数码中的一个来显示,例如用数码 1 显示输出,那么 FPGA 输出端对应的 4 位端口名分别是 PIO19~PIO16。通过查表(参考 2.2),就能查到它们对应于 EP4CE55 的具体引脚。

将以上讨论归纳后得表 3-1。确定了锁定引脚编号后就可以完成以下引脚锁定了:

(1) 假设现在已打开了 CNT10 工程。如果刚打开 Quartus II,应选择 File→Open Project 命令,并单击工程文件 CNT10,打开此前已设计好的工程。

(2) 选择 Assignments→Pin Planner,即进入如图 3-19 所示引脚锁定编辑窗。此图显示,在 Fitter Location 列已经有了锁定好的引脚。只是在 Quartus 对工程编译后自动对电路信号给出的引脚锁定,并不是设计者给出引脚情况。

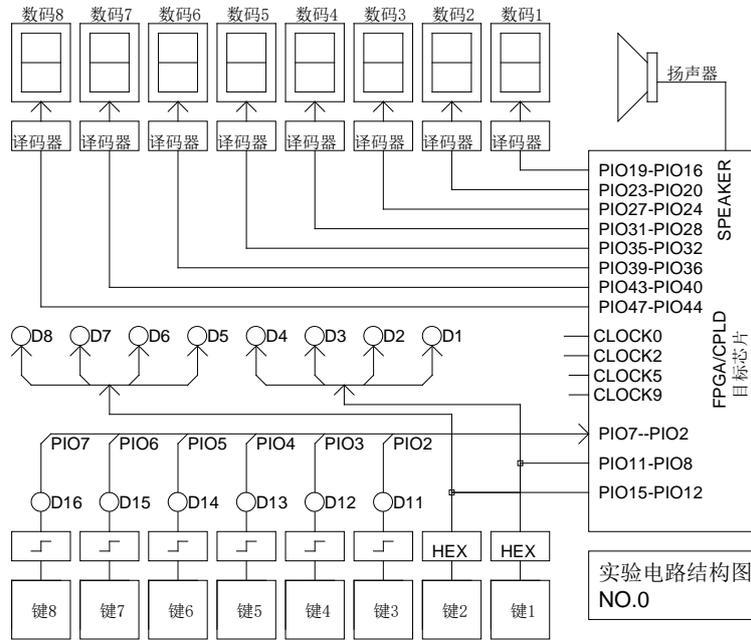


图 3-18 模式 0 的对 FPGA 的实验电路

表 3-1 基于 KX-CDS 的引脚锁定情况（可通过 2.2 的列表获得）

计数器信号名	CLK	EN	LOAD	RST	DATA(3)	DATA(2)	DATA(1)
模式 0 电路控制	键 8	键 7	键 6	键 5	键 1:D4	键 1:D3	键 1:D2
模式 0 电路信号	PIO7	PIO6	PIO5	PIO4	PIO11	PIO10	PIO9
EP4CE55F484/10CL055YF484	AB6	Y7	AA6	AB3	AB5	AA3	W2
EP4CE6/10E22C8N	84	34	75	67	74	70	65
EP3C16F484C6N(DE0)	U14	W13	V12	V11	U13	R10	Y10
5CEBA4F23C7N(DE0-CV)	M18	L17	K17	P18	L19	K19	R15
5CSEMA5F31C6N (DE1-SOC)	AJ20	AK21	AD20	AE19	AD19	AE18	AF20
计数器信号名	DATA(0)		COUT	DOUT(3)	DOUT(2)	DOUT(1)	DOUT(0)
模式 0 电路控制	键 1:D1		数码 2:a 段	数码 1	数码 1	数码 1	数码 1
模式 0 电路信号	PIO8		PIO20	PIO19	PIO18	PIO17	PIO16
EP4CE55F484/10CL055YF484	U2		AA1/Y14	V2	W1	R2	U1
EP4CE6/10E22C8N	60		51	46	49	43	44
EP3C16F484C6N(DE0)	T8		T10	U9	T9	Y7	U8
5CEBA4F23C7N(DE0-CV)	R16		F12	G16	G15	G13	G12
5CSEMA5F31C6N (DE1-SOC)	AF19		AG22	AF24	AF23	AE22	AD21

(3) 双击图 3-19 所示的 Location 栏对应的的信号位置，根据表 3-1 键入对应的引脚，再按回车键，依次下去，输入所有的引脚信号。完成后的情况如图 3-20 所示。

(4) 注意在输入所希望的引脚编号时，有可能显示不出来，说明此引脚不正确，或是因为此脚只能作输入口，不能作输出口；或者不存在此引脚名等原因。当然即使接受此引脚名，也不能说明此引脚一定合法。编译后有可能报错。总之，读者在设计前还应该了解更多的有关当前 FPGA 的信息。

最后必须再编译一次，即启动 Start Compilation。以后每改变一次引脚或其他设置，都要重新编译后才能将引脚锁定信息编译进编程下载文件中。此后就可以准备将编译好的文件下载到实验系统的 FPGA 中了。

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location
in CLK	Input				PIN_J4
out COUT	Output				PIN_T8
in DATA[3]	Input				PIN_R8
in DATA[2]	Input				PIN_R10
in DATA[1]	Input				PIN_T9
in DATA[0]	Input				PIN_V6
out DOUT[3]	Output				PIN_R9
out DOUT[2]	Output				PIN_T5
out DOUT[1]	Output				PIN_R6
out DOUT[0]	Output				PIN_P8
in EN	Input				PIN_T7
in LOAD	Input				PIN_R7
in RST	Input				PIN_P4

图 3-19 编译完成后刚打开的 Pin Planner 窗

Node Name	Direction	Location
in CLK	Input	PIN_AB6
out COUT	Output	PIN_AA1
in DATA[3]	Input	PIN_AB5
in DATA[2]	Input	PIN_AA3
in DATA[1]	Input	PIN_W2
in DATA[0]	Input	PIN_U2
out DOUT[3]	Output	PIN_V2
out DOUT[2]	Output	PIN_W1
out DOUT[1]	Output	PIN_R2
out DOUT[0]	Output	PIN_U1
in EN	Input	PIN_Y7
in LOAD	Input	PIN_AA6
in RST	Input	PIN_AB3

图 3-20 引脚锁定完成后的情况

### 3.3.2 编译文件下载

将编译产生的 SOF 格式配置文件下载进 FPGA 中，进行硬件测试的步骤如下：

(1) 连接好下载器，a、如是康芯核心板，先用提供的 USB 线连接电脑的 USB 口（最好是电脑后面 USB 口），另一端连接下载器标有“ALTERA USB-Blaster”的那个口，下载器的另一端的十芯口用十芯线一端连接，十芯线另一端连接核心板的 JTAG 的十芯口。b、如果是 DE 系列核心板，直接 USB 线把电脑和核心板上 USB 口连接即可。

#### (2) USB-Blaster 驱动程序安装方法

对于有的核心板，在初次使用 USB-Blaster 编程器前，需首先安装 USB 驱动程序。

(3) 将 USB Blaster 编程器一端插入 PC 机的 USB 口，这时会弹出一个 USB 驱动程序对话框，根据对话框的引导，选择用户自己搜索驱动程序，这里假定 Quartus II 安装在 E 盘，则驱动程序的路径为 E:\altera\quartus\drivers\usb-blaster\usbblstr.inf。如没有弹出对话框，需手动安装，在“此电脑”右键->属性->资源管理器->通用资源管理器->扫描硬件-> ALTERA USB-Blaster->点击“!”->更新驱动程序->手动安装（不要自动安装）->选择下面“从计算机列表中选择”->点击“从磁盘安装”->点击“浏览”找到安装软件 altera\quartus\drivers\usb-blaster\usbblstr.inf 路径。**如是 DE1-SOC 核心板，采用 USB-BlasetII 下载方式，设置下载驱动请浏览本公司提供的文件夹“DE-SOC 下载驱动设置”里的友晶公司提供的文件。**

安装完毕后，打开 Quartus II，选择编程器，单击图 3-21 左上角的 Hardware Setup 按钮，在弹出的窗口中双击 USB-Blaster 项。此后就能如同前面介绍的编程器一样使用了。

(1) 打开编程窗和配置文件。在工程管理窗口（如图 3-8 所示）中选择 Tools→Programmer 命令，弹出如图 3-21 所示的编程窗口。在 Mode 下拉列表框中有 4 种编程模式可以选择：JTAG、Passive Serial、Active Serial Programming 和 In-Socket Programming。

为了直接对 FPGA 进行下载（配置），在编程窗口的编程模式 Mode 中选择 JTAG（默认），并选中（打勾）下载文件右侧的第一个复选框。注意要仔细核对下载文件路径与文件名，确定就是当前工程生成的编程文件（注意此文件所在路径的文件夹是 output\_files）。如果此文件没有出现，可单击左侧的 Add File 按钮，手动选择配置文件 CNT10.sof。

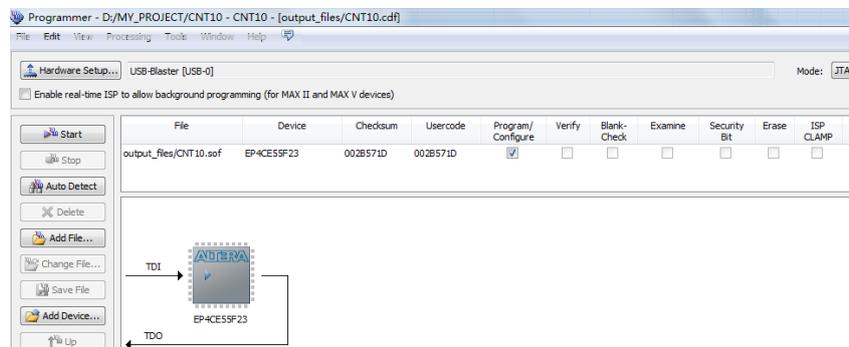


图 3-21 选择 JTAG 编程模式，将 SOF 文件载入 FPGA

(2) 设置编程器。若是初次安装的 Quartus II，在编程前必须进行编程器选择操作。这里准备选择 USB-Blaster。单击图 3-21 左上角的 Hardware Setup 按钮，在弹出的窗口中设置下载接口方式（如图 3-22 所示）。在 Hardware Setup 对话框中，双击此选项卡中的 USB-Blaster 选项之后，单击 Close 按钮，关闭对话框即可。这时应该在编程窗口右上方显示出编程方式：USB-Blaster，如图 3-21 所示。

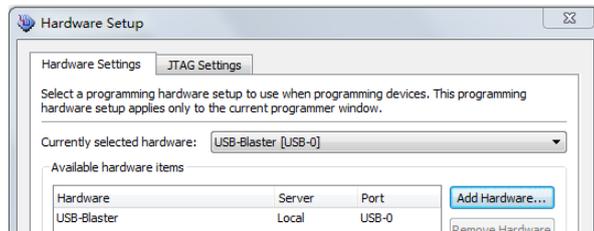


图 3-22 加入编程下载方式

如果在如图 3-22 所示的窗口中 Currently selected hardware 右侧下拉列表框中显示 No Hardware, 则必须加入下载方式。即单击 Add Hardware 按钮, 在弹出的窗口中单击 OK 按钮, 再双击 USB-Blaster, 使 Currently selected hardware 右侧下拉列表框中显示 USB-Blaster。

设定好下载模式后可以先删去图 3-21 所示的 SOF 文件, 再单击 Auto Detect 按钮。如果 JTAG 口的设置以及开发板的连接没有问题, 应该测出板上的 FPGA 的型号。

如图 3-21 所示, 向 FPGA 下载 SOF 文件前, 要选中 Program/Configure 复选框。最后单击 Start 按钮, 即进入对目标器件 FPGA 的配置下载操作。当 Progress 显示出 100% 时, 表示编程成功。

(3) 硬件测试。对于图 3-18 预先选择的控制情况, 让各键输出对应功能的电平或脉冲, 观察系统的输入和输出情况, 再与图 3-17 的仿真波形进行对照。下载后, 复位一下主系统, 使系统稳定, 键 5 低电平复位, 键 1 输入二进制信号如“0101”显示于 D1~D4, 键 7 使能高电平有效, 键 8 每按两次计一次数, 键 6 低电平把键 1 预制数加载进去。

### 3.4 编程配置器件

#### 3.3.3 通过 AS 口对配置芯片进行直接编程

为了使 FPGA 在上电启动后仍然保持原有的配置文件, 并能正常工作, 必须将配置文件烧写进专用的配置芯片 EPCSx 中。EPCSx 是 Cyclone 系列器件的专用配置器件, Flash 存储结构, 编程周期 10 万次。编程模式为 Active Serial 模式, 编程接口为 Usb-Blaster 或 II。以下给出编程流程:

##### 1. 选择编程模式和编程目标文件(适合 DE 系列)

在如图 3-21 所示窗口的 Mode 栏, 选择“Active Serial Programming”编程模式, 打开编程文件, 选中文件 cnt10.pof, 并选中打勾 3 个编程操作项目。

##### 2. AS 模式编程下载

单击窗口的 Start。编程成功后 FPGA 将自动被 EPCS 器件配置而进入工作状态。此后每次上电, FPGA 都能被 EPCS 自动配置, 进入正常工作状态。

#### 3.3.4 通过 JTAG 口对配置芯片进行间接编程

对于一般用户的开发板, AS 直接模式下载涉及复杂的保护电路, 为了简化电路, 以下介绍利用 JTAG 口对配置器件进行间接配置的方法。具体方法是首先将 SOF 文件转化为 JTAG 间接配置文件, 再通过 FPGA 的 JTAG 口, 将此文件对 EPCS 器件进行编程。

##### 1. 将 SOF 文件转化为 JTAG 间接配置文件

选择 File→Convert Programing Files 命令, 在弹出的窗口中作如下设置(如图 4-23 所示)。

(1)首先在 Programming file type 下拉列表框中选择输出文件类型为 JTAG 间接配置文件类型: JTAG Indirect Configuration File, 后缀为.jic。

(2)然后在 Configuration device 下拉列表框中选择配置器件型号, 选择 EPCS16, 这是由于核心板 KX-4CE55 上的配置器件就是 EPCS16(容量 16Mb)。

(3)再于 File name 文本框中输入输出文件名, 如: EPCS16\_file.jic。

(4)单击最下方 Input files to convert 栏中的 Flash Loader 项, 然后单击右侧的 Add Device 按钮, 这时将弹出 Select Devices 器件选择窗口。在此窗口左栏中选定目标器件的系列, Cyclone IV E; 再于右栏中选择具体器件: EP4CE55; 单击(选中) Input file to convert 栏中的 SOF Data 项, 然后单击右侧的 Add File 按钮, 选择 SOF 文件 CNT10.sof。

(5)选择压缩模式。单击选中加入的 SOF 文件名, 再单击右侧的 Properties 按钮, 选中 Compression 复选框(如图 4-23 下侧小窗), 单击 OK 按钮完成。最后单击 Generate 按钮, 即生成所需要的 JIC 编程文件。

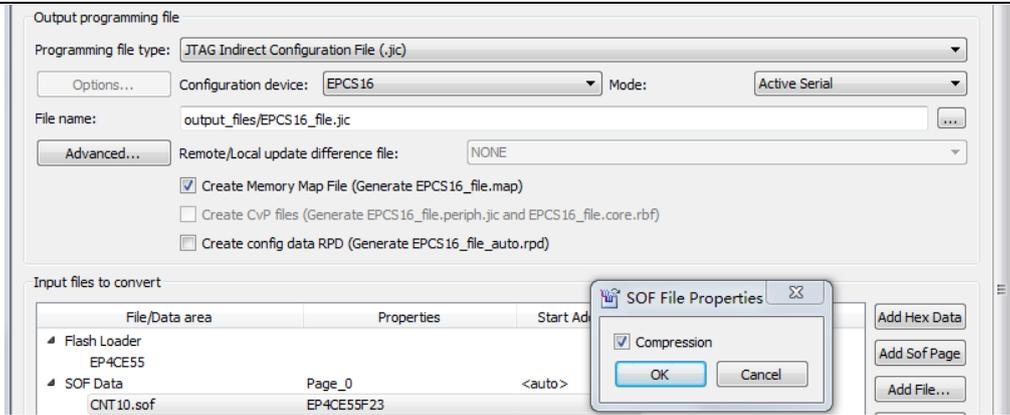


图 4-23 设定 JTAG 间接编程文件

## 2. 下载 JTAG 间接配置文件

选择 Tool→Programmer 命令（JTAG 模式），加入 JTAG 间接配置文件 EPCS16\_file.jic，如图 4-24 所示作必要的选择（注意一些打钩的操作项），单击 Start 按钮后进行编程下载。为了证实下载后系统是否能正常工作，在下载完成后，必须关闭系统电源，然后再打开电源，以便启动 EPCS 器件对 FPGA 的配置。然后观察计数器的工作情况。

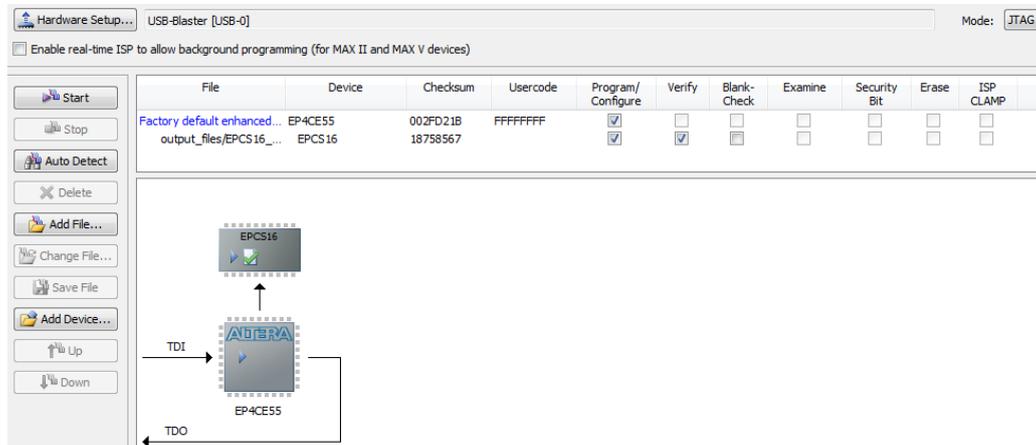


图 4-24 用 JTAG 模式将间接配置文件烧入配置器件 EPCS16 中

## 第四章、EDA 基础及综合实验

本章主要介绍 KX-CDS 系列系统的经典实验与设计示例项目，其中绝大多数实验都有配备的源程序和对应的实验指导课件 ppt 文件（针对 KX-CDS 系统）。列出实验包括文件夹：

\DEMO\EDA\_DEMO\LED\_DEMO 是主系统上 LED、数码管显示的 EDA 实验

\DEMO\EDA\_DEMO\MODU\_DEMO 是主系统上扩展模块的 EDA 实验

### 实验 4.1 十六进制计数器

实验指导路径：**DEMO\EDA\_DEMO\LED\_DEMO\DEMO1\_4Bit\_COUNTER\LED\_示例实验指导演示\DEMO1\_四位二进制计数器 1.PPT。**

#### (1) 实验目的：

1、学习计数器的设计、进一步熟悉实验系统的锁定引脚方法和性能。

#### (2) 实验原理：

4 位二进制计数方法，通过时钟 CLK 上升沿触发一次，计数一次，从'0000'一直计到'1111'，在通过实验系统上的译码器显示 0~F。

#### (3) 实验步骤 1：

对程序进行编辑、编译、综合、适配、仿真。说明例中各语句的作用。给出其所有信号的时序仿真波形，根据波形详细描述此设计的功能特点，包括 RST、CLK 等信号等异步和同步特性。查阅编译后的计数器的时序特点，从时序仿真图和编译报告中了解计数时钟输入至计数数据输出的延时情况，包括设定不同优化约束后的改善情况以及当选择不同 FPGA 后的延时情况，给出分析报告。

#### (4) 引脚锁定:

在此选择 KX-CDS 系统的电路模式 No.5 (图 2-8)，查 2.2 表，4C55/10LC55 板确定引脚分别为：主频时钟 CLK 主系统上键 2 查表是脚 R1；复位 RST 则接电路模式 No.5 的键 1 (PIO0 对应第 N1 脚，注意键序与引脚号码并无对应关系)；4 位输出数据总线 Q[3..0]可由数码 1 来显示，通过分别接 PIO19、PIO18、PIO17、PIO16 (它们对应的引脚编号分别为 V2, W1, R2, U1)。(如是 DE0 板，CLK:接 W6,RST:U7, Q[3..0]分别接 U9, T9, Y7, U8) (如是 DE0-CV 板，CLK:接 T18,RST:T15, Q[3..0]分别接 G16, G15, G13, G12)、(如是 DE1-SOC 板，CLK:接 AG20,RST:AJ21, Q[3..0]分别接 AF24, AF23, AE22, AD21)如是 4EC6/10 板，CLK:接 P55,RST:P52, Q[3..0]分别接 P46、P49、P43、P44。

#### (5) 下载测试:

实验步骤 1、成功下载 cnt4.sof 后，选择实验电路模式 5；

实验步骤 2、选键 1 置高电平，后置低电平，使 RST 产生复位信号，按动 CLK 键 2。观察数码 1 了解计数器工作情况，每按动一次键 2，数码 1 计数一次，1~F 循环显示。

#### (6) 实验内容 1:

使用 SignalTap II 对此计数器进行实时测试，给出报告。

#### (7) 实验内容 2:

为此项设计加入一个可用于 SignalTap II 采样的独立的时钟输入端 CLK。计数时钟可以低一点，而采样时钟可高一些，如选择 CLK=2MHz，并进行实时测试。

#### (8) 实验内容 3:

建立一个原理图工程，转换原理图 CNT4 元件。然后按照此图的连接方式完成设计。对此电路进行仿真，并说明此电路的功能特点。如何利用此电路设计一个不同模的计数器，或可预置的分频器？最后在开发板上硬件实现，验证分频情况。

#### (9) 实验报告:

将实验原理、设计过程、编译仿真波形和分析结果、硬件测试结果写进实验报告。

#### (10) 实验任务 1:

把 CLK 时钟更改为利用主系统上连续时钟输入，利用单线一端连接主系统 CLKB0 端 (J17)，另一端选择标注“9”的时钟源端，(可从高到底选择时钟输入)，查表得到对应芯片型号的引脚，看计数情况。

对于配置器件，如没有 AS 模式下载方式，可以通过 sof 文件转换成 jic 文件通过间接方式也可以对配置器件进行编程康芯给出间接配置方法可浏览文件夹“重要文件/FPGA\_单片机\_编程”。

#### (11) 实验内容 2:

从设计中去除 SignalTap II，要求全程编译后，将生成的 SOF 文件转变成用于配置器件 EPCS4 的压缩的间接配置文件 \*. jic，并使用 USB-Blaster 对实验板上的 EPCSX 进行编程，最后进行验证。编程和全程编译前，设定所有控制和参数。

## 实验 4.2 7 段译码器设计

实验系统演示示例(含实验指导 pdf 课件): DEMO\EDA\_DEMO\LED\_DEMO\ DEMO7\_DECODER

### (1) 实验目的:

1、学习 7 段数码显示译码器的硬件设计。

### (2) 实验原理:

7 段数码是纯组合电路。通常的小规模专用 IC, 如 74 或 4000 系列的器件只能作十进制 BCD 码译码, 然而数字系统中的数据处理和运算都是二进制的, 所以输出表达都是十六进制的。为了满足十六进制数的译码显示, 最方便的方法就是利用 Verilog/VHDL 译码程序在 FPGA 中来实现。所以首先要设计一段程序。该程序可按照 case 语句表述方法, 再根据真值表写出程序。设输入的 4 位码为 A[3:0], 输出控制 7 段共阴数码管 (图 4-1) 的 7 位数据为 LED7S[6:0]。输出信号 LED7S 的 7 位分别接图 4-3 的共阴数码管的 7 个段, 高位在左, 低位在右。例如当 LED7S 输出为“1101101”时, 数码管的 7 个段 g、f、e、d、c、b、a 分别接 1、1、0、1、1、0、1; 接有高电平的段发亮, 于是数码管显示“5”。这里没有考虑表示小数点的发光管, 如果要考虑, 需要增加段 h, 然后将 LED7S 改为 8 位输出。

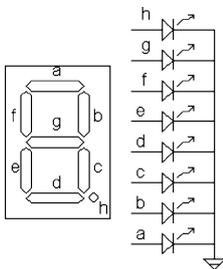


图 4-1

A	0001	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	0000	0001
LED7S	H 06	3F	06	5B	4F	66	6D	7D	07	7F	6F	77	7C	39	5E	79	71	3F	06

图 4-2 7 段译码器仿真波形

### (3) 实验内容:

将设计好的 VHDL 译码器程序在 Quartus II 上进行编辑、编译、综合、适配、仿真, 给出其所有信号的时序仿真波形。提示: 设仿真激励信号时用输入总线的方式给出输入信号仿真数据, 仿真波形图如图 4-2 所示。

#### 【例 4-7】

文本: DECL7S

### (4) 引脚锁定:

引脚锁定及硬件测试。引脚锁定及硬件测试。建议选系统的实验电路模式 6, 用数码 8 显示译码输出 (PI046-PI040), 键 8、键 7、键 6 和键 5 四位控制输入接 (PI013-PI010), 硬件验证译码器的工作性能。

### (5) 实验现象

根据上面输入的 A 段输入四位信号, 键 5-键 8, 对照 CASE 语句, 看数码 8 译码情况。

### (6) 实验任务 1:

用例化语句, 连接成顶层设计电路 (用 VHDL 表述), 用一个 4 位二进制加法计数器 (例 4-2), 模块 DECL7S 即为例实体元件, 重复以上实验过程。注意图 tmp 是 4 位总线, led 是 7 位总线。对于引脚锁定和实验, 建议选电路图 2-9 共阴数码管, 模式 6, 用数码 8 显示译码输出。

### (7) 实验报告:

根据以上的实验内容写出实验报告, 包括程序设计、软件编译、仿真分析、硬件测试和实验过程; 设计程序、程序分析报告、仿真波形图及其分析报告。

## 实验 4.3 十六进制七段译码设计

实验指导路径：DEMO\EDA\_DEMO\LED\_DEMO\DEMO2\_4Bit\_COUNTER\_7SEG

### (1) 实验目的：

- 1、熟悉 Quartus II 的文本设计流程全过程，学习计数器的设计、仿真和硬件测试。掌握原理图与文本混合设计方法。
- 2、了解 CASE 语句的设计方法
- 3、掌握原理图与文本混合的设计方法

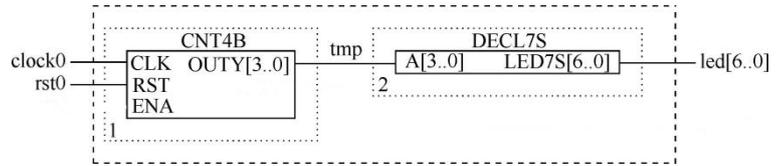


图 4-3 原理图示意图

### (2) 实验原理：

实验 4.1 已经完成 4 位计数器，输出连接主系统已有的实验 4.3 译码器，那么这个实验译码器自己设计，利用 4 位计数器连接 7 段译码，数码管进行显示控制，如图 4-1。利用一个输出四位二进制计数器作为七段译码器的输入端，输出利用 CASE 语句对数码 a、b、c、d、e、f、g 进行点亮，‘0’不亮，‘1’亮。

例【4-3】：

文本文件：DECL7S

### (3) 实验步骤 1：

对程序进行编辑、编译、综合、适配、仿真。说明例中各语句的作用。给出其所有信号的时序仿真波形，根据波形详细描述此设计的功能特点，包括 RST、CLK 等信号等异步和同步特性。查阅编译后的计数器的时序特点，从时序仿真图和编译报告中了解计数时钟输入至计数数据输出的延时情况，包括设定不同优化约束后的改善情况以及当选择不同 FPGA 后的延时情况，给出分析报告。

### (4) 引脚锁定：

采用不同方式锁定引脚并硬件下载测试。选择模式 2，键 1 作为复位接 PIO48，键 2 为时钟输入接 PIO49，输出选择数码 5 接 PIO16~PIO22，查表引脚锁定后进行编译、下载和硬件测试实验。将实验过程和实验结果写进实验报告。硬件实验中，注意测试所有控制信号和显示信号，包括 RST、CLK 等的同步、异步特性，进位信号等。时钟 CLK 换不同输入：手动有抖动或无抖动键输入，或主系统统上 1Hz 或 4Hz 时钟脉冲输入。

### (5) 下载测试：

编译下载后，复位主系统，数码 2，随着时钟的输入，显示 0-F 循环显示。

## 实验 4.4 多路选择器设计

实验路径：DEMO\EDA\_DEMO\LED\_DEMO\DEMO3\_MUX41

### (1) 实验目的：

- 1、进一步熟悉 Quartus II 的文本设计流程，组合电路的设计仿真和硬件测试。
- 2、掌握硬件描述语言对于组合电路描述的核心语法规则、基本设计方法和与之相关的逻辑电路设计经验。

### (2) 实验原理：

四选一多路选择器的电路模型如图 4-4 所示，图中的 a、b、c、d 是 4 个输入端口；s1 和 s0 为通道选择控制信号端，y 为输出端。当 s1 和 s0 取值分别为 00、01、10 和 11 时，输出端 y 将分别输出来自输入口 a、b、c、d 的数据。图中显示，当 a、b、c、d 四个输入口分别输入不同频率信号时，针对选通控制端 s1、s0 的不同电平选择，则输出端 y 有对应的信号输出利用 Quartus II 完成 4 选 1 多路选择器的文本编辑输入。根据此电路的功能要求，可得到相应的逻辑描述，此例中，4 组数据的获得通过时钟分频出 4 组数据来驱动扬声器的发声的高低。

例【4-4】

输入文本：MUX41A

### (3) 实验步骤 1：

对程序进行编辑、编译、综合、适配、仿真，如图 4-5 仿真结果

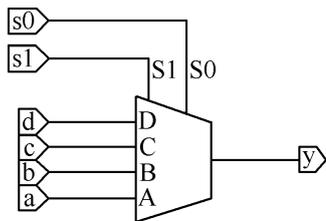


图 4-4

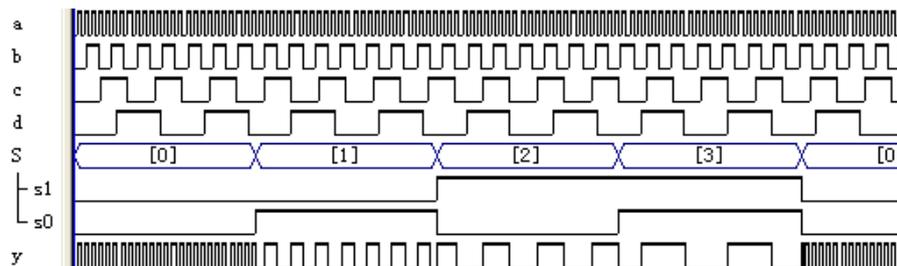


图 4-5 四选一多路选择器 MUX41a 的时序波形

#### (4) 引脚锁定:

在实验系统上硬件测试，选择模式 5 验证此设计的功能。引脚锁定 CLK 锁定核心板提供的 DE 板的 50MHz 或 KX 板的 20MHz 时钟源，通过时钟计数分频分别供给 a、b、c 和 d，再通过选择键 1、键 2 接来自不同的时钟；输出信号接蜂鸣器。

引脚 CLK 接 DE1-SOC 的时钟如图 4-6，选择时钟 CLOCK\_50，引脚号为 AA16，DE0、DE0-CV 分别浏览对应引脚文件，KX 的 EPEC6/10/55，查找提供的核心板说明文档。选择通道利用键 1、键 2，扬声器连接主系统“J7”十芯座的“DBT1”，查表得引脚号。

板时钟输入端名	FPGA 引脚名	时钟信号	I/O Standard
CLOCK_50	PIN_AF14	50 MHz clock input	3.3V
CLOCK2_50	PIN_AA16	50 MHz clock input	3.3V
CLOCK3_50	PIN_Y26	50 MHz clock input	3.3V
CLOCK4_50	PIN_K14	50 MHz clock input	3.3V
HPS_CLOCK1_25	PIN_D25	25 MHz clock input	3.3V
HPS_CLOCK2_25	PIN_F25	25 MHz clock input	3.3V

图 4-6 DE1-SOC 提供的时钟引脚名

#### (5) 硬件连接:

硬件连接，扬声器利用单线连接主系统“J7”十芯座的“DBT1”最后进行编译、下载和硬件测试。

#### (6) 下载测试:

编译下载后，复位主系统，实验（通过选择键 1、键 2，控制 s0、s1，分别输入“00、01、10、11 可使蜂鸣器输出不同音调）。

#### (7) 实验报告:

根据以上的实验内容写出实验报告，包括程序设计、软件编译、仿真分析、硬件测试和详细实验过程；给出程序分析报告、仿真波形图及其分析报告。参考对应的 ppt 实验指导课件。

## 实验 4.5 逻辑门电路

实验路径: DEMO\EDA\_DEMO\LED\_DEMO\DEMO\_logic

#### (1) 实验目的:

- 1、了解 EDA 软件环境的原理图及器件库等实验环境。
- 2、理解二输入与门工作原理。

#### (2) 实验内容:

输入与门。首先还是要建立一个文件夹，取一个文件名，具体参照第三章，这里不再赘述。

1、打开原理图编辑窗。打开 Quartus II，选菜单 File→New，在弹出的 New 对话框中选择原理图文件编辑输入项 Block Diagram/Schematic File (如图 4-7 所示)，按 OK 按钮后将打开原理图编辑窗口。

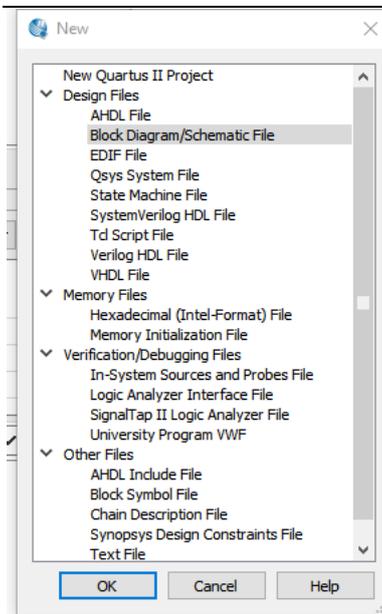


图 4-7 原理图输入选择



图 4-8 选择打开元件输入窗

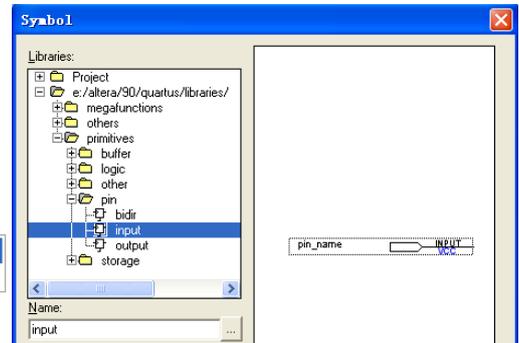


图 4-9 在元件输入对话框输入引脚

(2) 建立一个初始原理图。在编辑窗口中的任何一个位置上右击鼠标，将出现快捷菜单，选择其中的输入元件项 Insert→Symbol(图 4-7)，或直接双击原理图编辑窗口，于是将弹出如图 4-9 所示的输入元件的对话框。在左下的 Name 栏键入输入引脚符号 input。然后单击 Symbol 窗口的 OK 按钮，即可将元件调入原理图编辑窗口中。在 Name 栏 输入 分别 AND2、输出端口

(3) 最终输入以一个与门的输入输出电路图原理图文件存盘。选择菜单 File→Save As，将此原理图文件先存于刚才建立的目录中，将已设计好的原理图文件取名为后缀为.bdf（注意默认的后缀是.bdf，而且此原理图，最终完成的半加器电路设计应该如图 4-10 所示），因为只加入了一个输入端口，并存盘在此文件夹内。

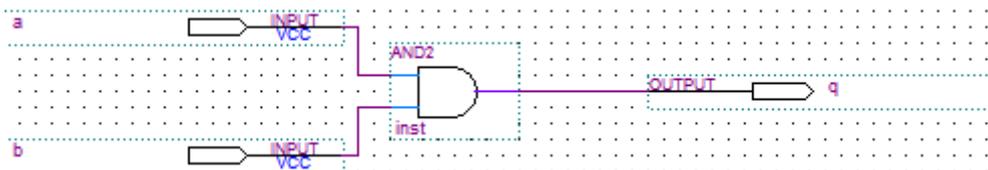


图 4-10，与门原理图

### (3) 实验步骤 1:

仿照第三章内容的流程进行编译，选择器件，仿真，锁定引脚步骤完成。

### (4) 引脚锁定:

选择模式 5，键 1、2 作为 a、b 的输入端，D1 灯作为 q 输出端来观察信号，查表得出引脚号。

### (5) 下载测试:

编译下载后，复位主系统，按动键 1 和键 2 输入高 00、01、10、11 电平，观察 D1 输出信号是否符合此电路逻辑。

## 实验 4.6 触发器

### 实验路径: DEMO\EDA\_DEMO\LED\_DEMO\ DEMO5\_DFF

最简单，最常用，并最具代表性的时序元件是 D 触发器，它是现代数字系统设计中最基本的底层时序单元，甚至是 ASIC 设计的标准单元。JK 和 T 等触发器都可由 D 触发器构建而来。D 触发器的描述包含了 VHDL 对时序电路的最基本和典型的表达方式，同时也包含了 VHDL 许多最具特色的语言现象。

### (1) 实验目的:

对 D 触发器的语言描述进行详细地分析，得出时序电路描述的一般规律和设计方法。

### (2) 实验原理:

具有边沿触发性能的 D 触发器（图 4-11）的工作时序如图 4-12 所示。波形显示，只有当时钟上升沿到来时，其输出 Q 的数值才会随输入 D 的数据而改变，在这里称之为更新。例 4.6 给出了文本对 D 触发器的一种常用描述形式。

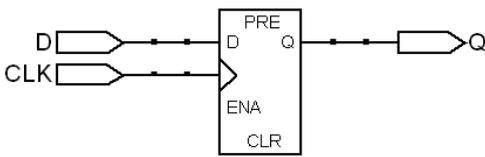


图 4-11 D 触发器模块图

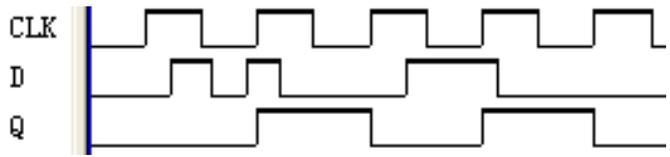


图 4-12 D 触发器时序波形

**(3) 实验内容 1:**

根据实验上述原理，设计触发器描述语言，给出程序设计、软件编译、仿真分析、硬件测试及详细实验过程。

**【例 4-6】**

文本：DFF1

**(4) 实验内容 2:**

根据上述语言描述，进行仿真测试。锁定引脚，下载硬件测试。

**(5) 引脚锁定:**

在 FPGA 中进行实测。对于 CDS\_EDA 实验系统，建议选择电路模式 5（第二章图 2-8），键 1 锁定 CLK 脚 PI0，键 2 锁定 D 脚 PIO2，D1 锁定 Q 作为输出信号 PIO8，查表锁定引脚。

**(6) 下载测试:**

编译下载后，复位主系统，当键 2 输入一个高电平，等待键 1 给予 CLK 信号，当 CLK 接收到信号，D 信号输入给 Q 在 D1 显示。

**(7) 实验任务:**

设计 JK 和 T 触发器。

设计一个异步复位和时钟使能 D 触发器。

## 实验 4.7 八位加法器设计

实验路径：\DEMO\EDA\_DEMO\LED\_DEMO\DEMO6\_ADDER”

**(1) 实验目的**

- 1、掌握并行加法器的原理及设计方法；
- 2、通过一个 8 位二进制加法器的设计，学习多位加法器的设计；
- 3、掌握用硬件描述语言实现简单组合逻辑电路的设计；

**(2) 实验原理**

加法器是数字系统中的基本逻辑器件，例如为了节省逻辑资源减法器 and 硬件乘法器都可由加法器来构成。宽位的加法器的设计是十分耗费硬件资源的，因此在实际的设计和相关系统的开发中需要注意资源的利用率和进位速度两方面的问题。多位加法器的构成有两种方式：并行进位和串行进位方式。并行进位加法器设有并行进位产生，逻辑运算速度较快；串行进位方式是将全加器级联构成多位加法器并行进位。实验表明，4 位二进制并行加法器和串行级联加法器占用几乎相同的资源。因此本实验中的 8 位加法器采用由两个 4 位二进制并行加法器级联的方案。

图 4-13，可以看出输入 2 个 8 位输入相加之和，与进位相加输出，并有溢出位输出，图 4-14，仿真波形。

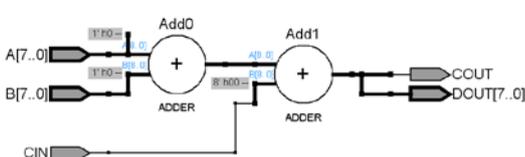


图 4-13 设计原理

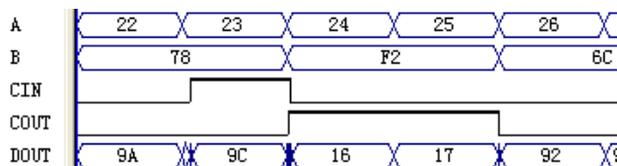


图 4-14 8 位加法器仿真波形

**(3) 实验内容 1:**

根据以上原理及波形图，首先给出 4 位加法器的语言描述

例【4-7-1】 4 位加法器

文本: ADDER4B

例【4-7-2】 8 位加法器

文本: ADDER8B

**(4) 实验内容 2**

语言编译通过，实验仿真，根据上述原理，建立仿真文件，进行仿真。

**(5) 引脚锁定**

编译仿真通过后，进行硬件测试，首先进行引脚锁定， 可选实验电路结构图 NO.1，8 位二进制被加数 A[7..0]选择键 2、1 接 PIO7~PIO0，键 2 为高位；8 位二进制加数 B[7..0]选择键 4、3 接 PIO15~PIO8，键 4 位高位；8 位和 S[7..0]接选择数码 6、5 接 PIO23~PIO16，数码 6 显示高位。进位输出位 COUT 由 PIO39 输出，当有进位时点亮发光二极管 D8。进位输入位 CIN 接 PIO49，由键 8 控制。查表锁定引脚。

**(6) 下载测试**

锁定好以后，编译通过，下载 sof 文件，复位主系统，输入键 1、2 的被加数，再输入键 3、4 的加数，此加法器的 8 位被加数 A 和加数 B 分别由键 2 与键 1、键 4 与键 3 输入，计算结果将显示于数码管 6（和的高 4 位）和数码管 5（低 4 位）。再输入键 8 作为进位位。D8 显示溢出位。

**(7) 实验任务:**

- 1、设计一个直接利用操作符“+”的 8 位及多位加法器。
- 2、设计一个 8 位减法器。

**4.8 数控分频器的设计**

实验路径: \DEMO\EDA\_DEMO\LED\_DEMO\DEMO8\_DVF

**(1) 实验目的:**

学习数控分频器的设计、分析和测试方法。

**(2) 实验原理:**

数控分频器的功能就是当在输入端给定不同输入数据时，将对输入的时钟信号有不同的分频比，数控分频器就是用计数值可并行预置的加法计数器设计完成的，方法是将计数溢出位与预置数加载输入信号相接即可，详细设计程序如例 4-8 所示。

实验【例 4-8】

文本: DVF

**(3) 实验分析:**

根据图 4-5 的波形提示，分析例 4-8 中的各语句功能、设计原理及逻辑功能，详述进程 P\_REG 和 P\_DIV 的作用，并画出该程序的 RTL 电路图。

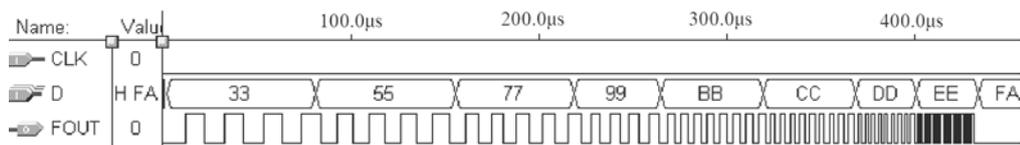


图 4-15 当给出不同输入值 D 时，FOUT 输出不同频率(CLK 周期=50ns)

**(4) 实验仿真:**

输入不同的 CLK 频率和预置值 D，给出如图 4-15 的时序波形。

### (5) 引脚锁定:

在实验系统上硬件验证功能。可选实验电路模式 1 (第二章图 2-4); 键 2/键 1 负责输入 8 位预置数 D(PIO7-PIO0); CLK 由“J17”CLKB1 单线连接输入, 频率选 65536Hz 或更高(确保分频后落在音频范围); 输出 FOUT 接扬声器(SPKER)。查表锁定引脚。

### (6) 硬件连接:

用一根单线一端连接“J17”的 CLKB1 端, 另一端连接时钟源区的 65536Hz, 作为时钟输入源。另一个单线一段连接“J7”的 DBT0 端, 另一端连接“J16”的蜂鸣器输入端。

### (7) 下载测试:

连接好, 编译下载后进行硬件测试: 复位主系统, 改变键 2/键 1 的输入值, 可听到不同音调的声音。

### (8) 实验任务 1:

- 1、将例 4-8 扩展成 16 位分频器, 并提出此项设计的实用示例, 如 PWM 的设计等。
- 2、怎样利用 2 个由例 4-8 给出的模块设计一个电路, 使其输出方波的正负脉宽的宽度分别由两个 8 位输入数据控制?

