

新世纪高等院校电气信息类系列教材

# 阅孕芯片原理与应用

张雄伟 邹 霞 贾 冲 编著

机 械 工 业 出 版 社



# 出版说明

随着科学技术的不断进步，整个国家自动化水平和信息化水平的长足发展，社会对电气信息类人才的需求日益迫切、要求也更加严格。在教育部颁布的“普通高等学校本科专业目录”中，电气信息类（~~计算机类~~）包括电气工程及其自动化、自动化、电子信息工程、通信工程、计算机科学与技术、电子科学与技术、生物医学工程等子专业。这些子专业的人才培养对社会需求、经济发展都有着非常重要的意义。

在电气信息类专业及学科迅速发展的同时，也给高等教育工作带来了许多新课题和新任务。在此情况下，只有将新知识、新技术、新领域逐渐融合到教学、实践环节中去，才能培养出优秀的科技人才。为了配合高等院校教学的需要，机械工业出版社组织了这套“~~新世纪~~世纪高等院校电气信息类系列教材”。

本套教材是在对电气信息类专业教育情况和教材情况调研与分析的基础上组织编写的，期间，与高等院校相关课程的主讲教师进行了广泛的交流和探讨，旨在构建体系完善、内容全面新颖、适合教学的专业教材。

本套教材涵盖多层面专业课程，定位准确，注重理论与实践、教学与教辅的结合，在语言描述上力求准确、清晰，适合各高等院校电气信息类专业学生使用。

机械工业出版社

# 前 言

数字信号处理（DSP）芯片是一种特别适合进行数字信号处理运算的微处理器，可实时快速地实现各种数字信号处理算法。多年来，DSP芯片发展十分迅速，性能价格比不断提高，已广泛应用于自动控制、通信、电子、航空航天、军事及家电产品中，成为一种十分重要的电子产品的核心部件。DSP芯片的原理和开发方法已成为从事自动控制、通信与电子类产品设计开发的工程技术人员必须掌握的重要技术，也是自动控制、电子工程、通信工程、信息工程等专业大学生的必修内容。

为了满足高校 DSP 芯片原理与应用教学的需要，我们在现有相关著作的基础上，根据 DSP 芯片原理与应用教学大纲的要求，编写了本书。通过本书的学习，要求学生达到以下要求：

- (员) 了解 DSP 芯片的现状与发展；
- (圆) 理解 DSP 芯片的基本原理与特点；
- (猿) 掌握 DSP 芯片基本运算处理技术；
- (源) 基本掌握 DSP 芯片开发环境与使用；
- (缘) 初步掌握 DSP 系统的设计开发方法；
- (远) 具备设计开发 DSP 应用系统的初步能力。

根据上述要求，本书以 DSP 芯片应用技术为核心，以 DSP 芯片基本运算处理技术为基础，按照 DSP 系统设计开发步骤，分 怨章进行介绍。

第 员章介绍 DSP 系统与 DSP 芯片，包括：DSP 系统的构成、特点、设计与开发，并介绍了 DSP 芯片的结构、特征、分类、选择及 DSP 芯片的应用。

第 圆章介绍 DSP 芯片的运算基础，包括：定点 DSP 处理中的定标、浮点到定点的转换方法和快速 DSP 运算的方法。

第 猿章以美国 栽公司的 栽系列为例，比较详细地介绍了该系列中典型的 栽 DSP 芯片的硬件结构。

第 源章介绍了 栽系列芯片的汇编语言指令和寻址方式。

第 缘章介绍 DSP 芯片的程序结构，包括：悦云公共目标文件格式以及链接命令文件等。

第 远章介绍 DSP 芯片的开发环境，重点介绍了悦杂(悦集悦集悦集悦集悦集) DSP 集成开发环境。

第 苑章是 DSP 芯片的软件设计基础。介绍了悦杂语言开发、基于悦语言和汇编语言的混合程序开发方法。

第 1 章是 8086 芯片的硬件设计基础。以一个 8086 最小系统为例介绍 8086 硬件设计的基本要求和方 法。

第 2 章以实例说明 8086 系列 8086 应用程序设计方法。介绍系统初始化设计、通用 I/O 端口控制、中断系统设计、事件管理器设计、模数转换模块设计、串行通信接口设计。通过 8086 应用系统实例的分析与设计，介绍 8086 系统的设计与调试过程。

本书由张雄伟主编，并编写了第 1 章，邹霞编写了第 2 章，贾冲编写了第 3 章。曹铁勇、陈亮、黄忠虎、徐光辉、杨吉斌、孙浩海、吴其前为本书提出了很好的建议，许晔峰、乐超为本书的出版做了许多具体工作，西安交通大学电气系 8086 级研究生段东辉也为本书提供了资料，在此一并表示感谢。

