

Y 936197

分类号: _____

密级: _____

U D C : _____

编号: _____

工学硕士学位论文

96 系列单片机仿真器研究与设计

硕士 研究生 : 张子红

指 导 教 师 : 金鸿章 教授

学 位 级 别 : 工学硕士

学 科、专 业: 控制理论与控制工程

所 在 单 位 : 自动化学院

论文提交日期: 2006 年 2 月

论文答辩日期: 2006 年 3 月

学位授予单位: 哈尔滨工程大学

摘 要

在线仿真器是单片机开发系统的重要组成部分之一，是开发单片机的强大工具。本文所设计的 96 系列单片机仿真器是在通用单片机仿真器主模块板的仿真接口上外接一个 96/98 BH 仿真扩展板构成的，将其与主机相连就可组成一个通用的 MCS-96 系列单片机开发系统。

本文首先对单片机开发系统和在线仿真器进行了全面的介绍。然后给出了通用单片机仿真器的结构和功能以及它的系统连接和使用方法，并给出了它所具有的软件功能。接着，介绍了九六系列单片机的各种型号产品并根据九六系列单片机所具有的内部结构和性能特点提出了 96 系列单片机在线仿真器的整体设计方案，并完成了仿真扩展板的硬件电路设计。同时对仿真扩展板中各主要元器件的选择及其在电路中所实现的功能进行了详细的论述。最后，给出了单片机仿真器的仿真软件设计，以流程图的方式对其进行了说明。通过试验可知，所设计的 96 系列单片机仿真器能够实现仿真时所需要的各种功能。

关键词：开发系统；单片机；在线仿真器；仿真扩展板

Abstract

In-circuit Emulator is one of the most important component parts of Microcomputer Development Systems. It is a powerful tool to develop microcontroller. The 96 family single chip microcomputer In-Circuit Emulator(ICE) designed in this paper is consist of a single chip Microcomputer In-Circuit Emulator and a 96/98 BH expandable board, it will be a general MCS-96 single chip development if we connect it to the host.

The Microcomputer Development System and ICE are introduced comprehensively at the beginning of this paper. After that, we give a description of architecture and function of single chip microcomputer In-Circuit Emulator .we especially explained how to use it and realize communication with the host system. In addition, it has rich software. And then, we introduced all kinds of products of 96 family single chip microcomputer and we build a design scheme on the base of internal architecture as well as its performance of MCS-96 and finished the hardware design. At the same time , we discussed the selection of major component in the expandable board and the function of component in the circuit in detail. Finally, we completed a software design about a single chip microcomputer In-Circuit Emulator and gave a particular description by flow chart. We concluded by experiment that 96 family single chip microcomputer In-Circuit Emulator can realize the function which the user need.

Keywords: development system; single chip; in-circuit emulator; expandable board

哈尔滨工程大学

学位论文原创性声明

本人郑重声明：本论文的所有工作，是在导师的指导下，由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出，并与参考文献相对应。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者（签字）： 张子江

日 期： 2006年 2月 27日

第 1 章 绪论

1.1 课题的来源及意义

随着电子技术的迅速发展，特别是随着大规模集成电路的产生而出现的微型计算机，给人类生活带来了根本性的改变。如果说微型计算机的出现使现代科学研究得到了质的飞跃，那么单片机技术的出现则给工业测控领域带来了一次新的技术革命。单片机又称为微控制器(Microcontroller，简称 MCU)，一般以某种微处理器内核为核心，芯片内部集成了 ROM, EPROM, E²PROM, Flash, RAM, A/D, D/A、总线逻辑、定时器/计数器、看门狗、I/O、串行接口、脉宽调制器等。

自 1976 年 9 月 INTEL 公司率先推出 MCS-48 系列单片机以来，伴随着半导体集成电路制造工艺的不断发展和电子技术应用领域的不断扩展，使得单片机不断更新换代，品种数量日益繁多，功能日益强大，价格不断下降，速度越来越快，根据最新的研究资料表明，利用最新的 SOI^[1] (Silicon-On-Insulator) 技术，单片机的频率可以达到 400MHZ。单片机以其高可靠性、高性能价格比向国民经济的各个领域渗透，在工业控制系统、数据采集系统、智能化仪器仪表、办公自动化等诸多领域得到极为广泛的应用，并已走入家庭，从洗衣机、微波炉到音响、汽车，到处都可见到单片机的踪影。

单片机本身只是一个电子元件，只有当它和其它的器件、设备有机的组合在一起，并配置适当的工作程序（软件）后，才能构成一个单片机的应用系统，完成规定的操作，具有特定的功能。我们把“一个单片机应用系统（或称目标系统）从提出任务到正式投入运行（或批量生产）的过程”称之为对单片机的开发^[2]。因此单片机的开发包括硬件和软件两个部分。

单片机本身没有自开发功能（通用计算机系统具有这种功能，用户可以在上面研制应用软件或对系统进行扩展），必须借助于开发工具来排除目标系统样机中硬件故障，生成目标程序，并排除程序错误。当目标系统调试成功

以后,还需要用开发工具把目标程序固化到单片机内部或外部 EPROM 芯片中。

单片机的开发工具通常是一个特殊的计算机系统-开发系统。开发系统是我们研制微机化产品或微机应用系统的过程中,必不可少的工具,特别是实时在线仿真的开发系统。用户可以通过开发系统的键盘下达各种控制命令,并通过显示器来了解程序运行中指令执行后的效果,进行调试、修改和联机运行实际系统。

目前,市场上各种各样的单片机开发工具层出不穷,各个生产厂家竞相设计、改进,并提出更完善的各种单片机开发系统,但总体设计思想基本没变,即在保证实现开发系统某些传统功能(运行、停止、单步、跟踪、查看寄存器、读写目标存储器)的基础上,增加了一些附加功能;或是通过使用高性能的器件来实现仿真时所需要的各种信号,使用户使用起来更加方便灵活,价格也越来越低廉。

在线仿真器是单片机开发系统的最重要的硬件组成部分,是开发单片机的强大工具之一,是开发软件和硬件最有效的手段之一。本课题是基于我校开发与设计减摇鳍工作的需要,设计了一套可开发96系列单片机的单片机在线仿真器,它与任何具有RS232串口的计算机相连再加上一些外部设备可组成一个完整的单片机开发系统。通过此次的研究与设计,对单片机的内部结构有了深入了解,对熟练应用单片机有很大的帮助,对今后从事单片机的开发工作也奠定了基础。

1.2 单片机开发系统的功能

用户的单片机应用系统的研制周期和他所选用的单片机开发系统性能优劣密切相关。一个单片机开发系统功能强弱可以从以下几个方面来分析^[2]:

- 在线仿真功能
- 排错功能
- 辅助设计功能

1.2.1 在线仿真功能

仿真就是通过仿真器的硬件和软件真实的模拟被开发应用系统的运行，还应使用户能最大限度的利用单片机的资源。开发系统中的在线仿真器应能仿真目标系统中单片机并能模拟目标系统中的 ROM、RAM 和 I/O 口，也要仿真单片机中断系统的运行，使在线仿真时目标系统的运行环境和脱机运行的环境完全“逼真”，以实现目标系统的完全的一致性仿真。仿真功能具体的体现在以下几个方面：

1. 单片机仿真功能

在线仿真时，开发系统应能将在线仿真器中的单片机完整地出借给目标系统，不占用目标系统单片机的任何资源，使目标系统在联机仿真和脱机运行时的环境（工作程序、使用的资源和地址空间）完全一致，实现完全的一次性的仿真。

单片机的资源包括：片上的 CPU、RAM、SFR、定时器、中断源、I/O 以及外部可扩充的程序存储器和数据存储器地址空间。这些资源应允许目标系统充分地自由地使用，不应受到任何限制，使目标系统能根据单片机固有的资源特性进行硬件和软件的设计。

2. 模拟功能

在开发目标系统的过程中，单片机的开发系统允许用户使用它内部的 RAM 存储器和输入输出替代目标系统中的 ROM 程序存储器、RAM 数据存储器以及 I/O，使用户在目标系统样机还未完全配置好以前，便可以借用开发系统提供的资源进行软件的开发。

最重要的是目标机的程序存储器模拟功能。因为在目标系统开始的初始阶段，目标程序还未生成，更谈不上在目标系统中固化程序。因此用户的目标程序必须存放在开发系统 RAM 存储器内，以便于调试过程中对程序的修改。开发系统所能出借的作为目标系统程序存储器的 RAM 我们常称之为仿真 RAM。开发系统中仿真 RAM 容量和地址映射应和目标机系统完全一致。

1.2.2 排错功能

开发系统必须具有排错功能。因为用户所编制的程序不会一次就完全编好，而且经常需要排错和修改。开发系统对目标系统软硬件的排错功能（也

称为调试功能)强弱,将直接关系到开发的效率。性能优良的单片机开发系统应具有下列排错功能:

(1) 运行控制功能

开发系统应能使用户有效的控制目标程序的运行,以便检查程序运行结果,对存在的硬件故障和软件错误进行定位。

a. 单步运行能力

使 CPU 从任意的目标程序地址开始执行一条指令后停止运行;

b. 断点运行

允许用户任意设置条件断点,启动 CPU 从规定地址开始运行后,当碰到断点条件(程序地址和设定断点地址符合或者 CPU 访问到指定的数据存储器单元)等条件符合以后停止运行;

c. 连续运行

能使 CPU 从指定地址开始连续地全速运行目标程序;

d. 启停控制

在各种运行方式中,允许用户根据调试的需要,来启动或者停止 CPU 执行目标程序。

(2) 对目标系统状态的读出/修改功能

当 CPU 停止执行目标系统的程序后,允许用户方便的读出/修改目标系统所有资源的状态,以便检查程序运行的结果、设置断点条件以及设置程序的初始参数。可供用户读出/修改的目标系统资源包括:

a. 程序存储器(开发系统中的仿真 RAM 存储器或目标机中的 ROM 存储器)。

b. 单片机片上资源:

- . 工作寄存器;
- . 特殊功能寄存器;
- . I/O 口;
- . RAM 数据存储器;
- . 位单元。

c. 系统中扩展的数据存储器和 I/O 口。

1.2.3 跟踪功能及其它辅助设计功能

1. 跟踪功能

高性能的单片机开发系统具有逻辑分析仪的功能，在目标程序运行过程中，能监督和存储目标系统总线上的地址、数据和控制信号的状态变化，跟踪存储器能同步地记录总线上的信息，用户可以根据需要显示跟踪存储器搜集到的信息，也可以显示某一位总线的状态变化的波形。使用户掌握总线上状态变化的过程，这对各种故障的定位特别有用，从而大大地提高工作效率。

2. 辅助设计功能

软件的辅助设计功能的强弱也是衡量单片机开发系统性能高低的重要标志。单片机应用软件的开发的效率在很大程度上取决于开发系统的辅助设计功能。这些功能包括：

(1) 程序设计语言

单片机实际上就是一台没有进行编程的微型计算机，开发系统首先要解决的就是如何编程问题。单片机的程序设计语言有机器语言、汇编语言和高级语言。

机器语言只在简单的开发装置中才使用，程序的输入、修改和调试都很麻烦。只能用来开发非常简单的单片机应用系统。

汇编语言具有使用灵活、程序容易优化的优点，容易设计出高质量的程序。这是单片机中最常用的程序设计语言。但是用汇编语言编写程序还是比较复杂，只有对单片机的指令系统非常熟悉，并具有一定的程序设计经验，才能研制出功能复杂的应用软件。

高级语言具有通用性好，功能强，程序设计人员只要掌握该种语言的特点和使用方法，不需要完全掌握具体的单片机指令系统，就可以编写程序。对于习惯高级语言研制应用软件的用户，乐于使用单片机的高级语言来进行程序设计。对于 MCS-51 系列单片机的语言有：PL/M51\C-51\MBASIC-51 等。解释型高级语言有 BASIC-52、TINYBASIC 等。解释型高级语言必须在解释程序支持下才能运行。编译型高级语言可生成机器码。

应指出的是，在程序设计时可以交叉使用汇编语言和高级语言。

(2) 程序编辑

程序输入的方法有机器码输入和利用行编辑或屏幕编辑对目标系统的源

程序进行编辑。

机器码的输入方式只在简单的开发装置中使用。几乎所有的单片机开发系统允许用户用汇编语言或高级语言编写程序，提供功能很强的编辑程序，例如在 PC 机上的 EDLIN 行编辑 PE、WS 等屏幕编辑程序，使用户方便地将源程序输入到开发系统中，生成汇编语言或高级语言的源文件。然后利用开发系统提供的汇编或编译系统，将源程序编译成可在目标机上直接运行的目标程序，并生成程序清单文件供打印。

(3) 其他软件功能

一些单片机的开发系统还提供反汇编功能，并提供供用户宏调用的子程序库、以减少用户软件研制的工作量。

单片机开发系统其他的功能指标和一般的计算机系统相类似：例如系统的可靠性、可维护性以及 I/O 的种类和存储器的容量等。

3. 程序固化功能

它能完成把调试好的程序固化到用户的程序存储器中去。

4. 人机信息交互功能

它包括运行结果显示、对运行的干预、人机对话和反汇编。

不同的开发系统都必须具备上述基本功能，但对于一个较完善的开发系统还应具备：不占用用户单片机的任何资源，包括内部 RAM、特殊功能寄存器，I/O 端口、串行端口、中断源等；提供给用户足够的仿真 RAM 空间作为用户的程序存储器，并提供足够的仿真 RAM 空间作为用户的数据存储器使用；有较齐全的软件开发工具，如配备有交叉汇编软件，将用户用汇编语言编写的应用程序生成可执行的目标文件；具有丰富的子程序库，汇编时连同用户设计的程序一起编译成目标程序，装入仿真 RAM 供调试和固化；具有高级语言编译系统，用户可以用高级语言进行编程；具有反汇编功能，对目标程序反汇编的结果可以打印或存入磁盘等。

1.3 单片机开发系统的研究现状及发展趋势

所谓单片机开发系统是一种用来进行单片机系统开发、调试、维修和分析的专用工具组成的系统。也有人用下面的等式简单地来表示单片机开发系

统：^[5]

MDS=OLE+SLD+Other Tools

其中：MDS指单片机开发系统（Microprocessor Development Systems）；

OLE指在线仿真器（On Line Emulator）；

SLD指源级调试器（Source Level Debugger）；

Other Tools 指开发系统中用到的其它一些工具。对单片机开发系统的基本要求是：首先把编译好的目标代码存入单片机，然后控制并追踪系统的执行，在设置的断点处可以更改一些寄存器中内容，分析一些基本数据。总的来说，对单片机开发系统的一个总的要求是对于系统组合和综合调试具备控制和分析的能力。

1.3.1 单片机开发系统的分类

目前开发系统的种类比较多，但大致可以分为以下五类^[8]：

1. 软件模拟开发系统

这是一种完全依靠软件手段进行开发的系统。开发系统与用户系统在硬件上无任何联系。只需在通用机上配置模拟开发软件即可对微处理器或单片机进行开发。该开发软件可对微处理器的硬件结构、指令执行、运行状态等进行模拟。模拟开发系统的最大缺点是不能进行硬件部分的诊断与实时在线仿真。

2. 专用型与通用型开发系统

专用型开发系统通常只支持开发特定微处理器生产厂家的微处理器应用。通用型开发系统则可以支持多种微处理器或单片机的开发工作。通用型开发系统比专用型开发系统在性能价格比上具有明显的优势，有很强的竞争优势。

3. 局部占用型与全透明型开发系统

局部占用型开发系统是指在仿真时要占用一部分用户系统资源，包括寄存器、存储器、I/O端口、中断源等，因此造成了“开发死区”，使用户对这部分被占用资源的仿真和调试无法进行。

全透明型开发系统则是使用自成体系的专门硬件和软件系统，通过接管

目标CPU去对用户系统进行开发。这种开发系统自备了仿真时所需的全部资源，因此没有‘开发死区’，能对各种复杂的用户系统进行完全的开发。

4.非交互式与全交互式开发系统

非交互式开发系统主要由仿真器构成，适用于软件开发工作，其优点是价格比较便宜，缺点是对硬件和实时系统软、硬件综合调试能力较弱。

全交互式开发系统由仿真器、仿真存储器、逻辑分析仪等多种设备构成，一般具有很强的在线仿真功能和逻辑分析功能。在交互式开发系统中，这些设备能同步工作并相互控制，因此具有强大的软件开发能力和对实时系统进行软、硬件综合调试能力。

5.单机仿真型与多机仿真型开发系统

单机仿真型开发系统与多机仿真型开发系统的主要区别在于，后者可以通过多台仿真器同时对用户系统上的多个目标处理器或单片机进行同步、实时仿真，而前者只能单独对其中的一个目标处理器或单片机进行仿真。

1.3.2 单片机开发系统的结构

开发系统的结构一般分为两大类^[16]，即主/从结构和单处理器结构。

1. 主/从结构

如图 1.1 所示，该结构有两个处理器，主处理器控制所有的开发功能，如编辑、汇编等。主处理器也控制存储设备及管理所有的 I/O。开发系统为主处理器编制的软件通常与从处理器(即目标处理器)不一样。从处理器与用户样机的典型连接是通过 40 引脚的连接器(引脚数随处理器的不同而变动)，该器件将主处理器和从处理器连接起来。

此结构的优点是:当从处理器连续仿真时，主处理器可以用于编辑、汇编等工作;主处理器的软件操作系统只需要编写一次，而且不管正在仿真的的是什么型号的微处理器，都能使用它;当要仿真的是一个新的从处理器时，只需改变仿真探头即可。

此结构的缺点是:存储器不连续，因为它被隔离成许多块。这种独立的总线结构使它在某些系统中工作起来很困难。在某些情况下，主/从结构的价格高也是一个缺点。

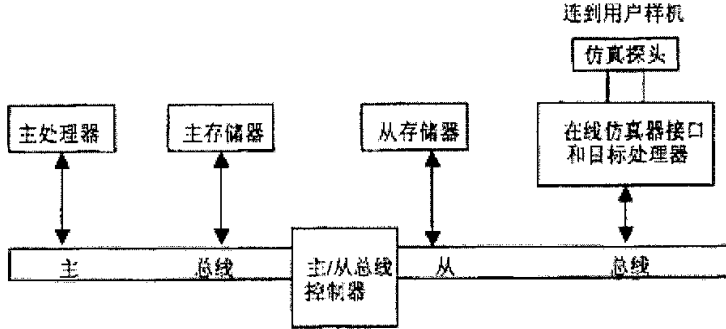


图 1.1 主/从结构

2. 单处理器结构

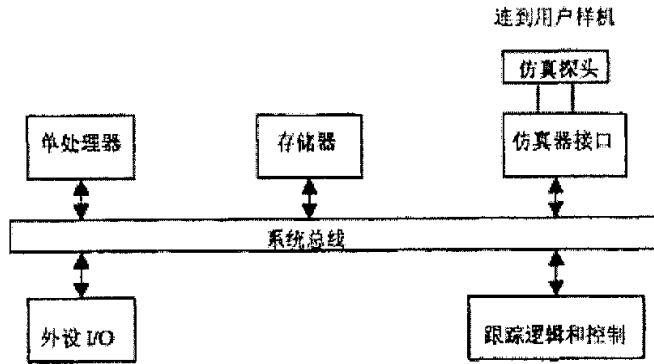


图 1.2 单处理器结构

如图 1.2 所示,单处理器结构就是用一个处理器来完成系统操作和目标仿真。系统处理器做两件事,即执行系统软件并充当目标处理器。

该结构的优点是:结构简单,价格低廉。

该结构的缺点是:系统软件随目标处理器的不同而需要改写,又由于系统软件必须驻留在仿真器使用的同一存储器中,因此,对仿真过程来说,不是所有的存储器都能用。

1.3.3 单片机开发系统的组成

开发系统是用来研制以微处理机为基础的系统(目标系统)的通用型设计和调试工具,即软件、硬件综合性研制工具。运用开发系统设计、调试成功的目标系统最终将不依赖于开发系统而独立工作。开发系统可以提供软件开发、实时仿真调试、逻辑分析及性能分析等综合性和交互性的开发功能。早期的开发系统与通用微型计算机系统无多大差别,它主要提供对软件开发的支持,自从出现了在线仿真器之后,开发系统才真正具有名副其实的独立意义。在线仿真器能够把开发系统的资源和调试功能直接扩展到目标系统中去。在调试环境下,在线仿真器既属于开发系统,又属于目标系统。无论在设计阶段,还是在调试阶段,用户通过在线仿真器能够最大限度地调用开发系统所提供的资源。在线仿真器通常被设计为开发系统的子系统或外部设备。对于每一种类型单片机的开发,都必须要有相应类型的在线仿真器。

