

嵌入式系统设计与开发系列

FPGA 嵌入式系统 设计与开发指南

林英撑 童晓华 刘向宇 编著

- 频率计设计
- 人机界面设计
- 信号源设计
- 高速数据采集系统设计
- SOPC设计及嵌入式智能设计
- LiveDesign实时设计及嵌入式软件调试
- 虚拟仪器调试及JTAG边界扫描
- 测量仪器平台实例



中国电力出版社
CHINA ELECTRIC POWER PRESS

嵌入式系统设计与开发系列

FPGA嵌入式系统设计 与开发指南

林英撑 童晓华 刘向宇 编著



中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书以 Altera 公司的飓风系列 FPGA 为硬件基础,讲述常见的 FPGA 应用模块和系统设计,并引入 Altium 公司开发的 Altium Designer 开发环境进行 SOPC 设计。

全书共 11 章,包括频率计设计、人机界面设计、信号源设计、高速数据采集系统设计、SOPC 设计及嵌入式智能设计、LiveDesign 实时设计及嵌入式软件调试、虚拟仪器调试及 JTAG 边界扫描、FPGA 到 PCB 的转换及同步更新、测量仪器平台实例。本书侧重 FPGA 的实战应用项目,以多个实践实例设计为切入点,全面进述系统的设计原理、框架设计、硬件设计、VHDL 设计、模块封装、顶层设计、系统设计、误差分析等,以求将整个系统设计的细节全面展现给读者。

本书实战性和指导性强,适合从事单片机、嵌入式系统开发的工程技术人员参考。

图书在版编目(CIP)数据

FPGA 嵌入式系统设计与开发指南/林英撑,童晓华,刘向宇编著. —北京:中国电力出版社,2012.1

(嵌入式系统设计与开发系列)

ISBN 978-7-5123-2600-2

I. ①F… II. ①林… ②童… ③刘… III. ①可编程序逻辑器件-系统设计-指南 IV. ①TP332.1-62

中国版本图书馆CIP数据核字(2012)第010347号

中国电力出版社出版、发行

(北京市东城区北京站西街19号 100005 <http://www.cepp.sgcc.com.cn>)

北京市同江印刷厂印刷

各地新华书店经售

*

2012年6月第一版 2012年6月北京第一次印刷

787毫米×1092毫米 16开本 19.75印张 475千字

印数 0001—3000册 定价 39.00元



敬告读者

本书封面贴有防伪标签,加热后中心图案消失

本书如有印装质量问题,我社发行部负责退换

版权专有 翻印必究



前言

FPGA 是英文 Field Programmable Gate Array 的缩写，即现场可编程门阵列。FPGA 最大的特点就是可以很方便地实现 ASIC，可以作为 ASIC 的验证原型，进而可以实现软件硬化的目标。也可以满足 ASIC 小批量的生产要求。

FPGA 是在 PAL、GAL、PLD 等可编程器件的基础上进一步发展起来的新型器件。目前生产 FPGA 器件的厂家很多，应用广泛的有 XILINX 公司、ALTERA 公司等。

FPGA 的烧写、加载方法和 CPLD 不尽相同。CPLD 的程序是烧写在片内非易失性存储器（如 EEPROM 或者 FLASH 之中），所以 CPLD 器件有烧写次数的限制。而 FPGA 是通过一个片外的非易失性存储器来存储程序，在上电之后，从片外存储器中加载程序，所以 FPGA 可以反复使用，而不必担心烧写次数的限制。

FPGA 内部有丰富的 I/O 资源和触发器资源，能兼容 CMOS、TTL 电平。FPGA 的应用领域越来越广泛，从医疗电子到通信行业、汽车电子、消费电子领域都有越来越多 FPGA 的应用实例。

目前市面上的 FPGA 图书中，介绍编程语言、FPGA 理论、FPGA 原理的书较多，而对于模块化设计、实例系统设计等实战型实例涉及甚少，实用性不强，本书就是为了弥补这种不足而编写的。