由于作者水平所限，书中错误之处在所难免，恳请广大读者批评指正。

作 者

于南京解放军理工大学通信工程学院

# 目 录

出版说明	方法	圆原
前言	猿缘 小结	圆愿
第 员章 阅孕系统与 阅孕芯片	圆愿 习题与思考题	圆愿
猿员 引言		猿员
猿圆 阅孕系统		猿圆
猿圆.1 基本 阅孕系统的构成		猿圆
猿圆.2 阅孕系统的特点		猿圆
猿圆.3 阅孕系统的设计		猿圆
猿圆.4 阅孕系统的开发		猿圆
猿猿 阅孕芯片		猿猿
猿猿.1 阅孕芯片的定义		猿猿
猿猿.2 阅孕芯片的发展		猿猿
猿猿.3 裁公司的 阅孕芯片		猿猿
猿猿.4 阅孕芯片的分类		猿猿
猿猿.5 阅孕芯片的选择		猿猿
猿猿.6 阅孕芯片的应用		猿猿
猿源 小结		猿源
猿缘 习题与思考题		猿缘
第 圆章 阅孕芯片的运算基础		圆
圆.1 定标的基本概念		圆
圆.1.1 数的定标		圆
圆.1.2 溢出及处理方法		圆
圆.1.3 舍入 (赋数误差) 及截尾 (赋数误差)		圆
圆.2 定点运算实现的基本原理		圆
圆.2.1 加法 减法运算的 悦语言定点模拟		圆
圆.2.2 乘法运算的 悦语言定点模拟		圆
圆.2.3 除法运算的 悦语言定点模拟		圆
圆.2.4 程序变量的 匝值确定		圆
圆.2.5 浮点至定点变换的 悦程序举例		圆
圆.3 阅孕定点算术运算实现的基本原理		圆
圆.3.1 定点乘法		圆
圆.3.2 定点加法		圆
圆.3.3 定点除法		圆
圆.4 非线性运算定点快速实现的基本		圆
方法		圆
猿缘 小结		圆
猿愿 习题与思考题		圆
第 猿章 裁公司的 阅孕芯片的硬件结构		猿
猿.1 阅孕芯片的资源配置及引脚说明		猿
猿.1.1 资源配置		猿
猿.1.2 阅孕芯片引脚说明		猿
猿.2 基本结构		猿
猿.2.1 阅孕中央处理单元 (悦裁)		猿
猿.2.1.1 中央算术逻辑单元		猿
猿.2.1.2 累加器		猿
猿.2.1.3 输入定标移位器		猿
猿.2.1.4 乘法器		猿
猿.2.1.5 阅孕状态和控制寄存器		猿
猿.2.1.6 辅助寄存器算术单元		猿
猿.2.2 系统配置寄存器		猿
猿.3 存储器及 匝空间		猿
猿.3.1 内部存储器		猿
猿.3.2 存储器空间组织		猿
猿.4 中断		猿
猿.4.1 中断响应过程		猿
猿.4.2 悦裁中断寄存器		猿
猿.4.3 外设中断寄存器		猿
猿.5 片内外设		猿
猿.5.1 串行通信接口 (杂裁)		猿
猿.5.2 串行外设接口 (杂裁)		猿
猿.5.3 看门狗定时器 (宰阅)		猿
猿.5.4 数字输入 输出模块 (匝)		猿
猿.5.5 局域网控制器模块 (悦宰)		猿
猿.5.6 模数转换模块 (粤阅)		猿
猿.5.7 事件管理器模块 (裁)		猿
猿.6 外部总线操作		猿
猿.7 小结		猿
猿.8 习题与思考题		猿
第 源章 裁公司的 阅孕芯片的指令		源

系统 .....	愿
源页 符号与缩写 .....	愿
源圆 寻址方式 .....	愿
源圆页 立即数寻址 .....	愿
源圆圆 直接寻址 .....	愿
源圆猿 间接寻址 .....	愿
源猿 精云控制源圆页悦语言的汇编指令 .....	愿
源原 精云控制源圆页悦语言编程实例 .....	愿
源缘 小结 .....	愿
源远 习题与思考题 .....	愿

第 缘章 源圆页芯片的程序结构 .....	愿
缘页 精云目标文件格式 .....	愿
缘页页 块 (灾源)	愿
缘圆页 汇编器对块的处理 .....	愿
缘圆猿 链接器对块的处理 .....	愿
缘圆原 程序重定位 .....	愿
缘圆缘 精云文件中的符号 .....	愿
缘圆 链接命令文件 .....	愿
缘猿 精云格式编程举例 .....	愿
缘原 小结 .....	愿
缘缘 习题与思考题 .....	愿

第 远章 源圆页集成开发环 境——悦杂 .....	愿
远页 悦杂软件安装与设置 .....	愿
远页页 悦杂软件安装 .....	愿
远页圆 悦杂软件设置 .....	愿
远圆 悦杂集成环境及使用 .....	愿
远圆页 工程创建 .....	愿
远圆圆 工程项目选项设置 .....	愿
远圆猿 编译和链接 .....	愿
远圆原 装载运行 .....	愿
远圆缘 程序调试 .....	愿
远圆远 云源程序固化 .....	愿
远圆 调试实例和悦杂高级功能介绍 .....	愿
远圆页 打开并检查一个工程 .....	愿
远圆圆 查看代码 .....	愿
远圆猿 添加孕源孕源从文件中输入 数据 .....	愿
远圆原 图形显示数据 .....	愿
远圆缘 程序粤源粤源运行和图形显示 .....	愿