一个完整的开发系统由硬件和软件两大部分组成^[1],其典型硬件配置如图 1.3 所示。

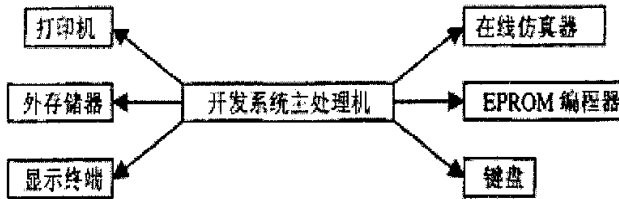


图 1.3 开发系统典型硬件配置

1. 主处理机

它是开发系统的核心,一般是一个带有监控程序的单板机或一个带有操作系统的微计算机。开发系统的硬件动作和软件运行,都由它来控制。

2. 控制台

控制台包括键盘和显示终端两部分,它是实现人机对话的必备部件。操作人员可通过键盘向开发系统下达各种命令。命令执行的结果通过显示终端显示出来,供操作人员检查正确与否。

3. 外存储器

通常指硬磁盘或软磁盘。主要是用来存放开发系统的系统软件和暂存用户的应用程序。

4. 打印机

在调试软件的过程中，时常需要打印许多中间信息，以便判别应用程序的故障所在。另外在用户最后开发工作完成时，为了整理用户系统的资料，要打印用户程序清单等。

5. 在线仿真器(In-Circuit Emulator, 简称 ICE)

这是开发系统的关键部件，缺少这一部件就不能称为开发系统。在单片机开发系统上编好应用程序后，一项重要的工作就是调试程序，而且最好是在用户环境中调试程序。在线仿真器正是起到了这个作用。它为用户程序的调试提供运行环境，把目标硬件(用户样机)与开发系统联系起来，使用户可以在开发的实际目标硬件环境下，全面调试用户程序。在线仿真器就是通过开发系统与目标系统的连接，用一个对用户来说具有“透明性”和“可控性”的仿真微处理器去取代用户目标系统中的目标处理器，仿真微处理器对用户目标系统的控制作用和工作过程几乎与实际的微处理器完全一样，从而实现开发系统对目标系统的控制和仿真操作。

6. EPROM (Electrically Programmable Read-Only Memory)电可编程只读存储器编程器

在完成程序调试后，用户必须把程序和固定数据写入 EPROM 中，使用户样机脱开发系统自己运行。EPROM 编程器可以将已调试完毕的用户程序及固定数据从单片机开发系统写入 EPROM 。

7. 开发系统软件

开发系统软件，一般由编辑程序、汇编(或编译)程序及动态调试程序三部分组成。其中编辑程序和汇编程序的运行并不需要在线仿真器的配合，而动态调试程序则要与在线仿真器配合才能运行。

近几年，随着 IBM-PC 个人计算机系统的不断普及，国内外推出了不少以 PC 机为基础的单片机开发系统，它们的典型结构如图 1.4 所示。微机开发系统和一般通用计算机系统相比，在硬件上增加了目标系统的在线仿真器、逻辑分析仪、编程器等部件，此外系统软件中除了一般计算机系统所具有的操作系统、编辑程序、编译等以外，还增加了目标系统的汇编和编译系统以及调试程序等。

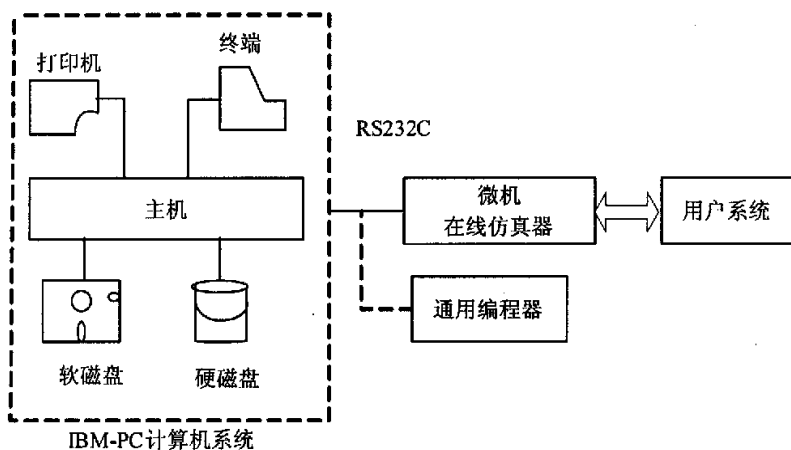


图 1.4 以 PC 机为基础的微机开发系统结构

1.3.4 国内开发系统的研究现状及发展趋势

自 1972 年 INTEL 公司研制成功了开发系统的第一个雏型 SIM—4—01 系统以来,各厂商纷纷不断推出自己的新产品。致使微机开发系统(MDS) 至今已经过了几代的发展,可归纳为以下几类:

独立系统— 一般只可开发某一种类型的微处理机,而且是单用户专用型的。

多用户通用系统— 可供多个用户同时使用。

开发站网络— 采用通过标准接口可与任何主机通讯的设备,使原来的独立系统既可独立工作,又可联入大系统中工作。

多功能系统— 在开发站网络的单工作站中加入逻辑分析和仿真能力,并配有各种类型的微处理机,大大提高了设计者的工作效率。目前,16 位 CPU, UNIX 操作系统,高级语言和模块化结构正被当前运行的多功能系统所采用。这种趋势,可能使今天的独立工作站扩展为未来的通用工程工作站。

由于微处理器当前有两个发展方向:一个是向高速多位并行微处理器方向发展;另一个是向多微处理器系统方向发展,构成集散控制系统、分布式系统、容错系统等多机系统。因此开发系统也正围绕着这两个方向加速发展:

- 1.越来越多的开发系统可支持 16 位和 32 位微处理器;
- 2.开发系统向着支持多微处理器系统的方向发展;

除了以上两个主要发展方向外,开发系统还应在以下几个方面有更大的突破:推出全透明、高性能及高速度的仿真器;加强状态分析和逻辑分析能力,提高对用户系统的软件、硬件综合测试能力;提供完善的软件支持;降低成本,使价格趋向合理化。

1.3.5 国内仿真器的研究现状及发展趋势

虽然开发系统有各种不同的结构和特点,但都具有在线仿真器这一重要部件。这是微机开发系统与一般微机系统的主要差别。

所谓仿真是指用一个拥有样机 CPU 所有功能的调试装置来取代该样机的 CPU,这种调试装置具有协调该样机软、硬件工作的能力。我们把这种调试装置称为在线仿真器。在线仿真器是 1975 年美国 INTEL 公司提出的概念,其后技术发展迅速。随着电子技术的更新换代数字仿真器取代了模拟仿真器。在各种控制系统的开发过程中在线仿真器为系统的软硬件联调提供了很大的方便。在线仿真器就是通过开发系统与目标系统的联接,用一个仿真微处理器取代目标系统中的微处理器,通过开发系统对目标系统进行控制和仿真操作。它综合软件开发、硬件调试和逻辑分析的功能,是一种综合性的开发工具。

仿真器是随可编程大规模集成电路的出现和在软件量急增的情况下产生的。它可帮助硬/软件人员调试他们所设计的系统和程序,与一般微机系统相比,以下几点描述了在线仿真器的所特有的功能:

- 当设计者测试他的软件时,将碰到的第一个问题是建立一个执行环境,在线仿真器通过映象部件提供了这一特性。如果样机可用,则通过提供在线仿真来完成这一功能,若样机不可用,则先进行资源转借。
- 硬件断点比较器这一强有力的断点机构提供了对程序流的完全控制。
- 借助于高速静态存储器,通过单步和跟踪的方式,可支持对程序流的观察。
- 通过显示存储单元和 I/O 口可查询程序变量和程序状态信息。

- 符号调试和对跟踪信息的解释为设计者提供了更为方便的调试手段。

到今天为止，国内在线仿真器发展大约有 20 年的历史，总的说来生产商的规模都不是很大，在新产品的研制、开发、服务上都不能投入很大的资金。根据仿真器使用的技术来划分，国内仿真器的设计大约可以分成以下几个时期：^[1]

1. 70 年代末期 · 80 年代中期

这个时期采用的技术主要是仿真开发系统。这种技术主要在仿真器的初级阶段使用。由于当时没有好的仿真芯片，在线仿真器被设计成了一个双平台的系统，并根据用户的要求在监控系统和用户系统中切换。这种仿真系统性能完全依赖于设计者的水平，实际的最终性能厂家之间相差很大。不过总的说来需要占用一定的用户资源，并且设计复杂。现在基本上已经淘汰，只是使用在一些开发学习系统中。

2. 80 年代末期 · 90 年代末期

这个时期主要采用的是 Bondout 技术。一般来说，人们常说的专用仿真芯片其实就是 Bondout。这种仿真芯片一般也是一种单片机，但是内部具有特殊的配合仿真的时序。当进入仿真状态后，可以冻结内部的时序运行，可以查找修改在静止时单片机内部的资源。使用 Bondout 技术制作的在线仿真器一般具有时序运行准确，设计制作成本低等优点。Bondout 芯片一般是由单片机生产厂家提供的，因此它只能仿真该厂商指定的单片机，仿真的品种很少。

3. 2000 年至今

2000 年至今主要采用 HOOKS 技术和嵌入式仿真技术。HOOKS 是 PHILIPS 公司拥有的一项仿真技术，主要解决不同品种单片机的仿真问题。使用该专利技术就可以仿真所有具有 HOOKS 特性的单片机，即使该单片机是不同厂家制造的。使用 HOOKS 技术制造的仿真器可以兼容仿真不同厂家的多种单片机，而且仿真的电气性能非常接近于真实的单片机。但是 HOOKS 技术对仿真器的制造厂家的技术要求特别高，不同的仿真器生产厂家同时得到 HOOKS 技术的授权，但是设计的仿真器的性能差别很大。即使到了今天，也不是每个仿真器厂家都能生产 HOOKS 仿真器，即使生产出来性能也差异很大。

嵌入式仿真技术是随着芯片技术的发展而产生的，现在很多单片机生产厂商在芯片内部嵌入了仿真功能，一般通过 JTAG 接口或 BDM (Background Mode Debug) 接口进行控制。为了降低成本和增加可靠性，内嵌的仿真部分一般功能比较简单。

我们知道，对于不同的仿真器，在命令状态下完成的功能都是基本相同的，区别只在于为了仿真不同的样机 CPU 而选用了不同的仿真处理器及相应的控制线路和软件。所以，我们可以设想，把在命令状态下仿真器的操作让一个专门的微处理器(称为控制 CPU)管理，而再用一个微处理器(称为仿真 CPU)，主要用于仿真样机 CPU。这样，由控制 CPU 管理的主要线路和相应的控制软件就可供多个仿真器所共享。为了仿真不同的样机 CPU，只需更换由仿真 CPU 控制的个性板即可，从而使仿真器具有模块化的结构^[4]。

1.4 本文要完成的主要工作

- 1.在熟悉单片机内部结构和性能特点的基础上，对九六系列单片机仿真器的总体结构进行规划。
- 2.完成仿真器的硬件设计。
- 3.详细论述了仿真器的工作原理。
- 4.论述了硬件电路中几个主要元器件的选择及要实现的功能。
- 5.对 96 系列单片机的各型号产品和它们各自的特点进行了介绍。
- 6.对仿真软件进行功能模块划分并画出程序流程图。
- 7.对仿真器进行联机调试。

第2章 开发系统与单片机仿真器

2.1 通用型单片机仿真器的简介

我们通常所说的开发系统一般是由一些开发工具构成, 其中一个主要的开发工具就是单片机在线仿真器, 在线仿真器是开发系统的关键设备, 它能以与用户处理器相同的时序执行用户程序, 并按用户需要产生各种断点响应, 同时也可接受主机系统的命令, 对用户系统进行全面测试和数据传送。在线仿真器通常由控制电路、存储器、仿真电缆、接口电路等组成, 这里我们采用的是一个通用型的单片机在线仿真器(即 Single Chip Microcomputer In-circuit Emulator)。它是一个独立型结构, 可以将仿真软件, 调试程序, 编辑汇编等系统程序全部存于仿真器内部的EPROM中。通用型单片机在线仿真器采用模块式结构^[2], 主模块板为8031/8032单片机仿真器, 在它的仿真接口上外接不同的扩展仿真板就可组成不同系列的单片机开发系统。如外接一块48扩展仿真板, 组成MCS-48系列单片机开发系统。外接51扩展仿真板组成8751仿真器, 外接96/98BH仿真扩展板, 组成MCS-96系列单片机开发系统。MCS-51为单CPU方式仿真器, MCS-48和MCS-96为双CPU方式仿真器。

当此仿真器为双CPU方式时, 主CPU是主模块板上的8031, 开发MCS-96系列单片机时, 从CPU是96/98BH仿真扩展板上的8097BH, 主CPU完成编辑、交叉汇编、命令解释、调试控制、各种I/O设备的输入/输出等功能, 从CPU由主CPU控制, 按主CPU的命令执行各种调试程序。

通用型单片机在线仿真器是开发和维修单片微机应用产品的理想工具。它和终端或任何具有RS232串行口的计算机系统相连就构成通用单片微机开发系统, 它可以开发国内外应用最广的INTEL公司的MCS-48、MCS-51、MCS-96以及飞利浦51系列等单片微机的应用系统。

通用型单片机在线仿真器具有完善的仿真功能, 联机在线调试用户样机时, 不占用用户系统的任何资源和存储空间, 使用户系统联结仿真器和脱离仿真器独立运行时的运行环境一致、执行的程序一致、资源使用一致。它具

有对样机硬件故障的诊断命令和单拍、断点、连续运行用户机程序的命令，排除硬件故障和软件错误迅速、方便。

联机调试用户应用系统时，通用型单片机在线仿真器不仅可将机内的仿真处理机（MCS-48、51、96 单片机）完整地出借给用户系统，而且可以出借仿真 RAM 作为用户的目标程序存储器或数据存储器。

在通用型单片机仿真器的内部还具有 56K 的源程序存储器，用于存放用户汇编语言或高级语言源程序。仿真存储器和源程序存储器都具有掉电保护功能，使机内的用户目标程序或源程序不因掉电而丢失，保持用户研制工作的连续性。

2.2 开发系统的结构和功能特性

2.2.1 开发系统的结构分析

这里采用的开发系统是由主模块板、扩展仿真板、输入输出设备、固化器以及终端或主机等所组成，用户可以根据自己所需开发的目标系统的机种和条件酌情选用适宜的模块或设备以组成最佳的单片机开发系统^[2]，开发系统框图如图 2.1 所示。

开发系统的主模块板是单 CPU 方式的 8031/8032/8344/80C51FA/FB 等单片机仿真器，将这些单片机的应用系统通过 40 芯的仿真插头和仿真接口（CN1）相连，就可以在线仿真调试。

对于 MCS-48 和 MCS-96 系列单片机采用双 CPU 仿真方式，开发系统的主模块板作为主控机，在它的仿真接口（CN1）外接扩展仿真板，通过仿真板在和相应的单片机应用系统相连，实现一次性在线开发用户系统。

两个 RS232 串行接口插座 CN4 或 CN5，都具有波特率自动检测、设置的功能，可以和任何具有 RS232 串行口的终端或主机相连，用户便可在终端或主机上操作，输入各种命令，进行用户系统的软硬件调试。

对于采用主机系统的用户可以借用主机的显示器、磁盘、打印机及各种软硬件资源，可以利用仿真器本身具有的行编辑功能，也可以用主机上原有的屏幕编辑程序生成源程序文件，可以利用主机的交叉汇编先将源程序编译

成目标程序文件再将目标程序传送到仿真器进行调试，也可以将源程序先传到仿真器的源程序存储器再用仿真器的汇编程序将源程序编译成目标程序，再装入仿真 RAM 进行调试，生成的打印清单文件可以传送到主机磁盘，并可利用主机的打印机打印出来。

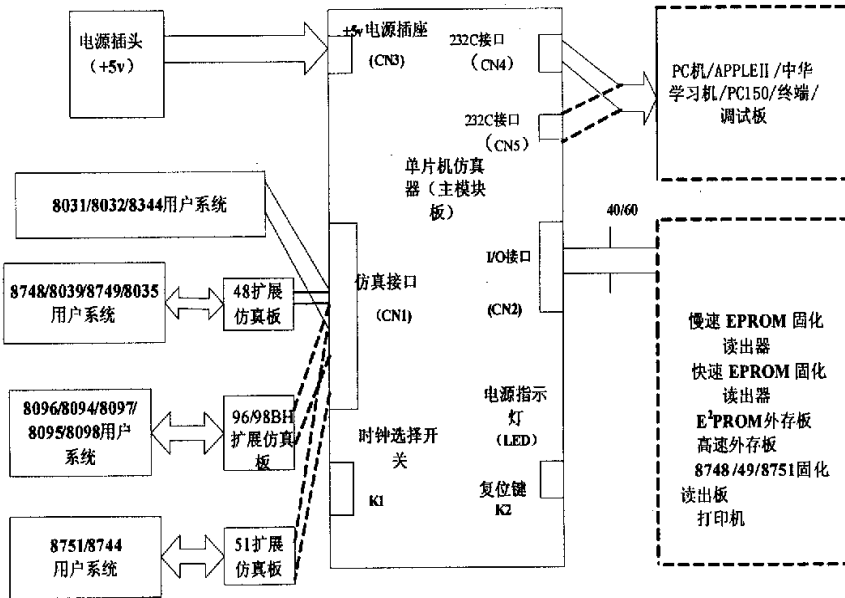


图 2.1 开发系统的结构框图

一个通用的 40 芯的输入输出接口 (CN2)，根据操作需要可外接相应的设备，其中通用 EPROM 固化板或高速 EPROM 固化板是必备的部件，因为最后调试好的目标程序文件都要固化到 EPROM 中；而打印机、快速外存板、E²PROM 外存板仅对没有主机或主机上没有打印机、磁盘驱动器的用户才是必须必备的，8748/8749/8751 固化读出板仅仅对于开发这些单片机的用户才是必须配置的。这些输入输出部件解决了终端或主机功能之不足，使用户能灵活地配置成一个功能齐全通用单片机开发系统。主模块板本身只需单一的+5v 电源即可工作，但在固化时，固化板需要固化电源 V_{pp} ， V_{pp} 电压随待固化的芯片类型不同而不同。

2.2.2 开发系统的软件功能

通用型开发系统具有极强的软件功能，内部具有固化的行编辑程序、MCS-48/MCS-51/MCS-96 汇编程序、ASM51 子程序库以及 MBASIC-51 高级语言编译程序。用户可以用汇编语言编写程序，监控软件中还包含有浮点四则运算、三角函数等子程序。用户可以宏调用各种子程序，也可以用 MBASIC-51 高级语言编写程序，这为用户研制目标机应用软件提供灵活而有效的手段，有利于提高开发新产品的速度。它的软件功能如下：