## 4.9 在 QuartusII 中用原理图输入法设计 8 位全加器

实验路径: \DEMO\EDA\_DEMO\LED\_DEMO\DEMO9\_Fadder8

### (1) 实验目的:

- 1、熟悉利用 Quartus II 的原理图和文本混合输入方法设计简单组合电路, 掌握层次化设计的方法。
- 2、通过一个 8 位全加器的设计把握文本和原理图输入方式设计的详细流程。

### (2) 实验原理:

一个 8 位全加器可以由 8 个 1 位全加器 (图 4-16) 构成, 一位全加器由两个半加器例【4-9-1】组成, 加法器间的进位可以串行方式实现, 即将低位加法器的进位输出  $c_{out}$  与相邻的高位加法器的最低进位输入信号  $c_{in}$  相接图 4-16。最终设计 8 位加法器图 4-17

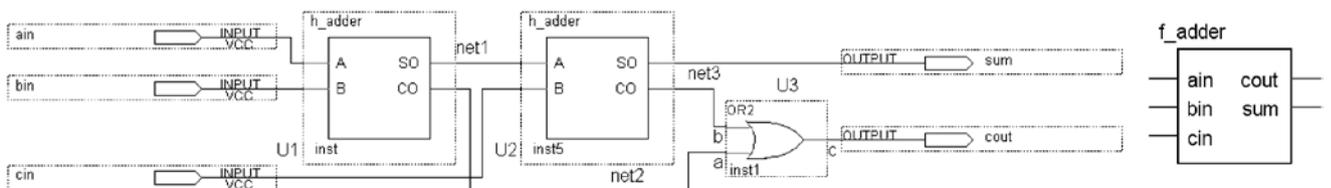


图 4-16 全加器 f\_adder 电路图及其实体模块 f\_adder

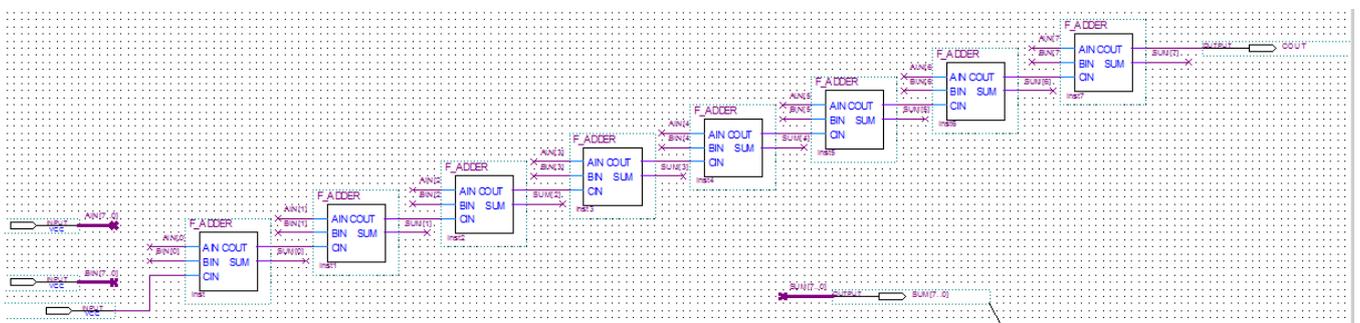


图 4-17 由 8 个全加器构成的 8 位加法器原理图方式

例【4-9-1】半加器

文本: H\_ADDER

### (3) 实验步骤 1:

按照 4-9-1 完成半加器完成 1 位全加器的设计, 包括用文本或原理图输入, 编译、综合、适配、仿真、

实验板上的硬件测试，可以选择模式 5，输入按键 1、键 2，输出接 LED D1 D2，这里就不提供具体操作方法，同学根据以上锁定引脚检验，自行完成。并将此全加器电路设置成一个元件符号入库。

#### (4) 实验步骤 2:

建立一个更高层次的原理图设计，利用以上获得的 1 位全加器构成 8 位全加器图 4-17，并完成编译、综合、适配、仿真和硬件测试图。

#### (5) 锁定引脚:

参照实验 4.6 方式，这里就不再赘述。

#### (7) 下载测试:

实验结果参照实验 4.6 方式。

## 4.10 移位运算器设计

实验路径: \DEMO\EDA\_DEMO\LED\_DEMO\DEMO10\_SHIFTER

#### (1) 实验目的:

完成一个带进位的 8 位移位运算器的功能时序电路。

#### (2) 实验原理:

带有同步预置控制功能的 8 位右移移位寄存器。CLK 是移位时钟信号，DIN 是 8 位并行预置数据端口，LOAD 是并行数据预置使能信号，QB 是串行输出端口，DOUT 是移位并行输出。此电路的工作原理是：当 CLK 的上升沿到来时进程被启动，如果这时预置使能 LOAD 为高电平，则将输入端口的 8 位二进制数并行置入移位寄存器中，作为串行右移输出的初始值；如果 LOAD 为低电平，则执行语句：

```
REG8(6 DOWNTO 0) <= REG8(7 DOWNTO 1);
```

此语句表明：

(1) 一个时钟周期后将上一时钟周期移位寄存器中的高 7 位二进制数，即当前值 REG8(7 downto 1) 赋给此寄存器的低 7 位 REG8(6 downto 0)。于是其串行移空的最高位始终由最初并行预置数的最高位填补。

(2) 将上一时钟周期移位寄存器中的最低位，即当前值 REG8(0) 向 QB 输出。

随着 CLK 脉冲的到来，就完成了将并行预置输入的数据逐位向右串行输出的功能，即将寄存器中的最低位首先输出。此例利用进程中的非完整条件语句构成了时序电路，同时又利用了信号赋值的“并行”特性实现了移位。

#### 【例 4-10】

文本: SHFT

#### (3) 实验仿真

建立仿真文件，由时序波形可见，由于第一个加载信号没有出现在时钟的上升沿处，数据未被载入；并行输入的数据“10011011”直到第二个加载信号出现才于第二个时钟上升沿处被载入。此时 DIN 口上的 8 位数据被锁入 REG8 中。第三个时钟，以及以后的时钟信号都是移位时钟。但应该注意的是，由于程序中赋值语句 QB<=REG8(0) 在 IF 语句结构的外面，因此它的执行并非需要当前的时钟信号，属于异步方式。即最低位的串行输出要早于移位时钟一个周期。这一点可以从波形图（图 4-18）中清楚地看出：在第二个执行并行数据加载的时钟后，QB 即输出了被加载的第一位右移数 '1'，而此时的 REG8 内仍然是“10011011”。

第三个时钟后，QB 输出了右移出的第二个位 '1'，此时的 REG8 内变为“CD”，其最高位被填为 '1'。如此进行下去，直到第八个 CLK 后，右移出了所有 8 位二进制数，最后一位是 '1'。此时 REG8 内是“FF”，即全部被 DIN 的最高位 '1' 填满。



图 4-18 例 5-17 的工作时序

#### (4) 锁定引脚:

选择实验模式 0 锁定引脚，DIN 作为待移位数据锁定键 1、2 (PI08-PI015)，键 3 锁定 LOAD 装载数据及移位控制 (PI02)，CLK 锁定键 8 (PI07)，DOUT 输出锁定 DE1-SOC/DE0/DE0-CV 板上的 10 个 LED (LEDR[0]-[7]，QB 接 LEDR[8] 最低位显示 (图 4-19)，引脚查看“友晶的原版资料。KX-4CE6/10/55，可利用主系统上 8 个 LED 放光管作为移位观察，通过主系统的 J6 十芯口和 J11 (LED) 控制端) 连接，QB 接 KX-4CE6/10 核心板上 LED P132 脚，KX-4CE55 的 C1 脚

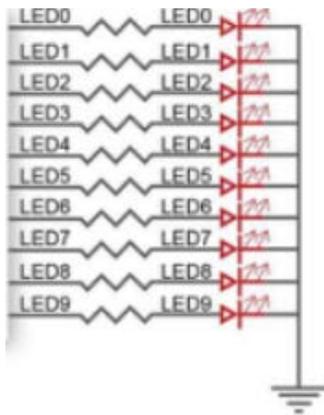


图 4-19 DE LED 原理图

#### (4) 实验操作:

下载 SOF 文件于实验系统，模式 0，复位系统；第一步：按键 1，键 2，输入待移位数据 (如 00100101H)，每按键一次，发光管累加一次，一直将此数将显示 8 个发光管 D8-D1 上：00100101；第二步：键 3 (LOAD) 置高电平；第三步：按动键 8 (CLK)，观察 8 个 LED 显示是否是和输入的数相吻合，如果不是，看语言描述是否有问题；第四步：键 3 (LOAD) 置低电平；第五步：按动键 8 (CLK)，每按动一次，观察 LED0-7 是否连续向右移位：00010010 LED8: 1, 00001001 LED8: 1...，一直移空为 0。

#### (5) 实验任务:

- 1、作业：设计一个 16 位并进/并出移位寄存器设计
- 2、仅用例 4.10 一个 8 位移位寄存器，再增加一些电路，如 3 位锁存器等，设计成为一个能为 8 位二进制数进行不同方式移位的移位寄存器。这个电路模型十分容易用到 CPU 的设计中。

## 实验 4.11 序列检测器

实验路径：\DEMO\EDA\_DEMO\LED\_DEMO\DEMO11\_SCHK

#### (1) 实验目的:

用状态机实现序列检测器的设计，了解一般状态机的设计与应用。

#### (2) 实验原理:

序列检测器可用于检测一组或多组由二进制码组成的脉冲序列信号，当序列检测器连续收到一组串行二进制码后，如果这组码与检测器中预先设置的码相同，则输出 1，否则输出 0。由于这种检测的关键在于正确码的收到必须是连续的，这就要求检测器必须记住前一次的正确码及正确序列，直到在连续的检测中所收到的每一位码都与预置数的对应码相同。在检测过程中，任何一位不相等都将回到初始状态重新开始检测。例 4-11 描述的电路完成对序列数“11100101”的检测，当这一串序列数高位在前(左移)串行进入检测器后，若此数与预置的

密码数相同，则输出“A”，否则仍然输出“B”。

**【例 4-11】**

文本：SCHK

**(3) 波形仿真：**

图 4-20 的波形显示，当有正确序列进入时，到了状态 s8 时，输出序列正确标志 SOUT=1。而当下一位数据为 0 时，即 DIN=0 时，进入状态 s3。这是因为测出的数据 110 恰好与原序列数的头 3 位相同。

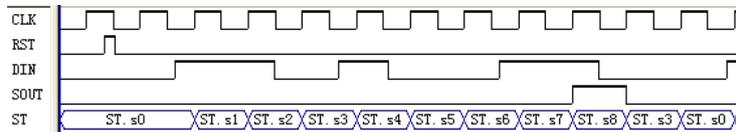


图 4-20 例 4-11 之序列检测器时序仿真波形

**(3) 实验内容：**

利用 QuartusII 对例 4-11 进行文本编辑输入、仿真测试并给出仿真波形，了解控制信号的时序，最后进行引脚锁定并完成硬件测试实验。

**(4) 引脚锁定：**

建议选择电路模式 No.8，用键 7(PIO11)控制复位信号 CLR；键 6(PIO9)控制状态机工作时钟 CLK；待检测串行序列数输入 DIN 接 PIO10(左移，最高位在前)；指示输出 AB 接 PIO39~PIO36(显示于数码管 6)。

**(5) 下载测试：**

下载后：①按实验板“系统复位”键；②用键 2 和键 1 输入 2 位十六进制待测序列数“11100101”；③按键 7 复位(平时数码 6 指示显“B”)；④按键 6(CLK)8 次，这时若串行输入的 8 位二进制序列码(显示于数码 2/1 和发光管 D8~D0)与预置码“11100101”相同，则数码 6 应从原来的 B 变成 A，表示序列检测正确，否则仍为 B。

**(6) 实验任务：**

提出的设计方案，重复以上实验内容(将 8 位待检测预置数由键 4/键 3 作为外部输入，从而可随时改变检测密码)。

**(7) 实验思考：**

如果待检测预置数必须以右移方式进入序列检测器，写出该检测器的 VHDL 代码(两进程符号化有限状态机)，并提出测试该序列检测器的实验方案。

## 4.12 8 位 16 进制频率计设计

**实验路径：** \ DEMO\EDA\_DEMO\LED\_DEMO\DEMO12\_FREQTEST16

**(1) 实验目的：**

设计 8 位 16 进制频率计，学习较复杂的数字系统设计方法。

**(2) 实验原理：**

根据频率的定义和频率测量的基本原理，需要三个模块组成，1、测频控制的时序；2、测频的计数；3、计得的数进行锁存并显示输出。

测频控制：测定信号的频率必须有一个脉宽为 1 秒的输入信号脉冲计数允许的信号；1 秒计数结束后，计数值被锁入锁存器，计数器清 0，为下一测频计数周期作好准备。测频控制信号可以由一个独立的发生器来产生，即图 4-22 中的 FTCTRL。根据测频原理，测频控制时序可以如图 4-21 所示。设计要求是：FTCTRL 的计数使能信号 CNT\_EN 能产生一个 1 秒脉宽的周期信号，并对频率计中的 32 位二进制计数器 COUNTER32B (图 4-22) 的使能端进行同步控制。当 CNT\_EN 高电平时允许计数；低电平时停止计数，并保持其所计的脉冲数。在停止计数期间，首先需要有一个锁存信号 LOAD 的上跳沿将计数器在前 1 秒钟的计数值锁存进锁存器 REG32B 中，并由外部的 16 进制 7 段译码器译出，显示计数值。设置锁存器的好处是数据显示稳定，不会由于周期性的清 0 信号而不断闪烁。锁存信号后，必须有一清 0 信号 RST\_CNT 对计数器进行清零，为下 1 秒的计数操作作

准备。根据以上原理，由测频的时序控制采集外部的频率信号数值，存入 32 位寄存器中显示，图 4-22 中，CLK 给出 1 秒的计数闸门，Fin 采集外部的信号端。以下语言描述为：

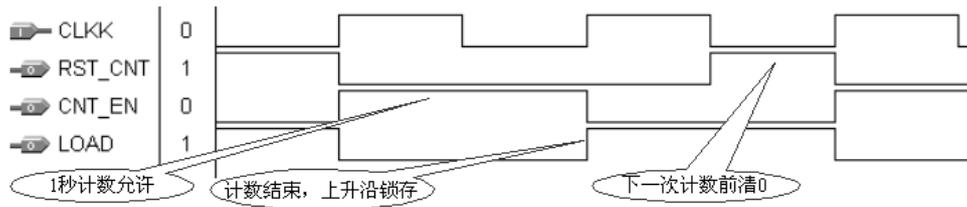


图 4-21 频率计测频控制器 FTCTRL 测控时序图

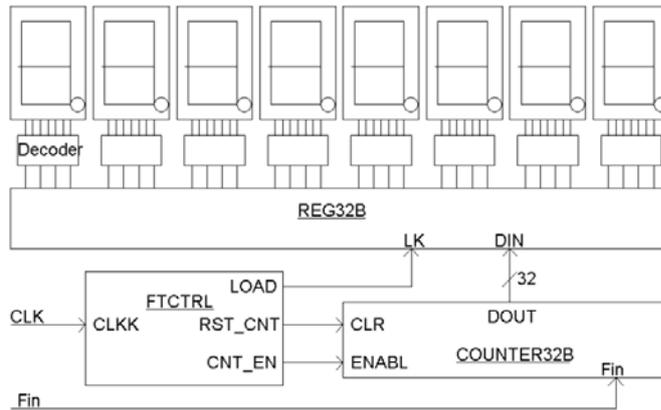


图 4-22 频率计电路框图

【例 4-12-1】测频控制的时序信号

文本：TESTCTL

【例 4-12-2】32 位锁存器

文本：REG32B

【例 4-12-3】32 位计数器

文本：COUNTER32B

【例 4-12-4】顶层文件

文本：FREQTTEST

### (3) 实验内容：

分别仿真测试模块例 4-12-1、例 4-12-2 和例 4-12-3，再结合例 4-12-4 完成频率计的完整设计和硬件实现，并给出其测频时序波形及其分析。

### (4) 引脚锁定：

建议选实验电路模式 5；8 个数码管以 16 进制形式显示测频输出(PIO16~PIO49)；待测频率输入 FIN 由主系统上单线连接 CLKB0，频率可选 4Hz、256Hz、3Hz...20MHz 等；1Hz 测频控制信号 CLK1HZ 可由 CLKB1 输入(用连线时钟源选 1Hz)。查表锁定引脚。注意，这时 8 个数码管的测频显示值是 16 进制的。

### (5) 硬件连接：

1Hz 的测控信号一端连接主系统“J17”的 CLKB1 端，另一端连接主系统左下方的时钟源区的 1Hz 的单针；待测频率端连接主系统“J17”的 CLKB0 端，另一端可利用主系统左下方的时钟源区选择时钟，

### (6) 下载测试：

当每插入一个时钟源作为待测时钟，看 8 个数码显示的数据是否和待测的时钟相吻合（要把测得的数据换算成 10 进制）。

### (7) 实验任务 1：

参考例 4.12，将频率计改为 8 位 10 进制频率计，注意此设计电路的计数器必须是 8 个 4 位的 10 进制计数

器，而不是 1 个。此外注意在测频速度上给予优化。

#### (8) 实验任务 2:

用 LPM 模块取代例 4-12-2 和例 4-12-3，再完成同样的设计任务。

#### (9) 实验任务 3:

用嵌入式锁相环 PLL 的 LPM 模块对实验系统的 50MHz 或 20MHz 时钟源分频率，PLL 的输出信号作为频率计的待测信号。注意 PLL 的输入时钟必须是器件的专用时钟输入脚，且输入频率不能低于 16MHz。

### 4.13 8 位 10 进制频率计设计

实验路径: DEMO\EDA\_DEMO\LED\_DEMO\ DEMO13\_Ftest10

#### (1) 实验目的:

根据上面的 16 进制的测频原理，设计一个 10 进制的测频实验。

#### (2) 实验原理

跟上面相同，但要计数器要设计一个 10 进制的计数器例【4-13】并用 8 个 10 进制的计数器串连成 32 位计数器，从低到高每个计算器的溢出位作为高位计数器的数据。

例【4-13】10 进制计数器

文本: CNT10

#### (3) 引脚锁定、及操作方法参考例 4-12

### 4.14 交通灯设计

实验路径: \DEMO\EDA\_DEMO\LED\_DEMO\DEMO14\_jiaotong(主系统上 LED)  
 \MODU\_DEMO\DEMO1\_jiaotong(模块交通灯)

#### (1) 实验目的

- 设计一个简单的交通灯，东西路口和南北路口分别由红、黄、绿控制了解交通灯的亮灭规律。
- 了解交通灯控制器的工作原理。
- 熟悉 VHDL 语言编程，了解实际设计中的优化方案。

#### (2) 实验原理

交通灯的显示有很多方式，如十字路口、丁字路口等，而对于同一个路口又有很多不同的显示要求，比如十字路口，车子如果只要东西和南北方向通行就很简单，而如果车子可以左右转弯的通行就比较复杂，本实验仅针对最简单的南北和东西直行的情况。根据以上情况，设计交通灯控制器，从整个方向红灯切换到绿灯需要 25S，其中绿灯转换成黄灯需 20S，其中黄灯占 5S，到绿灯共计 25S，根据这思路如下图所示：

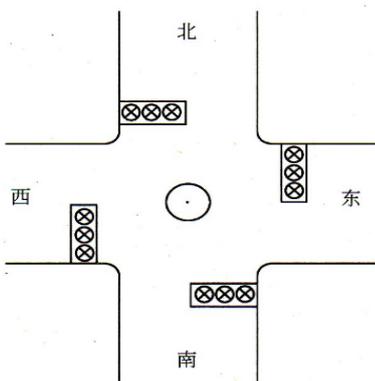


表 4-2 交通灯控制要求

东西	信号	绿灯	黄灯	红灯	
	时间	20S	5S	25S	
南北	信号	红灯		绿灯	黄灯
	时间	25S		20S	5S

图 4-23 交通灯示意图

根据以上要求，如下语言描述：

例 4-14

文本：traffic

### (3) 实验内容：

根据以上的语言描述，进行编译，仿真，进行波形分析。

### (4) 引脚锁定：

建议选择模式 5，r1、y1、g1 锁定分别锁定 D1、D2、D3(PIO8~PIO10)，代表东西方向，r2、y2、g2 锁定分别锁定 D8、D7、D6(PIO15~PIO13)，CLK 输入锁定主系统“J17”的 CLKB1 脚。

### (5) 硬件连接

这里只需要用一根单线，一端连接主系统“J17”的 CLKB1 脚，另一端连接左下脚的时钟源 1Hz，作为等待时间。

### (6) 实验现象：

1、用模式键选模式“5”

2、下载 jiaotong 中的 jiaotong.sof，到 FPGA 中，再按一次系统的复位键。

3、主系统左下方时钟源 1Hz 频率作为限时时间；

4、LED 灯 D1 D2 D3 和 D8 D7 D6 分别代表两个不同方向的红 黄 绿指示，D1 一直亮 25S，D6 亮 20S 时，跳转 D7 亮 5S，D7 结束，D8 亮一直亮 25S，D1 灭，D3 亮 20S，转 D2 亮 5S，再转换 D1 亮。这样循环转换。

注：另提供模块化交通灯模块实验模块在 \DEMO\EDA\_DEMO\MODU\_DEMO\DEMO1\_jiaotong

### (7) 实验任务：

设计一个读秒指示的交通灯显示，可以利用时钟计数器来完成。

## 4.15 多人抢答器

实验路径：\DEMO\EDA\_DEMO\LED\_DEMO\DEMO15\_qiangdaqi

### (1) 实验目的

- 1、了解电子抢答器的原理。
- 2、掌握异步时序逻辑电路的设计方法

### (2) 实验原理

从原理上来说抢答器的功能就是判断最先产生抢答信号的来自哪一条信号线，并用数码管显示出来。为判断最先产生的抢答信号的来源，首先要把所有的抢答信号都引到同一逻辑中，如在初始状态下，各个信号都出租无效状态，最先产生的抢答信号优先输出图 4-24。

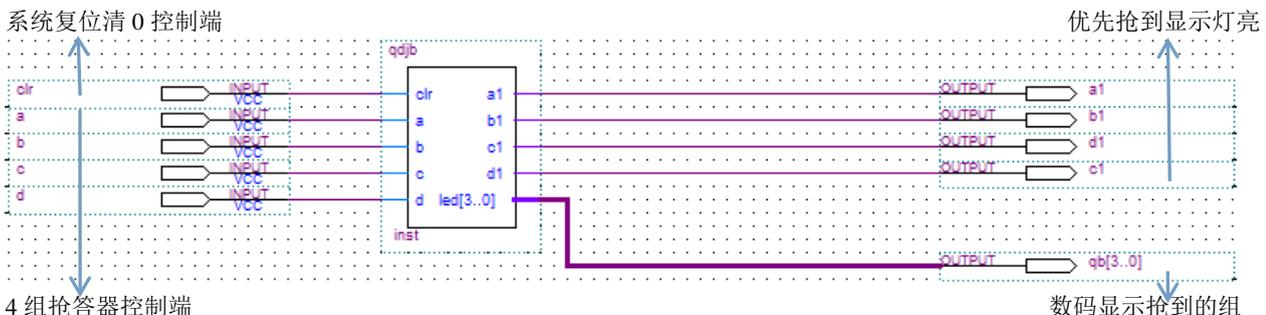


图 4-24 原理图

例 4-15

文本：qdjb

### (3) 实验内容：

根据以上设计，进行仿真，锁定引脚下载到系统中验证。

#### (4) 锁定引脚：

建议选择模式 3，键 1-键 4 锁定抢答端 a、b、c、c (PIO0-PIO3)，D1-D4 锁定显示抢答到的组 a1、b1、c1、d1 (PIO8-PIO11)，数码 1 显示抢答到的组数 QB(PIO16-PIO19)。

#### (5) 下载测试：

下载 qiangdaqi.sof 到系统中，选择模式 3，系统复位。按下键 1-4，看是否对应 LED 灯亮，并数码 1 显示对应组数；clr 清 0。

## 实验 4.16 出租车计费器

实验路径：\ DEMO\EDA\_DEMO\LED\_DEMO\DEMO16\_taxi

### (1) 实验目的

了解出租车计费器的工作原理，掌握出租车计价器的设计思路和设计方法。

### (2) 实验原理

出租车计费器一般都是按公里计费，通常是起步价 xx 元 (xx 元可以行走 x 公里)，然后再是 xx 元/公里。所以要完成一个出租车计费器，就要有两个计数单位，一个用来计公里，另外一个用来计时费用。

本实验中，规定计费的标准为：①起价为 3 公里以内 10 元 (如这时有等待时间超过 150S, 则同时要根据等待的时间来计费)；②3KM 以上按 2 元/KM 计，但每走 0.5KM, 计 1 元 (即达到 3KM 时计费为 11 元，达到 3.5KM 时计费为 12 元，达到 4KM 时计费为 13 元)；③等待时间按 1 元/150S 计，行驶过程中如车速小于 2M/s, (即 7.2KM/h,) 则计等待时间，同时要计千米数。

根据下图 4-25，计价器除去分频器以外，分为三个模块：行驶模拟模块、计费模块、显示模块。输入信号有：clk1Hz、clk10Hz、和 clk1KHz、为时钟信号。clr 为复位信号；speedup 为加速信号；speeddown 为加速信号，start 为计价器启动信号；mode 为模式切换信号。输出信号：data[23..0] 接 6 个数码管显示。

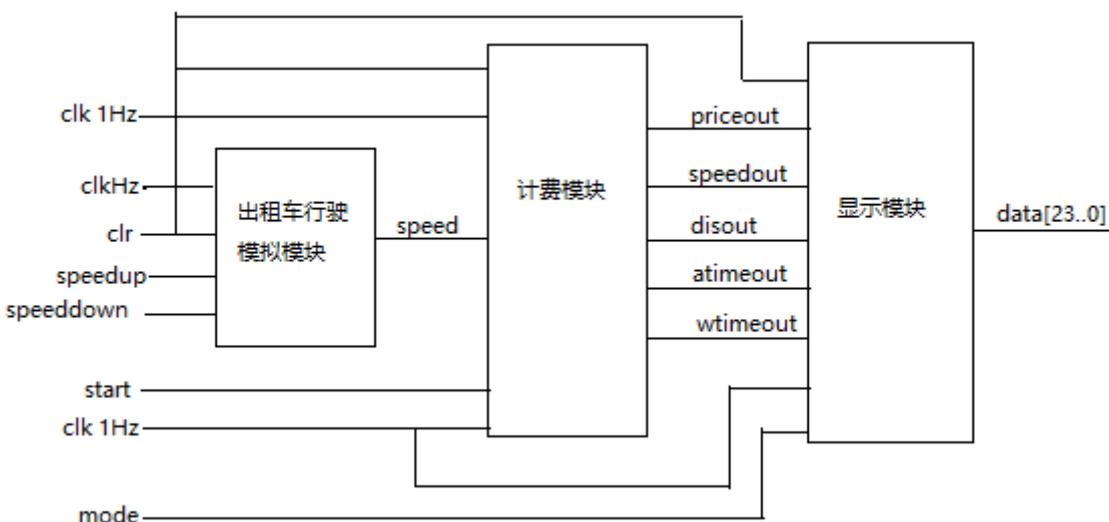


图 4-25

根据以上原理，每个模块功能写出描述语言，最终搭成一下顶层原理图 4-26

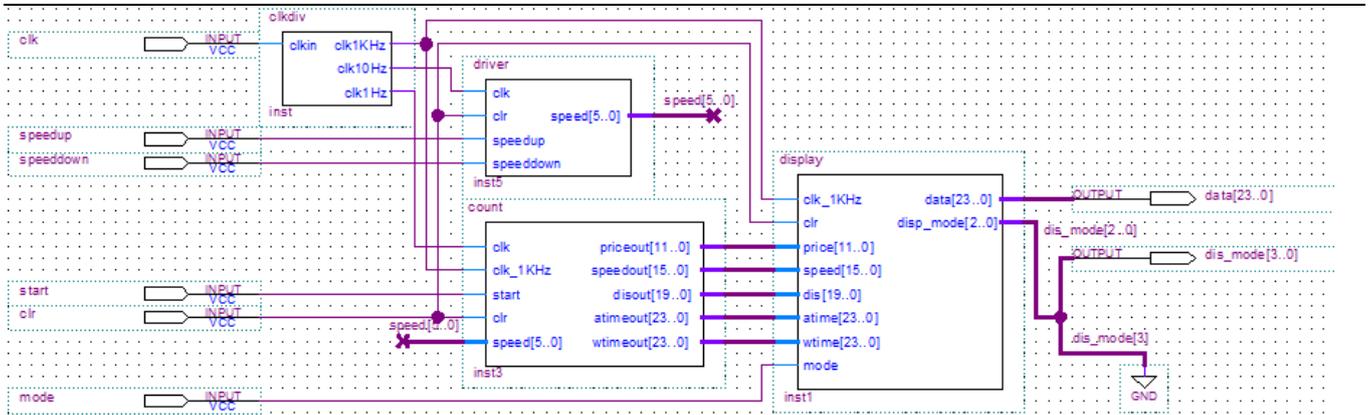


图 4-26

#### (4) 实验内容:

根据文本，测试文件，仿真波形，是否达到功能要求。

#### (5) 引脚锁定:

建议选择模式 3， Clk:时钟信号，DE 板时钟 50MHz，KX 板锁定 20M。

Start:计价器启动信号，接按键 1 (PIO0)。

Clr:复位信号：接按键 2 (PIO1)。

Speedup:加速信号，接按键 3 (PIO2)。

Speeddown:减速信号，接按键 4 (PIO3)。

Mode:显示切换按键，接按键 5 (PIO4)。

Data[23..0]在 mode 信号驱动 6 个数码管 6~1 分别接： (PIO16~PIO43)。

Dip\_mode[3..0]，显示模式，接数码管 8(PIO44~PIO47)。

根据以上引脚名和目标芯片的具体型号，查出引脚号锁定。

#### (6) 实验步骤:

1、下载\DEMO16\_taxi\taxi.sof，到 FPGA 中，复位主系统。

2、选择模式 3，按键 1~键 8 为琴键式；

#### (7) 下载测试:

(1) 按下键 2 进行复位。

(2) 通过键 3 和键 4 调整速度，观察速度上下限控制是否正常；首先按键 5，使得模式显示数码 8 显示 1，切换到显示速度数码 4—1，然后按键 3，观察速度值，应从 0 开始，按 3.6KM/h 的步长递增（长时间按住键不放，则可以实现速度调整），使得速度值增加到 63m/s=63X3.6km/h=226.8km/h；按键 4，观察速度值，应按 3.6kn/h 步长递减，按键 3，将行使速度设置值最大为 226.8km/h,相当于 3.78km/min=0.063km/S，

(3)然后按键 1，起动计价器开始计费 and 计时。

(4) 按键 5，切换显示各变量数值；

a、按键 5，使得数码 8 显示 0，切换显示计费（数码 3—1），观察起步价应为 10 元。

b、按键 5,使得数码 8 模式显示 2,切换到显示距离(数码 5—1),由于行使速度设置最大值 226.8km/h,则四十几秒后,当行使距离变为 3km 时,立刻显示计费,观察此时计费应为 11 元,继续观察,每走 0.5km 计费是否加 1 元。

c、按键 5，使得模式数码 8 显示 1，切换显示等待时间，按一下键 4，使速度减至 3.6kn/h(1m/s)，按住键 5，使得数码 8 显示模式为 4，切换到显示时间，观察等待计时，（数码管 6—1）是否正确，观察等待时间超过 150s,计费是否加 1 元。

#### (8) 实验任务:

将实验数据记录到表 4-3 中，并分析说明数据是否正确。

表 4-3 出租车计价器实验记录

	速度最大值 /km/h	起步价	距离为 3 公里的 计费	距离为 4 公里 的计费	速度小于 7.2km/h 的 等待计时	距离 /km	对应计费/元
理论值							
实测值							

## 实验 4.17 数字钟

实验路径: \DEMO\EDA\_DEMO\LED\_DEMO\DEMO17\_clock

### (1) 实验目的

掌握电子时钟的原理。

熟悉同步异步时钟时序电路和相应计数器的设计方法。

### (2) 任务要求:

- (1) 设计一个数字钟。
- (2) 具有时,分,秒计数显示功能,以 24 小时循环计时。
- (3) 具有清零,调节小时、分钟功能。
- (4) 具有整点报时功能,整 LED 灯花样显示。

### (3) 设计原理:

图 4-27,除时钟源模块以外,共有六个模块,分别是秒、分、时控制模块,另对分、时控制调节模块、整点报警模块。

秒模块,有时钟 1HZ 作为计时基准时钟,设计 60 进制的计数器,当计到 59 秒时,向分输入进位位,同时分计数,同样,计到 59 分向时输出进位位,当时、分、秒分别计到 24 整时,报时模块蜂鸣器响 10 秒,3 个发光管进行循环移位操作。

显示部分输出有 6 个数码管承担显示任务,秒、分、时分别输出 2 位十进制。

调节模块,可以对分、时进行输入调整。

根据以上原理写出语言描述例【4-17-1】、例【4-17-2】、例【4-17-3】。

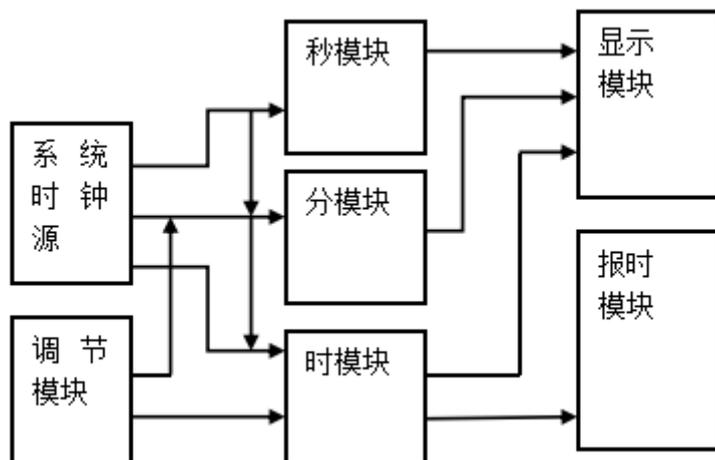


图 4-27

例【4-17-1】秒计数器模块:

文本: second

例【4-17-2】分计数器模块:

文本: minute

例【4-17-3】时计数器模块

hour

例【4-17-3】报警模块的

文本: alert

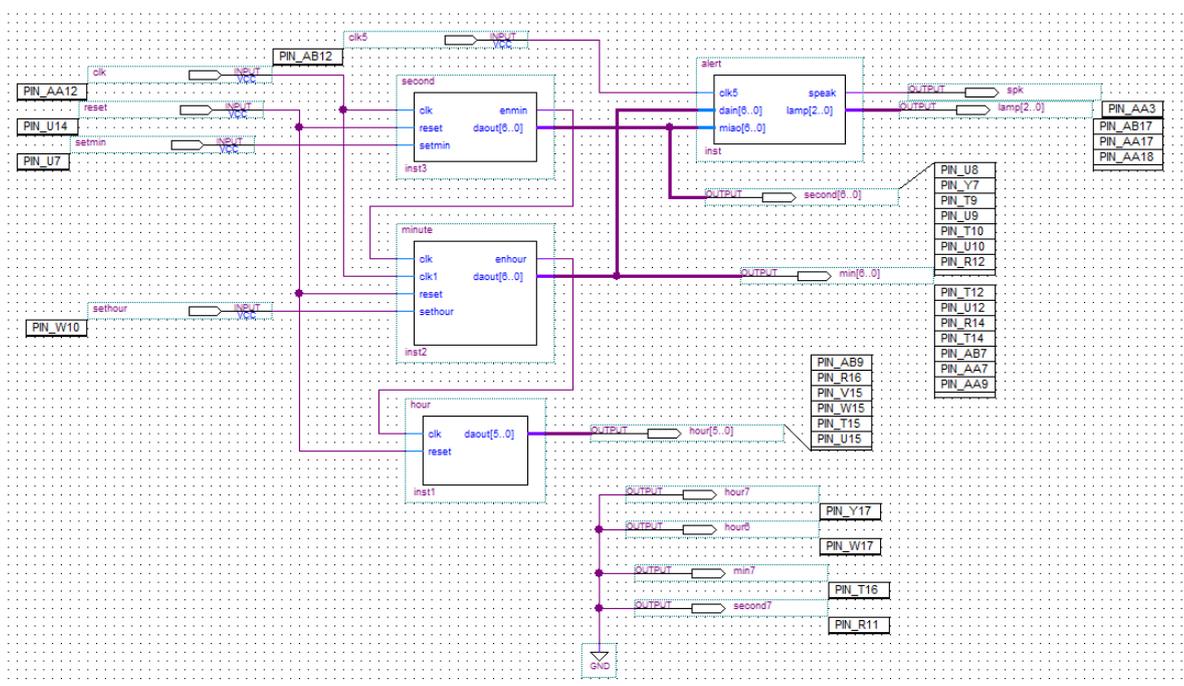


图4-28 顶层文件

#### (4) 实验内容:

根据文本，测试文件，仿真波形，是否达到功能要求。

#### (5) 引脚锁定:

选择模式7

CLK 计时基准时钟由主系统时钟源区提供1Hz，锁定 J17 的 CLKB0 端。

CLK5 报警时钟由主系统时钟源区提供10MHz，锁定 J17 的 CLKB1 端。

setmin 调节分锁定键4 (PI03)。

sethour 调节时锁定键7 (PI06)。

second[6..0] 显示秒，锁定数码管1、2 (PI016-PI023)。

min[6..0] 显示分，锁定数码管4、5 (PI024-PI031)。

hour[5..0] 显示分，锁定数码管7、8 (PI032-PI039)。

sperk 整点报时输入端锁定主系统左下方 J20，输出端锁定 J7 的 DBT1。

Lamp[2..0] 灯计时时循环闪烁锁定 D1、D2、D3 (PI040-PI042)。

#### (6) 硬件连接

准备三个杜邦单线，一根一端连接 J17 的 CLKB1 端，另一端接主系统左下方时钟源区的 1Hz，作为计时的基准时钟。第二根单线一端连接 J17 的 CLKB0 端，另一端接主系统左下方时钟源区的 1024Hz，作为正点报时的基准时钟，第三根线一端连接 J7 的 DBT1 端，另一端连接主系统左下方 J20，作为蜂鸣器整点报时的驱动。

#### (7) 下载测试:

硬件验证的方法如下：选择实验模式7；下载后，复位主系统，首先置键8为高电平，这是数码管开始显示秒的数据，同时三个发光二极管进行循环移位操作，当数码1、2计到59时，下一个为00，这时数码管4、5为1的分钟显示，数码管7、8为时的计时，当24時計完成，循环计数。当时钟为整点的时候，同时蜂鸣器持续响10秒。

键8为低电平时，时钟清零；高电平使能，键4调分操作；键7为调时操作。

#### (8) 课后任务:

设计一个日历、时钟的电子表。可以选择模式3。

## 实验 4.18 硬件电子琴

实验路径：\DEMO\EDA\_DEMO\LED\_DEMO\DEMO18\_ECHOR

### (1) 实验目的：

根据之前实验 4-8 学过的数控分配器，设计一个可以预置音阶频率字的电子琴。

### (2) 实验原理：

利用数控分频器设计硬件电子琴，当按下琴键时，扬声器发出该琴键相对应的音阶，同时数码管显示音阶数字，可选择低、中低、中高、高音的 4 档音阶。

整个系统由三个文件组成，例【4-18-1】是顶层设计文件，其内部有两个功能模块（如图 4-29 所示）：音阶发生器通过琴键选择预制数向数控分频器发出预置数，详见示例中的 Tone，预置数据对应的音阶的获得查相关资料。模块 TONE 是音阶发生器，当 8 位发声控制输入 INDEX 中某一位为高电平时，则对应某一音阶的数值将从端口 TONE 输出，作为获得该音阶的分频预置值；同时由 CODE 输出对应该音阶简谱的显示数码，如‘5’，并由 HIGH 输出指示音阶高 8 度显示。Tone 其语句结构只是类似与真值表的纯组合电路描述，其中的音阶分频预置值，如  $Tone \leq 1290$  是根据产生该音阶频率所对应的分频比获得的。此模块当中做了 4 组不同音频的输出，通过一个计数器来选择，当 00 分别是低音；当选择 01 时，是中音，但选择 10 的时候是高音，其他都为低八度音。

模块 SPEAKER 中的主要电路是一个数控分频器，它由一个初值可预置的加法计数器构成，当模块 SPEAKER 由端口 TONE 获得一个 2 进制数后，将以此值为计数器的预置数，对端口 CLK20MHZ 输入的频率进行分频，之后由 SPKOUT 向扬声器输出发声。

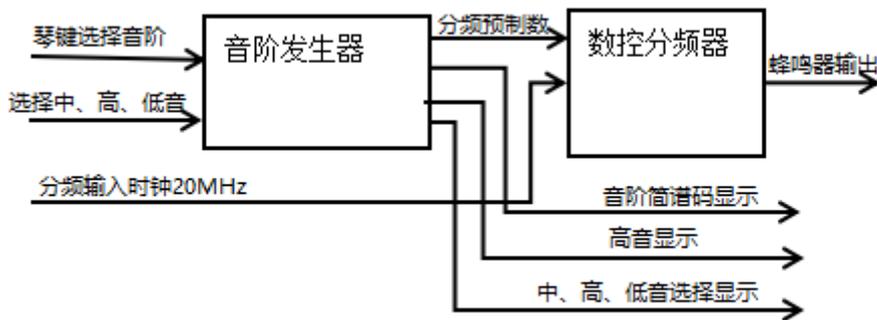


图 4-29

### 1、顶层模块的设计(文本：TOP)模块的设计

语言描述采用的是自顶向下的设计方式，顶层模块由音调发生器（tone）和数控分频器（speaker）两个模块组成。图 4-29 即是顶层设计原理图。其中音调发生器部分又包括了键盘编码。设置八个按键(Index1)输入信号，再对其进行编码，输出的 Tone 都是 11 位二进制数，对应数控分频模块的输入。数控分频模块根据输入的预置数 Tone1，对基准脉冲进行相关分频后，输出对应音阶的频率，驱动蜂鸣器发音。

### 2、音调发生器(文本：tone)模块的设计

音调发生器的作用是产生获得音阶的分频预置值。当 8 位发声控制输入 index 中的某一位为低电平时，则对应某一音阶的数值将以端口 tone 输出，作为获得该音阶的分频预置值，该值作为数控分频器的输入，来对 20MHz 的脉冲进行分频，由此得到每个音阶相应的频率，例如输入  $index="11111101"$ ，即对应的按键是 2，产生的分频系数便是 912；由 code 输出对应该音阶简谱的显示数码；由 high 输出指示音阶高 8 度的显示，低电平有效。

对于 tone 的计算很重要，tone 是根据产生该音阶频率所对应的分频比获得的。此值是通过查表和计算所到，查表我们可以知道每个乐谱符号的频率，然后计算出要得到该乐谱的频率所需的分频系数。例如音符‘1’的发音，通过查表得它的频率是 785HZ 左右，我们要计算 tone 的值可由公式：

音符频率=振荡频率/(16#7FF#-tone 的值)

振荡频率是经过分频之后得到的,在此是 1MHZ,由公式可计算出 tone 的值为:773。

### 3、数控分频模块(文本: speaker)的设计

数控分频模块的目的是对基准脉冲分频,得到 1,2,3,4,5,6,7 七个音符对应频率。该模块的描述中包含了三个进程。首先对 20MHz 的基准脉冲进行分频得到 1MHz 的脉冲,然后按照 tone1 输入的分频系数对 1MHz 的脉冲再次分频,得到的便是所需要的频率。而第三个进程的作用是在音调输出时再进行二分频,将脉冲展宽,同时使其占空比为 50%,以使扬声器有足够功率发音。

例【4-18】顶层文件

文本: TOP

#### (4) 实验内容:

根据文本,测试文件,仿真波形,是否达到功能要求。

#### (5) 引脚锁定:

选择模式3

CLK 计时基准时钟由主系统时钟源区提供20MHz,锁定 J17的 CLKB1端。

INDEX1[6..0]琴键发生锁定键1-键7 (PI00-PI06)。

HORL1中、高、低音选择键锁定键8(PI07)。

CODE1[3..0]显示简谱码锁定数码管1 (PI016-PI019)。

NUME1[3..0]显示选择中、高、低音的显示锁定数码管8 (PI044-PI047)。

SPOUT 驱动蜂鸣器发音输入端锁定主系统左下方 J20,输出端接 J7的 DBT1。

HIGH1高音显示锁定 D8 (PI015)。

根据以上引脚名,查表锁定引脚。

#### (6) 硬件连接

准备 2 个杜邦单线,一根一端连接 J17 的 CLKB1 端,另一端接主系统左下方时钟源区的 20MHz,作为分频基准时钟,第 2 根线一端连接 J7 的 DBT1 端,另一端连接主系统左下方 J20,作为蜂鸣器发音。

#### (7) 下载测试:

硬件验证的方法如下:选择实验模式3;下载后,复位主系统,首先按动键8,数码管显示0、1、2、3。0 档低音,1档中音,2档高音,再按动键1-键7发出不同的音阶。当按键按下去时,数码1显示对应的简谱码,D8 显示高音

## 4.19 乐曲硬件演奏电路设计

实验路径: \DEMO\EDA\_DEMO\LED\_DEMO\DEMO19\_SONGER

#### (1) 实验目的:

应用硬件描述语言,设计一个乐曲硬件演奏电路,它能将一首预先设置存储好的乐曲自动播放出来,除此之外,也能够通过按键的方式输入音符,使其具备简易电子琴的功能。通过此项研究,能够深切的体会利用 EDA 工具开发的优越性,在此基础上,对乐曲硬件演奏电路功能进行丰富,具有一定的社会实用性。

#### (2) 实验原理:

主系统由 3 个模块组成,图 4-30 是顶层设计文件,其内部有 3 个功能模块: TONETABA、NOTETABS 和 SPEAKER。。

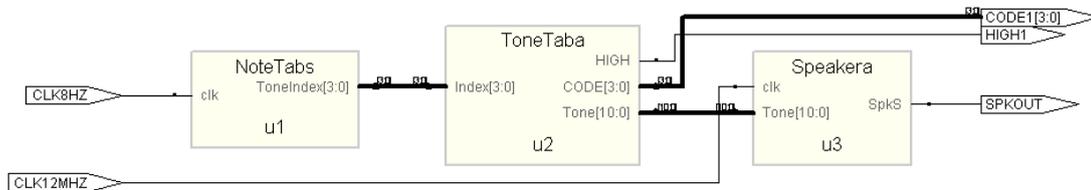


图 4-30 硬件乐曲演奏电路结构 (Synplify 综合)