本书的特色

本书以 Altera 公司的飓风系列 FPGA 为硬件基础，讲述常见的 FPGA 应用模块和系统设计，并引入 Altium 公司开发的 Altium Designer 开发环境进行 SOPC 设计。本书讲述的模块例子、实例均可用于 Altera 公司的 EP1C、EP2C、EP3C 等各个“飓风”FPGA 芯片上面，还可以移植到其他类型、型号的 FPGA 上面，通用性很好。考虑到很多大中院校、研究所、实验室都以 EP2C 系列 FPGA 作为硬件平台，所以，本书的实例基本以 EP2C8 为硬件平台。同时，本书提供了改进型 Cyclone III 配套教学电路板，以 EP3C 全系列为主芯片，详细信息请访问 <http://www.techruisi.com>。

本书侧重于 FPGA 实战应用项目，以多个实践实例设计为切入点，全面讲述系统的设计原理、框架设计、硬件设计、VHDL 设计、模块封装、顶层设计、系统设计、误差分析等各个方面，以求将整个系统设计的细节全面展现给读者朋友。

本书在讲述实战的同时，也会涉及部分理论知识，因为理论知识是基础，是基石，是指引我们设计的指向标。在理论的指引下，更好地付诸于实践，才能发现问题，解决问题。所以本书提倡一种学习、使用 FPGA 的思想：以理论为指导，以实践为平台，两者结合，才能更快地掌握 FPGA。FPGA 和单片机、DSP、ARM 最大的不同是并行处理，本书自始至终都以此为主线，也希望大家能更深入地理解并行处理的精髓。

主要内容

第 1 章以频率计为设计实例，讲述如何用 FPGA 来实现等精度测频法。

第 2 章以人机界面为实例，讲述 FPGA 系统中如何构建人机交互界面，如输入部分、输出部分。

第 3 章以任意函数发生器为实例，讲述用 FPGA 来实现 DDS 直接频率合成，以产生任意波形的原理、设计思路和实现方法。

第 4 章高速数据采集系统，讲述基于 FPGA 的高速数据采集、FIFO 设计、后端 CPU 配合等。

第 5 章讲述 FPGA 的 SOPC 设计及嵌入式智能设计。

第 6 章讲述如何实现 LiveDesign 实时设计及嵌入式软件调试。

第 7~9 章，以 Altium 公司的 Altium Designer 为开发环境，讲述如何对 SOPC 进行库设计、模块设计。

第 10 章，以 Altium 公司的 Altium Designer 为开发环境，讲述如何用 AD 开发环境来设计 FPGA 模块，进而形成 PCB 文件，并导入到 FPGA 芯片中。

第 11 章综合前面章节讲述的模块设计，组建一个以测量为主线的 FPGA 系统。

致 谢

在本书的编写过程中，参阅了大量的书籍、文献资料以及网络相关资源，并在书中引用了其中的部分文字和插图，在此向这些文字和插图的版权所有者和原作者表示感谢。尹欣、钟伟辉、颜训理、颜业等为本书做了很多工作，在此一并向他们表示感谢！此外，还要特别感谢我的家人，他们在本书的编写过程中给了我很大的鼓励和支持。

限于作者水平和时间，书中难免存在疏漏和不足，欢迎大家批评和指正。

本书配套的 VHDL 代码及设计图纸，请登录 <http://52dsk.com> 下载。

凡是购买本书的读者，将有机会免费获得本书配套的 PCB 空板，数量有限，先申请先得，详情请登录 <http://bbs.52dsk.com>。

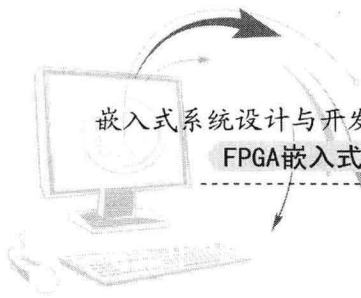
(1) 本 PCB 空板由睿思科技 (www.techruisi.com) 赞助提供，仅供大家学习参考之用，请勿用于商业用途。欢迎大家来论坛参加 FPGA 学习、讨论。