1. 编辑功能

仿真器内部具有行编辑程序 FD-EDIT，其功能和使用方法类似于 IBM-PC 机的 EDLIN 行编辑。它还具有 56K 源程序存储器，可用于存放 MCS-48、MCS-51 和 MCS-96 汇编语言源程序或 MBASIC-51 高级语言源程序。用户可以在终端上或主机的键盘上输入源程序，并可以随时列表显示、打印或修改源程序，还可以将源程序转贮到 E²PROM 或高速 RAM 外存储器板中，也可以存入到主机系统的磁盘上，也可将外存板或主机磁盘上的源程序读到仿真器的源程序存储器中，另外，对于连接 IBM-PC 等主机的系统，可以利用主机功能更强的屏幕编辑在主机上生成相应的单片机汇编语言源程序或 MBASIC-51 高级语言源程序，然后读到仿真器的源程序存储器中。

2. 汇编和编译功能

仿真器内装有 MCS-48、MCS-51 和 MCS-96 的各种汇编程序，它们都有用户符号处理功能和各种伪指令，能生成汇编清单供打印或存入主机的磁盘中，并能直接把产生的目标程序装入相应的仿真 RAM 存储器中供程序的调试、转贮或固化，它还具有功能特别强的 MBASIC-51 高级语言编译程序。

(1) FD-ASM51:

仿真器的主模板内配有 MCS-51 扩展汇编程序 FD-ASM51。它不同于一般仿真器的行汇编功能，是一个真正的 MCS-51 汇编系统。它能对 Intel 的《MCS-51 用户手册》中规定的标准格式 MCS-51 汇编指令进行汇编，允许指令中使用 8031、8032、8344、80C51FA/FB/80C552 的特殊功能寄存器名和各种位名，允许使用程序标号和各种符号。FD-ASM51 除了有 ORG、DB、DW、EQU 等常规伪指令外还具有功能特别强的宏调用指令，使用户可在汇编程序中直接调用浮点加、减、乘、除等各种运算和三角函数、指数、对数、开平方等各类函数运算符程序，在汇编时连同用户的源程序一起将宏调用的子程序汇

编成机器码装配到用户目标程序中。

(2) FD-ASM96:

在其中的一块扩展仿真板内配有 MCS-96 的汇编程序 FD-ASM96。它可对 Intel 的《MCS-96 用户手册》中规定的标准格式 MCS-96 汇编指令进行汇编, 允许在指令中使用各种特殊功能寄存器名 (如 P1、P2、T1、SP、BAUD、SBUF、IOCI 等) 和预定义的各种寄存器名 (如 AX、AH、AL、BX、CX、DX 等), 允许使用程序标号和各种用户符号。它除了具有 ORG、DB、DW、EQU 等常规伪指令外, 还具有数据段定义伪指令 DSEG、DSW、DSB 等。它允许使用寄存器直接寻址、立即寻址、寄存器间接寻址、自动加 1 间接寻址、加短变址、减短变址、长变址和直接寻址等 8 种汇编指令寻址方式。FD-ASM96 也具有极强的宏调用子程序的功能, 使用户能在汇编语言源程序中直接调用三字节或四字节浮点数加、减、乘、除子程序和各种函数运算符程序。在汇编时, 将用户编写的程序以及宏调用的子程序一起汇编成目标 (机器码) 程序装入到仿真 RAM, 使固化后, 用户样机脱离单片机仿真器后, 也能正常运行。这样可完全省去用户复杂的数据运算程序的编制和调试工作, 以利于加快产品的研制进度。

(3) MBASIC-51:

MBASIC-51 是一种适合于实时控制计算使用的编译型高级语言, 它用 MCS-51 汇编语言编写, 固化于仿真器的监控存储器中, 能对 MBASIC-51 高级语言源程序进行编译, 产生用户的目标程序, MBASIC-51 具有赋值语句、FOR-NEXT、IF-THEN、串行口输入输出等各种语句允许使用位、字节、整数和浮点等数据类型, 并能对各种数据类型进行转换, 能使用常数、变量和绝对地址变量数组, 具有 SIN、ATN、EXP、LOG、SQR、INT 等各种函数 MBASIC-51 采用汇编语言的标号结构, 并允许在源程序中任意使用汇编语言语句。MBASIC-51 是一种高速高效的编译系统, 用它生成的 MCS-51 目标程序具有运行速度快、程序长度短等特点。特别适合于实时控制和实时计算等场合使用, MBASIC-51 可以帮助用户高效率的研制 MCS-51 产品的应用软件。

3. 反汇编功能

除了汇编程序外, 仿真器还分别配有 MCS-48、MCS-51 和 MCS-96 等各种反汇编程序。可以对目标程序进行反汇编, 也可以对调试运行的指令进行反汇编。对目标程序进行反汇编所得到的程序清单可以打印或存入主机磁盘。

4. 调试功能

单片微机仿真器能控制用户系统以单拍、断点、跟踪、连续方式运行用户程序，在运行过程中，在屏幕上显示出的 CPU 的基本状态, PC, PSW, SP, ACC 以及各种特殊功能寄存器 I/O 口和工作寄存器的内容以及当前指令的汇编码，并可以输入各种修改命令来读出或修改单片机内部 RAM、SFR 和外部 RAM、I/O 的状态，使程序运行的结果显示在屏幕上，操作人员一目了然。

5. 全符号化调试功能

具有全符号化的调试功能。它允许使用汇编语言的标号作为运行控制命令中的起始地址或断点地址，因此可以按源程序进行调试，不必查找程序中指令的绝对地址，尤其在程序局部改动后（特别是插入一些指令）程序地址变动了，若按十六进制地址调试必须重新打印程序清单或者通过反汇编查出程序绝对地址才能进行调试。有了全符号化调试功能，则只要程序中标号名不变，就仍可按原来程序清单进行调试。

由于 MBASIC-51 采用汇编语言的标号结构，在调试用 MBASIC-51 高级语言编写的程序时，也可以用标号作运行控制命令中的起始地址和断点地址，逐个语句或逐段程序地进行调试，用户可以不必关心某语句或某程序段是由哪些指令组成，以及它们的存储地址。

2.3 开发系统中的主模块板的结构和功能

开发系统中采用的主模块板是单 CPU 方式的 MCS-51 系列单片微机仿真器。若采用 8032 则能开发 8031/8032 的用户系统；若采用 8344 则能开发 8344 的用户系统。若采用 80C552 可开发 80C552, 80C652 的用户系统。因此在开发不同机种的单片机产品时，我们可以对其中的 CPU 作相应的调整。开发系统中的主模块板的逻辑框图如图 2.2 所示。

主模块板内部具有 32KEPROM 作监控程序存储器，它分成若干块，离散地分布在 0~FFFFH 的空间内。内部的 0~7FFFFH 的 32K 仿真 RAM 仅作为用户的目标程序存储器；0~7FFFFH 的 32K 源程序存储器仅作为用户的源程序存储器；8000H~DFFFFH 的 RAM 是多用途存储器，可以作源程序存储器，也可以出借给用户系统作为用户的目标程序存储器，也可以借给用户作为数据存储器使用，

用户可以灵活地使用这个区域，使目标程序存储器或源程序扩大为 56K；0E000~FFFFH 为系统工作区。内部的这些 RAM 存储器具有掉电保护功能，使用户的源程序或目标程序不因掉电而丢失，使用户研制工作保持连续性。

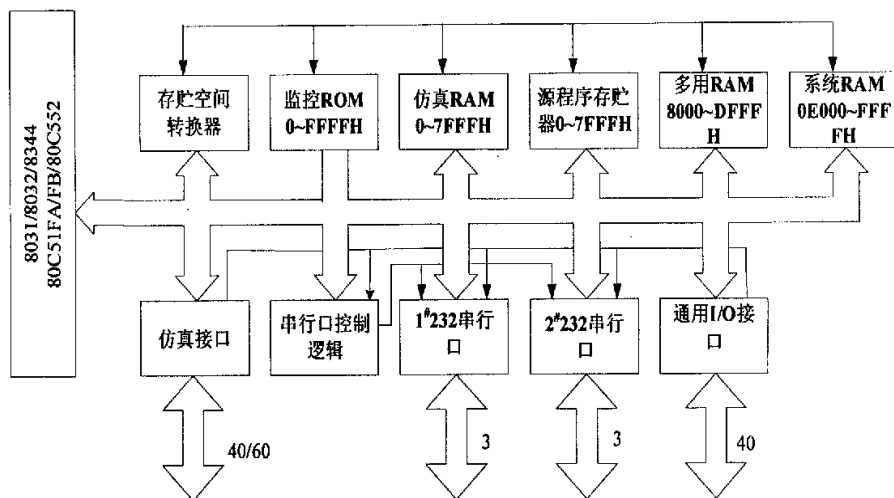


图 2.2 开发系统主模块板逻辑框图

主模块板内的监控、各种 RAM、用户 ROM 等存储空间的转换由存储空间转换器控制。当采用单 CPU 方式在线仿真调试 8031 目标系统时，采用切换 CPU 的方法，实现仿真器和目标系统对 8031 的资源共享。仿真器中的 8031 有两个状态：监控状态和目标状态。在总线控制器的控制下，使 CPU 在这两个状态之间切换。在目标状态时，仿真器能将 8031 完整地出借给目标系统，用户可以把它视为目标系统中的 CPU。在联机调试目标系统的过程中，仿真器可以将零地址开始的 16K 字节仿真 RAM 作为目标系统的程序存储器。

主模块板内具有两个 RS-232 的串行口其中 2# RS-232 (CN5) 是由 8031/8032 的串行口经串行接口控制逻辑引出的，1# RS-232 (MCS-51 串行口) 是由 8250 异步串行接口电路经串行口控制逻辑引出的。用户可把终端或主机和 1# 串行接口相连，也可以和 2# 串行接口相连，监控将自动检测用户的终端或主机连接到哪一个串行口上，并自动检测终端或主机的串行波特率，并设置主模块板内串行口的波特率和它们一致。当主机或终端连接到 2#232 串行口时，则用户系统和仿真器的监控系统分时使用 8031/8032 的串行口，这种连接方法对于调试用户 MCS-51 串行通讯程序特别有效，这时允许用户系统借

用主模块板上的 CN5 串行口上的终端或主机，使用户程序的输出信息直接在仿真器的终端或主机屏幕上显示出来，主机或终端键盘上输入的命令被用户程序所接受，用这种方法非常方便用户串行通讯程序的调试，而不必要一定要在样机上接好终端等设备后才能调试通讯程序，这可节省用户系统研制过程中所需的设备。

主模块板中有一个 40 芯的通用输入输出接口可用于外接各种外部设备。还有一个 40/60 芯的仿真接口，它可以连 8031/8032/8344/80C552 用户系统，也可连不同的扩展仿真板，将仿真器扩展成为 MCS-48 或 MCS-96 的开发系统。

主模块板内部共有 56K 字节 RAM，这些 RAM 在开发 MCS—96 时，0~0F28H 约 4K，用作系统 RAM，存放用户符号表及系统的控制信息。0F29H~0BFFFH，共 52K 多，用作用户的源程序存储区。

2.4 开发系统的组成和使用方法

将主模块板上的 CN4 和 CN5 两个 RS232 串行口中的任何一个连到终端或 IBMPC 机上，可构成一个完整的开发系统。这两个串行口具有相同的数据传送方式和功能。通常主机或终端应连到 CN4 接口，特别是当用户样机中 MCS-51 的 P3.0、P3.1 用作第一功能输入输出而不作为串行接口使用或开发 8344 时，终端或主机应接到 CN4 串行口接口。当用户 8031/8032 的 P3.0、P3.1 作为串行口使用时，终端或主机可以连到 CN4 也可以连到 CN5 上这里的串行通讯采用 8 位数据位、2 位停止位、无奇偶校验的格式，可支持的波特率有 9600、4800、3600、2400、1800、1200、900、600、300 和 150。这两个串行口的波特率都采用自动测试和设置的方法，它能按照外接的终端或主机的波特率自动跟踪设置仿真器串行口的波特率，使之和主机或终端的波特率相一致。

我们在使用这个仿真器时，先用 RS-232 专用连接插头和终端或主机相连，然后接通单片机仿真器和终端电源，若连主机则打开主机电源的同时还需运行相应的通讯程序并设置好波特率，这时在键盘上按一下回车键以启动仿真器开始工作并进行波特率检测和设置，如果连接无误屏幕上将显示相应的提示符。此仿真器具有行内编辑功能，可以接收大写或小写字母，数字及

各种符号，对于小写字母，可以在内部把它转换成大写字母，并配有暂停、重新启动和停止等快捷键来帮助命令的执行。还可以将屏幕上显示的信息在接上打印机的情况下打印出来。

2.5 本章小结

本章主要介绍了通用型单片机仿真器和它所组成的开发系统。首先介绍的是通用型单片机仿真器的特点及它所能实现的仿真功能。同时给出了开发系统的结构和功能。接下来重点介绍了主模块板的结构并给出了主模块板的逻辑框图，结合逻辑框图对它的一些主要的功能模块进行了详细的论述。通用型单片机仿真器具有极其丰富的软件功能其中包括编辑、汇编和编译以及反汇编功能，并且还有全符号化调试功能。最后，从一个完整的仿真系统的角度叙述了它的系统组成以及使用方法。

第 3 章 MCS-96 系列单片机及其开发系统

将通用型单片机仿真器 (Single Chip Microcomputer In-circuit Emulator) 的主模块板的仿真接口上外接一块 96/98BH 仿真扩展板, 再连接终端或主机构成一个可以开发 MCS-96 系列单片机的开发系统。其结构如图 3.1 所示。

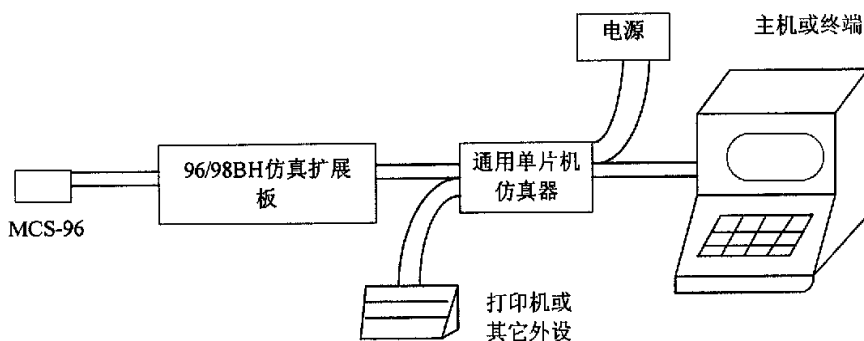


图 3.1 MCS-96 系列单片机开发系统构成

连接时, 96/98BH 扩展仿真板与主模块板的 8031 仿真插座相连, 在主模块板的扩展插座上可外接打印机, 也可外接 EPROM 读出写入板或 E²PROM 外存板、快速外存板等。主机如 IBM-PC、APPLE-II、LASER-310 等或智能终端通过 RS-232 插头与主模块板的两个串行口中的任何一个相连。这个开发系统连接主机的方法与其主模块板的相同。它所使用的电源由主模块板供给, 故不需另外接电源。

3.1 MCS-96 系列单片机简介

MCS-96 系列 16 位单片机由 Intel 公司于 1983 年推出, 特别适用于自动控制系系统。该产品包含 2 个子系列: 8096 和 8096BH。8096BH 完全兼容 8096, 但前者在性能上比后者更强, 且 8096BH 系列提供了带 8k 字节内部 EPROM 的型号产品, 该 EPROM 能采用几种编程方式, 包括在运行中编程的方式,

而 8096 系列不提供带 EPROM 的型号。性能上, 8096BH 的 A/D 转换速度高于 8096 的速度。8096BH 中还多一个芯片配置寄存器 (CCR)。除了以上所叙述的的两个子系列的差别之外, 还可以根据有无内部 A/D 转换器, 有无内部 ROM 或 EPROM, 是 68 引脚还是 48 引脚, 以及不同的封装形式, 把 MCS-96 系列细分为不同的型号产品, 见表 3.1 所示。

表 3.1 MCS-96 系列产品型号

| | | 无 A/D | 有 A/D |
|---------|------|-------------------|-------------------|
| 无 ROM 型 | 48 脚 | 8094—90 8094BH | 8095—90 8095BH |
| | 68 脚 | 8096—90 8096BH | 8097—90 8097BH |
| ROM 型 | 48 脚 | 8394—90 8394BH | 8395—90 8395BH |
| | 68 脚 | 8396—90 8396BH | 8397—90 8397BH |
| EPROM 型 | 48 脚 | 8794BH | 8795BH |
| | 68 脚 | 8796BH | 8797BH |

MCS-96 的主要性能特点: MCS-96 是具有最高性能的单片微型计算机系列产品之一。它的主要性能特点如下:

1. 16 位的 CPU (中央处理器)

该 CPU 在结构上的最大特点是: 没有采用习惯的累加器结构, 改用寄存器—寄存器结构, CPU 的操作直接面向 256 字节的寄存器空间, 消除了一般结构中存在的累加器的瓶颈效应, 提高了操作速度和数据吞吐能力。

16 位 CPU 支持位、字节和字操作, 在部分指令中还支持 32 位双字操作, 如 32 位除 16 位。

2. 高效的指令系统

该指令系统可以对不带符号和带符号数进行操作, 有 16 位乘 16 位和 32 位除 16 位的乘除指令, 有符号扩展指令, 还有数据规格化指令 (有利于浮点

计算)等等。

MCS-96 的很多指令既可用双操作数,也可用 3 操作数。后者包括两个源操作数和 1 个目的操作数,指令执行后 2 个源操作数的值不变。

3. 10 位 A/D 转换器

在 MCS-96 系列的部分型号产品中,有一个 8 通道或 4 通道的 10 位 A/D 转换器,使得这些型号产品特别适用于多路数据采集系统、智能仪器、控制系统等应用领域。

4. 脉宽调制输出

作为 D/A 转换器输出, MCS-96 单片机可以直接提供脉宽调制信号,某些电机可以用它直接驱动。脉宽调制输出信号经过积分就可以获得直流输出。

5. 全双工串行口

MCS-96 具有与 MCS-51 兼容的全双工串行口。这个串行口有 4 种操作模式,使它方便地用于 I/O 扩展、与 CRT 终端等设备进行通信及多处理机通信。

6. 高速输入/输出器

高速输入器可以相对于内部定时器产生的实时时钟,记下某个外部事件发生的时间,共可以记下 8 个事件;而高速输出器可以按规定的时刻去触发某一事件,任何时候,都可以悬挂起 8 个事件。“高速”意味着这些功能是“自动地”实现的,无需 CPU 的干预。

7. 5 个 8 位标准输入/输出口

除了上述输入/输出器之外, MCS-96 还有 5 个 8 位 I/O 口。

8. 8 个中断源

8 个中断源对应着 8 个中断矢量,而有些中断矢量又对应着多个中断事件,共对应 20 种中断事件。

9. 16 位监视定时器

产生软硬件故障时,监视定时器将使系统复位,提供了一种软硬件故障的恢复能力。

10. 可动态配置的总线

在运行过程中, MCS-96 的总线可以动态地配置成 8 位的或 16 位的,以便适应对外部存储器进行字节操作或字操作的不同需要。

11. 8K 字节的内部 ROM

MCS-96 的总存储器空间为 64K 字节, 在带内部 ROM 的芯片中内部 ROM 的容量为 8K 字节。

12. 256 字节的寄存器阵列和专用寄存器

其中 232 字节为内部寄存器阵列, 它兼有一般微处理机中通用寄存器和高速 RAM 的功能。另外 24 字节为专用寄存器。

13. 2 个 16 位定时器

其中定时器 1 在系统中作实时时钟用, 系统运行时, 不停地循环计数。定时器 2 根据外部引脚的触发而计数, 实际上是一个外部事件计数器。

14. 4 个软件定时器

4 个软件定时器受高速输出器控制, 到达预定时间时, 设置相应的软件定时器标志, 可以激活软件定时器中断。

3.2 MCS-98BH 简介

MCS-98BH 是 Intel 公司 1986 年在 MCS-96 的基础上改进生产的一个 MCS-96 增强型 16 位单片机系列, 该系列包含 8094BH、8095BH、8096BH、8097BH、8394BH、8395BH、8396BH、8397BH、8794BH、8795BH、8796BH、8797BH 和 8098 等型号, 其中 809×BH 和 8098 内无 ROM, 839×BH 有 8K×8 片内 ROM, 879×BH 有 8K×8 片内 EPROM, 最常用的为 8097BH 和 8098, 价格最低、使用比较方便的为 8098。下面简要介绍 809×BH 和 8098 的功能。