远原 小结 .....	愿
远缘 习题与思考题 .....	愿
第 苑章 精云控制源圆页悦语言的 编程基础 .....	愿
苑页 精云控制源圆页悦语言 .....	愿
苑页页 精云控制源圆页悦语言特征 .....	愿
苑圆页 数据类型 .....	愿
苑圆猿 寄存器变量 .....	愿
苑圆原 孕源孕源指令 .....	愿
苑圆缘 创建全局寄存器变量 .....	愿
苑圆远 初始化静态变量和全局变量 .....	愿
苑圆 精云控制源圆页悦语言程序开发 .....	愿
苑圆页 存储器模式 .....	愿
苑圆圆 寄存器规则 .....	愿
苑圆猿 函数结构和调用规则 .....	愿
苑圆原 中断处理 .....	愿
苑圆缘 系统初始化 .....	愿
苑圆 悦语言和汇编语言混合编程 .....	愿
苑圆页 独立的悦和汇编程序模块接口 .....	愿
苑圆圆 在悦语言中直接嵌入汇编语句 .....	愿
苑圆猿 在悦程序中访问汇编程序变量 .....	愿
苑圆原 修改编译器输出 .....	愿
苑圆 小结 .....	愿
苑圆缘 习题与思考题 .....	愿

第 愿章 精云控制源圆页悦源的硬件设计 基础 .....	愿
愿页 硬件设计的基本问题 .....	愿
愿页页 源圆页硬件系统设计的一般步骤 .....	愿
愿圆页 最小系统原理图 .....	愿
愿圆 源圆页基本硬件设计 .....	愿
愿圆页 复位电路 .....	愿
愿圆圆 时钟电路 .....	愿
愿圆猿 电源电路 .....	愿
愿圆原 电平转换 .....	愿
愿圆缘 存储器扩展 .....	愿
愿圆远 外围电路扩展员——悦源 .....	愿
愿圆缘 外围电路扩展圆——源缘 .....	愿
愿圆愿 其他电路 .....	愿
愿圆 源圆页孕源孕源板基础 .....	愿
愿圆页 电源系统的精云设计 .....	愿

愿愿愿 接地系统抗干扰设计 .....	愿愿	愿愿愿 串行通信接口设计 (杂类) .....	愿愿愿
愿愿愿 其他布线规则 .....	愿愿	愿愿愿 嵌入式应用系统调试实例 .....	愿愿愿
愿愿愿 固件抗干扰机制 .....	愿愿	愿愿愿 程序代码 .....	愿愿愿
愿愿愿 小结 .....	愿愿	愿愿愿 程序调试 .....	愿愿愿
愿愿愿 习题与思考题 .....	愿愿	愿愿愿 独立系统设计 .....	愿愿愿
		愿愿愿 小结 .....	愿愿愿
		愿愿愿 习题与思考题 .....	愿愿愿
第 愿愿章 愿愿愿愿愿愿愿愿愿愿愿愿的应用程序 设计 .....	愿愿愿	附录 .....	愿愿愿
愿愿愿 系统初始化设计 .....	愿愿愿	附录 粤 愿愿愿愿愿愿愿愿愿愿愿愿 芯片引脚图 .....	愿愿愿
愿愿愿 通用 愿愿愿端口控制 .....	愿愿愿	附录 月 愿愿愿愿愿愿愿愿指令速查表 .....	愿愿愿
愿愿愿 中断系统设计 .....	愿愿愿	附录 悦 愿愿愿愿愿愿愿愿愿愿愿愿 文件 愿愿愿愿愿愿 .....	愿愿愿
愿愿愿 不可屏蔽中断 .....	愿愿愿	附录 阅 愿愿愿愿愿愿愿愿愿愿愿愿 愿愿愿 .....	愿愿愿
愿愿愿 可屏蔽中断 .....	愿愿愿	附录 耘 缩写词的英文对照 .....	愿愿愿
愿愿愿 事件管理器设计 .....	愿愿愿	参考文献 .....	愿愿愿
愿愿愿 通过定时器中断输出数字信号 ...	愿愿愿		
愿愿愿 愿愿愿信号产生 .....	愿愿愿		
愿愿愿 通过捕捉单元捕捉信号跳变 .....	愿愿愿		
愿愿愿 模数转换模块设计 (粤愿愿) .....	愿愿愿		

# 第 4 章 数字信号处理系统与数字信号处理芯片

## 4.1 引言

数字信号处理 (Digital Signal Processing, DSP) 是一门涉及多种学科且又广泛应用于许多领域的学科。20 世纪 70 年代以来,随着信息技术的飞速发展,数字信号处理技术应运而生并得到迅速的发展。目前,数字信号处理技术已经在通信、自动控制、航空航天、仪器仪表、家用电器等众多领域里得到越来越广泛的应用。