(2) 申请 PCB 空板时，请发邮件至 support@52dsk.com 的专门技术支持邮箱进行申请。空板的邮寄费用需自付。

(3) 关于本书内容及电路板的使用问题可以在 <http://bbs.52dsk.com> 中讨论，本人将会定时上线和大家交流。作者邮箱：fcgexcellence@163.com，或者微博@fcgexcellence。

作 者

2012 年 5 月



目 录

前言

第 1 章 频率计设计	1
1.1 频率计原理	1
1.1.1 直接测频法	2
1.1.2 同步测频法	2
1.1.3 测周期法	3
1.2 原理构架	4
1.3 硬件组成	4
1.3.1 总体硬件图	4
1.3.2 FPGA 硬件设计	4
1.3.3 前端调理电路设计	4
1.3.4 方波生成电路设计	6
1.3.5 LCD 1602 电路设计	7
1.4 FPGA 设计	7
1.4.1 顶层设计图	7
1.4.2 LCD 1602 控制器	8
1.4.3 产生方波信号	11
1.4.4 计数器模块	12
1.5 本章小结	13
第 2 章 人机界面设计	14
2.1 原理构架	14
2.2 硬件组成	16
2.2.1 总体硬件图	16
2.2.2 FPGA 部分	16
2.2.3 PS2 键盘部分	19
2.2.4 液晶模块	20
2.2.5 VGA 模块	20
2.2.6 蜂鸣器模块	21

2.3	FPGA 设计.....	21
2.3.1	PS2 控制模块.....	21
2.3.2	液晶控制模块.....	22
2.3.3	VGA 驱动模块.....	27
2.3.4	蜂鸣器驱动模块.....	30
2.4	改进要点.....	34
2.4.1	添加触摸屏模块.....	34
2.4.2	VGA 模块的改进方法.....	35
2.4.3	其他模块的改进方法.....	35
2.5	本章小结.....	36

第 3 章 信号源设计 37

3.1	信号发生原理.....	37
3.1.1	信号的产生.....	37
3.1.2	DDS 原理.....	38
3.1.3	DDS 的优点.....	39
3.1.4	DDS 参数.....	40
3.1.5	DDS 实现方法.....	42
3.2	原理构架.....	43
3.3	硬件组成.....	44
3.3.1	总体硬件图.....	44
3.3.2	人机交互部分.....	46
3.3.3	DAC 部分.....	47
3.3.4	信号调理设计.....	47
3.3.5	晶振输入部分.....	48
3.4	FPGA 设计.....	48
3.4.1	相位累加器模块.....	48
3.4.2	ROM 模块.....	50
3.4.3	人机交互模块.....	55
3.4.4	波形选择模块.....	58
3.5	改进要点.....	58
3.5.1	压缩 ROM 表.....	58
3.5.2	更换 DAC.....	59
3.5.3	改进构架.....	60
3.6	本章小结.....	60

第 4 章 高速数据采集系统设计 61

4.1 数据采集方法	61
4.2 数据采集设计思路	62
4.2.1 传统数据采集模型	62
4.2.2 高速数据采集模型	62
4.2.3 基于 FPGA 数据采集原理	63
4.3 原理构架	64
4.4 硬件组成	65
4.4.1 硬件总体框架	65
4.4.2 高速 A/D	65
4.4.3 信号调理电路	68
4.4.4 后端 CPU	68
4.4.5 PCB 布线注意事项	70
4.5 FPGA 设计	70
4.6 后端处理设计	76
4.6.1 软件流程图	76
4.6.2 读取 FIFO	76
4.6.3 数据处理和显示	77
4.7 改进要点	77
4.7.1 提高采样率	77
4.7.2 提高分辨率	78
4.7.3 其他改进要点	78
4.8 本章小结	80