3.2.1 809×BH 的结构和特性

809×BH 和 8098 与原来的 809×—90 向上兼容, 它具有 809×—90 的一切功能, 如有 232 字节片内寄存器、高速 I/O 系统、全双工串行口 (包括波特率发生器)、脉冲宽度调制输出、16 位监视定时器、10 位 A/D 转换器等, 指令系统也与 809×—90 完全相同, 如有 16 位乘 16 位积为 32 位的乘法指令和 32 位除以 16 位的除法指令等。它的一些管脚的性能如下:

NMI: 非屏蔽中断, 当此引脚有正跳变时, 清除监视定时器, 形成一个

指向外部存储器 0000H 单元的中断矢量，而外部存储器 00H 至 0FFH 是保留给 Intel 开发系统用的。

INST：在读外部存储器期间此脚输出高电平，则表示是取指读操作。在整个总线周期内 INST 都有效。

$\overline{\text{ALE}}/\overline{\text{ADV}}$ ：地址锁存（ALE）或地址有效输出（ADV），由 CCR 选择。两者都提供了一个锁存信号，以便把地址从地址/数据总线中分离出来。当选定 $\overline{\text{ADV}}$ 功能时，在总线周期结束时，此引脚变高。 $\overline{\text{ADV}}$ 可作为外部存储器的片选信号。只有在外部存储器访问期间才会激活 $\overline{\text{ALE}}/\overline{\text{ADV}}$ 。

$\overline{\text{WR}}/\overline{\text{WRL}}$ ：写外部存储器（ $\overline{\text{WR}}$ ）或写外部存储器低位字节 $\overline{\text{WRL}}$ 的输出信号，由 CCR 选择。选 $\overline{\text{WR}}$ 功能时，每次写外部存储器时， $\overline{\text{WR}}$ 都变低，选 $\overline{\text{WRL}}$ 功能时，只有写外部存储器的偶数字节(低位)时， $\overline{\text{WRL}}$ 才变低。

$\overline{\text{BHE}}/\overline{\text{WRH}}$ ：总线高位字节允许 $\overline{\text{BHE}}$ 或写外部存储器高位字节 $\overline{\text{WRH}}$ 信号输出，由 CCR 选择， $\overline{\text{BHE}}=0$ ，选择连接至数据总线高位字节的存储器， $\overline{\text{A0}}=0$ 选择连接至数据总线低位字节的存储器，因此，访问一个 16 位宽的存储器时，若 $(\overline{\text{A0}}=0 \wedge \overline{\text{BHE}}=1)$ ，访问的是低位字节，若 $(\overline{\text{A0}}=1 \wedge \overline{\text{BHE}}=0)$ ，访问的是高位字节；若 $(\overline{\text{A0}}=0 \wedge \overline{\text{BHE}}=0)$ ，则同时访问 2 个字节。如果选择 $\overline{\text{WRH}}$ 功能，当写存储器奇数字节（高位）时，此引脚变低。

READY：准备就绪输出信号，用来延长外部存储器访问周期，以便与慢速或动态存储器接口，或为了总线共享。若此脚为高，则 CPU 继续正常的操作。若此脚在 CLKOUT 下降沿之前变低，则片内存储器控制器进入等待状态。直至 CLKOUT 脚发生下一个负跳变且 READY 变高时为止。不用外部存储器时，READY 不起作用。通过 CCR 可以控制插入到总线周期中的等待状态数。READY 有微弱的内部上拉作用，因此没有外部驱动时，此脚为高。

3.2.2 809×BH 的一些特殊功能

809×BH 除了具有 809×—90 的功能外，还增加了以下一些特殊功能：

1. 数据总线宽度可变

809×—90 的外部数据总线为 16 位，而一般的存储器芯片（包括 EPROM 和 RAM）均为 8 位字长，这样在扩展 EPROM 或 RAM 时，必须偶数个芯片，

一半为高 8 位、另一半为低 8 位。对于一个具有 EPROM 和 RAM 的 809×—90 的最小系统，至少需要两片 EPROM、两片 ROM 和两片地址锁存器（74LS373）。

为了简化存储器，809×—BH 具有两种数据总线宽度：16 位或 8 位。在 16 位工作方式，809×BH 与 809×—90 基本相同。在 8 位工作方式时，809×BH 的扩展方法与 MCS-51 等 8 位机基本相同，只是 809×BH 没有程序存储器和数据存储器之分，它只有一个统一的 64K 地址空间。这时，我们可使用一片地址锁存器，一片 EPROM 和一片 RAM 来构成一个具有 EPROM 和 RAM 的最小系统。在 8 位工作方式时，对外部存储器的字（16 位）读入和写操作将分两次进行，但内部操作仍为 16 位，所以它的执行速度比 16 位工作方式慢 1/3 左右。

除了在复位可选择数据总线宽度外，68 脚的 8096BH 和 8097BH 在 16 位工作方式时，还允许在运行时通过 BUSWIDTH 输入脚动态地改变总线宽度，例如取指令可采用 16 位方式，而读写 RAM 可采用 8 位方式，这使系统的构成更加灵活。

2. 写脉冲控制

809×—90 具有 \overline{WR} 写脉冲，在进行字节方式写入时， \overline{WR} 必须与 \overline{BHE} 和 A0 组合，产生 \overline{WRH} 和 \overline{WRL} 信号，并且 \overline{BHE} 为动态信号，必须用触发器由 ALE 下跳时锁存 \overline{BHE} 。

为了简化外接电路，809×BH 允许在复位时置 \overline{WR} 和 \overline{BHE} 为 \overline{WRL} 和 \overline{WRH} 信号，从而省去了外接电路。同时也可置它们仍为 \overline{WR} 和 \overline{BHE} ，以使 809×BH 与 809×—90 兼容。

3. ALE 控制

809×—90 有一根地址锁存允许控制线 ALE，用作地址锁存器 74LS373 的锁存脉冲。为了方便地址译码控制，809×BH 允许在复位时置 ALE 为 \overline{ADV} ，这时它既可作为地址锁存器的锁存脉冲，又可用作地址有效信号（0 有效）。 \overline{ADV} 可用作外部存储器片选控制信号。

4. READY 控制

由于 MCS-96 的速度较快，如 12MHZ 晶振时，写脉冲宽度仅为 146ns，小于许多 I/O 接口芯片的规定值，所以 MCS-96 有一根 READY 输入线，用它

可使 MCS-96 进入等待状态。一般在访问慢速 I/O 部件时, 从外部使 READY 变为低电平, 在等待时间到后, 再使 READY 变为高电平, 从而结束 MCS-96 的等待。这样, READY 线的控制比较复杂, 一方面在访问慢速 I/O 时使 READY 变为低电平, 这比较容易实现—可由 I/O 的片选信号去触发, 另一方面需有定时功能, 在延迟时间到时, 使 READY 恢复高电平, 这需要计数器或单稳电路来实现。

为了方便 READY 控制, 809×BH 增加了内部 READY 控制功能, 在复位时可控制它的 READY 方式和延迟时间。在允许内部 READY 控制功能时, 当 READY 输入线变低时, CPU 进入等待状态, 然后延迟 1~3 个机器周期后, 自动退出等待状态, 不必在 READY 输入线变高后才退出等待状态。这大大方便等待的控制。

5.A/D 变换器

809×BH 的 A/D 变换器比 809×—90 有较大的改进。它的转换分辨率仍为 10 位, 但转换时间缩短一半, 在 12MH 晶振时, 只需 22 μ s。

809×BH 的另一个优点是内部具有采样保持电路。在启动某一路 A/D 后, 自动进行采样, 然后进行变换。这样 809×BH 的 4~8 路 10 位 A/D 使用将更方便, 适用面也将更广。

3.2.3 809×BH 的使用方法

在 809×BH 片内有一个芯片构成寄存器 CCR, 它控制数据总线的宽度、写脉冲方式、地址有效信号和内部 REDAY 控制功能等。

8096BH/8098 子系列产品在性能上比 8096 子系列改进了很多, 主要表现在它们多了一个用于存储芯片操作方式信息的 CCR (Chip Configuration Register) 寄存器。

通过 CCR 选择不同的操作方式, 可以简化存储器系统、接口要求和总线控制等。

1. CCR 寄存器

CCR 寄存器是一个特殊的专用寄存器, 它并不属于内部 RAM 中的专用寄存器, 因此在系统工作时, 无法通过对内部 RAM 访问的指令来改变 CCR

的内容。

8096BH/8098 的存储器空间中，有一个特殊的单元，叫芯片配置字节 (CCB)，它的地址是 2018H。在一般用户系统中，这个单元应属于程序存储器，事先加以编程。当系统复位且只有复位时，CCB 的内容才被装载到 CCR 中。装载后，CCR 的值直至下一次复位之前，都不能改变。在每个总线周期，8096BH/8098 都读一次 CCR。

访问 CCR 的方法如下：

- (1) $\overline{EA}=0$ ，由外部存储器访问 CCB；
- (2) $\overline{EA}=1$ ，由内部 ROM/EOPROM 访问 CCB；
- (3) $\overline{EA}=+12.5V$ ，即在编程时，CCR 是由另外一个称为编程芯片配置字节 PCCB 的单元来装载的，这个单元不是存储器映射单元，而是一个独立的 EPROM 单元。

图 3.2 给出了 809x BH 的 CCR 的寄存器各位的定义。

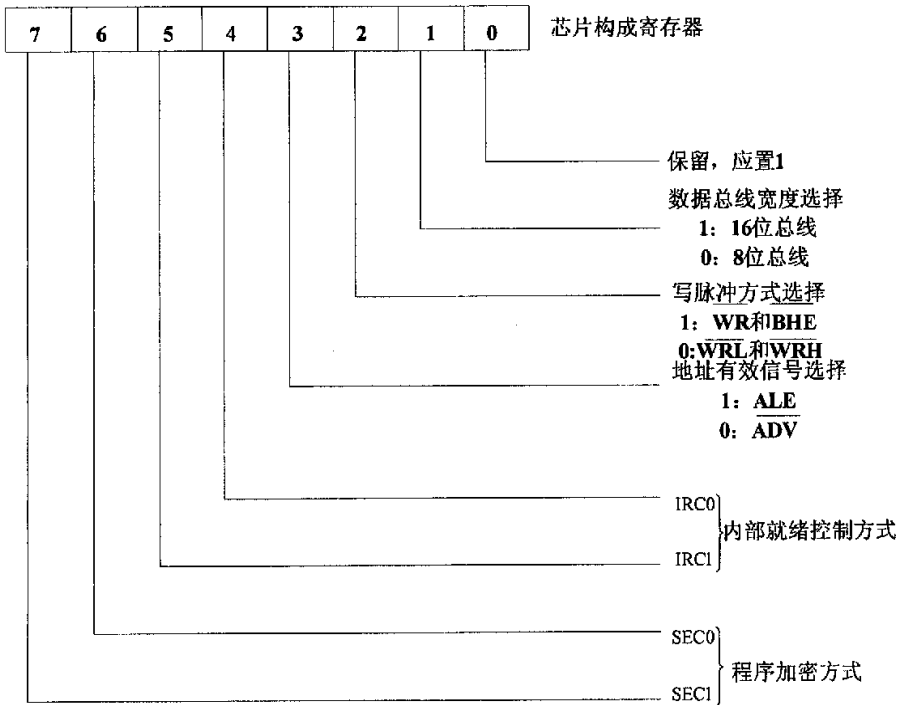


图 3.2 CCR 的定义

CCR 的内容在复位时从 2018H 单元读入。CCR 本身只能在复位时写入，在下次复位前它总保持复位时读入的值。不管总线为何种方式，809×BH 都能正确地在复位时读入 2018H 单元的内容。在使用 809×BH 时，必须在 2018H 单元放入芯片构成的控制信息。

2. 总线宽度选择

8096BH/8098 产品的外部总线可以在运行中动态的改变配置，配置成标准的 16 位多路转换的地址/数据总线，或配置成 8096 最小方式型的 16 位地址/8 位数据总线。

16 位总线周期中，P3 和 P4 作为 16 位地址和 16 位数据的多路转换总线，ALE 信号用于锁存地址。8 位总线周期中，地址仍为 16 位，数据是 8 位，P3 用作多路转换的地址/数据总线，P4 只作高 8 位地址线。在整个 8 位总线周期内，P4 上的地址一直是有效的。

在 68 脚芯片中，总线宽度受 CCR.1 和引脚 BUSWIDTH 的控制，当 CCR.1=0 或 BUSWIDTH=0，外部访问是通过 16 位地址/8 位数据总线进行的。CCR.1=1 或 BUSWIDTH=1，则外部访问是通过 16 位地址/16 位数据总线进行的。内部访问时，总线宽度总是 16 位。

CCR.1=1 时，总线宽度实际上受 BUSWIDTH 信号的控制。因此，通过运行中改变 BUSWIDTH 上的值，就可以动态地选择总线宽度。例如，我们可以把引脚 INST 的信号馈送给 BUSWIDTH 脚，这样在取指周期中，INST=1，使得从外部 EOPROM 取指时，总线宽度为 16 位，取 16 位的指令，在访问外部 RAM，即存取数据时，总线宽度为 1 个字节。

在 48 脚的芯片中，没有 BUSWIDTH 脚，故总线宽度由 CCR.1（只能选 0）单独决定。

3. 总线控制

用 CCR 可安排 8096BH/8098 使之提供若干类型的总线控制信号。为减少片外硬件，ALE 和 \overline{WR} 两条控制信号线被赋予了双重功能。由 CCR 的位 2 和位 3 指定。

当 CCR.2 和 CCR.3 均为 1 时，标准总线方式，那么它们所设定的乃是标准的控制信号 ALE 和 \overline{BHE} 和 \overline{WR} 。每次写操作都要产生 \overline{WR} 信号，当总线上开始出现地址码时，ALE 将升为高电平，此后的下降沿将作为时钟信号以

便把地址码的低 8 位锁存在外部锁存器中。在整个总线周期内 $\overline{\text{BHE}}$ 都有效，A0、 $\overline{\text{BHE}}$ 和 $\overline{\text{WR}}$ 组合以后以译码形成 $\overline{\text{WRH}}$ 和 $\overline{\text{WRL}}$ 信号。

当 CCR.2=0 且为 16 位总线周期时，为写选通方式 $\overline{\text{WRL}}$ 和 $\overline{\text{WRH}}$ 取代了 $\overline{\text{WR}}$ 和 $\overline{\text{BHE}}$ ，不再需要译码电路就可得奇偶字节的写选通信号。在 8 位总线周期中，CCR.2=0 仍提供 $\overline{\text{WR}}$ 信号。

当 CCR.3=0 时，为地址有效选通方式，ALE 的功能由地址有效 $\overline{\text{ADV}}$ 功能所取代，外部地址建立起来后， $\overline{\text{ADV}}$ 就变低，直至总线周期结束。 $\overline{\text{ADV}}$ 可以作为外部存储器的片选信号。

当 CCR.2 和 CCR.3 均为 0 时，在 16 位总线周期中选择了 $\overline{\text{ADV}}$ 和 $\overline{\text{WRL}}$ ， $\overline{\text{WRH}}$ 功能，8 位总线周期中仍能保持 $\overline{\text{WR}}$ 功能。

4. 就绪控制

为了简化就绪控制，8096BH/8098 内部提供了 4 种就绪控制方式这些方式的选择取决于 CCR.4 和 CCR.5 的值，如表 3.2 所示。

表 3.2 内部就绪控制

| IRC1(CCR.5) | IRC0 (CCR.4) | 说明 |
|-------------|--------------|------------|
| 0 | 0 | 最多延迟一个机器周期 |
| 0 | 1 | 最多延迟二个机器周期 |
| 1 | 0 | 最多延迟三个机器周期 |
| 1 | 1 | 禁止内部就绪控制 |

内部就绪控制逻辑用来限制等待的状态周期数，采用慢速器件时，可以把等待插入到总线周期内。这一特点使就绪控制变得较简单。

5. 程序保密

ROM/EPROM 型 839×BH 和 879×BH 芯片，对内部存储器可以采取 4 种程序保密方式，采用哪种方式由 CCR.6 和 CCR.7 的值决定如表 3.3 所示。写保护只对 EPROM 型芯片而言。

只有从内部存储器执行的代码可以读被保护的存储器。但是，即使是内部执行的代码也不能写被保护的存储器。

为了确保程序保密而同时又允许验证和测试，对于 839×BH 和 879×BH

芯片，在因编程和测试而需要读被保护的存储器之前，要求验证一下保密键码只有内外部保密键码吻合，才允许访问被保护存储器。

表 3.3 程序保密方式

| SEC1(CCR.7) | SEC2 (CCR.6) | 保护方式 |
|-------------|--------------|-------|
| 0 | 0 | 读和写保护 |
| 0 | 1 | 写保护 |
| 1 | 0 | 读保护 |
| 1 | 1 | 不保护 |

3.3 MCS-96 系列单片机开发系统的特点

MCS-96 系列单片机的开发系统是在通用单片机仿真器 (Single Chip Microcomputer In-circuit Emulator) 的主模块板的基础上外扩一个仿真扩展板构成的，它是一个真正的 MCS-96 十六位单片机开发系统，它具有仿真功能强、软件丰富、使用方便、外设齐全等特点。它可用于开发 8094BH、8095BH、8096BH、8097BH、8098、8094-90、8095-90、8096-90、8097-90 等 MCS-96 单片机，它的各种性能超过 Intel 公司的 iSBE-96 仿真器，是一种价廉物美的 MCS-96 开发系统。

MCS-96 开发系统的基本特点如下：

1. 可开发各种结构的用户系统：可通过命令来设置总线宽度为 16 位或 8 位，也可通过仿真插头的 BUSWIDTH 输入脚在运行时动态地改变总线的宽度，可通过命令来置写脉冲为 \overline{WR} 和 \overline{BHE} ，也可置为 \overline{WRL} 和 \overline{WRH} ；可通过命令置地址锁存线为 \overline{ALE} 或 \overline{ADV} ，可通过命令来置内部 READY 控制功能。这样 EM-96BH 可开发任意的 809×BH（包括 8098）用户系统。

2. MCS-96 开发系统不占用用户系统的任何存储空间，用户可任意使用 64k 地址空间。

3. 有 64k 仿真 RAM，可部分或全部借给用户系统作程序存储器，并可部分或全部借给用户系统作数据存储器，共有 8 种影像方式。

4. 具有 FD-EDIT 行编辑功能，有 52k~56k 源程序存储器，可输入、修改

汇编源程序。

5.具有 FD-ASM96 扩展汇编程序，可汇编用 MCS-96 标准格式汇编指令编写的源程序，具有标号和用户符号功能，并允许标号运算。

6.具有 CALL 宏子程序调用指令，可自动生成浮点加、减、乘、除、开平方、正弦、反正切、指数、对数、多项式、二翻十、整数转化成浮点数等 24 个子程序。这些程序具有速度快、长度短、使用方便等特点。