与利用微处理器 (CPU 或 MCU) 来实现乐曲演奏相比, 以纯硬件完成乐曲演奏电路的逻辑要复杂得多, 如果不借助于功能强大的 EDA 工具和硬件描述语言, 仅凭传统的数字逻辑技术, 即使最简单的演奏电路也难以实现。本实验设计项目作为“梁祝”乐曲演奏电路的实现。我们知道, 组成乐曲的每个音符的发音频率值及其持续的时间是乐曲能连续演奏所需的两个基本要素, 问题是如何来获取这两个要素所对应的数值以及通过纯硬件的手段来利用这些数值实现所希望乐曲的演奏效果。图 4-30 中, 模块 U1 类似于弹琴的人的手指; U2 类似于琴键; U3 类似于琴弦或音调发声器。

下面首先来了解图 4-30 的工作原理例【4-19-1】:

1、音符的频率可以由图 4-30 中的 SPEAKER 【4-19-2】获得, 这是一个数控分频器 (其详细工作原理可参考实验 4.8)。由其 clk 端输入一具有较高频率 (这里是 12MHz) 的信号, 通过 SPEAKER 分频后由 SPKOUT 输出, 由于直接从数控分频器中出来的输出信号是脉宽极窄的脉冲式信号, 为了有利于驱动扬声器, 需另加一个 D 触发器以均衡其占空比, 但这时的频率将是原来的 1/2。SPEAKER 对 clk 输入信号的分频比由 11 位预置数 Tone[10..0] 决定。SPKOUT 的输出频率将决定每一音符的音调, 这样, 分频计数器的预置值 Tone[10..0] 与 SPKOUT 的输出频率, 就有了对应关系。例如在 TONETABA 模块中若取 Tone[10..0]=1036, 将发音符为“3”音的信号频率。

2、音符的持续时间须根据乐曲的速度及每个音符的节拍数来确定, 图 4-30 中模块 TONETABA 【4-19-3】的功能首先是为 SPEAKER 提供决定所发音符的分频预置数, 而此数在 SPEAKER 输入口停留的时间即为此音符的节拍值。模块 TONETABA 是乐曲简谱码对应的分频预置数查表电路, 其中设置了“梁祝”乐曲全部音符所对应的分频预置数, 共 13 个, 每一音符的停留时间由音乐节拍和音调发生器模块 NOTETABS 的 clk 的输入频率决定, 在此为 4Hz。这 13 个值的输出由对应于 TONETABA 的 4 位输入值 Index[3..0] 确定, 而 Index[3..0] 最多有 16 种可选值。输向 TONETABA 中 Index[3..0] 的值 ToneIndex[3..0] 的输出值与持续的时间由模块 NOTETABS 决定。

3、在 NOTETABS 【4-19-4】中设置了一个 8 位二进制计数器 (计数最大值为 138), 作为音符数据 ROM 的地址发生器。在 4-18 电子琴示例基础上, 相当于原来用手工弹奏的音阶, 我们改成自动输出音阶节拍。这个计数器的计数频率选为 4Hz, 即每一计数值的停留时间为 0.25 秒, 恰为当全音符设为 1 秒时, 四四拍的 4 分音符持续时间。例如, NOTETABS 在以下的逻辑描述中, “梁祝”乐曲的第一个音符为“3”, 此音在逻辑中停留了 4 个时钟节拍, 即 1 秒时间, 相应地, 所对应的“3”音符分频预置值为 1036, 在 SPEAKER 的输入端停留了 1 秒。随着 NOTETABS 中的计数器按 4Hz 的时钟速率作加法计数时, 即随地址值递增时, 音符数据 ROM 中的音符数据将从 ROM 中通过 ToneIndex[3..0] 端口输向 TONETABA 模块, “梁祝”乐曲就开始连续自然地演奏起来了。

4、定制 NoteTabs 模块中的音符数据 ROM “music” 【4-19-5】。该 ROM 中的音符数据已列, 注意该例数据表中的数据位宽、深度和数据的表达类型。此外, 为了节省篇幅, 例中的数据都横排了, 实用中应该以每一分号为一行来展开, 否则会出错。最后对该 ROM 进行仿真, 确认音符数据已经进入 ROM 中。

【例 4-19-1】 硬件演奏电路顶层设计

文本: Songer

【例 4-19-2】 音符频率

文本: SpeakeRa

【例 4-19-3】 音符节拍

文本: TONETABA

**【例 4-19-4】** 音符地址发生器

文本: NoteTabs

**【例 4-19-5】**

WIDTH = 4 ; -- “梁祝” 乐曲演奏数据

DEPTH = 256 ;

ADDRESS\_RADIX = DEC ;

DATA\_RADIX = DEC ;

CONTENT BEGIN --注意, 以下的数据排列方法只是为了节省空间, 实用文件中要展开以下数据, 每一组占一行

```

00: 3 ; 01: 3 ; 02: 3 ; 03: 3; 04: 5; 05: 5; 06: 5;07: 6; 08: 8; 09: 8;
10: 8 ; 11: 9 ; 12: 6 ; 13: 8; 14: 5; 15: 5; 16: 12;17: 12;18: 12; 19:15;
20:13 ; 21:12 ; 22:10 ; 23:12; 24: 9; 25: 9; 26: 9; 27: 9; 28: 9; 29: 9;
30: 9 ; 31: 0 ; 32: 9 ; 33: 9; 34: 9; 35:10; 36: 7; 37: 7; 38: 6; 39: 6;
40: 5 ; 41: 5 ; 42: 5 ; 43: 6; 44: 8; 45: 8; 46: 9; 47: 9; 48: 3; 49: 3;
50: 8 ; 51: 8 ; 52: 6 ; 53: 5; 54: 6; 55: 8; 56: 5; 57: 5; 58: 5; 59: 5;
60: 5 ; 61: 5 ; 62: 5 ; 63: 5; 64:10; 65:10; 66:10; 67:12; 68: 7; 69: 7;
70: 9 ; 71: 9 ; 72: 6 ; 73: 8; 74: 5; 75: 5; 76: 5; 77: 5; 78: 5; 79: 5;
80: 3 ; 81: 5 ; 82: 3 ; 83: 3; 84: 5; 85: 6; 86: 7; 87: 9; 88: 6; 89: 6;
90: 6 ; 91: 6 ; 92: 6 ; 93: 6; 94: 5; 95: 6; 96: 8; 97: 8; 98: 8; 99: 9;
100:12 ;101:12 ;102:12 ;103:10;104: 9;105: 9;106:10;107: 9;108: 8;109: 8;
110: 6 ;111: 5 ;112: 3 ;113: 3;114: 3;115: 3;116: 8;117: 8;118: 8;119: 8;
120: 6 ;121: 8 ;122: 6 ;123: 5;124: 3;125: 5;126: 6;127: 8;128: 5;129: 5;
130: 5 ;131: 5 ;132: 5 ;133: 5;134: 5;135: 5;136: 0;137: 0;138: 0;
END ;

```

**(3) 实验内容:**

根据文本, 测试文件, 编译。

**(5) 引脚锁定:**

选择模式3

CLK8Hz 节拍时钟输入端锁定输入 J17的 CLKB0 (查表)。

CLK12MHz 音调时钟锁定主系统时钟源区提供20MHz, 输入 JP7的 CLKB1 (查表) 端。

CODE[3..0]显示简谱码锁定数码管1 (PI016-PI019)。

SPOUT 驱动蜂鸣器发音输入端锁定主系统左下方 J20, 输出端接 J7的 DBT1。

HIGH1高音显示锁定 D8 (PI015)。

根据2.2\2.4查表锁定引脚

**(6) 硬件连接**

准备 3 个杜邦单线, 第一根一端连接 J17 的 CLKB1 端, 另一端接主系统左下方时钟源区的 20MHz, 音调分频时钟。第二根一端连接 J17 的 CLKB0 端, 另一端接主系统左下方时钟源区的 4Hz, 节拍时钟, 第三根线一端连接 J7 的 DBT1 端, 另一端连接主系统左下方 J20, 作为蜂鸣器发音。

**(7) 下载测试:**

硬件验证的方法如下: 选择实验模式3; 下载后, 复位主系统, 乐曲自动演奏, 数码1显示对应的简谱码, D8显示高音。

## 4.20 5 首乐曲演奏

实验路径: \DEMO\EDA\_DEMO\LED\_DEMO\ DEMO20\_MUSICTEA

**(1) 实验目的:**

通过 4.19 实验原理, 添加 5 首演奏电路。

## (2) 实验原理:

以上实现单首演奏电路原理这里不再重复。我们知道，添加乐曲的频率，也就是简谱对应的频率在 NOTETABA 模块中，我们只要做五个这样的模块，每个模块对应的乐曲。填入新的乐曲，如“采茶舞曲”、或其它熟悉的乐曲在 Songer 中，连接五个不同乐曲的 NOTETABA 文件例【4-20】做好以后，再接入一个 5 进制的计数器，对播放乐曲进行选择如图 4-31。

例【4-20】顶层文件

文本: Songer

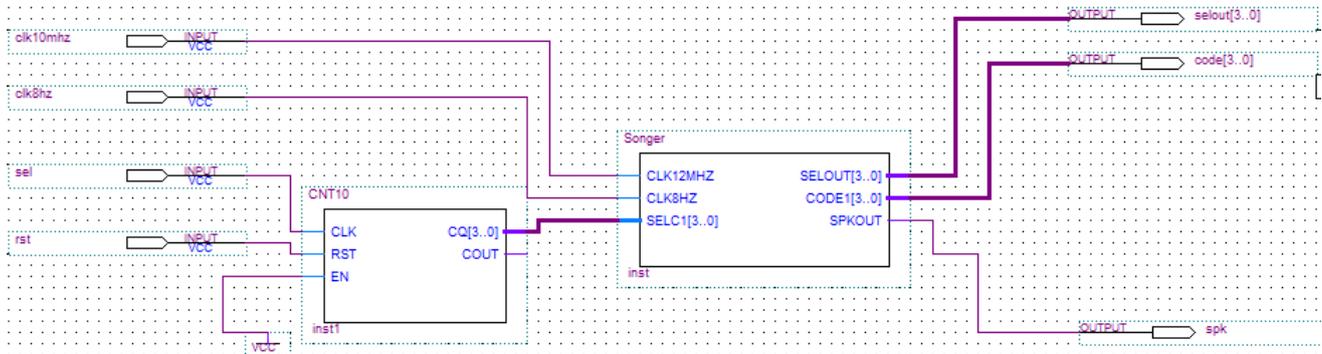


图 4-31 多首乐曲播放原理图

图 4-31 中，CNT10 是一个计数器模块，用来选择不同乐曲，sel 控制端用来计数输入的计数时钟，每输入一次，对应 selout 显示乐曲的数目，rst 系统复位，按一下清 0。

## (3) 实验内容:

根据文本，测试文件，编译。

## (5) 引脚锁定:

选择模式3

CLK8Hz 节拍时钟输入端锁定 J17的 CLKB0 (查表)。

CLK12MHz 音调时钟由主系统时钟源区提供20MHz 输入输入锁定 J17的 CLKB1 (查表) 端。

CODE[3..0]显示简谱码锁定数码管1 (PI016-PI019)。

SPOUT 驱动蜂鸣器发音输入端锁定主系统左下方 J20, 输出端接 JP7的 DBT1。

HIGH1高音显示锁定 D8 (PI015)。

sel, 曲目选择键锁定键2 (PI00)。

rst 复位键锁定键1 (PI01)。

根据2.2\2.4查表锁定引脚

## (6) 硬件连接

准备 3 个杜邦单线，第一根一端连接 J17 的 CLKB1 端，另一端接主系统左下方时钟源区的 20MHz，音调分频时钟。第二根一端连接 J17 的 CLKB0 端，另一端接主系统左下方时钟源区的 4Hz，节拍时钟，第三根线一端连接 J7 的 DBT1 端，另一端连接主系统左下方 J20，作为蜂鸣器发音。

## (7) 下载测试:

硬件验证的方法如下：选择实验模式3；下载 Block1.sof 文件后，复位主系统，再按键1复位，键2选择曲目，数码1显示对应曲目，数码8显示对应的简谱码，D8显示高音。

对应曲面：

1、梁祝； 2、采茶舞曲； 3、敖包相会； 4、没有共产党就没有新中国； 5、十送红军。

## (8) 实验任务:

1、利用 ROM 文件争取可以在一个 ROM 装上多首歌曲，可手动或自动选择歌曲。

## 4.21 乒乓球游戏电路设计

实验路径: \DEMO\EDA\_DEMO\LED\_DEMO\DEMO21\_TENNIS

### (1) 实验目的:

了解乒乓球游戏机的工作原理,学习乒乓球游戏电路设计方法。  
掌握复杂数字系统的设计思路和方法。

### (2) 实验要求:

设计一个乒乓球游戏,实现如下功能:

①使用 8 个发光二极管 D1~D8 模拟乒乓球运动,使用 2 个琴键式按键分别模拟左右选手的乒乓球拍,使用数码管显示双方得分情况。

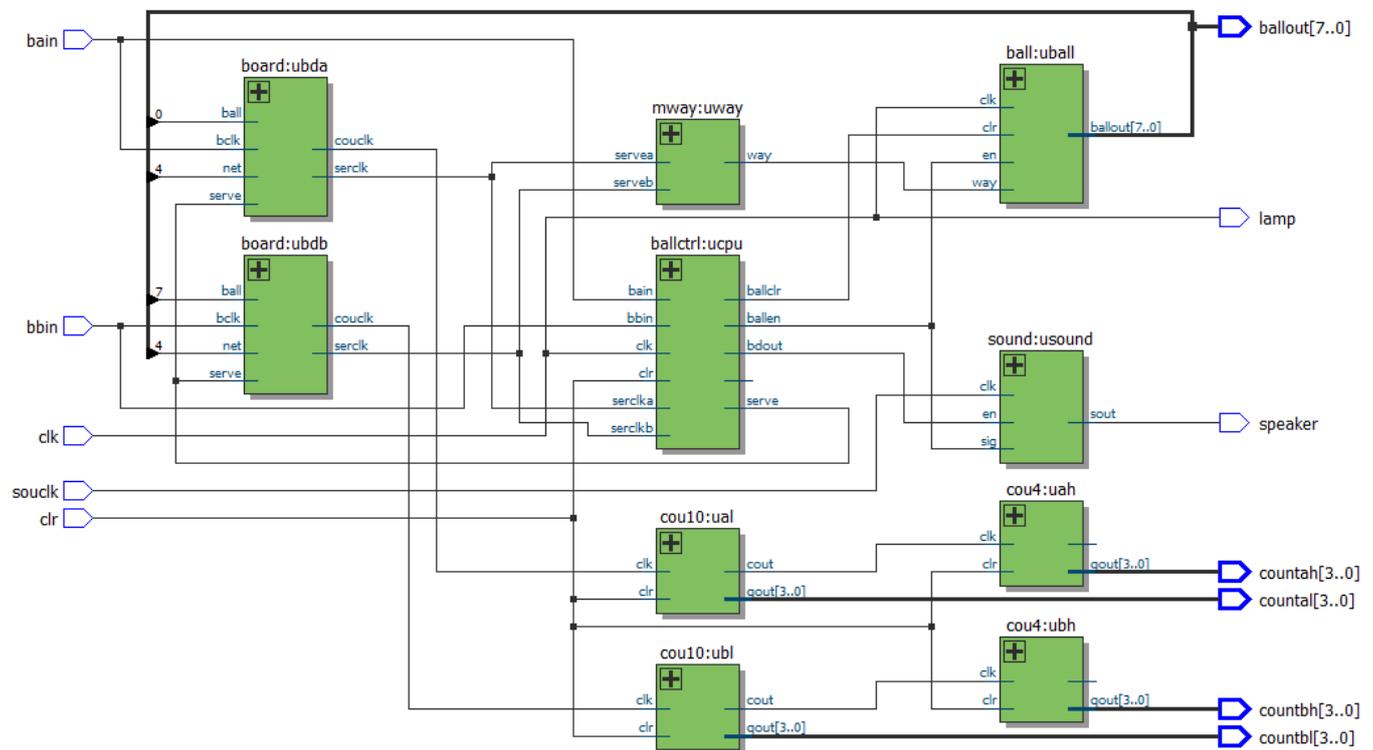


图 4-32 RTL 级电路

### (3) 实验原理:

8 个 VHDL 文件是一个乒乓球游戏电路的完整设计。其中,模块 tennis 是顶层设计例【4-21-1】,在 quartusII 中可设其为工程文件;

board 例【4-21-2】是乒乓板接球控制模块,甲乙双方各一个,相当于球拍,即当发光管亮到最后一个的瞬间,若检测到对应的表示球拍的键的信号,立即将“球”反向运行,如果此瞬间没有接到键信号,将给出错鸣叫,同时为对方记 1 分,并将记分显示出来;

ball 例【4-21-4】是模拟乒乓球行进路径的发光管亮灯控制模块,在游戏中,以一排发光管交替发光指示乒乓球的行进路径,其行进的速度可由输入的时钟信号 clk 控制;

cou4 例【4-21-6】和 cou10 例【4-21-5】分别是甲乙双方失球计数器的高低位计数模块;

mway 例【4-21-7】是乒乓球行进方向控制模块,主要由发球键控制;

sound 例【4-21-3】是失球提示发声模块。

ballctrl 例【4-21-8】,为总控制模块,接收甲乙双方发来的球进行处理,对乒乓球运行的路线清 0 和使能,并向接球对是否鸣叫发出信号。

例【4-21-1】乒乓球游戏顶层文件

文本: TENNIS

例【4-21-2】乒乓拍模块

文本: board

例【4-21-3】

文本: sound

例【4-21-4】乒乓球灯模块

文本: ball

例【4-21-5】十进制计数器用来做失球低位计数

文本: cou10

例【4-21-6】四进制计数器用来做失球高位计数

文本: cou4

例【4-21-7】乒乓球前进方向产生模块

文本: mway

【例 4-21-8】总控制模块

文本: ballctrl

### (3) 实验内容:

根据文本, 测试文件, 编译。

### (5) 引脚锁定:

选择模式3

souclk 蜂鸣器时钟输入端锁定 J17的 CLKB0端 (查表)。

clk 球运行轨迹时钟由主系统时钟源区提供4Hz, 输入 J17的 CLKB1 (查表) 端。

bain 甲方发球按键8 (PI07)

bbin 乙方发球按键1 (PI00)

clr 清0按键67 (PI06)

ballout[7..0] 球运行轨迹接 D1~D8 (PI08~PI015)。

countah[3..0]countal[3..0] 显示 a 得分, 分别接数码管3 (PI027-PI024) 数码管2 (PI023-PI020)。

countbh[3..0]countbl[3..0] 显示 b 得分, 分别接数码管7 (PI043-PI040) 数码管6 (PI039-PI036)

sperker 驱动蜂鸣器发音输入端接主系统左下方 J20, 输出端接 J7的 DBT1。

根据2.2\2.4查表锁定引脚

### (6) 硬件连接

准备3个杜邦单线, 第一根一端连接 JP、17的 CLKB1端, 另一端接主系统左下方时钟源区的4Hz, 球运行速度时钟。第二根一端连接 J17的 CLKB0端, 另一端接主系统左下方时钟源区的1024Hz, 蜂鸣器时钟, 第三根线一端连接 J7的 DBT1端, 另一端连接主系统左下方 J20, 作为蜂鸣器发音。

### (7) 下载测试:

硬件验证的方法如下: 选择实验模式3; 下载后, 复位主系统, 键7复位。分别按动键1模拟一方发球, 当D1发光管顺着向右运行发送到D8时, 立即按动键8, 否则判对方得一分, 数码3、2累计分或数码7、6累计分。如此往复, 看谁得分多谁赢。

## 4.22 八位移位相加硬件乘法器设计 参考: 《EDA技术实用教程》第五版第四章

实验路径: \DEMO\EDA\_DEMO\LED\_DEMO\DEMO22\_MULT18X8

### (1) 实验目的:

学习应用移位相加原理设计8位乘法器。

### (2) 实验原理:

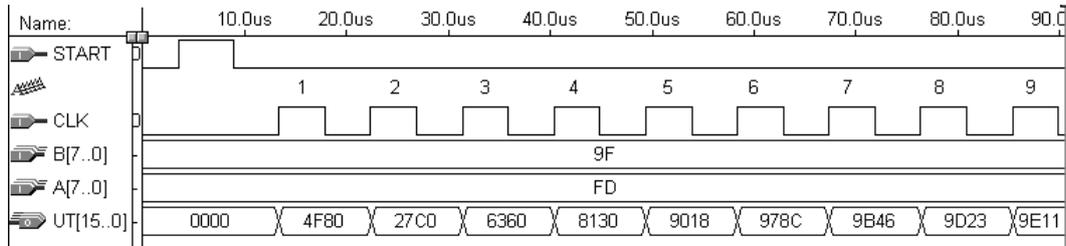


图 4-33 8 位移位相加乘法器运算逻辑波形图

该乘法器是由 8 位加法器构成的以时序方式设计的 8 位乘法器。原理是：乘法通过逐项移位相加来实现相乘。从被乘数的最低位开始，若为 1，则乘数左移后与上一次的和相加；若为 0，左移后以全零相加，直至被乘数的最高位。从图 4-33 的逻辑图及其乘法操作时序(示例中的相乘数为 9FH 和 FDH)上可以清楚地看出此乘法器的工作原理。为了更好了解其工作原理，图 3-33 中没有加入控制电路(例 4-22-5)。图 4-33 中，START 信号的上跳沿及其高电平有两个功能，即 16 位寄存器清零和被乘数 A[7..0]向移位寄存器 SREG8B 例【4-22-1】加载；它的低电平则作为乘法使能信号。CLK 为乘法时钟信号。当被乘数被加载于 8 位右移寄存器 SREG8B 后，随着每一时钟节拍，最低位在前，由低位至高位逐位移出。当为 1 时，1 位乘法器 ANDARITH 例【4-22-3】打开，8 位乘数 B[7..0]在同一节拍进入 8 位加法器，与上一次锁存在 16 位锁存器 REG16B 例【4-22-4】中的高 8 位进行相加 ADDER8B 例【4-22-2】，其和在下一时钟节拍的上升沿被锁进此锁存器。而当被乘数的移出位为 0 时，与门全零输出。如此往复，直至 8 个时钟脉冲后，最后乘积完整出现在 REG16B 端口。在这里，1 位乘法器 ANDARITH 的功能类似于 1 个特殊的与门，即当 ABIN 为‘1’时，DOUT 直接输出 DIN，而当 ABIN 为‘0’时，DOUT 输出全“0000000”。

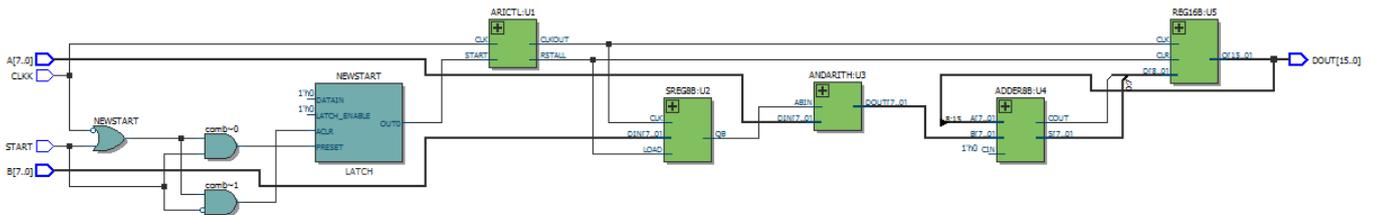


图 4-34 RTL 生成电路

8 位移位相加原理构成的乘法器比用组合电路直接设计的同样功能的电路的资源(逻辑宏单元 LCs)耗用要小许多，由编译报告可知，前者是 52，后者是 169。从波形图图 4-33 可见，当 9FH 和 FDH 相乘时，第 1 个时钟上升沿后，其移位相加的结果(在 REG16B 端口)是 4F80H，第 8 个时钟上升沿后，最终相乘结果是 9D23H。

乘法器的各功能块的 VHDL 描述如下：

例【4-22-1】 8 位右移寄存器

文本：SREG8B

例【4-22-2】 8 位加法器

文本：ADDER8B

例【4-22-3】 1 位乘法器

文本：ANDARITH

例【4-22-4】 6 位锁存器/右移寄存器

文本：REG16B

例【4-22-5】 移位相加

文本：ARICTL

例【4-22-6】 8 位乘法器顶层设计

文本：MULTI8X8

### (3) 实验仿真：

根据给出的乘法器逻辑原理图及其各模块的 VHDL 描述，在 QuartusII 上完成全部设计，包括编辑、编译、综合和仿真操作等。以 87H 乘以 F5H 为例，进行仿真，对仿真波形作出详细解释，包括对 8 个工作时钟节拍中，每一节拍乘法操作的方式和结果，对照波形图给以详细说明，根据顶层设计例【4-22-6】，结合图 4-33，画出乘法器的详细电路原理框图。

#### (4) 引脚锁定:

选择模式1

CLKK 计算时钟输入端锁定 J17的 CLKB1端 (查表)。

A[7..0]被乘数锁定键1、2(PI07-PI00)。

B[7..0]乘数锁定键1、2(PI015-PI08)。

START 开始计算开始锁定键7 (PI048)

DOUT[15..0]计算结果显示锁定数码5~数码8。

#### (5) 硬件连接

准备 1 个杜邦单线, 第一根一端连接 J17 的 CLKB1 端, 另一端接主系统左下方时钟源区的 4Hz 或更高时钟。此乘法器的控制模块, 接受实验系统上的连续脉冲, 如 4Hz, 当给定启动/清 0 信号后, 能自动发出 CLK 信号驱动乘法运算, 当 8 个脉冲后自动停止。或时钟用键 8 代替, 把时钟改为 PIO49(查表), 连续按动 8 次, 看计算时数码管的变化和仿真做比较。

#### (6) 下载测试:

编程下载, 进行实验验证。实验电路选择 No.1, 8 位乘数用键 2、键 1 输入; 8 位被乘数用键 4 和键 3 输入; 16 位乘积可由 4 个数码管 (数码管 8、7、6、5) 显示; 键 7 输入 START (注意, START 由高到低是清 0, 由低到高电平是允许乘法计算)。详细观察每一时钟节拍的运算结果, 并与仿真结果进行比较。

乘法时钟连接实验系统上的连续脉冲, 可利用实验箱左下脚 CLK, 设计一个此乘法器的控制模块, 接受实验系统上的连续脉冲, 如启动/清 0 信号后, 能自动发出 CLK 信号驱动乘法运算, 当 8 个脉冲后自动停止。

## 4.23 8 位动态扫描设计

实验路径: \DEMO\EDA\_DEMO\MODU\_DEMO\DEMO2\_SCAN8\_LED\DEMO2\_seg\_led\_dynamic\_1(秒表)

#### (1) 实验目的:

学习硬件扫描显示电路的设计。

#### (2) 实验原理:

图 4-35 所示的是 8 位数码扫描显示电路, 其中每个数码管的 8 个段 h、g、f、e、d、c、b、a (h 是小数点) 都分别连在一起, 8 个数码管分别由 8 个选通信号 k1~k8 来选择。被选通的数码管显示数据, 其余关闭。如在某一时刻, k3 为高电平, 其余选通信号为低电平, 这时仅 k3 对应的数码管显示来自段信号端的数据, 而其他 7 个数码管呈现关闭状态。根据这种电路状况, 如果希望在 8 个数码管显示希望的数据, 就必须使得 8 个选通信号 k1~k8 分别被单独选通, 同时在段信号输入口加上希望该对应数码管上显示的数据, 于是随着选通信号的扫变, 就能实现扫描显示的目的。例 4-23

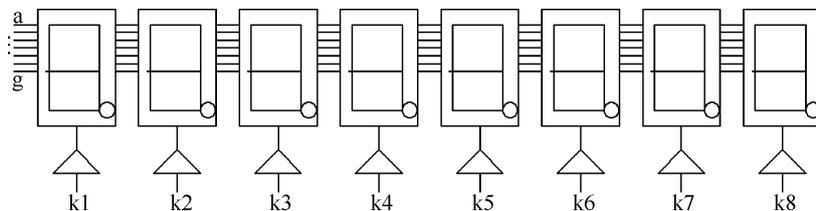


图 4-35 8 位数码扫描显示电路

例【4-23】

文本: SCAN\_LED

#### (3) 实验仿真:

根据描述语言进行仿真测试, 分析波形



图 4-36 动态扫描模块



图 4-37 主系统连接口

**(4) 硬件准备:**

这里需要一块 8 位动态扫描模块在康芯系统上进行硬件测试：图 4-36 是康芯提供的一个 8 位动态扫描模块，在本讲义第二章中介绍过此模块，提供 8 组七段译码动态扫描数码，左边十芯口是数码段的接口端，分别是 a、b、c、d、e、f、g、p(小数点)，8 个数码管串行在一起，右边是 8 数码的位控制端。只要对位进行选择，当点亮某个数码管，只要把此数码管置‘1’，其余置‘0’。此数码管就被点亮，再对此数码管的七段进行译码。此数码管显示你想要的数字。原理图请参考文件夹“原理图\SCAN8D.pdf”。

**(4) 引脚锁定:**

引脚锁定跟左半边的选择模式不一样，引脚要通过连接线连接，如图4-37，在主系统上提供了6组十芯、1组14芯外接口，这些接口是 FPGA 的 IO 口。注意：这里除了 J6、J7口以外，其他外接口都和模式电路有冲突。如果没有用到模式电路可选择其他口，优先选择 J6、J7口。

此模块插在 KX-CDS 的右下方 4 组插座，其中任意一组都可以插扩展模块。IO 口锁定这里我们选择 J6、J7 作为位和数码管译码控制端。主系统每个十芯口有 8 个 IO 口，刚好对应扫描模块位和七段译码 8 个端口，注意：根据十芯座的缺口方向为基准对应引脚，主系统缺口朝上，扩展板缺口朝下，所以引脚刚好相反方向。

所以引脚分别对应为： 表 4-2

示例引脚名	主系统十芯座标的引脚名	动态扫描模块引脚名
CLK(时钟控制)	J17: CLKB0	主系统左下端的时钟源
BT[0]	J7: DB10	D1(位选 1)
BT[1]	J7: DB11	D2(位选 2)
BT[2]	J7: DB12	D3(位选 3)
BT[3]	J7: DB13	D4(位选 4)
BT[4]	J7: DBT0	D5(位选 5)
BT[5]	J7: DB14	D6(位选 6)
BT[6]	J7: DBT1	D7(位选 7)
BT[7]	J7: DB15	D8(位选 8)
SG[0]	J6: DB9	a 段
SG[1]	J6: DB8	b 段
SG[2]	J6: DB7	c 段
SG[3]	J6: DB6	d 段
SG[4]	J6: DB5	e 段
SG[5]	J6: DB3	f 段
SG[6]	J6: DB1	g 段

由于我们主系统可以配不同型号的 FPGA 型号，所以我们在主系统上统一了标注引脚名（2.4 节），再根据引脚名查出对应的引脚号，本实验引脚号对应如图 4-38：

## 2.4. FPGA 核心板扩展至康芯主系统引脚对照表（模块类）

（此表是针对扩展板提供的 IO 口查找表，根据主系统上扩展的十芯或十四芯口的标号对应查找引脚号）

主系统扩展标号	KX-10CE5 5 板	KX-4CE55 板	KX-4C6/ 10E 板	DE0板	DE0-CV 板	DE1-SOC 板	DE10-n ane	模式查找号
CLKB0	W22	W22	90	AB12	N16	AC18	V12	
CLKB1	W21	W21	91	AA12	M16	AD17	W12	
DBT0	AA10	AA10	98	AB3	M22	AA18	D12	
DBT1	AA8	AA8	86	AA3	L22	AE17	C12	
DB0	Y22	Y22	133	AB16	B16	Y17	E8	
DB1	Y21	Y21	129	AA16	C16	Y18	D11	
DB2	AA21	AA21	127	AA15	D17	AK16	D8	PE0(51核)
DB3	AA20	AA20	126	AB15	K20	AK18	AH13	
DB4	W20	W20	124	AA14	K21	AK19	AF7	PE2(51核)
DB5	AB20	AB20	121	AB14	K22	AJ19	AH15	
DB6	AA18	AA17	119	AB13	M20	AJ17	AF4	
DB7	AB4	AB17	115	AA13	M21	AJ16	AH3	
DB8	V16	V16	111	AB10	N21	AH18	AD5	
DB9	U16	U16	112	AA10	R22	AH17	AG14	
DB10	AA14	AA14	106	AB8	R21	AG16	AE23	
DB11	AB15	AB15	110	AA8	T22	AE16	AE6	
DB12	Y13	Y13	103	AB5	N20	AF16	AD23	
DB13	AB13	AB13	104	AA5	N19	AG17	AE24	
DB14	AB10	AB10	99	AB4	P19	AA19	AD20	
DB15	AB8	AB8	87	AA4	P17	AC20	AD17	

图 4-38 引脚查找表

根据查表锁定引脚为如图 4-39(核心板是 DE1-SOC，根据实际目标板不同，对应的引脚也不同)

To	Assignment Name	Value	Enabled
out BT[1]	Location	PIN_AE16	Yes
out BT[2]	Location	PIN_AF16	Yes
out BT[3]	Location	PIN_AG17	Yes
out BT[4]	Location	PIN_AA18	Yes
out BT[5]	Location	PIN_AA19	Yes
out BT[6]	Location	PIN_AE17	Yes
out BT[7]	Location	PIN_AC20	Yes
in CLK	Location	PIN_AC18	Yes
out SG[0]	Location	PIN_AH17	Yes
out SG[1]	Location	PIN_AH18	Yes
out SG[2]	Location	PIN_AJ16	Yes
out SG[3]	Location	PIN_AJ17	Yes
out SG[4]	Location	PIN_AJ19	Yes
out SG[5]	Location	PIN_AK18	Yes
out SG[6]	Location	PIN_Y18	Yes
out BT[0]	Location	PIN_AG16	Yes

图 4-39 DE1-SOC (举例)



图 4-40 硬件连接

(5) 硬件连接

准备 1 个杜邦单线和 2 根十芯线，单线一端连接 J17 的 CLKB0 端，作为时钟输入端，另一端连接主系统左下方时钟源，另两根十芯线连接如图 4-40。

(6) 下载测试：

编程下载，进行实验验证。下载后如图 4-40，8 个数码管分别显示显示 1、3、5、7、9、A、b、C,根据上述的语言描述，可以更改显示段。输入的时钟在左下方可以选择高频或低频，可以观察数码管扫描的速度。

## 4.24. 4X4 阵列键盘键信号检测电路设计

实验路径：\ DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO3\_SCAN\_4X4KEY

(1) 实验目的：

了解 4X4 键盘的 8 根线扫描 16 个键的方法。

(2) 实验原理：

4X4 阵列键盘十分常用，图 4-41 是其电路和接口。假设其两个 4 位口，A[3:0]和 B[3:0]都有上拉电阻图 4-42。在应用中，当按下某键后，为了辨别和读取键信息，一种比较常用的方法是，向 A 口扫描输入一组分别只含一个 0 的 4 位数据，如 1110, 1101, 1011 等。若有键按下，则 B 口一定会输出对应的数据，这时，只要结合 A, B 口的数据，就能判断出键的位置。如当键 S0 按下，对于输入的 A=1110 时，那么输出的 B=0111。于是 {B,A}=0111\_1110 就成了 S0 的代码。当检测到某个键时，对应输出键盘码。例 4-24 就是根据此原理给出的描述语言【例 24-1】设计程序。

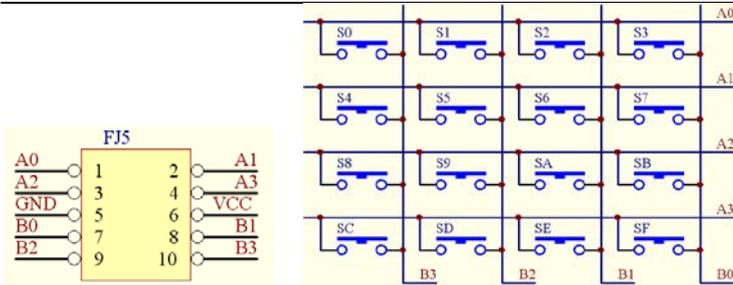


图 4-41 4X4 键盘电路和 10 芯接口

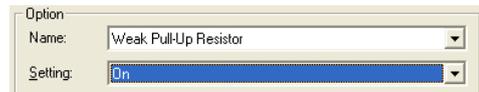


图 4-42 设置端口上拉

## 【例 24-1】

文本： K4X4

## (3) 实验仿真：

根据描述语言进行仿真测试，分析波形

## (4) 硬件准备：

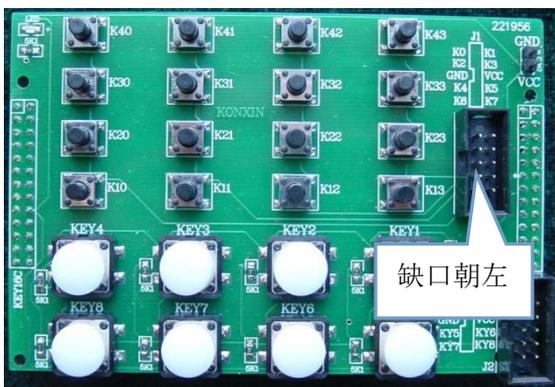


图 4-43 4X4+8 个单脉冲综合键盘模块

图 4-43 此键盘是综合使用键盘，上面的黑色键盘采用 8 个线扫描方式接 16 个键，可参考图 4-41，下面 8 个白色键盘是独立的单脉冲键盘，原理图请参考文件夹“原理图\4X4KEYs\_8KEYs\_SCH.pdf”。

## (5) 引脚锁定：

引脚锁定键盘引脚要通过连接线连接，此模块插在 KX-CDS 的右下方 4 组插座，其中任意一组都可以插扩展模块。IO 口锁定这里我们选择 J6 或其他。按键码输出我们接模式 6 的数码管 8 来显示。时钟源接核心板上的 DE 板 50MHz、KX 板 20Mhz 脚或主系统上时钟源。

A[3..0] (K0\K1\K2\K3, 键盘) 分别接：DB0\DB1\DB3\DB5(主系统 J6)。

B[3..0]K4\K5\K6\K7, 键盘) 分别接：DB6\DB7\DB8\DB9(主系统 J6)。

RR 输出显示脚，选择模式 6 的数码管 8(PIO40-PIO46)。

CLK 根据自己需要来连接核心板主系统。

根据 4.23 实验方法查找引脚锁定。

## (6) 硬件连接

准备 1 个杜邦单线和 1 根十芯线，单线一端连接 J17 的 CLKB0 端（如需要），作为时钟输入端，另 1 根十芯线连接 4X4 键盘对应的十芯口和主系统的 J6 口。

## (7) 下载测试：

编程下载，进行实验验证。选择模式 6，分别按下 4X4 键盘模块的 16 个键，再观察模式部分的数码管 8 上的键盘码是否和语言描述的译码器一样。

## (8) 实验任务

(1) 修改程序例 4-24，使第二个 case 语句成为一个独立的有时钟敏感信号的过程结构，再根据教材[2]6.4.2 节的方法使程序自动生成 ROM 替代模块（要检测键功能是否被改变）；或干脆例化进一个 LPM\_ROM 以取代

第二个 case 语句。检测其功能和资源的变化情况。

(2) 实验任务：利用此项电路进而设计一个 4 位加、减、乘法计算器。

(3) 实验任务：为了更实用，参考教材[2]7.8 节，为键盘电路加上抖动电路模块。

(4) 实验任务：首先回答问题：例 4-24 的程序中为何没有加 default 语句，它希望借此实现什么功能？

如果加上 default 语句会有什么后果？分别用仿真波形说明之。在 default 语句存在的条件下，需要增加什么电路才能实现例 4-24 同样的功能？试给出完整程序，并硬件验证之。

## 4.25 硬件消抖动电路设计

实验路径：\ DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO4\_Di\_TRemble

### (1) 实验目的：

掌握消抖动电路的原理。

### (2) 实验原理：

分别用两个计数器去对输入信号的高电平和低电平的持续时间（脉宽）进行计数（在时间上是同时但独立计数）。只有当高电平的计数时间大于某值，则判为遇到正常信号，输出 1；若低电平的计数时间大于某值，则输出 0。此例的仿真波形如图 4-44 所示。



图 4-44 例 10-11 消抖动电路仿真波形

由波形图可见，其输出信号脉宽比逻辑方式输出的信号要宽得多。此例的输出脉宽与正常信号高电平 KH 的位宽和工作时钟频率共同决定，不单纯由时钟决定。所以优于以上的逻辑方式。例 4-25 给出的设计比前面的电路要容易控制，且效果更好，只是耗用资源比较多。此电路同样能用于消除来自不同情况的干扰、毛刺和电平抖动。其中的工作时钟 CLK 的频率大小要视干扰信号和正常信号的宽度而定。对于类似键抖动产生的干扰信号，频率可以低一些，数万赫即可；若为比较高速的时钟信号，则可利用 FPGA 内的锁相环，使 CLK 能到达 200MHz 以上。此外，KH 和 KL 的计数位宽和计数值都可以根据具体情况调节。例 4-25

### 例【4-25】

文本： ERZP

在此实验，输出一路信号作为一个计数器计数时钟，再做一个双路输出的选择器，用一个开关，当开关为低电平时，这路信号有抖动输出，当这个开关为高电平时，这路信号无抖动输出，看计数结果。

### (3) 硬件准备：

图 4-43 模块中，下面 8 个白色键盘是独立的单脉冲键盘，没有进过消抖动处理，选择其中一个键作为一路输入信号，在用主系统的模式电路 5 中键 1 作为选通信号，数码 1、2 作为计数显示。

### (5) 引脚锁定：

PCLK 作为一路信号键 KY1(键盘)输入分别锁定：DB0(主系统 J6)。

S 作为选通键锁定 模式 5 的键 1 (PIO0)。

B 作为输出计数显示，选择模式 5 的数码管 1、2(PIO16-PIO23)。

CLK 根据自己需要来连接核心板的时钟源或主系统。

根据 4.23 实验方法查找引脚锁定。

### (6) 硬件连接

准备 1 个杜邦单线，连接 4X4 键盘下方十芯口 KY1 针和主系统的 J6 口的 DB0 针。

### (7) 下载测试：

编程下载，进行实验验证。选择模式 5，先置模式 5 上键 1 为低电平，为无消抖动电路，再按下 4X4 键盘模块下方的白色按键的 KY1 键，每按一次，观察模式 5 数码管 1 上计数是否稳定。再把模式 5 上的键 1 置高电

平，为有消抖动电路。对比一下两方面的计数情况。

### (8) 设计任务：

FPGA 中的去抖动电路十分常用，在以后的实验中会多次用到。参考教材[2]8.8 节，设计不同类型的消抖动电路，比较并讨论它们的适用范围和性能特点。提出自己的设计方案，并硬件验证之。较方便的方法是用原理图作顶层设计，调入不同的消抖动模块进行比较测试。4-43 模块的 8 个键都有抖动，可以利用它们进入一个计数器，按键后观察其计数情况，即能很容易了解去抖动效果。

## 4.26 ADC 采样控制电路设计（参照《EDA 技术实用教程》第十章）

实验路径：\DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO5A\_ADC0809

### (1) 实验目的：

学习设计状态机对 A/D 转换器 0809 采样的控制电路。

### (2) 实验原理：

ADC0809 的采样控制原理已在教材[2]中作了详细说明（实验程序用教材[2]）。ADC0809 是 CMOS 的 8 位 A/D 转换器，片内有 8 路模拟开关，可控制 8 个模拟量中的一个进入转换器中。转换时间约  $100\ \mu\text{s}$ ，含锁存控制的 8 路多路开关，输出由三态缓冲器控制，单 5V 电源供电。

利用状态机例【4-26-1】，主要控制信号。START 是转换启动信号，高电平有效；ALE 是 3 位通道选择地址（ADDC、ADDB、ADDA）信号的锁存信号。当模拟量送至某一输入端（如 IN0 或 IN1 等）时，由 3 位地址信号选择，而地址信号由 ALE 锁存；EOC 是转换情况状态信号，当启动转换约  $100\ \mu\text{s}$  后，EOC 产生一个负脉冲，以示转换结束；在 EOC 的上升沿后，若使输出使能信号 OE 为高电平，则控制打开三态缓冲器，把转换好的 8 位数据结果输至数据总线。至此，ADC0809 的一次转换结束。

例【4-26-1】

文本：ADCINT

### (3) 实验任务 1：

利用 Quartus II 对文本编辑输入和仿真测试，并给出仿真波形。最后进行引脚锁定并进行测试，硬件验证教材[1]例 810-2 电路对 ADC0809 的控制功能。为此，图 4-45 给出了此项实验的原理图顶层设计。其中的模块 ADC0809。图中的锁相环输入 20MHz，设置两个时钟输出：c0 输出 5MHz，作为状态机工作时钟；c1 频率是 500kHz，作为 ADC0809 的工作时钟。如 DE 板上是输入 50M，KX 板上输入是 20M。

### (4) 硬件准备：

需要一块扩展板，详细浏览 2.3.11，此板的 J2 口是 AD0809 的控制端口，CLK/ENA/ALE/ADDB/ADDA/ADDC/EDO，J3 口是 D0~D7 口的数据口，这里通过 FPGAIO 口连接这两十芯口进行控制。

### (5) 引脚锁定：

主频时钟 CLK 可以接核心板上，或锁定 J17 的 CLKB0；

控制 AD 采样时钟通过 FPGA IO 口输入，CLKOUT 锁定 DB13；

ADDA/ADDB/ADDC 分别锁定 DB10/DB11/DB12；

OE 锁定 DB14；

EOC 锁定 DBT1；

START 锁定 DBT0

选择模式 5

CLR 锁定键 1 (PIO0)；

转换的数据 D[7..0]（分别接：DB0\DB1\DB3\DB5\DB6\DB7\DB8\DB9(主系统 J6)）。

转换后的数据 Q 锁定数码 1 和 2(PIO16~PIO23)。

根据 2.2/2.4 查表实验方法查找引脚锁定。

### (6) 硬件连接

扩展板插主系统右方 4 组插座，其中任意一组都可以插扩展模块。准备 1 个杜邦单线和 2 根十芯线，单线一端连接 J17 的 CLKB0 端，另一端接主系统的时钟源区 20MHz（如需要），作为时钟输入端，1 根十芯线连接扩展板的十芯口 J2 和主系统的 J7 口，另 1 根十芯线连接扩展板的十芯口 J3 和主系统的 J6 口。