第 5 章 FPGA 的 SOPC 及嵌入式智能设计 81

5.1 嵌入式应用程序开发	81
5.1.1 创建一个嵌入式项目	81
5.1.2 代码编辑基本知识与通用设置	81
5.1.3 编辑器	87
5.2 TASKING 工具链	88
5.2.1 嵌入式软件构建流程	88
5.2.2 绑定目标项目	89
5.2.3 处理器选项	89
5.2.4 C 编译器选项	90
5.2.5 汇编语言编译器选项	98
5.2.6 链接选项	99

5.2.7	编译器输出	100
5.2.8	第一次编译	100
5.3	利用微处理器软核	101
5.3.1	在 FPGA 原理图中使用微处理器软核	101
5.3.2	片上调试功能	101
5.3.3	放置一个软核到 FPGA 原理图	102
5.4	连接一个嵌入式项目到硬件设计	105
5.4.1	连接一个嵌入式项目到 FPGA 项目	105
5.4.2	连接程序存储器到处理器核	106
5.4.3	连接项目	107

第 6 章 LiveDesign 实时设计及嵌入式软件调试 109

6.1	初始化 LCD.....	109
6.1.1	使用提供的 LCD 控制模块	110
6.1.2	开发一个自定义的 LCD 软件模块	110
6.2	使用 LiveDesign 初始化 LCD	110
6.2.1	调整软件定时环路	110
6.2.2	校验 LCD 写操作	118
6.2.3	校验 LCD 读操作	120
6.2.4	LCD 指令定义.....	121
6.3	调试设计.....	122
6.3.1	调试菜单	122
6.3.2	嵌入式控制面板	123
6.3.3	代码调试	129

第 7 章 FPGA 设计提高——Wishbone 130

7.1	设计一个视频显示系统	130
7.1.1	视频捕获具体细节	130
7.1.2	在触摸屏上显示视频的具体细节	131
7.1.3	系统的高层次观察	131
7.2	使用微处理器软核	131
7.2.1	在 FPGA 原理图中使用软核	132
7.2.2	访问软核使用资料	133
7.2.3	TSK3000 处理器	133
7.2.4	放置一个软核到 FPGA 原理图	135
7.3	Wishbone 互联架构.....	135
7.3.1	Wishbone 兼容的核	135

7.3.2	Wishbone 界面信号描述.....	136
7.3.3	Wishbone 信号管脚命名原则.....	137
7.4	放置 FPGA 外设.....	138
7.5	总线互联和仲裁.....	140
7.5.1	Wishbone 互联器件 WB_INTERCON.....	141
7.5.2	配置 Wishbone 互联元件.....	141
7.5.3	连接外设.....	143
7.5.4	管理存储器访问.....	144
7.5.5	多主公用一从的仲裁方法.....	144
7.5.6	Wishbone 双主或多主器件.....	145
7.5.7	配置 Wishbone 多主器件.....	146
7.5.8	多主器件的定义.....	147
7.6	与外部存储器接口连接.....	148
7.6.1	配置 SRAM 控制器.....	148
7.6.2	布局和配置存储器控制器.....	149
7.6.3	从器件的多个存储器.....	150
7.6.4	放置元件的端口接插件.....	151
7.6.5	连接 FPGA 设计.....	153
7.7	配置处理器存储器.....	153
7.7.1	处理器存储器的划分.....	154
7.7.2	配置处理器存储器.....	155
7.7.3	配置处理器外设.....	155
7.7.4	定义处理器外设.....	156
7.8	确定设计、编译和错误处理.....	157
7.9	新建 FPGA 项目的嵌入式系统.....	157
7.9.1	编辑基础.....	157
7.9.2	绑定目标项目.....	158
7.9.3	项目选项.....	158
7.9.4	链接嵌入式项目到 FPGA 项目.....	165
7.9.5	链接嵌入式项目到 FPGA 项目.....	167
7.9.6	构建整个设计并查看实时更新软件的效果.....	167
7.10	更新设计以显示视频.....	170
7.10.1	定义段.....	170
7.10.2	实时更新软件.....	171
7.10.3	更新软件以显示视频.....	172
7.10.4	更新 Wishbone 端口 I/O.....	174
7.10.5	使用虚拟仪器.....	175
7.10.6	添加虚拟仪器.....	176
7.10.7	重新构建带有虚拟仪器的项目.....	178

