

硬件工程师手册

华为技术有限公司

第一章 概述

第一节 硬件开发过程简介

§ 1.1.1 硬件开发的基本过程

产品硬件项目的开发，首先是要明确硬件总体需求情况，如CPU处理能力、存储容量及速度，I/O端口的分配、接口要求、电平要求、特殊电路（厚膜等）要求等等。其次，根据需求分析制定硬件总体方案，寻求关键器件及电咱的技术资料、技术途径、技术支持，要比较充分地考虑技术可能性、可靠性以及成本控制，并对开发调试工具提出明确的要求。关键器件索取样品。第三、总体方案确定后，作硬件和单板软件的详细设计，包括绘制硬件原理图、单板软件功能框图及编码、PCB布线，同时完成开发物料清单、新器件编码申请、物料申领。第四，领回PCB板及物料后由焊工焊好1~2块单板，作单板调试，对原理设计中的各功能进行调测，必要时修改原理图并作记录。第五，软硬件系统联调，一般的单板需硬件人员、单板软件人员的配合，特殊的单板（如主机板）需比较大型软件的开发，参与联调的软件人员更多。一般地，经过单板调试后在原理及PCB布线方面有些调整，需第二次投板。第六，内部验收及转中试，硬件项目完成开发过程。

§ 1.1.2 硬件开发的规范化

上节硬件开发的基本过程应遵循硬件开发流程规范文件执行，不仅如此，硬件开发涉及到技术的应用、器件的选择等，必须遵照相应的规范化措施才能达到质量保障的要求。这主要表现在，技术的采用要经过总体组的评审，器件和厂家的选择要参照物料认证部的相关文件，开发过程完成相应的规定文档，另外，常用的硬件电路（如ID.WDT）要采用通用的标准设计。

第二节 硬件工程师职责与基本技能

§ 1.2.1 硬件工程师职责

一个技术领先、运行可靠的硬件平台是公司产品质量的基础，硬件工程师职责神圣，责任重大。

- 1、硬件工程师应勇于尝试新的先进技术之应用，在产品硬件设计中大胆创新。

2、坚持采用开放式的硬件架构，把握硬件技术的主流和未来发展，在设计中考虑将来的技术升级。

3、充分利用公司现有的成熟技术，保持产品技术上的继承性。

4、在设计中考虑成本，控制产品的性能价格比达至最优。

5、技术开放，资源共享，促进公司整体的技术提升。

§ 1.2.1 硬件工程师基本素质与技术

硬件工程师应掌握如下基本技能：

第一、由需求分析至总体方案、详细设计的设计创造能力；

第二、熟练运用设计工具，设计原理图、EPLD、FPGA调试程序的能力；

第三、运用仿真设备、示波器、逻辑分析仪调测硬件的能力；

第四、掌握常用的标准电路的设计能力，如ID电路、WDT电路、 π 型滤波电路、高速信号传输线的匹配电路等；

第五、故障定位、解决问题的能力；

第六、文档的写作技能；

第七、接触供应商、保守公司机密的技能。

第二章 硬件开发规范化管理

第一节 硬件开发流程

§ 3.1.1 硬件开发流程文件介绍

在公司的规范化管理中，硬件开发的规范化是一项重要内容。硬件开发规范化管理是在公司的《硬件开发流程》及相关的《硬件开发文档规范》、《PCB投板流程》等文件中规划的。硬件开发流程是指导硬件工程师按规范化方式进行开发的准则，规范了硬件开发的全过程。硬件开发流程制定的目的是规范硬件开发过程控制，硬件开发质量，确保硬件开发能按预定目的完成。

公司硬件开发流程的文件编号为4/QM-RSD009，生效时间为1997年月21日。

硬件开发流程不但规范化了硬件开发的全过程，同时也从总体上，规定了硬件开发所应完成的任务。做为一名硬件工程师深刻领会硬件开发流程中各项内容，在日常工作中自觉按流程办事，是非常重要的，否则若大一个公司就会走向混乱。所有硬件工程师应把学流程、按流程办事、发展完善流程、监督流程的执行作为自己的一项职责，为公司的管理规范化做出的贡献。

§ 3.2.2 硬件开发流程详解

硬件开发流程对硬件开发的全过程进行了科学分解，规范了硬件开发的五大任务。

- 硬件需求分析
- 硬件系统设计
- 硬件开发及过程控制
- 系统联调
- 文档归档及验收申请。

硬件开发真正起始应在立项后，即接到立项任务书后，但在实际工作中，许多项目在立项前已做了大量硬件设计工作。立项完成后，项目组就已有产品规格说明书，系统需求说明书及项目总体方案书，这些文件都已进行过评审。项目组接到任务后，首先要做的硬件开发工作就是要进行硬件需求分析，撰写硬件需求规格说明书。硬件需求分析在整个产品开发过程中是非常重要的环节，硬件工程师更应对这一项内容加以重视。

一项产品的性能往往是由软件和硬件共同完成的，哪些是由硬件完成，哪些是由软件完成，项目组必须在需求时加以细致考虑。硬件需求分析还可以明确硬件开发任务。并从总体上论证现在的硬件水平，包括公司的硬件技术水平是否能满足需求。硬件需求分析主要有下列内容。

- 系统工程组网及使用说明

基本配置及其互连方法

运行环境

硬件整体系统的基本功能和主要性能指标

硬件分系统的基本功能和主要功能指标

功能模块的划分

关键技术的攻关

外购硬件的名称型号、生产单位、主要技术指标

主要仪器设备

内部合作，对外合作，国内外同类产品硬件技术介绍

可靠性、稳定性、电磁兼容讨论

电源、工艺结构设计

硬件测试方案

从上可见，硬件开发总体方案，把整个系统进一步具体化。硬件开发总体设计是最重要的环节之一。总体设计不好，可能出现致命的问题，造成的损失有许多是无法

挽回的。另外，总体方案设计对各个单板的任务以及相关的关系进一步明确，单板的设计要以总体设计方案为依据。而产品的好坏特别是系统的设计合理性、科学性、可靠性、稳定性与总体设计关系密切。

硬件需求分析和硬件总体设计完成后，总体办和管理办要对其进行评审。一个好的产品，特别是大型复杂产品，总体方案进行反复论证是不可缺少的。只有经过多次反复论证的方案，才可能成为好方案。

进行完硬件需求分析后，撰写的硬件需求分析书，不但给出项目硬件开发总的任务框架，也引导项目组对开发任务有更深入的和具体的分析，更好地来制定开发计划。

硬件需求分析完成后，项目组即可进行硬件总体设计，并撰写硬件总体方案书。硬件总体设计的主要任务就是从总体上进一步划分各单板的功能以及硬件的总体结构描述，规定各单板间的接口及有关的技术指标。硬件总体设计主要有以下内容：

- 系统功能及功能指标

系统总体结构图及功能划分

单板命名

系统逻辑框图

组成系统各功能块的逻辑框图，电路结构图及单板组成

单板逻辑框图和电路结构图

关键技术讨论

关键器件

总体审查包括两部分，一是对有关文档的格式，内容的科学性，描述的准确性以及详简情况进行审查。再就是对总体设计中技术合理性、可行性等进行审查。如果评审不能通过，项目组必须对自己的方案重新进行修订。

硬件总体设计方案通过后，即可着手关键器件的申购，主要工作由项目组来完成，计划处总体办进行把关。关键元器件往往是一个项目能否顺利实施的重要目标。

关键器件落实后，即要进行结构电源设计、单板总体设计。结构电源设计由结构室、MBC等单位协作完成，项目组必须准确地把自已的需求写成任务书，经批准后送达相关单位。

单板总体设计需要项目与CAD配合完成。单板总体设计过程中，对电路板的布局、走线的速率、线间干扰以及EMI等的设计应与CAD室合作。CAD室可利用相应分析软件进行辅助分析。单板总体设计完成后，出单板总体设计方案书。总体设计主要包括下列内容：

- 单板在整机中的的位置：单板功能描述

单板尺寸

单板逻辑图及各功能模块说明

单板软件功能描述

单板软件功能模块划分

接口定义及与相关板的关系

重要性能指标、功耗及采用标准

开发用仪器仪表等

每个单板都要有总体设计方案，且要经过总体办和管理办的联系评审。否则要重新设计。只有单板总体方案通过后，才可以进行单板详细设计。

单板详细设计包括两大部分：

- 单板软件详细设计

单板硬件详细设计

单板软、硬件详细设计，要遵守公司的硬件设计技术规范，必须对物料选用，以及成本控制等上加以注意。本书其他章节的大部分内容都是与该部分有关的，希望大家在工作中不断应用，不断充实和修正，使本书内容更加丰富和实用。。

不同的单板，硬件详细设计差别很大。但应包括下列部分：

单板整体功能的准确描述和模块的精心划分。

接口的详细设计。

关键元器件的功能描述及评审，元器件的选择。

符合规范的原理图及PCB图。

对PCB板的测试及调试计划。

单板详细设计要撰写单板详细设计报告。

详细设计报告必须经过审核通过。单板软件的详细设计报告由管理办组织审查，而单板硬件的详细设计报告，则要由总体办、管理办、CAD室联合进行审查，如果审查通过，方可进行PCB板设计，如果通不过，则返回硬件需求分析处，重新进行整个过程。这样做的目的在于让项目组重新审查一下，某个单板详细设计通不过，是否会引起项目整体设计的改动。

如单板详细设计报告通过，项目组一边要与计划处配合准备单板物料申购，一方面进行PCB板设计。PCB板设计需要项目组与CAD室配合进行，PCB原理图是由项目组完成的，而PCB画板和投板的管理工作都由CAD室完成。PCB投板有专门的PCB样板流程。PCB板设计完成后，就要进行单板硬件过程调试，调试过程中要注意多记录、总结，勤于整理，写出单板硬件过程调试文档。当单板调试完成，项目组要把单板放到相应环境进行单板硬件测试，并撰写硬件测试文档。如果PCB测试不通过，要重新投板，则要由项目组、管理办、总体办、CAD室联合决定。

在结构电源，单板软硬件都已完成开发后，就可以进行联调，撰写系统联调报告。联调是整机性能提高，稳定的重要环节，认真周到的联调可以发现各单板以及整体设计的不足，也是验证设计目的是否达到的唯一方法。因此，联调必须预先撰写联调计划，并对整个联调过程进行详细记录。只有对各种可能的环节验证到才能保证机器走向市场后工作的可靠性和稳定性。联调后，必须经总体办和管理办，对联调结果进行评审，看是不是符合设计要求。如果不符合设计要求将要返回去进行优化设计。

如果联调通过，项目要进行文件归档，把应该归档的文件准备好，经总体办、管理办评审，如果通过，才可进行验收。

总之，硬件开发流程是硬件工程师规范日常开发工作的重要依据，全体硬件工程师必须认真学习。

第二节 硬件开发文档规范

§ 2.2.1 硬件开发文档规范文件介绍

为规范硬件开发过程中文档的编写，明确文档的格式和内容，规定硬件开发过程中所需文档清单，与《硬件开发流程》对应制定了《硬件开发文档编制规范》。开发人员在写文档时往往会漏掉一些该写的内容，编制规范在开发人员写文档时也有一定的提示作用。《硬件开发文档编制规范》适用于中央研究部立项项目硬件系统的开发阶段及测试阶段的文档编制。规范中共列出以下文档的规范：

- 硬件需求说明书
- 硬件总体设计报告
- 单板总体设计方案
- 单板硬件详细设计
- 单板软件详细设计
- 单板硬件过程调试文档
- 单板软件过程调试文档
- 单板系统联调报告
- 单板硬件测试文档
- 单板软件归档详细文档
- 单板软件归档详细文档
- 硬件总体方案归档详细文档
- 硬件单板总体方案归档详细文档
- 硬件信息库

这些规范的具体内容可在HUAWEI服务器中的“中研部ISO9000资料库”中找到，对应每个文档规范都有相应的模板可供开发人员在写文档时“填空”使用。模块在rndI服务器中的文档管理数据库中。

&2.2.2 硬件开发文档编制规范详解

1、硬件需求说明书

硬件需求说明书是描写硬件开发目标，基本功能、基本配置，主要性能指标、运行环境，约束条件以及开发经费和进度等要求，它的要求依据是产品规格说明书和系统需求说明书。它是硬件总体设计和制订硬件开发计划的依据，

具体编写的内容有：系统工程组网及使用说明、硬件整体系统的基本功能和主要性能指标、硬件分系统的基本功能和主要性能指标以及功能模块的划分等。

2、硬件总体设计报告

硬件总体设计报告是根据需求说明书的要求进行总体设计后出的报告，它是硬件详细设计的依据。编写硬件总体设计报告应包含以下内容：

系统总体结构及功能划分，系统逻辑框图、组成系统各功能模块的逻辑框图，电路结构图及单板组成，单板逻辑框图和电路结构图，以及可靠性、安全性、电磁兼容性讨论和硬件测试方案等。

3、单板总体设计方案

在单板的总体设计方案定下来之后应出这份文档，单板总体设计方案应包含单板版本号，单板在整机中的位置、开发目的及主要功能，单板功能描述、单板逻辑框图及各功能模块说明，单板软件功能描述及功能模块划分、接口简单定义与相关板的关系，主要性能指标、功耗和采用标准。

4、单板硬件详细设计

在单板硬件进入到详细设计阶段，应提交单板硬件详细设计报告。在单板硬件详细设计中应着重体现：单板逻辑框图及各功能模块详细说明，各功能模块实现方式、地址分配、控制方式、接口方式、存储器空间、中断方式、接口管脚信号详细定义、时序说明、性能指标、指示灯说明、外接线定义、可编程器件图、功能模块说明、原理图、详细物料清单以及单板测试、调试计划。有时候一块单板的硬件和软件分别由两个开发人员开发，因此这时候单板硬件详细设计便为软件设计者提供了一个详细的指导，因此单板硬件详细设计报告至关重要。尤其是地址分配、控制方式、接口方式、中断方式是编制单板软件的基础，一定要详细写出。

5、单板软件详细设计

在单板软件设计完成后应相应完成单板软件详细设计报告，在报告中应列出完成单板软件的编程语言，编译器的调试环境，硬件描述与功能要求及数据结构等。要特别强调的是：要详细列出详细的设计细节，其中包括中断、主程序、子程序的功能、入口参数、出口参数、局部变量、函数调用和流程图。在有关通讯协议的描述中，应说明物理层，链路层通讯协议和高层通讯协议由哪些文档定义。

6、单板硬件过程调试文档

开发过程中，每次所投PCB板，工程师应提交一份过程文档，以便管理阶层了解进度，进行考评，另外也给其他相关工程师留下一份有参考价值的技术文档。每次所投PCB板时应制作此文档。这份文档应包括以下内容：单板硬件功能模块划分，单板硬件各模块调试进度，调试中出现的问题及解决方法，原始数据记录、系统方案修改说明、单板方案修改说明、器件改换说明、原理图、PCB图修改说明、可编程器件修改说明、调试工作阶段总结、调试进展说明、下阶段调试计划以及测试方案的修改。

7、单板软件过程调试文档

每月收集一次单板软件过程调试文档，或调试完毕（指不满一月）收集，尽可能清楚，完整列出软件调试修改过程。单板软件过程调试文档应当包括以下内容：单板软件功能模块划分及各功能模块调试进度、单板软件调试出现问题及解决、下阶段的调试计划、测试方案修改。

8、单板系统联调报告

在项目进入单板系统联调阶段，应出单板系统联调报告。单板系统联调报告包括这些内容：系统功能模块划分、系统功能模块调试进展、系统接口信号的测试原始记录及分析、系统联调中出现的问题及解决、调试技巧集锦、整机性能评估等。

9、单板硬件测试文档

在单板调试完之后，申请内部验收之前，应先进行自测以确保每个功能都能实现，每项指标都能满足。自测完毕应出单板硬件测试文档，单板硬件测试文档包括以下内容：单板功能模块划分、各功能模块设计输入输出信号及性能参数、各功能模块测试点确定、各测试参考点实测原始记录及分析、板内高速信号线测试原始记录及分析、系统I/O口信号线测试原始记录及分析，整板性能测试结果分析。

10、硬件信息库

为了共享技术资料，我们希望建立一个共享资料库，每一块单板都希望将的最有价值最有特色的资料归入此库。硬件信息库包括以下内容：典型应用电路、特色电路、

特色芯片技术介绍、特色芯片的使用说明、驱动程序的流程图、源程序、相关硬件电路说明、PCB布板注意事项、单板调试中出现的典型及解决、软硬件设计及调试技巧。

第三节 与硬件开发相关的流程文件介绍

与硬件开发相关的流程主要有以下几个：

项目立项流程

项目实施管理流程

软件开发流程

系统测试工作流程

中试接口流程

内部接收流程

§ 3.3.1 项目立项流程：

是为了加强立项管理及立项的科学性而制定的。其中包括立项的论证、审核分析，以期做到合理进行开发，合理进行资源分配，并对该立项前的预研过程进行规范和管理。立项时，对硬件的开发方案的审查是重要内容。

§ 3.3.2 项目实施管理流程：

主要定义和说明项目在立项后进行项目系统分析和总体设计以及软硬件开发和内部验收等的过程和接口，并指出了开发过程中需形成的各种文档。该流程包含着硬件开关、软件开发、结构和电源开发、物料申购并各分流程。

§ 3.3.3 软件开发流程：

与硬件开发流程相对应的是软件开发流程，软件开发流程是对大型系统软件开发规范化管理文件，流程目的在对软件开发实施有效的计划和管理，从而进一步提高软件开发的工程化、系统化水平，提高华为公司软件产品质量和文档管理水平，以保证软件开发的规范性和继承性。软件开发与硬件结构密切联系在一起的。一个系统软件和硬件是相互关联着的。

§ 3.3.4 系统测试工作流程：

该流程规定了在开发过程中系统测试过程，描述了系统测试所要执行的功能，输入、输出的文件以及有关的检查评审点。它规范了系统测试工作的行为，以提高系统测试的可控性，从而为系统质量保证提供一个重要手段。

项目立项完成，成立项目组的同时要成立对应的测试项目组。在整个开发过程中，测试可分为三个阶段，单元测试、集成测试、系统测试。测试的主要对象为软件系统。

§ 3.3.5 中试接口流程

中试涉及到中央研究部与中试部开发全过程。中研部在项目立项审核或项目立项后以书面文件通知中试部，中试部以此来确定是否参与该项目的测试及中试准备的相关人选，并在方案评审阶段参与进来对产品的工艺、结构、兼容性及可生产性等问题进行评审，在产品开发的后期，项目组将中试的相关资料备齐，提交《新产品准备中试联络单》，由业务部、总体办、中研计划处审核后，提交中试部进行中试准备，在项目内部验收后转中试，在中试过程中出现的中试问题，由中试部书面通知反馈给项目组，进行设计调整直至中试通过。

由上可见中试将在产品设计到验收后整个过程都将参与，在硬件开发上，也有许多方面要提早与中试进行联系。甚至中试部直接参与有关的硬件开发和测试工程。

§ 3.3.6 内部验收流程

制定的目的是加强内部验收的规范化管理，加强设计验证的控制，确保产品开发尽快进入中试和生产并顺利推向市场。项目完成开发工作和文档及相关技术资料后，首先准备测试环境，进行自测，并向总体办递交《系统测试报告》及项目验收申请表，总体办审核同意项目验收申请后，要求项目组确定测试项目，并编写《测试项目手册》。测试项目手册要通过总体办组织的评审，然后才组成专家进行验收。

由上可见，硬件开发过程中，必须提前准备好文档及各种技术资料，同时在设计时必须考虑到测试。

第三节 常用的接口及总线设计

§ 3.3.1 接口标准:

常用的接口类型，按物理电气特性划分，大致可分为以下几类:

1. **TTL电平接口**: 最通用的接口类型, 常用做板内及相连板间接口信号标准。其信号速度一般限制在二、三十兆HZ以内。驱动能力一般为几毫安到几十毫安, 产品设计特别是总线设计时必须考虑负载能力。
2. **CMOS电平接口**: 速度范围与TTL相仿, 驱动能力要弱一些。
3. **EIL电平接口**: 为高速电气接口, 速率可达几百兆, 但相应功耗较大, 电磁辐射与干扰与较大。
4. **RS-232电平接口**: 为低速串行通信接口标准, 电平为 $\pm 12V$, 用于DTE与DCE之间的连接。
5. **差分平衡电平接口**: 能实现较远距离, 较高速率的传输, 2MHZ信号在匹配适当地情况下, 传输距离在15m以上。
6. **光隔离接口**: 能实现电气隔离, 允许信号带宽一般在10M以内, 更高速率的器件价格较昂贵。
7. **线圈耦合接口**: 电气隔离特性好, 但允许信号带宽有限。

常用的板极接口类型, 按接口实现的逻辑形式, 主要有以下几类:

共享内存接口

并行口 (PIO) 接口

串行口 (STI) 接口

多总线技术

§ 3.3.2 串口设计:

