国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
一九八五年第二期　总第五十期　　No. 2 1985 Sum.50

一种微机应用开发的实际方法

欧阳鄂

摘　要　在微机应用中缺乏必需的开发工具是经常会遇到的情况，本文介绍了利用一般的微型计算机来研制微机应用程序的方法，讨论了有关的硬件和软件问题。

一、引　言

目前微机的应用十分广泛。由于微机是一个可编程的器件，它的功能十分复杂。因此应用程序的开发就成为经常遇到的一个课题。

微机应用程序的开发，根据用戶所拥有的手段，可以循不同的途径。

许多的微机应用程序是由人工直接编写机器码完成的。这对比较短的程序来说是一种可取的方法。但当程序较大时，人工编写所需工作量太多，容易出错而且不易排除错误。

利用专用的微机开发系统，这是微机生产厂家推荐的工具。它除了一般微计算机系统的基本组成部份之外，还包含有在线仿眞器和编程器等，并有丰富的软件支援，是一种有力的开发工具。但它是厂家针对自己的微机系列设计的，所以缺乏通用性，而且价格较贵，并不值得普遍推荐。

另一种方法是在小型以上计算机上利用交叉汇编和仿眞程序来研制微机软件。目前一般用戶不具备这样的手段，而且仿眞程序难以模仿复杂的实时的I/O环境。

目前微型计算机价格便宜，日趋普及，已达到了个人计算机的程度。它一般具备鍵盘、显示、打印及磁盘以及比较完整的系统软件，是一个相当可观的资源。如果能加上适当的接口来进行微机应用的开发，我们便有了一个经济的开发工具，而且使用自己熟悉的计算机，对用戶也是很大的方便。本文的目的就是讨论这种方法。

二、开发系统的配置

图1中微型计算机系统（以下简称主机）包括CPU、鍵盘、显示、打印及磁盘；待开发的微机（以下简称微机）与主机之间通过双端口的随机存贮器来接口，并根据需要可以按不同的复杂程度来配置，具体方法如下述。

1. 利用随机存贮器M1作为程序存贮器，由主机将程序写入，接着微机便执行这个程序。M1相当于以后要固化的程序存贮器，微机对M1是只读。405教研室曾采用

一九八五年一月收到

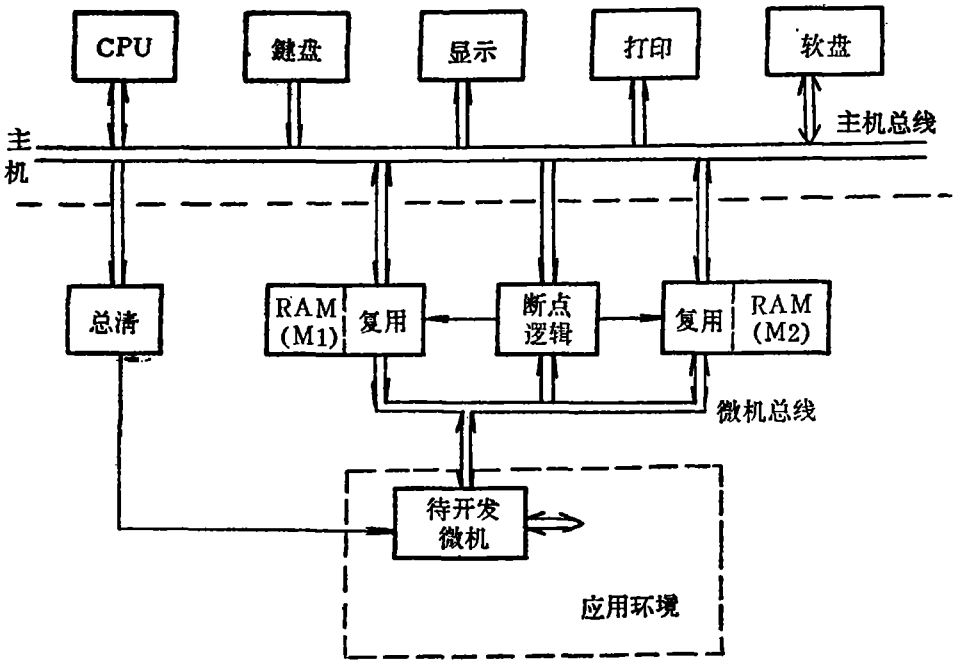

图 1 开发系统的配置

这样的配置利用TRS—80为单片微机Z8研制智能数传机的控制程序和信号处理程序。结果说明这种方法是简易可行的。

2. 为了进一步增强调试程序的能力，可以增加一个随机存贮器M2。在调试程序的过程中，微机可以将它的状态信息写入M2，然后由主机读走并以适当的形式显示在荧光屏上。M2中还可以装入调试用的程序（例如后面要提到的断点服务程序）让微机执行。为便于程序的执行，M1和M2可占用同一地址空间，M1和M2的转接需要相应的控制信号。