### (7) 下载测试:

编程下载，进行实验验证。选择模式 5，键复位，旋转扩展板上的旋钮，向 AD 输入从低到高的模拟电压，再观察模式部分的数码管 1 上转换成的数据 00~FF。

建议仿真前先除去锁相环（引脚锁定标注是基于 CDS 系统的）当旋转电位器时可以看到数码管显示的采样数据的变换。KX-CDS 实验系统演示示例(含实验指导 pdf/PDF 课件):

### (8) 实验任务 2:

利用教材[2]介绍的内容用不同的编码形式设计此例。例如在不改变原代码功能的条件下将教材表达成用状态码直接输出型的状态机；或用其他编码形式设计此状态机，然后进行硬件验证和讨论。

### (9) 实验任务 3:

利用多种方法设计安全可靠的状态机，并对这些方法作比较，总结安全状态机设计的经验。

### (10) 实验任务 4:

利用以上的设计完成一简易数字电压表设计。测试电压范围是 0 至 5V（可直接利用扩展模块 B6 上的电位器输出电压），精度 1/256；在数码管上用十进制数显示，如 3.1V 等。提示：首先找到 ADC 转换的 16 进制数与满度电压的对应关系，如可查表式计算技术，获得相关的数据，利用 ROM 存储计算表格。当 ADC 转换好的 8 位二进制数作为 ROM 的地址信号输入 ROM 后，ROM 的数据线将能直接输出对应的电压值显示。

实验报告：根据以上实验要求、实验任务和实验思考题写出实验报告。

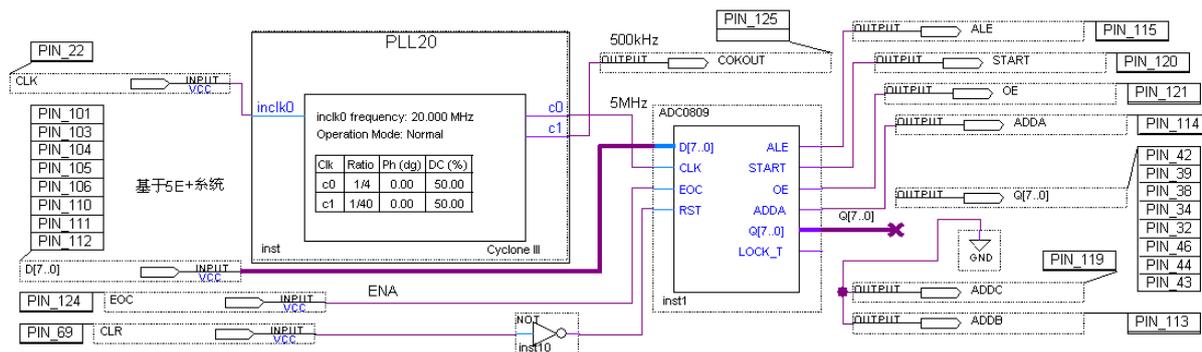


图 4-45 ADC0809 采样控制实验电路与引脚锁定指示

## 4.27 直流电机综合测控系统设计

实验路径：\DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO6\_DC\_MOTO

### (1) 实验目的:

学习直流电机 PWM 的 FPGA 控制。掌握 PWM 控制的工作原理，对直流电机进行闭环速度控制、旋转方向控制、变速控制。

### (2) 实验原理:

一般的脉宽调制 PWM 信号是通过模拟比较器产生的，比较器的一端接给定的参考电压，另一端接周期性线性增加的锯齿波电压。当锯齿波电压小于参考电压时输出低电平，当锯齿波电压大于参考电压时输出高电平。改变参考电压就可以改变 PWM 波形中高电平的宽度。若用单片机产生 PWM 信号波形，需要通过 D/A 转换器产生锯齿波电压和设置参考电压，通过外接模拟比较器输出 PWM 波形，因此外围电路比较复杂。

FPGA 中的数字 PWM 控制与一般的模拟 PWM 控制不同。用 FPGA 产生 PWM 波形，只需 FPGA 内部资源就可以实现。用数字比较器代替模拟比较器，其一端接设定值计数器输出，另一端接线性递增计数器输出。当线性计数器的计数值小于设定值时输出低电平，当计数值大于设定值时输出高电平。与模拟控制相比，省去

了外接的 D/A 转换器和模拟比较器，FPGA 外部连线很少、电路更加简单、便于控制。脉宽调制式细分驱动电路的关键是脉宽调制，转速的波动随着 PWM 脉宽细分数的增大而减小。

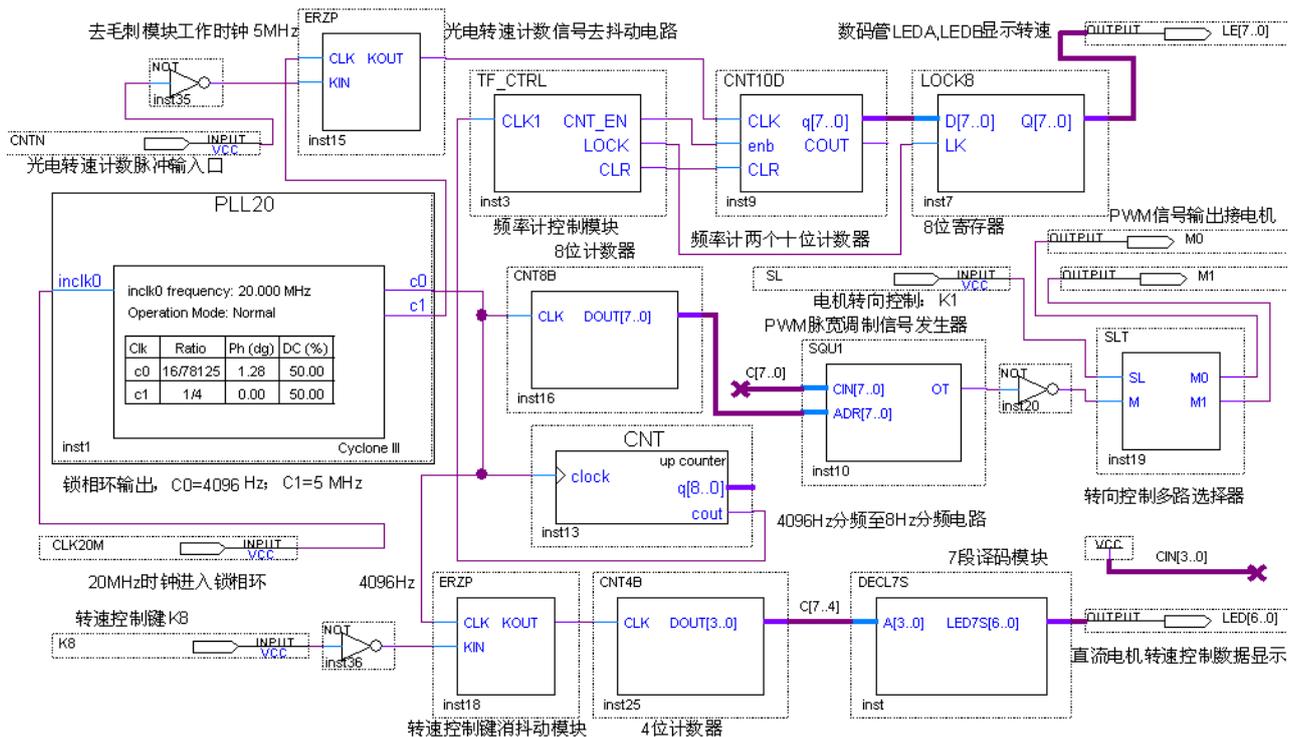


图 4-46 直流电机驱动控制电路顶层设计

图 4-46 是直流电机控制电路顶层设计，主要由以下 3 个大部分组成：

(1) PWM 脉宽调制信号发生模块 SQU1 (例 4-27)。此模块是 FPGA 中的 PWM 脉宽调制信号产生电路。它的输出接电机转向控制电路模块，此模块输出的两个端口接电机。通过控制 SL 端 (键 K1)，可以改变电机转向。SQU1 的输入端之一来自模块 CNT8B。这是一个 8 位计数器，输出的数据相当于锯齿波信号，此信号的频率就是输出 PWM 波的频率，它由来自锁相环的 C0 的频率决定，频率选择 4096Hz。SQU1 模块的另一端来自键控的 8 位数据，其中低 4 位 CIN[3..0] 设定为恒定 1111，高 4 位由计数器 CNT4B 产生，计数器的时钟来自键。于是可以通过手动按键控制电机的转速。

如果键有抖动，在键输入进计数器前加了一个消抖动模块 ERZP，由于键抖动的脉冲频率比较低，故 ERZP 的工作时钟选择 4096Hz。

#### 【例 4-27】

文本：SQU1

(2) 电机转速测试系统。电机转速的测定很重要，一方面可以直观了解电机的转动情况，更重要的是，可以据此构成电机的闭环控制，即可以设定电机的某一转速后，确保负载变动时仍旧能保持恒定转速和恒定输出功率。本项实验是通过红外光电测定转速的。每转一圈光电管发出一个负脉冲，由图 4-46 的左上方的 CNIN 口进入。由于此类方法测转速，会附带大量毛刺脉冲，所以在 CNIN 的信号后必须接入消毛刺模块 ERZP，其工作时钟频率是 5MHz。ERZP 的输出信号进入一个 2 位十进制显示的频率计。图 4-22 中频率测量功能模块 TF\_CTRL 是测频时序控制电路，其功能可参考教材[2]；模块 CNT10D 是双十位计数器，模块 LOCK8 是 8 位寄存器，由 74374 担任。两位十进制频率数显示在系统实验电路模式上数码管 LEDA, LEDB。

(3) 工作时钟发生器。这主要由锁相环 PLL20 模块担任，或直接从主系统直接提供。其输入频率是 20MHz；输出两个频率：C0=4096Hz，C1=5MHz。

#### (4) 硬件准备：

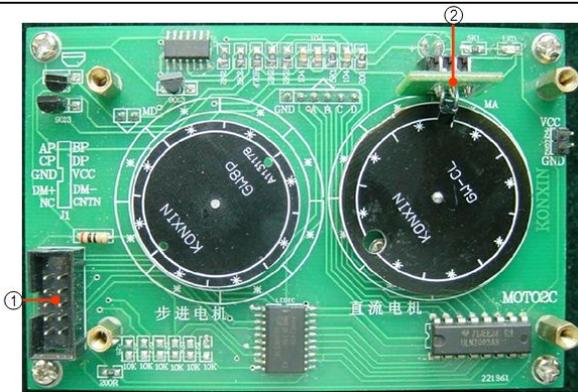


图 4-47 直流步进电机模块

图 4-47 电机模块包括步进电机和直流电机扩展模块。原理图请参考文件夹“原理图\MOTO\_SCH.pdf”。

标注“1”的标识 DM+和 DM- 是直流电机接口，步进电机 AP,BP,CP,DP 控制端口，CNTN 是直流电机计数端。标注“2”是直流电机转动圈数红外接收模块。

#### (5) 引脚锁定:

电机模块插主系统右方 4 组插座，其中任意一组都可以插扩展模块。直流电机 DM+、DM-控制端口接 FPGA 的 IO 口锁定这里我们选择 JP6 或其他。显示的转速和加速的级别分别选择模式 5 的数码管 7、8 和数码 1 来显示，键 1 控制正反转，键 2 选择加速级别输入。时钟源锁定主系统上时钟源。

DM0、DM1 电机正反转分别锁定：DB6\DB7(主系统 J6)。

CNTN 转速红外接收锁定：DB9(主系统 J6)。

SPD[7..0]转速输出显示，选择模式 5 的数码管 7、8(PIO40~PIO47)。

DAT[3..0]选择转速级别数字显示选择选择模式 5 数码管 1 (PIO16~PIO19)

SL 控制正反转锁定模式 5 键 1 (PIO0)。

K8 控制输入速度级别锁定模式 5 键 2 (PIO1)。

CLK1 锁定主系统 J17 的 CLKB0。

CLK2 锁定主系统 J17 的 CLKB1。

根据 4.23 实验方法查找引脚锁定。

#### (6) 硬件连接

准备 2 个杜邦单线和 1 根十芯线，第 1 根单线一端连接 J17 的 CLKB0 端，另一端接主系统时钟 4096Hz 作为电机转速时钟输入端，第 2 根单线一端连接 J17 的 CLKB1 端，另一端接主系统时钟 5MHz，作为电机测速时钟输入端，另 1 根十芯线连接 J1 对应的十芯口和主系统的 J6 口。

#### (7) 下载测试:

编程下载，进行实验验证。选择模式 5，分别测试键 1 正反转、键 2 的增减速度级别，再观察模式部分的数码管 7、8 测速情况。

#### (8) 实验任务:

1: 完成直流电机控制电路所有模块进行定制、设计，并分别进行仿真，给出电机的驱动仿真波形，与示波器中观察到的电机控制波形进行比较。

讨论其工作特性。最后完成整个系统的验证性实验。

2: 增加逻辑控制模块，用测到的转速数据控制输出的 PWM 信号，实现直流电机的闭环控制，要求旋转速度可设置。转速范围为 10~40 转/秒。

3: 了解工业专用直流电机转速控制方式，利用以上原理测速和控制电机，实现闭环控制。要求在允许的转速范围转矩功率不变。

4、完成以图 4-45 所示的直流电机控制电路所有模块进行定制、设计，并分别进行仿真，给出电机的驱动仿真波形，并与示波器中观察到的电机控制波形进行比较。讨论其工作特性。最后完成整个系统的验证性实验。

5、增加逻辑控制模块，用测到的转速数据控制输出的 PWM 信号，实现直流电机的闭环控制，要求旋转速度可设置。转速范围每秒 10 至 40 转。

6、了解工业专用直流电机转速控制方式，利用以上原理测速和控制电机，实现闭环控制。要求在允许的转速范围转矩功率不变。

## 4.28 步进电机控制电路设计

实验路径: \ DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO7\_STEP\_MOTO

### (1) 实验目的:

了解四项步进电机的工作原理。

### (2) 实验原理

步进电机是一种离散运动的装置, 步进电机驱动器通过外加控制脉冲, 并按环形分配器决定的分配方式, 控制步进电机各相绕组的导通或截止, 从而使电机产生步进运动<sup>[9]</sup>。步进电机的驱动电路根据控制信号工作, 控制信号由各类控制器来产生。其基本原理作用如下:

步进电机正、反转控制:

步进电机的正、反转控制可通过改变步进电机各绕组的通电顺序来改变其转向, 四相双四拍步进电机通电顺序为 AB—BC—CD—DA—AB……时电机正转; 当绕组按 AD—DC—CB—BA—AD……顺序通电时电机反转。因此, 可以通过 PLC 输出的方向控制信号改变硬件环形分配器的输出顺序, 或经编程改变输出脉冲的顺序来改变步进电机绕组的通电顺序实现。

步进电机运动速度的控制:

步进电机的转速取决于输入的脉冲频率。如果给步进电机发一个控制脉冲, 它就转一步, 再发一个脉冲, 它会再转一步。两个脉冲的间隔越短, 步进电机就转得越快。调整控制器发出的脉冲频率, 就可以对步进电机进行调速, 从图 4-48 可以看出, 当改变输入脉冲的周期时, A、B、C、D 四相绕组高低电平的宽度将发生变化。这就导致通电和断电变化的速率发生变化, 使电机转速发生变化。所以调节输入脉冲的周期就可以控制步进电机的运动速度。

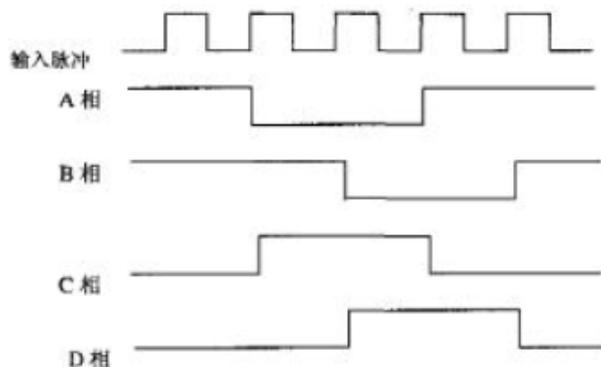


图 4-48 脉冲分配波形图

根据步进电机的工作原理, 我们设计一个依次向四项发送脉冲信号, 来驱动电机转动。例【4-28】

例【4-28】

文本: PW

### (3) 实验仿真

对此实验进行仿真, 并分析波形情况。

### (4) 硬件准备:

在图 4-47 电机模块中左边的那个电机为步进电机。原理图请参考文件夹“原理图\MOTO\_SCH.pdf”。

标注“1”的标识中步进电机 AP,BP,CP,DP 控制端口。

### (5) 引脚锁定:

电机模块插主系统右方 4 组插座，其中任意一组都可以插扩展模块。步进电机 AP\BP\CP\DP 四个控制端口接 FPGA 的 IO 口锁定这里我们选择 JP6 或其他。键 1 控制正反转。时钟源锁定主系统上时钟源。

Q[3..0]驱动电机的四个信号分别锁定：DB0\DB1\DB3\DB5(主系统 J6)。

SS 控制电机的正反转锁定模式 5 键 1 (PIO0)。

CLK 锁定主系统 J17 的 CLKB0。

根据 4.23 实验方法查找引脚锁定。

#### (6) 硬件连接

准备 1 个杜邦单线，一端连接 J17 的 CLKB0 端，另一端接主系统时钟 4096Hz 作为电机转速时钟输入端。另 1 根十芯线连接 J1 对应的十芯口和主系统的 J6 口。

#### (7) 下载测试：

编程下载，进行实验验证。选择模式 5，测试键 1 正反转情况。

## 4.29 步进电机细分控制电路设计

实验路径：\DEMO\EDA\_DEMO\MODU\_DEMO\DEMO8\_FDIV\_STEP\_MOTO

#### (1) 实验目的：

学习用 FPGA 控制步进电机，包括能对步进电机进行细分控制的模块设计。

#### (2) 实验原理：

步进电机的驱动是靠给步进电机的各相励磁绕组轮流通以电流，实现步进电机内部磁场合成方向的变化来使步进电机转动的。设矢量  $T_A$ 、 $T_B$ 、 $T_C$ 、 $T_D$  为步进电机 A、B、C、D 四相励磁绕组分别通电时产生的磁场矢量； $T_{AB}$ 、 $T_{BC}$ 、 $T_{CD}$ 、 $T_{DA}$  为步进电机中 AB、BC、CD、DA 两相同时通电产生的合成磁场矢量。当给步进电机的 A、B、C、D 四相轮流通电时，步进电机的内部磁场从  $T_A \rightarrow T_B \rightarrow T_C \rightarrow T_D$ ，即磁场产生了旋转。一般地，当步

$$\theta_B = \frac{\theta_M}{N_r}$$

进电机的内部磁场变化一周 ( $360^\circ$ ) 时，电机的转子转过一个齿距，因此，步进电机的步距角  $\theta_B$  可表示为：

式中， $N_r$  为步进电机的转子齿数； $\theta_M$  为步进电机运行时两相邻稳定磁场之间的夹角。 $\theta_M$  与电机的相数

( $M$ ) 和电机的运行拍数有关。当电机以单四拍方式运行时， $\theta_M = 90^\circ$ ；当电机以四项八拍方式运行时， $\theta_M = 45^\circ$ 。和单四拍方式相比， $\theta_M$  和  $\theta_B$  都减小了一倍，实现了步距角的二细分。但是在通常的步进电机驱动线路中，由于通过各相绕组的电流是个开关量，即绕组中的电流只有零和某一额定值两种状态，相应的各绕组产生的磁场也是一个开关量，只能通过各项的通电组合来减小  $\theta_M$  和  $\theta_B$ 。因此，这样可达到的细分数很有限。以四相反应式步进电机为例，最多只能实现二细分，对于相数较多的步进电机可达到的细分数稍大一些，但也很有限。因此，要使可达到的细分数较大，就必须能控制步进电机各相励磁绕组中的电流，使其按阶梯上升或下降，即在零到最大相电流之间能有多个稳定的中间电流状态，相应的磁场矢量幅值也就存在多个中间状态，这样，相邻两相或多相的合成磁场的方向也将有多个稳定的中间状态。四相步进电机八细分时的各相电流是以 1/4 的步距上升或下降的，在两相  $T_A$ 、 $T_B$  中间又插入了 7 个稳定的中间状态，原来一步所转过的角度  $\theta_M$  将由 8 步完成，实现了步距角的八细分。由此可见，步进电机细分驱动的关键在于细分步进电机各励磁绕组中的电流。本课题即采用了四相步进电机八细分方案。

为了对步进电机的相电流进行控制，从而达到细分步进电机步距角的目的，人们曾设计了多种步进电机细分驱动电路。最初对电机相电流的控制是由硬件来实现的，每一相绕组的相电流用  $n$  个晶体管构成  $n$  个并联回路来控制，靠晶体管导通数的组合来控制相电流。这种细分驱动电路线路复杂，体积大，成本高，而且电路一旦制造出来就难以改变其细分数，缺乏柔性，因此在目前的实用中已经很少采用这种方法。

实践表明，如果使用 FPGA 进行数字控制，将为步进电机的细分驱动带来很大的便利。目前，最常用的开关型步进电机细分驱动电路有斩波式和脉宽调制 (PWM) 式两种。

脉宽调制式细分驱动电路的关键是脉宽调制，它的作用是将给定的电压信号调制成接近连续的信号，角速度的波动也随着细分数的增大而减小。一般角速度波动与步距角成正比，与细分数成反比。

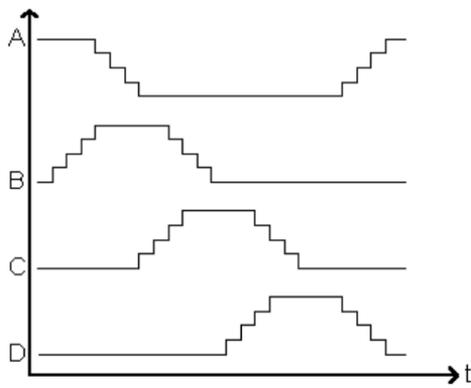
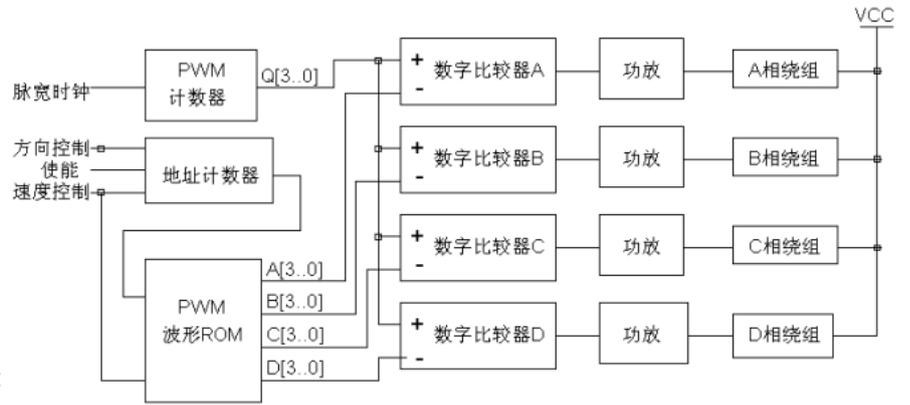


图 4-49 四相步进电机 8 细分电流波形图



4-50 步进电机细分驱动电路结构图

步距细分的系统构成

图 4-49 所示为四相步进电机的八细分电流波形，从图中可以看出，一般情况下总有二相绕组同时通电。一相电流逐渐增大，另一相逐渐减小。对应于一个步距角，电流可以变化 N 个台阶，也就是电机位置可以细分为 N 个小角度，这就是电机的一个步距角被 N 细分的工作原理。或者说，步距角的细分就是电机绕组电流的细分，从而可驱动步进电机平滑运行。

图 4-50 所示为步进电机细分驱动系统结构图，直观地反映了步进电机细分驱动的原理。

该系统是由 PWM 计数器、波形 ROM 地址计数器、PWM 波形 ROM 存储器、比较器、功放电路等组成。其中，PWM 计数器在脉宽时钟作用下递增计数，产生阶梯形上升的周期性锯齿波，同时加载到各数字比较器的一端；PWM 波形 ROM 输出的数据 A[3..0]、B[3..0]、C[3..0]、D[3..0] 分别加载到各数字比较器的另一端。当 PWM 计数器的计数值小于波形 ROM 输出数值时，比较器输出低电平；当 PWM 计数器的计数值大于波形 ROM 输出数值时，比较器输出高电平。由此可输出周期性的 PWM 波形。根据图 4-49 所示步进电机八细分电流波形的要求，将各个时刻细分电流波形所对应的数值存放于波形 ROM 中，波形 ROM 的地址由地址计数器产生。通过对地址计数器进行控制，可以改变步进电机的旋转方向、转动速度、工作\|停止状态。FPGA 产生的 PWM 信号控制各功率管驱动电路的导通和关断，其中 PWM 信号随 ROM 数据而变化，改变输出信号的占空比，达到限流及细分控制，最终使电机绕组呈现阶梯形变化，从而实现布局细分的目的。输出细分电流信号采用 FPGA 中 LPM\_ROM 查表法，它是通过在不同地址单元内写入不同的 PWM 数据，用地址选择来实现不同通电方式下的可变步距细分。根据此原理设计出步进电机细分驱动控制器的电路原理图如图 4-50 所示。

图 4-51 中，其中输入 S 选择是否输出细分控制信号。模块 STEP1 产生非细分控制信号；CNT10B 产生分频时钟；模块 CNT8 产生锯齿波比较信号（也可用三角波）；ROM3 并行存放 4 路准正弦波数据（可以用 4 个 ROM 产生更精细的波形）；CMP3 是比较器。

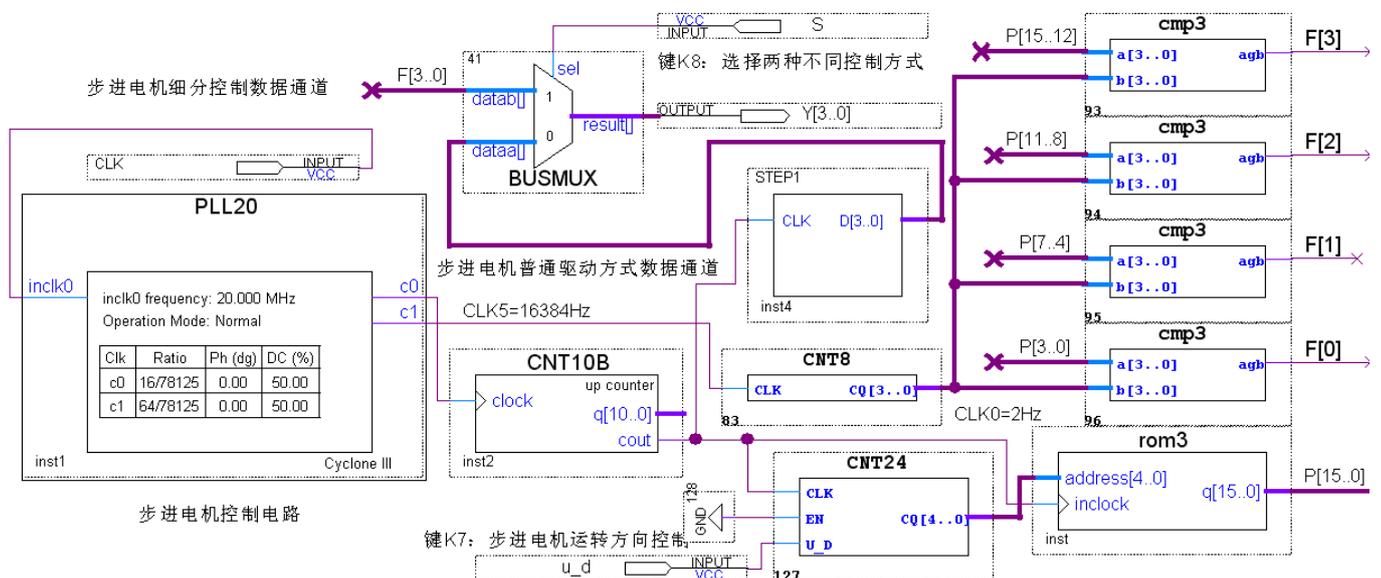


图 4-51 步进电机细分控制电路

文本：STEP1

#### (4) 实验仿真

对此实验进行仿真，并分析波形情况。

#### (4) 硬件准备：

在图 4-47 电机模块中左边的那个电机为步进电机。原理图请参考文件夹“原理图\MOTO\_SCH.pdf”。标注“1”的标识中步进电机 AP,BP,CP,DP 控制端口。

#### (5) 引脚锁定：

电机模块插主系统右方 4 组插座，其中任意一组都可以插扩展模块。步进电机 AP\BP\CP\DP 四个控制端口接 FPGA 的 IO 口锁定这里我们选择 JP9 或其他。键 1 控制正反转。时钟源锁定主系统上时钟源。

Y[3..0]驱动电机的四个信号分别锁定：DA11\DA10\DA8\D9(主系统 J9)。

u\_d 控制电机的正反转锁定模式 5 的键 2 (PIO1)。

S 控制电机细分锁定模式 5 键 1 (PIO0)。

CLK 锁定主系统 J17 的 CLKB0。

CLK1 锁定主系统 J17 的 CLKB1。

根据 4.23 实验方法查找引脚锁定。

#### (6) 硬件连接

准备 2 个杜邦单线和 1 根十芯线，第 1 根单线一端连接 JP17 的 CLKB0 端，另一端接主系统时钟 16384Hz 作为电机细分时钟输入端，第 2 根单线一端连接 JP17 的 CLKB1 端，另一端接主系统时钟 4096Hz 作为分频时钟输入给 ROM，另 1 根十芯线连接电机模块对应的十芯口和主系统的 J9 口。

#### (7) 下载测试：

编程下载，进行实验验证。选择模式 5，测试键 1 高电平时细分，电平时非细分。当键 1 置高电平时，键 2 置高电平时，电机细分逆时针转，反之，电机顺时针细分转。电机非细分情况下，键 2 对电机无控制正反转功能。如设置此功能，把 STEP1 模块换成 4-27 实验的程序 PW，再 u\_d 脚连接可正反转的 S 端。

#### (8) 实验任务

(1) 根据例 4-28 设计步进电机基本控制模块，要求能控制转向。

(2) 查阅相关资料，根据演示示例及图 4-50，设计 4 相步进电机细分控制模块，对各模块进行仿真测试。分别实现 1/4、1/8、1/16 步距细分控制。

(3) 为了实现细分转动的平稳性，对图 4-51 的设计做一些改善，如用 4 个 ROM 存储不同相位正弦波形数据；提高精度位数；将 CNT8 改成三角波等。讨论细分控制的实用方案。

## 4.30 点阵 LED 实验控制

实验路径：\DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO14\_seg70\DEMO14\_seg7\_1(16\*16 点阵显示)

#### (1) 实验目的：

学习 LED 点阵的控制扫描显示方法。

#### (2) 实验原理

8X8 点阵的实际外观图如图 4-52：一共 64 个发光点构成，上下两排引脚，每排 8 个，一共 16 个点阵侧面有文字的那面对准自己。

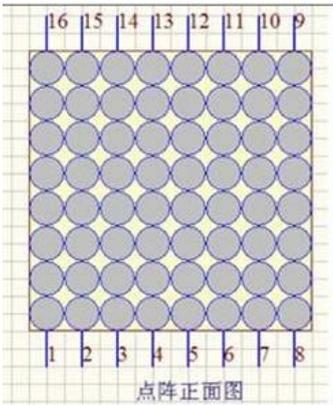


图 4-52 排列

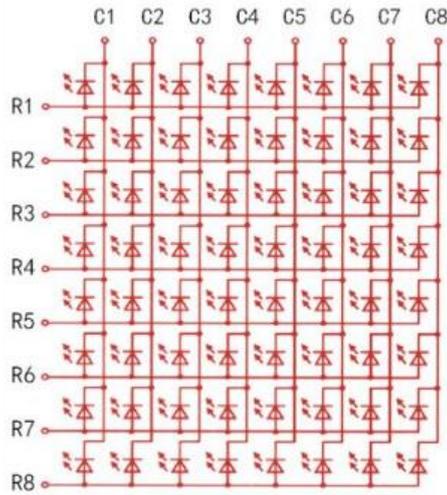


图 4-53 行列

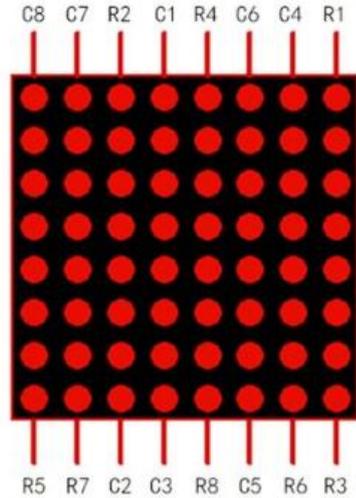


图 4-54 8\*8 点阵实际引脚端

8X8 点阵的显示原理:

图 4-53，8X8 点阵共由 64 个发光二极管组成，R 为列，C 为行，且每个发光二极管是放置在行线和列线的交叉点上，当对应的某一行置 1 电平，某一列置 0 电平，则相应的二极管就亮；如要将第一个点点亮，则 R1 脚接高电平 C1 脚接低电平，则第一个点就亮了；如要将第一行点亮，则第 R1 脚要接高电平，而 C1~C8 这些引脚接低电平，那么第一行就会点亮；如要将第一列点亮。则第 R1 脚接低电平，而 C1~C8 接高电平，那么第一列就会点亮。图 4-54，实际点阵图，每个引脚对应的引脚名。

根据以上点阵原理，语言描述例 4-29

例【4-30】:

文本: seg70

以上描述，康芯点阵模块，共两个 8\*8，也就是 8 列，16 行，我们设计了一个“正”字和一个心形。

(3) 硬件准备:

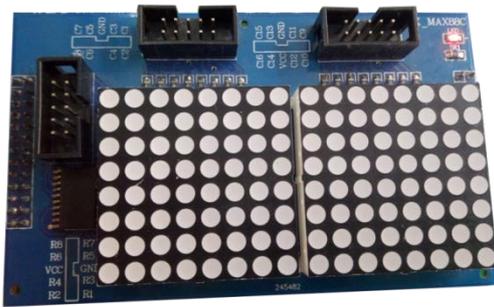


图 4-55 点阵模块 2 组 8\*8 点阵组成

图 4-54 2 组 8\*8 点阵，构成 8\*16 点阵模块，三个十芯接口，（上→下）R1-R8 控制列，（左→右）C1-C16 控制行。LED8X16 点阵模块是由两片 8X8 点阵串联一起组成的。板上分别标出引脚名，锁定时一一对应。原理图参考：“原理图\KX\_MAX88.pdf”。

(5) 引脚锁定:

此模块插主系统右方 4 组插座，其中任意一组都可以插扩展模块。

dataout[7..0]作为列 R 扫描分别锁定: DA14\DAT0\DA13\DA12\DA11\DA10\DA8\DA9(主系统 J9)。

En[15..0]作为行 C 扫描分别锁定: DB15\DAT1\DB14\DBT0\DB13\DB12\DB11\DB10(主系统 J6)、DB9\DB8\DB7\DB6\DB5\DB3\DB1\DB0(主系统 J7)。

rst 复位键选择锁定模式 5 的键 1 (PIO0)。

clk 锁定核心板时钟 DE50MHz, KX20MHz。

(6) 硬件连接

准备 3 根十芯线，第 1 根十芯线连接点阵模块左侧对应的十芯口和主系统的 J9 口、第 2 根十芯线连接点

阵模块上左侧对应的十芯口和主系统的 J6 口、第 3 根十芯线连接点阵模块上右侧对应的十芯口和主系统的 J7 口。

#### (7) 下载测试:

编程下载，进行实验验证。选择模式 5，键 1 复位显示如下：左边“正”字，右边心形如图 4-56。

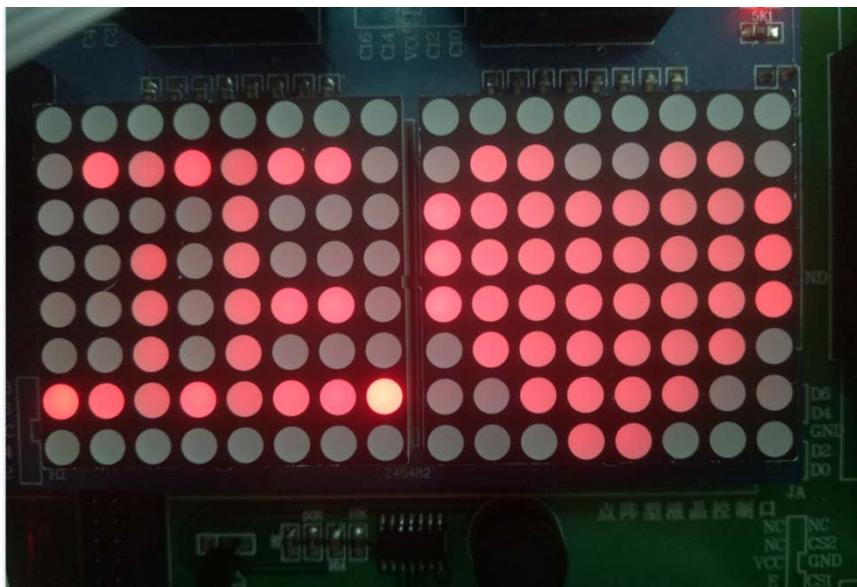


图 4-56 下载后显示的图形

## 4.31 数字彩色液晶显示控制电路设计

实验路径: \DEMO\EDA\_DEMO\MODU\_DEMO\DEMO9\_COLOR\_LCD

### (1) 实验目的:

学会对 TFT 液晶的时序及色调控制方法。

### (2) 实验原理

设计点阵彩色液晶显示控制电路。提供液晶为 7 寸 800\*480TFT 彩屏液晶，具体手册查阅 \VIVI\_FILE\LCD\_FILE 中的“AT070TN83V.1”等文件。

用状态机设计数字 TFT 800×480 彩色点阵型液晶显示控制电路，注意此类液晶控制原理与 VGA 相同，TFT 800×480 彩色液晶接口板上的跳线 U/D 是控制上下扫描方式的；跳线 L/R 是控制左右扫描方式的；跳线 MODE 是选择 DE 或 HV 时序控制方式。方式是普通 LCD 时序控制方式，而 HV 方式是类似 VGA 的控制方式，其行场控制信号分别是 HS 和 VS；若选择 DE 方式，则另两跳线分别选择 DCLK 和 DE；若选择 HV 方式，则另两跳线分别选择 HS 和 VS。根据以上原理，lcd\_demo.vhd 文件给出了描述语言例 4-31。

例【4-31】

文本: lcd\_demo

### (3) 硬件准备:



图 4-57 7 寸彩屏

图 4-57 7 寸彩屏，右侧有三个接口，上面第一十四芯 JP1 和第二个十芯口 JP2 分别是液晶控制端口及时钟输入口，下端十芯口 JP3 为触摸控制端口。设置方法参考本讲义的 2.2.6 原理图请参考文件夹“原理图\LCD\_CL1.pdf

#### (5) 引脚锁定:

液晶模块插主系统右方 J27 和 J28 座，FPGA 连接方式分别把液晶的 JP1 和 JP2 连接主系统的 J1 和 J3 口。

B[5..0]锁定 DA23\DA26\DA27\DA28\DA29\DA24。

G[5..0]锁定 DA10\DA13\DA12\DAT0\DA14\DA22。

R[5..0]锁定 DA4\DA5\DA6\DA9\DA8\DA11。

DE 锁定 DA25。

DCLK(液晶时钟) 锁定 DA2。

CLKIN 锁定核心板的时钟。

CCLK 选择锁定模式 1 的键 1 (PIO0)。

以上锁定的引脚都要根据表 2.2 (模式) 和 2.4 (模块) 查表对应型号的引脚号。

#### (6) 硬件连接:

准备 1 根十芯线，1 根十四芯线，十四芯线连接液晶模块上侧对应的十四芯口 JP1 和主系统的 J1 十四芯口，1 根十芯线连接液晶模块上第二个对应的十芯口 JP2 和主系统的 J3 口。

#### (7) 下载测试:

编程下载，进行实验验证。选择模式 3，键 1 选择不同的显示彩条。

#### (8) 实验任务:

- 1、分别用 DE 和 HV 两种时序控制方式显示彩条和彩色方块图像。
- 2、将逻辑分析仪采样的 8 路数字波形信号用彩色 LCD 显示。
- 3、将简易存储示波器采样的模拟波形信号用彩色 LCD 显示。
- 4、将 DDS 函数信号发生器的波形用彩色 LCD 显示。主要波形有正弦波、方波、三角波、锯齿波、移相双路正弦波、里萨如图、AM、FM、FSK 和扫频等。

## 4.32 模块化交通灯电路设计

实验路径: \DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO1\_jiaotong\DEMO1\_1\_top\_traffic\_1(倒计时)

#### (1) 实验目的、原理

参照实验 4.14，硬件实现在原来模式上的 LED 模拟交通灯实现改为交通灯模块实现，更直观。



图 4-58 交通灯模块

## (2) 硬件准备:

图 4-58 交通灯模块从右向左依次提供四组红、黄、绿、蓝交通模式，下方两十芯口分别是对应的灯的引脚端口。引脚名都已标出，一一对应即可，原理图请参考文件夹“原理图\TRAFIC.pdf”。

## (4) 引脚锁定:

模块插主系统右方四组插座，任意一组都可以，FPGA 连接方式分别将模块 2 个十芯口连接主系统的 J6 和 J7 口。

EG 锁定 DB12; ER 锁定 DB10; EY 锁定 DB11; NG 锁定 DB8; NR 锁定 DB6; NY 锁定 DB7; SG 锁定 DBT1; SR 锁定 DBT0; SY 锁定 DB14; WG 锁定 DB3; WR 锁定 DB0; WY 锁定 DB1; clk 锁定 CLKB0。

以上锁定的引脚都要根据表 2.4 (模块) 查表对应型号的引脚号。

## (6) 硬件连接

准备 2 根十芯线和 1 根单线，1 根连接模块左侧对应的十芯口 JP1 和主系统的 J6 十芯口, 1 根十芯线连接模块右侧十芯口 JP2 和主系统的 J7 口, 1 根单线一端连接主系统的 CLKB0 端, 另一端连接主系统的时钟源区 1Hz。

## (7) 下载测试:

编程下载，进行实验验证。观察显示情况。

## 4.33. 正弦信号发生器设计 (参照《EDA 技术实用教程》第六章)

实验路径: \DEMO\EDA\_DEMO\MODU\_DEMO\DEMO17\_SIN\_WAVE\_DAC0832

### (1) 实验目的:

进一步熟悉 Quartus II 及其 LPM\_ROM 与 FPGA 硬件资源的使用方法。

### (2) 实验原理:

利用定制完成的 LPM\_ROM 设计一个简易的正弦信号发生器。如图 4-59 所示的简易正弦信号发生器的结构由如下 4 个部分组成:

1、计数器或地址信号发生器，这里根据以上 ROM 的参数，选择 7 位输出。

2、正弦信号数据存储 ROM (7 位地址线，8 位数据线)，含有 128 个 8 位波形数据 (一个正弦波形周期) 图 4-60 的 LPM\_ROM: ROM78。

3、工程 SIN\_GNT 顶层原理图设计 8 位 D/A (设此示例之实验器件选择 DAC0832)。



图 4-59 正弦信号发生器结构框图

图 4-60 所示的信号发生器结构图中，顶层文件是原理图工程 SIN，它包含两个部分：ROM 的地址信号发生器，由 7 位计数器担任；正弦数据 ROM，由 LPM\_ROM 模块构成。地址发生器的时钟 CLK 的输入频率  $f_0$  与每周期的波形数据点数 (在此选择 128 点)，以及 D/A 输出的频率  $f$  的关系是： $f = f_0/128$

Addr	+0	+1	+2	+3	+4	+5	+6	+7
00	80	86	8C	92	98	9E	A5	AA
08	B0	B6	BC	C1	C6	CB	DO	D6
10	DA	DE	E2	E6	EA	ED	FO	F3
18	F5	F8	FA	FB	FD	FE	FE	FF
20	FF	FF	FE	FE	FD	FB	FA	F8
28	F5	F3	F0	ED	EA	E6	E2	DE
30	DA	D5	D0	CB	C6	C1	BC	B6
38	B0	AA	A5	9E	98	92	8C	86
40	7F	79	73	6D	67	61	5A	55
48	4F	49	43	3E	39	34	2F	2A
50	25	21	1D	19	15	12	0F	0C
58	0A	07	05	04	02	01	01	00
60	00	00	01	01	02	04	05	07
68	0A	0C	0F	12	15	19	1D	21
70	25	2A	2F	34	39	3E	43	49
78	4F	55	5A	61	67	6D	73	79

图 4-60 sin.mif 文件

图 4-61 是此正弦信号发生器的顶层设计原理图，其中包含作为 ROM 的地址信号发生器的 7 位计数器模块和 LMP\_ROM 的 ROM78 模块。此后的设计流程包括编辑顶层设计文件、创建工程、全程编译、观察 RTL 电路图、仿真、了解时序分析结果、引脚锁定、再次编译并下载、对 FPGA 的存储单元在系统读写测试和嵌入式逻辑分析仪测试等。

(2) 实验内容 1: 实验原理参考教材[2]相关内容。在 Quartus II 上完成简易正弦信号发生器设计，包括建立工程，生成正弦信号波形数据，仿真等。最后在实验系统上实测，包括 SignalTap II 测试、FPGA 中 ROM 的在系统数据读写测试和利用示波器测试。最后完成 EPCSx 配置器件的编程。信号输出的 D/A 使用 DAC0832，注意其转换速率是  $1\mu s$ 。