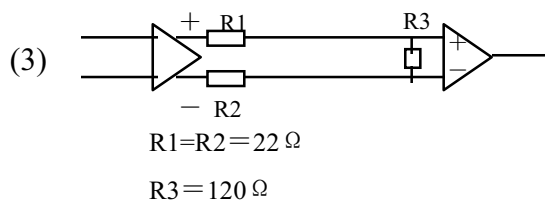
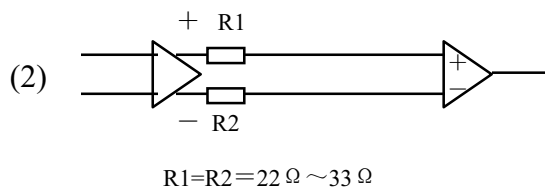
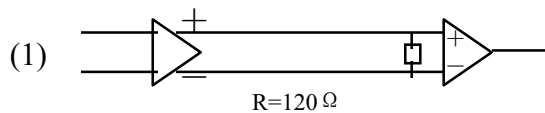
异步串口: 速度可达几百Kbps, 在速度大于15kbps时, 为可靠传输, 一般推荐用EIA-402或EIA 485电平接口标准进行传输。

同步串口: 常用的接口芯片如82525、20320, 速率为2Mbps或4Mbps。

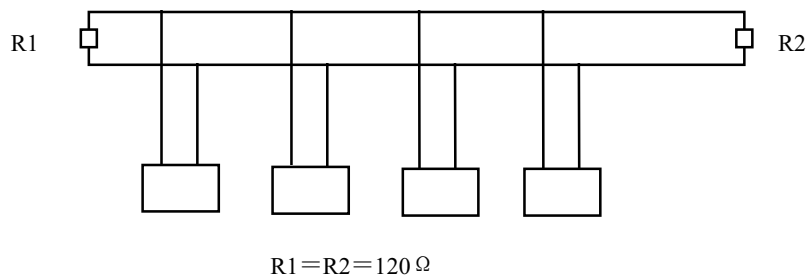
无论同步或异步, 在较高速率传输时, 除在板内或传输线长度较短的板间传输时可用TTL电平接口外, 一般要求用差分平衡电平接口: EIA422、端到端传输

FIA-485: 差分总线传输、终端可达329。

使用差分平衡电平传输时, 为保证传输质量, 应进行信号匹配, 对EIA-422, 常用的匹配方法:



对于EIA-485差分总线传输，需在端点进行匹配：



§ 3.3.3 并口设计及总线设计：

以AT总线为例，总线设计的几个主要注意事项如下：

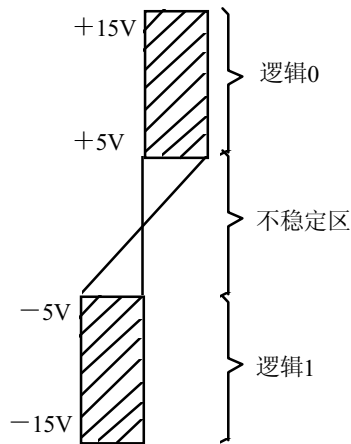
1. 总线驱动能力设计；根据实际应用中可能接入总线的板卡板，应使系统线具有足够的驱动能力。
2. 总线隔离挂接在总线上的各单板应使用总线隔离器件。一方面减轻时总线的负载要求，另一面对故障进行隔离，另外也确保信号是总线方式而非星形方式传送。
3. 时延考虑：各类总线都有一个速率限制范围，这就限制了总线的物理尺寸、设计时应考虑这点。另外，在底板尺寸较大时，地址、数据、控制，总线任一槽位的时延确保其时序联系正确。
4. 防串扰考虑：总线设计时，应尽量避免总线长距离相邻走线，如PCB板尺寸允许，线线之间应加入保护隔离地线。

§ 3.3.4 RS-232接口总线

RS-232 总线是采用按位串行的通讯总线，但它并不限制所传送的数据类型和数据帧长。可用于同步通信也用于异步通讯。

RS-232对其所传输的数据格式约定为：信息起始位、数据块停止位、奇偶校验位及若干数据位。

RS-232为了增加抗干扰能力，采用负逻辑电路，其逻辑电平范围如下图所示：



RS-232的接口电气：微机接口及内部电路是采用TTL和CMOS型电路。这些电路都不能直接与RS-232相连，中间必须要进行电平转换，常用的转换芯片有MC1488、MC1489、MAX232等。MC1488、MC1489需接 $\pm 12V$ 的电源，其中MC1488用在通讯设备的发送端，MC1489用在通讯设备的接收端。MAX232是单5V供电。片内集成有升压电路产生 $\pm 12V$ 的电压，使用时片外需匹配相应的电容。

RS-232传输的速率有300、600、1200、2400、4800、9600等标准的波特率。

RS-232最大传输距离是30m，如果要以较高的波特率传输较远的距离，则可选用RS-422，其传输距离可达1500m。

§ 3.3..5 RS-422和RS-423标准接口联接方法

1、RS-423接口

RS-232应用广泛，但对于现代通信网络来说存在以下问题：①数据传输速率慢，只有20KB/s；②传输距离短，一般只允许20~30m；③抗干扰能力差，为提高传输过程的抗干扰能力，RS-232采用高压（ $\pm 10V$ ）传输数据。即使如此，由于它采用不平衡方式传输，即传输线中一条为信号线，另一条为地线，干扰电平对信号线的影响与较大，影响传输的接口的可靠性。因此，为提高抗干扰能力和加大传输距离，可采用RS-422或RS-423标准接口。

RS-423接口实际上是用双端差分接收器代替单端接收器，联接方式如图1所示。由于两条传输线是双曲线，所受的干扰基本相同，对差分接收器的输入端来说相当于共模信号，不影响其差值，故对接收器无影响。

RS-423规定为单端输出线接到接收器的一个输出端，接收器的另一个输入端接驱动器的参考电平（地），驱动器与接收器没有公共地线，即为双端非平衡联接方式。要求逻辑1电平为 $-0.2 \sim -6V$ ，逻辑0电平为 $+0.2 \sim 6V$ 。RS-423与RS-232兼容， $\pm 6V$ 信号可作为RS-232一起工作时，其标准与RS-232一致。

与RS-232相比，RS-423抗干扰能力强，传输距离较远，传输速率也较高。传输距离为90m时，最大传输速率为100KB/s，则传输长度可达到1200m。

2、RS-422接口

RS-422采用双端平衡传输方式，即输入输出均为差动方式。其中一条线是逻辑1时，另一条线为逻辑0。由于两条双扭线传送的是一对互补信号，故抗干扰能力强、传输速率高。应用RS-422驱动器和接收器时，最大传输速率为10MB/s，这种情况下传输长度为120m。传输速率降低时，传输距离可达1200m。平衡驱动器的双端输出信号差要求在 $\pm 2V$ 以上。RS-422联接方式如图2所示。

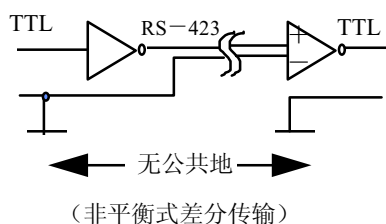


图1 RS-423联接方法

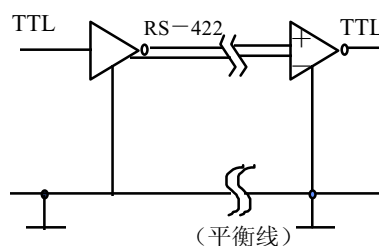


图2 RS-422联接方法

3、RS-423接口电平转换电路与联接方法

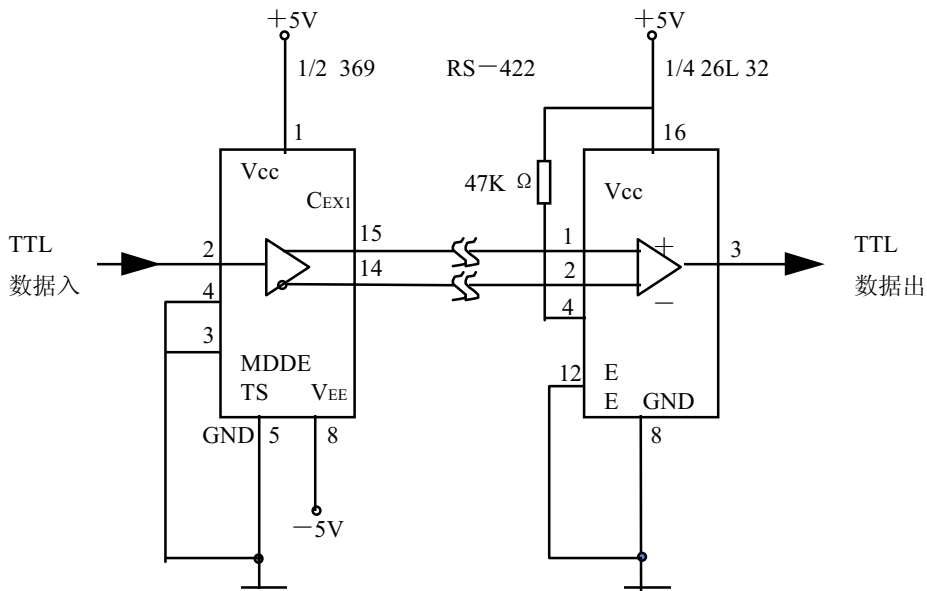
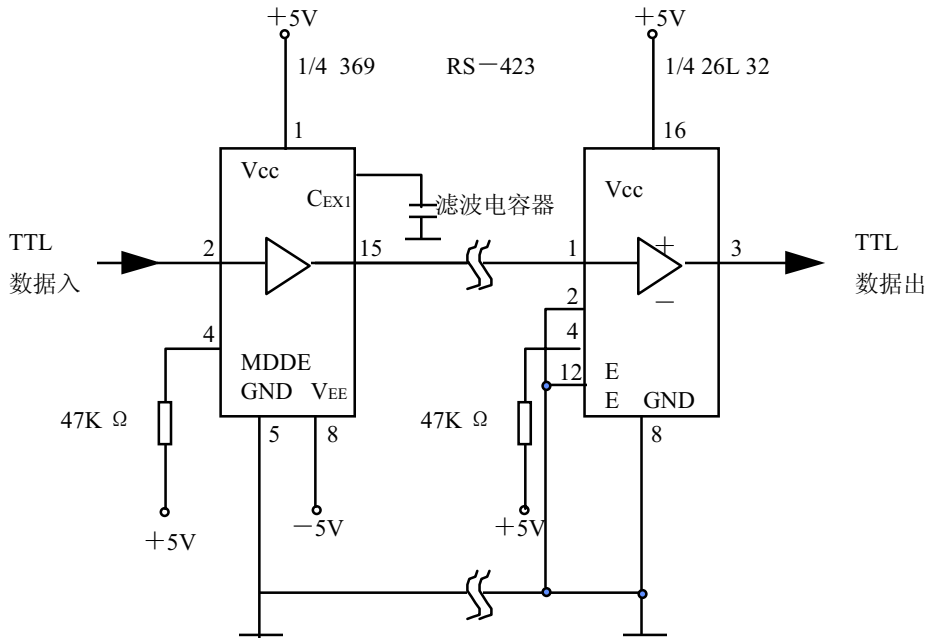
RS-423接口标准对接口电路的要求是：传输速率为100KB/s，传输距离最大12m，接收器为双端接收器。高电平必须在 $+4\sim+6V$ 范围内，低电平必须在 $-4\sim-6V$ 范围内。图12-30是使用3691驱动器和26L32接收器时的RS-423接口联接方法。

4、RS-422接口电平转换电路与联接方法

它要求驱动器双端输出电平在 $\pm 2V\sim\pm 6V$ 之间，接收器可检测到的信号为 $\pm 200mV$ 。RS-422接口电平转换电路及联接方法如图 所示。有些驱动/接收器具有三态控制，用适当的信号控制芯片的三态控制端，就可实现几个设备在一对接口传输线上采用半双工方式接收和发送数据。

§ 3.3.6 RS-485标准接口与联接方法

与RS-422类似，RS-485也是采用平衡驱动、双端平衡差分输入方式，技术性能规范与RS-422相同，接口联接方法与一样。它与RS-422驱动器具有较强负载能力，接收器的负载较小，允许一个驱动器驱动多个从机输入上，多个从机的输出（每个时刻只有一个有效）需联到主机的输入端。在这种一个驱动器连到多个连接器的场合，应使用RS-485接口。



常用的RS×485接口发送器为MC/SN75174、接收器为MC/SN75175，以及发送驱动器/接收器为MC75176，它包括一个驱动器和一个接收器。SN75174和SN75175引脚与结构如图 所示。在应用时如果为双机通信，可不使用驱动器/接收器的允许控制端，即把75175的允许端接固定高电平。在多机通信时，从机的驱动器必须加以控制，在收到主机传送的地址与本机相符时，才允许接通该驱动器。

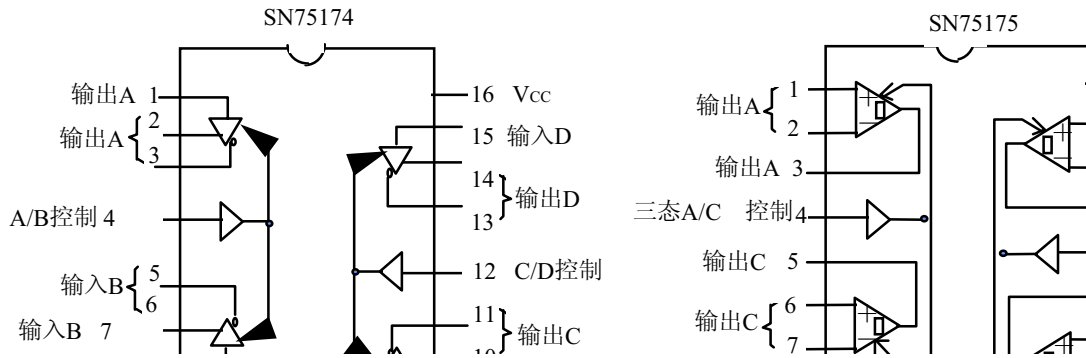
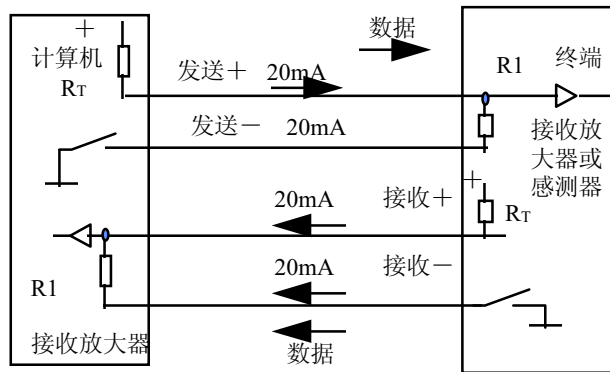


图3 SN75174和SN75175的管脚与内部结构

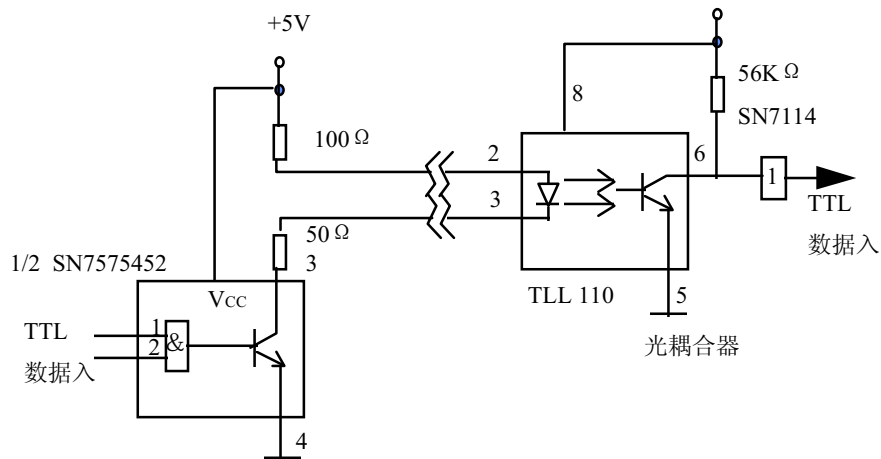
§ 3.3.7 20mA电流环路串行接口与联接方法

20mA电流环路串行接口也是目前串行通信中广泛使用的接口，但未形成正式标准的接口电路。这种接口比RS-232接口简单得多，只有4条线，如图4所示。正、负发送端和正、负接收端形成一个输入电流回路和一个输出电路回路。当发送数据时，根据数据的逻辑1或0，分别使回路形成通断状态，如图4中的开关所示。



20mA电流环路是异步串行接口，每次发送数据时必须以无电流的起始位作为每一个字符的起始位，接收端检测到起始位时便接收一个字符数据。这种接口方法的最大优点是低阻传输线对噪声不敏感，并且容易实现光电隔离。

20mA电流环路串行接口采用信号电流传输，接口电路的联接方法如图 所示。图中先将TTL电路转换成电流信号，传输后经光耦合器再转换成TTL电平信号。这是一个对地隔离的电流环路传送和接收电路。



本节阐述设计人员在设计TTL电路时需考虑的各种要求。

1、电路的极限参数和规范参数

极限参数是保证集成电路能可靠工作的权限值。为了防止电路损坏，电路的工作条件不应超过所规定的极限范围。规范参数是保证集成电路能够正常工作的参数，所以电路必须严格按器件参数规范值进行测试和使用。

目前，国内外推出的各种产品其种类，特性参数，即使是同一种功能的TTL器件其工作特性也不完全相同，因此在选用一种新器件以前，一定要仔细阅读厂家提供的参数。

2、电源和地

对电源的纹波和稳定度的要求：

I类、III类 $\leq \pm 5\% (\pm 250\text{mV})$

III类 $\leq \pm 10\% (\pm 500\text{mV})$

同时根据不同系列电路的速度指标和电流变化率的差异还必须加接旁路滤波电容。其容量通常为 $0.01\mu\text{F} \sim 0.1\mu\text{F}$ 。

低频滤波电容器应连接在印制电路板两边电源线入口处，通常用两个 $20\mu\text{F} \sim 50\mu\text{F}$ 的钽电容或电解电容。

3、电路的工作环境

国产各系列TTL电路的适用环境温度可分为下述三类：

I类 $-55^\circ\text{C} \sim +125^\circ\text{C}$

II类 $-40^\circ\text{C} \sim +85^\circ\text{C}$

III类 $-10^\circ\text{C} \sim +70^\circ\text{C}$

在此工作温度范围内，电路的功能和参数应完全符合产品规定的要求。

4、电路外引线端的连接

(1) 在使用TTL电路时，不能将电源V_{cc}和地线颠倒错接，否则将引起很大的电流而造成电路失效。

(2) 电路的各输入端不能直接与高于+5.5V和低于-0.5V的低内阻电源连接。因为低内阻电源能提供较大电流，会由于过流而烧坏电路。

(3) 不允许将电路的输出端与低内阻电源直接相连接，但可以通过电阻相连以提高输出高电平。当使用集电极开路输出电路时，必须通过计算选用阻值合适的外接电阻。

(4) 具有图腾柱或达林顿输出结构的TTL电路不允许并联使用。只有三态或集电极开路输出结构的电路可以并联使用。当若干个三态逻辑门并联使用时只允许其中一个门处于使能状态（“0”态或“1”态），其它所有门应处于高阻态。当将集电极开路门输出端并联使用时，只允许其中一个门处于低电平输出状态，其它门则应处于高电平输出状态，否则将会出现逻辑错误。

(5) 当将一些集电极开路门路的输出端并联而使电路具有“线与”功能时，通常应在其公共端出端加接一个上拉负载电阻R_L到V_{cc}端。

(6) 集电极开路输出电路中输出管的击穿电压一般在10V以上，有的可达20V。只要在输出管所允许的驱动能力和击穿电压范围内就可任意选用工作电压值，用作电平转换接口，指示灯驱动等。

5、对输入信号边沿的要求

驱动TTL电路的输入信号必须具有较快的转换时间。当输入信号上升或下降时间大于1 μ S时，有可能在输出端出现信号振荡。这种振荡信号送入触发器或单稳态触发器中就可能引起逻辑错误。

一般组合电路的输入信号上升或下降沿变化速率应小于100ns/V，时序电路输入信号上升或下降沿变化速度应小于50ns/V。施密特触发器把缓慢变化的信号边沿变成陡变的边沿。对于慢边沿输入信号，必须加整形器。

6 TTL 门、扩展器和触发器的使用要求

(1)、输出端的连接

TTL电路的输出端不允许与电源短路，但允许瞬间接地。当一个管壳内封装有若干个单元电路时，不允许其中的几个单元电路的输出端同时瞬间接地。

(2)、不使用的输入端的处理方法

A • 若电源电压不超过5.5V，与门电路的使用端输入端和触发器不使用的置位、复位端可直接连到电源V_{cc}上；也可将不用的输入端分别通过一个大于或等于1kΩ的电阻连到V_{cc}上；或把几个不使用输入端通过一个公共电阻连到电源V_{cc}上；也可以把同一块电路的不使用的输入端并联到该电路的一个已被使用的输入端上。