7.具有 MCS-96 反汇编程序，可对执行的指令或存储器中的机器指令进行反汇编。

8.采用硬件断点，允许设置多至 8000 个断点、数据读或写断点，并允许区域断点。允许用户任意使用 TRAP 指令和软中断。具有夭折处理能功能。

9.具有单拍、跟踪、全速软断点、全速硬断点、连续等各种运行方式。

10.具有全符号化调试功能，可使用汇编源程序中的符号（包括标号加数字）来启动和设置断点。

11.能实时仿真用户系统，仿真时不占用用户的任何资源。用户任意使用中断、定时器、AD 及 HSIO 等部件、用户堆栈可设在内部或外部，仿真时系统仅占用 2 级（4 个字节）堆栈区，少于 iSBE-96 和其他 MCS-96 仿真器。

12.能以字或字节形式读出或修改 8097 内部 RAM，也能对 8097 内部的各种特殊功能寄存器进行读出或写入操作。

13.能以字或字节形式读出或修改仿真 RAM 或用户的数据存储器的内容。

14. 能读出或调试用户样机板上 EPROM 中的程序。

15.可任意选择晶振，可使用 EM-96 内的 12MHZ 晶振，也可使用用户系统的晶振(6~12MHZ)。

16.可外接打印机，打印源程序或汇编清单，也可打印屏幕上显示的任何信息。

17.可外接 E²PROM 外存板或快速外存板来存放源程序。在外接主机时，可使用主机的磁盘作源程序存储设备，并可从主机接收 Intel 的 HEX 格式文件，装入仿真 RAM，使它可与 Intel 公司的各种软件兼容。

18.可外接 EPROM 读出/写入板，可对 2716、2732、2764、27128、27256、27512 等 EPROM 芯片进行固化或读出操作，并可自动进行奇偶地址转换、

分页。

3.4 本章小结

本章首先介绍了 MCS-96 系列单片机开发系统的构成。同时对 MCS-96 系列单片机所包含的子系列，以图表的方式给出了它们之间的区别。结合 MCS-96 系列单片机所具有的主要性能特点，对其中一些主要引脚的功能进行了说明。通过芯片配置寄存器的配置引出了 809×BH 的使用方法。最后给出了 MCS-96 系列单片机开发系统的一些基本特点。

第 4 章 96/98BH 仿真扩展板的硬件设计

96/98BH 是我们在开发 MCS-96 单片机时用到的的一块仿真扩展板，它的硬件电路由以下几大模块组成：存储器模块、接口电路模块、控制电路模块和 CPU 模块。各部分模块中所用到的元器件在以下各节有详细的介绍。

4.1 96/98BH 仿真扩展板的硬件结构

96/98BH 仿真扩展板的硬件结构框图如图 4.1 所示。

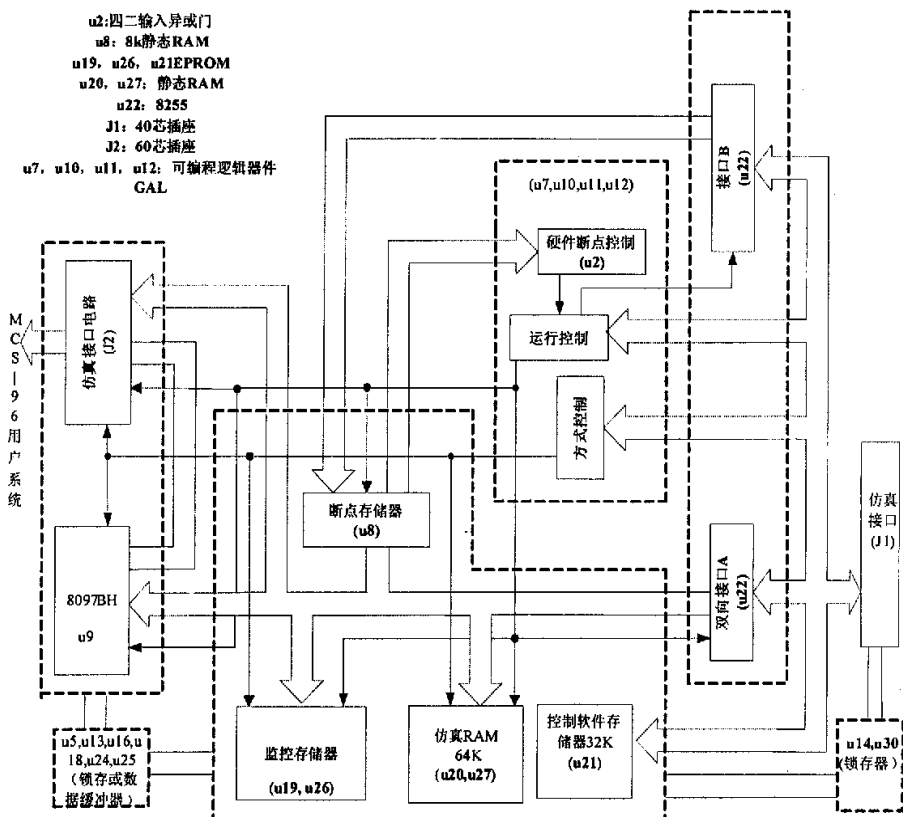


图 4.1 96/98BH 扩展仿真板硬件结构框图

图中 CPU 采用 8097BH,控制电路采用门阵列芯片,因而集成度高,结构简单。仿真扩展板中装有 64k 仿真 RAM,它们可以用作用户的程序存储器,还可以部分或全部借给用户作数据存储器。

96/98BH 仿真扩展板在控制软件的控制下运行。它允许设置多达 8k 个程序断点、数据读或写断点,断点设置由主模块板通过接口 B 完成。主模块板控制 8097BH 的工作方式和运行状态。主模块板通过双向接口 A 和 8097 通讯,当 8097 处于监控状态时,它接收主模块板的命令,可将 MCS-96 目标程序装入 96/98BH 扩展仿真板的仿真 RAM 中,也可将 8097 的内部状态或扩展的存储器、I/O 的内容送至主模块板,还可以根据用户命令修改现场状态或仿真 RAM 的内容。

96/98BH 仿真扩展板在开发用户系统时,通过仿真接口电路与用户样板板接口,样板板的故障一般不会影响开发系统的运行,可用各种操作命令来检查用户系统的故障。

4.2 96/98BH 仿真扩展板中元器件的选择及功能介绍

根据总体设计的要求合理的选择元器件,并根据 MCS-96 技术手册和有关参考资料,完成硬件电路的设计工作。对所选的元器件不仅要充分实现它的价值而且还要求对它的技术参数留有一定的余地。

4.2.1 GAL 可编程逻辑器件的选择

GAL是通用型阵列逻辑(Generic Array Logic)的简称,是美国Lattice半导体公司推出的低密度可编程逻辑器件。它采用的是E²CMOS技术,具有高速度、低功耗、可反复编程及编程后的信息能长期保存的优点。它是在PAL的基础上发展起来的,其间的差异主要是输出结构不同: GAL器件采用了可编程的与阵列和固定的或阵列。但它的输出结构采用了可编程的输出逻辑宏单元OLMC。因而可根据需要选择组合逻辑或寄存器输出,还带有三态门控制信号、I/O反馈途径及输出极性控制,使GAL在使用时更灵活。通过对OLMC的编程可改变输出的结构,从而决定电路的具体形式。GAL器件的或

阵列包括在OLMC中，因此GAL器件由一定数量的可编程的与阵列与OLMC构成。由于GAL器件的输出结构能通过编程灵活改变，因而此器件具有更强的通用性。

一个GAL器件的输入输出端的数量有限，尽管可通过编程来确定某些引脚处于输入或输出工作状态，但一个器件的输入输出端的总数受到引脚数量的限制。例如GAL16V8的输入输出端的总数为18个。此芯片的输入端的数量最多为16个，但此时其输出只能有2个，而选择其输出端的数量为最大值8时，其输入端仅为10个。因而在划分电路时，必须考虑到实现这一部分电路时可能采用的器件的输入输出端的数量。

GAL16V8宏单元的工作方式可以分为三类共5种方式：

1. 寄存器方式,在这种方式下，宏单元作为寄存器输出或作为I / O功能。
2. 复合方式，在这种方式下，宏单元可构造I / O功能或输出功能。
3. 简单方式，在这种方式下，宏单元可以构造三态门常通，输出端有反馈或无反馈及三态门常闭而该引脚作为输出端用。

GAL器件中的输出逻辑宏单元并不是随意扩展成某一形式的电路，它有一定的电路结构模式，对16V8控制码存放格式，如图4.2所示。

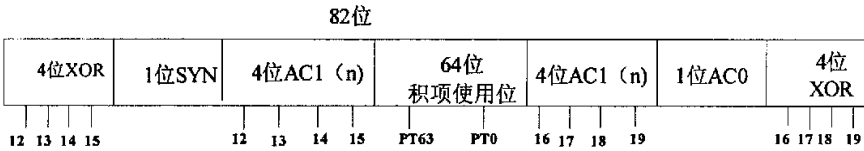


图4.2 GAL16V8控制码格式

图中SYN及AC0是由8个OLMC共用的，64位的PTD (Product Term Disable)用来对与阵列的大小作调整，某位为1，对应的积项送至或阵列，否则就不送，XOR控制输出为低态有效还是高态有效，AC1控制I / O的构成方式，由SYN及AC0的组合构成了GAL16V8的三种工作方式，如表4.1所示。

表4.1 GAL的三种工作方式

| SYN | AC0 | 方式 | SYN | AC0 | 方式 |
|-----|-----|-------|-----|-----|------|
| 0 | 0 | 不用 | 1 | 0 | 简单方式 |
| 0 | 1 | 寄存器方式 | 1 | 1 | 复合方式 |

96/98 BH扩展仿真板中共用了4个可编程逻辑器件GAL,这四个GAL器件作为控制电路分别负责完成不同的任务。U7的功能主要是用来实现断点地址的译码,产生非屏蔽中断信号,即对硬件断点进行控制。U10是对64K仿真RAM的输出信号进行译码选择,并且它作为8255双向端口A的控制联络信号,通过它来实现主模块板和8097BH之间的双向数据通讯。U11是一个译码器,它对仿真RAM存储器的片选信号进行译码,对8097BH读写信号译码输出到仿真接口信号端,对扩展板中EPROM监控存储器的片选信号进行译码输出,对64K仿真RAM的写信号进行译码输出。U12完成了仿真RAM中用户存储器地址空间8种映像方式的选择。

4.2.2 8255 可编程接口芯片的应用

8255 是一个通用的可编程并行 I/O 接口芯片,内部具有 3 个带锁存或缓冲的数据端口,可与外设并行进行数据交换。用户可用程序选择多种操作方式,因而通用性强,使用灵活。可为 CPU 与外设提供并行输入输出通道。其框图如 4.3 所示。

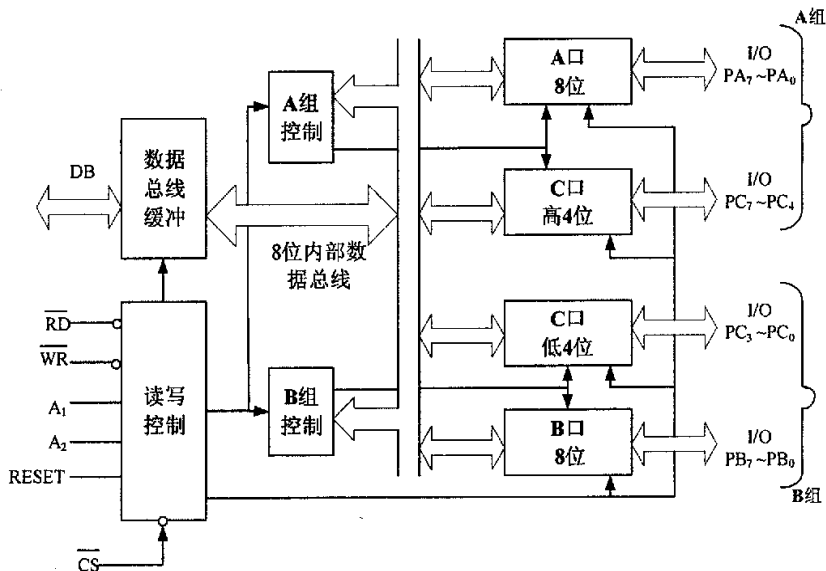


图 4.3 8255 内部结构

8255 可编程通用并行接口:

1. 8255 的结构

8255 具有 3 个可编程并行 I/O 口, A 口、B 口和 C 口。这 3 个 8 位 I/O 口的功能完全由编程决定, 但每个口都有自己的特点。

8255 可编程接口由以下 4 个逻辑结构组成。

1) 数据总线驱动器

这是双向三态的 8 位驱动口, 用于和单片机的数据总线相连, 以实现单片机与 8255 芯片间的数据传送。

2) 3 个并行 I/O 端口

A 口: 具有一个 8 位数据输出锁存/缓冲器和一个 8 位数据输入锁存器。是最灵活的输入输出寄存器, 它可编程为 8 位输入输出或双向寄存器。

B 口: 具有一个 8 位数据输入/输出锁存/缓冲器和一个 8 位数据输入缓冲器(不锁存), 可编程作为 8 位输入或输出寄存器, 但不能双向输入/输出。

C 口: 具有一个 8 位数据锁存/缓冲器和一个 8 位数据输入缓冲器(不锁存)。这个口可分为两个 4 位口使用。C 口除作输入输出使用外, 还可以作 A 口、B 口选通方式操作时的状态控制信号。

3) 读/写控制逻辑

它用于管理所有的数据、控制字或状态字的传送。它接收单片机的地址线和控制信号来控制各个口的工作状态。

4) A 组和 B 组控制电路

这是两组根据 CPU 的命令字控制 8255 工作方式的电路。每组控制电路从读、写控制逻辑接受各种命令, 从内部数据总线接收控制字(即指令)并发出适当的命令到相应的端口。

A 组控制电路控制 A 口及 C 口的高 4 位。

B 组控制电路控制 B 口及 C 口的低 4 位。

2. 8255 的外部特性。8255 共有 40 个引脚, 下面根据功能分类说明:

1) 数据总线: D0~D7、PA0~PA7、PB0~PB7、PC0~PC7, 此 32 条数据线均为双向三态。

D0~D7 用于传送 CPU 与 8255 之间的命令与数据, PA0~PA7、PB0~PB7、PC0~PC7 分别与 A、B、C 口相对应, 用于 8255 与外设之间传送数据。

2)控制线 \overline{WR} 、 \overline{RD} 、RESET

\overline{RD} ：读信号线，低电平有效。当这个引角输入低时，控制 8255 送出数据或状态信息到 CPU，即 CPU 对 8255 进行读操作。

\overline{WR} ：写信号线，低电平有效。当这个引角输入低时，控制把 CPU 输出的数据或命令写到 8255，即 CPU 对 8255 进行写操作。

RESET：复位信号线，高电平有效。当此引脚为高电平时，所有 8255 内部寄存器都清 0。所有通道都设置为输入方式。24 条 I/O 引脚，为高阻状态。

3)寻址线： \overline{CS} 、A0、A1

\overline{CS} ：为芯片选择线，低电平有效。当此引角为低电平时，本芯片被 CPU 选中。

A0、A1：这是两个输入信号，通常一一对应接到地址总线最低两位 A0 和 A1 上。当 \overline{CS} 有效时，这两位的 4 种组合 00、01、10、11 分别用来选择 A、B、C 口和控制寄存器。所以一片 8255 共有 4 个地址单元。

3. 8255 的工作方式

8255 有三种工作方式，即方式 0、方式 1、方式 2。

方式 0（基本输入/输出方式）：这种方式不需要任何选通信号。A 口、B 口及 C 口的高 4 位和低 4 位都可以被设定为输入或输出。作为输出口时，输出的数据被锁存，作为输入口时，其输入的数据不锁存。

方式 1（选通输入/输出方式）：在这种工作方式下 A、B、C 3 个口将分为两组。A 组包括 A 口和 C 口的高 4 位，A 口可由编程设定为输入口或输出口，C 口的高 4 位则用来作为输入/输出操作的控制和同步信号；B 组包括 B 口和 C 口的低 4 位，B 口可由编程设定为输入口或输出口，C 口的低四位则用来作为输入/输出操作的控制和同步信号。A 口和 B 口的输入数据或输出数据都被锁存。

方式 2（双向总线方式）：在这种方式下，A 口为 8 位双向总线口，C 口的 PC3~PC7 用来作为输入/输出的控制同步信号。应注意的是，只有 A 口允许作为双向总线口使用。这时 B 口和 PC0~PC2 则可编程为方式 0 或方式 1 工作。

4. 8255 的控制字

8255 的工作方式选择是通过控制口输入控制字（或称命令字）的方式来实现的。控制字有方式选择控制字和 C 口置/复位控制字。

96/98 BH 扩展仿真板中的 8255 的 A 口是按照方式 2 工作的，即 A 口作为双向数据总线端口，既可以发送数据，又可以接收数据。这时 A 口既可工作于查询方式，又可工作于中断方式。其控制联络信号是由 C 口的 PC4—PC7 实现的。PC3 作为中断请求信号与 J1 端口相连，在输入输出方式时可用于向 CPU 发中断请求。在这里可将 8097BH 看作是主模块的一个外设，是通过 8255 可编程并行接口实现的。PC4 作为 8097 的选通输入信号，PC5 作为输入缓冲器满信号，这两个信号作为输入时的握手信号。PC6 可作为外设响应信号，用来启动 A 口三态输出缓冲器输出数据，PC7 作为输出缓冲器满信号，是 8255 输出给外设的联络信号，表示 CPU 已把数据输出到 A 口。8255 B 口与断点寄存器的高五位相连，因此可通过 B 口来设置断点。

4.2.3 RAM 和 ROM 存储器的选择及存储空间的划分

96/98 BH 仿真扩展板中共扩展 2 片 8K 字节的 EPROM 和一片 32K 字节的 EPROM，还扩展了 1 片 8K 字节的 RAM 和两片 32K 字节的 RAM。这些存储器中 2 片 8K 字节的 EPROM 中装有 8097BH 的监控程序，由 8097 来运行这些程序。两片 32K 字节的 RAM 共 64K 字节空间称为仿真 RAM。它的 13 根地址线与 8097BHA1~A13 相连。它可出借给用户作程序存储器或数据存储器。在开发 MCS-96 用户系统时，它可以有 8 种基本的存储器映像方式。如表 4.2 所示。32K 字节的 EPROM 是控制软件寄存器。它是上电自动工作的，由主模板来控制运行。一个 8K 字节的 RAM 是断点地址寄存器。

表 4.2 用户存储器映像表

| 方式 | 0~0FF | 100~7FF | 800~1FFF | 2000~7FFF | 8000~9FFF | A000~FFFF |
|----|------------|------------------|----------|------------------|-----------|-----------|
| 0 | 8097 内部寄存器 | 用户板空间 | | 仿真 RAM 作数据和程序存储器 | | |
| 1 | 同上 | 仿真 RAM 作数据和程序存储器 | | | | 用户板空间 |

| | | | | | |
|---|----|----------------|-------|----------------|-------|
| 2 | 同上 | 仿真RAM作数据和程序存储器 | 用户板空间 | 仿真RAM作数据和程序存储器 | 用户板空间 |
| 3 | 同上 | 仿真RAM作数据和程序存储器 | | | |
| 4 | 同上 | 用户板空间 | | 仿真RAM作程序存储器 | |
| 5 | 同上 | 仿真RAM作程序存储器 | | | 用户板空间 |
| 6 | 同上 | 用户板空间 | | 仿真RAM作程序存储器 | 用户板空间 |
| 7 | 同上 | 用户板空间 | | | |

这里需要说明的是：用户板空间表示该空间用户可以任意使用，并且它位于用户系统中。仿真RAM作程序存储器，表示该空间处于96/98BH的仿真RAM中，并且作程序存储器，它不能由用户指令写入，只能由仿真器命令写入，故不能用作用户数据存储器，更不能用作堆栈区，其内容在运行时不会改变。