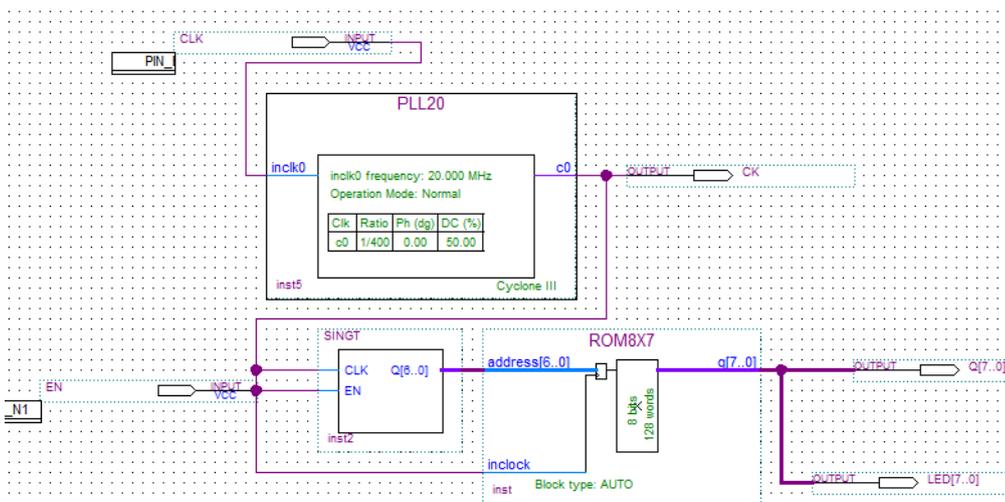


图 6-61 顶层原理图

(4) 硬件准备:

需要一块扩展板，详细浏览 2.3.11，此板 J5/J6 口分别是双 DA0832 是 A 和 B 的 D0~D7 口的数据口，这里通过 FPGAIO 口连接这两十芯口进行控制。

(5) 引脚锁定:

- 主频时钟 CLK 可以接核心板上，或锁定主系统 J17 的 CLKB0;
- 转换的数据 D[7..0] (分别接: DA25\DA24\DA29\DA28\DA27\DA26\DA23\DA22(主系统 J3))
- 选择模式 5
- EN 使能锁定键 1 (PIO0);
- 转换后的数据 LED[7..0]锁定数码 1 和 2(PIO16~PIO23)。
- 根据 2.2/2.4 查表实验方法查找引脚锁定。

(6) 硬件连接

扩展板插主系统时需注意，因 DA 涉及到运放需 $\pm 12V$ ，在扩展的 5 组座中，其中“实验模块 3 (J29/J30)”上输出 $\pm 12V$  供给此板，直接插上此板在此座中，自动供给 $\pm 12V$ ，不需外接 $\pm 12V$ ，或另一种方法：插任意座，可利用主系统上“电源指示区”上的输出的 $\pm 12V$  的连接到此扩展板右上角的 $\pm 12V$  输入端。另准备 1 根十芯线，连接扩展板的十芯口 J5 或 J6DA0832 其中一个数据口和主系统的 J3 口连接。用示波器连接 DA0832A 或 B 对应的 OUT/GND 口。

(7) 下载测试:

编程下载，进行实验验证。选择模式 5，键 1 高电平使能，观察示波器的正弦波形，再观察模式部分的数码管 1 上转换成的数据 00~FF。

**(8) 实验内容 2:**

按照教材[2],用原理图方法设计正弦信号发生器,硬件实现时可以通过 SignalTap II 观察波形。

**(9) 实验内容 3:**

设计一任意波形信号发生器(可使用附录 1 的软件),可以使用 LPM 双口 RAM 担任波形数据存储器,利用单片机产生所需要的波形数据,然后输向 FPGA 中的 RAM。

**4.34 DDS 正弦信号发生器设计**

**实验路径:** \DEMO\EDA\_DEMO\MODU\_DEMO\DEMO10\_DDS\_SIN\_WAVE

**(1) 实验目的:**

学习利用 EDA 技术和 FPGA 实现直接数字频率综合器 DDS 的设计。

**(2) 实验原理:**

实验原理参考教材[2]第六版。详细叙述 DDS 的工作原理,根据教材[2]完成整体设计和仿真测试,深入了解其功能,并由仿真结果进一步说明 DDS 的原理。完成编译和下载,用示波器观察输出波形。

**(4) 硬件准备:**

需要一块扩展板,详细浏览 2.3.11,此板的 J5/J6 口分别是双 DA0832 是 A 和 B 的 D0~D7 口的数据口,这里通过 FPGAIO 口连接这两十芯口进行控制。

**(5) 引脚锁定:**

主频时钟 CLK 可以接核心板上,或锁定主系统 J17 的 CLKB0;

正弦波数据 PA[7..0] (分别接: DA25\DA24\DA29\DA28\DA27\DA26\DA23\DA22(主系统 J3))

选择模式 5

CIN 频率字输入锁定键 1 (PIO0);

LED 频率字显示[3..0]锁定数码 8(PIO44~PIO47)。

根据 2.2/2.4 查表实验方法查找引脚锁定。

**(6) 硬件连接**

扩展板插主系统时需注意,因 DA 涉及到运放需 $\pm 12V$ ,在扩展的 5 组座中,其中“实验模块 3(J29/J30)”上输出 $\pm 12V$ 供给此板,直接插上此板在此座中,自动供给 $\pm 12V$ ,不需外接 $\pm 12V$ ,或另一种方法:插任意座,可利用主系统上“电源指示区”上的输出的 $\pm 12V$ 的连接到此扩展板右上角的 $\pm 12V$ 输入端。另准备 1 根十芯线,连接扩展板的十芯口 J5 或 J6DA0832 其中一个数据口和主系统的 J3 口连接。用示波器连接 DA0832A 或 B 对应的 OUT/GND 口。

**(7) 下载测试:**

编程下载,进行实验验证。选择模式 5,按动键 1 输入频率字,数码 8 显示频率字数据,随着频率字越大观察示波器的正弦波形。

**(8) 实验内容 2:** 在设计中增加一些元件,设计成扫频信号源,扫频速率、扫频频域、扫频步幅可设置。所有控制可以用单片机完成,如使用 8051 核来完成。

**(9) 实验内容 3:** 将此设计改成频率可数控的正交信号发生器,即使电路输出两路信号,且相互正交,一路为正弦(sin)信号,一路为余弦(cos)信号,它们所对应 ROM 波形数据相差 90 度,可用附录 1 介绍的软件生成。

**(10) 实验内容 5:** 利用此电路设计一个 FSK 信号发生器,并硬件实现之。

**(11) 思考题:** 如果不作截断,此例的频率精度和相位精度分别是多少?

**4.35 移相信号发生器设计**

**实验路径:** \DEMO\EDA\_DEMO\MODU\_DEMO\DEMO11\_DDS\_PHASE\_WAVE

**(1) 实验目的:**

了解移向控制的方法。

## (2) 实验原理:

图 4-62 是基于 DDS 模型的数字移相信号发生器的电路模型图。FWORD 是 10 位频率控制字，控制输出信号的频率；PWORD 是 10 位相移控制字，控制输出信号的相移量；ADDER32B 和 ADDER10B 分别为 32 位和 10 位加法器；SIN\_ROM 是存放正弦波数据的 ROM，10 位数据线，10 位地址线，设其中的数据文件是 LUT10X10.mif，可由附录中的软件或用 MATLAB 生成；REG32B 和 REG10B 分别是 32 位和 10 位寄存器；POUT 和 FOUT 分别为 10 位正弦信号输出，可以分别与 DA0832 或两个高速 10 位 D/A 相接，它们分别输出参考信号和可移相正弦信号。它的地址线没有经过移相用的 10 位加法器，而直接来自相位累加器，所以用于基准正弦信号输出。180MHz 时钟来自锁相环。

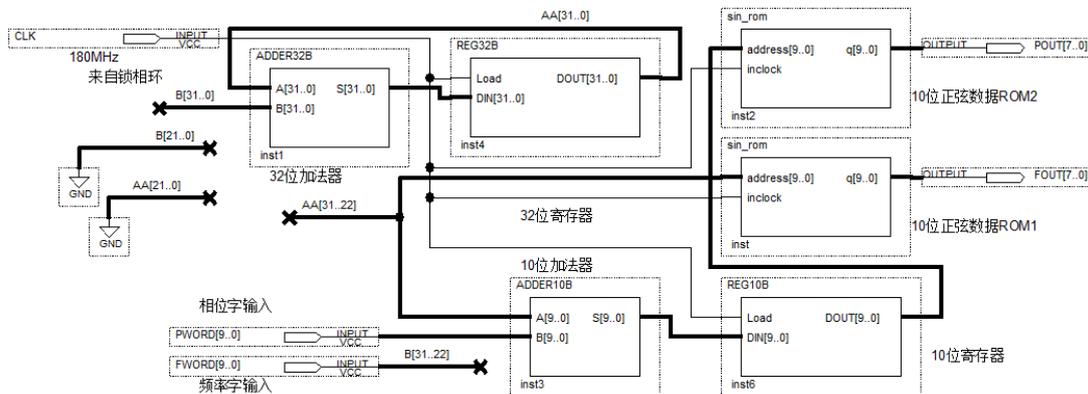


图 4-62 全数字移相信号发生器电路原理图

## (3) 硬件准备:

需要一块扩展板，详细浏览 2.3.11，此板的 J5/J6 口分别是双 DA0832 是 A 和 B 的 D0~D7 口的数据口，这里通过 FPGAIO 口连接这两十芯口进行控制。

## (4) 引脚锁定:

主频时钟 CLK 可以接核心板上，或锁定主系统 J17 的 CLKB0；

正弦波基准 PA[7..0] (分别接: DA25\DA24\DA29\DA28\DA27\DA26\DA23\DA22(主系统 J3))

待移相正弦波 PB[7..0] (分别接: DA21\DA20\DA19\DA18\DA17\DA16\DA15\DA14(主系统 J2))

选择模式 5

p 相位字输入锁定键 1 (PIO0)；

F 频率字输入锁定键 2 (PIO1)；

FF 频率字显示[3..0]锁定数码 8(PIO44~PIO47)。

PP 频率字显示[3..0]锁定数码 7(PIO40~PIO43)。

根据 2.2/2.4 查表实验方法查找引脚锁定。

## (5) 硬件连接

扩展板插主系统时需注意，因 DA 涉及到运放需 $\pm 12V$ ，在扩展的 5 组座中，其中“实验模块 3 (J29/J30)”上输出 $\pm 12V$  供给此板，直接插上此板在此座中，自动供给 $\pm 12V$ ，不需外接 $\pm 12V$ ，或另一种方法：插任意座，可利用主系统上“电源指示区”上的输出的 $\pm 12V$  的连接到此扩展板右上角的 $\pm 12V$  输入端。另准备 2 根十芯线，连接扩展板的十芯口 J5 和 J6 的 DA0832 两个数据口和主系统的 J3、J2 口连接。用示波器两路 CH1/2 连接 DA0832A 和 B 对应的 OUT/GND 口。

## (6) 下载测试:

编程下载，进行实验验证。示波器出现两路正弦波。选择模式 5，按动键 2 输入频率字，数码 8 显示频率字数据，随着频率字越大观察示波器的正弦波形频率越大。按动键 1 输入相位字，数码 7 显示相位字数据，随着相位字越大观察示波器的正弦波形向右移动的相位越大。

**实验任务 1:** 完成 10 位输出数据宽度的移相信号发生器的设计，要求使用锁相环，设计正弦波形数据 MIF 文件；给出仿真波形；最后进行硬件测试。

**思考题 1:** 如果频率控制字宽度是 32 位，相位控制字宽度是 10 位，输出 10 位，时钟为 60MHz。计算频率、

相位和幅度三者分别的步进精度是多少，给出输出频率的上下限。

**思考题 2:** 给出基于此项设计的李萨如图信号发生器的设计方案。

## 4.36 高速 ADC\_DAC 存储示波器

实验路径: \DEMO\EDA\_DEMO\MODU\_DEMO\DEMO12\_Hi\_Seed\_DAC

### (1) 实验目的

学习控制高速 AD 采样再通过 DA 输出。

### (2) 实验原理

做一个控制 AD5540 采样的控制电路及存储器，时钟是通过 FPGA 给出，存再 32 位 ROM 中，再截取高 10 位连接一路 DA5651，转换模拟信号输出，另一路 DA 利用输入的频率字利用存在 ROM 里的正弦波数据输出，可参考实验 4.32，接示波器，可观察采样波形。

### (4) 硬件准备:

需要一块扩展板，详细浏览 2.3.12，四组十芯口从左分别是 AD5540/DA5651A/时钟接入/DA5651B 数据口，通过主系统输出的 IO 口连接扩展板的 4 组十芯口。

### (5) 引脚锁定:

主频时钟 CLK 可以接核心板上，或锁定主系统 J17 的 CLKB0;

DIN[7..0]是 AD5540 转换的数据输出端口锁定 DB0\DB1\DB3\DB5\DB6\DB7\DB8\DB9(主系统 J6)

PA[9..0]正弦波输出给一路 DA(分别接:DA10\DA11\DA1\DA0\DA2\DA3\DA4\DA5\DA6\DA7(主系统 J8\J9)

PB[9..0] 采样数据输出给另一路 DA (分别接:

DA13\DA12\DB10\DB11\DB12\DB13\DBT0\DB14\DBT1\DBT1\DB15(主系统 J7\J9)

ADCLK 是通过 FPGAIO 口输入给 AD5540 锁定 DA14; CLK1/CLK2 分别是两路 DA5651 的输入时钟，分别锁定 DAT0/DA9。

选择模式 5

F 频率字输入锁定键 2 (PIO1);

FF 频率字显示[3..0]锁定数码 3(PIO24~PIO27)。

根据 2.2/2.4 查表实验方法查找引脚锁定。

### (6) 硬件连接

扩展板插主系统时需注意，因 DA 涉及到运放需+-12V，在扩展的 5 组座中，其中“实验模块 3 (J29/J30)”上输出+-12V 供给此板，直接插上此板在此座中，自动供给+-12V，不需外接+-12V，或另一种方法：插任意座，可利用主系统上“电源指示区”上的输出的+-12V 的连接到此扩展板右上角的+-12V 输入端。另准备 4 根十芯线，一端分别连接扩展板的四组十芯口：

扩展板“ADC\_DATA”连接主系统 J6;

扩展板“DA\_B\_DATA”连接主系统 J7;

扩展板“MIXP”连接主系统 J9;

扩展板“DA\_A\_DATA”连接主系统 J8;

用示波器两路 CH1/2 连接两路 DA 和 A 和 B 对应的 OUT/GND 口。

### (7) 下载测试:

编程下载，进行实验验证。用双路示波器观察。选择模式 5，按动键 1 输入频率字，数码 3 显示频率字数据，随着频率字越大观察示波器的一路正弦波形频率越大。用一根单线，一端连接主系统时钟源区选择一个频率，比如 4096Hz 以下，另一段连接扩展板 AD5540 的模拟信号输入端“AIN”端，作为模拟信号输入，观察另一路示波器波形，应该是显示方波。

## 4.37 PS2 键盘控制模型电钢琴子电路设计

实验路径: \DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO13\_PS2\_PIANO

## (1) 实验目的:

学习对 PS2 键盘数据程序的设计, 以及掌握 PS2 键盘应用技术。

## (2) 实验原理:

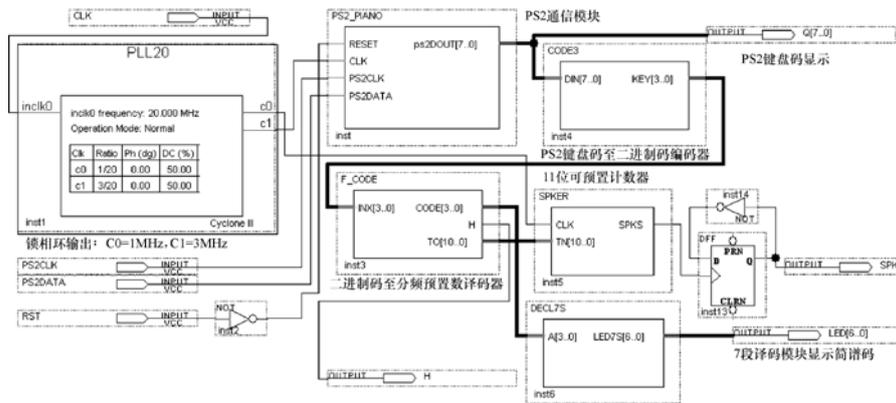


图 4-63 PS2 键盘控制模型电子琴电路顶层设计

图 4-63 是 PS2 键盘控制模型电子琴电路顶层设计。除了 PS2 通信模块 PS2\_PIANO (例 4-37), CODE3 模块等稍有不同外, 此电路所有其他模块及电路功能与以上实验 4.18 完全相同, 工作原理也类似。对此不再重复说明。

表 4-3 是 PS2 键盘键控与输出码对照表。PS2 键盘接口是个 6 脚连接器。其 4 个脚的功能分别是时钟端口、数据端口、+5V 电源端口和电源接地端口。PS2 键盘依靠 PS2 端口提供+5V 电源。PS2 是双端口双向通信模式, 即遵循双向同步通信协议。通信的双方过时钟口同步, 然后通过数据口进行数据通信。通信中, 主机若要控制另一方通信选择, 可把时钟拉至低电平即可。PS2 通信过程中, 数据以帧为单位进行传输, 每帧包含 11 至 12 位数据, 具体方式是:

通信的双方过时钟口同步, 然后通过数据口进行数据通信。通信中, 主机若要控制另一方通信选择, 可把时钟拉至低电平即可。PS2 通信过程中, 数据以帧为单位进行传输, 每帧包含 11 至 12 位数据, 具体方式是:

1. 数据的第 1 个位起始位逻辑恒为 0;
2. 接下来是 8 个数据位, 低位在前; 时钟下降沿读取;
3. 然后是一个 1 位奇偶校验位, 作奇校验; 接下的第 11 位是停止位, 恒为 1;
4. 必要时 1 个答应位, 用于主机对设备的通信。

通信过程中当 PS/2 设备等待发送数据时, 首先检查时钟端以便确认其高电平; 如果为低电平, 则认为是主机抑制了通信, 此时必须缓存待发送的数据, 直到获得总线控制权。如果时钟信号为高电平, 则 PS/2 设备将数据发送给主机。由于 PS/2 通信协议是一种双向同步串行通信协议, 数据可以从主机发往设备, 也可以由设备发往主机。实际应用中, PS/2 键盘作为一种输入设备, 都是由键盘往主机发送数据, 主机读取数据。这时由 PS/2 键盘产生时钟信号, 发送数据时按照数据帧格式顺序发送。其中数据位在时钟为高电平时准备好, 在时钟的下降沿主机就可以读取数据。因此通常情况只有键盘向主机发送数据, PS2 键盘的两个工作端口都是单向输出口。对于 PS2 通信的详细情况可参考相关资料。

**KX-CDS 实验系统演示示例(含实验指导 pdf/PDF 课件):** \DEMO\EDA\_DEMO\MODU\_DEMO\MODU\_实验示例演示\DEMO13\_PS2 键盘控制电子琴设计.ppt 模块是用状态机实现的, CLK 是状态机工作时钟)。

## 【例 4-37】

文本: ps2\_PIANO

表 4-3 PS2 键盘键控与输出码对照表

Key	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Data	1C	32	21	23	24	2B	34	33	43	3B	42	4B	3A	31	44
Key	P	Q	R	S	T	U	V	W	X	Y	Z	0	1	2	3
Data	4D	15	2D	1B	2C	3C	2A	1D	22	35	1A	45	16	1E	26
Key	4	5	6	7	8	9	`	-	=	\	]	;	'	,	.
Data	25	2E	36	3D	3E	46	0E	4E	55	5D	5B	4C	52	41	49
Key	/	[	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	KP0
Data	4A	54	05	06	04	0C	03	0B	83	0A	01	09	78	07	70
Key	KP1	KP2	KP3	KP4	KP5	KP6	KP7	KP8	KP9	KP.	KP-	KP+	KP/	KP*	END
Data	69	72	7A	6B	73	74	6C	75	7D	71	7B	79	4A	7C	69
Key	BKSP	SPACE	TAB	CAPS	LSHFT	LCTRL	LCUI	LALT	R SHFT	R CTRL	R CUI				
Data	66	29	0D	58	12	14	1F	11	59	14	27				
Key	R ALT	APPS	ENTER	ESC	INSERT	HOME	PG UP	DELETE	PG DN	NUM					
Data	11	2F	5A	76	70	6C	7D	71	7A	77					
Key	UARROW	LARROW	D ARROW	R ARROW	KP EN	SCROLL	PRNT SCR N	PAUSE							
Data	75	6B	72	74	5A	7E	12	7C	14						

### (3) 硬件准备:

准备一个 PS/2 的键盘和 1 根单线。

### (5) 引脚锁定:

PS/2 的引脚 KX-4CE6/10 上已标出, KX-4/10CE55 在“核心板说明”文档里, DE 板在相关原厂提供的相关资料里查找, DE1-S0C 引脚资料如下表 4-4: 有两组 IO 口连接, 可选择其中一组。

表 4-4 Table 3-21 Pin Assignment of PS/2

Signal Name	FPGA Pin No.	Description	I/O Standard
PS2_CLK	PIN_AD7	PS/2 Clock	3.3V
PS2_DAT	PIN_AE7	PS/2 Data	3.3V
PS2_CLK2	PIN_AD9	PS/2 Clock (reserved for second PS/2 device)	3.3V
PS2_DAT2	PIN_AE9	PS/2 Data (reserved for second PS/2 device)	3.3V

PS2CLK 和 PS2DATA 分别查找相关资料。

CLK 接 DE 核心板 50MHz, KX 板接 20MHz 时钟。

选择模式 5 电路:

RST 锁定键 1 (PI00)。

H 高音显示锁定 D1 (PI08)。

SPK 蜂鸣器锁定 DA0。

CD[3..0] 显示简码锁定数码管 1 (PI019~PI016)。

CODE 简谱码[7..0] 键盘码锁定数码 3、2 (PI027~PI020)。

### (6) 硬件连接

准备 1 根单线一端连接 J8 的 DA0 端, 另一端连接 J16 的蜂鸣器输入端。把准备好的 PS/2 键盘口插入核心板的对应端口。

### (7) 下载测试:

编程下载, 复位主系统, 进行实验验证。选择模式 5, 键 1 复位, 按动键盘按键, 观察数码 1 上的简谱码和蜂鸣器发出的音阶及数码 2/3 的键盘码。

## 4.38 VGA 彩条信号显示控制电路设计

实验路径: \ DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO14\_VGA

### (1) 实验目的:

学习 VGA 图像显示控制电路设计。

### (2) 实验原理:

计算机显示器的显示有许多标准,常见的有 VGA、SVGA 等。一般这些显示控制都用专用的显示控制器(如 6845)。在这里不妨尝试用 FPGA 来实现 VGA 图像显示控制器,用以显示一些图形、文字或图象,这在产品开发设计中有许多实际应用。

常见的彩色显示器一般由 CRT (阴极射线管) 构成,彩色是由 R、G、B (红: Red, 绿: Green, 蓝: Blue) 三基色组成,用逐行扫描的方式解决图象显示。阴极射线枪发出电子束打在涂有荧光粉的荧光屏上,产生 R、G、B 三基色,合成一个彩色像素。扫描是从屏幕的左上方开始的,从左到右,从上到下,进行扫描。每扫完一行,电子束回到屏幕的左边下一行的起始位置,在这期间,CRT 对电子束进行消隐,每行结束时,用行同步信号进行行同步;扫描完所有行,用场同步信号进行场同步,并使扫描回到屏幕的左上方,同时进行场消隐,预备下一场的扫描。

对于普通的 VGA 显示器,其引出线共含 5 个信号,即: R、G、B 是三基色信号; HS 是行同步信号; VS 是场同步信号。对于 VGA 显示器的 5 个信号的时序驱动要注意严格遵循“VGA 工业标准”,即  $640 \times 480 \times 60\text{Hz}$  模式。图 6-33 是 VGA 行扫描、场扫描的时序图,表 6-2、6-3 分别列出了它们的时序参数。VGA 工业标准要求的频率:

时钟频率 (lock frequency) : 25.175 MHz (像素输出的频率)

行频 (Line frequency) : 31469 Hz

场频 (Field frequency) : 59.94 Hz (每秒图像刷新频率)

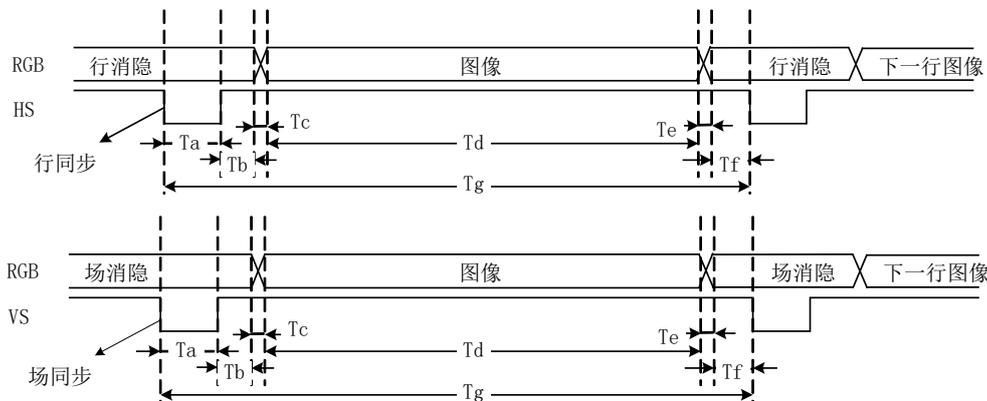


图 4-64 VGA 行扫描、场扫描时序示意图

VGA 工业标准显示模式要求: 行同步、场同步都为负极性,即同步头脉冲要求是负脉冲。设计 VGA 图像显示控制要注意两个问题: 一个是时序驱动,这是完成设计的关键,时序稍有偏差,显示必然不正常;另一个是 VGA 信号的电平驱动(注意 VGA 信号的驱动电平是模拟信号),详细情况可参考相关资料。对于一些 VGA 显示器,HS 和 VS 的极性可正可负,显示器内可自动转换为正极性逻辑。在此以正极性为例,说明本示例中的 CRT 工作过程: R、G、B 为正极性信号,即高电平有效。

当  $VS=0$ 、 $HS=0$  时,CRT 显示的内容为亮,此过程即正向扫描过程,约需  $26\mu\text{s}$ 。当一行扫描完毕,行同步  $HS=1$ ,约需  $6\mu\text{s}$ ;其间,CRT 扫描产生消隐,电子束回到 CRT 左边下一行的起始位置 ( $X=0$ ,  $Y=1$ );当扫描完 480 行后,CRT 的场同步  $VS=1$ ,产生场同步使扫描线回到 CRT 的第一行第一列 ( $X=0$ ,  $Y=0$ ) 处(约为两个行周期)。HS 和 VS 的时序图如图 4-65 所示:  $T_1$  为行同步消隐(约为  $6\mu\text{s}$ );  $T_2$  为行显示时间(约为  $26\mu\text{s}$ );  $T_3$  为场同步消隐(两行周期);  $T_4$  为场显示时间(480 行周期)。

为了节省存储空间,本示例中仅采用 3 位数字信号表达 R、G、B (纯数字方式): 三基色信号,因此仅可显示 8 种颜色,表 4-7 是此 8 色对应的编码电平。例 6-2 程序设计的彩条信号发生器可通过外部控制产生如下 3 种显示模式,共 6 种显示变化(表 4-58)。

图 4-65 是对应例 4.38 的 VGA 图像显示控制器接口电路图。首先按照图 4-66 的方式将 VGA 显示器(液晶或

CRT 管都可) 插入 CDS 系统的 VGA 接口。将编译文件下载进 FPGA 后, 即可控制键 K1, 每按一次键换一种显示模式, 6 次一循环, 其循环显示模式分别为: 横彩条 1、横彩条 2、竖彩条 1、竖彩条 2、棋盘格 1 和棋盘格 2。时钟信号必须是 20MHz, 如果是 12MHz 或 50MHz, 则必须改变程序中的分频控制, 或外围做锁相环输出为 20MHz, 对此例 4-38 已作了标注。

表 4-5 行扫描时序要求: (单位: 像素, 即输出一个像素 Pixel 的时间间隔)

		行同步头			行图像		行周期
对应位置	Tf	Ta	Tb	Tc	Td	Te	Tg
时间(Pixels)	8	96	40	8	640	8	800

表 4-6 场扫描时序要求: (单元: 行, 即输出一行 Line 的时间间隔)

		行同步头			行图像		行周期
对应位置	Tf	Ta	Tb	Tc	Td	Te	Tg
时间(Lines)	2	2	25	8	480	8	525

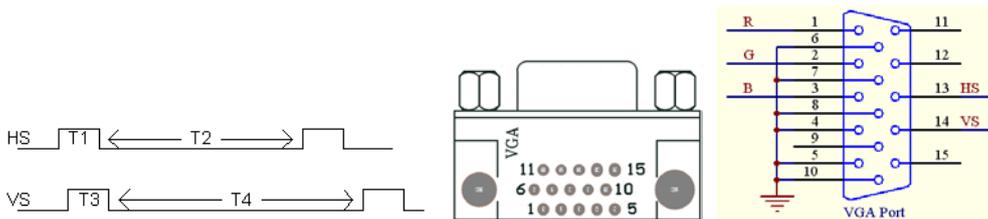


图 4-65 HS 和 VS 的时序图

图 4-66 VGA 接口电路图, 左接口从上往下看

表 4-7 颜色编码:

颜色	黑	蓝	红	品	绿	青	黄	白
R	0	0	0	0	1	1	1	1
G	0	0	1	1	0	0	1	1
B	0	1	0	1	0	1	0	1

表 4-8 彩条信号发生器 3 种显示模式,

1	横彩条	1: 白黄青绿品红蓝黑	2: 黑蓝红品绿青黄白
2	竖彩条	1: 白黄青绿品红蓝黑	2: 黑蓝红品绿青黄白
3	棋盘格	1: 棋盘格显示模式 1	2: 棋盘格显示模式 2

【例 4-38】

文本: COLOR

(3) 实验内容 1:

根据 VGA 的工作时序, 详细分析并说明例 4.33 程序的设计原理, 给出仿真波形, 并说明之。然后完成 VGA 彩条信号显示的硬件验证实验。

(3) 硬件准备:

FPGA 核心板集成 VGA 接口。

(5) 引脚锁定:

选择模式 3 电路:

显示模式锁定键 1 (PI00)。

R、G、B、HS、VS 的引脚, 如果是 KX-4CE6/10/55 在核心板上已标出, 如是 DE 系列板的引脚请浏览“DE 核心板使用及引脚说明”文件夹。

CLK 锁定核心板 20MHz (KX 板已标注在板上, 如果核心板 DE 是 50MHz, 可以利用连线连接主系统 20MHz, 或锁相环, 或修改语言控制的时序和实际提供的时钟相匹配。

(6) 硬件连接

将显示器的 VGA 接线口连接核心板上的 VGA 口。

#### (7) 下载测试:

编程下载, 复位主系统, 进行实验验证。选择模式 3, 按动键 1, 观察显示器上彩条的变化。

(8) **实验任务 1:** 设计可显示横彩条与棋盘格相间的 VGA 彩条信号发生器。

(9) **实验任务 2:** 设计可显示英语字母的 VGA 信号发生器电路。

(10) **实验内容 3:** 设计可显示移动彩色斑点的 VGA 信号发生器电路。

## 4.39 基于 FT245BM 的 USB 通信控制模块设计

实验路径: \ DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO16\_USB\_FT245

#### (1) 实验目的:

通过上位机软件设计 PC 与 FT245USB 口相互通信原理

#### (2) 实验任务:

本项实验设计实用意义明显。即利用 FT245BM 进行 USB 通信, 进而设计相关产品, 如 USB-Blaster、虚拟仪表、高速采样模块、编程器、仿真器、图像采集与处理等等, 十分常用。

#### (3) 实验原理:

本实验是利用 FPGA 的控制电路 (实用产品中为降低成本多使用 CPLD) 与 FT245BM 通信, 通过此器件, 经由 USB 接口与 PC 机通信。FT245BM 的开发资料可参考本实验文件夹下面的说明文档, 驱动软件: \ VIVI\FT245BM\_USB 驱动\FT245BM\_USB。

根据相关 FT245 资料原理, 编写代码, 例 4.39

例 4.39

文本: usb245i

#### (4) 硬件准备:

**需要一块扩展板, 详细浏览 2.3.16, 上方两个十芯口分别是连接 FPGA 的 FT245 控制和数据端口。**

#### (5) 引脚锁定:

主频时钟 CLK 锁定 KX 核心板上 20M 时钟, DE 系列板锁定主系统上 J17 的 CLKB0;

USB\_d[7..0]通信数据口分别锁定 DB9\DB8\DB7\DB6\DB5\DB3\DB1\DB0(主系统 J6)

usb\_rd\usb\_wr\usb\_rxe\usb\_txe\usb\_siwu 通信控制口分别锁定 DB10\DB11\DB13\DBT0(主系统 J7)

选择模式 5

reset 作为系统的复位选择键 1 锁定 PIO0, sendkey 作为系统上向 PC 发送数据选择键 2 锁定 PIO1, key1 和 key2 分别作为系统上向 PC 待发送的 8 位数据键分别选择键 2 和键 3, 锁定 PIO2\PIO3, 前面为地位, LL[3..0]作为待发送低 4 位数据的显示选择数码 1 锁定 (PIO16~PIO19); HH[3..0]作为待发送高 4 位数据的显示选择数码 2 锁定 (PIO20~PIO23); H[7..0]作为 PC 机发来的接收数据显示选择数码 3 和 4 锁定 (PIO24~PIO31)。

#### (6) 硬件连接

准备 1 根单线两个十芯线, 1 根 USB 线, 单线一端主系统就 J17 的 CLKB0 端, 另一端连接主系统时钟源 20MHz, 一根十芯线连接板扩展板的左边十芯口和主系统 J6 口连接, 另一个十芯线板扩展板的右边十芯口和主系统 J7 口连接; USB 线把把扩展板和 PC 机连接, 当连接好时, PC 机跳出安装 USB 驱动, 路径: VIVI\_FILE\FT245BM\_USB\FTD2XX.d11, 安装好以后, 资源管理器上会出现如图 4-67。好了以后准备下载测试。

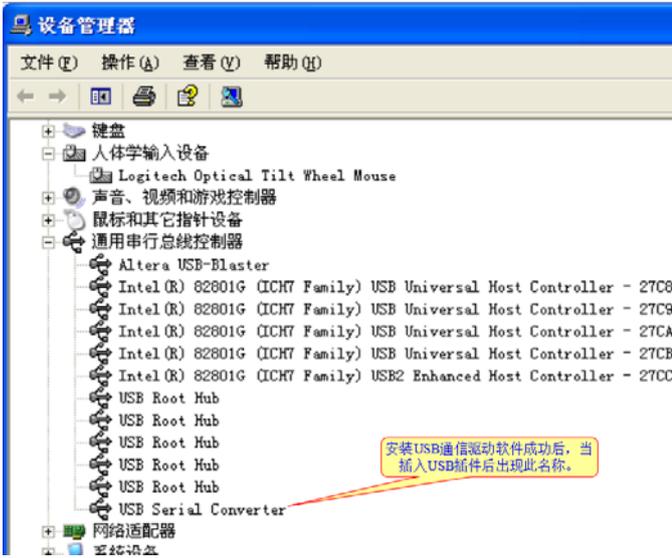


图 4-67

名称	修改日期	类型	大小
DRIVERS	2017-07-17 13:30	文件夹	
EE.BIN	2017-12-01 16:17	BIN 文件	1 k
USBDEMO	2005-05-02 14:02	应用程序	216 k

图 4-68

(7) 下载测试:

编程下载, 复位主系统, 进行实验验证。打开通信软件, 在本例程文件夹下: TestSoft\USBDEMO, 如图: 4-68, 打开后如图 4-69, 上面窗口是接收系统上的信号, 下面是向系统上发生信号。选择模式 5, 按动键 1 复位, 键 1 和键 2 分别输入待像 PC 发送数据, 比如: B5, 显示数码 1、2; 按动键 2 发送, 这样图 4-69 的接收信息窗口会出现发送来的数据 B5; 反过来, 在窗口下输入待向系统发送的数据, 比如: 3A, 在下发选择“十六进制显示”, 再点击“发送十六进制”按钮, 在系统的数码 3 和 4 会显示同样的数据。



图 4-69

### 4.40 五功能智能逻辑笔设计

实验路径: \ DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO18\_LOGIC\_PEN

(1) 实验目的:

学习用状态机设计实用电路。

(2) 实验原理:

图 4-70 是 5 功能智能逻辑笔电平信号采样电路。它有 3 个端口与 FPGA 相接: Vo1, Vo2 和 TEST。Vo1 和 Vo2 输入进 FPGA; TEST 由 FPGA 输出。此信号从 FPGA 端口通过一个 TTL 两反向器与 TEST 相接, 因为 FPGA 输出的 3.3V 电平不够高, 驱动力也不够。

设计前首先查阅有关 LM393 的资料，它是一个双比较器。在图 4-67 的 LM393 元件的第 3、6 脚是二比较器的两个输入端。左端 3 脚接参考电压  $V_{rh}=2.6V$ ，作为高电平的分界线；右端 6 脚接参考电压  $V_{rl}=0.9V$ ，作为低电平的分界线。

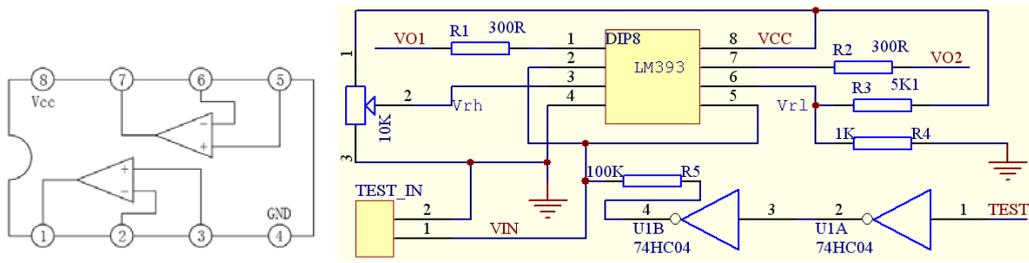


图 4-70 五功能智能逻辑笔电平信号采样电路，左图是 LM393 引脚图

另外请注意电平测试口  $V_{in}$  除与两个比较器的输入端 2 和 5 相连外，还通过一个 100K 欧姆的电阻与 FPGA 的输出口相接。这是测试高阻态必须电阻。还有，接输出电平  $V_{o1}$  和  $V_{o2}$  口处都应该分别接 5K1 上拉电阻于 3.3V 电平，因为 FPGA 的 I/O 高电平是 3.3V。（对于 KX-CDS 系统，可选择对应模块完成此实验）

通过 FPGA 的状态机对“TEST”分别输出 1 或 0，以及结合来自  $V_{o1}$  和  $V_{o2}$  测试到的电平组合，完全可以判断出测试端  $V_{in}$  的逻辑信号是高电平、低电平、中电平、高阻态还是连续脉冲信号。

可以这样来定义：测到的电平  $V_{in}$  大于  $V_{rh}$ ，判定为高电平 1；小于  $V_{rl}$ ，则判定为低电平 0；若所测结论是  $V_{rl} < V_{in} < V_{rh}$ ，则判定为中电平，即不稳定电平；再若 TEST 输出 1 时判定为高电平，TEST 输出 0 时判定为低电平，则最终判定为高阻态。又若高低电平不断变化，则判为连续脉冲。建议状态机的工作时钟频率不要太高，可以在 250Hz 左右。例【4-40】

例【4-40】

LGC\_PEN

(4) 硬件准备：

需要一块扩展板，详细浏览 2.3.18，图 4-71，左下方十芯口 FPGA 控制连接端口。

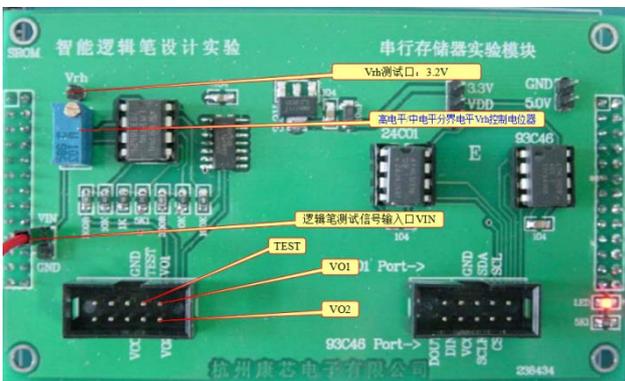


图 4-71

(5) 引脚锁定：

- 主频时钟 CLK 可以接核心板上，根据核心板的主时钟修改输入时钟或锁定主系统 J17 的 CLKB0；
- VO[0]锁定 DA25(主系统 J3)
- VO[1]锁定 DA24(主系统 J3)
- TEST 锁定 DA29(主系统 J3)
- 选择模式 5
- LED[3..0]锁定 D4~D1 (PIO11~PIO8)；分别指示“D4:高阻态、D3:中电平、D2:高电平、D1:低电平
- 根据 2.2/2.4 查表实验方法查找引脚锁定。

(6) 硬件连接

模块插主系统右方四组插座，任意一组都可以。  
扩展板左下方十芯口和主系统 J3 口用十芯线连接；

(7) 下载测试：

编程下载，进行实验验证。选择模式 5，用一根单线，一端连接扩展板的“VIN”端口，另一端连接要测试的电平，如果是低于 2.5V 的 D1 亮；低于 3.3V~2.5V，D3 亮；高于 3.3V~5V 的，D2 亮；悬空的，高阻态，D4 亮。

## 4.41 无线编码收发电路设计

实验路径：\ DEMO\EDA\_DEMO\MODU\_DEMO\ DEMO19\_WRIELESS\_CORD

### (1) 实验目的：

了解 PT2262 发送及 PT2272 接收信号原理。具体这两款器件请查阅 DEMO19\_WRIELESS\_CORD 的文档说明。

### (2) 实验原理：

编码芯片 PT2262 发出的编码信号由：地址码、数据码、同步码组成一个完整的码字，由 TE 发出，解码芯片 PT2272 接收到信号后，其地址码经过两次比较核对后，VT 脚才输出高电平，与此同时相应的数据脚也输出高电平，如果发送端一直按住按键，编码芯片也会连续发射。根据以上工作原理，设计由 FPGA IO 口向 PT662 D1~D4 发送信号分别为“0000、0001、0010、... 1111，每当有一个编码，按动 TE 发射按键，这时接收端 LED 会显示相同对应的码。根据以上原理设计例【4-41】

例【4-41】

文本：code

### (3) 硬件准备：

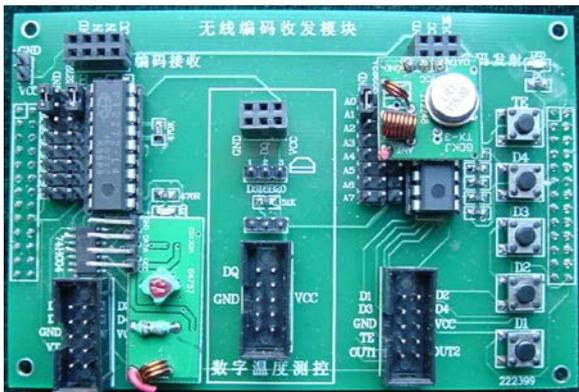


图 4-72 无线发射模块

图 4-72 此模块是基于 PT2272 和 PT2262 的数字编码无线通信收/发模块，其中最左边是接收 PT2272 模块，在对应的下方有个十芯口是相关引脚，最右边是发送 PT2262 模块，对应下方是相关引脚。测试方法，可同时按动 TE 键和 D1-D4 的任意键，如果对应的接收模块的对应的小灯亮，表示接收成功。原理图参考：“原理图\RADIO\_TRANS\_RESV\_SCH.pdf”

### (4) 引脚锁定：

模块插主系统右方四组插座，任意一组都可以，FPGA 连接方式分别模块发送十芯口（右边）连接主系统的 JP6 口。

d[3..0]锁定 DB0/DB1/DB3/DB5(输入待发送的码)。

Led[3..0]锁定模式 5 的数码 1，PIO19/PIO18/PIO17/PIO16（16 进制形式显示待发送的码）。

clk 锁定模式 5 键 1。

rst 复位清 0

以上锁定的引脚都要根据表 2.4（模块）查表对应型号的引脚号。

### (6) 硬件连接

准备 1 根十芯线连接模块右下方对应发送编码的十芯口和主系统的 J6 十芯口。

### (7) 下载测试：

编程下载，主系统复位。按动键 1，这时数码管 1 对应显示'1'，二进制码表示'0001'，待发送，再按动模块板上 TE 键发送，这时对应的模块板 LED 发光管显示'0001'接收码和对应的发送码一样。这样的方式重复上面的步骤数码管分别显示" 0~F,对应的模块板显示对应的二进制码。

#### 4.42 SDRAM 测试实验 (4S、10S、)

实验路径: \ DEMO\EDA\_DEMO\CDS\_DEMO\ DEMO41\_sdram\_test

#### 4.43 串口收发实验 (4、4S、10S、3ES、5V)

实验路径: \ DEMO\EDA\_DEMO\CDS\_DEMO\ DEMO42\_uart\_test

#### 4.44 ds1302 数码管显示 RTC 时间实验 (4、4S、10S、3ES、5V、5S)

实验路径: \ DEMO\EDA\_DEMO\CDS\_DEMO\DEMO43\_rtc\_ds1302ok

#### 4.45 sd 卡测试实验 (4S、10S、5V)

实验路径: \ DEMO\EDA\_DEMO\CDS\_DEMO\DEMO44\_sd\_card\_test

#### 4.46 SD 卡音乐播放实验 (4S、10S、5V)

实验路径: \ DEMO\EDA\_DEMO\CDS\_DEMO\DEMO45\_sd\_card\_audio

#### 4.47 SD 卡读取 BMP 图片显示实验 (4S、10S、5V)

实验路径: \ DEMO\EDA\_DEMO\CDS\_DEMO\DEMO46\_1\_sd\_sdram\_vga\DEMO46\_2\_sd\_sdram\_an070\_lcd

#### 4.48 RS485 通信实验 (4、4S、10S、3ES、5V、5S)

实验路径: \ DEMO\EDA\_DEMO\CDS\_DEMO\DEMO49\_rs485\_uart\_top

#### 4.49 温度测试实验 (4、4S、10S、3ES、5V、5S)

实验路径: \ DEMO\EDA\_DEMO\CDS\_DEMO\DEMO50\_temp\_disp

#### 4.50 HDMI 输入输出控制 (4、4S、10S、3ES、5V、5S)

实验路径: \ DEMO\EDA\_DEMO\CDS\_DEMO\DEMO51\_in\_out

# 第五章 单片机系统综合实验

对应教材：《单片机原理与应用技术》清华大学，作者：潘明 黄继业 潘松

## 5.1 单片机基本实验

### 实验 5-1. 存储器块清零程序设计

(1) 实验目的：1. 掌握存储器读写方法；2. 了解存储器的块操作方法。

(2) 实验内容：1. 指定内部 RAM 中某块的起始地址和长度，要求能将其内容清零。2. 指定外部 RAM 中某块的起始地址和长度，要求能将其内容清零。3. 指定外部 RAM 中某块的起始地址和长度，要求能将其内容置为某固定值（如 012H）。