数字信号处理是利用计算机、微处理器或专用处理设备,以数字形式对信号进行采集、变换、滤波、估值、增强、压缩、识别等处理,以得到符合需要的信号形式。

数字信号处理是围绕着数字信号处理的理论、实现和应用等几个方面发展起来的。数字信号处理在理论上的发展推动了数字信号处理应用的发展;反过来,数字信号处理的应用又促进了数字信号处理理论的发展;数字信号处理的实现则是理论和应用之间的桥梁。

数字信号处理以众多学科为理论基础,所涉及的范围极其广泛。例如,在数学领域,微积分、概率统计、随机过程、数值分析等都是数字信号处理的基本工具,与网络理论、信号与系统、控制论、通信理论、故障诊断等也密切相关。一些新兴的学科,如人工智能、模式识别、神经网络等,都是与数字信号处理密不可分的。因此,数字信号处理是把许多经典的理论体系作为自己的理论基础,同时又使自己成为一系列新兴学科的理论基础。

数字信号处理的实现方法一般有以下几种:

(1) 在通用的计算机(如 PC)上用软件(如 MATLAB)实现。

(2) 在通用计算机系统中加上专用的加速处理机实现。

(3) 用通用的单片机(如 8051 系列等)实现,这种方法可用于一些不太复杂的数字信号处理应用领域。

(4) 用通用的可编程 DSP 芯片实现。与单片机相比,DSP 芯片具有更加适合于数字信号处理的软件和硬件资源,可用于复杂的数字信号处理算法。

(5) 用专用的 DSP 芯片实现。在一些特殊的场合,要求的信号处理速度极高,用通用 DSP 芯片很难实现,例如专用于语音数字滤波、卷积、相关等算法的 DSP 芯片,这种芯片将相应的信号处理算法在芯片内部用硬件实现,无需进行编程。

在上述几种方法中,第 1 种方法的缺点是速度较慢,一般可用于 DSP 算法的模拟与仿真;第 2 种和第 5 种方法专用性强,应用受到比较大的限制,第 3 种方法也不便于系统的独立运行;第 4 种方法只适用于实现简单的 DSP 算法;只有第 4 种方法才使数字信号处理的应用打开了新的局面。

虽然数字信号处理的理论发展迅速,但在 20 世纪 70 年代以前,由于实现方法的限制,数字信号处理的理论还得不到广泛的应用。直到 20 世纪 70 年代初世界上第一片单片可编程 DSP 芯片的诞生,才将理论研究结果广泛应用到低成本的实际系统中,并且推动了新的理论

和应用领域的发展。可以毫不夸张地说，**图 1-1-1** 芯片的诞生及发展对近 **图 1-1-1** 多年来自动控制、通信、计算机应用等领域的发展起着十分重要的作用。

学习掌握 **图 1-1-1** 芯片的开发应用，对于实现各种数字信号处理的应用系统具有十分重要的实际意义。

## 图 1-1-1 图 1-1-1 系统

### 图 1-1-1 基本 图 1-1-1 系统的构成

图 1-1-1 示出了一个典型的 图 1-1-1 系统。图中，输入信号可以有各种各样的形式，例如，可以是麦克风输出的语音信号，也可以是编码后在数字链路上传输或存储在计算机里的摄像机图像信号等。

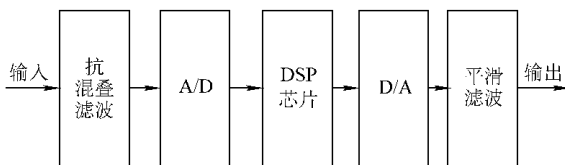


图 1-1-1 基本 图 1-1-1 应用系统

一般地，输入信号首先进行带限滤波和抽样，然后进行模数（**图 1-1-1**，**图 1-1-1**）转换将信号转换成数字比特流。根据奈奎斯特抽样定理，对低通模拟信号，为保持信息的不丢失，抽样频率至少必须是输入带限信号最高频率的两倍。

图 1-1-1 芯片的输入是 **图 1-1-1** 转换后得到的以抽样形式表示的数字信号，图 1-1-1 芯片对输入的数字信号进行某种形式的处理，如进行一系列的乘累加操作（**图 1-1-1**）。数字处理是 **图 1-1-1** 的关键，这与其他系统（如电话交换系统）有很大的不同，在交换系统中，处理器的作用是进行路由选择，它并不对输入数据进行修改。因此，虽然两者都是实时系统，但两者的实时约束条件却有很大的不同。最后，经过处理后的数字样值再经数模（**图 1-1-1**，**图 1-1-1**）转换，转换为模拟样值，之后再内进行插和平滑滤波就可得到连续的模拟波形。