B • 如果前级驱动器具有足够的驱动能力，可将与或非门的不使用的与输入端直接连到与或非门的已使用的某一个输入端上，但也可把不使用的或输入端接地。

C • 为了达到最低功耗，可将不使用的与或非门和或非门等器件的所有输入端接地，同时也可将它的输出高电平连到不使用的与输入端上。

D • 不使用的输入端可以悬空但决不允许带开路长线，以免产生“低频效应”，造成单拍工作失常。

(3) 扩展器、触发器输入端的连接

带扩展的门应尽量与其所带的扩展器靠近，使其连线最短，以避免增加扩展器连线的分布电容。一个或扩展门最多可带两个或扩展器，而且每带一个或扩展器级延迟将增加40%左右。

不应将触发器的不使用输入端悬空。触发器的输入端连接线应尽量短。当用若干触发器构成时序电路时，由于时钟信号沿传输线传送时存在着时间偏差，到达或进入触发器的时钟信号将产生时钟偏移 t_{skew} 。应使 t_{skew} 尽可能小。

(4)、触发器对输出负载和信号源的要求

A 应正确区分锁存触发器，D型触发器和J—K触发器对电位输入和时钟信号的要求，即电位输入信号（数据）与时钟信号的时间关系必须满足推荐使用条件中的建立时间 t_{set} 和保持时间 t_h 的要求。

B 一般不允许触发器的输出直接驱动各种指示灯、长传输线和电感负载，否则必须附加缓冲门。

三 CMOS电路的工程设计

CMOS IC是压敏器件，其输入阻抗很高，用CMOS IC直接带动CMOS IC时，其直流负载是很轻的。主要考虑电平匹配就可以。

1、对总线终端的驱动

高速CMOS系统中，以CMOS IC驱动总线是很常见的事。工作时不能让总线浮空，而是通过上拉或下拉把总线接到V_{cc}或V_{ss}上。为了提高工作速度，根据一般充电慢于放电的特点，在实际中大多数采用常态时将总线上拉为高电平的方法。加入上拉或下拉电阻后，可以大大减少总线受噪声的干扰，使总线工作在所有三态总线驱动器全部处在三态时，也不会被悬浮起来的状态。

在总线设计中，终端电阻阻的选择，往往要考虑速度和或功耗的折衷。总线的工作速度是与总线相关寄生电容和终端电阻形成的RC时间常数的函数，终端电阻越低，总线工作速度越快，总线功耗也越大。Motorola建议用户在使用HCMOS IC系列电路时，选取总线终端电阻的阻值可在 $1k\Omega$ - $1M\Omega$ 之间。

当数据长距离传送时，数据的通过的导线可以看作是长线。当然，线距离的长短，应当是相对于数据传输率而言的。高速工作的总线、PCB板上的长线、同轴线和带状电缆都可视为长线。所有的长线都可以接成低阻抗终端。低阻终端有利于减少噪声，减少瞬时扰动，减少过冲和串扰，低阻终端还可更快地将传至接收终端，然而却加大了功耗。所以长线终端阻值的选定，仍旧是功耗、数率速度及传输线长度诸因素之间的折衷。

当然可以使用更高的电阻阻值，但却减慢了速度且可能使信号退化，然而可以降低功耗。

虽然所有的导线都具有传线特性，但并不是对每一根导线都要考虑它的传输线效应。一般认为，只有在驱动器输出波形上升时间和下降时间等于或小于长线（即传输）延时的3倍时，才需要考虑传线效应，这种终端的功耗比较大，往往超过逻辑元件本身的功耗，因此在高速CMOS IC系统中，一般不提倡使用它。

四 ECL电路的工程设计

ECL电路（即发射极耦合逻辑电路）是一种非饱和型的数字逻辑电路。与DTL、TTL、S-TTL等逻辑电路不同，ECL电路内部的晶体管工作在线性区或截止区，从根本上消除了限制速度提高的少数载流子的“存储时间”。因此，它是现有各种逻辑电路中速度最快的一种电路形式，也是目前唯一能够提供亚毫微秒开关时间的实用电路。

1、基本门电路的结构

典型的ECL基本门电路的结构由三部分组成：差分放大器输入电路：温度—电压补偿（跟踪）偏压网络（参考源）和射极跟随器输出电路。

ECL电路一般能用于驱动传输线，因此通常设计成射极开路输出的形式。此时，传输线的终端匹配电阻 R_L 即为输出负载。

2 ECL电路的特点及应用范围

ECL电路是根据高速噪声数字的应用要求设计的，它具有以下独特的优点：

（1）、速度快

速度快是高速数字系统线咱设计者广泛采用ECL电路的一个重要原因。ECL基本门电路的典型传输延迟时间已达到亚毫微秒量级，其触发器、计数器的工作频率也在1GHz范围。因此，一个ECL系统与等效的TTL系统相比，其工作速度至少可以快一倍以上。

(2)、逻辑功能强

ECL电路能同时提供互补逻辑输出，这样不仅可以节省系统所用的组件数，减小系统功耗，而且由于互补输出具有相同的传输延迟时间，因此可以消除一般逻辑电路中为产生互补逻辑功能而设置反相器所增加的时间延迟，进而提高了系统的速度。

(3)、扇出能力高

输入阻抗高（通常约 $10\text{K}\Omega$ ），输出阻抗低（约 7Ω ）是ECL电路的性能特点之一，这种特点允许电路有高的扇出能力以 $CE10K$ 。

(4)、噪声低

系统噪声的大小直接与噪声源的能量、逻辑的消噪性能和互连线的阻抗等有关。就噪声的产生来说，ECL电路的内部噪声较小。

(5)、便于数据传输

ECL电路具有互补、大电流驱动能力输出特别适合于以差分方式驱动和接收双绞线或其它平衡线上的信号。ECL电路的差争线接收器具有 1V 或者更大的共态噪声抑制能力。这是因为差分工作时，耦合到双绞线上的任何噪声一般是等同地出现在该双绞线的每股线上（共态），即串扰是等同地被线拾取，而接收器只响应两条线上的电压差，所以可大大抑制引线串扰的影响，从而易于实现远距离的数据传输。驱动同轴电缆时，其距离只受电缆频带宽度的限制，而且可以改善系统的性能，驱动双绞线的长度可以在 300m (约 1000ft)以上，并且较同轴电缆经济。

除了上面介绍的主要特点以外，ECL电路的结构还提供了其它若干有益的特性，它们是：

(1) 可以简化电源。ECL电路对电源电压的同步变化是不太敏感的，因此可以在某些应用中相对地放松对电源波纹、偏差和分配的要求。有时允许ECL电路的电源电压范围可宽至 $\pm 10\%$ 。由于ECL电路工作时电源电流基本上恒定（不随逻辑状态变化而变化，也不随工作频率增加而增加），因此可以考虑放宽对电源内阻的要求。加上参考电源是设计在电路内部，因此整个电路可以由单一电源供电。所有这些，使电源系统设计简单、成本降低。

(2) 逻辑功能变化范围宽，适应性强。ECL电路的差分放大器设计允许它作线性方面的应用。ECL线接收器可以用施密触发器和线性放大器用。由于这一应用灵活性，许多功能可以用标准的ECL电路来完成。

(3) 由频率提高引起的附加功耗小。开关工作时因对寄生电容充放电而要消耗一定的能量，对于有电源电流尖峰的逻辑电路来说消耗的能量更大。由于每次充放电都要消耗能量，所以TTL电路的功耗在高频范围随开关频率提高而显著增加。由于存储在杂散电容中的能量与电压平方成正比，而ECL电路的信号摆幅又比TTL电路小 $3\sim 4$ 倍，所以它因杂散电容引起的附加功耗要较

TTL电路小一个数量级。特别是，它没有电源电流尖峰引起的附加功耗。ECL电路的功耗基本上不随频率而变化，关于这一点在高频领域是甚为重要的。

(4) 便于实现各种规模的集成。

当然，ECL电路并不是完美无缺的，它的主要缺点是直流功耗大。从某种意义上来说，ECL电路开关速度的提高是以牺牲功耗换取的。

2、应用范围

ECL电路在高速信息系统中有着广泛的应用，主要包括：

(1) 大型高速电子计算机；(2) 高速计数器、缓冲存储器；(3) 高速模拟/数字转换系统；(4) 数字通信系统；(5) 航天和通信卫星系统、雷达系统；(6) 频率合成器；(7) 高速数字仪器和仪表；(8) 微波测量系统；(9) 数据传输、情报处理系统。

PECL电路

PECL电路是单正电源供电的ECL电路，其特点同ECL电路，由于其单正电源供电，简化了整个系统的电源设计。目前在高速光模块电路中，使用十分广泛。

六 传输线电路

线电路是指在一个系统内部或在几个系统之间和总线相接的、用于数字通信的专用集成电路。它是向传输线发送或从传输线接收信号的电路。

当系统互连线的距离增加和信号传送频率提高时，信号在传输线上的反射、串扰、衰减以及共地噪音等问题对信号传输的可靠性的影响就不能忽略。

1、信号在传输线上的传输

(1)、传输线的概念及长线 and 短线的区分

传输线是具有分布参数的传输信号的长线，其特性阻抗 Z_0 和信号在线上的传输速度由传输线单位长度的分布参数决定。

长线和短线的概念是相对于信号本身来说的。当信号沿线传播的延迟时间比信号的变化时间（例如信号的上升时间 t_r 或下降时间 t_f ）短得多时，信号在线上的任何反射仅反映在信号的边沿上，这样的线就称为短线。当信号沿线传播的延迟时间与信号变化时可以相比拟时，就必须考虑信号传播的延迟时间。此时，由于长线不均匀或负载不匹配而产生的信号反射将在线上出现“振铃”，这种较长的传输线就叫做长线。所以长线和短线的概念是相对的。

在数字信号传输领域中，常采用下述定义：线的传输延迟时间（ t_{pd} ）大于驱动电路信号上升时间（ t_r ）的一半，即大于 $(t_r/2)$ 的线称为长线，反之就是短线。例如，对于ECL和STTL电路来说，几十厘米长的线就算是长线，而对中速CMOS逻辑电路来说，则要几米长的线才算是长线。

(2)、传输线的反射和“振铃”

信号在传输线上传输时，若遇到阻抗不连续的情况，则将出现反射现象。

反射波将按正负反射系数在线上来回传输，形成所谓“振铃”现象。这往往会增加系统的噪声干扰，通常应采用匹配的方法来消除或抑制反射及“振铃”现象。

(3)、线间串扰

串扰是指一条线上的信号通过互感和互容在被干扰的线上产生不希望的耦合信号。

串扰分为电感性串扰和电容性串扰两种（参看图4-2）。电感性串扰正比于信号线上电流变化率；电容性串扰正比于线上信号电压的变化率。

串扰是系统噪音的一个主要来源，系统设计者必须仔细加以考虑。

串扰还分近端（异向）串扰和远端（同向）串扰两种。近端串扰是电容串扰和电感串扰之和；远端串扰是电感串扰和电容串扰之差。在TTL系统中，双绞线间的近端串扰幅度可达0.8V左右。应该尽量避免产生近端串扰，将平行传输信号的发送线和接收线分开，或在要求严格的场合，应采用如同轴电缆等屏蔽优良的线。

(4)、共式噪音

一个好的系统必须有一个很好的地系统和电源系统，否则通过系统共用的阻抗（电阻 R 和电感 $j\omega L$ ）将产生较大的共式噪音而使系统发生故障。

为了减少在传输信号时流经下线电流对电源和地的扰动，驱动器和接收器的接地端应该直接与电缆的地线相连，同时它们的电源和地之间要加去耦电容，这样就保证了信号传输时，线电流回路不对其它电路产生共式干扰。

上面讨论的是系统内部噪音。若走线很长，外部的噪音就成为一个重要因素。随着地线长度的增加，获得一个好的地线就更加重要，但也更加困难。同时当回路的几何尺寸很大时，它相当于一个好的天线，接收外部电磁场的干扰。为了避免形成回路，信号地线系统必须和安全地线分开，最后只通过一点接大地。

在噪音环境恶劣或对噪音要求严格的场合，信号的驱动和接收都必须采用差分平衡方式，通过抑制共式干扰（串扰也是共式干扰），而实现可靠的传输。

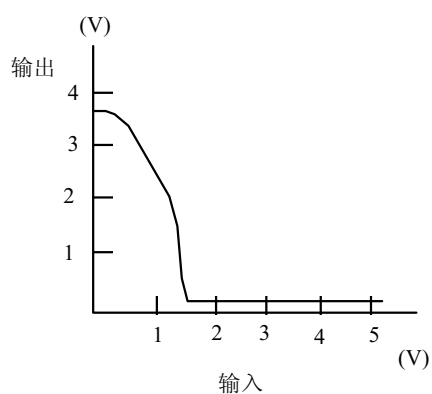
七 逻辑系列的特性TTL，CMOS，ECL及设计选择

选择最佳逻辑系列是所有数字系统设计的关键一步。有的设计要求高速操作，有的需要较低的功耗，还有的可能要求较低的成本。

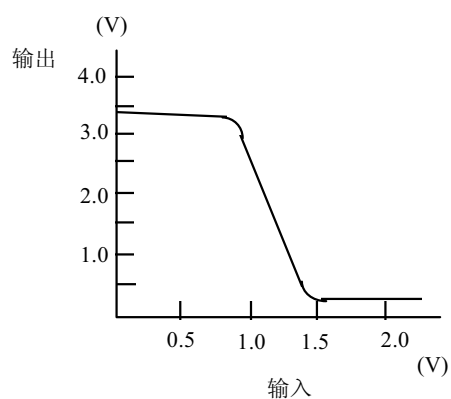
1、电压传输特性

各逻辑系列的典型电压传输特性如图1.2所示。这些曲线含有电路设计人员感兴趣的某些信息，例如，输出导电，截止电压，以及直流噪声容限，均为输入导电、截止电压的函数。除此之外，这些曲线还表明了以下四点：

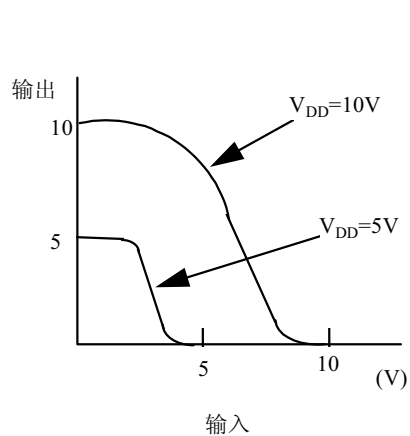
- 1、作为电源电压的函数，传输特性随电源电压的改变而改变。
- 2、作为温度的函数，传输特性随温度的改变而改变。
- 3、转换时的功率（设在同一图上描绘出电源电流曲线）。
- 4、滞后特性（对施密特触发器有用）。



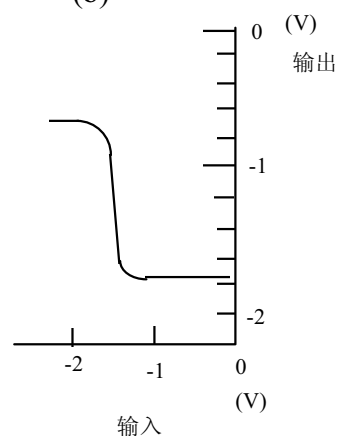
(a)



(b)



(c)



(d)

由于CMOS电路通常驱动能力较弱，必须先进行TTL转换后再驱动ECL转换芯片，反之亦然。

在三种逻辑电平进行转换时，主要考虑以下几点：

- 1· 电平关系，必须保证在各自的电平范围内工作，否则，不能满足正常逻辑功能，严重时烧毁芯片。
 - 2· 驱动能力，必须根据器件的特性参数仔细考虑，计算和试验，否则很可能造成隐患，在电源波动，受到干扰时系统就会崩溃。
 - 3· 时延特性，在高速信号进行逻辑电平转换时，会带来较大的延时，设计时一定要充分考虑其容限。
 - 4· 选用电平转换逻辑芯片时应慎重考虑，反复对比。通常逻辑电平转换芯片为通用转换芯片，可靠性高，设计方便，简化了电路，但对于具体的设计电路一定要考虑以上三种情况，合理选用。
- (a)TTL; (b)肖特基TTL; (c)CMOS; (d)ECL

图 电压传输特性

2、速度或传输延迟

当设计一个系统时，逻辑系列可能的运行速度是需要考虑的重要因素。速度一般通过“传输延迟”予以说明，而传输延迟则定义为信号通过器件传输的所需要的时间。

应注意，需说明两个延迟时间。一个是 t_{plh} ，它定义了输出从低状态改变到高状态的传输延迟。而另一个是 t_{pnl} ，它定义了输出从高状态改变到低状态的传输延迟。通过一个电路的总传输延迟等于该电路中每个器件单个传输延迟之和。因此，正确地确定各器件的状态转换是十分重要的。

工厂用另外一种方法说明器件的速度，常叫做“触发频率”，像触发器这样的器件，转换到另外一种状态的最高频率，便是极大触发频率。若转换速率快于极大触发频率，将产生不确定的输出状态，自然这是必须避免的。

3、功耗

必须严格限定电源电流值时，或者当电路的热耗成为必须满足的关键条件时，功耗便特别重要。

电源电压和电源供给电路的平均电流的乘积定义为功耗。一般来说，功耗是针对电源对每个门的耗散定义的。为了估算总功耗，必须将单个门的耗乘上系统或电路中的等效门的个数。

对TTL而言，直至5MHz之前每个门的功耗几乎是不变的，然后随着频率尖锐地增长。而CMOS逻辑系列的每个门的功耗随着频率线性变化。

5、抗扰度

没有一个逻辑系统是绝对完善的。噪声是影响系统正常工作的客观现实，因此，设计人员必须认真地予以处理。显然，噪声可能产生不应有的逻辑状态，并会引起错误的系统操作。消除有害的噪声可从两方面着手处理。一个方法是努力降低噪声源，传输线技术、去耦、屏蔽是用于降

低噪声源的几种可供采用的方案；第二个方法是努力使噪声接收端不易受噪声源的。逻辑系列的抗扰度体现了在噪声的环境下正常操作的能力。通常，缓慢的逻辑系列对噪声不敏感，这是因为它们对噪声尖峰响应缓慢。

6、负载加载

设计任何一个系统，均必须将一些逻辑组件相互连接起来，以实现某种逻辑功能，即必定碰到逻辑门的加载问题。将一个逻辑器件可能驱动的逻辑器件的数目定义为负载能力。它可以通过“扇出”和“扇入”两个参数很好地予以说明。“扇出”是逻辑器件驱动能力的量度。“扇入”是逻辑器件所呈现的输入加载的量度。扇入系数 N_i 是TTL门电路输入端的数目，在制造时确定，一般 $N_i \leq 8$ 。

表1.2比较了各逻辑系列的负载特性。

表1.2 逻辑系列的负载特性

驱动级	接收端						
	TTL	S-TTL	LS-TTL	AS-TTL	ALS-TTL	CMOS(5V)	ECL
TTL	10	8	40	8	40	*>100	**
S-TTL	12	10	50	10	50	*>100	**
LS-TTL	5	4	20	4	20	*>100	**
AS-TTL	12	10	50	10	50	*>100	**
ALS-TTL	5	10	20	4	20	*>100	**
CMOS	0	0	1	0	1	>100	**
ECL	**	**	**	**	**	**	△

* 设采用了上拉电阻R

** 因逻辑电平的差异，不能直接兼容，需采用电平移位器才能互相连接。

△ 参考工厂的器件手册

§ 3.5.2 TTL、ECL、MOS互连与电平转换

一、TTL集成电路与CMOS集成电路的连接

一般CMOS电路既不能提供出大电流也不允许灌入大电流。因此两者之间的连接主要是电平转换，和驱动能力的问题。

通常在靠近TTL端，接一上拉电阻，具体接口参数与具体的芯片有关，要保证可靠的工作，一定要根据芯片的参数进行计算选型。

二· TTL集成电路与ECL电路的接口

用ECL电路驱动TTL电路时，其间的连接稍许复杂，分别为TTL至ECL和ECL至TTL的专用电平转换接口电路，可达到高速转换的要求。如XEL22，XEL23等芯片。

由于CMOS电路通常驱动能力较弱，必须先进行TTL转换后再驱动ECL转换芯片，反之亦然。

在三种逻辑电平进行转换时，主要考虑以下几点：

- 1· 电平关系，必须保证在各自的电平范围内工作，否则，不能满足正常逻辑功能，严重时烧毁芯片。
- 2· 驱动能力，必须根据器件的特性参数仔细考虑，计算和试验，否则很可能造成隐患，在电源波动，受到干扰时系统就会崩溃。
- 3· 时延特性，在高速信号进行逻辑电平转换时，会带来较大的延时，设计时一定要充分考虑其容限。
- 4· 选用电平转换逻辑芯片时应慎重考虑，反复对比。通常逻辑电平转换芯片为通用转换芯片，可靠性高，设计方便，简化了电路，但对于具体的设计电路一定要考虑以上三种情况，合理选用。