7.10.8	更新嵌入式项目以使用虚拟仪器	178
7.10.9	添加附加的源文件	178
7.10.10	更新 main.c 主函数	180
7.10.11	用虚拟仪器来监控设计	185

第 8 章 嵌入式软件代码调试及 C 到硬件 188

8.1	处理器核实时调试	188
8.1.1	片上调试	188
8.1.2	仿真	188
8.1.3	嵌入式控制面板	189
8.1.4	Nexus 调试器	194
8.2	C 到硬件	195
8.2.1	C 到硬件编译器的优点	196
8.2.2	使用 CHC 编译器	196
8.2.3	C 到硬件的实现	197
8.2.4	为 C 到硬件提供空间	197
8.2.5	添加特殊应用处理器 ASP	197
8.2.6	给 ASP 访问到存储器	198
8.2.7	配置 ASP	201
8.2.8	通过 ASP 完善性能	202

第 9 章 虚拟仪器调试及 JTAG 边界扫描 204

9.1	虚拟仪器简介	204
9.2	使能嵌入式虚拟仪器	209
9.3	使用嵌入式虚拟仪器	209
9.4	FPGA 项目与试验板交互	213
9.4.1	Nanoboard 通信	213
9.4.2	技术背景	214
9.4.3	FPGA I/O 状态实时显示	216
9.4.4	实时交叉探针	218
9.4.5	在 Nanoboard 上观察 PWM 项目	218

第 10 章 FPGA 到 PCB 转换及同步更新 219

10.1	移植设计到预备的 FPGA 器件	219
10.1.1	选择适合该设计的 FPGA 器件	219
10.1.2	绑定为一个新的 FPGA 器件	220

10.1.3	灵活配置.....	222
10.1.4	绑定设计到一个新的 FPGA 器件	225
10.1.5	压缩设计.....	225
10.2	从 FPGA 到 PCB 项目	227
10.2.1	各文档间的关系.....	227
10.2.2	FPGA 到 PCB 向导的应用	229
10.2.3	FPGA 配置.....	229
10.2.4	FPGA 管脚分配.....	230
10.2.5	选择目标 PCB 项目	232
10.2.6	配置 FPGA 器件原理图.....	232
10.2.7	配置顶层原理图及其方框图	232
10.2.8	运行 FPGA 到 PCB 向导	233
10.2.9	修改自动生成的原理图.....	235
10.2.10	关于 FPGA 特殊用途的管脚	235
10.2.11	重新创建自动生成的原理图	236
10.3	保持项目间同步.....	236
10.3.1	FPGA 工作区映射图 Workspace Map	236
10.3.2	同步对话框.....	238
10.3.3	同步匹配信号.....	239
10.3.4	同步未匹配信号.....	240
10.4	配置 FPGA I/O.....	243
10.4.1	配置 I/O 标准.....	243
10.4.2	使用 FPGA 信号管理器.....	244
10.5	手动链接 FPGA 和 PCB 项目	245
10.5.1	支持器件.....	246
10.5.2	创建链接.....	247
10.5.3	链接自动生成的原理图到现有 PCB 项目	249
10.5.4	手动连接 PCB 到 FPGA 项目	249
10.6	管脚交换.....	250
10.6.1	PCB 文档中管脚交换	250
10.6.2	FPGA 项目中管脚交换.....	256
10.6.3	在 PCB 和 FPGA 项目中同步管脚交换.....	257
10.6.4	管脚交换和同步更新.....	257