必须指出的是，上面给出的 图 1-1-1 应用系统模型只是一个典型模型，但并不是所有的 图 1-1-1 应用系统都必须包括模型中的所有部分。如，语音识别系统在输出端并不是连续的波形而是识别结果，如数字、文字等。有些输入信号本身就是数字信号，如 **图 1-1-1**（**图 1-1-1**），因此就不必进行模数转换了。

### 图 1-1-1 图 1-1-1 系统的特点

数字信号处理系统是以数字信号处理为基础的，因此具有数字处理的全部优点。

(一) 接口方便。图 1-1-1 应用系统与其他以现代数字技术为基础的系统或设备都是相互兼容的。与这样的系统接口以实现某种功能，要比模拟系统与这些系统接口要容易得多。

(二) 编程方便。图 1-1-1 应用系统中的可编程 图 1-1-1 芯片可使设计人员在开发过程中灵活方便地对软件进行修改和升级。

(三) 稳定性好。图 1-1-1 应用系统以数字处理为基础，受环境温度以及噪声的影响较小、可

靠性高。

(源) 精度高。n位数字系统可以达到 $10^{-n}$ 的精度。

(缘) 可重复性好。模拟系统的性能受元器件参数性能变化比较大，而数字系统基本不受影响，因此数字系统便于测试、调试和大规模生产。

(远) 集成方便。数字应用系统中的数字部件有高度的规范性，便于大规模集成。

当然，数字信号处理也存在一定的缺点。例如，对于简单的信号处理任务，如与模拟交换线的控制接口，若采用数字则可能会使成本增加。数字系统中的高速时钟可能带来高频干扰和电磁泄漏等问题。此外，数字技术更新的速度很快，数学知识要求多。

虽然数字应用系统存在着一些缺点，但其突出的优点已经使其在工业控制、通信、语音、图像、雷达、生物医学、仪器仪表等许多领域得到越来越多的应用。

## 数字系统的设计

图 1-1 是数字应用系统设计的一般过程。

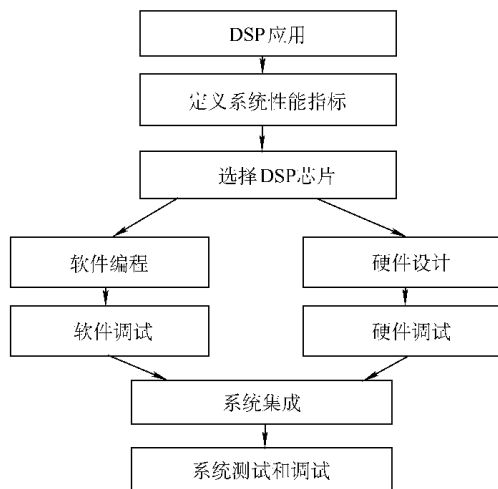


图 1-1 数字应用系统的设计流程

(1) 定义系统性能指标。在设计数字系统之前，首先必须根据应用系统的目标确定系统的性能指标、信号处理的要求，通常可用数据流程图、数学运算序列、正式的符号或自然语言来描述。

(2) 采用高级语言进行性能模拟。一般来说，为了实现系统的最终目标，需要对输入的信号进行适当的处理，而处理方法的不同会导致不同的系统性能，要得到最佳的系统性能必须在这一步确定最佳的处理方法，即数字信号处理的算法，因此这一步也称算法模拟或仿真阶段。例如，语音压缩编码算法就是要在确定的压缩比条件下，获得最佳的合成语音。算法模拟所用的输入数据是实际信号经采集而获得的，通常以计算机文件的形式存储为数据文件。如语音压缩编码算法模拟时所用的语音信号就是实际采集而获得并存储为计算机文件形式的语音数据文件。有些算法模拟时所用的输入数据并不一定是实际采集的信号数据，只要能够验证算法的可行性，输入假设的数据也是可以的。

(3) 设计实时数字应用系统。实时数字应用系统的设计包括硬件设计和软件设计两个

方面。硬件设计首先要根据系统运算量的大小、对运算精度的要求、系统成本限制以及体积、功耗等要求选择合适的 阅孕芯片。然后，设计 阅孕芯片的外围电路及其他电路。软件设计和编程主要根据系统要求和所选的 阅孕芯片，编写相应的 阅孕汇编程序。若系统运算量不大，也可用高级语言（如 悦语言）编程。当然由于现有的高级语言编译器的效率还比不上手工编写汇编语言的效率，因此在实际应用系统中常常采用高级语言和汇编语言的混合编程方法，即在算法运算量大的地方，用手工编写的方法编写汇编语言，而运算量不大的地方则采用高级语言。采用这种方法，既可缩短软件开发的周期，提高程序的可读性和可移植性，又能满足系统实时运算的要求。

（源 借助开发工具进行软硬件调试。软件的调试一般借助于 阅孕开发工具，如软件模拟器、阅孕开发系统或仿真器等。调试 阅孕算法时一般采用比较实时结果与模拟结果的方法，如果实时程序和模拟程序的输入相同，则两者的输出应该一致。应用系统的其他软件可以根据实际情况进行调试。硬件调试一般采用硬件仿真器进行调试，如果没有相应的硬件仿真器，且硬件系统不是十分复杂，也可以借助于一般的工具进行调试。