(3) 实验说明。通过本实验，学生可以了解单片机的存储器结构及读写存储器的方法，同时也可以了解单片机编程、调试方法。

(4) 参考程序框图。1、程序参考流程如图 6-1 所示；2、参考程序如下：

```

BLOCK    EQU    0400H
          ORG    0000H
          LJMP   START
          ORG    0030H

START:    MOV    DPTR,#BLOCK    ; DPTR←起始地址
          MOV    R0,#0FFH      ; R0←块长度
          MOV    A,#0          ; A←0

LOOP:    MOVX   @DPTR,A        ; 当前地址内容清 0
          INC    DPTR          ; 地址加 1
          DJNZ   R0,LOOP       ; 计数器减 1，循环
          SJMP   $             ; 结束
          END
  
```

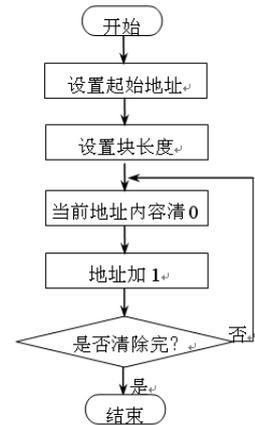


图 5-1 程序流程图

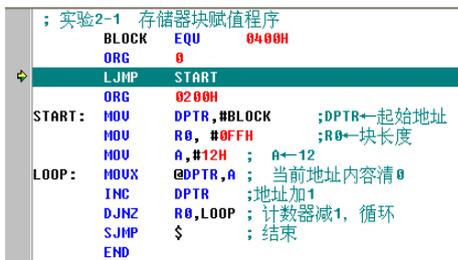


图5-2 程序窗口（存储器块赋值程序）

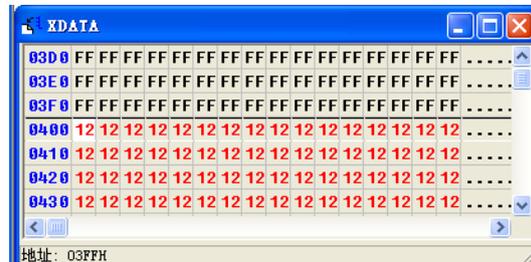


图5-3 外部数据存储器XDATA空间窗

例如若要将从 0400H 开始的 255 个字节外部数据存储器块的内容置成 12H 值，请修改以上程序完成此操作，注意在程序调试过程中数据的变化情况。注意在程序调试过程中数据的变化情况。3、程序调试过程。程序窗口如图 5-2 所示。可通过观察外部数据存储器窗口 XDATA 了解程序运行结果，运行结果如图 5-3 所示。

(5) 思考题。如何将存储器块的内容置成某固定值(例全填充为 0FFH)? 请用户修改程序，完成此操作。

### 实验 5-2 二进制到 BCD 转换程序设计

(1) 实验目的：掌握简单的数值转换算法。基本了解数值的各种表达方法。

(2) 实验内容：1、将给定的一个单字节二进制数，转换成非压缩的二—十进制(BCD)码；2、将给定的一个单字节二进制数，转换成压缩的二—十进制(BCD)码。

(3) 实验说明：计算机中的数值有各种表达方式，这是计算机的基础。掌握各种数制之间的转换是一种基本功。有兴趣的同学可以试试将 BCD 转换成二进制码。

五、参考程序框图。(1) 程序参考流程如图 5-4 所示；(2) 参考程序

通过本实验学习简单的数值转换算法，了解数值的各种表达方法。同时进一步熟悉单片机实验集成开发软件环境的使用方法。学习“单步”、“跟踪”、“执行到光标处”等调试功能。启动程序运行到最后一条指令处，打开 8051 内部数据存储器空间，观察程序执行结果。若 A 中原来的内容“123”（7BH）转换为三个 BCD 码“01”、“02”和“03”，并分别存入了从

(30H)开始的三个单元中，如图5-5所示。

程序执行后的结果：

	ORG	0000H	
	LJMP	START	
	ORG	0030H	
START:	MOV	A,#123	;A←待转换的二进制数
	MOV	B,#100	;A÷100，得百位数
	DIV	AB	
	MOV	30H,A	;保存百位数
	MOV	A,B	
	MOV	B,#10	;余数÷10
	DIV	AB	; (A)为十位，(B)为个位
	MOV	31H,A	;保存十位数
	MOV	32H,B	;保存个位数
	SJMP	\$	

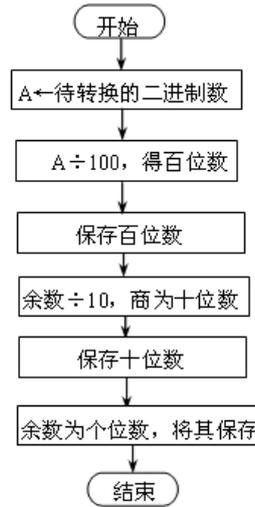


图 6-4 程序流程图

图 5-4

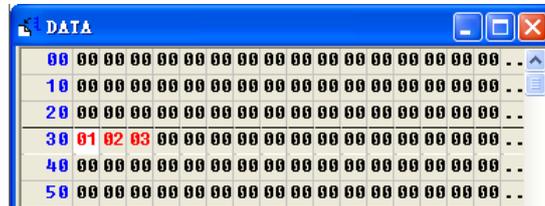


图 5-5 内部数据存储器 DATA 空间窗口

### 实验5-3 十六进制到ASCII码转换程序设计

- (1) 实验目的：1. 了解十六进制数和 ASCII 值的区别。 2. 了解如何将十六进制数转换成 ASCII 值。 3. 了解如何查表进行数值转换及快速计算。

(2) 实验内容：给出一个十六进制数，分别用查表法和逻辑运算方法将其转换成 ASCII 值。

(3) 实验说明：此实验主要让学生了解数值的十六进制数和 ASCII 码的区别，学会用查表法快速地进行数值转换并进一步了解数值的各种表达方式。

(4) 参考程序框图（查表法）

方法一：（通过查表实现数制转换）。程序流程图如图 5-6 所示。入口条件：待转换的 8 位二进制数在累加器 A 中。

出口信息：高四位的 ASCII 码在 31H 中，低四位的 ASCII 码在 30H 中。

方法 2：逻辑运算方法。HASC 功能：8 位二进制数转换成双字节 ASCII 码。入口条件：待转换的 8 位二进制数在累加器 A 中。

出口信息：高四位的 ASCII 码在 A 中，低四位的 ASCII 码在 B 中。

	ORG	0000H	
	LJMP	START	
	ORG	0030H	
START:	MOV	A,#56H	; A←待转换数 56H
	MOV	B,A	; 暂存待转换数
	LCALL	ASC_1	; 调用 ASC 查表子程序
	MOV	30H,A	; 保存高位 ASCII 码
	MOV	A,B	; 取 BCD 码低 4 位
	SWAP	A	;
	LCALL	ASC_1	; 调用 ASC 查表子程序
	MOV	31H,A	; 保存低位 ASCII 码
	SJMP	\$	; 结束
ASC_1:	MOV	DPTR,#ASCTAB	; ASC 查表子程序
	ANL	A,#0FH	
	MOVC	A,@A+DPTR	
	RET		
ASCTAB:	DB	"0123456789ABCDEF"	; 0~F 的 ASCII 码表

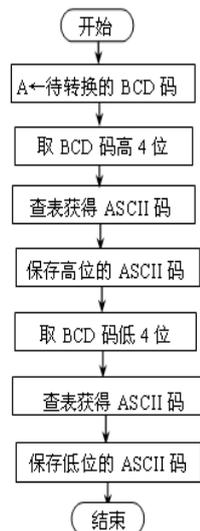


图 6-6 BIN2ASC 流程图

图 5-6

```
HASC:    MOV    B,A      ; 暂存待转换的单字节十六进制数
         LCALL HAS1     ; 转换低四位
         XCH   A,B      ; 存放低四位的 ASCII 码
         SWAP  A        ; 准备转换高四位
HAS1:    ANL   A,#0FH   ; 将累加器的低四位转换成 ASCII 码
         ADD  A,#90H
         DA   A
         ADDC A,#40H
         DA   A
         RET
```

### 实验 5-4 存储块移动程序设计

(1) 实验目的:

1. 了解内存的移动方法;
2. 加深对存储器读写的认识。

(2) 实验内容: 将指定源地址和长度的存储块移到指定目标位置。

(3) 实验说明: 块移动是计算机常用操作之一, 多用于大量的数据复制和图象操作。本程序是给出起始地址, 用地址加一方法移动块, 请思考给出块结束地址, 用地址减一方法移动块的算法。另外, 若源块地址和目标块地址有重叠, 该如何避免?

(4) 参考程序框图。

1、程序流程图如图 5-7 所示;

2、参考程序。将外部 RAM 存储器 XDATA 中 3000H 开始的 256B 数据移动到 4000H 开始的存储单元中。参考程序如上:

(5) 思考题。1.若源块地址和目标块地址有重叠, 该如何避免? 2.请思考给出块结束地址, 用地址减一方法移动块的算法。

3.如何将存储器块的内容置成某固定值(例全填充为 0FFH)? 请用户修改程序, 完成此操作。

### 实验 5-5 多分支程序

(1) 实验目的: 1. 了解程序的多分支结构。2. 了解多分支结构程序的编程方法。

(2) 实验内容: 在多分支结构的程序中, 能够按调用号执行相应的功能。

(3) 实验说明: 多分支结构是程序中常见的结构, 若给出调用号来调用子程序, 一般用查表方法, 查到子程序的地址, 转到相应子程序, 完成指定操作。分支结构程序设计也是最基本的程序设计技术。在实际的程序设计中, 有很多情况往往还需要程序按照给定的条件进行分支。这时就必须对某一个变量所处的状态进行判断, 根据判断结果来决定程序的流向。在编写分支程序时, 关键是如何判断分支的条件。在 MCS—51 单片机指令系统中 有 JZ(JNZ)、CJNE、JC(JNC)、DJNZ 及 JB (JNB) 等丰富的控制转移指令, 他们是分支结构程序设计的基础, 可以完成各种各样的条件判断、分支。

```
ORG 0
MOV R0, #30H
MOV R1, #00H
MOV R2, #40H
MOV R3, #00H
MOV R7, #0
LOOP:  MOV DPH, R0
        MOV DPL, R1
        MOVX A, @DPTR
        MOV DPH, R2
        MOV DPL, R3
        MOVX @DPTR, A
        INC R1
        INC R3
        DJNZ R7, LOOP
        LJMP $
END
```

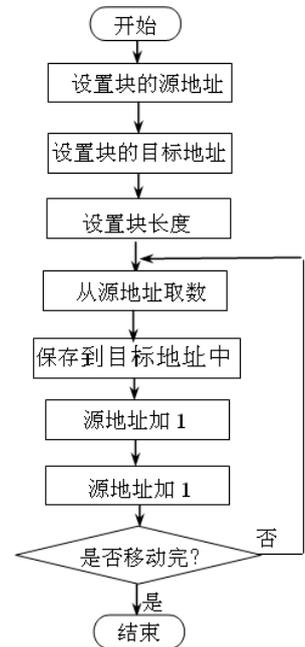


图 6-7 数据块移动程序流程图

图 5-7

常用的分支结构有以下几种:

1、统计: 例 1: 统计全班 60 名同学不及格人数, 成绩以 BCD 码放在 30H 起, 结果放入 R7。

2、分类: 例 2: 统计全班 60 名同学的成绩, 大于等于 90 分的、大于等于 75 分的、大于等于 60 分的及小于 60 分的人数, 分别放入 R3、R4、R5、R6 中。

例 1

例 2

```

MOV R0,#30H
MOV R1,#3CH
MOV R3,#00H
MOV R4,#00H
MOV R5,#00H
MOV R6,#00H
XX8: MOV A,@R0
CJNE A,#90H,XX1
XX1: JC XX2
INC R3
LJMP XX3
XX2: CJNE A,#75H,XX4
JC XX5
INC R4
LJMP XX3
XX5: CJNE A,#60H,XX6
XX6: JC XX7
INC R5
LJMP XX3
XX7: INC R6
XX3: INC R0
DJNZ R2,ZZ3
RET
ZZ3: MOV A,@R0
CJNE A,#60H,ZZ1
ZZ1: JNC ZZ2
INC R7
ZZ2: INC R0
DJNZ R2,ZZ3
RET

```

3. 程序散转：可以有多种方法实现程序散转。

```

ORG 0030H
KEY_JMP: MOV A, B
MOV DPTR,#TAB
MOVC A, @A+DPTR
JMP @A+DPTR
TAB: DB PR0-TAB
DB PR1-TAB
DB PR2-TAB
DB PR3-TAB
DB PR4-TAB
PR0: 处理程序 0
PR1: 处理程序 1
PR2: 处理程序 2
PR3: 处理程序 3
PR4: 处理程序 4
END

```

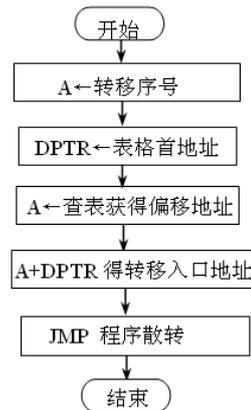


图 6-8 程序散转流程图

图 5-8

(1) 方法 1：利用地址偏移量表实现散转。将转移目标地址与表首地址差列表，作为转移目标地址。利用了伪指令的数学计算功能，标首地址加上处理程序与标首地址差转到处理程序。实验内容：有 5 个按键 0, 1, 2, 3, 4，根据按下的键转向不同的处理程序，分别为 PR0, PR1, PR2, PR3, PR4。

(2) 方法 2：利用转移指令表实现转移。将转移到不同程序的转移指令列成表格，判断条件后查表，执行表中的转移指令。用直接转移指令组成一个转移表，然后把菜单号读入累加器(A)，转移表首地址放入 DPTR 中,利用 JMP @A+DPTR 实现散转。【入口参数】(R3)=存输入键盘码 1~9。 【出口参数】转移到相应的子程序入口。

```

ORG 0030H
KEY_JMP: MOV DPTR, #TAB1 ; 子程序入口首地址送 DPTR
MOV A, R3 ; 把键盘输入缓冲区内容送累加器(A)
DEC A ; 由于输入数码为 1~9, 因此需减 1
MOV B, #03H
MUL AB ; 由于长跳转指令 LJMP 占用 3B, ; 各子程序入口地址相距 3B
JMP @A+DPTR ; 根据输入码, 执行相应的子程序
ORG 0500H ; 程序入口地址表
TAB1: LJMP NO1
LJMP NO2
LJMP NO3
LJMP NO4

```

```

LJMP NO5
LJMP NO6
LJMP NO7
LJMP NO8
LJMP NO9
END

```

(3) 方法 3: 利用转向地址表实现转移。将转移地址列成表格, 将表格的内容作为转移的目标地址。根据 R3(0~n)的内容转向对应的程序; 处理程序的入口符号地址分别为 PR0~PRn( $n < 256$ )

- ① 将 PR0 ~PRn 入口地址列在表格中, 每一项占两个单元;
- ② PRn 在表中的偏移量为 2n, 因此将 R3 的内容乘“2”即得 PRn 在表中的偏移地址;
- ③ 从偏移地址 2n 和 2n+1 两个单元分别取出 PRn 的高 8 位地址和低 8 位地址送 DPTR 寄存器, 用 JMP @A+DPTR 指令(A 先清零)即转移到 PRn 入口执行。

【入口参数】(R3)=0~n 转移序号。【出口参数】转移到相应的子程序入口。

```

                ORG    0030H
KEY_JMP:       MOV    DPTR, #TAB
                MOV    A,R3
                ADD A,R3    ; (A)←(R3)*2
                JNC    NADD
                INC    DPH    ; (R3*2)>256
NADD:          MOV    R3,A
                MOVC   A, @A+DPTR
                XCH A, R3    ; 转移地址高 8 位
                INC A
                MOVC   A, @A+DPTR
                MOV    DPL, A ; 转移地址低 8 位
                MOV    DPH, R3
                CLR A
                JMP @A+DPTR
TAB:           DW    PR0,PR1,PR2, . . . . . PRn
PR0:           处理程序 0
PR1:           处理程序 1
                ...
                ...
PRn:           处理程序 n
                END

```

实验要求: 根据以上实例程序, 按以下要求编写程序

1、全班30个人的成绩放在1000H起, 统计及格和不及格的情况, 30H放及格人数, 31H开始放成绩, 50H放不及格人数, 51H起放成绩。

2、统计全班30名同学的成绩(20H起), 大于等于90分的、大于等于75分的、大于等于60分的及小于60分的人数, 分别放入 R3、R4、R5、R6 中, 找出成绩最高的放入50H起的单元。

(4) 实验报告要求

1. 根据实验实例程序, 完成实验要求中的程序以及程序的流程图。
2. 观察数据窗口中数据的变化过程。记录实验过程中遇到的问题和解决方法。 四、

(5) 思考题。1、分类程序中, 如何设计判断的次序使90分、75分、60分合理的归档? 2、针对不同的判断条件, 相应的流程图应如何正确画出以使程序的编写更为清晰?

## 实验5-6 数据排序程序设计

(1) 实验目的: 1. 了解数据排序的简单算法。2. 了解数列的有序和无序概念。

(2) 实验内容: 给出一组随机数, 编写程序将此组数据排序, 使之成为有序数列。有序的数列更有利于查找。

```

QUE: CLR 00H      ;清除交换标志
      MOV R3,#9H   ;10 个数据循环次数
      MOV R0,#50H  ;设置数据区地址
L2:   MOV A,@R0    ;取数据 A
      INC R0       ;地址加 1
      MOV R2,A
      SUBB A,@R0   ; A 与 B 比较
      MOV A,R2
      JC L1
      SETB 00H     ;设置交换标志
      XCH A,@R0    ;交换 A、 B 位置
      DEC R0
      XCH A,@R0
      INC R0
L1:   DJNZ R3,L2   ;数据比较完?
      JB 00H,QUE   ;全部有序?
      RET          ;结束

```

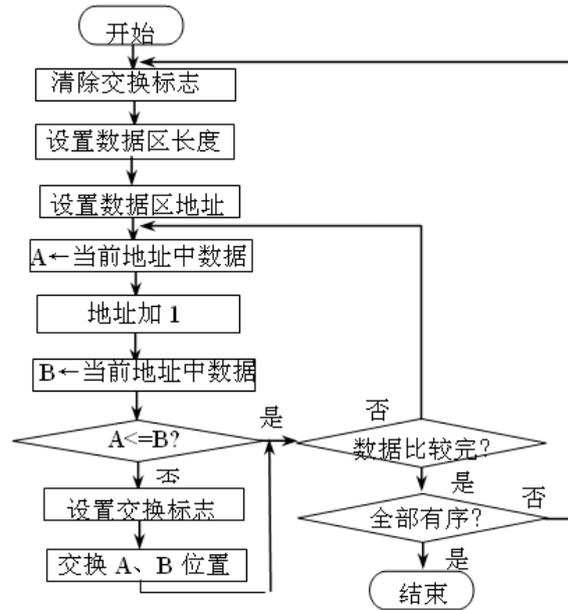


图 6-9 排序程序流程图

图 5-9

(3) 实验说明：有序的数列更有利于查找。本程序用的是“冒泡排序”法，算法是将一个数与后面的数相比较，如果比后面的数大，则交换，如此将所有的数比较一遍后，最大的数就会在数列的最后面。再进行下一轮比较，找出第二大数据，直到全部数据有序。通过本实验可以了解数据有序和无序的概念以及数据排序的简单算法。

(4) 参考程序框图。1、程序参考流程如图 5-9 所示。2、参考程序：将片内 RAM50H-59H 中的数据按从小到大的顺序排序

3、调试步骤①设置待排序数字。点击存储器空间菜单中的“DATA”选项，打开单片机内部数据存储器空间DATA，将鼠标指向其中从50H 开始的单元，单击右键，将其中内容修改为一组无序的数据，如图5-10所示，

②运行程序。点击程序运行键，再用“执行到光标处”功能，启动程序运行到最后一条指令处，可以看到此时数据已经按从小到大的顺序排列了，如图6-11所示。

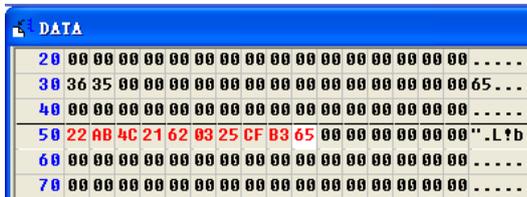


图5-10 未排序数据

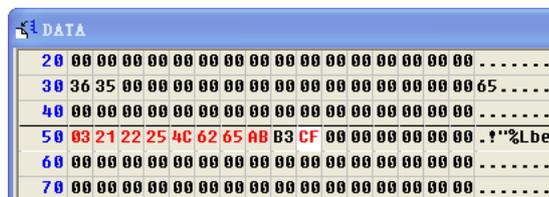


图5-11排序后的数据

### 实验 5-7 P1 口输入、输出实验

(1) 实验目的：1. 学习 P1 口的使用方法。2. 学习延时子程序的编写和使用。

(2) 实验内容：P1 口做**输出**口，接八只发光二极管（其输入端为高电平时发光二极管点亮），编写程序，使发光二极管循环点亮。P1.0、P1.1 作**输入**口接两个拨动开关 S0、S1；P1.2、P1.3 作**输出**口，接两个发光二极管，编写程序读取开关状态，将此状态在发光二极管上显示出来。编程时应注意 P1.0、P1.1 作为输入口时应先置 1，才能正确读入值。

(3) 实验电路连线。

- |                   |                   |
|-------------------|-------------------|
| P1.0 ●-----● LED0 | P1.0 ●-----● S0   |
| P1.1 ●-----● LED1 | P1.1 ●-----● S1   |
| P1.2 ●-----● LED2 | P1.2 ●-----● LED2 |
| P1.3 ●-----● LED3 | P1.3 ●-----● LED3 |
| P1.4 ●-----● LED4 |                   |
| P1.5 ●-----● LED5 |                   |
| P1.6 ●-----● LED6 |                   |
| P1.7 ●-----● LED7 |                   |

实验 1-1: P1 口循环点灯

实验 1-2: P1 口输入输出

(4) 实验说明。P1口是准双向口。它作为输出时与一般的双向口使用方法相同。由准双向口结构可知当P1口用作输入时必须先对它置“1”。若不先对它置“1”，读入的数据是不正确的。延时子程序的延时计算问题

对于程序 Delay: MOV R7,#200 DEL1: MOV R6,#123 NOP DEL2: DJNZ R6, DEL2 DJNZ R7, DEL1 RET	查指令表可知执行 MOV 指令需用 1 个机器周期, DJNZ 指令需用 2 个机器周期, 在 12MHz 晶振时, 一个机器周期时间长度为 1 μs, 所以该段程序执行时间为: $[1 + (1+1+2*123+2) * 200+2] * 1 \mu s \approx 50ms$
--	---

(5) 参考程序框图

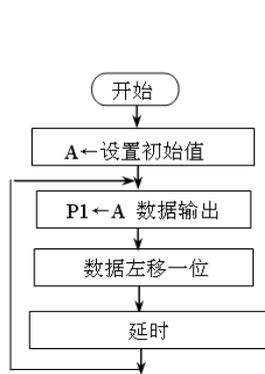


图 5-12 P1 循环点灯流程图

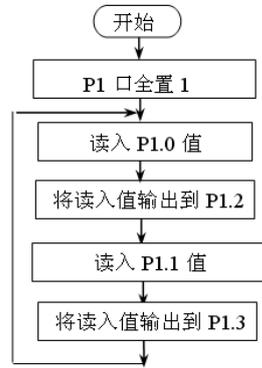


图 5-13 P1 口输入输出流程图

实验 1 参考程序

```

ORG 0
LOOP: MOV A,#01
      MOV P1,A
      RL A
      LCALL Delay
      SJMP LOOP
    
```

实验 2 参考程序:

```

ORG 0
LOOP: MOV P1,#0FFH
      MOV C,P1.0
      MOV P1.2,C
      MOV C,P1.1
      MOV P1.3,C
      SJMP LOOP
    
```



图 5-14 P1 端口输出情况



图 5-15 P1 端口输入/输出

(6) 程序调试。在 WAVE 集成软件调试环境下, 输入以上程序, 编译通过后, 打开外设端口, 单步运行, 观察 P1 口的输出情况, 实验 1 运行情况如图 5-14 所示。调试实验 2 程序时, 采用单步运行方式, 用鼠标点击 P1.0 和 P1.1 改变其输入状态, 观察 P1.2 和 P1.3 的输出情况, 实验 2 运行情况如图 5-15 所示。

实验 5-8 交通灯控制 (软件延时法)

- (1) 实验目的: 1. 学习数据输出程序的设计方法。学习模拟交通灯控制的方法。3. 进一步了解软件延时方法。
- (2) 实验内容: 用 CPU 的 P1 口输出控制信号, 控制六个 LED 灯(红, 绿, 黄), 模拟交通灯管理。
- (3) 实验电路连接:

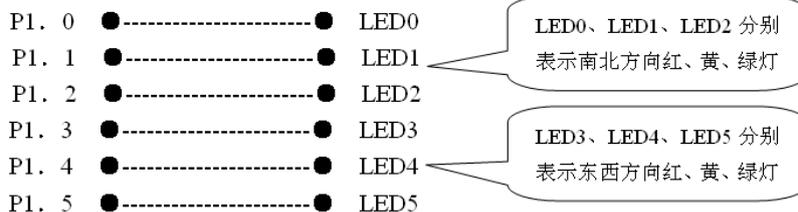


图 5-16

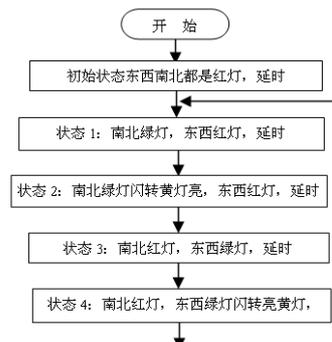


图 6-16 交通灯控制参考程序框图

(4) 实验说明。1. 因为本实验是交通灯控制实验, 所以要先了解实际交通灯的变化规律。假设一个十字路口为东西南北走向。

初始为状态 0。

**状态 0:** 东西红灯，南北红灯；然后转

**状态 1:** 南北绿灯通车，东西红灯；过一段时间转

**状态 2:** 南北绿灯闪几次转黄灯亮，延时几秒，东西仍然红灯；再转

**状态 3:** 东西绿灯通车，南北红灯；过一段时间转

**状态 4:** 东西绿灯闪几次转亮黄灯，延时几秒，南北仍然红灯；最后循环至**状态 1**。

2. 各用一组红、黄、绿色 LED 分别表示南北方向和东西方向红绿灯。
3. 延时程序参考硬件实验一中的例子，具体时间长短由同学自己确定。

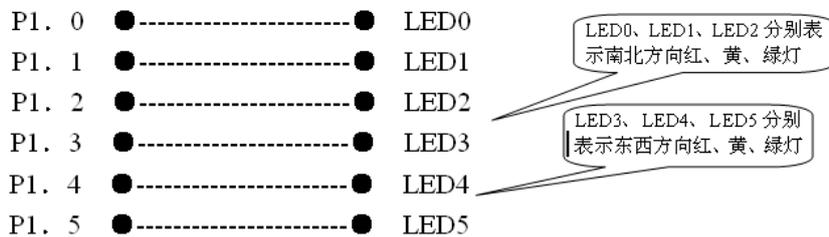
(5) 参考程序框图如图 5-16 所示。

### 实验 5-9 交通灯控制（定时器延时法）

- (1) 实验目的：1. 进一步理解单片机内部定时器/计数器的工作原理和使用方法。2. 学习模拟交通灯控制的方法。3. 学习数据输出程序的设计方法。4. 学习中断处理程序的编程方法。

(2) 实验内容：用 CPU 的 P1 口输出控制信号，控制六个 LED 灯(红，绿，黄)，模拟交通灯管理。

(3) 实验电路连接：



(4) 实验说明。

1. 因为本实验是交通灯控制实验，所以要先了解实际交通灯的变化规律。假设一个十字路口为东西南北走向。初始为状态 0。

**状态 0:** 东西红灯，南北红灯；然后转入

**状态 1:** 南北绿灯通车，东西红灯；过一段时间转入

**状态 2:** 南北绿灯闪几次转黄灯亮，延时几秒，东西仍然红；再转入

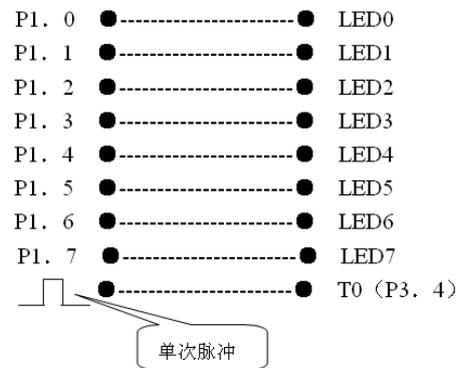
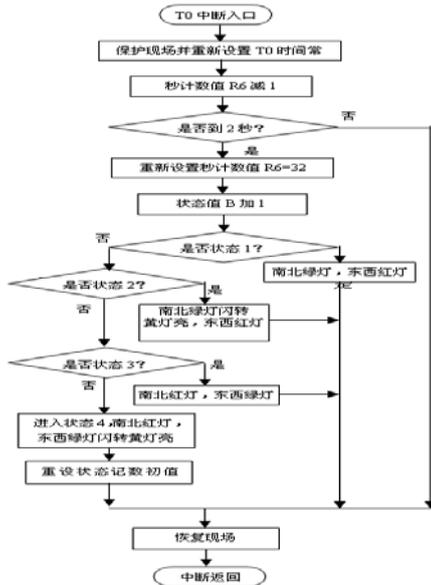
**状态 3:** 东西绿灯通车，南北红灯；过一段时间转入

**状态 4:** 东西绿灯闪几次转黄灯亮，延时几秒，南北仍然红灯;最后循环至**状态 1**。

2. 各用一组红、黄、绿色 LED 分别表示南北方向和东西方向。



图 3-17 主程序流程图



3. 由定时器来产生通车延时时间，时间长短 1 秒以上（由同学自己确定）。 图 5-18

**提示:**内部计数器用作定时器时,是对机器周期计数。每个机器周期的长度是 12 个振荡器周期。因为实验系统的晶振是 12MHZ,本程序定时器工作于方式 1（16 位方式）时，最大定时时间为： $2^{16} * 1 \mu s = 65536 \mu s = 65.536ms$

所以需要配合软件计数。如要延时 2 秒，则需要 T0 中断 32 次，所用时间为： $65536 * 32 = 2097120 \mu s \approx 2s$

因此在 T0 中断处理程序中，要判断中断次数是否到 32 次，若不到 32 次，则只使中断次数加 1，然后返回，若到了 32

次，定时 2 秒时间到。

4. 用软件延时方法产生“闪”延时时间（参考硬件实验 6-8）。

(5) 参考程序框图如图 5-17、5-18

### 实验 5-10 计数器应用实验

(1) 实验目的：学习 51 单片机内部定时/计数器使用方法。

(2) 实验内容：51 单片机内部定时计数器 T0，按计数器模式工作于方式 1，对 P3. 4(T0)引脚进行计数。将其数值按二进制数在 P1 口驱动 LED 灯上显示出来。

(3) 实验电路连线如右图所示。

(4) 实验说明。本实验中内部计数器起计数器的作用。外部事件计数脉冲由 T0 (P3. 4) 脚引入入定时器 T0。单片机在每个机器周期采样一次输入波形，因此单片机至少需要两个机器周期才能检测到一次跳变，这就要求被采样电平至少维持一个完整的机器周期，以保证电平在变化之前即被采样。同时这就决定了输入波形的频率不能超过机器周期频率。

(5) 参考程序框图如图 5-19 所示。

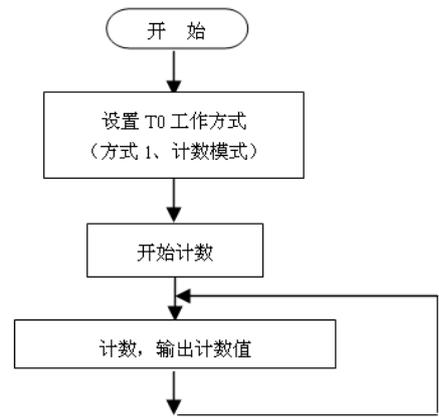


图 6-19 计数器实验参考程序框图

### 实验 5-11 外部中断实验

(1) 实验目的：1、学习外部中断技术的基本使用方法。2、学习中断处理程序的编程方法。

(2) 实验内容：用单次脉冲申请中断，在中断处理程序中对信号进行反转输出。

(3) 实验电路连线如右图。

(4) 实验说明。编写中断处理程序需要注意的问题是：1. 保护进入中断时的状态，并在退出中断之前恢复进入时的状态。

2. 必须在中断处理程序中设定是否允许中断重入，即设置 EX0 位。

本例中使用了 INT0 中断，一般进入中断处理程序时应保护 PSW, ACC 以及中断处理程序使用但非其专用的寄存器（保护现场）。本例的 INT0 中断处理程序保护了 PSW, ACC 等三个寄存器并且在退出前恢复了这三个寄存器（恢复现场）。另外中断处理程序中涉及到关键数据的设置时应关中断，即设置时不允许中断重入。本例中没有涉及这种情况。3. INT0 端接单次脉冲发生器。P1. 0 接 LED 灯，以查看信号反转。

(5) 参考程序框图如图 5-20,5-21



### 实验 5-12 定时器实验 1

(P1 口状态取反)

(1) 实验目的：1. 学习单片机内部计数器的使用和编程方法；2. 进一步掌握中断处理程序的编程方法。

(2) 实验内容：用单片机内部定时器中断方式计时，实现每 1 秒钟 P1 口输出状态发生一次反转。

(3) 实验说明。1. 关于内部定时器/计数器的编程主要是时间常数的设置和有关控制寄存器的设置。内部定时器/计数器在单片机中主要有定时和计数两种功能。本实验使用的是定时功能。

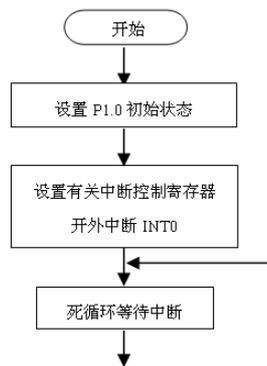


图 5-20 主程序框图

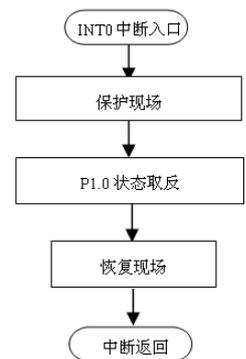


图 5-21 INT0 中断处理程序框图

2..与定时器有关的寄存器有工作方式寄存器 TMOD 和控制寄存器 TCON。TMOD 用于设置定时器 / 计数器的工作方式 0-3，并确定用于定时还是用于计数。TCON 主要功能是为定时器在溢出时设定标志位，并控制定时器的运行或停止等。

3.内部计数器用作定时器时，是对机器周期计数。每个机器周期的长度是 12 个振荡器周期。因为实验系统的晶振是 6MHz，本程序定时器工作于方式 1（16 位方式）时，最大定时时间为： $2^{16} * 2 \mu s = 131072 \mu s \approx 131ms$

设定时间为 100ms，可计算定时器初值为 X

$$100ms = (2^{16} - X) \times 2\mu s, \quad X = 2^{16} - \frac{100ms}{2\mu s} = 15536 = 3CB0H$$

再利用软件记数,当 T0 中断 10 次时，所用时间为 2S。因此在 T0 中断处理程序中，要判断中断次数是否到 10，若不到 10，则只使中断次数加 1，然后返回，若到了 10 次，则使输出状态位取反。

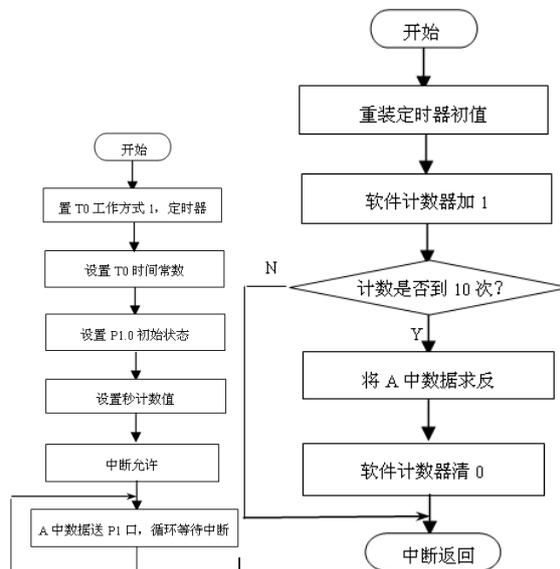
4.在设置时间常数前要先关对应的中断，设置完时间常数之后再打开相应的中断。

(4) 实验参考程序:

```

ORG 0000H
LJMP MAIN ;转主程序
ORG 000BH ;T0 中断入口
LJMP TIME0 ;转 T0 中断服务程序
ORG 0100H ;主程序
MAIN: MOV A,#0FFH ;
MOV R2,#0 ;计数器清 0
MOV TMOD,#01H ;设置 T0 工作方式 1
MOV TH0,#3CH;设置 T0 的初值
MOV TL0,#0B0H
SETB ET0 ;允许 T0 中断
SETB EA ;开放总中断允许
SETB TR0 ;启动 T0 工作
L0: MOV P1,A ;A 中数据送 P1 口
SJMP L0 ;主程序结束
TIME0: MOV TH0,#3CH;T0 中断服务程序
MOV TL0,#0B0H ;重装 T0 计数初值
INC R2 ;软件计数器加 1
CJNE R2,#10,L1 ;判断是否到 10 次计数?
CPL A ;到 10 次后, 将 A 中数据取反
MOV R2,#0 ;软件计数器清 0
L1: RETI ;中断返回
END

```



(5) 实验步骤: 1. 设计应用程序, 包括主程序和T0中断服务程序(见参考程序)。 2. 运行程序。点击程序运行键, 打开外设窗口, 观察端口输出情况。将光标停留在中断服务程序的中, 对软件计数器清0语句“MOV R2,#0”处, 再用“执行到光标处”功能, 可以看到每中断10次计数时间到以后, P1口的输出状态都会发生改变。

## 实验 5-13 定时器输出 PWM 实验

(1) 实验目的: 1. 了解脉宽调制(PWM)的原理; 2. 学习用 PWM 输出模拟量; 3. 熟悉 51 系列单片机的延时程序。

(2) 实验说明: PWM 是单片机上常用的模拟量输出方法, 通过外接的转换电路, 可以将脉冲的占空比变成电压。程序中通过调整占空比来调节输出模拟电压。占空比是制脉冲中高电平与低电平的宽度比。

(3) 实验内容及步骤。P1.0 输出 PWM 信号, 输出信号送数字电压表显示。本实验需要用到单片机最小应用系统(F1 区)。

1. 选用单片机最小应用系统模块, 用导线将 P1.0 电压输出接电压表“+”端, 电压表“-”端接地。

2. 打开 Keil uVision2 仿真软件, 首先建立本实验的项目文件, 输入源程序, 进行编译, 直到编译无误。

3. 全速运行程序, 观察电压表显示值, 并做记录, 改变 PWM 的占空比。修改源程序 LOOP 程序段两次, 给累加器 A 的赋值, 改为①“MOV A, #1” ②“MOV A, #9”, 重新编译后运行, 记录电压表显示值, 这是占空比 1:9 的 PWM。同样, 用户可做占空比 9:1 的 PWM, 并做记录。比较三种 PWM 信号转换电压的大小, 与理论值相比较。

4. 也可以把源程序编译成可执行文件, 把可执行文件用 ISP 烧录器烧录到 89S52/89S51 芯片中运行。

(4) 流程图及源程序。流程图和源程序如下:

```

ORG 0
LJMP START
ORG 000BH ;T0 中断入口
LJMP T0INT ;转中断服务程序
ORG 0030H ;主程序
START: MOV TMOD,#01H ;T0 方式 1
MOV TH0,#0ECH ;设 T0 时间常数

```

```

MOV    TLO, #78H    ;
SETB  TR0          ;启动 T0 工作
SETB  ET0          ;允许 T0 中断
SETB  EA           ;开放总中断
MOV    R2, #0       ;计数器清 0
SETB  P1.0         ;P1.0 输出 1
SJMP  $            ;循环

TOINT: MOV    TLO, #78H    ;T0 中断服务程序
      MOV    TH0, #0ECH    ;重装 T0 初值
      INC    R2            ;计数器加 1
      CJNE  R2, #5, NX1    ;判断 PWM 高电平持续时间 5%
NX1:   JC    NX2            ;小于设定值, 输出 P1.0=1
      CLR   P1.0          ;大于设定值, 输出 P1.0=0
      CJNE  R2, #100, NX3  ;判断? PWM 周期=100
      MOV   R2, #0         ;若 PWM 周期=100, 则计数器清 0
      SJMP  NX3           ;转中断返回

NX2:   SETB  P1.0         ; 输出 P1.0=1
NX3:   RETI              ; 中断返回
    
```

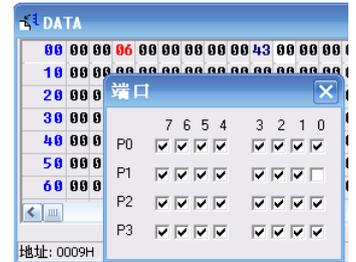
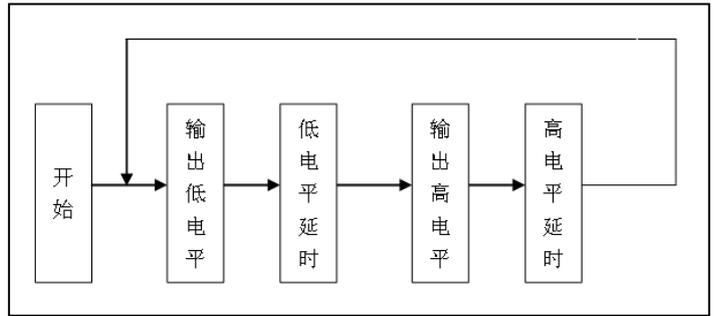


图 5-24 P1.0 输出状态

(5) 参考程序设计。设单片机晶振频率为 6MHz，定时器 0.01s 产生中断，定时器工作方式 1、初值为 X：

$$10ms = (2^{16} - X) \times 2\mu s, \quad X = 2^{16} - \frac{10ms}{2\mu s} = 60536 = EC78H$$

若要求 P1.0 输出 PWM 的占空比为 5% 的脉冲信号。

- (6) 思考题。1.分析 PWM 转换电路的原理。2.改变延时子程序 R0 的值，观察转换电压如何改变。  
 3. 如何采用 T0 定时，用查询方式使 P1.0 输出占空比为 1%~99% 的 PWM 信号？  
 4. 如何用中断方式，使 P1.0 输出占空比为 1%~99% 的 PWM 信号？如何改变占空比？

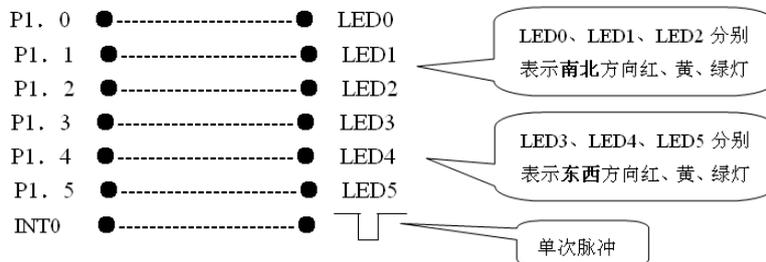
(7) 仿真调试。在 WAVE 软件环境下输入参考程序，打开外设输出端口和 DATA 窗口，将光标停留在中断返回语句，再用“执行到光标处”功能，可以看到每次中断以后，R2 计数器的计数情况和 P1.0 口的输出 PWM 电平的情况，如图 5-24 所示。

### 实验 5-14 外部中断实验

(1) 实验目的：1. 学习外部中断技术的基本使用方法。 2. 学习中断处理程序的编程方法。

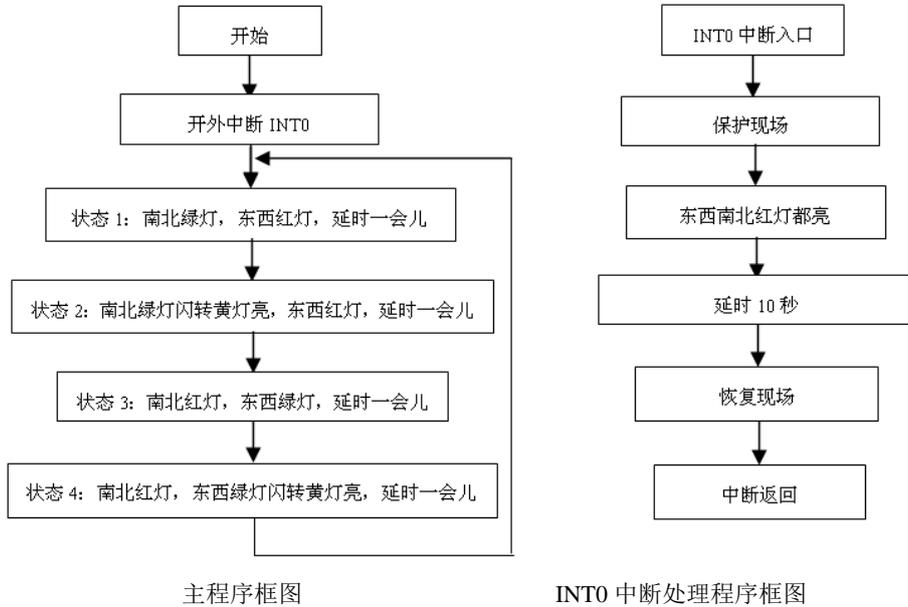
(2) 实验内容：在硬件实验二的基础上增加允许急救车优先通过的要求，有急救车到达时，两个方向交通信号灯全红，以便让急救车通过。设急救车通过路口时间为 10 秒，急救车通过后，交通恢复正常，本实验用单次脉冲申请中断，表示有急救车通过。