第 11 章 测量仪器平台系统设计 259

11.1	平台设计思路及框架.....	259
11.1.1	原理框架.....	260
11.1.2	模块划分.....	260

11.1.3	模拟电路模块	260
11.1.4	高速 ADC 模块	262
11.2	硬件组成	267
11.2.1	硬件总体框架	267
11.2.2	模拟前端	268
11.2.3	高速 A/D	269
11.2.4	FPGA 芯片	275
11.2.5	后端 CPU	276
11.2.6	辅助电路	280
11.3	FPGA 设计	281
11.4	后端处理设计	291
11.4.1	读取 FIFO	292
11.4.2	数据处理算法——FIR	292
11.4.3	数据处理算法——IIR	293
11.4.4	数据处理算法——卷积	293
11.4.5	数据处理算法——傅里叶变换	295
11.4.6	数据处理算法——A 律	298
11.5	误差分析	300
11.6	应用实例	300
11.6.1	超声测量	300
11.6.2	数字示波器	301
11.7	改进要点	301
11.8	本章小结	302

频率计设计

频率计是检测信号频率的仪器。在平常的电子设计工作中，频率计、示波器、万用表都是常用的仪器仪表，对电路调试有着重要的测试参考作用。作为常用的电子类仪器仪表，频率计在平常的电子研发、测试和生产过程中起着很重要的作用，所以，了解频率计的原理、设计一款自己的频率计，也不失为一个有趣而有意义的事情。

通过数学计算公式可以得知：频率的倒数就是周期。所以，通过频率计除了能获得频率值之外，还可以获得周期。现在，功能强大的高级频率计除了测量频率、周期外，还可以测量脉冲宽度、占空比等参数。

测频率是通过在标准时间内对被测信号的计数来实现的。其测量方法很多，总结起来，也就是：在预先设定的时间内（如 0.5s），通过计数器对输入信号进行计数，得到的就是单位时间内的变化次数，这就是待测信号的频率。

不同的测频率的方法，对于预定时间（称为闸门时间或者门限时间）的设定和计数器的工作方法有着不同的定义，所以，不同的测频率方法有着各自的优点和缺点。

本章将讲述用 FPGA 来设计一个数字频率计。利用 FPGA 所特有的频率高、并行执行的优点，可使得这种频率计具有效率高、高度集成硬件电路、速度快等优点。

以前纯电路的测频电路是通过多片芯片、门电路等分立电路来组成的，器件不一致容易导致测量结果不准，现在全部电路都集中在一片 FPGA 里，有利于外部电路板设计的精简，并保持性能稳定。

1.1 频率计原理

频率计的设计思路一般可有以下几种。

(1) 专用芯片 ASIC。这种方法，测量性能好，有成套成熟的方案，缺点：功能固定，很难扩展更多的功能。

(2) 通用单片机。这种方法，成本低廉，软件可自行更改，缺点：频率计的性能受限于单片机的外设和性能，单片机的主频往往不是很快。

(3) FPGA/CPLD。这种方法，利用硬件并行执行的优点，加上 CPLD/FPGA 能达到很快的主频，所以性能优越、执行速度快、成本合适。

本节将讲述频率计的原理和各种测频率方法的优/缺点。

频率计的基本原理就是在预先设定的时间内（如 0.5s），通过计数器对输入信号进行计数，之后通过运算算出待测的频率。不同的测频率的方法对于闸门时间和计数器的工作方法有着不同的定义，也就造就了不同测频方法的优、缺点。



常见的测频法有直接测频法、同步测频法和测周期法。这三种方法对于时间基准和计数基准的定义不尽相同。

1.1.1 直接测频法

直接测频法：在闸门开启时间（如 1s）内，测量输入频率的计数值设为 N ，从而得到测量信号的频率是 $f=N$ ，如图 1.1 所示。在 1s 时间内，若计数是 7，则待测信号是 7Hz。