第六节 母板设计指南

母板为安装在模块插框背面的印制板，它通过板内的连接器使得每个电路板相互联接，单扳间信号的传送，电源的供给，均由母板上印制线来实现，避免了电路板间的电缆缠接，从而使整个系统具有很高的可靠性与易维护性，也保证整机背面连线的条理性。CC08机机所需要的母板的共同特点是面积大，走线长，插座多，传输信号速度快，常常需用多层印制板来实现，其设计必须按高速传输线理论及电磁干扰理论来设计。

§ 3.6.1 公司常用母板简介

CC08机交换机的母板主要有主控框母板、用户框母板、中继框母板和时钟框母板四种类型，它们都是为了单板的连接、信号的传送而设计的，由于功能不一，其特点不尽相同。

1、主控框母板

主控框母板是为连接、MPU、控制板、通信板、网板、信令板而设计的PCB板，根据用途不同它分为模块主控框母板（如A型机、C型机模块、B型机）和AM/CM控制框母板（如8模块AM/CM、16模块AM/CM、32模块AM/CM、128模块AM/CM）两大类。图1为B型机主控框母板示意图，它的面积为两个机框的尺寸，图2为32模块主控（通信）框母板示意图，它面积为一个机框尺寸，它们的共同特点是面积大，插板多且种类复杂，板内信号速度高，各种总线很多并形成大量的平行线，线间距较小，常常用6层PCB板完成。合理的布局走线，正确的信号匹配是满足整个系统工作稳定的关键。

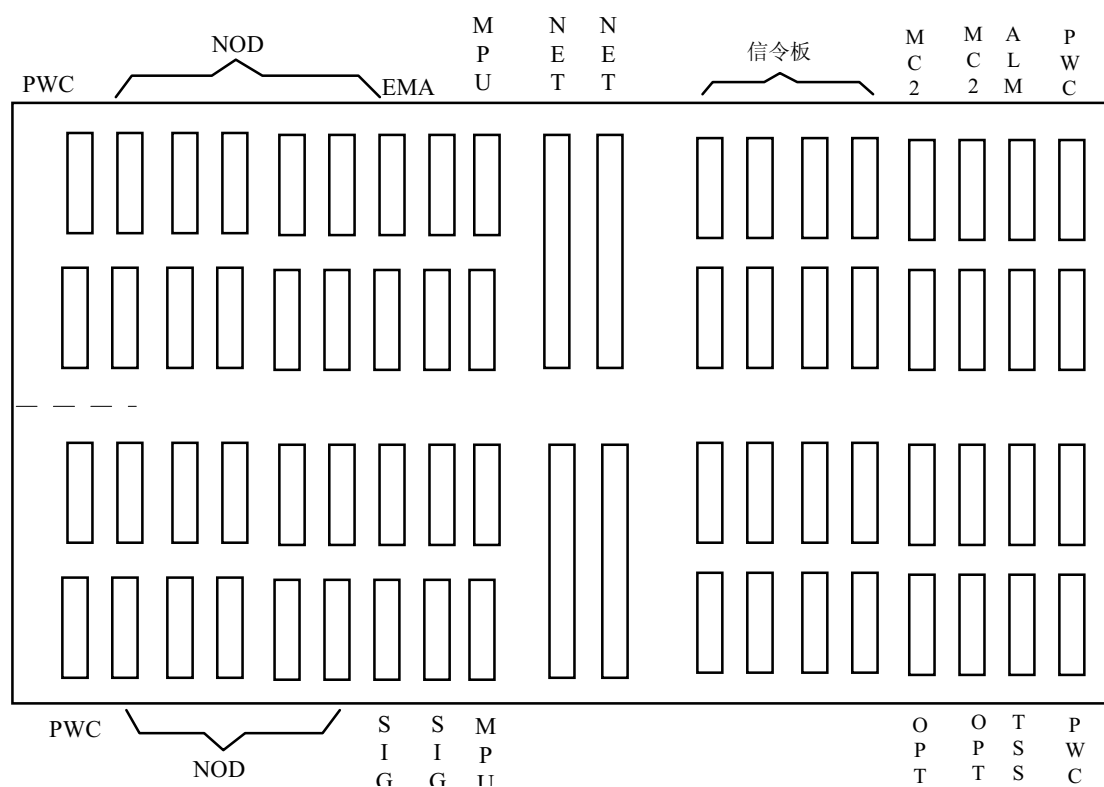


图1 B型机主控母板示意图

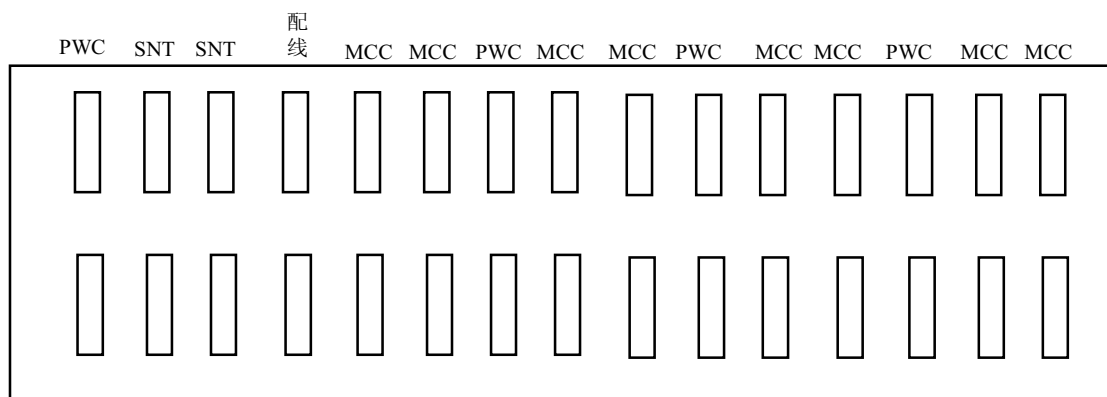


图2 32模块主控（通信）母板示意图

2、时钟框母板

时钟板（CK2,CK3），通过时钟母板与其它框相连，为整个交换机系统提供各类时钟信号。该板上提供较多的同轴插座和背板配线座，以满足对时钟的平衡或不平衡传输的要求，由于对时钟的抖动要求较高，故母板设计时，应考虑减少线间耦合及变化负载的匹配，并注意同类时钟走线的一致性。如图3，是CC08时钟框的母板示意图，它有一个机框的面积。

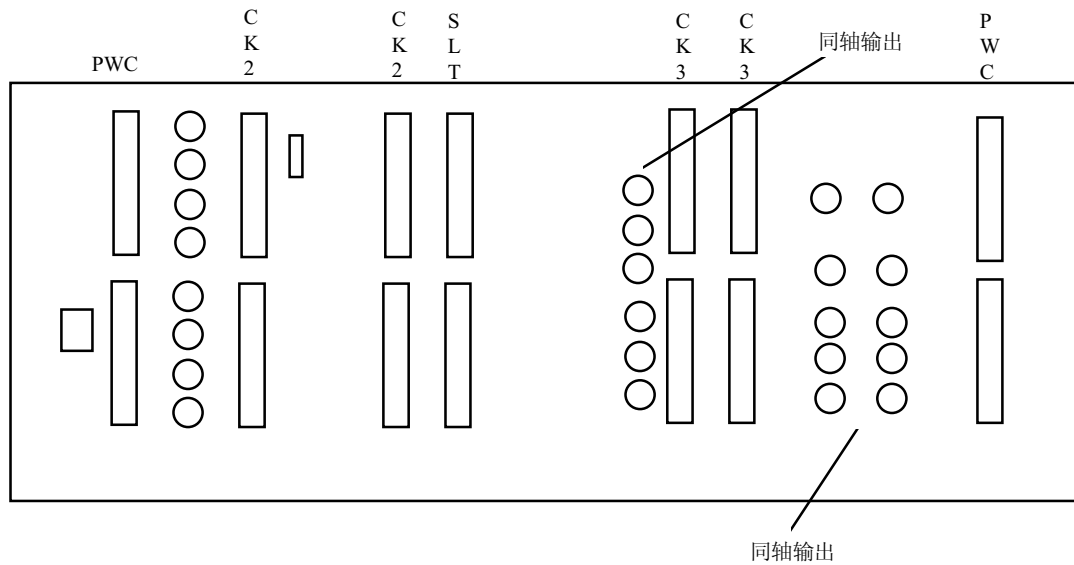


图3 时钟框母板示意图

3、中继框母板

中继框母板是用于连接DTM与外界而设计的PCB板，该板的同轴座多，背面出线，空间紧凑、信号变化幅度大，接插件的合理布局是本板设计重点，该板设计一般用4层板完成。如图4是26槽位新万门PMB，它大小为一个框的尺寸。

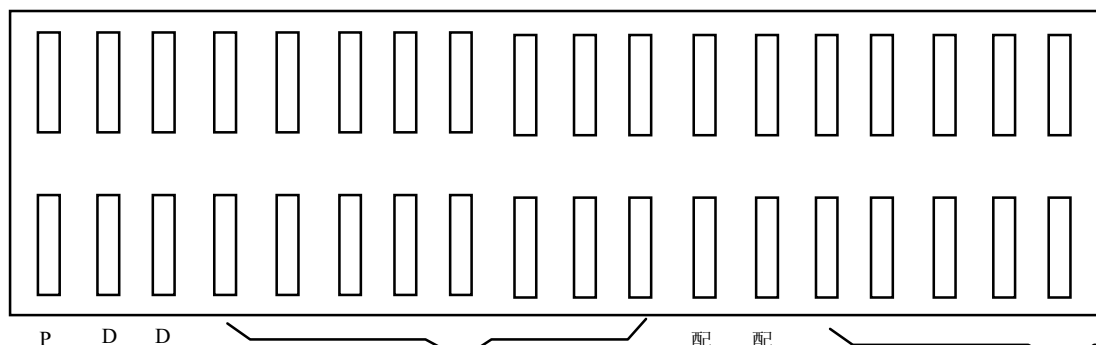


图4 中继框母板示意图

4、用户框母板

用户框母板完成用户板与外界的连接，它不仅使DRV板与各用户板间进行信号的传递，还向外提供配线的插座，该板有部分信号电压较高，合理地分配走线，减少线间耦合是该板设计的关键。

图5是26个槽位可插24块用户板（304线）的母板示意图，它一般可用2层PCB板完成（现在也有用4层）。

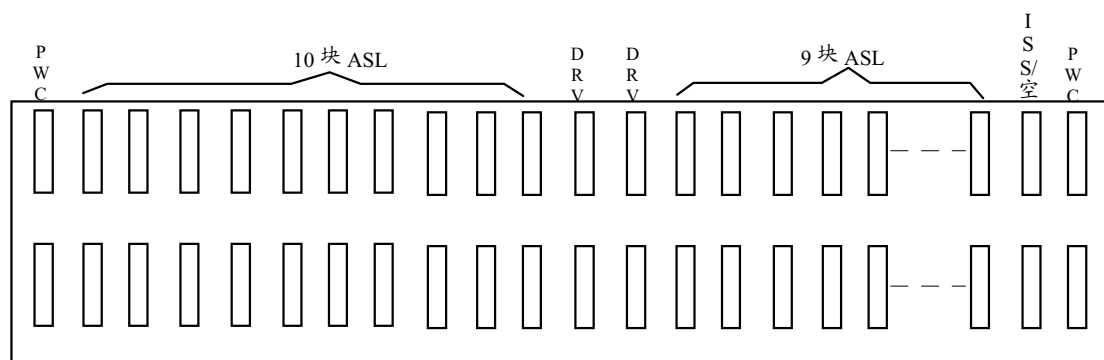


图5 CC01SLM用户框母板

CC08机及其它产品采用的母板种类很多，公司母板产品达几十种，在此不作介绍，请参见有关的文档。在设计母板时，怎样保证母板上单板布局合理、保证信号传输质量，提高生产工艺，是每一块母板设计必须考虑的。针对以往设计母板的经验，保证高速信号传输质量是母板设计的关键和难点。简单的物理逻辑连接，常常会产生严重的振铃和串扰现象，使系统的稳定性受到影响，因此有必要运用高速传输理论以及电磁干扰原理设计母板。

§ 3.6.2 高速传线理论与设计

母板上传输的信号速度快，传输距离长，当信号波长（或上、下边沿时间）与印制传输线传输时间相当时，其每根线都应认为是传输线，必须按传输线理论进行考虑，而母板繁杂的信号又必须进行合理的布局和处理，以减少串扰影响。长线信号的可靠传输和线间串扰的减少是母板设计的两个关键，下面对有关的概念进行说明。

一、信号的长线传输

图6是一个点对点传输的简单模型。

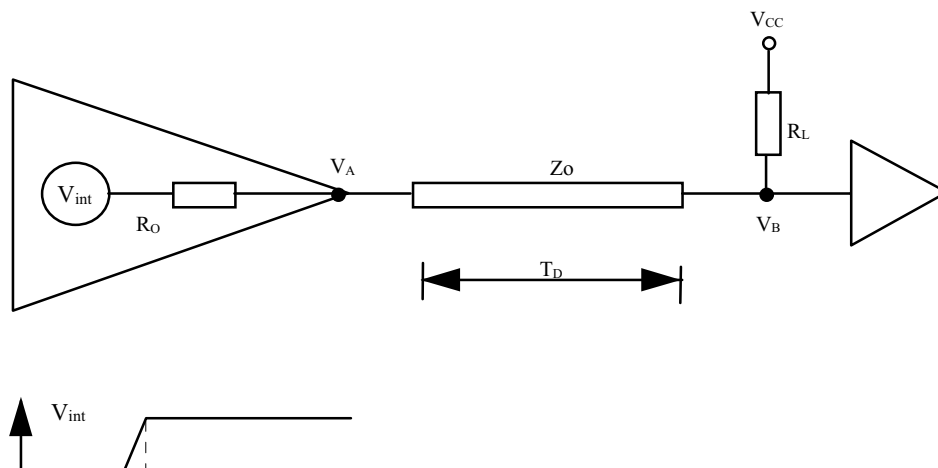


图6

在图中，驱动信号的上升时间 t_r 小于传输线的传输时延 T_D ，也就是说，完整的电平转移将发生在传输线的接收端收到脉冲之前。

这将引起反射（或振铃）。在传输线上，A点的电压变化可用（1）式来表示：

$$\Delta v_A = \Delta V_{int} \left(Z_o / (R_o + Z_o) \right) \quad (1)$$

式中， V_{int} 为驱动器输出的内部电压； R_o 为驱动门的输出阻抗； R_L 为负载阻抗； Z_o 为传输线的特征阻抗； V_A 为传输线输入端的源电压。

因为 R_o 与传输线相比较小，A点的电压变化 ΔV_A 将约等于内部电压 ΔV_{int} 的变化。电压变化将沿着传输线传播下去，北过传输线的传输时延 T_D 后，就可在B点处观察到电压变化。

在B点，根据（2）有一部分信号被反射回A点：

$$\rho_L = (R_L - Z_o) / (R_L + Z_o) \quad (2)$$

式中， ρ_L 称为电压反射系数，它实际上是反射电压与入射电压之比。

考察（2）式，可以发现 $-1 \leq \rho \leq +1$ 。显而易见，如果 $R_L = Z_o$ 就不会发生反射。也就是说，只要根据传输线的特征阻抗进行终端匹配，就能消除反射。从原理上说，反射波的幅度可以大到入射电压的幅度，极性可以为正，也可以为负。

反射系数的概念既适用于传输线的发送端，又适用于接收。也即：

$$\rho_s = (R_o - Z_o) / (R_o + Z_o) \quad (3)$$

高速逻辑电路板的分布特性如果处理不当，这时常会导致振铃现象，这是未经终端匹配的传输线从其终端多次反射的结果。未经终端匹配的传输线没有负载阻抗 $R_L = \infty$ ，因而其阻抗是失配的。这种失配传输线与高摆率器件相连接时，波形在传输线中的传播情况如图7所示（作为一个例

子进行理解)。在 $t=0$ 之前，传输线上各点都为2.5V的稳态直流电压（图7a）。在 $t=0$ 时，在A点产生从2.5V到0.5V的初始电压过渡即A点电压下降到0.5V，向B点送出一个2V的负脉冲（图7b）。在时间 T_D 之后，信号达到了B点，被负载反射回来。负载的反射系数为 ρ_L 。

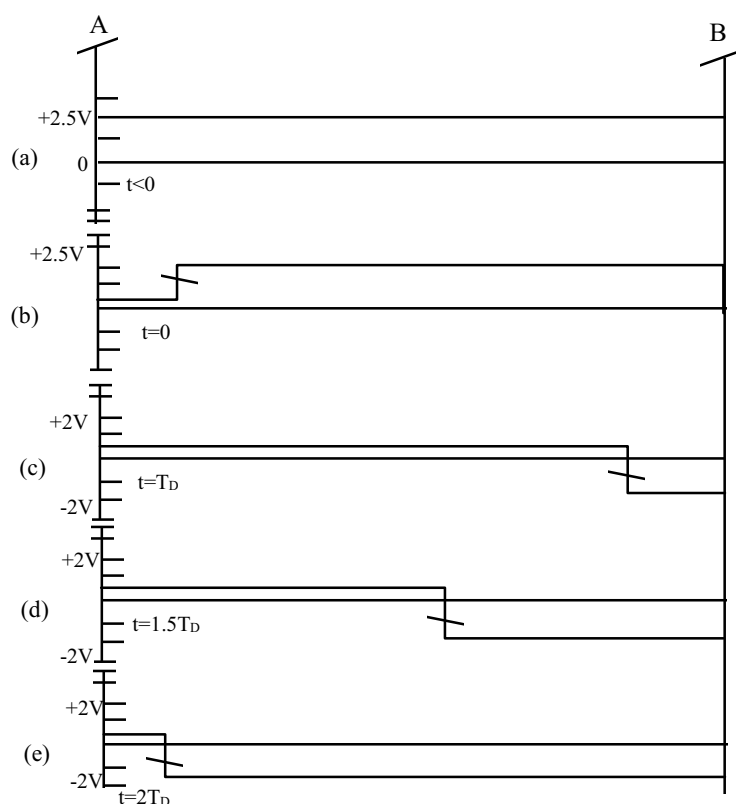


图7

在B点，器件的输入阻抗相对于 Z_0 很高； R_L 可视为无穷大。将其代入（2）式，反射系数 ρ_L 约为+1。也就是说，从负载反射的电压约等于入射电压。在 $t=T_D$ 时，负脉冲从B点反射回来，与传输线上的0.5V相迭加，向A点发出1.5V的负脉冲（图7c）。反射波沿着信号通道朝A点反向传播（图7d）。

在传线的发送端A点， R_0 近似为0，重复上述运算，可得到-1的源反射系数 ρ_s 。也就是说，源和负载都存在着反射，但源所反射的是其入射波的反向波（图7e）。

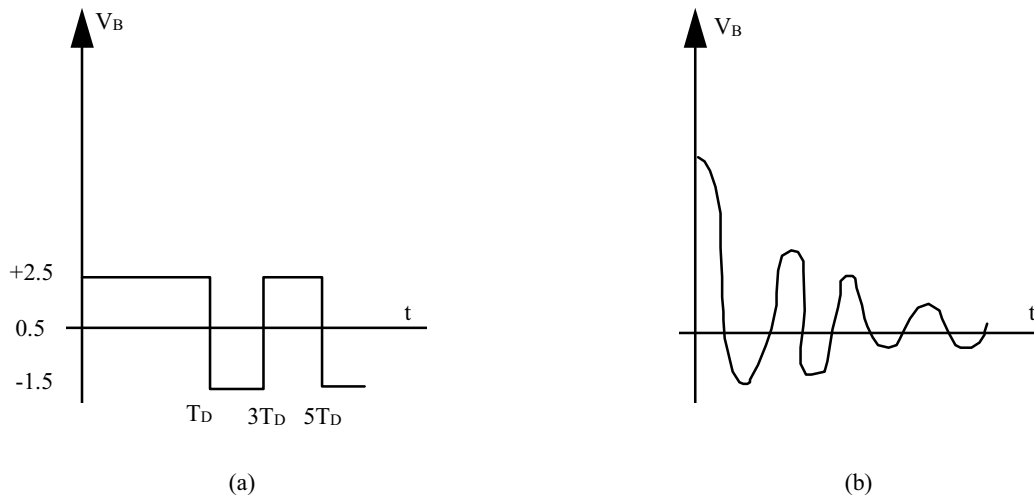


图8

从B点信号的变化来看， $t=0$ 时的单一阶跃电压过渡引发了振荡，其电压摆幅为4V，即早先电压过渡的两倍。电压加倍的原因大致上由于B点的电压是同一点上的入射波与反射波之和（图8a）。事实上，由于实际电路板的非理想特性（有限的输入阻抗和输出阻抗，传线的损失等等）， ρ_L 将小于+1， ρ_s 也将大于-1。结果，反射将越来越小，成为我们所熟悉的衰减振荡现象（图8b）。