仿真RAM作数据和程序存储器时，表示该空间处于96/98BH的仿真RAM中，并且它可由用户指令写入或读出，所以它可作用户的数据存储器，也可用作程序存储器。但在用户程序运行出错时，可能会改变它的内容。

对于OFFFEH和OFFFFH，在监控状态下不能用命令读出或写入。但在运行状态，能由用户指令正确读出和写入。8种存储器映像方式是这样实现的：U20的片选信号线 \overline{CS} 是由可编程逻辑器件U11的14管脚控制的。U27的片选信号线 \overline{CS} 是由可编程逻辑器件U11的15管脚控制的。它们的逻辑方程式如下：

$$\begin{cases} U11(14) = E * F * \overline{G} + D * \overline{E} * F * \overline{G} = E * F * \overline{G} + D * F * \overline{G} \\ U11(15) = E * F * \overline{G} + \overline{D} * \overline{E} * F * \overline{G} = E * F * \overline{G} + \overline{D} * F * \overline{G} \end{cases}$$

从U11的逻辑方程式中可以推出，要使U20和U27这两个片子能够正常工作则有：

$$\begin{cases} E * F * \bar{G} = 0 \\ \bar{D} * F * \bar{G} = 0 \\ D * F * \bar{G} = 0 \end{cases}$$

由此可得： $F * \bar{G} = 0$ 要使等式成立，可知 $F=0$ 或 $G=1$ 。这里的D、E、F、G都是U11片子的输入信号。

当 $G=1$ 时，有 $AD_{15}=AD_{14}=AD_{13}=AD_{12}=AD_{11}=AD_{10}=AD_9=AD_8=0$ ，映射的地址空间为 $0\sim 0FF$ 。

当 $F=0$ 时，则由可编程逻辑器件U12（16）的逻辑方程式决定。经过对逻辑方程式的分析可得如下结论：

当 $G=1$ ，且

| | |
|---|---|
| { | $J_1(5) = 0$, $J_1(32) = 0$, $J_1(33) = 0$ 时为方式4。 |
| | $J_1(5) = 0$, $J_1(32) = 0$, $J_1(33) = 1$ 时为方式5。 |
| | $J_1(5) = 0$, $J_1(32) = 1$, $J_1(33) = 0$ 时为方式0。 |
| | $J_1(5) = 0$, $J_1(32) = 1$, $J_1(33) = 1$ 时为方式1。 |
| | $J_1(5) = 1$, $J_1(32) = 0$, $J_1(33) = 0$ 时为方式2。 |
| | $J_1(5) = 1$, $J_1(32) = 0$, $J_1(33) = 1$ 时为方式3。 |
| | $J_1(5) = 1$, $J_1(32) = 1$, $J_1(33) = 0$ 时为方式6。 |
| | $J_1(5) = 1$, $J_1(32) = 1$, $J_1(33) = 1$ 时为方式7。 |

这就是8种基本的存储器映像方式。根据存储器映像表，用户可以设计自己的存储器空间。用户在调试目标系统时，可通过系统命令来选用其中的任何一种存储器映像方式。

4.2.4 晶振电路的设计和其它元器件的选择

晶振电路的选择如图 4.4 所示：96/98 BH 扩展仿真板中晶振电路的选择是通过一个双向转换开关 S1 来控制实现的。开关的一侧接有 8097BH 的两个管脚 XTAL1 和 XTAL2，开关的另一侧接有一个 12MHZ 的晶振和 4 个电容，P2 是一个两端子的引出线，可以与用户板的晶振相连。用户可以通过转换开关来选择仿真时所需的时钟信号。这样设计使仿真器既可以采用用户的晶振电路提供的时钟频率工作，也可以选用仿真器内部晶振电路提供的时钟频率工作。

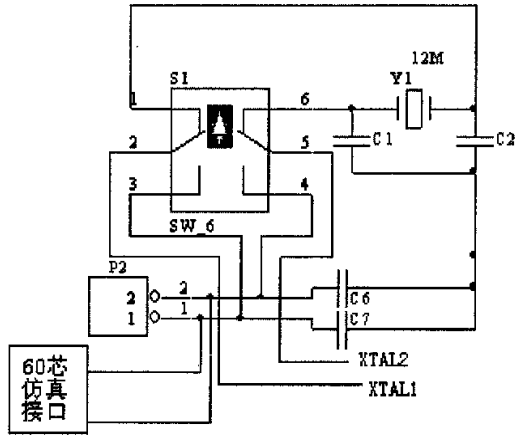


图 4.4 晶振电路原理图

8097BH 仿真扩展板还选用了两个分别为 40 芯和 60 芯的仿真接口插座。40 芯插座是用来与仿真器的主模块板的仿真扩展口通讯的。60 芯插座是通过一个电缆线与仿真插头连接的，仿真插头是在仿真用户板时将用户板中的 CPU 拔出，将仿真插头插入用户板的 CPU 插座中。96/98 BH 仿真扩展板与 MCS-96 的用户板之间是通过这个仿真接口电路连接，其中在 \overline{WR} 、 \overline{RD} 、ALE 均加有驱动，他们可驱动 8 个 LSTTL 负载， \overline{BHE} 负载能力比原来减低，其它线均与 8097 相同。

扩展板通过 60 芯电缆与用户系统相连，其中包括：8097 的 P0、P1、P2、P3、P4 五个 8 位口，HSI 与 HSO8 根线，BUSWIDTH (TEST)、 \overline{BHE} 、ALE、 \overline{WR} 、CLKOUT、VSS、READY 及 VREF、AGND、 \overline{RESET} 、INST，一共 60 根线。 \overline{EA} 已在内部接地，Vcc、Vpp 已在内部接+5v，NMI 由仿真器使用 V_{BB} 已在内部通过 0.01 μ F 电容接到 AGND，这些均没有接到用户系统。READY 线在 EM-96 内通过 5K 电阻接到+5V，不需延迟时可不接，需延迟时，可通过 READY 延迟，但时间不得超过 Intel 手册的规定。

仿真扩展板中其余的大部分器件就是用来作为地址锁存器或实现数据缓冲的数据总线缓冲器。另外扩展板中用到的每一个芯片在它的电源和地之间都并联一个电容，它们的作用来滤除电路干扰的。板上还有一个发光二极管，作为电路板上 5V 电源指示灯，扩展板上的电源是一个 5V 电源它是由主模块板提供的。

4.3 96/98BH 仿真扩展板的工作原理介绍

结合图 4.1 可知 EM-96/98BH 工作过程如下：两片 EPROM 存储器 U19 和 U26 都是监控程序存储器，共 16K 字节。它们的地址线是通过两片地址锁存器连到 CPU 的总线 AD1~AD13 上，U19 的数据线分别经两片 74LS245 连到 CPU 的 AD0~AD15 的数据线上，且分别由各自的 74LS245 的 DIR 信号控制数据的传输方向。U26 的数据线直接和 CPU 的低 8 位 AD0~AD7 数据线相连。它们的输出允许信号 \overline{OE} 直接与 CPU 的 \overline{RD} 相连。两片 EPROM 存储器可动态工作在 8 位或 16 位总线方式。当工作在 8 位总线方式时，通过各自的片选工作信号来选通。当 AD0=1 时，U19 被选通工作。当 AD0=0 时，U26 被选通工作。当工作在 16 位总线工作方式时，两片 EPROM 存储器同时被选通工作。U19 的数据总线通过两片 245 连接到 CPU 的 16 位总线上，16 位工作方式时它是作为奇数地址存储单元进行工作的，即接数据线 D15~D8。8 位总线工作方式时，它可作为一个独立的 8 位的存储器工作。

8K 字节的 6264 静态 RAM 存储器 U8 是断点存储器，最多可以设置多达 8K 个程序断点、数据读写断点。断点设置是通过 8255 的 PB 口来完成的。U8 的 A0~A7 的 8 位地址线经地址锁存器与 J1 端口相连，A8~A12 直接和 8255 的 PB 口相连。U8 的数据线有三颗连到 8255 的 PC 口，还有三颗连到 J1 端口上，其余两颗悬空。这样用户通过主控板和 8255 的接口就可完成断点设置。六颗数据线中有三颗是用来设置断点地址的，所设置的断点值应该大于等于 100H，因为 0~0FF 区域内为 MCS-96 的特殊功能寄存器和内部 RAM 区。另外三颗数据线是用来设置断点种类的，分程序地址断点、数据写地址断点、数据读地址断点、数据读/写地址断点。设置好的断点就存在这个 8K 字节的 RAM 中。当主模块板向 8097BH 发出命令通知它运行用户程序时，它会将 U8 的输出置为允许状态。这时 PC 中送出的地址值不仅选通仿真 RAM 中的某一个存储单元，同时选中相应的断点地址寄存器中的相应单元，断点地址寄存器的相应单元中数据输出送到一个异或门与 PC 中送出的高三位地址值在异或门中进行异或，从而实现断点地址的比较功能。异或门的输出去可编程器件 U7 的输入端，U7 的输出端 17 引脚输出信号去触发 8097 BH 的 NMI 引脚，产生非屏蔽中断。在中断

程序中，用户程序的现场得到保护，用户的一些特殊功能寄存器的状态送到一个预先所设置的缓冲区中，也可在终端或计算机中显示出来供用户查询。中断返回后进入监控状态，等待用户发出新的命令。这就是硬件断点的实现过程。它的好处是可在用户的程序存储器中任意设置断点，而软件断点不能实现这个功能。另外，在设置断点地址值时，在8K地址区域内，断点可为任意值，但大于8K地址时，两个断点的低13位不能重复。

仿真扩展板中的U22也就是8255接口芯片起着非常重要的作用。主控板通过8255的PA口来实现和8097BH的双向通讯。因为只有A口允许在方式2下工作，即可以进行双向数据传输。A口的8颗数据线与8097BH的低8位地址数据线相连，也就是主模块板通过8255接口A将8097BH作为自己的外设部分，8255的操作是受主模块板的控制。8255的8颗数据线直接与端口J1的8个引脚相连。这样就可以实现双向数据通讯了。8255的读/写控制信号和片选信号 \overline{CS} 与单片机仿真器的主控板相连。A0和A1经过地址锁存器和主模块板相连。即8255内部端口的口地址是由主模块板进行译码选择的。复位信号RESET接至复位电路中。复位电路是在电源信号线VCC和地信号线GND之间连了一个 $10\mu\text{F}$ 电容和一个510欧姆的电阻。上电后，复位电路开始工作。用户下达的命令由主模块板接收后经过解释后，通过8255A口通知8097BH，从而实现它们之间的双向通讯。

U21是32K字节的EPROM存储器，它是上电自动工作的。它的低8位地址线经地址锁存器和主模块板相连。数据线直接连至主模块上。它是控制软件存储器，共32K字节。里面装有MCS-96的汇编程序FD-ASM96。

U20和U27是两片32K字节的静态RAM，两片联合作为64K仿真RAM使用。它们的地址线A0~A14经过地址锁存器和8097BH的AD1~AD15相连。它们的读/写控制信号线和片选信号线是由GAL和主模块板的信号经译码后选通工作的。

两片74LS245（U15和U23）在这里是用来驱动数据地址总线AD0~AD7和AD8~AD15以提高仿真插座的带载能力。两片245的使能端脚19均直接接地，以保证时序不被后延而影响可靠性，而245的方向控制端脚1受一个或门控制，保证数据的传输不发生冲突。

4.4 不可屏蔽中断的使用和全速硬件断点的实现

仿真扩展板采用MCS-96的不可屏蔽中断作为控制手段，使用MCS-96的0~0FFH的程序存储区作监控存储器区，故用户不能使用NMI及0~0FFH的程序存储区（用户的0~0FFH为MCS-96的特殊功能寄存器和内部RAM区）。以上这两点是Intel专为MCS-96的开发系统保留的功能，它们只能由开发系统所使用。

所谓硬断点，就是当程序执行到断点时，靠硬件电路打断用户程序，返回到监控程序中，简单的方法就是利用中断来实现。因为MCS-96有一个非屏蔽中断可以利用它来实现。单片机先将设定的断点地址保存在断点地址锁存器中作为比较电路的一部分输入，比较电路的另一部分输入直接接至地址总线。这个比较电路是由一个四二输入异或门来实现的。

96/98 BH仿真扩展板中装有8K硬件断点寄存器，可由断点写入命令或全速运行命令写入程序地址或数据地址断点。在运行用户程序时，由硬件电路把用户8097的16位地址总线的内容与硬件断点寄存器的内容实时进行比较，如果两者符合，并且条件也符合，如对程序断点则为在8097读指令时地址符合；对数据读断点，则为在8097读数据时地址符合；对数据写断点，则为在8097写数据时地址符合；对数据读写断点，则为在8097读或写数据时地址符合，发出控制信号，使用户8097停止运行。在中断程序中，监控程序将保存用户程序执行地址和状态标志等信息，并将这些信息发送给系统机，然后监控程序将恢复断点处的原用户程序指令，最后跳转到监控程序的主等待程序，等待系统主机发出新的命令，转回到监控状态，并显示这时用户8097的现场。采用这种方法，可实现全速仿真，并具有如下几个优点：

- 1.具有数据读写断点功能，可监视CPU对某些存储器或I/O设备的读写。
- 2.具有区域断点功能，可方便程序的调试。
- 3.不影响用户程序运行，还可调试用户板上EPROM中的程序。
- 4.允许用户任意使用TRAP指令和软中断功能。
- 5.具有夭折处理功能，在遇不到断点时，键入回车可停止用户程序执行，并显示当前CPU现场。

在使用全速硬件断点时，必须注意它是采用监控用户8097地址总线的方

法来进行控制。而MCS-96单片机具有先行指令栈，即它在执行本条指令时，可再取进多达三个字节的后断指令。使用这种技术，加快了MCS-96执行指令的速度，但也给硬件断点的控制带来一定的麻烦。因为MCS-96在执行一条指令时，可能已经取进了后断指令，如果硬件断点设在后断指令地址上，则该预取指令操作将触发硬件断点控制电路，使程序停止运行。如果本条指令为非转移指令（断点前一条指令）。则停止时用户PC正好等于断点地址。但是如果该指令为转移指令（包括无条件转移、条件转移和子程序调用、返回）。则停止时用户PC值等于转移地址，而不等于断点地址，即这时停止地址不符合断点控制的要求。

4.5 MCS-96 的仿真特性及通讯电路的实现

在单片机仿真器主模块板仿真接口上外接 96/98 BH 仿真板，就可以开发 8094-90、8095-90、8096-90、8097-90、8094BH、8095BH、8096BH、8097BH、8098 等 MCS-96 系列单片机应用系统。

在联机仿真时，96/98BH 仿真板能将用户样机所需要的 MCS-96 单片机完整地出借给用户系统，不占用用户系统 MCS-96 的任何资源；能以各种方式调试 MCS-96 的各种中断和各种特殊的 I/O 或定时器功能；能控制用户系统的扩展总线以 8 位或 16 位的数据传送方式工作，96/98BH 仿真板共有 64K 仿真 RAM；能出借给用户作为程序存储器或数据存储器；能控制用户系统以单拍、跟踪、全速断点或连续方式运行仿真 RAM 或样机 EPROM 内的程序，允许用户设置多达 8K 个程序地址断点（取指令操作周期）或数据地址断点（读写数据存储器周期），并且也具有分析功能，能使用户系统停止执行程序，并读出用户系统资源状态，分析程序运行的结果。

本系统采用的是主从串行通信方式，其特点是结构简单方便，通信距离远。PC 机通常有一个 25 针和 9 针的 D 型异步通信接口，它采用 RS232 电平，可提供通信所需的控制信号和数据信号，因而在与单片机仿真器连接时，十分方便，只需将其发送端 TXD 和接收端 RXD 与单片机的数据接收端 RXD 和数据发送端 TXD 通过电平转换芯片 1488，1489 分别相连即可，如图 4.5 所示。

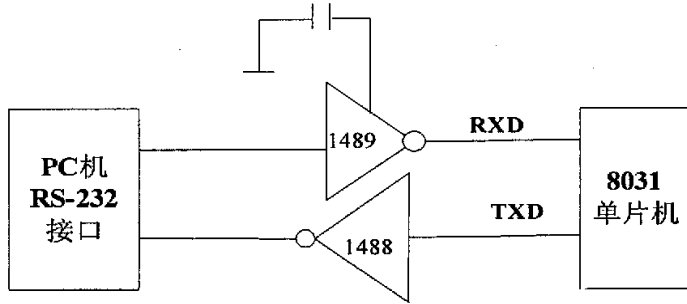


图 4.5 单片机与 PC 机串行通讯硬件连接原理图

4.6 本章小结

本章从 96/98 BH 仿真扩展板的硬件结构框图出发，简单介绍了它的工作过程。同时，对扩展板中所用到的各主要元器件的选型及功能分别进行了详细的说明。接着结合所选用的元器件阐述了仿真板的工作原理，说明了仿真板中硬件断点的实现及它所具有的优点，在本章的最后总结了 MCS-96 的仿真特性及通讯电路的实现。

第 5 章 仿真软件设计

单片机仿真器内具有各种仿真软件，其中包括处理寄存器及存储器单元的显示和修改、目标代码程序的单步运行、断点运行和 EPROM 编程等功能，每一个命令处理都对应着一个子程序段，并把子程序的入口地址集中存放在命令入口地址表中，以便监控程序根据命令进行查表，转移到相应的命令处理子程序去执行。

用户可通过终端键盘输入各种命令，并在屏幕上显示执行结果，可使用仿真器上外接的打印机打印出程序清单；可将源程序写入外存设备和用 EPROM 读出/写入器将目标码写入 EPROM。将通用单片机仿真器与任何具有 RS-232 接口的计算机相连，在主机的通信控制软件支持下，利用主机的硬件和软件资源可构成一个高性能的通用单片微机开发系统。这时，单片机仿真器主要靠 PC 机显示屏提供用户界面。用户通过 PC 机向仿真器装载代码文件，发出各种操作命令。仿真器向 PC 机提供各种状态信息，送回要贮存的文件等。PC 机与仿真器之间通过 RS-232 口进行双向串行通信。因此整个仿真系统的软件包括两部分，一部分是 PC 机中的通讯控制软件，另一部分是仿真器板的监控软件。

5.1 监控程序设计

监控软件界于使用者和仿真器之间，用户通过监控软件的协调管理才能使用仿真器。为了使监控软件具有较大的灵活性和灵活能力，监控软件使用了模块化接口法，将仿真器的所有功能划分为若干个具有独立功能的部分，使程序显得简单，清晰，且程序易于调试和修改。

仿真器的各种操作命令共有几十条，其中包括系统命令、读出命令、读出修改命令、传送命令、运行命令、符号化调试命令、编辑命令。其软件框图如图 5.1 所示。使用这些命令可输入、编辑、汇编源程序，并可汇编生成目标程序，然后可用单拍、跟踪、断点、全速等各种运行控制命令，对应用

系统进行调试。这时可读出或修改应用系统的 CPU 内部各种状态和寄存器的内容，也可读出或修改应用系统的程序存储器或数据寄存器的内容。这些操作命令具有使用方便、功能强的特点。监控软件除了可完成一般仿真功能外，还具有汇编程序，可对标准的格式的汇编源程序进行汇编，具有符号化调试功能，能直接用汇编中的标号作为调试时的启动或断点地址。另外，还能用特殊功能寄存器的名字来读出或修改它们的内容。这种符号化调试功能是微机开发系统所特有的，用它可实现在汇编语言的基础上调试用户程序，大大方便了程序调试，加快了开发进程；除此之外，它还有反汇编功能，可反汇编显示执行的命令，也可反汇编程序寄存器的内容，另外，它还具有 16 进制文件读入和输出功能，可直接接收或保存目标程序。