(3) 实验电路及连线如下图：



(4) 实验说明。1、编制中断处理程序要注意的问题是：保护进入中断时的状态（保护现场），并在退出中断之前恢复进入时的状态（恢复现场）。2、本例中使用了 INT0 中断，一般中断处理程序进入时应保护 PSW，ACC 以及中断处理程序使用但非其专用的寄存器。本例的 INT0 程序中应保护 PSW，ACC 等，以便保护中断前红绿灯的状态（保护现场），并且在退出前恢复原来红绿灯状态（恢复现场）。INT0 端接单次脉冲发生器。3、本实验中交通信号灯 LED0~LED5 与 P1.0~P1.5 连接。

(5) 参考程序框图。



## 5.2 单片机扩展和接口实验与设计

本节实验主要参考教材[2]是清华大学出版社的《单片机原理及实用技术》

### 实验 5-15 单片机串口扩展

- (1) 实验任务 1: 参考教材[3]的图 5-5(a), 利用单片机串行口扩展 24 个发光二极管和 8 个按键, 要求画出电路图并编写程序, 使 24 个发光二极管按照不同的顺序发光 (发光的时间间隔为 1S)。选择模块 B9 完成此实验。
- (2) 实验任务 2: 说明电路图 5-25 的工作原理, 并编写此电路的数码管控制与显示程序, 完成硬件设计实验。
- (3) 实验任务 3. 某单片机系统中用 P0 口驱动共阴极 LED 数码管的笔画段, 用 P2.0~P2.5 通过反相驱动电路驱动 0~5 位, 请画出电路图。并编写显示子程序, 将显示缓冲器 50H~55H 内的 6 个 10 进制数字显示一遍, 已知 0 的显示代码为 3FH, 且有延时子程序 Delay 可供调用。
- (4) 实验任务 4. 说明电路图 5-26 的工作原理, 并编写此电路的数码管控制显示程序。

### 实验 5-16 键盘与液晶显示控制

- (1) 实验任务 1: 设计一个 89S51 单片机与液晶 LCM1602 和 4X4 十六键键盘接口的电路 (图 5-25), 并编写程序, 实现 4 位十进制数相加减的功能, 由键盘输入数据和控制加减, 由 LCM1602 液晶屏显示数据输入和计算结果。
- (2) 实验任务 2: 设计一个 89S51 单片机与液晶 LCM1602、DS18B20、蜂鸣器、4X4 十六键键盘接口的电路 (图 5-26), 并编写程序, 实现温度测量和显示功能。而且由键盘输入温度上下限报警数据, 温度脱离上下限后即报警。

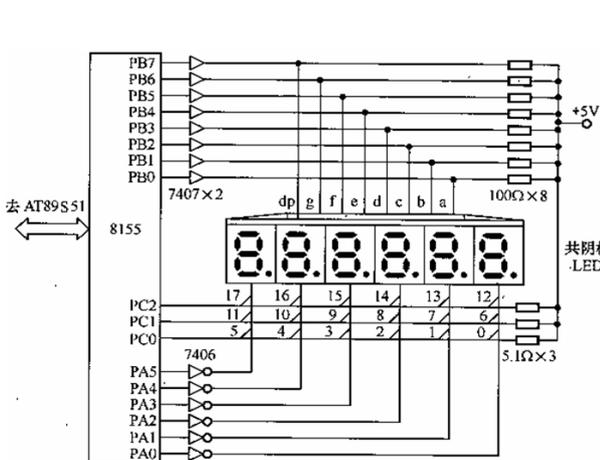


图 5-25

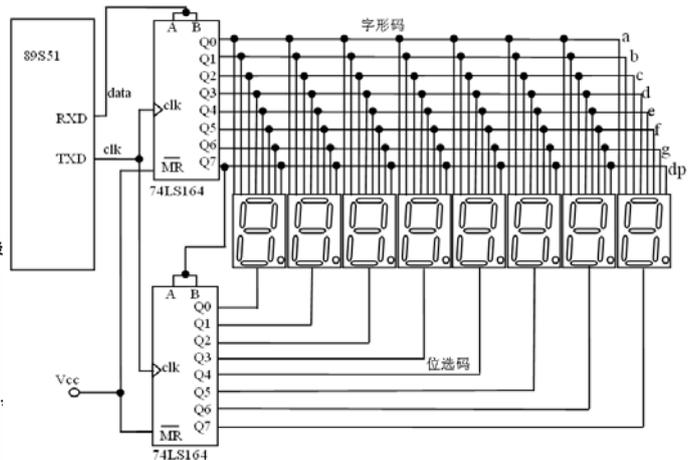


图 5-26

### 实验 5-17 单片机串行通信和红外双向通信

- (1) 实验目的. 1、掌握串行通信的基本原理; 2、掌握单片机串行通信技术和程序设计的综合知识; 3、掌握用单片机控制

红外线调制/解调模块实现双机通信的方法。

(2) 实验内容。1、用两台单片机实现双机通信(有线通信),通过 4×4 小键盘和 LCD1602 显示模块,输入和显示信息。2、实现两台单片机之间的无线通信(红外线通信),通过 4×4 小键盘和 LCD1602 显示模块,输入和显示信息。

实验中需要用到前面实验的相关内容有:键盘扫描和程序散转, LCD1602 的控制显示程序,定时器/计数器的控制,查询和中断的编程控制方法。

(3) 实验要求。采用全双工通信方式实现单片机之间双机通信,两台单片机都可同时发送和接收信息。当按动键盘上的不同按键时,在发送单片机和接收单片机的 LCD 显示屏上分别显示发送键的编号和从串行通信口接收到的信息。

(4) 实验基本原理。采用全双工异步通信方式,单片机之间的连接方法分别如图 5-27(有线通信)和图 5-28(红外线通信)所示。红外通信系统主要由红外发射装置、红外接收装置、及单片机等组成的数字系统(见图 5-28)。发射器件采用红外线发光二极管,如图 5-29(a)所示。接收装置采用一体化红外线接收器,其外形如图 5-29(b)所示,它是一种集红外线接收、带通滤波、选频放大、整形于一体的集成电路,不需要任何外接元件,就能完成从红外线接收到输出与 TTL 电平信号兼容的所有工作,没有收到红外信号时为高电平,收到红外信号时为低电平。红外线通信数据传输波形如图 4 所示。目前常用的红外发光二极管发出的红外线波长为 940nm,常用的载波频率为 37.9kHz。

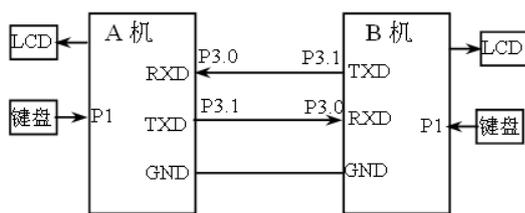


图 5-27 单片机双机通信(有线)

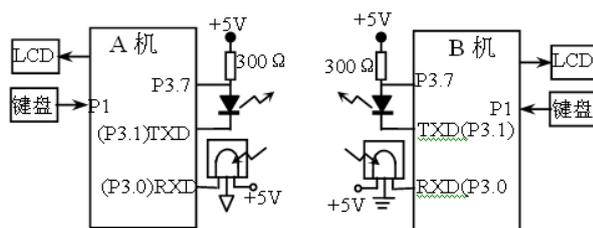


图 5-28 单片机双机通信(无线)

为了能够实现单片机之间的红外通信,要求发射端单片机产生频率为 37.9kHz 载波信号,并将串行通信的数字信号调制到载波信号上,通过红外发光二极管发射出去。接收端单片机用一体化红外线接收器将调制信号解调后还原成数字信号。从图 5-30 波形可以看出,调制波形与解调后的波形正好反相,即当发射数据为“1”或“空”时,输出为低电平,无调制波输出;而当发射数据为“0”时,有调制波输出。

载波信号的产生。利用单片机的定时器 T0 工作在方式 2,具有自动重装功能。通过对系统时钟信号分频,在 T0 的中断服务程序中对 P3.7 求反,即可在 P3.7 输出 37.9kHz 的载波信号。通信波特率发生器。异步通信时,要求通信双方须采用相同的波特率,利用单片机 T1 工作在方式 2,作为波特率发生器,产生串行通信收发所需的移位脉冲时钟信号。

信号调制的实现方法。红外发光二极管的正端接 P3.7 输出的载波信号,负端接 TXD(P3.0)串行通信数据输出口。当 TXD 输出逻辑“1”(高电平)时,红外发光二极管截止,无调制信号输出;而当 TXD 输出逻辑“0”(低电平)时,红外发光二极管导通,发射出 37.9kHz 的红外线调制信号,这样就可以通过红外发光二极管发射调制后的数字信号。单片机 I/O 口的输出电流非常有限,在载波信号输出端 P3.7 外接 300 欧姆的上拉电阻,来提高红外发光二极管的正向导通电流,可以增加红外发射功率。

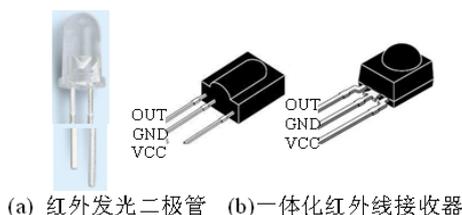


图 5-29 红外线发射器 / 接收器

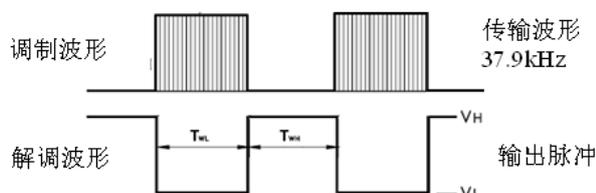


图 5-30 红外通信调制解调波形

(5) 实验步骤。1、学习串行通信相关知识;2、根据实验内容和实验要求,提出实验方案,画出程序流程图,编写实验程序。将实验程序 TX\_1.asm 分别下载到两台通信的单片机中;3、实验电路连接:

1. 连接单片机模块、4×4 小键盘模块和 LCD1602 显示模块。

2. 单片机之间(有线)全双工通信,如图 5-27 所示。用三根导线连接两台单片机的 TXD、RXD、GND。

3. 单片机之间(无线)全双工通信,如图 5-28 所示,在 P3 口接红外线收发模块。一体化红外线接收器的三个引脚分别是:

□OUT 信号输出;□GND 地;□Vcc 电源+5V。一体化红外线接收器与单片机的连接:

□OUT—RXD (P3.0); □GND 地; □Vcc 电源+5V;

红外线发射器与单片机的连接:

□红外发光二极管的正极接单片机的 37.9kHz 载波信号输出端(P3.7), (P3.7)与 Vcc 之间接 300 欧姆的上拉电阻;□红外发光二极管的负极接 TXD(P3.1)。

(6) 实验内容 1。双机通信(有线),单片机之间连接如图 5-27 所示,两台单片机可以相互通信。按动小键盘上的按键,可

以在本机的 LCD1602 的发送区显示键号。对方单片机接收键号后可以在接收区显示接收到的信息。

(7) 实验内容 2。双机通信（无线），单片机之间无导线连接，连线如图 5-28 所示，两台单片机之间通过红外线可以相互通信。按动小键盘上的按键，可以在本机的 LCD1602 的发送区显示键号。通过红外线无线传输，对方单片机接收信息后可以在接收区显示接收到的信息。

(8) 实验内容 3。分别采用串行通信的方式 1、方式 2 和方式 3 进行双机通信。

(9) 实验内容 4。发送和接收程序分别采用查询方式和中断方式实现。

(10) 实验内容 5。用一台单片机外接 DS18B20 测温，通过红外发射测温数据，在另一台单片机上显示温度数据。

(11) 实验报告要求。1. 分析程序，说出程序的功能，画出软件流程图；2. 源程序加注释。3. 画出硬件接口原理电路图。

4. 实验小结（包括理论知识的总结和实验结果的分析）

## 实验 5-18 单片机扩展 X5045 看门狗器件

**KX-CDS 实验系统演示示例(含实验指导 pdf/PDF 课件): DEMO\MCU\_TECH\_DEMO\ADC\_DAC\X5045**

(1) 实验原理：X5045 是 XICOR 公司推出的电压监控芯片，

其引脚封装如图 5-31 所示。引脚功能说明：1、CS/WDI：片选输入/看门狗复位输入；2、SO：串行输出；3、WP：写保护输入；4、Vss：地；

5、Vcc：电源；6、RESET：复位输出；7、SCK：同步时钟输入；8、SI：串行输入。X25045 的状态寄存器描述器件的当前状态，各位意义如表 5-1 所列。

表 5-1 状态寄存器

7	6	5	4	3	2	1	0
0	0	WD1	WD0	BL1	BL0	WEL	WIP

其中，WD1、WD0 是看门狗定时时间设置位；BL1、BL0 是存储单元写保护区设置位；WEL 是只读标志，1 表明写使能开关打开；WIP 也是只读标志，1 代表芯片内部正处于写周期。电复位时，各位都被清零。

X25045 芯片具有以下 4 种功能（实验模块 B23）：

1、上电复位控制。在对 X25045 通电时，RST 引脚输出有效的复位信号，并保持至少 200ms，使 CPU 有效复位。

2、电源电压监控。当检测到电源电压低于内部门槛电压 VTRIP 时，RESET 输出复位信号，直至电源电压高于 VTRIP 并保持至少 200ms，复位信号才被撤消。VTRIP 的出厂值根据芯片型号不同共有 5 个级别的电压范围。对于需要电源电压精确监控的应用，用户可以搭建编程电路，对芯片内 VTRIP 电压进行微调。

3、看门狗定时器。芯片内部状态寄存器的 WD1、WD0 是看门狗定时设置位，通过状态寄存器写指令 WRSR 修改这 2 个标志位，就能在 3 种定时间隔中进行选择或关闭定时器。对看门狗的复位由 CS 输入电平的下降沿完成。表 5-2 是 WD1、WD0 组合的含义。

4、串行 E2PROM。芯片内含 512 字节存储单元，10 万次可靠写，数据保持时间 100 年。XICOR 设计了 3 种保护方式防止误写。包括：WP 写保护引脚，当引脚被拉低时，内部存储单元状态寄存器都禁止写入；存储区域写保护模式，通过对状态寄存器的 BL1、BL0 位的设置，可以选择对不同的存储区域进行写保护；在进行任何写操作前都必须打开写使能开关，而且在上电初始化写操作完成时，写使能开关自动关闭。显然，在几方面的保护之下，产生误写的可能性极小，表 5-3 是 BL1、BL0 组合的含义。

对 X25045 的操作是通过 4 根口线 CS、SCK、SI 和 SO 进行同步串行通信来完成的。SCK 是外部输入的同步时钟信号。在对芯片定改指令或数据时，时钟前沿将 SI 引脚信号输入；在读数据时，时钟后沿将数据位输出到 SO 引脚上。数据的输入/输出都是高位。数据的输入/输出都是高位在先。芯片内部共有 6 条指令，如表 6-4 所列，表中：

1、WREN 和 WRDI 是写使能开关的开/关指令。它们都是单字节指令。

2、RDSR 和 WRSR 是状态寄存器的读/写指令。在从 SI 输入指令后，RDSR 的执行结果，即状态寄存器内容须从 SO 读出；而 WRSR 需要紧接着输入修改数据。

3、READ 和 WEITE 是存储单元的读/写指令。输入指令后（指令码第三代表存储单元地址的最高位），接着输入低八位地址，最后就可以连续读出或写入数据。其中，读指针和写指针的工作方式完全不同，读指针的全部 8 位用来计数，0FFH 溢出后变成 00H；写指针只用最低两位计数，XXXXXX11B 溢出后变成 XXXX XX00B，所以连续写的实际结果是在 4 个单元中反复写入。另外，由于 E2PROM 的写入时间长，所以在连续两条写指令之间应读取 WIP 状态，只有内部写周期结束时才可输入下一条写指令。

表 5-2 看门狗定时器设置位

表 5-3 存储单元写保护区设置位

表 5-4

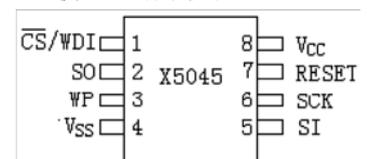


图 5-31 X5045 的引脚封装

命令名称	命令格式	内容
WREN	0000 0110	打开写使能开关
WRDI	0000 0100	关闭写使能开关
RDSR	0000 0101	读状态寄存器
WRSR	0000 0001	写状态寄存器
READ	0000 A <sub>8</sub> 011	读存储单元
WRITE	0000 A <sub>8</sub> 010	写存储单元

WD1	WD0	看门狗定时值
0	0	1.4s
0	1	600ms
1	0	200ms
1	1	禁止看门狗工作

BL1	BL0	写保护的单元地址
0	0	没有保护
0	1	180H~1FFH
1	0	100H~1FFH
1	1	000H~1FFH

(2)实验说明。**1. X5045 与单片机的连接：**1、SO—P2.5：串行数据输出引脚；2、SCK—P2.6：串行数据输入引脚；3、/CS—P2.7：片选 CS 及复位看门狗引脚；4、SI—P2.4：串行数据输入端；5、WP—P2.3：WP---写保护；

**2. LCD 与单片机连接**

LCD Module LMB1602 与单片机连接：

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Vss	Vdd	Vo	RS	R/W	E	D0	D1	D2	D3	D4	D5	D6	D7	BLA	BLK
0V+5V	0V	P32	P33	P34	-----	P1[0..7]	-----	+5V	0V						

**3. LCD显示功能说明。**LCD1602分为4个显示区(图5-32)：

- 显示 4×4 键盘按键时，按键的编号；
- 显示从 X5045 的 I2C 存储器中读出的 8 个字节的内容；
- 显示程序中对 X5045 的访问次数。若程序中能够在 WTC 溢出之前对 /CS 访问，则 WTC 不会产生 RESET 复位信号，反之则会产生 RESET 复位信号。□显示 WTC 的复位次数。

**4. 操作说明。**X5045 的 RST 接单片机的 P3.5。程序中用单片机的 T1 作为外部脉冲计数器，可记录 X5045 产生的复位信号。单片机 P1 口连接 4×4 键盘；用 8 芯排线连接单片机 P2 口与实验板 X5045 模块。通过串行口将程序 X5045\_1.hex 下载到 STC89C52。单片机程序运行时，通过 LCD1602 可以看到从 X5045 中读出的 8 字节内容和 WTC 复位次数的变化情况。

**5. 键盘功能说明。**1) 键 K04，允许程序在每次大循环中对 X5045 的 /CS 访问。按 K04 键后，可以使 WTC 计数器清 0，LCD 的 □ 区数字会连续变化，显示对 /CS 的访问次数，而 □ 区的数字停止不动。2) 键 K00，禁止程序在每次大循环中对 X5045 的 /CS 访问。按 K00 键后，LCD 的 □ 区数字会连续变化，显示 WTC 的复位次数；而 □ 区的数字停止不动。

3) 按 4×4 小键盘的其他键，在 LCD1602 的 □ 区显示按键的编号 (图 5-33)。

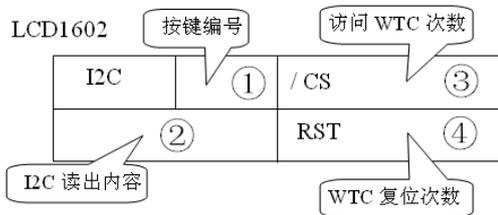


图 5-32 LCD1602 显示区

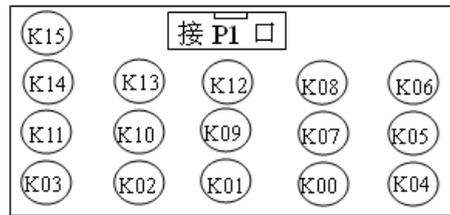


图 5-33 小键盘定义

**6. 单片机示例程序说明。**在单片机的主程序中对 X5045 进行了初始化，对芯片内部状态寄存器的 WD1、WD0 看门狗定时设置位进行设置，令 WD1、WD0=01，即每 600ms 产生一次溢出，使 RESET 引脚产生复位脉冲。因此，可以用单片机的 T1 对 RESET 溢出脉冲进行计数，显示在 LCD 的 □ 区。

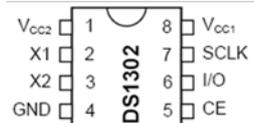


图 534 DS1302 引脚

**实验 5-19 单片机扩展 DS1302 时钟/日历器件**

**KX-CDS 实验系统演示示例 (含实验指导 pdf/PDF 课件)：**

**\\DEMO\MCU\_TECH\_DEMO\ADC\_DAC\DS1302**

(1) 实验原理。DS1302 是 DALLAS 公司推出的时钟芯片，内含有一个实时时钟/日历和 31 字节静态 RAM，通过简单的串行接口与单片机进行通信。实时时钟/日历电路提供秒、分、时、日、日期、月、年的信息，每月的天数和闰年的天数可自动调整，时钟操作可通过 AM/PM 指示决定采用 24 或 12 小时格式。DS1302 与单片机之间能采用同步串行的方式进行，DS1302 的引脚命名如图 6-34 所示。通信仅需用到三根信号线：1、CE 片选；2、I/O 数据线；3、SCLK 串行时钟。

DS1302与CPU的连接如图6-35所示。时钟/RAM的读/写数据以一个字节或多字节的字符组方式通信，DS1302工作时功耗很低，保持数据和时钟信息时功率小于1mW。DS1302具有双电源管脚，用于主电源和备份电源供应Vcc1，为可编程充电电源附加七个字节存储器，它广泛应用于电话传真、便携式仪器以及电池供电的仪器仪表等产品领域。

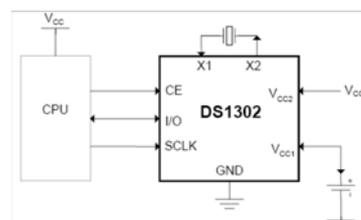


图 5-35 DS1302 与 CPU 接口

DS1302主要的性能指标：1、实时时钟具有能计算2100 年之前的秒、分、时、日、日期、星期、月、年的能力，还有闰年调整的能力；2、31×8 位暂存数据存储RAM；3、串行I/O 口方式，使得管脚数量最少；4、宽范围工作电压2.0~ 5.5V；5、工作电流2.0V时,小于300nA；6、读/写时钟或RAM数据时有两种传送方式：单字节传送和多字节传送字符组方式；7、8 脚DIP 封装或可选的8 脚SOIC 封装根据表面装配；8、简单3线接口；9、与TTL 兼容Vcc=5V。DS1302的读写模式工作时序如图5-36和图5-37所示。

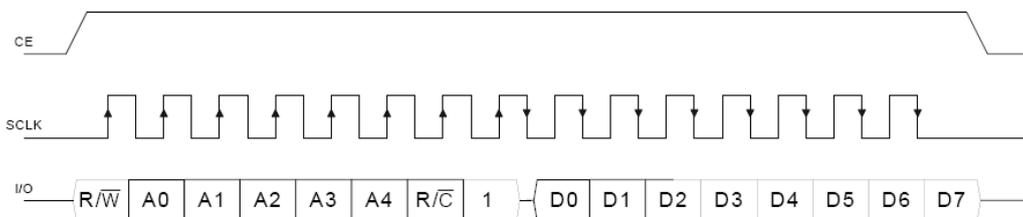


图5-36 单字节读模式

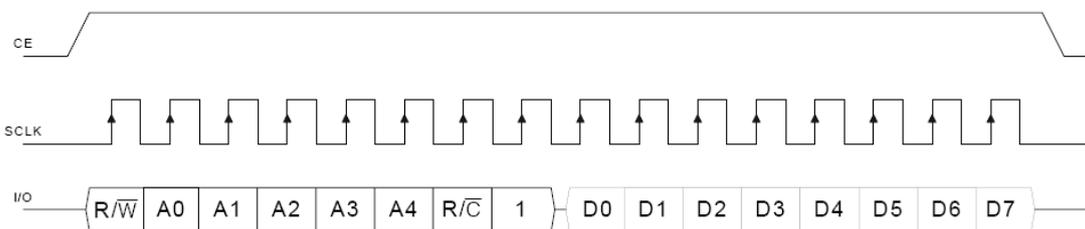


图5-37 单字节写模式

DS1302 内部寄存器的地址定义如表 6-5 所示。

(2) 实验说明。1. DS1302与51单片机的连接：IO—P2.7：串行数据输入/输出引脚；SCLK—P2.6：串行时钟引脚；CE—P2.4：片选CE

2. LCD 与单片机连接：LCD1602 显示格式如图 5-38 所示。LCD Module LMB1602 与单片机连接：

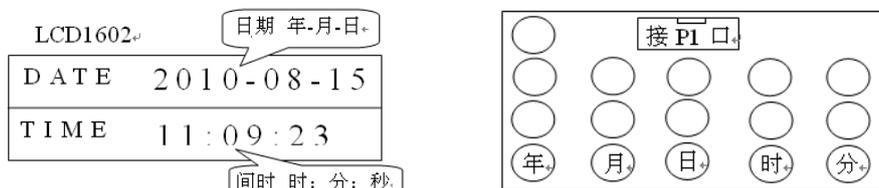
```

*****
;1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16
;Vss Vdd Vo RS R/W E D0 D1 D2 D3 D4 D5 D6 D7 BLA BLK
;0V +5V 0V P32 P33 P34 ----- P1[0..7] ----- +5V 0V
*****
    
```

表 5-5 寄存器的地址及定义

READ	WRITE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	RANGE	
81h	80h	CH	10 Seconds			Seconds				00-59	
83h	82h		10 Minutes			Minutes				00-59	
85h	84h	12/24	0	10 AM/PM	Hour	Hour				1-12/0-23	
87h	86h	0	0	10 Date		Date				1-31	
89h	88h	0	0	0	10 Month	Month				1-12	
8Bh	8Ah	0	0	0	0	0	Day				1-7
8Dh	8Ch	10 Year			Year					00-99	
8Fh	8Eh	WP	0	0	0	0	0	0	0	—	
91h	90h	TCS	TCS	TCS	TCS	DS	DS	RS	RS	—	

(3) 实验步骤。1、单片机模块的P2口与DS1302实验模块的接口用10芯排线连接；2、单片机模块的P1口与4×4小键盘连接；3、通过串行口将程序DS1302\_1.hex下载到STC89C52；4、可以通过小键盘调整日期和时间，键盘定义如图6-39。



### 实验 5-20 SPI 串行 DAC TLV5637 与单片机的接口

**KX-CDS 实验系统演示示例(含实验指导 pdf/PDF 课件): \DEMO\MCU\_TECH\_DEMO\ADC\_DAC\TLV5637**

(1) 实验原理。TLV5637 是 TI 公司生产的一款 10 位电压输出型 DAC，串行输入接口，双通道 DAC 输出，转换时间是 1μs，含可编程内部精密参考电压基准。TLV5637 的引脚命名如图 5-40 所示。该器件的 3 个串行信号线：SCLK、DIN 和  $\overline{CS}$  构成 SPI 接口通信线，故其支持对 TI 公司的 TMS320 的 DSP 系列的 SPI、QSPI 和 Microwire 串行标准接口；图 5-40 中 OUTA 和 OUTB 分别是两路模拟信号输出口，REF 是参考电平输入口。

图 5-41 是 TLV5637 和 SPI 接口的连接示意图。从图中可以看出，主控器件通过 SPI (Serial Peripheral Interface) 接口向 TLV5637 进行写操作。图 5-42 是 TLV5637 的 SPI 接口时序图。如图所示，TLV5637 的数据采样是在 SCLK 时钟的上升沿发生，而且该器件只支持 16 位的数据格式。详细内容参考教材[3]。

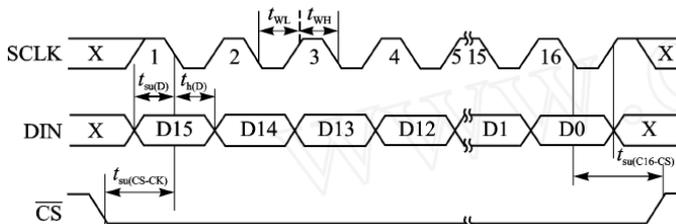
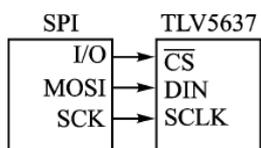
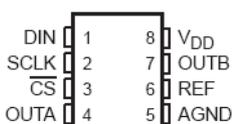


图 5-40 TLV5637 引脚图

图 5-41 TLV5637 与 SPI 口的连接

图 5-42 TLV5637 SPI 时序图

(2) 实验要求 1。参考教材[3]，试完成单片机对 TLV5637 的控制实验。设 TLV5637 与 AT89S5 的 P2 口连接，画出接口电路。要求使 TLV5637 分别从 A 通道和 B 通道输出锯齿波和三角波，试编写相应的汇编程序。并由双踪示波器显示此两路波形。

(3) 实验要求 2。参阅参考文献[1]、[2]和本书第 8 章，用 FPGA 控制 TLV5637，试于 KX-CDS 实验平台上完成以上实验要求，并将单片机与 FPGA 分别控制 TLV5637 输出波形（设每周期相同的波形数据点数）的最高频率进行比较（假设同时还要对某 ADC 进行采样控制），对比较结果进行评论。演示程序对应 TLV5637 模块。

### 实验 5-21 串行精密 ADC 器件 ADS1100 与单片机的接口

**KX-CDS 实验系统演示示例(含实验指导 pdf/PDF 课件): \DEMO\MCU\_TECH\_DEMO\ADC\_DAC\ADS1100**

(1) 实验原理。ADS1100是具有自校准功能的16位精密A/D转换器，使用可兼容的I<sup>2</sup>C串行接口，在2.7V至5.5V的单电源下工作。以电源作为基准电压，转换按比例进行。有单次和连续两类数据转换模式，而且体积小，功耗低，是嵌入式仪器仪表高分辨率采样测量电路的理想A/D转换芯片。ADS1100的封装为小型SOT23-6，引脚与结构如图6-43所示。ADS1100可每秒采样8、16、32或128次进行转换；片内可编程的增益放大器PGA提供高达8倍的增益，允许对较微弱的信号进行测量，并且具有高分辨率。在单周期转换方式中，ADS1100在一次转换之后自动掉电，在空闲期间极大地减少了电流消耗。

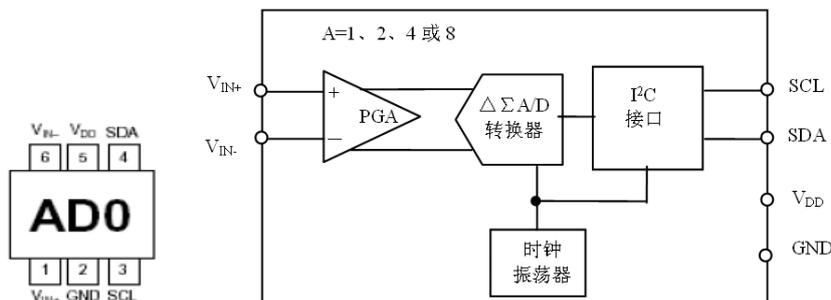


图 5-43 ADS1100 的引脚与结构

ADS1100的读操作时序如图5-44所示。首先在SCL为高电平时，SDA出现一个下降沿启动I<sup>2</sup>C总线,然后发送的第一个字节就是 ADS1100 的地址，其中第8位为读写方向位(“1”表示接受ADS1100 的数据，“0”表示发送数据给ADS1100)。ADS1100接收到字节信号后要返回一个应答信号，建立主从握手成功后，若是进行读操作，ADS1100依次发送输出寄存器高位字节、低位字节和方式寄存器字节数据；单片机每收到一个字节后都要返回一个应答信号，也可只读一个字节，若多于三个字节，后面收到的将是FFH。

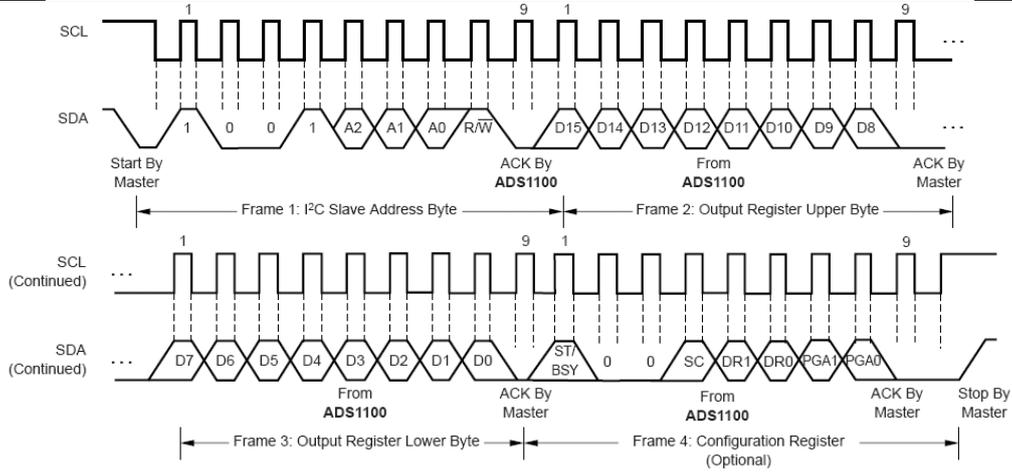


图5-44 ADS1100的读操作时序

(2) 实验要求。参考教材[3]，将 ADS1100 的 A/D 转换结果折算为所测的电压值，并送至 1602 液晶显示器显示。试设计 A/D 转换器与单片机接口的电路，编写相应的汇编程序，并完成全部实验。

### 实验 5-22 串行高速 ADC 器件 ADS7816 与单片机的接口

**KX-CDS 实验系统演示示例(含实验指导 pdf/PDF 课件): \DEMO\MCU\_TECH\_DEMO\ADC\_DAC\ADS7816**

(1) 实验原理。ADS7816 是一种 12 位，200kHz 转换速率的 A/D 转换器。它具有自动关闭电源时的低功耗工作方式，具有同步串行接口，模拟信号采用差分输入，以及可编程的参考电压和分辨率。其参考电压可改变范围是 100mV 至 5V；可改变的分辨率范围是 24uV 至 1.22mV。此外，还有低功耗，自动断电，体积小等特点，因此是使用电池供电系统的理想选择，也适用于要求大量信号同时进行采集的应用系统，以及远程和/或隔离的数据采集理想选择。ADS7816 提供三种封装形式：8pin-PDIP，8pin-SOIC，或 8pin-MSOP 封装。ADS7816 的技术指标和主要特点：

- 1、串行接口，200kHz 的采样率；差分输入；
- 2、低功耗：200kHz 和 12.5kHz 时分别是 1.9mW 和 150mW；掉电时最大电流 3mA；
- 3、8-pin 微型 DIP，SOIC 和 MSOP 封装

ADS7816 的引脚与结构如图 5-45 所示。ADS7816 的工作时序如图 6-46 所示，在此时序图中，串行时钟 DCLOCK 用于同步数据转换，每位转换后的数据在 DCLOCK 的下降沿开始传送。因此，从 Dout 引脚接收数据时，可在 DCLOCK 的下降沿期间进行，也可以在 DCLOCK 的上升沿期间进行。

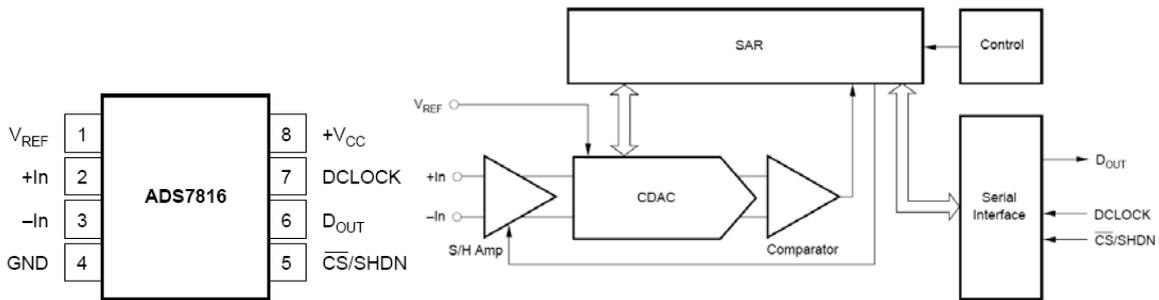


图 5-45 ADS7816 的引脚和结构

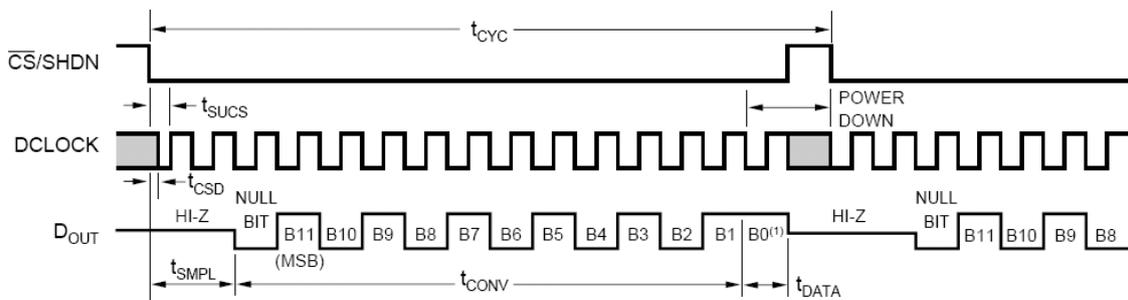


图 5-46 ADS7816 的工作时序

(2) 实验要求 1。参考教材[3]，将 ADS7816 的 A/D 转换结果送至 1602 液晶显示器显示。试设计 A/D 转换器与单片机接口的电路，编写相应的汇编程序，并完成全部实验。演示程序对应的 ADS7816 的 ADC 实验模块是 C6。

(3) 实验要求 2。参阅参考文献[1]、[2]和本书第 8 章，用基于 FPGA 的状态机控制 ADS7816，试于附录 2 的实验平台上完成以上实验要求，并将单片机与状态机分别控制 ADS7816 的最高采样速率进行比较（假设同时还要对某 DAC 进行波形输出控制）。

制)，对比较结果进行评论。

## 实验 5-23 高速低功耗串行 ADC 器件 TLV2541 与单片机的接口

**KX-CDS 实验系统演示示例(含实验指导 pdf/PDF 课件): \DEMO\MCU\_TECH\_DEMO\ADC\_DAC\TLV2541**

(1) 实验原理。TLV2541/2/5 是高性能的 12 位低功耗 CMOS 系列模/数转换器。TLV254x 系列产品在单端 2.7 V 至 5.5 V 的电源电压下工作，器件带有单通道双通道或单通道伪差分 (single pseudo differential) 输入，每个器件均有一个片选  $\overline{CS}$  端，一个串行时钟 SCLK 和一个串行数据输出 SDO。这为最常用的微处理器提供了一个三线接口，SPI 接口。若与 DSP 连接，会以一个帧同步信号 FS 来标示所有器件的  $\overline{CS}$  端，或者 TLV2541 的 FS 端上的串行数据帧的开始。TLV2541/2/5 的设计允许它们在极低的功耗下工作，而且以自动掉电的方式使其低功耗特点得到进一步增强，该系列产品可以为带有 SCLK 的微处理器提供一个高速串行链接，其速率可高达 20 MHz。TLV254X 系列产品使用内置的振荡器作为转换时钟，可提供 3.5  $\mu$ s 的转换时间。TLV2541 的引脚图和结构图如图 5-47 所示。

TLV254x 的技术指标和主要特点：1、小型 8 引脚 MSOP 和 SOIC 封装，内置转换时钟；2、SFDR: 85 dB  $f_i = 20$  kHz；INL/DNL: 最大  $\pm 1$  LSB SINAD 72 dB；3、可兼容 SPI/DSP 的串行接口，带自动掉电的低功耗；4、单端电源电压 2.7 V 直流至 5.5 V 直流，带 500 kHz 带宽的轨对轨的模拟输入；5、工作电流：2.7 V 电压时 1 mA，5V 电压时 1.5 mA；6、自动掉电：2.7 V 电压时 2  $\mu$ A，5V 电压时 5  $\mu$ A。

(2) 实验要求 1。参考教材[3]，将 TLV2541 的 A/D 转换结果送至 1602 液晶显示器显示。试设计 A/D 转换器与单片机接口的电路，编写相应的汇编程序，并完成全部实验。

(3) 实验要求 2。参阅参考文献[1]、[2]和本书第 8 章，用基于 FPGA 的状态机控制 TLV2541，试于 KX-CDS 实验平台上

完成以上实验要求，并将单片机与状态机分别控制 TLV2541 的最高采样速率进行比较（假设同时还要对某 DAC 进行波形输出控制），对比较结果进行评论。

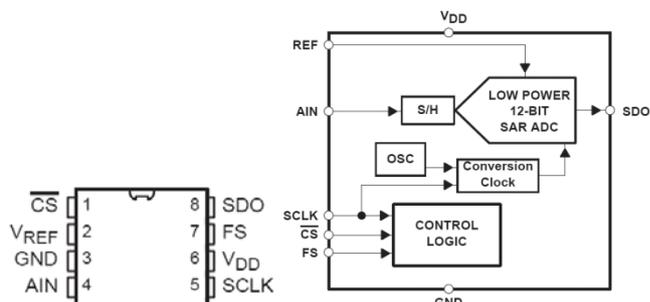


图 6-47 TLV2541 引脚图和结构图

## 实验 5-24 双通道 A/D 转换芯片 ADC0832 与单片机的接口

**KX-CDS 实验系统演示示例(含实验指导 pdf/PDF 课件): \DEMO\MCU\_TECH\_DEMO\ADC\_DAC\ADC0832**

将 ADC0832 的 A/D 转换结果送至 1602 液晶显示器显示。试设计 A/D 转换器与单片机接口的电路，编写相应的汇编程序，并完成全部实验。

## 实验 5-25 高速同步 10 位串行 A/D 转换器 TLV1572 与单片机的接口

**KX-CDS 实验系统演示示例(含实验指导 pdf/PDF 课件): \DEMO\MCU\_TECH\_DEMO\ADC\_DAC\TLV1572**

(1) 实验原理。TLV1572 是高速同步的 10 位 A/D 转换芯片，单电源 2.7~5.5V 供电，8 引脚 SOIC 封装。低功耗（3V 供电时功耗为 8mW，5V 供电时 25mW），当 A/D 转换不进行期间，自动进入省电模式。当采用 5V 供电，时钟频率 20MHz 时，最高转换速率为 625kSPS。

TLV1572 的引脚排列如图 5-48 所示。1、 $\overline{CS}$ ：片选使能，低电平芯片工作使能；2、VREF：基准电压输入。最小值 2.7V，最大值 VCC；3、GND：模拟地，芯片参考 0 电位（地）；4、AIN：模拟输入。最小值 GND，最大值 VREF；5、SCLK：串行时钟输入；6、VCC：电源。最小值 2.7V，最大值 5.5V；7、FS：DSP 方式时帧同步信号输入，在帧同步信号的下降沿，A/D 转换数据从；8、DO 引脚串行输出；SPI 方式时，和 VCC 连接在一起为高电平；9、DO：A/D 转换串行数据输出。

TLV1572 与微处理器的接口的 SPI 时序如图 6-49 所示。TLV1572 在片选信号  $\overline{CS}$  的下降沿通过检测帧同步引脚 FS 的电平状态，来辨别系统是工作在微控制器  $\mu$ C 模式还是 DSP 模式。若 FS 引脚是低电平，则系统工作于 DSP 模式；否则系统工作于  $\mu$ C 模式。当 TLV1572 工作于  $\mu$ C 模式时，FS 接高电平，在  $\overline{CS}$  的下降沿，ADC 开始传输数据到微处理器。输出数据采用 16 位的数据格式，所以传输数据时，在 ADC 的 10 位转换结果前要补 6 个 0；在  $\overline{CS}$  变为低电平后，从 SCLK 的第一个下降沿开始传输 0 值；在第 6 个 SCLK 的上升沿，6 个 0 位全部送出。

此后，在 SCLK 的每个上升沿，ADC 真正的 10 位转换结果按次逐位传送出去。而微处理器在每个 SCLK 的下降沿接收数据。图 5-49 中，数据输出 DO 线在第 16 个 SCLK 的上升沿时变为高阻态，在下一个  $\overline{CS}$  的下降沿时从高阻态中恢复回来。而系统会在第 17 个 SCLK 的下降沿时进入自动掉电模式。在下一个  $\overline{CS}$  的下降沿时从掉电模式中恢复出来，开始下一轮的转换和数据传输。

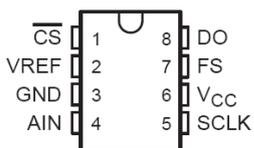


图 5-48 TLV1572 引脚图

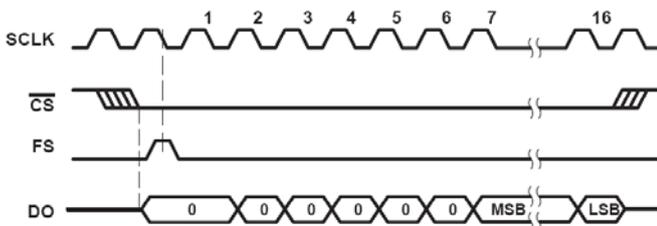


图 5-49 TLV1572 与微控制器接口的  $\mu C$  模式时序图

(2) 实验要求 1. 参考教材[3], 将 TLV1572 的 A/D 转换结果送至 1602 液晶显示器显示。试设计 A/D 转换器与单片机接口的电路, 编写相应的汇编程序, 并完成全部实验。(3) 实验要求 2. 参阅参考文献[1]、[2]和本书第 8 章, 用基于 FPGA 的状态机控制 TLV1572, 试于 KX\_DN 的实验平台上完成以上实验要求, 并将单片机与状态机分别控制 TLV1572 的最高采样速率进行比较(假设同时还要对某 DAC 进行波形输出控制), 对比较结果进行评论。