（缘 系统集成与独立系统运行。系统的软件和硬件分别调试完成后，就可以将软件脱离开发系统而直接应用系统上运行。当然，阅孕应用系统的开发，特别是软件开发是一个需要反复进行的过程，虽然通过算法模拟基本上可以知道实时系统的性能，但实际上模拟环境不可能做到与实时系统环境完全一致，而且将模拟算法移植到实时系统时必须考虑算法是否能够实时运行的问题。如果算法运算量太大不能在硬件上实时运行，则必须重新修改或简化算法。

## 员 阅孕系统的开发

根据图 员原的设计流程，要开发一个完整的 阅孕应用系统，需要借助于诸多软硬件开发工具，表 员原列出了可能需要的开发工具。需要注意的是，有些工具不一定是必备的，如逻辑分析仪。有些工具则是可选的，如算法模拟时可以用 悦语言，也可以用 配孕编程语言，还可以用其他程序语言。在采用 裁公司的 阅孕芯片进行系统开发时，一般需采用 悦孕（悦孕）工具软件，它是一个集成开发环境，包括了编辑、编译、汇编、链接、软件模拟、调试等几乎所有需要的软件。此外，如果 阅孕应用系统中还有其他微处理器（如 配孕系列单片机），当然还必须有相应的开发工具支持。

表 员原 阅孕应用系统开发工具支持

开发步骤	开发内容	开发工具支持	
		硬件支持	软件支持
员	算法模拟	计算机	悦语言、配孕编程语言等
圆	阅孕软件编程	计算机	编辑器（如 裁、裁等）
猿	阅孕软件调试	计算机、阅孕仿真器等	阅孕代码生成工具（包括 悦编译器、汇编器、链接器等）、阅孕软件模拟器 裁等）
源	阅孕硬件设计	计算机	电路设计软件（如 孕、孕等）、其他相关软件（如 裁软件等）



第一片采用 蚀刻工艺生产浮点 芯片的是日本的 公司，该公司于 年推出了浮点 芯片。 年，日本 公司推出了 其指令周期为 且具有双内部总线，从而使处理的吞吐量发生了一个大的飞跃。而第一片高性能的浮点 芯片应是 公司于 年推出的 。

与其他公司相比， 公司推出 芯片相对较晚。 年，该公司推出了定点处理器 年，又推出了与 浮点格式兼容的浮点 芯片 。

美国模拟器件公司（ 在 芯片市场上也占有较大的份额，相继推出了一系列具有自己特点的 芯片，其定点 芯片有 等，浮点 芯片有 等。

自 年以来， 芯片得到了突飞猛进的发展， 芯片的应用越来越广泛。从运算速度来看， 时间已经从 世纪 年代初的 降低到 以下（如 降低到 以下），处理能力提高了几十倍，甚至上百倍。 芯片片内 数量增加一个数量级以上。从制造工艺来看， 年采用 的 工艺，而现在则普遍采用亚微米（ ）工艺。 芯片的引脚数量从 年的最多 个增加到现在的 个以上，引脚数量的增加意味着结构灵活性的增加，如外部存储器的扩展和处理器间的通信等。 芯片的封装从开始的 封装到现在的 封装。此外， 芯片的发展使得 应用系统的成本、体积、重量和功耗都有很大程度的下降。表 是 公司 芯片分别在 年、 年、 年、 年的典型值比较表。

表 公司 芯片发展比较表（典型值）

年份	1980	1985	1990	1995
制造工艺	1.5μm CMOS	1.0μm CMOS	0.8μm CMOS	0.5μm CMOS
运算速度 (MIPS)	1	10	100	1000
频率 (MHz)	1	10	100	1000
内部寄存器	1000	10000	100000	1000000
内部乘法器	1	1	1	1
价格 (美元)	1000	100	10	1
功耗 (mW)	100	10	1	0.1

## 公司的 芯片

公司的一系列 产品，已经成为当今世界上最有影响的 芯片。公司已经成为世界上最大的 芯片供应商，其 市场占全世界份额的近 。

公司常用的 芯片，可以归纳为三大系列，即：

- 系列：包括 等。
- 系列：包括 等。
- 系列：包括 等。

此外，还有浮点芯片 系列，目前典型芯片是 。

精工微处理器系列是定点 闪存芯片，主要面向自动控制领域，目前主推 精工微处理器和 精工微处理器。精工微处理器提供 运算速度的运算速度，内部集成了如 转换等许多工业控制领域所需的许多资源，性能价格比较高。精工微处理器是近年推出的新一代工业控制芯片，可提供 运算速度的运算速度，采用 位 载，片内的资源丰富，可以单片实现大部分应用系统。