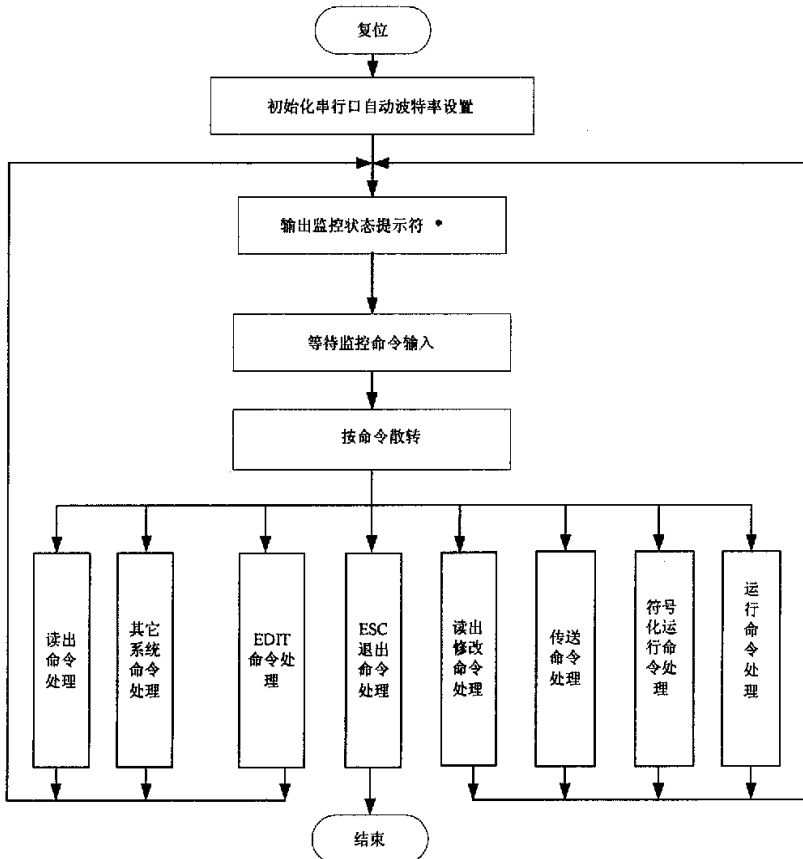


图 5.1 监控程序流程图

5.2 编辑命令

我们从图 5.2 中可以看出，在编辑状态下，可输入并执行各种编辑和控制命令，也可进行汇编。编辑命令作为监控状态下的一个子程序。

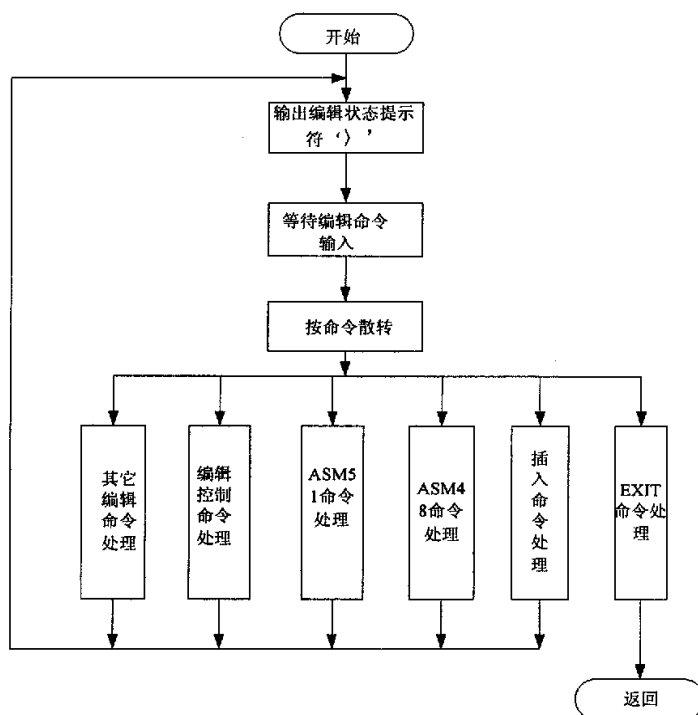


图 5.2 EDIT 命令处理流程图

复位后，仿真器进入监控状态，在键入转向编辑命令 EDIT 或 EDITA 后，从监控状态转到编辑状态，这时会显示编辑状态提示符“>”。仿真器的编辑程序为行编辑程序，它的编辑命令的格式和使用方法与 IBM—PC 的 EDLIN 行编辑程序基本相同，具有插入命令 I，列表显示命令 L、打印命令 P、行修改命令 N、删除命令 D、字符串搜索命令 S、字符串替换命令等。在利用编辑程序输入完源程序后，可键入汇编命令 ASM51 对源程序进行汇编。在完成了编辑和汇编后，键入 EXIT 命令即可退出编辑状态返回到系统状态（提示符由“>”改为“*”）。以后，如果想再次进入编辑状态对已输入的源程

序进行修改,则必须键入 EDITA 命令,待出现提示符“>”后(表示已进入编辑状态)便可对上次已输入的源程序进行修改、列表显示、汇编等操作。在编辑状态,不能使用监控状态的一切命令,只能使用编辑命令和汇编命令。只有 EXIT 命令,才使它再回到监控状态。

5.2.1 编辑控制命令

仿真器上配有锂电池,可长期保存机内 RAM(包括源程序存储器)的内容。如果把仿真器与带有磁盘的个人计算机如 IBM-PC 等相连,同时配上这些个人计算机用的通讯控制程序,则可以使用它们的磁盘作为 FD-EDIT 的源程序外存设备。这时可使用 MSV、MLD、MLA 等命令来读写源程序。这些命令为主机磁盘操作命令。在使用这些命令时应该注意的一点是它们不是以回车符作为结束标志的,而是以 ©V 或 ©W 结束的,在连接 IBM-PC 并使用 TERM 主模块板通讯软件时,键入命令后,主机要求输入源程序文件名,文件名的格式必须符合主机操作系统的要求,键入文件名后,这时就可按照命令要求完成相应的操作。

除了使用主机磁盘作为源程序的外存设备外,还可以使用快速外存板或 E²PROM 外存板作为源程序的外存设备。E²PROM 是由可改写的可编程序的 ROM 器件构成的存储器件,它在写入后,数据可一直不变,又可随时对任何一个单元进行改写,每个存储单元容许写入 10000 次以上,写入后数据可保存十年以上。E²PROM 外存板上有一个写入允许开关。平时它应处于禁止写入的位置,以防止误操作破坏存储的数据。在进行源程序存外存板前,应首先打开写入允许开关,然后再键入 ESV 命令,此命令就可以把主模块板源程序存储器中的内容写入到外接的 E²PROM 外存板上。写入时,约 20ms 写入一个字符。写入完毕,出现编辑状态提示符 ‘)’,这时应及时关闭 E²PROM 写入允许开关。也可以使用命令将 E²PROM 外存板上的内容读入源程序存储器区中。

快速外存板是通用单片机仿真器的一种专门用于存放源程序的外存设备。它的读出、写入速度极快,远快于 E²PROM 外存板;它的速度甚至比主机磁盘操作还要快得多。它的使用也比较方便,是一种最佳的外存设备。它

的板上也具有一个写入允许开关，快速外存板在使用前，必须使用命令进行初始化。此外，它还有目录显示命令，删除命令，源程序写入、读出命令。

5.2.2 其他编辑控制命令

1.源程序存储区大小检测命令：CK

显示源程序存储的下一个空闲单元地址。

主模块板中，源程序存储区的首地址为 0002H，末地址取决于所需的用户目标程序存储区（出借 RAM）大小，一般为 0DFFFH，（如果用户目标程序存储区大于 32K，则源程序存储区末地址为 7FFFFH）。所以，一般源程序存储区容量为 56K。

2.源程序存储区清除命令：CLR

它通过清 0 源程序行数指针和源程序存储区末地址指针，达到清除源程序存储区的目的。

3.退回监控状态命令：EXIT

执行该命令，保存现在的源程序指针，然后从编辑状态退回到监控状态，这时提示符变为 ‘*’。以后可键入监控命令。

4.后备文件命令：BAKUP

主模块板在需要仿真 RAM 小于 32k 及不借用数据存储器时，可具有 56K 源程序存储区。如果用户的源程序小于 28K，则可在主模块板中存放两个源程序文件。在已输入一个源程序后，键入 BAKUP 可把源程序存放到从 7000H~DFFFH 的区域中。

5.后备文件变换命令：CHANG

在用 BAKUP 命令存放过一个源程序且以后又输入过另一个源程序后，如果想重新使用以前的源程序，键入 CHANG 后可把 7000H~0DFFFH 的源程序与 0000~6FFFH 的源程序区进行交换。这样新的源程序变成后备文件，而原来的后备文件变成现行源程序。

由于仿真器的源程序存储区只能存放一个源程序，故它的编辑和汇编都没有文件控制功能（但是和主机如 IBM—PC 或 APPLE—II 相连时，可使用主机的文件系统）。汇编时，采用二次扫描方法，第一次生成用户符号表，并

进行语法检查，第二次生成机器码。汇编时生成的符号表保留在系统 RAM 中，供符号化调试命令使用。如果使用了宏子程序调用指令，则先登记所有调用的子程序，然后在第二次扫描后，生成所调用的各种子程序的机器码。它的具体实现方法类似于一般高级语言的 LINK 程序，即在把子程序装配于用户目标程序后面时，需同时按实际地址修改所有的调用指令的地址。

5.3 运行控制命令

调试用户程序对于单片机仿真器是必须具备的也是最重要的功能，它包括连续运行、断点运行和单步运行，用户在调试过程中经常用到这些功能，但实现起来比较复杂。

1、连续运行

连续运行时监控软件将控制权让出，完全交给用户程序，不再进行干预，直到强制复位中止用户程序的运行。这时相当于用户程序独立工作，实现其预定的各种功能。要实现这一功能，监控软件需要完成的任务主要是恢复用户现场和利用转移指令去执行用户程序。

2、单步运行

简单地说，单步运行是指一条一条地运行用户程序，每执行一条用户程序都要返回监控状态等待下一步操作，这时用户可以检查各种数据、寄存器状态等来判断用户程序运行是否正常。单步也允许用户在执行下一条指令之前跟随程序逻辑每次一步的检查所有过程信息。通常，在调试过程检查数据总线、地址总线、状态标志及控制总线的状态是很有帮助的。在每一步期间，借用检查过程状态，调试者就能查找出这些程序故障，如非正确的跳转，非正确的寻址和错误的操作码等。图 5.3 是单步运行流程图。

当利用单步进行硬件调试时，时序错误，非正确的中断及 DMA (Direct Memory Access) 错误是检查不出来的。由于单步方式对目标处理的每一条指令要做大量的处理工作，因此单步比起实时程序的执行速度要慢很多。由于这个原因，单步调试通常只限于小段程序的调试，而整个程序就不一定要用单步调试。单步运行是一种重要的调试方法，可以有效地排除用户系统的软硬件故障。

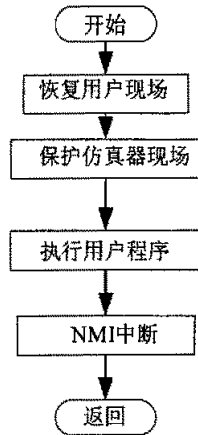


图 5.3 单步运行流程图

3.跟踪运行

在设置了启动地址和断点(停止)地址及循环次数后开始执行命令，是从启动地址开始以单拍方式连续执行用户程序，直到遇到断点地址且其断点次数符合或人工中止命令的执行为止。在跟踪执行时，可用暂停命令停止执行，以后可键入任何键来继续执行命令。跟踪时，执行速度比较慢，并与终端或主机的波特率有关。在需执行一段较长的程序时，一般应采用断点运行方式。

96/98BH 真扩展板的单拍和跟踪调试采用 MCS-96 的不可屏蔽中断 NMI 实现（该中断是 INTEL 开发系统专用的，不能由用户系统所使用）。在使用时，不能以单拍运行方式响应中断。

4.断点运行

单步调试允许用户每执行完一条指令之后，就停止程序执行。但是，应当指出，这种方式是很慢的，在需要全速执行程序时是不使用这种方法的。最好是全速执行程序，然后在要检查的特殊指令处暂停程序。断点运行是指连续运行用户程序到某一断点地址处返回监控程序。这时用户可以查询中间结果和寄存器状态，借以判断程序运行是否正常。断点程序就是允许用户插入一条称为断点的专门指令，以便在用户希望程序暂停的地方暂停。当执行断点指令时，控制将返回监控程序而用户就可以检查过程状态，观察程序是如何操作的。

通常有二种断点系统，即硬件断点和软件断点。硬件断点使用硬件监控

系统地址总线并在执行程序时检查是否到达给定的断点。当发现断点时，硬件使用处理器控制总线暂时中止处理器以便检查，或使处理器执行中断而进入断点程序，产生非屏蔽中断，从而返回监控状态，进行断点后的处理，如保护用户现场等操作。硬件断点能用来调试以 ROM 和 RAM 为基础的程序。

所谓软件断点是指断点功能完全由软件完成，通过修改用户代码实现断点，一般是用一条程序转移指令替换断点处的用户指令，这样当用户程序执行到断点地址时，就不再执行用户的原指令，而是跳转到监控程序的断点处理程序，从而完成断点用户功能。不过当返回监控程序后，仍需将被修改的内容加以恢复，故监控软件需要作相应处理。本系统是采用在断点处的用户程序中插入一条 TRAP 指令 (0F7H) 的方法来实现断点控制，且断点地址不能出错，应为一条指令的开始地址，将 TRAP 指令操作码代替断点处指令的操作码，并把被代替指令的操作码保存在系统堆栈中，以后每当执行到指定的断点处时，处理器就会自动响应软中断，实现对运行状态和结果的检查，最后把堆栈中保存的原指令操作码取出放回原处，等待用户键入“继续执行”键后就往下运行。单片机仿真器的软件断点运行命令，具有夭折处理功能，在遇不到断点时，可键入回车，停止用户程序的运行，并将如同遇到断点一样，恢复断点处的用户指令和用户软件中断向量，然后显示断点处的 CPU 状态，包括 PC 值。软件断点运行方式具有断点位置准确的特点，但它只能用于设置程序地址断点，并且只能用于仿真 RAM，因为断点指令必须插进要执行的程序中。不能设置数据读写地址断点，也不能用于用户 EPROM 作程序存储器的场合。由于软件断点需要修改用户程序，如果断点设置不当，程序可能出现崩溃，这个为软件断点的不足之处。

断点对调试程序是极为有用的，因为它允许用户在停止执行之前执行大部分程序。这样可以执行整个分程序，然后暂停以检查处理器寄存器的状态看是否符合执行的分程序。当断点不需要保留长时间，就把它移去再执行正常的程序。

断点运行的流程图如图 5.4 所示。监控程序在收到命令后执行图(a)所示的断点处理和用户现场的恢复，然后执行用户程序到断点处。在断点处执行图(b)所示的流程，即返回监控状态。

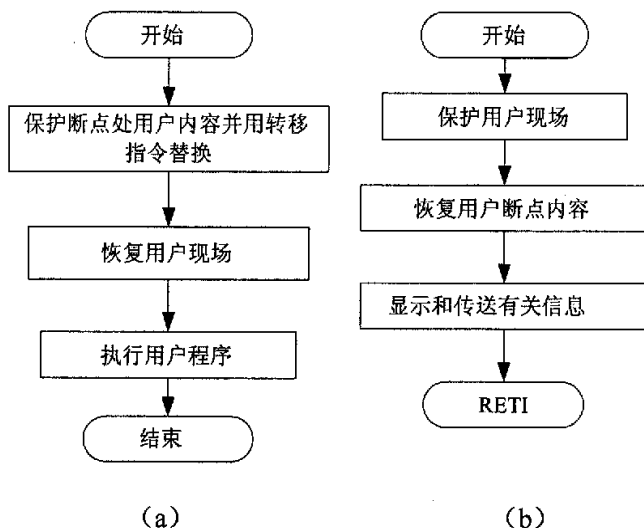


图 5.4 断点程序流程图

大部分调试程序都是由单步和断点程序组合而成，用户可以在要求的位置上插入断点并让程序一直执行到断点处为止。当程序停在断点处，用户可以使用单步方式每次一条指令一条指令地检查程序。

5.4 符号化调试命令

通用单片机仿真器具有全符号化调试功能，允许使用汇编程序中的标号作为启动地址和断点地址，也允许用数据符号作为数据断点地址。它们的功能和执行方式与运行控制命令相同，只是现在参数不再是 16 进制数，而是符号或数据，命令中的参数可以是用户汇编程序的标号、十进制数或以数字开头 ‘H’ 结尾的 16 进制数，也可以为 \$（它代表当前 PC 的值），并且还可是标号 \$ 再加上数字 M，该数字 M 为 1~255 之间，它代表在该标号或 PC 值后 M 条指令的地址。另外，在设置数据地址断点时，还可使用汇编源程序中数据地址符号，符号化调试命令包括单拍运行命令，跟踪运行命令，全速软件断点运行命令，硬件断点设置命令，区域断点设置命令，全速硬件断点运行命令。符号化调试命令，必须是用仿真器配置的汇编程序进行汇编后才能使用。如果使用主机的交叉汇编，则必须在交叉汇编后，得到汇编的用户符

号表，用命令装入用户的目标程序后，再用相应的命令接收用户符号表。这样以后再执行符号化调试命令时，它将按仿真器中的符号表进行查表，得出相应的 16 进制数地址。符号化调试命令与非符号化调试命令可任意组合使用。在使用符号化调试命令时，必须注意该类命令至少应该有一个参数，如不需参数，则应该用对应的非符号化调试命令。在符号化调试命令中使用数字地址时，必须注意数据的格式：十进制数或以数字开头以 ‘H’ 结尾的 16 进制数。

5.5 汇编命令

通用单片机仿真器具有三条汇编命令，分别是 ASM51、ASM48 和 ASM96。在编辑状态下键入汇编命令后，对源程序存储器区中的汇编程序进行汇编。它们的汇编采用二次扫描方法，第一次生成用户符号表，并进行语法检查，第二次生成机器码。汇编时生成的符号表保留在系统 RAM 中，供符号化调试命令使用。如果使用了宏子程序调用命令，则先登记所有调用的子程序，然后在第二次扫描后，生成所调用的各种子程序的机器码。汇编后的目标程序存放在由 ORG 指令所指出的仿真 RAM 单元中。ASM48 和 ASM96 的使用方法相同。

我们以 ASM96 为例，在把 MCS—96 汇编程序存放主模块板的源程序存储器区后，就可使用 MCS—96 汇编命令，把它变成机器码。使用方法如下：先在 MCS—96 仿真状态下进入 FD—EDIT 编辑状态，然后输入 MCS—96 汇编源程序。最后在编辑状态下，键入) ASM96，即可开始对 MCS—96 汇编源程序进行汇编。汇编后的目标程序存放于由 ORG 指令所指出的 96/98BH 的仿真 RAM 单元中。然后开始对源程序进行第一遍扫描。在这遍扫描中，生成用户符号表，并对程序进行语法检查。第一遍扫描中，如有错误，则显示出错信息，扫描完，显示出错数目，然后返回编辑状态。这时可对源程序进行修改，如没有错误，则询问是否需要列表显示汇编结果，是否需要打印汇编结果。

接着开始第二遍扫描。如回答需要列表显示汇编结果的，则一面汇编，一面在显示器上逐行显示汇编的结果。如不要求显示，则只进行第二遍扫描。

在第二遍扫描中逐行生成机器码,放入由 ORG 命令所指示的 RAM 单元中。在第二遍扫描完成后,如有错误,则显示出错内容,然后回到编辑状态。如没有错误,则显示用户符号表,然后显示下一个空闲地址,再返回编辑状态,如果 ORG 指定的地址不在 96/98 BH 仿真扩展板的仿真 RAM 中,则可用 ASM96n 来进行汇编, n 为十进制数或以数字开头 H 结尾的 16 进制数,这时目标程序存放于由 n 指出的地址开始的 96/98 BH 仿真扩展板的仿真 RAM 中,而它的转向地址则仍取决于 ORG,与存放地址无关。