## 第六章 基于单片机 IP 核的 FPGA 片上系统 SOC 设计

可参考清华大学出版社出版、潘松等编著的《单片机原理与应用技术》

单片机系统设计的另一种途径是利用单片机 IP 软核 (Intellectual Property Soft Core), 将所有软硬件控制模块都放在单片 FPGA 中, 实现所谓 SOC (System On a Chip) 的单片系统设计, 这里称之为基于单片机 IP 软核的 SOC 设计方案, 或简称为单片 FPGA 系统方案。这是一个更具优势和发展前景的设计方案 (如图 6-1 所示)。如果将图 5-1 中虚线框中的所有内容都集成于一片大规模 FPGA 中, 就能构成一个单片系统, 或可称片上系统 (SOC)。图 5-1 就是一个较简单的片上系统, 这是单片机系统构建的另一个设计方案, 即 SOC 方案。基于这个设计方案的 FPGA 中将包含了一个单片机软核、多个不同类型和用途的存储器以及存储器控制模块、一个能提供不同功能操作和通信接口的宏功能模块、硬件算法模块等, 以及数个能提供不同时钟源的锁相环。其开发特点是首先设计和构建硬件环境, 包括 CPU 硬件工作平台, 然后对其进行硬件测试与仿真, 最后是针对单片机 CPU 核的工作完成软件设计与调试。片上系统的特点就是在一个单片集成电路模块中 (这里主要指 FPGA) 包含一个或多个处理器软核或硬核, 所有必须的存储器, 以及各种功能模块、控制模块、通信模块和接口模块。片上系统是一个软硬件有机结合的综合系统模块。

本章将给出多则根据图 5-1 基于 FPGA 的片上系统设计示例, 所有示例都是将上一章的基于图 6-1 电路模型中的所有模块, 包括单片机和存储器等都并入一片 FPGA 中。如前所述, 此结构可以归类为一个简单的可在 FPGA 中重构的片上系统 SOC。片上系统的设计和应用无疑是现代电子设计技术发展的方向, 读者可以通过本章的学习和实践, 初步了解和掌握基于 FPGA 的单片系统的软硬件基本构建和应用方法, 以及调试和测试技术。从本质上说, 本章设计项目的基本功能和实现方法与上一章的内容相同, 只是在实现技术上有所区别。

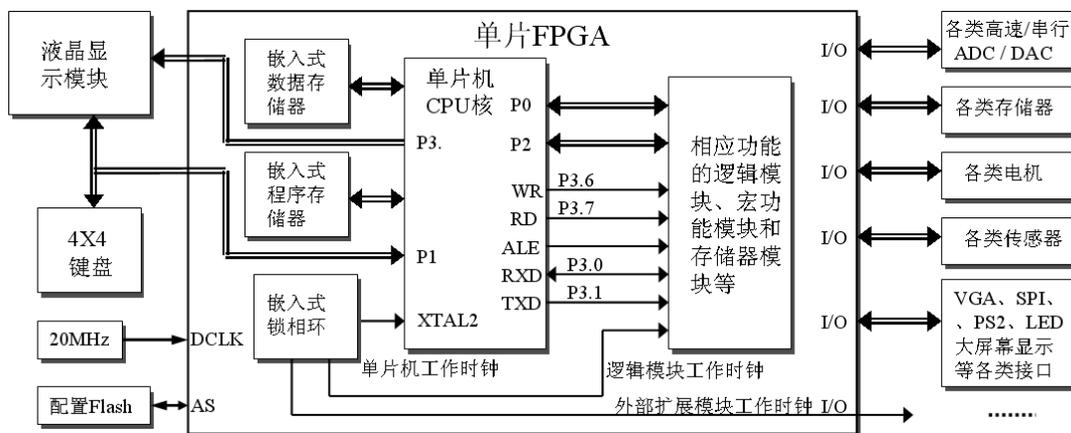


图 6-1 基于单片机软核的 FPGA 单片系统模块图

### 实验 6-1. FPGA 片上系统 8051 核测试电路

实验路径: \DEMO\8051Core\_DEMO\DEMO1\_8051Core\_BASIC\_V13

**(1) 实验目的:**

掌握单片机扩展 FPGA 软硬件综合设计基本方法。

**(2) 实验原理:**

设计一个软件语言测试 P1O 口的输出端，当 P3.0 为低电平时，允许软件计数并输出显示与数码管，当接收到 P3.2 口高电平，系统接收中断信号数码管显示“55”。

程序如下：(DEM01\_8051Core\_BASIC\_V13\ASM\TEST.ASM) 把生成的 HEX 文件装载到 ROM 中去，

```

ORG 0000H
    LJMP START
    ORG 0003H
    AJMP INTN
    ORG 0070H
START:  MOV SP,#60H
        MOV 40H,#22H
        MOV P1,#11H
        SETB IT0
        SETB EA
        SETB EX0
RRD:   PUSH 40H
        NOP
        POP 40H
        JB P3.0,RRD
TOINC: INC A
        LCALL DELAY
        MOV P1,A
        MOV 30H,A
        LJMP TOINC
INTN:  MOV P1,#55H
        INC 40H
        RETI
DELAY: MOV R7,#0FFH
WWW:   MOV R0,#0FFH
NMN:   DJNZ R0,NMN
        DJNZ R7,WWW
        RET
        END

```

**(3) 引脚锁定:**

选择模式 5

CLK 锁定核心板 50M 时钟，可以根据资料提供的引脚查找。

RST 单片机核复位选择键 1 锁定 PI00。

K2\_P30 计数允许选择键 2 锁定 PI01。

K3\_P32 中断允许选择键 2 锁定 PI02。

L[3..0] 计数低位显示选择数码 1 锁定 PI019~PI016。

H[3..0] 计数低位显示选择数码 2 锁定 PI023~PI020。

另每例必锁定引脚 NO\MT\POE[0]\POE[2]。

以上锁定的引脚都要根据表 2.2 查表对应型号的引脚号。

**(4) 下载测试:**

编程下载，主系统复位。选择模式 5，当键 1 复位键置高电平，键 2 置高电平，数码 1、2 开始计数，这时，如果键 2 置低，键 3 置高，键 1 置高接着再置高，数码显示“11”，再键 3 置高，数码显示“55”。

### (5) 实验任务:

根据提供的程序，修改程序，用不同的方式显示

## 实验 6-2. 串进并出/并进串出 8051 核测试电路

实验路径: \DEMO\8051Core\_DEMO\DEMO02\_8051Core\_S2P74164

### (1) 实验目的:

掌握单片机扩展 FPGA 串行扩展功能片上系统设计技术，熟悉软硬件综合设计基本方法。

### (2) 实验原理:

参考教材[3]8.3.1 节以及相关参考资料。

### (3) 实验任务:

根据图 6-2 的电路以及参考教材[3]8.1.2 节的设计流程，完成 FPGA 单片系统综合设计与测试。

对于本项设计任务，以及以下多数实验中同类设计任务，不再需要传统的单片机了，只要一个单片 FPGA 即可，因为可在其中调用 8052 CPU 核进行系统设计，因此可以使用 5KX-CDS 系统。当然也可使用同类的 FPGA 开发系统，只要其中的资源满足设计的要求就可以了。但推荐使用 CycloneIII~V 型 FPGA，因为其硬件资源更符合本章介绍的示例。单片机文件: /ASM/L1602.asm, 生成的 HEX 文件装载到 ROM 中去，或用在系统存储器编辑的方式实时下载到 ROM 中，进行调试。

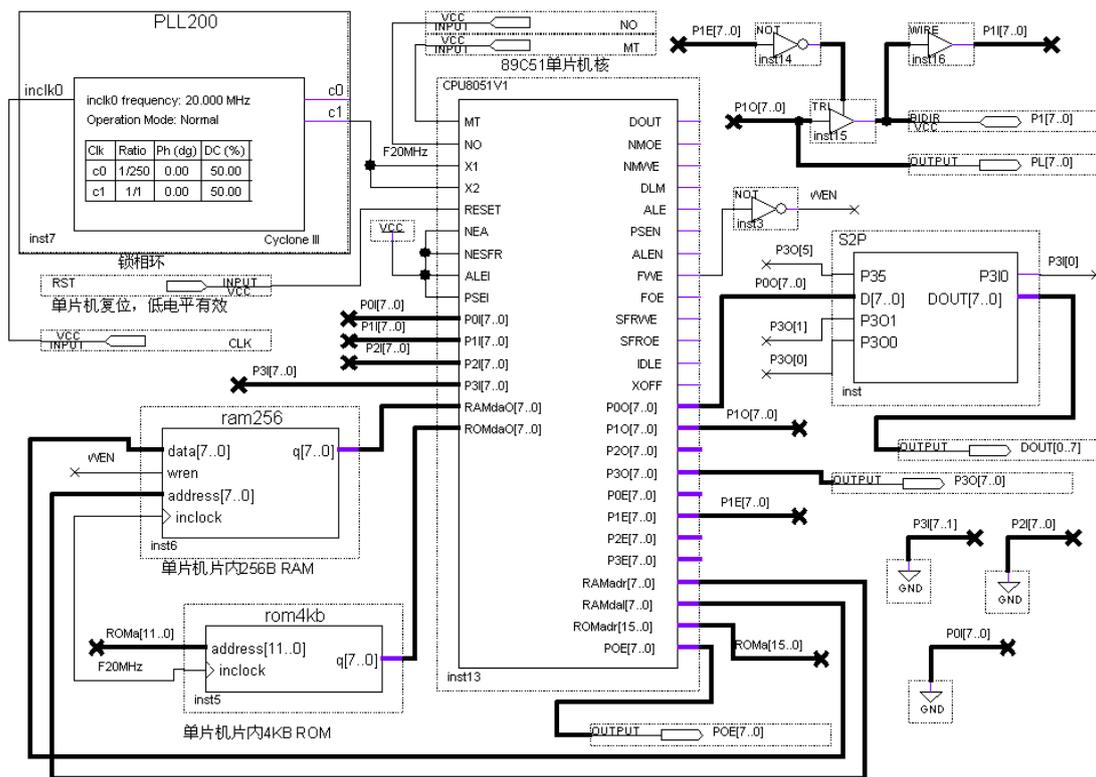


图 6-2 单片机扩展串进并出/并进串出模块的 FPGA 片上系统电路图

### (3) 硬件准备:

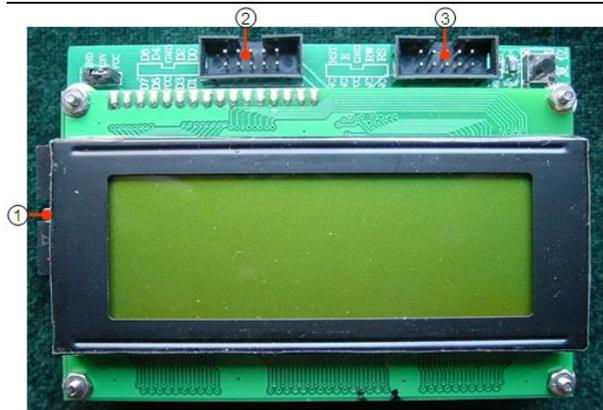


图 6-3 2004 字符液晶

图 6-3 字符液晶模块 2004 液晶其使用手册在文件夹“VIVI\_FILE\LCD\_FILE”里。

标注“1”做为 DDS 模块显示时，此接口是通过 14 芯线连接到 DDS 模块的标注“4”的接口上，在把标注“2”端口利用 10 芯线连接到 4X4 键盘的标注“1”接口上。这样就构成了 DDS 的硬件操作系统。

标注“2”是此液晶的 8 位数据控制端口。

标注“3”是此液晶的功能控制端口。原理图请参考文件夹“原理图\LCD4X20\_SCH.pdf”。

#### (4) 引脚锁定:

选择模式 5

CLK 锁定核心板 50M 时钟，可以根据资料提供的引脚查找。

RST 单片机核复位选择键 1 锁定 PI00。

P10 计数允许选择键 2 锁定 PI01。

P[2..0] 串入串出允许分别选择键 4、3、2 锁定 PI03~PI01。

PI03[4..2] 分别是字符液晶的 E\RW\RS 控制端口分别锁定主系统 JP6 的 DB6\DB3\DB0。

PL[7..0] 分别是字符液晶的 D7~D0 数据控制端口分别锁定主系统 J7 的 DB10\DB11\DB12\DB13\DBT0\DB14\DBT1\DB15。

另每例必锁定引脚 NO\MT\POE[0]\POE[2]。

以上锁定的引脚都要根据表 2.2 和 2.4 查表对应型号的引脚号。

#### (5) 硬件连接

准备 2 根十芯线，1 根连接模块右方十芯口和主系统的 J6 十芯口。1 根连接模块左方十芯口和主系统的 J7 十芯口。

#### (6) 下载测试:

编程下载，主系统复位。选择模式 5，当键 1 复位键置高电平，键 4 置高电平，键 3 置高电平，键 2 置低->高电平，字符液晶显示 74165 串入数据“9A”，当键 4 置高电平，键 2 置高电平，键 3 置低->高电平，数码管 1、2 显示 74164 串出数据“DA”。

### 实验 6-3. 超声波 8051 核测距电路

实验路径: \DEMO\8051Core\_DEMO\DEMO15\_8051Core\_US100\_5s

#### (1) 实验目的:

用单片机核掌握超声波测距的原理。

#### (2) 实验原理:

US100 超声波使用手册在“DEMO15\_8051Core\_US100\_5s\US100 说明”，根据其工作原理进行写出软件代码生成 ptest.HEX 文件装载到 ROM 中去。



图 6-4 超声波和 WIFI 模块

### (3) 硬件准备:

图 6-4 左边为超声波 US100 控制测距模块，对应的接口引脚在其下方，TX 和 RX 分别是接收和发送，可通过十芯线连接主系统上 FPGA 的 IO 口。

### (4) 引脚锁定:

选择模式 0

CLK 锁定核心板 50M 时钟，可以根据资料提供的引脚查找。

RST 单片机核复位选择键 8 锁定 PI07。

US100\_RS 和 US100\_TX 控制端口分别锁定主系统 J6 的 DB8\DB0。

P00[7..0]\P10[7..0]\P20[7..0] 分别是测得数据的显示分别接数码管 1 至 6PI016~PI039

以上锁定的引脚都要根据表 2.2 和 2.4 查表对应型号的引脚号。

### (5) 硬件连接

准备 1 根十芯线，1 根连接模块左下方十芯口和主系统的 J6 十芯口。

### (6) 下载测试:

编程下载，主系统复位。选择模式 0，键 8 复位，低电平复位，高电平测距，数码管 1 至 6 显示测距数据。

## 实验 6-4. 扩展存储器的 FPGA 单片系统设计

实验路径\DEMO\8051Core\_DEMO\DEMO3\_8051Core\_SRAM

### (1) 实验目的:

掌握单片机扩展 FPGA 中存储器的单片系统设计技术。

### (2) 实验原理:

参考教材[3]8.3.2 节以及相关参考资料。图 6-5 的单片系统功能与参考教材[3]8.2.3 节讨论的电路系统功能相同。即为单片机扩展一个数据存储器模块 SRAM。利用单片机的 MOVX 指令和总线口线对扩展 RAM 读写。图 5-5 中的 SRAM 模块内部的电路结构与教材[3]图 8-7 基本相同。只是为了适应单片机 CPU 核输入输出分开的特点，拆除了图中 P0 口的双向控制结构，其结果如图 6-6 所示。为了节省篇幅，图 6-5 中截去了与原图结构与功能相同的电路图（以下类似电路图也是同样目的），突出了单片机核扩展模块的电路接口特点。

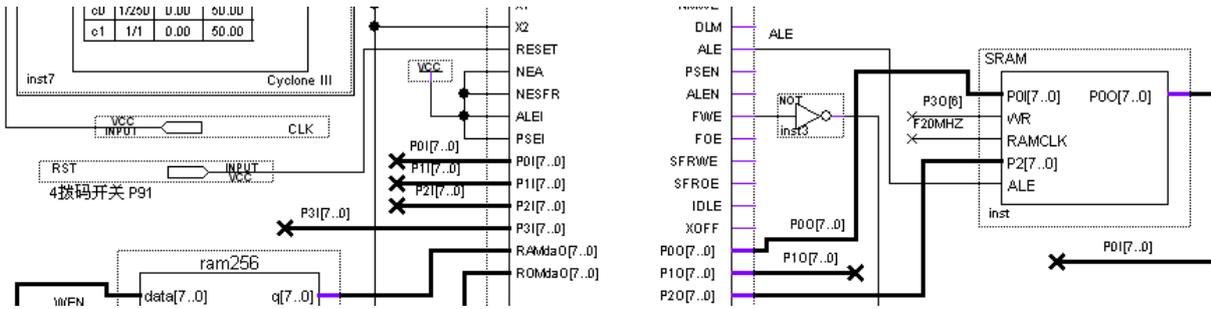


图 6-5 单片机扩展 SRAM 模块的 FPGA 片上系统电路图

结合图 6-6，考察图 6-5 的 SRAM 模块的接口情况，电路是按照单片机总线控制特点连接的：SRAM 的数据输出口与单片机 P0 口的输入口 P01[7..0]相连；而其数据输入口则与单片机 P0 口的输出口 P00[7..0]相连；SRAM 的 8 位高位地址信号线与单片机的 P20[7..0]相连；其写允许信号 WR 和地址锁存控制信号 ALE 分别与单片机的 P30[6]和 ALE 相连。P30[6]的第二功能恰是数据写允许控制信号。软件程序在 /MCU\_ASM/LCD1602.asm

**(3) 实验任务：**

根据图 6-5 的电路，完成 FPGA 单片系统综合设计与测试：

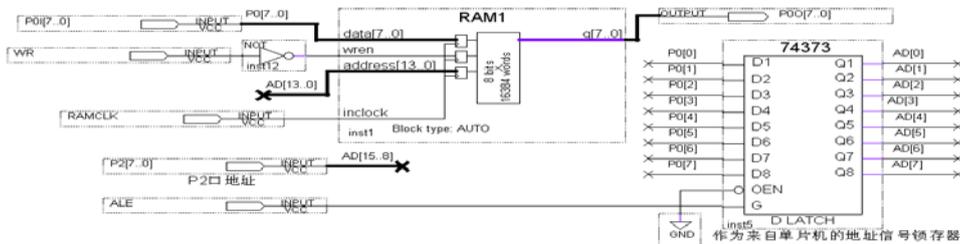


图 6-6 模块 SRAM 中的电路结构

**(4) 硬件准备：**

选择模式 5 和一块字符液晶图 6-3

**(5) 引脚锁定：**

选择模式 5

CLK 锁定核心板 50M 时钟，可以根据资料提供的引脚查找。

RST 单片机核复位选择键 1 锁定 PI00。

P10 读 RAM 允许选择键 2 锁定 PI01。

PI03[4..2]分别是字符液晶的 E\RW\RS 控制端口分别锁定主系统 JP6 的 DB6\DB3\DB0。

PL[7..0] 分别是字符液晶的 D7~D0 数据控制端口分别锁定主系统 J7 的 DB10\DB11\DB12\DB13\DBT0\DB14\DBT1\DB15。

另每例必锁定引脚 NO\MT\POE[0]\POE[2]。

以上锁定的引脚都要根据表 2.2 和 2.4 查表对应型号的引脚号。

**(6) 硬件连接**

准备 2 根十芯线，1 根连接模块右方十芯口和主系统的 JP6 十芯口。1 根连接模块左方十芯口和主系统的 JP7 十芯口

**(7) 下载测试：**

编程下载，主系统复位。选择模式 5，键 1 低电平复位，高电平使能，键 2 高电平读允许，液晶显示“78 AC 16 1E”RAM 数据。

**实验 6-5. 直流电机测控 FPGA 单片系统设计**

\DEMO\8051Core\_DEMO\DEMO11\_8051Core\_DC\_MOTO

### (1) 实验目的:

学习电机实用控制技术;

### (2) 实验原理

可参考教材[3]8.3.5节及相关资料。图6-7描述的单片系统顶层设计的功能与参考教材[3]8.2.10节讨论的直流电机单片机控制功能相同，其中的扩展模块DCMOTO内部的电路则与考教材[3]图8-21相同（不含锁相环）。如图5-7所示，模块DCMOTO的电机转速数据由输出口P2P[7..0]进入单片机P2口的输入口P2I[7..0]；而驱动电机的PWM信号脉宽控制数据由单片机的P0O[7..0]进入此模块的输入端口POP[7..0]。

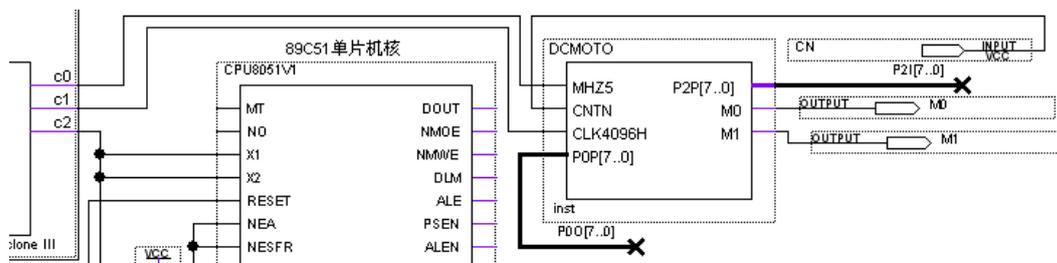


图 6-7 单片机扩展直流电机控制模块的 FPGA 片上系统电路图

图6-7中的锁相环为系统提供三个时钟信号，其中c0(5MHz)用作DCMOTO模块中光电测速脉冲毛刺消除电路时钟信号；c1(4096Hz)用作转速测量频率计的工作时钟；c2(30MHz)为单片机CPU提供工作时钟。

### (3) 实验任务:

根据图5-7的电路，完成FPGA单片系统综合设计与测试。

### (4) 硬件准备:

选择模式5和一块字符液晶图6-3、一块电机模块图4-46

### (5) 引脚锁定:

选择模式5

CLK锁定核心板50M时钟，可以根据资料提供的引脚查找。

RST单片机核复位选择键1锁定PI00。

P1[0]电机转速选择键2锁定PI01。

P1[1]控制电机正反转选择键3锁定PI02。

P1[2]电机测速选择键4锁定PI03。

PI03[4..2]分别是字符液晶的E\RW\RS控制端口分别锁定主系统JP6的DB6\DB3\DB0。

PL[7..0]分别是字符液晶的D7~D0数据控制端口分别锁定主系统J7的DB10\DB11\DB12\DB13\DBT0\DB14\DBT1\DB15。

M0、M1电机控制正反转锁定主系统J3的DA27和DA26。

CN是测速计数红外接收端口锁定主系统DA22。

CLKM0是驱动电机的时钟锁定CLKB0。

另每例必锁定引脚NO\MT\POE[0]\POE[2]。

以上锁定的引脚都要根据表2.2和2.4查表对应型号的引脚号。

### (6) 硬件连接

准备2根十芯线，1根连接液晶模块右方十芯口和主系统的JP6十芯口。1根连接模块左方十芯口和主系统的JP7十芯口。另一个十芯线连接电机模块J1口和主系统JP3口。

### (7) 下载测试:

当键4、键3为高电平，键2由高到低，电机控制转速，液晶上STEP:会显示数据；当键4、键2为高电平，

键 3 由高到低，电机控制正反转，液晶上 N 或 P；当键 2、键 3 为高电平，键 4 由高到低，电机测速，液晶上显示 SPEED XXC/C。如图 6-8

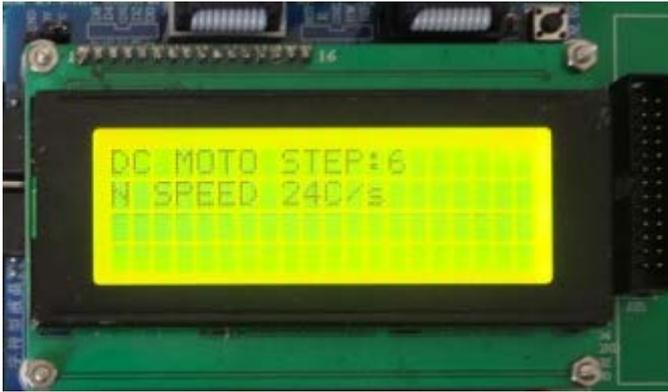


图 6-8

## 实验 6-6. 等精度频率计 FPGA 单片系统设计

\DEMO\8051Core\_DEMO\DEMO6\_8051Core\_FRQ\_TEST

### (1) 实验目的:

学习高精度频率、脉宽和占空比测试电路设计，以及单片系统设计技术。

### (2) 实验原理:

参考教材[3]8.3.7 节以及相关参考资料。

等精度频率计设计项目是一个十分典型的单片机扩展 FPGA 的应用实例。在这项设计中，单片机和 FPGA 都发挥了相互之间不可替代的作用，实现了硬件设计和软件设计、硬件功能和软件功能间默契配合后的上佳的功能项目和技术指标的表现。这在 8.2.9 节和参考教材[2]、[3]中都给出了详细的说明。如果 8.2.9 讨论的设计项目全部装进一单片 FPGA 中，将在更多的技术指标方面向实用领域有更大的迈进。图 5-9 给出的电路结构就是由单片机核与等精度频率测试硬件模块相结合的单片系统设计方案。图中 et 模块的结构、功能和端口都与参考教材[3]图 8-19 的同名模块相同，而且单片机模块的接口形式也与图 8-19 相同。因此 8.2.9 节给出的系统中，单片机的程序可以不加更改地直接用到图 5-9 的单片模块中。由于单片机核的 I/O 口是按输入输出方向分开的，所以读者必须详细了解和分辨图 6-9 中 et 模块各端口与单片机核 I/O 口的接口方式，以及与教材[3]图 8-19 电路模块接口方式的不同之处。

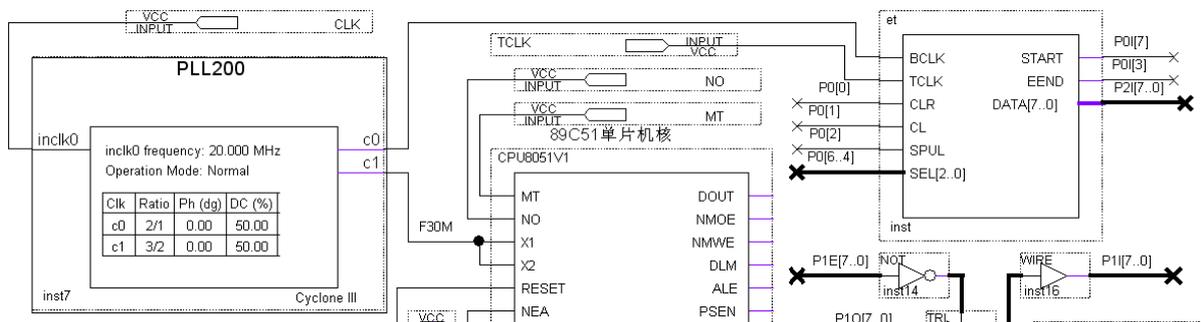


图 6-9 单片机扩展等精度频率测试模块的 FPGA 片上系统电路图

### (3) 实验任务:

根据图 5-9 的电路，完成 FPGA 单片系统综合设计与测试。

### (4) 硬件准备:

选择模式 5 和一块字符液晶图 5-3。

### (5) 引脚锁定:

选择模式 5

CLK 锁定核心板 50M 时钟，可以根据资料提供的引脚查找。

RST 单片机核复位选择键 1 锁定 PIO0。

P1[0]测频选择键 2 锁定 PIO1。

P1[1]测脉宽选择键 3 锁定 PIO2。

P1[2]测空比选择键 4 锁定 PIO3。

PIO3[4..2]分别是字符液晶的 E\RW\RS 控制端口分别锁定主系统 JP6 的 DB6\DB3\DB0。

PL[7..0]分别是字符液晶的 D7~D0 数据控制端口分别锁定主系统 J7 的

DB10\DB11\DB12\DB13\DBT0\DB14\DBT1\DB15。

TCLK 待测时钟锁定 CLKB0。

另每例必锁定引脚 NO\MT\POE[0]\POE[2]。

以上锁定的引脚都要根据表2.2和2.4查表对应型号的引脚号。

## (6) 硬件连接

准备 2 根十芯线，1 根连接液晶模块右方十芯口和主系统的 JP6 十芯口。1 根连接模块左方十芯口和主系统的 JP7 十芯口。

## (7) 下载测试：

选择模式5，用一根单线一端连接J17的CLKB端，作为待测频率输入端，另一端连接时钟源J13的任意时钟，作为待测时钟输出端。键1高电平复位，电平测试，键2由高到低，液晶显示测频数据，对照一下选择的时钟源是多少，是否对应；再按动键1由高到低，按键2，由高到低，液晶显示脉宽数据。再次按动键1复位，置低电平，按动键3，液晶显示成功比例。

### 实验6-7 8051控制WIFI和实验系统的实验

\DEMO\8051Core\_DEMO\DEMO13\_Basic8051\_R2WiFi\_XLW210A

### 实验6-8 8051控制网口和实验系统的实验

\DEMO\8051Core\_DEMO\DEMO12\_Basic8051\_W5200

### 实验6-9 8051控制GPS的实验

\DEMO\8051Core\_DEMO\DEMO7\_8051Core\_GPS

### 实验6-10 8051控制AD0809实验

\DEMO\8051Core\_DEMO\DEMO4\_8051Core\_ADC0809

### 实验6-11 8051控制点阵液晶实验

\DEMO\8051Core\_DEMO\DEMO5\_8051Core\_LCD128X64

### 实验6-12 8051控制DDS移相信号发生器实验

\DEMO\8051Core\_DEMO\DEMO8\_8051Core\_DDS\_PHASE

### 实验6-13 8051控制李萨如图输出实验

\DEMO\8051Core\_DEMO\DEMO9\_8051Core\_LEES\_WAVE

### 实验6-14 8051控制占空比可调方波输出实验

\DEMO\8051Core\_DEMO\DEMO10\_8051Core\_MD\_SQR

### 实验6-15 805118B20数字温度测控实验

\DEMO\8051Core\_DEMO\DEMO14\_8051Core\_DS18B20

## 附录 1 MIF 文件生成器使用方法（文件路径：\VIVI\_FILE\mif 任意波编辑）

本讲义中给出的一些有关 LPM RAM 或 ROM 的实验都将用到 mif 格式初始化文件，这可以用不同方法获得，但比较方便的方法是使用 mif 文件生成器。这里介绍康芯公司为本书读者提供的免费的 mif 生成软件 Mif Maker 的使用方法(索取：[www.kx-soc.com](http://www.kx-soc.com))。双击打开 Mif\_Maker2010，如图 F1-1 所示。首先对所需要的 mif 文件对应的波形参数进行设置。如图 F1-2 所示，选择“查看”，并于此下拉菜单中选择“全局参数设置”。如选择波形参数：数据长度 256，输出数据位宽 8，数据表示格式 16 进制（有的情况下需要选择有符号类型，如实验中的 AM 信号发生器的设计），初始相位 120 度（如设计 SPWM 中要用到此相位设定），按“确定”后，将出现一波形编辑窗。然后再选择波形类型。选择“设定波形”，再选择“正弦波”，如图 F1-3 所示。



图 F1-1 打开“Mif\_Maker2010”



图 F1-2 设定波形参数

这时，图 F1-3 将出现正弦波形。如果要编辑任意波形，可以选择“手绘波形”项，在下拉菜单中选择“线条”（图 F1-4），表示可以手工绘制线条。然后即可以在图形编辑窗中在原来的正弦波形上绘制任意波形（图 F1-4）。最后选择“文件”中的“保存”，将此编辑好的波形文件以 mif 格式保存（图 F1-5）。如取名为 WAVE1.mif。如果要了解编辑波形的频谱情况可以选择“查看”项的“频谱”。如图 F2-6 所示的锯齿波的归一化频谱显示于图 F1-7 上。

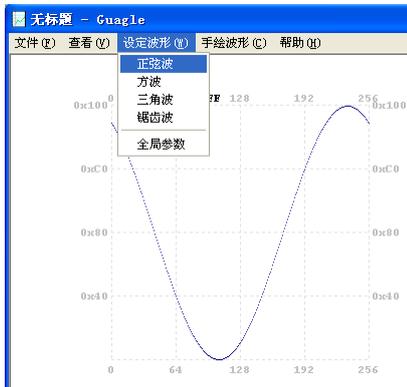


图 F1-3 选择波形类型

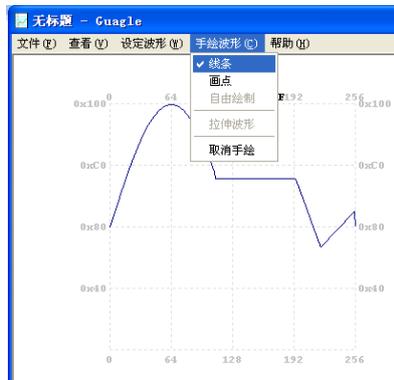


图 F1-4 手动编辑波形

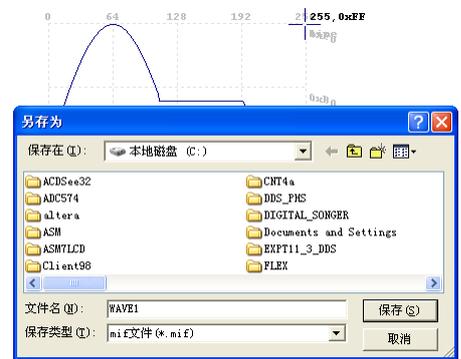


图 F1-5 存储波形文件

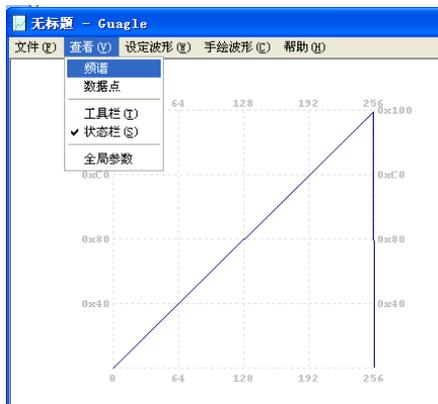


图 F2-6 选择频谱观察功能

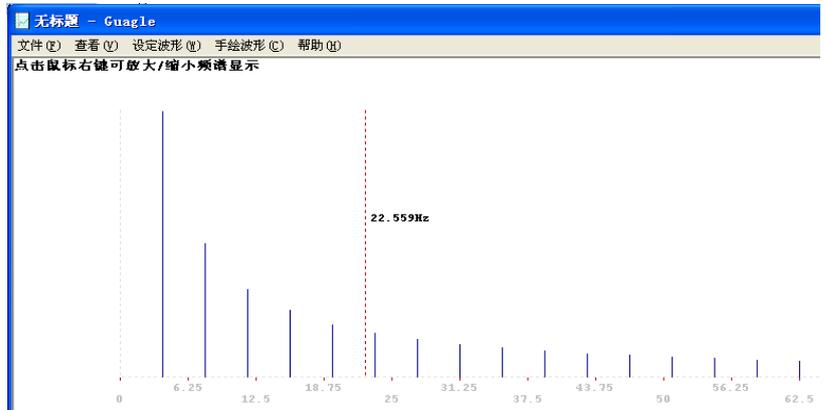


图 F2-7 锯齿波频谱

## 附录 2 STC89C 单片机编程下载方法 (编程软件:\VIVI\_FILE\stc89c51 编程下载.rar)

要将应用程序下载到 STC89C 单片机，首先须安装好 STC\_ISP 软件。安装完成后，打开 STC\_ISP v4.80 下载软件的界面如图 F2-1 所示。

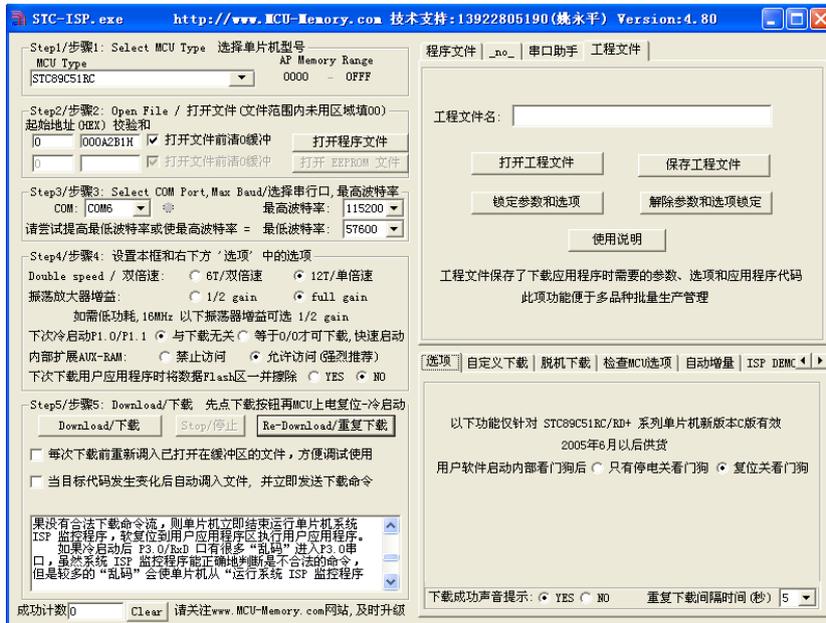


图 F2-1 STC 单片机下载界面

软件设置步骤如下：

步骤1：选择单片机型号。在**MCU Type**栏目下选择所使用的单片机型号，如STC89C51, STC12C5410, STC12C5410AD 等，如图F2-2所示。

步骤2：打开烧录文件。先确认硬件连接正确，然后点击“**打开程序文件**”并在对话框内找到所要下载的\*.hex文件或\*.bin文件，如图F2-3所示。

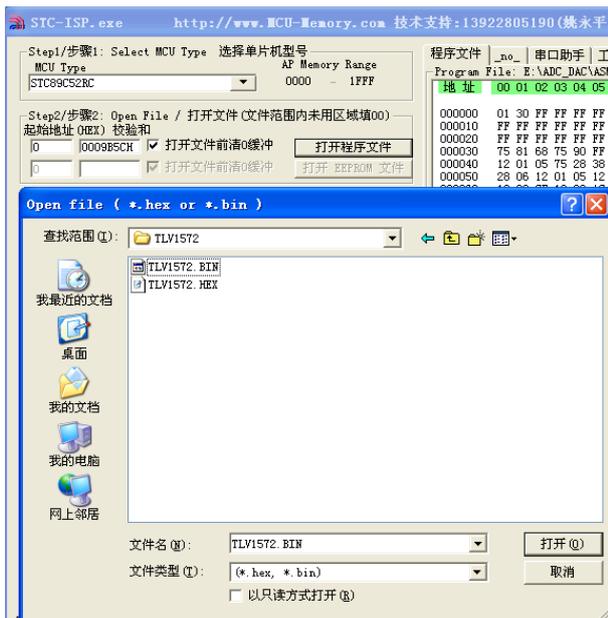


图 F2-2 选择 STC 单片机型号

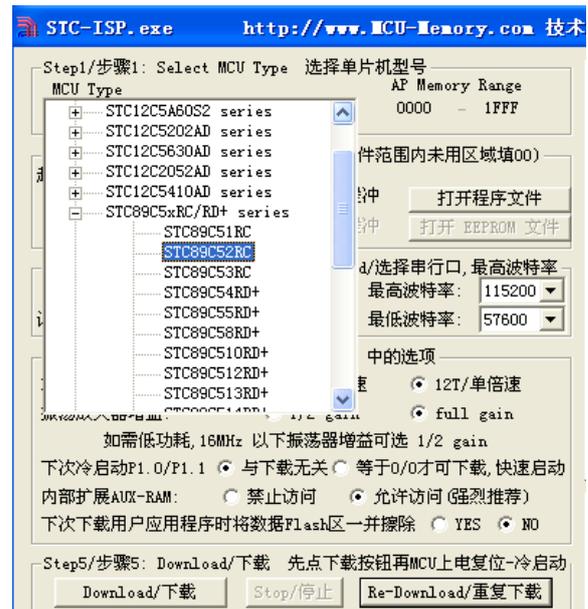


图 F2-3 选择需烧录的程序

步骤3：选择电脑的串行口。根据下载数据线连接情况选择COM端口和下载波特率。如串行口1--COM1, 串行口2--COM2,.... 有些新式笔记本电脑没有RS-232 串行口,可通过USB-RS232 转接器下载。有些USB-RS232 转接器与电脑连接后，还需要安装相应的驱动程序。再根据安装后的USB-RS232 转接器，选择下载口，如图F2-4所示。下载波特率的设置方法如图F2-5所示。

步骤4：**设置是否双倍速**，双倍速选中Double Speed即可。STC89C51RC/RD+ 系列可以反复设置 双倍速/单倍速，新的设置停电后重新冷启动后才能生效。

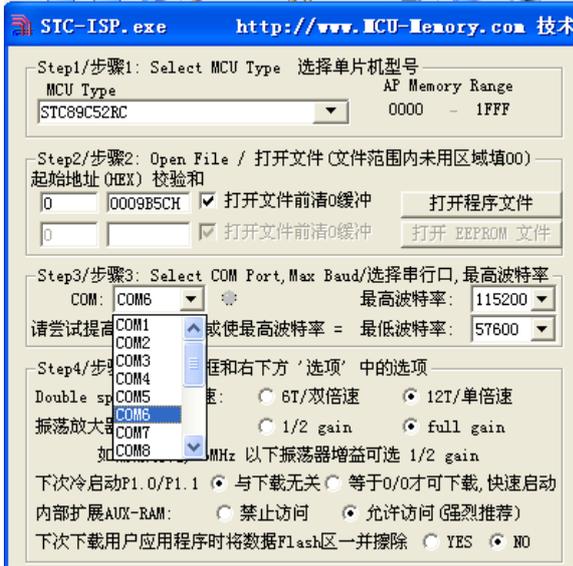


图 F2-4 选择 COM 下载口

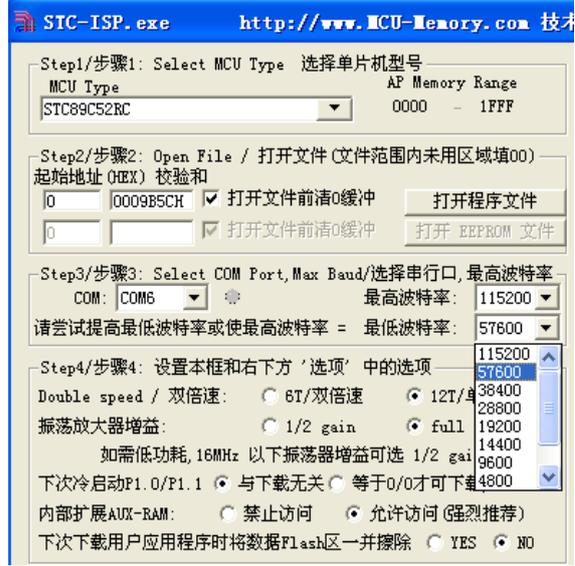


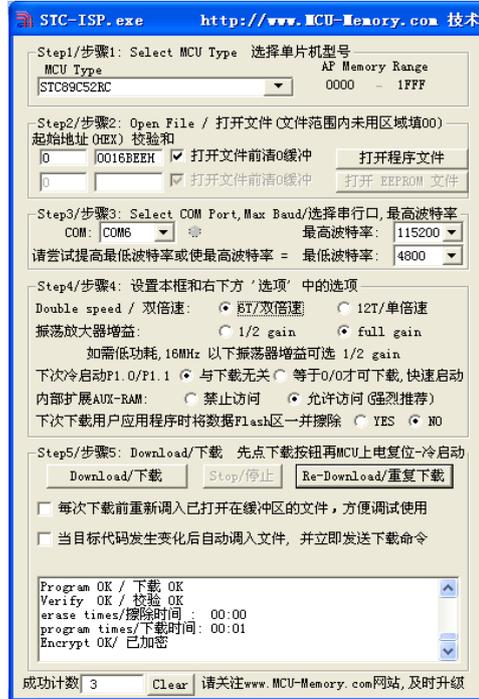
图 F2-5 选择下载波特率

编程说明文件: \KX-CDS\重要文件\单片机编程说明.pdf

USB-RS232 驱动文件: :\\VIVI\_FILE\INF\_USB\_RS232\_DRIV\98ME\_20011\_2kXP\_20024

Chinese:正在尝试与 MCU/单片机 握手连接 ...  
 Chinese:连接失败, 请尝试以下操作:  
 1.在单片机停电状态下, 点击下载按钮, 再给单片机上电  
 2.停止下载, 重新选择 RS-232 串口, 接好电缆  
 3.可能需要先将 P1.0/P1.1 短接到地  
 4.可能外部时钟未接  
 5.因 PLCC、PQFP 转换座引线过长而引起时钟不振荡, 请调整参数  
 6.可能要升级电脑端的 STC-ISP.exe 软件  
 7.若仍然不成功, 可能 MCU/单片机内无 ISP 系统引导码, 或需退回升级, 或 MCU 已损坏  
 8.若使用 USB 转 RS-232 串口线下载, 可能会遇到不兼容的问题, 可以让我们帮助购买兼容的 USB 转 RS-232 串口线  
 仍在连接中, 请给 MCU 上电...

图F2-6 在线编程提示信息

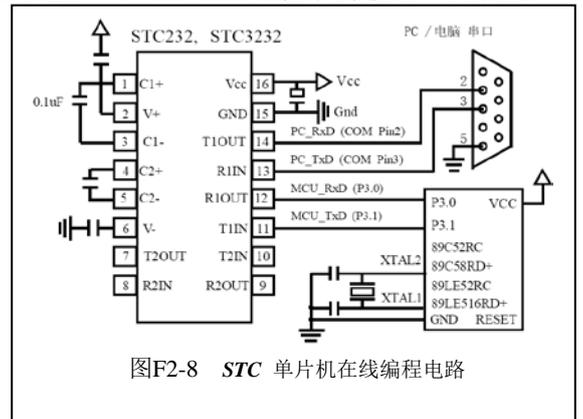


图F2-7 下载成功后的提示信息

步骤5: 选择“Download/ 下载”按钮下载用户的程序进单片机内部, 也可选择“Re-Download/ 重复下载”按钮。下载过程中, 左下角的窗口会出现提示信息, 如图F2-6所示。若选中“步骤5”中的两个条件项, 就可以在每次用KEIL编译用户程序时, 将HEX代码自动加载到STC-ISP。点击“Download/下载”, 然后手动接通单片机的电源开关便可把可执行文件\*.HEX写入到单片机内。程序写入完毕后, 出现如图F2-7所示信息。目标板开始运行用户应用程序。

关于硬件连接: STC 单片机在线编程电路如图F2-8所示。

- (1). MCU/ 单片机 RXD(P3.0) --- RS-232 转换器 --- PC/ 电脑 TXD(COM Port Pin3)
- (2). MCU/ 单片机 TXD(P3.1) --- RS-232 转换器 --- PC/ 电脑 RXD(COM Port Pin2)



图F2-8 STC 单片机在线编程电路