如果振铃幅度足够大的话，就会在其后继器件的输入端上产生非法的电平过渡，并有可能出现影响逻辑设计的寄生逻辑状态。在某些情况下，振铃的幅度可能会大得足以以后继器件的输入端损坏。

从上面的分析可看出，要使信号传质量达到要求，必须采取措施消除或最大地减少反射出现的振铃现象，最直接的办法就是匹配，有关匹配的方法在下一节讲述。作为设计人员来说，必须确定是否该加匹配，匹配的值应怎样取，这两项参数是由信号传输时延大小和传输线的特性阻抗决定的。这里作一讨论。

信号传输的时延估算：

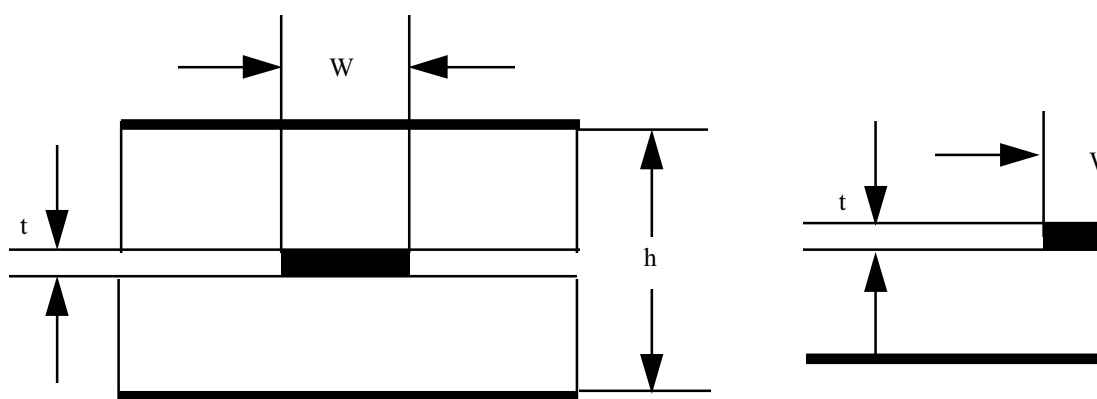
假定现代高速逻辑器件的典型上升时间为2ns，那么当电路板的传输时延到达1ns时其线路长度为何？对于具有连续地线层且信号走线在其邻近层的印刷电路板，传时延仅取决于一个变量，这就是电路板材料的电介质常数。延迟时间由下式决定：

$$t_{PD}=3.33660(0.475\epsilon_R+0.65)^{1/2}(\text{ns/m}) \quad (4)$$

对于通常用的FR4材料制成的板子，电介质常数 ϵ_r 为4.7至4.9。如果取其平均值4.8，就可求得每米长度的 t_{PD} 为5.73ns，也即每17.45cm的时延为1ns。所以作为一种习惯性做法，一旦线路长度超过17cm就应当将其作为传输线来考虑并采取相应的措施。

传输线特征阻抗估算

传输线的特征阻抗是传输线的一个重要指标，不同的条件有不同的计算方法，针对母板设计的实际情况，可近似归纳成stripline和Microstrip两类。如图9，图10。



对于微波带状线（stripline），特征阻抗：

$$Z_o = \frac{60}{\sqrt{\epsilon_r}} \ln \frac{4h}{0.67\pi w(0.8 + t/w)} \quad \Omega \quad (5)$$

对于微带线（microstrip），特征阻抗：

$$Z_o = \frac{87}{\sqrt{\epsilon_r + 1.41}} \ln \frac{5.98h}{0.8w + A} \quad \Omega \quad (6)$$

其中， ϵ_r 为PCB板介质的介电常数。

根据（5）、（6）式和实际情况，就可估算出匹配时，匹配电阻的取值。

二、信号的串扰

串扰是一导体的对其邻近导体不希望的耦合。随着信号摆率的上升，平行线的增多，增长，串扰将成为越来越严重的问题，它可能使原理上毫无瑕疵的逻辑设计毁于一旦，串音通常作用于

靠近的电路和导体上，并用电路和导线的互容和互感来表征。对带有大面积地平面的PCB带状线来说互容耦合为主，为了计算方便，可简单地用下式对耦合电压进行估算。

$$V_c = C \cdot V_1 \quad (7)$$

(7)式中， V_1 为激励源印制线上的激励电压（平均值）

C 为线间耦合系数

V_c 受激励源印制线上的感应电压（平均值）

有关线间耦合系数的计算可参见《电磁兼容性原理及运用》，国防出版社1996.4 (ISBN 7-118-01514-8)。下面提供线间耦合系数 C 的查表。

针对母板，可等效为PCB两层间的PCB线，图9和PCB表面的一对微带线图10，若假设两线均匹配，可得表1，表2。

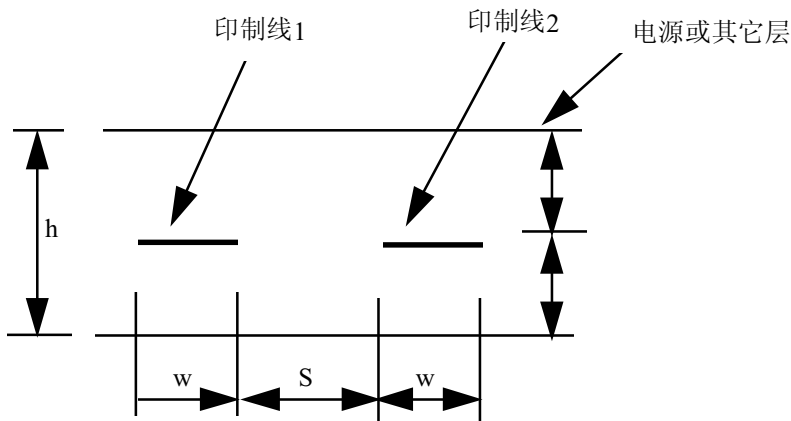


图9

表1 图9带状线串音系数

w/h	s/h	$\epsilon_r=4$	$\epsilon_r=2$
		C	C
0.3	0.01	0.55	0.55
0.3	0.10	0.38	0.38
0.3	0.50	0.09	0.09

0.5

0.01

0.48

0.480.50.100.270.270.50.500.080.08

1.0

接上表

				0.01
				0.35
0.351.00.100.200.20				
1.0	0.50	0.05	0.05	

表2 图10微带线串音系数C

w/h	s/h	$\epsilon_r=4$	$\epsilon_r=2$
		C	C
0.2	0.01	0.55	0.55
0.2	0.05	0.56	0.56
0.2	0.20	0.42	0.42
0.2	0.50	0.27	0.27
0.2	1.00	0.15	0.15
0.5	0.05	0.52	0.52
0.5	0.20	0.38	0.38
0.5	0.50	0.26	0.26
0.5	1.00	0.15	0.15
1.0	0.05	0.45	0.45
1.0	0.20	0.33	0.33
1.0	0.50	0.23	0.23
1.0	1.00	0.14	0.14
2.0	0.05	0.37	0.37
2.0	0.20	0.26	0.26
2.0	0.50	0.18	0.18
2.0	1.00	0.12	0.12

从上面看到，欲减小耦合干扰，即串扰，就应使耦合系数C降至最低，具体措施见3.6.4节。

§ 3.6.3 总线阻抗匹配、总线驱动与端接

消除信号反射最直接的方法是根据传输线的特征阻抗进行匹配，对于一般的总线驱动电路，在发送端和接收端都可以进行匹配。

1、终端并联匹配：

根据 § 3.6.2 节的 (2) 式，当终端电阻 $R_L = Z_0$ 时，反射系数 $\rho_L = 0$ ，这意味着传输线上不再有反射或失真，除去时延 T_D 之外，线路将变得如同直流电路一般。故终端并联匹配电阻一般取 Z_0 的值。应当注意的是，终端匹配电阻应尽可能靠近接收电路。

2、始端串联匹配

对于始端的串联匹配，其匹配的原理是使得 $\rho_s = 0$ 和 $\rho_L = +1$ 为此，应使得 R_L 等于无穷大（不加任何匹配）在信号源端串入电阻，以使得总的源阻抗等于传输线的特性阻抗。

$$R_s + R_o = Z_{oL} \quad (8)$$

然而，使 $R_s + R_o$ 等于 Z_{oL} 会产生信号分压，结果有一半的信号幅度电压降落在传输线上，另一半则降落在 R_s 和 R_o 上。因此，在采用串联终端匹配的情况下，传输波形的幅度为无匹配情形之半。

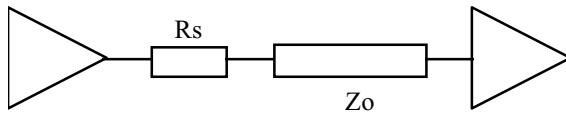


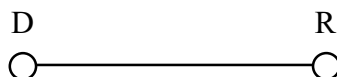
图11

更为有趣的是，未经匹配的传输线接收端能够精确地补偿上面的半幅度现象。其原因如下：在接收端，在收到半幅度波形的同时又反射出半幅度波形。但切记这是两个不同的波形，它们的幅度将在反射点相迭加。由于迭加的结果，我们在传线的接收端就只能观察到全幅度的波形。

串联终端匹配的主要缺点是接收信号的负载器件必须位于线路的终端而不能沿着线路分布。与并联匹配相比，串联终端匹配的突出优点是无需与电源相连接，并可保护电路过流。

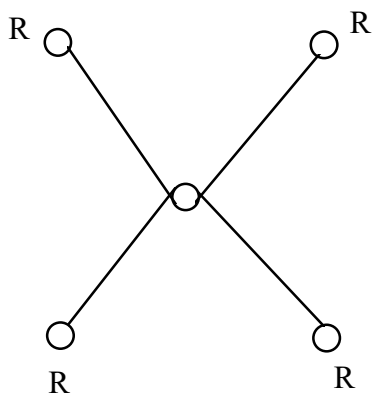
在主板设计中，PCB板上的连线多种多样，针对不同的拓补结构，应采用不同类型的匹配技术来解决信号的反射问题。

(1) “点一点”结构



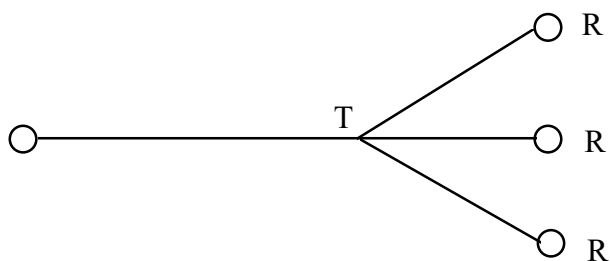
常见的一种驱动端接一个负载端，可在驱动端串小阻值电阻或并联一个阻值为 Z_0 的电阻在终端。

(2) 星形结构



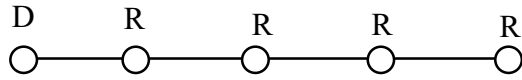
对于星形结构来讲，其负载分散，各负载之间影响小，但需提高驱动能力，减小驱动端的输出阻抗，可采用始端串阻方式实现。需要注意的是，这种方法改善有限，因此一般不要采用此种结构。

(3) 远端簇型结构



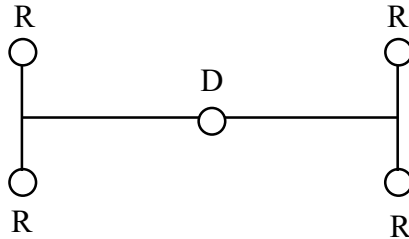
远端簇形结构是星形结构的一种变化，当各负载端接近时，需考虑相互之间的影响。在布线时应减小接头T到各负载端的距离。这种结构可在各负载终端处加电阻并联匹配。电阻阻值为 $n \cdot Z_0$ (n 为分枝数)。

(4) 菊花链结构



菊花链结构中负载相互有一定影响，但总线传输较稳定，常在最后一个终端，根据特性阻抗加上/下拉匹配电阻或并联AC方式匹配，不能用串阻方式解决反射问题。推荐使用这种方式。

(5) H型结构



H型结构也是星形结构的一种变化，各负载干扰小，比星形结构易驱动，可在各负载端加并联上/下拉电阻或并联AC方式。

在主板设计时，需要了解长线的驱动问题对于这类问题，只考虑直流负载，也即常见的扇出和扇入数。未免难以反映出代高速逻辑器件的问题。在高速逻辑电路的设计中，更为有意义的是输入和输出的交流负载。

由于器件的输入电容将影响到逻辑电路的总体性能，因此当选取某种器件来进行设计时，尤其应当注意其输入电容。为保证特定的性能，某一器件所驱动的总负载电容，包括导线的分布电容在内，应当不超过该器件所规定的容性负载。许多高速逻辑器件的最大负载为50pF。作为一条经验之谈，为保证最佳速度性能和负载性能，任何一种逻辑器件的最大负载都不应当超过4~6个。当然，目前市场上已有一些输入驱动能力强的高摆率器件。

另外，在选器件时，只要满足要求，尽可能选低速器件。与主板相连的还有许多差分电路，当线路过长时也需要进行端接匹配，具体见附录1。

为了保证信号的传输，线路特性阻抗必须考虑一致，由于有诸多限制，实际很难达到，不过，主板设计时应注意：

- (1) 线宽尽量保证一致；
- (2) 线上过孔尽可能少，尽量把线走在同一层；
- (3) 少走折线、弯线。

§ 3.6.4 布线策略与电磁干扰

设计一个优质的母板，不仅要解决好信号的传输问题，还要消除电磁耦合的影响，即减少串扰，从3.6.2节知道，只要减小耦合系数，就可达到降低相互影响的目标，具体的方法很简单。

- 1、扩大线间距
- 2、尽可能少走平行线
- 3、减少线长
- 4、不走环形线

显然，由于电路板的面积，总体安装位置等的限制，按上述原则设计会受到一定的限制。因此，在设计时，需要对布局和布线分别考虑，最终目的就是要尽可能地达到上述要求。

一、布局

当总体方案定后，单板的种类、数量已定，所有送到母板上的信号也定下来了，因此，布局就是要调整单板的位置，以及信号的出线位置（有的单板不能改变出线位置）使整个板面尽可能满足如下条件：

- 1、总的连线尽可能短，关键信号线最短
- 2、交叉线最少，过孔最少
- 3、地线层和电源层没有连线
- 4、高电压、大电流信号与小电流，低电压的弱信号完全分开
- 5、模拟信号与数字信号分开
- 6、高频信号与低频信号分开
- 7、高频数字信号的间隔要大
- 8、有可能情况下，减小电源地层与信号层层距布线

二、布线

布线时应考虑如下因素：

- 1、增加线间距，减少平行走线长度
- 2、增加线宽度，降低其特性阻抗
- 3、重要信号间，可采用平行地线的方法隔离
- 4、尽可能少折线，不走90°折线
- 5、少走过孔
- 6、重要线不要走插座脚间穿过，频率高的线也应尽量避免

实际中，会碰到各种情况的串扰影响，除了上述的一些原则外，还可考虑屏蔽、抵消等方法，具体的可参看有关电磁干扰的书藉。在此不在论述。

第七节 单板软件开发

§ 3.7.1、常用CPU介绍

C&C08数字程控交换采用了三级程控分散控制结构，一个2000门模块采用了近200个嵌入处理器，软、硬件协同完成系统功能。单板软件在系统中，主要完成各种硬件电路控制、信号检测功能，同时还必须具有实现板间通讯功能。

目前，我司产品中使用许多厂家不同系列的处理器，其性能、处理能力、应用场合差别较大，主要有以下类型：

1、MCS—51系列

以ATMEL公司AT89C51使用量最大，因所有ASL板均采用该芯片，其它一些控制功能较简单的板也采用，如光板FBC、OPT等；其次8031在除ASL外的大部分单板中被采用，如DRV、NOD、DTM等常用板；部分单板对处理能力有较高的要求，已开始采用Intel公司升级产品251系列，如DTF、MFC板均采用了80C251SB。

2、X86系列

主要有80C186/188、80386EX两种，386SC、486、586在主控板上也有使用，188在公司早期开发的一些单板MC2、MC4、NO7、FCP等使用较多，186在96年改板中使用较多，386EX作为嵌入式处理器正越来越广泛地在各项目中使用。

3、683XX系列

主要有68302、68360各种型号，68040等，其68EN360，68FE360为8MH360使用较多，其主要优势在于处理器片上包含了QUICC协议处理部分，在通讯领域中使用较多。

4、其它

其它厂商处理器现使用较少，主要是在早期产品中使用，如Zilog Z80系列，Intel 8098等，现已停用。

§ 3.7.2 开发环境

目前，单板软件开发语言主要采用C汇编两种，个别情况有用PLM的。

对MCS51系列，有C51编译器、ASM51汇编程序，主要是Intel公司工发工具，地186系列，现可采用Intel IC86、ASM86，也可采用BC/CH，VC/CH等常用开发语言。386EX现主要用MCC386、HIGHC两种编译器。

§ 3.7.3 单板软件调试

调试有两种方法：硬件仿真器、软件调试器。

硬件仿真器主要用于开发初期目标板硬件系统尚未稳定期间的软、硬件调试。当目标板稳定后仍可用仿真器调试，查找软件疑难问题。缺点是调试时借助其强大的硬件查错功能、硬件断点、硬件跟踪功能，常见人仿真器有北京三环公司产品MDS—55系列不太方便，需要采用仿真器连到目标板上，有8051、80186/188、8098、Z80等型号，Applied Microsystems Corporation公司产品Code TAP 186XL，386EX等，Microtek 386仿真器，HP公司360仿真器等等。

软件调试器一般是通过PC串口与目标板串口相连，通过目标板上监控程序（monitor）下载程序在目标板中进行调试，重点调试软件的流程功能等，其前提条件下目标板硬件应已经完全没有问题，至少是CPU程序、RAM、串口等部分可正常运行。常见软件调试器主要是Microtec Research公司的XRAY，有386EX和360两种版本在公司使用，另外现刚刚引入Paradigm DEBUG RT186，可调试186、386EX实模式软件，其界面操作Borland Turbo Debugger完全相同，可加快软件调试进度。

§ 3.7.4 编程规范

现在微系统软件研究部已初步拟订关于软件编程的初步规范，正在试行中，单板软件库也正在建设中，软件开发共享库建立起来后将大大促进软件开发的效率。

单板软件编程规范(试行)

目的：为了开发人员之间更好地进行交流，提高代码的可读性，可维护性，特制订本规范，作为程序编写的指导文件。本规范只涉及到源码书写的格式，希望能有较统一的编程风格。将来若有和公司相关规定冲突的地方，本规范有可能作适当修改。

范围：本规范暂时适用于微系统软件研究部开发人员。

内容：

1. 模块描述

模块是为了实现某一功能的函数的集合，文件名使用缺省的后缀，在每一模块的开头应有如下的描述体：

```
/******  
* PROJECT CODE: 项目代号或名称 *  
* CREATE DATE : 创建日期 *
```

```

* CREATED BY   : 创建人                                     *
* FUNCTION     : 模块功能                                   *
* MODIFY DATE  : 修改日期                                   *
* DOCUMENT    : 参考文档                                   *
* OTHERS      : 程序员认为应做特别说明的部分, 如特别的编译开关 *

```

*****/

不同的修改人应在修改的地方加上适当的注释, 包括修改人的姓名。另外, 如有必要, 要注明模块的工作平台, 如单板OS、DOS、WINDOWS等。注明适用的编译器和编译模式。

2. 函数描述

函数是组成模块的单元, 一般用来完成某一算法或控制等。在每一函数的开头应有如下的描述体:

/******/

```

* FUNCTION NAME: 函数名称                                     *
* CREATE DATE  : 创建日期                                   *
* CREATED BY   : 创建人                                     *
* FUNCTION     : 函数功能                                   *
* MODIFY DATE  : 修改日期                                   *
* INPUT       : 输入参数类型(逐个说明)                   *
* OUTPUT      : 输出参数类型(逐个说明)                   *
* RETURN      : 返回信息                                   *

```

*****/

可选的描述有:

```

* RECEIVED MESSAGES: 收到的消息                           *
* SENT MESSAGES   : 发送的消息                             *
* DATABASE ACCESS : 存取的数据库                           *
* CALLED BY      : 该函数的调用者                           *
* PROCEDURES CALLED: 调用的过程                             *
* RECEIVED PRIMITIVES: 收到的原语                           *
* SENT PRIMITIVES : 发送的原语                             *

```

及其它程序员认为应有的描述。标题可以只大写第一个字母。例如: **Function Name:**

3. 命名规则:

A) 函数: 函数名应能体现该函数完成的功能, 关键部分应采用完整的单词, 辅助部分若太长可采用缩写, 缩写应符合英文的规范。每个单词的第一个字母大写。如: ShowPoints, CtrlDestBoard, SendResetMsg 等。

B) 变量: 变量的命名规则部分采用匈牙利命名规则(鼓励完全使用匈牙利命名规则)。变量的第一个或前两个字母小写, 表示其数据类型, 其后每个词的第一个字母大写。推荐的类型前缀如下:

前缀	含义	前缀	含义
a	数组	n	short int
b	BOOL	np	短指针

byBYTEp指针 ccharlLONGcb字节记数 lp长指针 cr颜色参考值 s串 cx,cy短型 (x,y长度的记数) sz以零结尾的串 dwDWORDtm文本 fn函数 wWORDhHANDLEx,y短型 (x或y的坐标) iintg_全局变量

m_	类的数据成员	uc	unsigned char
----	--------	----	---------------

如iCurrentValue, uTransitionCount等。对于其他复合类型或自定义类型, 请用适当的前缀来表示。除局部循环变量外, 不鼓励单个字母的变量名。

对于常用的类型定义, 尽量使用WORD、BOOL、LPWORD、VOID、FAR、NEAR等惯用写法, 避免使用char、long、void、far、near等小写格式。不使用_UC、_UL等华为公司以前一些人的习惯写法。

C) 结构: 结构的定义有两个名称, 一个是该结构的类型名, 一个是变量名。按照C语言的语法, 这两个名称都是可选的, 但二者必有其一。我们要求写类型名, 类型名以tag做前缀。下面是一个例子:

```

struct tagVBXEVENT
{
    HCTL      hControl;
    HWND     hWindow;
    int       nID;
    int       nEventIndex;
    LPCSTRlpEventName;
    int       nNumParams;
    LPVOID   lpParamList;
} veMyEvent;
tagVBXEVENT  veMyEvent[MAXEVENTTYPE], *lpVBXEvent;

```

对于程序中常用的结构, 希望能使用 typedef 定义, 格式如下:

```

typedef struct tagMYSTRUCT
{
    struct members .....
} TMYSTRUCT,* PTMYSTRUCT,FAR * LPTMYSTRUCT;
struct 后的类型名有tag前缀, 自定义的结构名称一律用大写字母, 前面可以加一大写的T。而结构类型变量定义则可以写为:
TMYSTRUCT variablename;
并可在定义 MYSTRUCT_S 同时根据需要, 定义其指针, 远指针和尺寸常量:
typedef tagMYSTRUCT
{
    struct members ... ..
} TMYSTRUCT,* PTMYSTRUCT,FAR * LPTMYSTRUCT;
#define MYSTRUCTSIZE  sizeof ( TMYSTRUCT );

```

结构变量的命名, 建议采用如下方式:

从结构名中, 取出二至三个词的首字母作为代表此结构的缩写, 小写作为变量前缀。例如: (取自Microsoft Windows示例)

```

struct OPENFILENAME  ofnMyFile
strcut CHOOSECOLOR ccScreenColor

```

D) 联合: 联合的命名规则和结构相似, 如:

```

union tagMYEXAMPLE

```

```

{
    int    iInteger;
    long  lLongInt;
} myExample;

typedef tagMYEXAMPLE
{
    int    iInteger;
    long  lLongInt;
} MYEXAMPLE,* PMYEXAMPLE,FAR * LPMYEXAMPLE;

```

4. 书写风格:

A) 函数: 函数的返回类型一定要写, 不管它是否默认类型, 函数的参数之间应用一逗号加一空格隔开, 若有多个参数, 应排列整齐。例如:

```

int SendResetMsg( PTLAPENTITY pLAPEntity, int iErrorNo )
{
    int iTempValue;
    .
    .
    .
}

```

函数的类型和上下两个括号应从第一列开始, 函数的第一行应缩进一个TAB, 不得用空格缩进。(按大多数程序范例, TAB为四个字符宽, 我们规定: TAB为四个字符宽。)

B) 语句: 循环语句和if语句等块语句的第一个大括号 ‘{’ 可跟在第一行的后面, 接下来的语句应缩进一个TAB, 如:

```

for ( count = 0 ; count < MAXLINE ; count++ ) {
    if ( (count % PAGELINE) == 0 ) {
        .
        .
        .
    }
    .
    .
}

```

也可另起一行, 如:

```

for ( count = 0 ; count < MAXLINE ; count++ )
{
    if ( (count % PAGELINE) == 0 )
    {
        .
        .
        .
    }
}

```

```
.  
. }  
}
```

两种写法在世界著名的程序员手下均可见到，我们尊重个人的习惯，但推荐使用后一种写法。

复杂表达式(两个运算符以上，含两个)必须用括号区分运算顺序，**运算符的前后应各有一空格**，习惯写在一行的几个语句(如IF语句)，中间应有一空格，其它语句不鼓励写在同一行。

空格加在适当的地方，如 **if (; for (;) {;**

语句的上下对齐也可使程序便于阅读，如：

```
myStruct.iFirstNumber      = 0;  
myStruct.lSecondNumber    = 1;  
myStruct.pThePoint        = NULL;
```

C) 常量：常量一般情况下可用宏定义，用大写的方式，单词之间用下划线隔开 如：

```
#define MAX_LINE    100  
#define PI          3.1415926
```

不鼓励在程序中出现大量的数字常数。

注：对于一些有必要说明的缩写，可以在模块描述内加以说明。

5. 头文件：头文件一般包括了数据结构的定义，函数原形的说明，宏定义等，不许包含函数体和变量实体，文件名使用缺省的后缀.h，不使用类似.DEF等非标准的后缀名，头文件的开始可包括如下的注释：

```
/*  
* CREATE DATE: 创建日期  
* CREATED BY : 创建人  
* MODIFIED BY : 修改人  
* USED BY : 由哪些模块使用  
*/
```

为了避免重编译，应加上条件编译语句，如文件headfile.h应包含下列语句：

```
#ifndef __HEADFILE_H  
#define __HEADFILE_H  
.  
.  
.  
#endif
```

6.1.1 预编译宏：对于代码中使用到的预编译宏，应在文件的适当处加以说明。对于连续多行的宏，在书写上也采用缩进的格式。例如：

```
#ifdef _BORLANDC_  
#define FAR      far  
#define NEAR    near  
#ifdef OS_DEBUG  
#define HHH      0x11  
#define JJJ      0x22  
#undef KKK  
#endif  
#else  
#define FAR
```

```

#define NEAR
#endif
对于夹在代码中的编译开关，在书写格式上当作源代码一样处理。例如：
...
lpPrim = (LPTL2L1PRIM)GetMessage( &queL2L1 );
if( lpPrim == NULL )
{
    #if OS_DEBUG
    Printf(“Queue L2->L1 is empty!”);
    #endif
    return;
}
...

```

7. **注释：**注释是源码中非常重要的部分，不应少于源码行数的15%，我们希望能达到25%或更多。注释用中英文都可以，但应易读易懂。不要在一个语句的中间插入注释。

本规范内容符合公司有关文档要求，参照了北京研究部的相关文档，并听取了大家的意见，希望诸位在软件开发过程中，能主动遵照本规范，写出高质量的源码！

附 页

本附页的内容不是“软件编程规范”的一部分，但对于较大的软件，值得推荐。

1. 接口函数（被其它模块调用的函数）及接口变量（被其它模块引用的变量），其命名可增加前缀表明其所归属的模块。

如：**NLS_MsgProc, NLP_GetPHTime,**
NLS_LogcnToConnid ...

由于C的函数都是全程的，其它函数也建议增加前缀来避免重名或混淆。

2. 头文件分开定义

A) 自用头文件：定义模块自用的函数，结构，变量等。

B) 输出头文件：用于该模块向外输出，即别的模块在使用该模块提供功能时需要引用的头文件，包含该模块的输出函数和输出数据结构，输出常量。

3. 建议在使用有一定关联的成组常量时使用枚举定义：

```

typedef enum
{
    MONDAY,
    .....
    SUNDAY,
} WEEKDAY;

```

在有特殊数值要求时在枚举内定义：

```

typedef enum
{
    MONDAY = 1,
    .....
    SUNDAY = 7,
} WEEKDAY;

```


好处：容易理解有关联的常量；枚举变量可以当作整型变量使用。

六、参考书目

- 1、3H Text View使用手册
- 2、ASM86 IC86使用手册
- 3、ASM51使用手册
- 4、Paradigm CT186/RT186 ^{User's Guide}

第十节 DSP技术

§ 3.10.1 DSP概述

1、基本概念

通常DSP（Digital Signal Processing）是指数字信号处理。DSP芯片是专用的数字信号处理器（Digital Signal Processor），它采用哈佛结数、流水线作业方式的并行处理技术，有专用的指令系统，尤其适用于数字信号的算法实现，编码变换等处理。

2、DSP的发展历史和现状

（1）DSP发展的历史背景

现代信号处理起源于17—18世纪的数学，并以1807年12月21日法国工程师J、B、Fourier提出FT变换为标志。我们现在讨论的数字信号处理，是随着数字电子计算机的发展而发展起来。在数字信号处理的初期（本世纪50~60年代），人们只是在通用的数字计算机上进行算法的研究和处理系统的模拟与仿真，1965年库利—图基（Cooley Tukey）提出快速付氏变换FFT是数字信号处理发展史上的一个里程碑。同时随着FFT及其它算法的提出，也促进了专用数字处理硬件的发展，并最终导致高速高位DSP的出现。

一般认为：70年代后期推出的Intel 2920是第一块脱离了通用型微处理器结构的DSP芯片，1980年前后推出的 μ PD7720具有专门的硬件乘法器，从而被认为是第一块单片DSP器件，1983年Ti推出的TMS系列，标准着实时数字信号处理领域的重大突破。

（2）DSP主要机种与生产商

DSP主要分浮点与定点运算两类

定点运算有Analog Derives ADSP2100系列，AT&T DSP 16系列 Motorola DSP5600系列、NEC μ PD77220、TMS320C2x/c5x/c54x系列等；

浮点运算有Analog Derives ADSP 2120、AT&T DSP32c/3210、Motorola DSP 96002、TMS3200 c3x/c4x等。

目前DSP主要生产商有：Analog Devices公司，其主要产品ADSP2100定点系列和ADSP21020定点系列；Motorola公司其主要产品：Motorola DSP 5600定点系列和Motorola 9600浮

点系列；Texas Instruments公司，其主要产品：TMS320c2x/c5x/c54x定点和TMS3x/4x系列浮点。并且，Ti公司宣传即将推出TMS320C6X/C62XX系列产品，速度1000MPIS以上。Analog Devices公司也将推出高性能的SHARC产品。

§ 3.10.2 DSP的特点与应用

1、DSP的特点

(1) DSP总线结构都采用哈佛结构或改进的哈佛结构。

通用的微处理采用冯·诺依曼结构，即程序指令和数据共用一个存储空间及单一的地址和数据总线。为进一步提高运算速度，以满足实时信号处理的要求，当前的DSP芯片采用哈佛结构。所谓哈佛结构，是将指令和数据的存储空间分开。这样使得处理指令和处理数据可以同时进行，从而大大提高处理效率。

(2) 采用流水线技术

DSP大多采用了流水线技术。计算机在进行一条指令时，要经过取指、译码、访问数据、执行等几个步骤，需若干个指令周期才能完成。流水线技术将各指令和执行时间可以重叠起来。即第一条指令取指后，在进行译码的同时，可进行第二条指令的取指；第一条指令访问数据时，第二条指令译码……。这样尽管每条指令的执行时间仍然是几个指令周期，但由于用了流水线的作业方式，使得总体处理速度大大提高。而DSP所采用的指令与数据存储空间分开的哈佛结构，为实现流水线作业方式提供了方便。

(3) 具有硬件乘法器和MAC单元

在数字信号处理算法中，乘法和累加是最基本和大量的运算，例如在卷积运算、数字滤波、快速付里叶变换、相关计算、矩阵运算等算法中大多都有乘加（乘法和累加）运算。通用的乘法运算是采用软件实现的，往往一次乘法运算需要若干个机器周期才能完成。DSP中都设置了硬件乘法器和MAC乘加（乘法并累加）一类指令，取两个操作数到乘法器中作乘法，并将乘积加到累加器中，这些操作在DSP芯片中往往可以在单个在指令周期内完成，使得DSP作乘和累加这种数字信号处理基本运算的速度大为提高。

(4) 具有循环寻址（circular addressing）及位倒序（bit reverse）寻址功能。

为满足FFT、卷积等数字信号处理的特殊要求，当前的DSP大多在指令系统中设置了循环寻址及位倒序指令及其它特殊指令、体现在作这些运算时寻址、排序及计算机速度大大提高。

另外，DSP系统设计和软件开发，往往需要相当规模的仿真调试系统，为方便用户的设计与调试，许多DSP在片上设置了仿真模块或仿真调试接口、如Motorola在DSP片是设置了一个QnCE (On-Chip Emulation)功能块、通过行脚、使用户可以检查片内存储器、寄存器及外设，用单步运行，设置断点、跟踪等方式控制与调试程序。Ti则在DSP片上设置了JTAG（Joint Test Action Group）标准测试接口及相应的控制器。在PC机上插入一块调试插板，接通JTAG接口，就可以在PC机上运行一个软件去控制它。

DSP有自己的汇编语言指令系统，为适用于高级语言编程，各公司也陆续推出适用于DSP的高级语言编译器，主要是C语言编译器，也有Ada、Pascal等编译器，程序员可用高级语言编程，通过编译器，将程序汇编、连接成DSP目标代码。

2、DSP应用领域

随着DSP性能的改善和成本的降低，DSP在通用数字信号算法处理的基础上，其应用几乎遍及电子学每个领域。

(1) 通信：网络通信，高速调制解调器，编/解码器，自适应均衡器，移动通讯，语音邮箱，回音消除，噪声对消，会议电视，扩频通信等。

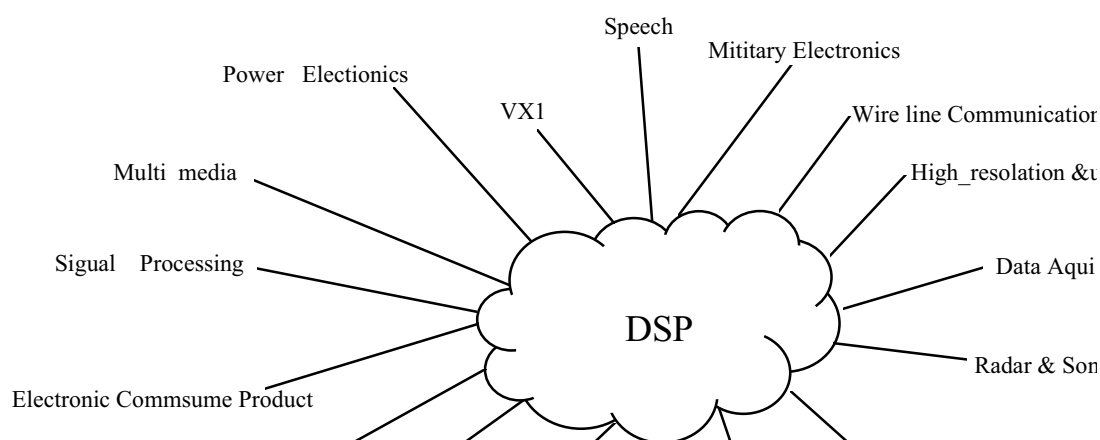
(2) 声音/语音处理、图像/图形处理：语言识别，语言合成、语音编码、图形变换、机器人视觉、模式识别、电子地图等。

(3) 仪器/仪表：谱分析、函数发生器、波形发生器、数据采集、状态分析、石油/地质勘探、VR航空（Visual Reality）试验，医学仪器等。

(4) 计算机：阵列处理器、图形加速器、神经网络、多媒体等。

(5) 消费电子：数字电视、高清晰度电视、VR游戏、汽车工业等。

(6) 军事：雷达与声纳信号处理、导航、制导、全球定位、搜索跟踪、VR式器试验等。如下图所示：



(下面以TMS320C54X系列为例介绍DSP的硬件结构及软件编程方法)

§ 3.10.3 TMS320 C54X DSP硬件结构

1、TMS 320C54X内部组成单元与总线结构

TMS320C54XDSP的结构简图如图1所示：

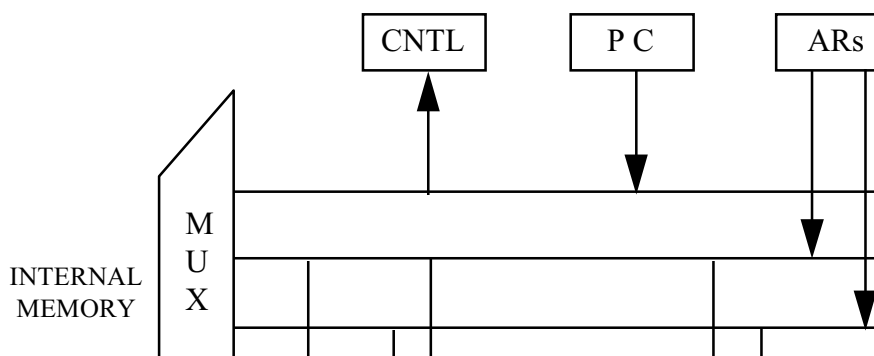


图1 TMS320C54X内部组件与总线连接

可见DSP内部的程序总线P、数据总线D、控制总线C和数据写存储器总线E都是独立分开的，以便于组件单元的并行处理和数据传送。

TMS320C54X C54X的结构部件其实可看作是围绕挂换在8束16 bit的总线上的。（见图2 TMS320 C54X内部硬件结构框图），其中：

（1）程序总线PB传送指令码和程序存储器的直接操作码。

（2）三束数据总线（CB、DB和EB）相互连接起各个单元，如CPU、数据地址产生逻辑（DAGEN）、程序地址产生逻辑（PAGEN）及其它芯片组件和数据存储器（其中CB、DB传送从数据存储器读出的数据，EB传送给存储器写的的数据）。

（3）四束地址总线（PAB、CAB、DAB和EAB）传送指令执行期间所需的地址数据。

2、内部存储器结构

54X的存储器由程序空间、数据空间和输入/输出空间三个分开的空间组成，包括随机存取存储器RAM和只读存储器ROM。RAM可当作DARAM（dual-access RAM）和SARAM（Single-access RAM）54X系列中各种芯片的ROM、DARAM和SARAM空间大小有所不同。

（1）片上ROM（On-Chip ROM）它是程序存储器的一部分，常用以接受boot Loading。

（2）DARAM（On-Chip Dual-Access RAM）。它由若干块组成，对DARAM中的数据，CPU在一个机器周期内可对它进行两次操作（读或写），DARAM总是映射在数据空间，其主要目的是存储数值，它也可映射于程序存储空间用以存储程序代码。

（3）SARAM（On-Chip Single-Access RAM）。它由若干块组成，每块在一个机器周期内，可被读或写一次，它一般映射在数据空间，主要是存储数值，它也可映射于程序空间存储程序代码。

（4）片上RAM保护。当设计片上RAM保护时，外部指令就不能对其进行读写。

3、CPU（Central Processing Unit）

TMS 320 C54X CPU包括：

a) 40-bit的算术逻辑单元 (ALU)。可实现2个40bit操作数的算术逻辑运算, 也可进行布尔运算。

b) 2个40-bit的累加器 (A、B)。可存储ALU或乘/加 (Multiplier/adder) 运算的输出, 也可提供ALU的二次输入。累加器A还可作乘/加运算的输入, 每个累加器都可划分成以下三部分:

i) 保留位 (bit 39-32);

ii) 高字节 (bit31-16);

iii) 低字节 (bit15-0)。

c) 桶式移位器 (Barrel shifter)。桶式移位器有一个来自40bit的累加器 (A或B) 或数据存储器 (通过CB或DB) 的输出, 其40bit的输入, 送到ALU或到数据存储器 (通过EB) 中。它可对输入产生0—31bit的左移和0~16bit的右移。并由指令进行控制。

d) 17×17-bit的乘法器。它实现两个17bit操作数的乘运算, 并与40bit的加法器等共同实现在一个指令周期内完成乘/加运算; 和AKU共同在一个指令周期内完成并行的乘/累加 (MAC) 运算。

e) 40-bit加法器 (Adder(40)), 配合Multiplier执行乘/加运算。

f) 比较、选择和存储单元 (CSSU), 实现累加器高、低字节中数据的大小比较测试或控制标志TC、状态寄存器ST0和发送寄存器 (TRN) 以保持发送状态等。

g) 数据地址产生单元 (DAGEN)

h) 程序地址产生单元 (PAGEN)

4、DSP外部信号引脚

DSP处理信号引脚包括: (C548)

(1) 外部总线接口包括: (External Bus Interface)

并行地址总线口 (23-bit)

并行数据总线口 (16-bit)

存储器选择MSTRB_、程序空间选择PS_、数据空间选择DS、I/O选择IOSTRB_、IS_读/写信号R/W、READY、信号保持HOLD_、指令获得IAQ_等。

(2) 控制信号 (Control Signals)

复位信号RS;

外部中断输入INTX_及中断响应IACK_等;

分支控制BIO-;

外部标志输出XF;

工作方式选择MP/MC等。

(3) 晶振/时钟信号 (Oscillator/Timer Signals)

晶振输出插入 X1、X2/CLKIN

时钟信号：包括CLKOUT、CKKMDX、TOUT等。

(4) 缓冲串行口信号 (Buffered Serial Ports Signals)