5.6 本章小结

本章介绍了通用单片机仿真器仿真软件的功能,它具有监控和编辑两个主要状态。给出了监控软件的程序流程框图,并结合流程图逐一介绍了它的编辑状态下的几条命令,对运行控制命令中单步和断点功能的实现也给出了程序流程,特别对于断点功能的具体实现以及各实现方法的优缺点都一一作了介绍。最后给出了符号化调试命令和汇编命令的功能。

结 论

随着半导体集成电路制造工艺的不断发展和电子技术应用领域的不断扩展,单片机以其性能好、体积小、价格优、功能全、功耗低等突出优点被广泛应用。单片机已成为电子信息领域增长速度最快的产品之一。单片机的开发离不开开发系统,在线仿真器作为开发系统最重要的组成部分,是目前单片机开发的主要设备。能否提供廉价的在线仿真器,较之能否提供高性能、低价位的单片机有着同等的重要性。

本文在阅读大量在线仿真器与开发系统相关文献的基础上,对在线仿真器的原理进行了深入的分析 and 研究,结合实验室的条件本论文完成了以下工作并得出如下结论:

(1)根据在线仿真器的基本仿真功能要求,并结合“96 系列单片机内部结构和指令系统的特点,提出了在线仿真器的整体设计构想,并且完成了仿真扩展板的硬件电路原理图的设计和印刷电路板的设计。

(2) 根据在线仿真器的基本仿真功能要求,完成了硬件断点控制电路的设计以及存储器 8 种地址映像方式的设计。

(3) 根据在线仿真器的基本仿真功能要求,完成了主模块板和仿真扩展板之间的双向数据通讯的设计。

(4)结合硬件电路,给出了仿真软件的设计。

(5)通过试验,本仿真器能够实现仿真时所需要的各种功能。

由于是首次进行在线仿真器的研究与设计,加之时间和个人能力的限制,对许多问题的研究还处于探索阶段。因此本文所设计的仿真器还存在着很多不足之处。如本文所设计的仿真器是在 DOS 操作系统下工作的,因此使用起来很不方便,硬件电路部分的体积也很大,软件部分也是采用传统的汇编语言编写的,不仅编程工作量,而且不利于移植。所以今后需要做以下几方面工作:

1. 将仿真器的工作界面改为在 Windows 操作系统下工作。
- 2.提高在线仿真器与 PC 主机之间通讯的可靠性和速度。在线仿真器 PC

主机之间通过 RS-232 串行通讯端口进行通讯，采用 RS-232 串行通讯端口的优点是通讯协议比较简单，编程容易，但是通讯速度受到了限制。可以考虑采用通用串行总线(Universal Serial Bus } USB)进行通讯，USB 通讯速度快，采用差分传输方式，具备检错和纠错功能，保证了数据的正确传输，并且用户安装方便。

3.将原来采用的汇编语言改为现在通用的结构化编程语言 C 语言。

4.采用 FPGA 来实现仿真控制电路，这样有利于实现在线仿真器的系统集成度和可靠性，使硬件结构简单，同时降低了设计的风险和成本，缩短了设计的周期。由于 FPGA 具有反复可擦写的特点，当要根据实际需要完善在线仿真器功能时，只需要修改仿真控制逻辑，然后将修改后生成的位流文写入 FPGA 芯片即可，因此可以方便的对在线仿真器进行升级换代。

参考文献

- [1] 高金山. 6502MCU 在线仿真器的研制. 湖南大学硕士学位论文. 2004, 1
- [2] 涂时亮, 张友德等. Super Sice 通用单片机仿真器及其应用. 复旦大学出版社, 1995
- [3] 孙涵芳. Intel 16 位单片机. 北京: 北京航空航天大学出版社, 1995
- [4] Inoue, Y. SOI technology for future SoC. Junction Technology, 2002. I WJT. Extended Abstracts of the Third International Workshop, 2002, 107-110P
- [5] 居水荣. 单片机开发系统综述. 半导体情报. 2002(1):19-22 页
- [6] 周永钊, 张雷, 陈铭. 通用逻辑阵列 GAL . 中国科学技术大学出版社, 1989
- [7] 薛青, 张华, 胡蕾, 杨广学. 单片机原理及接口技术 (MCS8096 系列). 哈尔滨工业大学出版社, 1998
- [8] 张毅刚, 彭喜源, 谭晓明. MCS-51 单片机应用设计. 哈尔滨工业大学出版社, 1997
- [9] 赵长德. MCS-51/98 单片机原理与应用. 机械工业出版社, 1997
- [10] 张迎新. 单片微型计算机原理, 应用及接口技术. 北京: 国防工业出版社, 1993
- [11] 刘振安等. 单片微机开发系统: 设计原理·软件仿真·在线仿真. 合肥: 中国科学技术大学出版社, 1994
- [12] 周杏鹏, 路兆梅. 8098 单片机开发器的设计与实现. 自动化仪表. 1995(9):35-38 页
- [13] 王东云. DSG-51 单片机仿真器硬件结构及监控程序分析. 郑州纺织工学院学报. 1991(4): 73-78 页
- [14] 郭家伟, 栾贵华, 龚刚华. KSJ-Z80 仿真器的设计及应用. 小型微型计算机系统. 1986(9):16-27 页
- [15] 余家春 . Protel 99 SE 电路设计实用教程. 北京: 中国铁道出版社, 2004
- [16] 张平. 单片机开发的一种简易方法. 上海第二工业大学学报. 1988 (2): 87-93 页

- [17] 张增秋. 微机开发系统的选择及应用. 电子产品世界. 1994 (1): 37 -38 页
- [18] 杨瑛洁. 微机与单片机实验平台的设计与开发. 电子科技大学硕士学位论文. 2004, 3
- [19] 雷自强. 微型计算机开发系统. 江苏: 江苏科学技术出版社, 1990
- [20] 修林成. MCS-51 单片机开发系统中的断点产生. 微型机与应用. 1993 (9): 4-5 页
- [21] 李朝阳. 单片机开发系统硬件断点的设计与实现. 微小型计算机开发与应用. 1995 (3): 15-20 页
- [22] 黄均鼎, 俞承芳, 蒋慧文, 卢烜. 可编程逻辑器件设计. 上海: 复旦大学出版社, 1997
- [23] 孟京生. C8051F $\times\times\times$ 系列单片机仿真器的研制. 河北工业大学硕士学位论文
- [24] 阎石. 数字电子技术基础. 北京: 高等教育出版社, 1998
- [25] 戴梅萼, 史嘉权. 微型计算机技术及应用. 北京: 清华大学出版社, 1996
- [26] 于慧明. 一种新型联机仿真器的研究. 合肥工业大学学报. 1982 (3): 91-99 页
- [27] Howard M. Bloom, Cita M. Furlani, Anthony J. Barbera . Emulation as a desingn tool in the development of real-time control systems. Proceedings of the 16th conference on Winter simulation. 1984(1):626-636P
- [28] John H. Mayer. ICEs rise to challenge of new processors. Computer design. 1997(3):113-115P
- [29] Rgincan, Fahir; Saatci, Ali. STAND-ALONE-CIRCUITEMULATOR. Microprocessing and Microprogramming . 1986(3): 159-167P
- [30] Schwerte, J. General-purpose in-circuit emulator. Elektronik. 1982(19):65-68P
- [31] P. K Rowe, B. Paquirek. Remedy: A Real-Time Multiprocessor System Level Debugger. 1987 IEEE Real-Time System Symposium. 230-240P
- [32] Colins, R. R. In-circuit emulation. Dr. Dobb's Journal. 1997(7):113-115P

- [33] 吴宣恒. 利用 8097BH 单片机开发多功能专用机系统. 微处理机. 1992 (4): 11-12 页
- [34] 夏华龙. 在线仿真器技术和 AEDK5198 仿真器. 航空电子技术. 1992 (3) :39-48 页
- [35] 费一民, 王晓峰. MCS-96 系列单片机开发系统. 电子工程师. 1998 (1) :12-16 页
- [36] 尹雪飞, 陈克安. 集成电路速查大全. 西安电子科技大学出版社, 2000
- [37] 张宇. 一种 8098 单片处理器的在线式仿真器. 电脑开发与应用. 1991 (3): 29-31 页
- [38] Ing-Jer Huang, Hsin-Ming Chen and Chuang-Fu Kao. Reusable embedded in-circuit emulator. Asia and South Pacific Design Automation Conference 2001. 2001, 33-34P
- [39] 张有志. 可编程逻辑器件 PLD 原理及应用. 中国铁道出版社, 1996
- [40] 微处理器开发系统性能评价. 电子产品世界. 1996 (7): 79-80 页
- [41] 方大寿, 李纪扣. 单片机开发系统的单步和断点功能的实现. 微计算机应用. 1995 (5): 14-19 页
- [42] 郭家伟, 栾贵兴, 龚刚华. KSJ-Z80 仿真器的设计及其应用. 小型微型计算机系统. 1986 (9): 16-27 页
- [43] 李华, 孙晓民, 李红青, 徐平, 张新宇. MCS-51 系列单片机实用接口技术. 北京: 北京航空航天大学出版社, 1993
- [44] 黄玉蓉. 单片机开发系统的设计与实现. 公安大学学报. 2001 (6): 31-33 页

致 谢

本文是在导师金鸿章教授的悉心指导下完成的，在跟随金老师学习的三年中，金老师严谨求实的科学态度、勤奋踏实的工作作风、高尚无私的人格品质都对我的人生道路产生了深远的影响。金老师不仅在学术上对我指导和帮助，而且在生活中也给予我热情的帮助。同时，教研室的各位老师都给予了我许多关心和指导，使我受益匪浅。在论文完成之际，谨向给予我指导和帮助的金老师表示衷心的感谢和崇高的敬意。

在攻读硕士学位期间，以及论文完成过程中，我还得到了实验室吉明副教授、杨生工程师以及实验室其他同学的帮助和指导。在此，一并向他们表示诚挚的谢意！

感谢我的家人和朋友对我的理解、支持与鼓励。

作者: [张子红](#)
学位授予单位: [哈尔滨工程大学](#)

参考文献(45条)

1. [参考文献](#)
2. [高金山](#) 6502MCU在线仿真器的研制[学位论文]硕士 2004
3. [涂时亮](#), [张友德](#) Super Sice通用单片机仿真器及其应用 1995
4. [孙涵芳](#) Intel 16位单片机 1995
5. [Inoue Y](#) SOI technology for future SoC. Junction Technology 2002
6. [居水荣](#) 单片机开发系统综述[期刊论文]-半导体情报 2001(1)
7. [周永钊](#), [张雷](#), [陈铭](#) 通用逻辑阵列GAL 1989
8. [薛青](#), [张华](#), [胡蕾](#), [杨广学](#) 单片机原理及接口技术(MCS8096系列) 1998
9. [张毅刚](#), [彭喜源](#), [谭晓明](#) MCS-51单片机应用设计 1997
10. [赵长德](#) MCS-51/98单片机原理与应用 1997
11. [张迎新](#) 单片微型计算机原理,应用及接口技术 1993
12. [刘振安](#) 单片微机开发系统:设计原理·软件仿真·在线仿真 1994
13. [周杏鹏](#), [路兆梅](#) 8098单片机开发器的设计与实现 1995(09)
14. [王东云](#) DSG-51单片机仿真器硬件结构及监控程序分析 1991(04)
15. [郭家伟](#), [栾贵华](#), [龚刚华](#) KSJ-Z80仿真器的设计及应用 1986(09)
16. [余家春](#) Protel 99 SE电路设计实用教程 2004
17. [张平](#) 单片机开发的一种简易方法 1988(02)
18. [张增秋](#) 微机开发系统的选择及应用 1994(01)
19. [杨瑛洁](#) 微机与单片机实验平台的设计与开发 2004
20. [雷自强](#) 微型计算机开发系统 1990
21. [修林成](#) MCS-51 单片机开发系统中的断点产生 1993(09)
22. [李朝阳](#) 单片机开发系统硬件断点的设计与实现 1995(03)
23. [黄均鼎](#), [俞承芳](#), [蒋慧文](#), [卢烜](#) 可编程逻辑器件设计 1997
24. [孟京生](#) C8051FXXX系列单片机仿真器的研制[学位论文]硕士
25. [阎石](#) 数字电子技术基础 1998
26. [戴梅萼](#), [史嘉权](#) 微型计算机技术及应用 1996
27. [于慧明](#) 一种新型联机仿真器的研究 1982(03)
28. [Howard M Bloom](#), [Cita M Furlani](#), [Anthony J Barbera](#) Emulation as a desingn tool in the development of real-time control systems 1984
29. [John H Mayer](#) ICEs rise to challenge of new processors 1997(03)
30. [Rgincan Fahir](#), [Saatci Ali](#) STAND-ALONE-CIRCUITEMULATOR 1986(03)
31. [Schwerte J](#) General-purpose in-circuit emulator 1982(19)
32. [P K Rowe](#), [B Paquirek](#) Remedy:A Real-Time Multiprocessor System Level Debugger 1987

33. [Colins R R In-circuit emulation](#) 1997(07)
34. [吴宣恒 利用 8097BH 单片机开发多功能专用机系统](#) 1992(04)
35. [夏华龙 在线仿真器技术和 AEDK5198 仿真器](#) 1992(03)
36. [费一民, 王晓峰 MCS-96系列单片机开发系统\[期刊论文\]-电子工程师](#) 1998(1)
37. [尹雪飞, 陈克安 集成电路速查大全](#) 2000
38. [张宇 一种 8098 单片处理器的在线式仿真器](#) 1991(03)
39. [Ing-Jer Huang, Hsin-Ming Chen, Chuang-Fu Kao Reusable embedded in-circuit emulator](#) 2001
40. [张有志 可编程逻辑器件PLD原理及应用](#) 1996
41. [微处理器开发系统性能评价](#) 1996(07)
42. [方大寿, 李纪扣 单片机开发系统的单步和断点功能的实现](#) 1995(05)
43. [郭家伟, 梁贵兴, 龚刚华 KSJ-Z80 仿真器的设计及其应用](#) 1986(09)
44. [李华, 孙晓民, 李红青, 徐平, 张新宇 MCS-51系列单片机实用接口技术](#) 1993
45. [黄玉蓉 单片机开发系统的设计与实现](#) 2001(06)

相似文献(10条)

1. 期刊论文 [麦范金, 蒋存波, 崔建明, 陆刚 单片机可视化通用开发系统模型研究与实现 -桂林工学院学报2003, 23\(4\)](#)
目前单片机开发系统的开发界面不友好,而且各种开发系统通常都是针对某一特定类型CPU的,从而制约了单片机的广泛应用与发展.在介绍单片机应用程序汇编语言源代码结构特点的基础上,通过对可视化、通用性的分析研究,给出一个汇编语言的单片机可视化通用开发系统的系统模型与软件的实现,并提供了8051系列CPU和8096系列CPU单片机应用程序的基本功能模块.
2. 期刊论文 [居水荣 单片机开发系统综述 -半导体情报2001, 38\(1\)](#)
简单介绍了单片机开发系统的概念、功能及分类;然后具体介绍单片机开发系统的构成,包括开发工具和开发过程以及单片机开发系统的选择.
3. 学位论文 [陈彬 TSC-51单片机实验开发系统的模拟与仿真](#) 2002
该论文以单片机的开发与应用为工程背景,探讨了目前国内外单片机的发展现状以及单片机模拟软件的现状,总结了单片机模拟与仿真的研究方法,全面分析了TSC-51单片机实验开发系统的结构和特点,运用面向对象的模拟仿真方法,经过合理的设计,确立了软件模型,实现了预期的各种功能.对比国内外相关的软件,该软件的优势在于有友好的人机界面,有强大的调试功能,有独特的实验开发平台界面,而且不需要占用单片机系统资源.该软件的安装程序的制作则采用当今软件工程产业最流行、最规范的InstallShield软件来完成.
4. 期刊论文 [郑丽群 谈8051单片机开发系统及开发过程 -中国新技术新产品2009\(4\)](#)
构建一个以单片机为核心的应用系统,从任务的提出到系统正式投入运行,称为单片机应用系统的开发.对目前最通用流行的单片机51系列,依照51单片机应用系统开发的基本步骤和思路,介绍单片机开发的基本知识和如何针对具体项目需求来设计解决方法及单片机一般开发过程.
5. 期刊论文 [章克明 8051单片机开发系统和开发过程 -科技咨询导报2006\(1\)](#)
构建一个以单片机为核心的应用系统,从任务的提出到系统正式投入运行,称为单片机应用系统的开发.对目前最通用流行的单片机51系列,依照51单片机应用系统开发的基本步骤和思路,介绍单片机开发的基本知识和如何针对一个具体的项目需求来设计解决方法及单片机的一般开发过程.
6. 期刊论文 [王迎旭, 梁慧林, 伍萍辉 基于总线的模块式单片机通用开发系统的设计 -半导体技术2002, 27\(6\)](#)
介绍了一种基于总线的模块式单片机通用开发系统的组成和设计方法,以及开发系统的应用.
7. 期刊论文 [邵剑龙, 何春, 卢诚, 杨嘉林, 金建辉, 方全忠 一种便携式普及型多功能8051系列单片机开发系统的设计 -计算机工程与应用2003, 39\(13\)](#)
目前在单片机应用系统开发中广泛使用仿真器或仿真软件,然而,由于常见仿真器不是价格较高,就是功能难以扩展,使用时不方便,且不能作为用户目标板使用,而仿真软件常常又不能进行实时硬件调试.该文以一种便携式普及型多功能51系列单片机开发系统OPEN-SC51的设计为例,提出了解决问题的思路与方法.
8. 学位论文 [伍群慧 可重构可编程开发系统的设计](#) 2005
本文阐述了可重构的数字与模拟可编程开发系统的设计.该系统以FPGA为主控电路,利用FPGA的在线重配置特性,实现系统结构的动态重构.该开发系统由单片机、可编程逻辑器件/和可编程模拟器件三大模块组成,并含有丰富的外围电路,包括RAM、A/D与D/A转换模块、键盘和按键等输入模块、以及数码管、LED点阵和LCD等显示模块.利用主控FPGA丰富的I/O端口资源,通过对FPGA加载不同的配置电路,可以方便地把各功能模块联接起来,组成多种模式的开发系统,适合各种场合下的应用,具有较大的灵活性.在本系统中,使用AT89C2051单片机专门对主控FPGA实现在系统动态配置,达到所需的系统功能.主控FPGA的多个工作模式配置文件存储在一个串行存储器中.为在有限的存储容量下能够多存储配置文件,对配置文件预先进行了无损压缩.单片机在配置FPGA的过程中,先对配置文件解压缩,再通过被动串行方式将配置数据加载到FPGA中.
9. 期刊论文 [高尧林, 罗友美 一种实用的基于本机资源构建的单片机简易开发系统 -仪表技术2002\(2\)](#)
提出一种构建简易开发系统的方法,在研制单片计算机应用系统的过程中,利用单片机应用系统的本机资源、添加相应程序,建立简易、实用的开发系统.
10. 期刊论文 [杨冰 基于MCS-51仿真开发系统的乒乓球游戏的设计 -商情2009\(16\)](#)
设计一款乒乓球游戏能让大家更方便快捷的感受到国球的魅力.该系统灵活方便,能任意改变球的球速和分制,采用开关操作,LED显示,分别对软件和硬

件进行调试,最终达到了软件和硬件的良好结合.该设计采用模块化的设计,程序结构简单,任务明确.

本文链接: http://d.g.wanfangdata.com.cn/Thesis_Y936197.aspx

下载时间: 2010年1月6日