测脉冲个数法就是在规定的定时时间内（可选择 1、100ms 等间隔时间）获取脉冲个数，之后根据以下公式算出频率

$$f = \frac{n}{T} \tag{1.1}$$

式中 n ——脉冲个数；

T ——时间间隔。

图 1.1 是理想情况，很多实际情况中由于闸门开通和关闭的时间与被测信号的跳变不一定同步，如图 1.2 所示。若闸门关闭的时间与频率跳变不同步，处于频率变化的中间时刻（也就是被测信号不同步），则会产生误差，且在低频段内测量的误差较大。

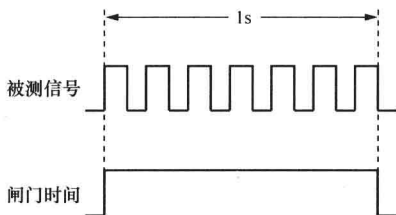


图 1.1 直接测频法原理框图

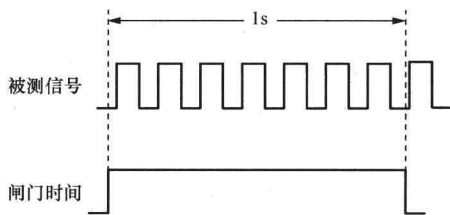


图 1.2 直接测频法误差示意图

1.1.2 同步测频法

综合上述两种测频方法的效果，可以得出这样的结论：高频信号用直接测频法，低频信号用测周期法。但实际应用中的信号是不定的，那么有没有一种通用、适应范围广泛的测频方法呢？有，那就是同步测频法。

同步测频法，其精髓就是和被测信号同步，同步的好处就是能尽可能地“贴近”被测信号，从而能将误差大大减小，且对于被测信号的频率没有“盲区”。

如图 1.3 所示，当预置闸门时间到来的同时，被测信号却不是上升沿，于是，计数器就不应该开始时间计数，而应该等待被测信号上升沿到来后再开始计数，即是实际闸门开启时间。经过预先设定好的脉冲计数之后，到达预置闸门时间附近，而此时不一定是被测信号的下降沿，也就不应该停止时间计数，而应该等到被测信号下降沿到来后才能停止，即是实际闸门的关闭时间。

通过这种开启和关闭的同步可以消除对被测信号计数的误差，可以计数到整数被测信号的个数，但对标准时间的计数会有 ± 1 的计数误差。由于标准时间是一个非常快速的信号（通常用几十兆到几百兆），所以一个标准时间周期的误差很小。相比直接测频法和测周期法而言，同步测频法的误差缩小很多。

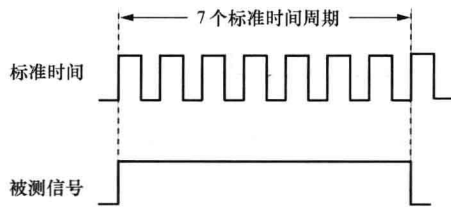


图 1.3 测周期法原理示意图

这种方法需要设计两个计数器，不仅用来对被测信号进行计数，还用于对标准信号进行计数。

1.1.3 测周期法

测周期法就是采用定时器和计数器，当第一个脉冲被捕获时，立即开始定时器计时，当第 N 个脉冲被捕获时 (N 可以自己决定，一般选取 1、2、4 等数值)，立即停止定时器，并查看这个定时点之间的时间间隔，用下式来算出频率。

$$f = \frac{N}{t_2 - t_1} \quad (1.2)$$

测周期法原理示意图如图 1.4 所示。图 1.4 表示的是捕获一个脉冲所计算的周期，比如是 7 个时钟周期，则最后算出的频率

$$f = 1 / (7 \times t) \quad (1.3)$$

式中 t ——标准信号的时钟周期。

图 1.4 中的测周期法是理想情况，而实际情况是：被测信号的下降沿到来时，并不一定正好是标准时间的整数个周期（也就是标准时间不同步），就会产生标准时间计数的误差，如图 1.5 所示。所以，测周期法对于高频信号误差较大。