接收时钟：BCLKRO、BCLKR1；发送时钟：BCLKXO、BCLKX1；

串行数据接收：BDR0、BDR1；串行数据发送BDX0、DBX1；

帧同步接收：BFSR0、BFSR1；帧同步发送BFSX0、BFSX1。

(5) TDM串口信号

TDM 接收时钟TCLKR；TDM发送时钟TCLKX

串行数据接收TDR；串行数据发送TDX

接收帧同步/地址 TFSR/TADD；帧同步发送TFSX/TFRM

(6) 仿真信号 (Emulation)

JTAG测试时钟TCK

JTAG数据输入TDI

JTAG口模式选择TMS

JTAG数据输出TDO

JTAG口复位TRST_

仿真引脚 EMU0、EMU1/OFF

(7) Host接口Host Port Interface

并行数据口 HD0~7

控制输入 HCNTL0、HCNTL1

字节识别输入 H+BIL

片选 1+CS

数据输入选择 HDS1_HDS1；

地址选通：HAS；

读/写信号：HRW、HRDY

串断输出：HINT；

HPI模式选择 HPIENA/VDD

如图3 “C54X信号引脚”和”C548信号引脚”。

5、DSP硬件设计的几个主要问题：

- (1) 总线控制方案
- (2) boot loading
- (3) Hardware wait_states
- (4) I/O与中断设计
- (5) Memory Map

(6) 串口工作方式与时序设计

(7) TAP接口

§ 3.10.4 TMS320C54X的软件编程

1、DSP编程工具与流程

DSP的设计目标是进行数字信号处理，在硬件设计的基础上选择好一定的优化算法并通过编程在DSP芯片上实现是DSP技术的核心内容。对DSP进行编程，目前最有效的语言工具仍是DSP汇编语言，同时为方便用户用高级语言进行编程开发，也相继推出了C语言编译器，ADa、Pascal等编译器。

图4是TMS C320C54X软件开发流程，图中有阴影的部分是通常开发的必需步骤，其它可根据需要选作，最后产生COFF（Common Object File Format）文件。

2、DSP汇编语言编程

(1) DSP汇编过程有以下几个步骤

a) 源程序的编写。将算法的详细实现过程用DSP指令系统的语句描述出来，其中源程序一般包括：data、bss和text三个基本组成内容。下面是一段程序例子，图5是COFF文件格式和object file在存储器中的分配情况。

```
2          *****
3          ** assemble an initialized table into data.**
4          *****
5 0000          data
6 0000 0011    coeff word 011h,022h,033h
7             0001 0022
8             0002 0033
9             *****
10          ** Reserve space in bss for a variable **
11          *****
12          bss buffer,10
13          *****
14          ** Still in .data.
15          *****
16          ptr word 0123h
17          *****
18          ** Assemble code into the text section ****
19          *****
20          text
21          add: LD 0Fh,A
22          aloop: SUB #1,A
23          BC aloop,AGEQ
24          *****
25          ** Another initialized table into data *****
26          *****
27          data
28          ival .word 0AAh,0BBh,0CCh
29          *****
30          ** Define another section for more variables**
31          *****
32          var2 .userct "newvars",1
33          inbuf .userct "newvars",7
34          *****
35          ** Assemble more code into text *****
36          *****
37          text
```

```

36 0005 110a      may: LD  0Ah,B
37 0006 f166      mloop: MPY #0Ah,B
      0007 000a
38 0008 f868      BC   mloop,BNOV
      0009 0006'
39
40                *****
41                *** Define a named section for int vectors *****
42                *****
42 0000                .sect  "vectors"
43 0000 0011          .word  011h,033h
44 0001 0033

```

b) cmd文件的编写。在使用不同的DSP芯片时，我们还要根据不同芯片的内部存储器情况，修改cmd文件，com文件中的内容必须按照系统的存储器地址分配确定。C54X系统与CMD文件配置情况见图6。

c) 编译与连接。通过对源程序 (*.asm) 的编译产生COFF目标文件 (*.obj)，再通过连接产生可执行的COFF文件 (*.out)，就可以进行仿真和加载。编译与连接格式如下：

编译: asm 500[input file [object file[list file]]][option]

连接: Link 500[-option]filename,···filenamen

3、DSP开发系统仿真

Ti提供了TMS320C54X与DSP开发仿真环境，包括Simulator和Emulator

Simulator是软件仿真器，它可仿真54X的全部指令、I/O和主要外设功能，装入由汇编器/连接器产生的目标代码后，可连接或单步运行，仿真观察各存储器、寄存器的内容，以作汇编、编辑与修改等。

Emulator是PC机插件形式的硬件仿真器及开发系统；它具有良好的用户界面，可作全速的在线仿真。

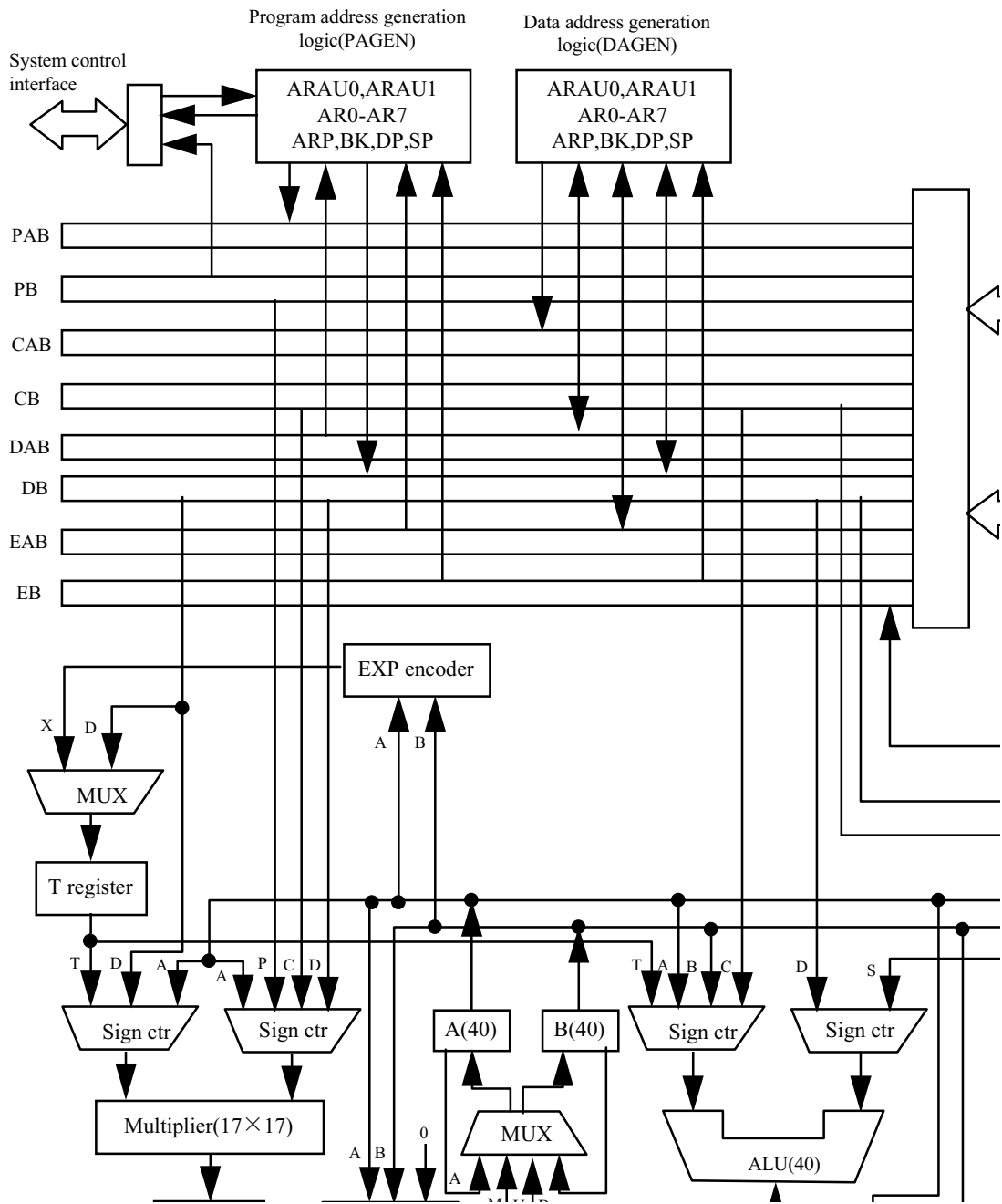


图2 TMS320 C54X硬件结构

图3 C542和C548的信号引脚

TMS320C54x Software Development Flow

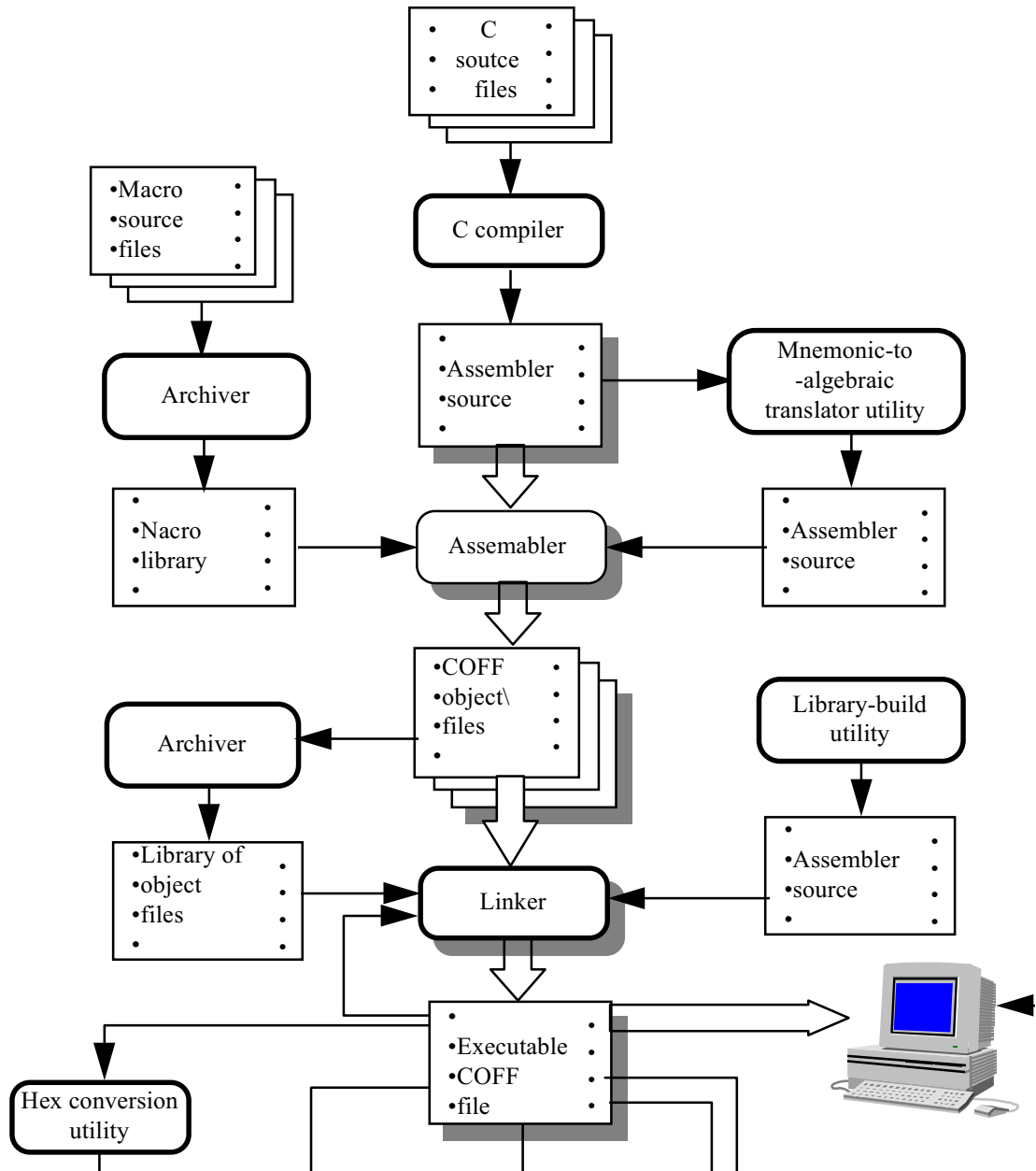


图4 TMS320 C54X软件开发流程

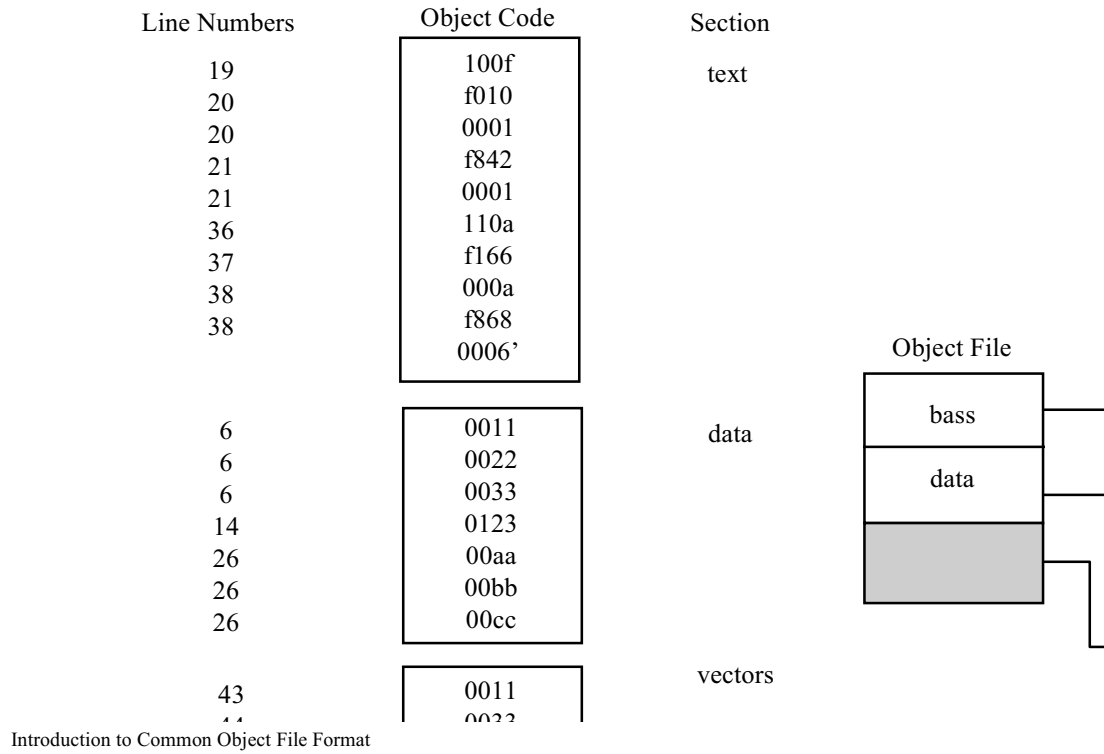
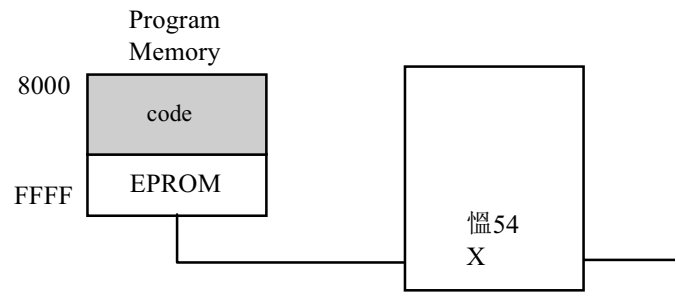


图5 COFF文件及Object File 在存储器中的分配

Example System



```

example1.obj
-o example1.out
-m example1.map
MEMORY
{
  Page 0: /* Program */
    EPROM: org=8000h,len=8000h
  page 1:/* Data */
    SRAM: org=4000h,len=2000h
    DEPRM: org=8000h,len=2000h
}

```

图6 C54X系统与CMD文件配置

硬件EMC设计规范

引言：

本规范只简述EMC的主要原则与结论，为硬件工程师们在开发设计中抛砖引玉。

电磁干扰的三要素是干扰源、干扰传输途径、干扰接收器。EMC就围绕这些问题进行研究。最基本的干扰抑制技术是屏蔽、滤波、接地。它们主要用来切断干扰的传输途径。广义的电磁兼容控制技术包括抑制干扰源的发射和提高干扰接收器的敏感度，但已延伸到其他学科领域。

本规范重点在单板的EMC设计上，附带一些必须的EMC知识及法则。在印制电路板设计阶段对电磁兼容考虑将减少电路在样机中发生电磁干扰。问题的种类包括公共阻抗耦合、串扰、高频载流导线产生的辐射和通过由互连布线和印制线形成的回路拾取噪声等。

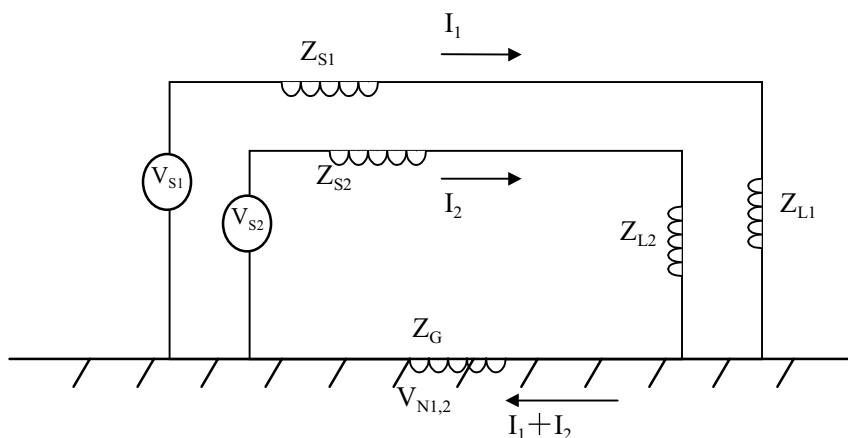
在高速逻辑电路里，这类问题特别脆弱，原因很多：

- 1、电源与地线的阻抗随频率增加而增加，公共阻抗耦合的发生比较频繁；
- 2、信号频率较高，通过寄生电容耦合到步线较有效，串扰发生更容易；
- 3、信号回路尺寸与时钟频率及其谐波的波长相比拟，辐射更加显著。
- 4、引起信号线路反射的阻抗不匹配问题。

一、总体概念及考虑

- 1、五一五规则，即时钟频率到5MHz或脉冲上升时间小于5ns，则PCB板须采用多层板。
- 2、不同电源平面不能重叠。
- 3、公共阻抗耦合问题。

模型：



$V_{N1} = I_2 Z_G$ 为电源 I_2 流经地平面阻抗 Z_G 而在 1 号电路感应的噪声电压。

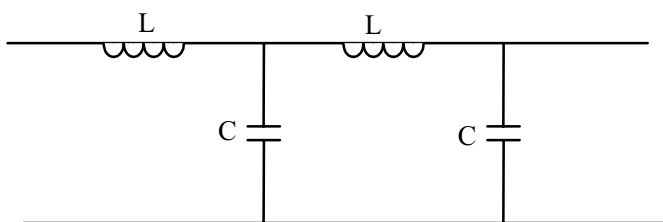
由于地平面电流可能由多个源产生，感应噪声可能高过模电的灵敏度或数电的抗扰度。

解决办法：

- ① 模拟与数字电路应有各自的回路，最后单点接地；
- ② 电源线与回线越宽越好；
- ③ 缩短印制线长度；
- ④ 电源分配系统去耦。

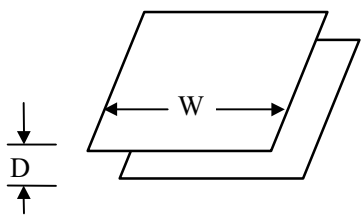
4、减小环路面积及两环路的交链面积。

5、一个重要思想是：PCB 上的 EMC 主要取决于直流电源线的 Z_0



电源线分布电感与电容

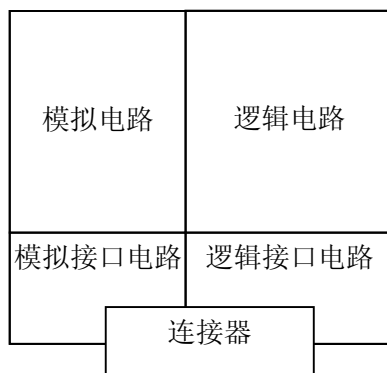
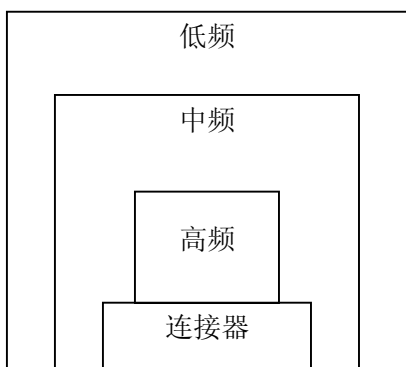
$C \rightarrow \infty$ ，好的滤波， $L \rightarrow 0$ ，减小发射及敏感。