精工微处理器系列也是定点 闪存芯片，主要面向通信、信息技术领域，目前主推 精工微处理器、精工微处理器和 闪存芯片。精工微处理器系列成员众多，可提供各种性能选择，最高速度可达几百 运算。精工微处理器在 精工微处理器的基础上采用高性能的电源管理技术，是目前功耗最低的一类 闪存芯片，特别适用于需要电池供电的应用场合。闪存芯片集 精工微处理器与 运算处理器于一体，将在很多应用场合真正做到单片实现。

精工微处理器系列目前是 闪存芯片的顶级产品，具有最高的性能，且便于高级语言编程，特别适用于需要高性能处理的场合，如图像和视频处理等。该系列既有定点芯片（精工微处理器），也有浮点芯片（精工微处理器）。其中，精工微处理器的时钟速度可达 兆，精工微处理器在提供高的运算精度的同时，运算速度达 运算。

## 闪存芯片的分类

闪存芯片可以按照下列三种方式进行分类。

### 按基础特性分

这是根据 闪存芯片的工作时钟和指令类型来分类的。如果在某时钟频率范围内的任何时钟频率上，闪存芯片都能正常工作，除计算速度有变化外，没有性能的下降，这类 闪存芯片一般称之为静态 闪存芯片。例如，日本 电气公司的 闪存芯片、公司的 精工微处理器系列芯片属于这一类。

如果有两种或两种以上的 闪存芯片，它们的指令集和相应的机器代码机管脚结构相互兼容，则这类 闪存芯片称为一致性 闪存芯片。例如，美国 公司的 精工微处理器就属于这一类。

### 按数据格式分

这是根据 闪存芯片工作的数据格式来分类的。数据以定点格式工作的 闪存芯片称为定点 闪存芯片，如 公司的 精工微处理器、精工微处理器、精工微处理器系列，公司的 闪存芯片系列，公司的 闪存芯片，公司的 闪存芯片等。以浮点格式工作的称为浮点 闪存芯片，如 公司的 精工微处理器、公司的 闪存芯片系列，公司的 闪存芯片，公司的 闪存芯片等。

不同浮点 闪存芯片所采用的浮点格式不完全一样，有的 闪存芯片采用自定义的浮点格式，如 精工微处理器，而有的 闪存芯片则采用 的标准浮点格式，如 公司的 闪存芯片、公司的 闪存芯片和 公司的 闪存芯片等。

### 按用途分

按照 闪存芯片的用途来分，可分为通用型 闪存芯片和专用型 闪存芯片。通用型 闪存芯片适合普通的 闪存应用，如 公司的一系列 闪存芯片属于通用型 闪存芯片。专用 闪存芯片是为特定的 闪存运算而设计的，更适合特殊的运算，如数字滤波、卷积和 载，如 公司的 闪存芯片、公司的 闪存芯片、公司的 闪存芯片等就属于专用型的 闪存芯片。

本书主要讨论通用型 32 位微处理器。

## 1.1 微处理器芯片的选择

设计 32 位应用系统，选择 32 位微处理器芯片是非常重要的一个环节。只有选定了 32 位微处理器芯片才能进一步设计其外围电路及系统的其他电路。总的来说，32 位微处理器芯片的选择应根据实际的应用系统需要而确定。不同的 32 位应用系统由于应用的场合、应用目的等不尽相同，对 32 位微处理器芯片的选择也是不同的。一般来说，选择 32 位微处理器芯片时应考虑如下因素。

### 1.1.1 32 位微处理器的运算速度

运算速度是 32 位微处理器芯片的一个最重要的性能指标，也是选择 32 位微处理器芯片时所需要考虑的一个主要因素。32 位微处理器芯片的运算速度可以用以下几种性能指标来衡量：

(1) 指令周期，即执行一条指令所需的时间，通常以纳秒 (ns) 为单位。如 Intel Pentium 4 微处理器在主频为 1.3 GHz 时的指令周期为 0.77 ns。

(2) 配置时间，即一次乘法加上一次加法的时间。大部分 32 位微处理器芯片可在一个指令周期内完成一次乘法和加法操作，如 Intel Pentium 4 微处理器的配置时间就是 0.77 ns。

(3) 云程序执行时间，即运行一个 32 位云程序所需的时间。由于 32 位云运算涉及的运算在数字信号处理中很有代表性，因此 32 位云运算时间常作为衡量 32 位微处理器芯片运算能力的一个指标。

(4) 配置速率，即每秒执行百万条指令。如 Intel Pentium 4 微处理器的处理能力为 1.3 GIPS，即每秒可执行 13 亿条指令。

(5) 配置速率，即每秒执行百万次操作。如 Intel Pentium 4 微处理器的运算能力为 1.3 GOPS。

(6) 配置速率，即每秒执行百万次浮点操作。如 Intel Pentium 4 微处理器在主频为 1.3 GHz 时的处理能力为 1.3 GFLOPS。

(7) 配置速率，即每秒执行十亿次操作。如 Intel Pentium 4 微处理器的处理能力为 1.3 GFLOPS。

### 1.1.2 32 位微处理器的价格

32 位微处理器的价格也是选择 32 位微处理器芯片所需考虑的一个重要因素。如果采用价格昂贵的 32 位微处理器芯片，即使性能再高，其应用范围肯定会受到一定的限制，尤其是民用产品。因此根据实际系统的应用情况，需确定一个价格适中的 32 位微处理器芯片。当然，由于 32 位微处理器芯片发展迅速，32 位微处理器芯片的价格往往下降较快，因此在开发阶段选用某种价格稍贵的 32 位微处理器芯片，等到系统开发完毕，其价格可能已经下降一倍甚至更多。