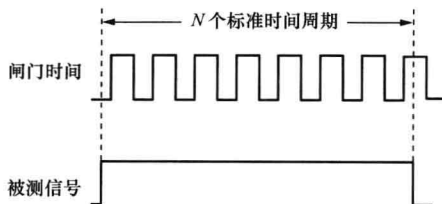


图 1.4 测周期法误差示意图

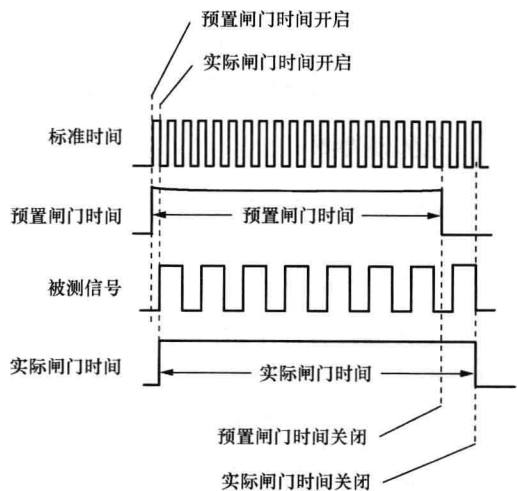


图 1.5 同步测频法原理示意图

1.2 原理构架

上一节介绍了几种测频的原理。本节将介绍如何在 FPGA 上将理论变为现实。

使用 FPGA 的好处：可以将几乎所有的数字电子集成到 FPGA，并且是可编辑、可改变的。在 FPGA 的外围仅仅需要少量的模拟电路、功率电路即可，这样的设计可大大简化整个电路结构，精简电路。

本实例的总体设计框图如图 1.6 所示。

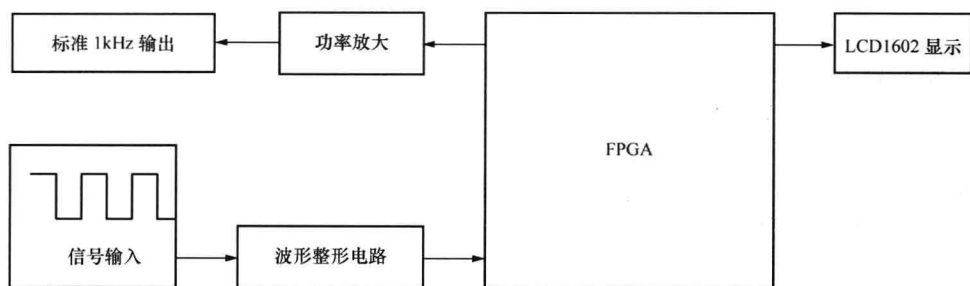


图 1.6 总体设计框图

图 1.6 所示中的左下方是信号输入模块，经过波形整形电路之后，输入 FPGA，在 FPGA 中设计测频器（可以运用不同的测频原理来设计），测出频率之后，再送到外围的液晶显示。另外，利用 FPGA 内设计的分频模块还可以实现不同频率方波的输出，从而在本系统内产生信号，供别的系统测试，或者供本系统自测。

1.3 硬件组成

本实例采用 Altera 公司的 Cyclone III 系列 EP2C8 作为硬件平台。下面先介绍总体硬件组成框架，并逐一介绍各模块的硬件组成。

1.3.1 总体硬件图

FPGA 频率计总体硬件图如图 1.7 所示。

1.3.2 FPGA 硬件设计

在 FPGA 硬件设计中留出脉冲输入（和波形整形电路连接）、LCD 控制（和 LCD1602 连接）和脉冲输出（和功率放大电路连接）。由于 FPGA 管脚多，不能全部显示，这里仅展示与本实例有关的管脚，如图 1.8 所示。

1.3.3 前端调理电路设计

外部输入的信号往往夹杂了很多干扰信号。可能会有毛刺，影响测频结果。所以，在信号输入单片机之前，需要进行简单的处理。