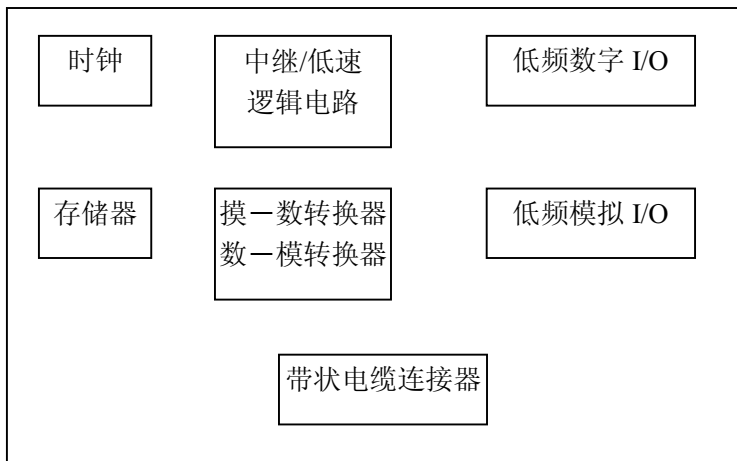


$$Z_0 = \sqrt{L/C} = 377(d/w) \sqrt{\mu_r / \epsilon_r}, \text{ 如果 } < 0.1 \Omega \text{ 极好。}$$

二、布局

下面是电路板布局准则：





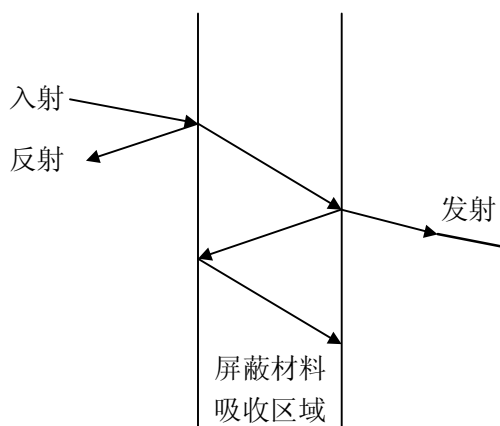
- 1) 晶振尽可能靠近处理器
- 2) 模拟电路与数字电路占不同的区域
- 3) 高频放在PCB板的边缘，并逐层排列用地填充空着的区域

三、布线

- 1、电源线与回线尽可能靠近，最好的方法各走一面。
- 2、为模拟电路提供一条零伏回线，信号线与回程线小与5: 1。
- 3、针对长平行走线的串扰，增加其间距或在走线之间加一根零伏线。
- 4、手工时钟布线，远离I/O电路，可考虑加专用信号回程线。
- 5、关键线路如复位线等接近地回线。
- 6、为使串扰减至最小，采用双面#字型布线。
- 7、高速线避免走直角。
- 8、强弱信号线分开。

四、屏蔽

1屏蔽 > 模型:



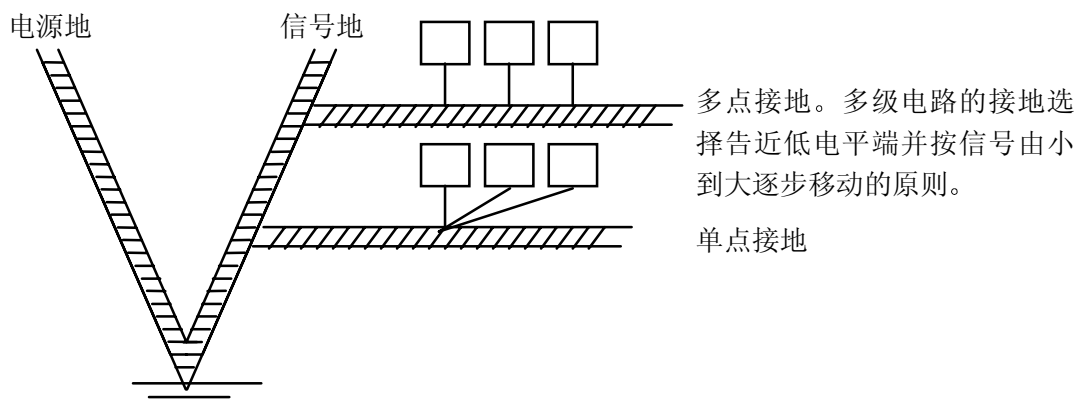
屏蔽效能 $SE(dB) = \text{反射损耗}R(dB) + \text{吸收损耗}A(dB)$

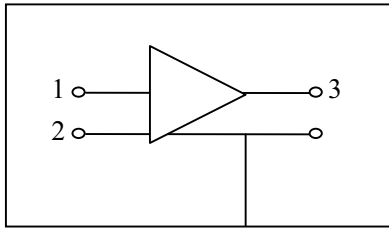
高频射频屏蔽的关键是反射，吸收是低频磁场屏蔽的关键机理。

- 2、工作频率低于1MHz时，噪声一般由电场或磁场引起，(磁场引起时干扰，一般在几百赫兹以内)，1MHz以上，考虑电磁干扰。单板上的屏蔽实体包括变压器、传感器、放大器、DC/DC模块等。更大的涉及单板间、子架、机架的屏蔽。
- 3、静电屏蔽不要求屏蔽体是封闭的，只要求高电导率材料和接地两点。电磁屏蔽不要求接地，但要求感应电流在上有通路，故必须闭合。磁屏蔽要求高磁导率的材料做封闭的屏蔽体，为了让涡流产生的磁通和干扰产生的磁通相消达到吸收的目的，对材料有厚度的要求。高频情况下，三者可以统一，即用高电导率材料(如铜)封闭并接地。
- 4、对低频，高电导率的材料吸收衰减少，对磁场屏蔽效果不好，需采用高磁导率的材料(如镀锌铁)。
- 5、磁场屏蔽还取决于厚度、几何形状、孔洞的最大线性尺寸。
- 6、磁耦合感应的噪声电压 $U_N = j\omega B \cdot A \cdot \cos\theta = j\omega M \cdot I_1$ ，(A为电路2闭合环路时面积；B为磁通密度；M为互感； I_1 为干扰电路的电流。降低噪声电压，有两个途径，对接收电路而言，B、A和 $\cos\theta$ 必须减小；对干扰源而言，M和 I_1 必须减小。双绞线是个很好例子。它大大减小电路的环路面积，并同时在校合的另一根芯线上产生相反的电动势。
- 7、防止电磁泄露的经验公式：缝隙尺寸 $< \lambda \min/20$ 。好的电缆屏蔽层覆视率应为70%以上。

五、接地

- 1、300KHz以下一般单点接地，以上多点接地，混合接地频率范围50KHz~10MHz。另一种分法是： $< 0.05 \lambda$ 单点接地； $> 0.05 \lambda$ 多点接地。
- 2、好的接地方式：树形接地





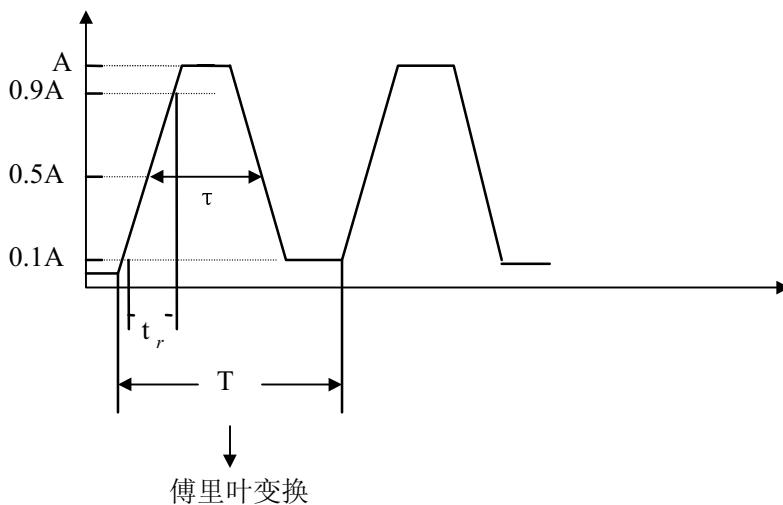
接地点选在放大器等输出端的地线上。

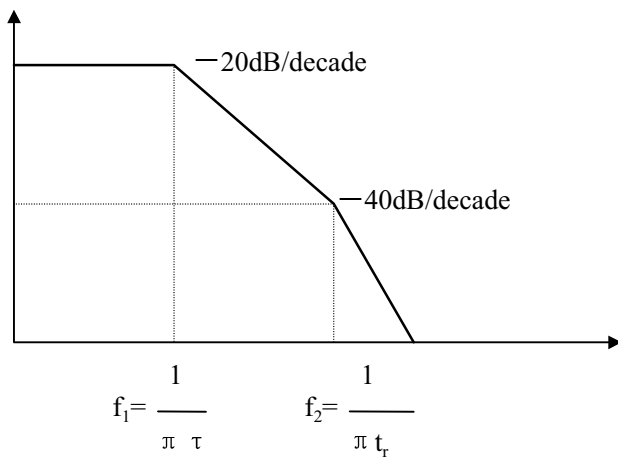
- 4、对电缆屏蔽层， $L < 0.15 \lambda$ 时，一般均在输出端单点接地。 $L < 0.15 \lambda$ 时，则采用多点接地，一般屏蔽层按 0.05λ 或 0.1λ 间隔接地。混合接地时，一端屏蔽层接地，一端通过电容接地。
- 5、对于射频电路接地，要求接地线尽量要短或者根本不用接线而实现接地。最好的接地线是扁平铜编织带。当地线长度是 $\lambda/4$ 波长的奇数倍时，阻抗会很高，同时相当 $\lambda/4$ 天线，向外辐射干扰信号。
- 6、单板内数字地、模拟地有多个，只允许提供一个共地点。
- 7、接地还包括当用导线作电源回线、搭接等内容。

六、滤波

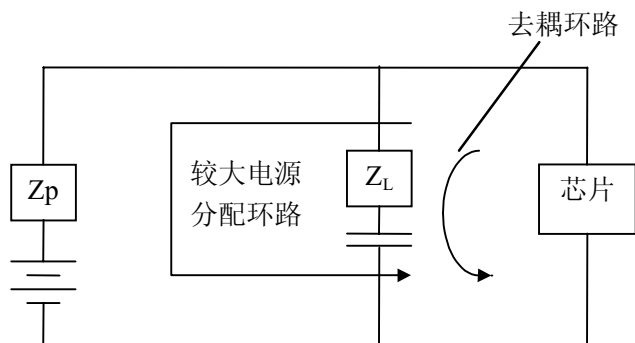
- 1、选择EMI信号滤波器滤除导线上工作不必要的高频干扰成份，解决高频电磁辐射与接收干扰。它要保证良好接地。分线路板安装滤波器、贯通滤波器、连接器滤波器。从电路形式分，有单电容型、单电感型、L型、 π 型。 π 型滤波器通带到阻带的过渡性能最好，最能保证工作信号质量。

一个典型信号的频谱：





- 2、选择交直流电源滤波器抑制内外电源线上的传导和辐射干扰，既防止EMI进入电网，危害其它电路，又保护设备自身。它不衰减工频功率。DM(差模)干扰在频率 $< 1\text{MHz}$ 时占主导地位。CM在 $> 1\text{MHz}$ 时，占主导地位。
- 3、使用铁氧体磁珠安装在元件的引线上，用作高频电路的去耦，滤波以及寄生振荡的抑制。
- 4、尽可能对芯片的电源去耦($1-100\text{nF}$)，对进入板极的直流电源及稳压器和DC/DC转换器的输出进行滤波(μF)。



$$C_{min} \approx \Delta I \Delta t / \Delta V_{max} \quad \Delta V_{max} \text{一般取} 2\% \text{的干扰电平。}$$

注意减小电容引线电感，提高谐振频率，高频应用时甚至可以采取四芯电容。电容的选取是非常讲究的问题，也是单板EMC控制的手段。

七、其它

单板的干扰抑制涉及的面很广，从传输线的阻抗匹配到元器件的EMC控制，从生产工艺到扎线方法，从编码技术到软件抗干扰等。一个机器的孕育及诞生实际上是EMC工程。最主要需要工程师们设计中注入EMC意识。

FPGA归档要求

FPGA(Field Programmable Gate Array)已广泛用于硬件的逻辑设计中，在其开发过程中，出现了多种版本的开发工具，而每一种版本，从原理图输入到最终形成加载文件需要对一些参数进行设定，为了更好地利用已开发的成果，避免在文件的传递过程中出现错误或误解，我们在归档时，除了按要求对硬件作必要说明，还应针对FPGA作专门说明。

（由于我公司用Xilinx公司的FPGA较多，这里以其为例，其它公司FPGA产品可类似）。

1、归档文件种类

归档文件必须有如下文件：

- (1) 原理图（包括所有子层的原理图）或VHDL语言描述的逻辑设计文件；
 - (2) 单层结构的网图文件，即XFF文件；
 - (3) 已布线的LCA文件；
 - (4) 可直接卸载的bit和MCS文件；
 - (5) 除标准库外，定义的所有库文件；
 - (6) Viewdraw.INI文件；
 - (7) 仿真文件；（当需要对某些重要波形进行控制时，提供）
- 28? 生成的布线指导文件Guide文件。（对布线要求严格，编译选项有Guide时, 提供）

2、加载文件的编程说明

- (1) FPGA的工作模式及加载方式；
- (2) 加载文件的编程工具；
- (3) 对MCS文件的编程过程；
- (4) 复位位说明。

3、使用开发工具，及参数选择说明

- (1) 开发工具的类型及版本：

如Viewlogic DOS 版FPGA开发工具,版本号4.00版

Foundation windows版FPGA开发工具,版本号2.00版

- (2) 采用器件系列库：

如：XC3000…… XC5000等

- (3) 开发使用的计算机环境（硬件、软件环境）

- (4) 使用FPGA具体器件的详细型号

如：XC4003EPLCC84—5

注：对有重点要求的参数如速率、温度等，需在此作一说明

(5) 原理图中定义的关键网络有特殊要求电路的说明

如：时延要求、长线驱动、快速I/O等

(6) 编译后，FPGA引脚定义

(7) 自动布线的参数说明

每一种FPGA开发工具都提供自动布线功能，当布线的参数选择不同时，编辑后的结果可能差别很大，必须对要求的参数选择进行说明，下面列出部分常用参数选项供参考。（说明：实际中，只需把要求的选项列出）。

附:自动布线的参数设置

(A) Foundation版本或Viewlogic Windows版本

Xilinx FPGA XC4000 implemenlation

1.Implementation

A.Placement Effort()1--4

Routing Effort ()1--4

B.Use Xact-Performance()

2.Optimization

()Trim Unconnected Signats

()Use Global Resources For High fan-out Signals

()Create RPMs for Register-based X-Blox Modules

()Merge Flip-Flop into I/Os

3.Guide/Resource

A.Resoutces Available fot Fouting

CLBs:()Partially Used ()Any

()Unused Gloabal buffers

B.Guide Placement

()AllBlocks ()Only block that Have Routed Signals

C. Guide Routing

locking Routing ()None ()Whole Signails ()All

Xilinx FPGA XC4000 Configuration

1.Configuration

Configuration Rate:()Fast (Slow)

TDO:()Float ()Pull-up ()Pull-down

M1:()flat ()Pull-up ()Pull-down

DONE: ()Pull-up ()Pull-down

()performance CRC Durng Configuration

Produce ASCII Configuration File

// XC4000E:Inputs threshold:()TTL ()CMOS

Output threshold:()TTL ()CMOS

Configuration Pins:

M0:()Float ()Pull-up ()Pull-down

M2:()Float ()Pull-up ()Pull-down

2.Startup

Startup clock:()clock ()user clock

Synchronize Startup to Done IN Pin

Output Events

Done: ()C1 ()2 ()C3 ()C4

Enable Outputs:()C2 ()C3 ()C4

Release Set/Reset: ()C2 ()C3 ()C4

3.Readback

Clock:()clock ()user clock

Capture CLBIOB Outputs when TRIG Goes Active

Abort Readback when TRIG Goes Inactive

XC4000 Optional Targets

Produce Timing Simulation Data

Produce Timing Report

Produce Configuration Data

Xilinx FPGA XC3000 Implementation

1.Implementation

A.Placement Effort ()1---4

Routing Effort ()1---4

B.Use Xact-Performance ()

2.Optimization

Trim Unconnected Signals

Pack Design

Use Global Resource For High Fan-out Signals

Create RPMs for Register-based X-Blox Modules

Merge Flip-Flop into I/Os

3.Guide/Resource

A.Resources Available fot Fouting

CLBs:()Partially Used ()Any ()None
Unused Global Buffers

B.Guide Placement

()All Blocks ()Only Block that Have Routed Signats

C. Guide Routing

Locking Routing ()None ()Whole Signals ()All

Silinx FPGA XC3000 Configuration

1.Configuration:

Inputs threashole :()TTL ()CMOS

Configuration Pins:

Done/Program:()Float ()Pull-up

Crystal Oscillator:()Disable ()Enable ()Enable (Divide by 2)

2.Startup

Statup clock:()clock ()user clock

()Synchornize Startup to Done INPin

Output Events:

Done:()C1 ()C2 ()C3 ()C4

Enable Outputs ()C2 ()C3 ()C4

Release Set/Reset: ()C2 ()C3 ()C4

3.Readback

Mode:()Never ()Once ()On Command

Output Events:

Done:()Before I/O Active ()After I/O Active

Reset:()Before I/O Active ()After I/O Active

SC3000 Optional Targets

()Produce Timing simulation Data

()Produce timing Report

() Produce Configuration Data

Foundation

Xilinx FPGA XC5200 Implemcntation

1.Implememtation

A.Placement Effort ()1--4

Routing Effort ()1--4

B.Use Xact-Performance ()

2.Optimization

()Trim Unconnected Signais

()use Global Resources For High Fan-out Signals

()Create RPMs for Register-based X-Blox Modules

3.Guide/Resource

A.Resources Available fot Fouting

CLBs:()partially used ()Any

Unused global Buffers

B.Guide Placement

()All Blocks ()Only block that Have Routed Signats

C. Guide Routing

Locking Routing ()None ()Whole Signals ()All

Xilinx FPGA XC5200 Configuration

1.Configuration

Inputs threshold:()TTL ()CMOS

configuration Rate:()Fast ()Slow ()Wed

configuration Pins:

Program:()flcat ()Pull-up

Done: ()Float ()Pull-up

Perform CRC During Configuration ()

Produce ASCII configuration File()

2.Startup

Startup clock :()clock ()user clock

()Synchronize Statuup to Done IN Pin

Output Events:

Done:()C1 ()C2 ()C3 ()C4

enable Outputs :()C2 ()C3 ()C4

Release Set/Reset: ()C2 ()C3 ()C4

3.Readback

Clock :()clock ()user clock

()Capture CLB/IOB Outputs when TRG Goes Active

()Abort Readback when TRIG Genes Inactive

Optional Targets

Produce Timing Simulation Data

Produce Timing Report

Produce Configuration Data

2.Startup

Startup clock : clock user clock

Synchronize Statuip to Done IN Pin

Output Events

Done: C1 C2 C3 C4

Enable Optputs: C2 C3 C4

Release Set/Reset : C2 C3 C4

3.Readback

Clock clock user clock

Capture CLB/IOB Outputs when TRG Goes Active

Abort Readback when TRIG Geses Inactive

Optional Targets

Produce Timing Simulaton Data

Produce Timing Report

Produce configuration Data

2.Startup

Startup clock : clock user clock

Synchronize Startup to Done IN Pin

Makebit

—I (Ignore Critical Net Flags on Nets)

—U (Enable Mackebits to Tle using critical nets)

—B (Create rawbits file in RBT)

—C (Cmos inputs)

—D (Create DRC report in DRC)

—J (Suppress making bitstream)

—L (Create Readback into file in .LL)

—M (Creare mack file in .MSK)

—MBO=(File) (Read Makebits options from FILE)

—N (Save tied desigt)

—O (Output filemame)

—P (No pullup on LCA Done pad)

- R0 (Disable readblack)
- R1 (Enable readback once)
- R2 (Enable readback many)
- S0 (Disable Oscillator)
- S1 (Enable Oscillator)
- S2 (Enable Oscillator Div 2)
- T (Tiedown Unused Interconnect)
- V (verbose Mode)
- XA (Done after IOBs active)
- XB (Done before IOBs active)
- YA (Reset after IOBs active)
- YB (Reset before IOBs active)

(B) ViewLog DOS版本

Xmake的部分选项

- X Use XNF files only
- L Use old library only
- B perform XBLOX optimization
- I <file> Use PPR Guide file
- G Generate MAK file only
- O Don't redirect output
- R Force re-execution of all programs
- V Verbose message made

Makebit

- Input TTL () COMS ()
- Donepad Pullup () NoPullup ()
- Read 0 () 1 () Cmd ()
- XtaIOsc Disable () Enable () DIV2 ()
- DoneTime Before () After ()
- RestTime Before () After ()