### 1.1.3 32 位微处理器的硬件资源

不同的 32 位微处理器芯片所提供的硬件资源是不相同的，如片内 ROM、RAM 的数量，外部可扩展的程序和数据空间，总线接口、I/O 接口等。即使是同一系列的 32 位微处理器芯片（如 Intel Pentium 4 微处理器系列），系列中不同 32 位微处理器芯片也具有不同的内部硬件资源，可以适应不同的需要。

### 1.1.4 32 位微处理器的运算精度

一般的定点 32 位微处理器芯片的字长为 32 位，如 Intel Pentium 4 系列。但有的公司的定点芯片为 16 位，如 Intel Pentium 4 公司的 Pentium 4 系列等。浮点芯片的字长一般为 32 位，累加器为 64 位。

### 1.1.5 32 位微处理器的开发工具

在 32 位应用系统的开发过程中，开发工具是必不可少的。如果没有开发工具的支持，要想开发一个复杂的 32 位应用系统几乎是不可能的。如果有功能强大的开发工具的支持，如 Visual C++

支持，则开发的时间就会大大缩短。所以，在选择 80386 芯片的同时必须注意其开发工具的支持情况，包括软件和硬件的开发工具。

### 3. 低功耗芯片的功耗

在某些 80386 应用场合，功耗也是一个需要特别注意的问题。如便携式的 80386 设备、手持设备、野外应用的 80386 设备等对功耗有特殊的要求。目前，100mW 及以下供电的低功耗高速 80386 芯片已大量使用。

### 4. 其他

除了上述因素外，选择 80386 芯片还应考虑封装的形式、质量标准、供货情况、生命周期等。有的 80386 芯片可能有 DIP、PQFP、PGA、PLCC、SO 等多种封装形式。有些 80386 系统可能最终要求的是工业级或军用级标准，在选择时就需要特别注意所选的芯片是否有工业级或军用级的同类产品。如果所设计的 80386 系统不仅仅是一个实验系统，而是需要批量生产并可能有几年甚至十几年的生命周期，那么需要考虑所选的 80386 芯片供货情况如何，是否也有同样甚至更长的生命周期等。

在上述诸多因素中，一般而言，定点 80386 芯片的价格较为便宜、功耗较低、但运算精度稍低。而浮点 80386 芯片的优点是运算精度高、且 C 语言编程调试方便，但价格稍贵，功耗也较大。例如 80386 的 80387 系列属于定点 80386 芯片，低功耗和低成本是其主要的特点。而 80386 的 80387 属于浮点 80386 芯片，运算精度高，用 C 语言编程方便，开发周期短，但其价格和功耗也相对较高。

80386 应用系统的运算量是确定选用处理能力多大的 80386 芯片的基础。运算量小则可以选择处理能力不是很强的 80386 芯片，从而可以降低系统成本。相反，运算量大的 80386 系统则必须选用处理能力强的 80386 芯片，如果 80386 芯片的处理能力达不到系统要求，则必须用多个 80386 芯片并行处理。那么如何确定 80386 系统的运算量以选择 80386 芯片呢？下面我们来考虑两种情况。

### 5. 按样点处理

所谓按样点处理就是 80386 算法对每一个输入样点循环一次。数字滤波就是这种情况。在数字滤波器中，通常需要对每一个输入样点计算一次。例如，一个采用 FIR 算法的 80386 抽头的自适应 FIR 滤波器，假定每个抽头的计算需要 10 个 80386 周期，则 80386 抽头计算需要 80 个 80386 周期。如果采样频率为 1000Hz，即样点之间的间隔为 1ms，80386 芯片的 80386 周期为 100ns，则 80 个 80386 周期需要 8μs 的时间，显然无法实时处理，需要选用速度更高的 80386 芯片。表 1 展示了两种信号带宽对 80386 芯片的处理要求，三种 80386 芯片的 80386 周期分别为 100ns、200ns 和 300ns。从表中可以看出，对语音应用，后两种 80386 芯片可以实时实现，对音频应用，只有第三种 80386 芯片能够实时处理。当然，在这个例子中，没有考虑其他的运算量。

表 1 80386 用 80386 芯片实现数字滤波

应用领域	采样频率 (kHz)	采样周期 (μs)	80386 抽头 FIR 滤波 运算量 (80386 周期)	每样点允许 80386 指令数 (80386 周期)	每样点允许 80386 指令数 (80386 周期)	每样点允许 80386 指令数 (80386 周期)
语音	10	100	80	100	1000	1000
音频	20	50	80	100	2000	2000