在图1中微机是与它所服务的设备直接联接的，这就是说应用程序是在微机的实际工作环境中调试的。微机在电子仪器设备中的应用环境可能是很复杂的。例如在一种数传机中采用了单片微机Z8，后者有一个可编程的串行接口，两个可编程的定时器，32根I/O线和4个优先矢量中断，微机控制着外部电路的动作，外部电路中的各种信号又对微机程序的运行施加影响，在这样复杂的控制和处理的环境中，相互作用是十分复杂的；企图利用仿真程序或在线仿真器来研制程序是难以奏效的；实际可行的办法就是如图1所示意的现场调试。当然，只有在调试不会对用户设备造成损害的条件下才可以作现场调试。

还有一个微机的启动问题，大多数的微机，其程序计数器不能从外部写入，而是在RESET之后从某一固定地址启动。一般在加电（POWER—UP）过程中执行一次RESET，所以可以在用户系统断电的情况下。由主机将程序写入M1，然后将用户系统加电即启动微机。较方便的方法是用户系统不断电，而由主机来控制微机的RESET。这时要注意RESET对微机以外的用户设备的影响，以免引起不正常的情况或损害。

三、软件支持

1. 程序的编写

如果待开发的微机与主机的CPU是同一类型，那就不成问题，可以直接利用主机的汇编程序。但往往会遇到所用的微机与主机的CPU类型不同的情况，这时就需要交叉汇编(CROSS—ASSEMBLING)。厂家所提供的交叉汇编程序多半用FORTRAN写成并运行在分时系统或较大的小型机上。如果只限于最起码的汇编功能，那末在微型计算机上用户自己编写和运行一个交叉汇编程序是可行的，最起码的汇编功能就是：(1)为每条指令分配地址；(2)将指令从汇编语言译成机器码。因为汇编语言程序所占的存贮量可能很大（一般比机器语言大一个数量级以上）而微计算机的内存有限，所以要充分利用磁盘文件功能并采取多次扫视(MULTI—SCAN)。具体做法可以分下面几步：(1)编写源程序（汇编语言）存入磁盘；(2)从磁盘读出源程序，为每一条指令分配地址并建立符号地址表，这是第一轮汇编，逐行进行，随即存入磁盘，结果形成中间文件；(3)再从磁盘读入中间文件，将指令译为机器码，这是第二轮汇编，也是逐行进行，结果在磁盘中得到目的程序；(4)打印程序清单。

一般微型计算机均配有BASIC语言，它是一种比较容易掌握的语言。只要具备字符串操作和文件管理功能，采用BASIC来编写交叉汇编程序也是一个可行的选择。

2. 程序的调试

一个应用程序经过反复的编辑和汇编之后，仍然难免存在错误。不但是程序的编写上可能有错误，而且所依据的处理方法、数学模型也需要在运行中来加以验证和改进。

按照图1的方式，支持调试程序的有效而易行的方法是利用断点(BREAKPOINT)。断点功能在调试程序时是很有用的，我们可以利用断点将程序分割成许多能分别进行测试和易于分析处理的小段，检查其执行情况。

根据用户的需要和复杂程度，设置断点可以用不同的方法：

(1) 最简单的方法就是在断点处直接用断点服务程序替换原有的程序（在存贮器M1中）。断点服务程序是一个短的程序，它的职务就是将断点处微机的状态信息和运算的中间结果送到存贮器M2，以便主机和操作人员监视。这个方法的缺点是每改变一个断点就要重写部分程序。

(2) 利用软件中断，即利用微机的软件中断功能，在要设置断点的地址上，用一条软件中断指令来代替被调试程序原来的指令。当程序执行到断点时便能转入断点服务程序。

显然，软件断点法受到一些限制。首先，如果在应用程序中已经使用了软件中断，此法就不便于使用。此外，软件断点只能在一条指令执行完毕之后生效，而不可能在其它时刻（例如在访问存贮器的时刻）设置断点。

(3) 硬件断点——增加一点硬件可以使断点功能更加完善，其方法就是设置一个硬件地址比较器，由主机将断点地址写入此比较器。当微机地址线上的地址与断点地址相符时，微机即停止M1中程序的执行并转而执行M2中的断点服务程序，M1与M2之间的转接是由地址比较器的符合信号实现的。

四、结 束 语

利用个人计算机，配上合适的接口和少量的软件就可以进行微机应用的开发，这种方法是经济，有效而易行的。

参 考 文 献

[1] Mc Cracken D, Hybrid Tool for Universal Microprocessor Development, Computer Design, V. 19, Apr. 1980

[2] Mc Cracken D., In-situ Emulation Paces New Micros, Computer Design, V. 20 Oct. 1981

[3] Conley S. W., Portable Microcomputer Cross-Assemblers in BASIC, Computer V.8, Oct, 1975

[4] Vincent Tseng, Microprocessor Dev and Dev Systems, Granada, 1982

[5] Korn G. A, Microprocessors and small Digital Computer Systems for Engineers and Scientists, Mc Graw-Hill, 1977

An Economic Approach to Microprocessor Application Development

Ouyang Eh

Abstract

Microprocessor application development based upon an existing microcomputer can be effective and very economic. This paper takes hardware and software into consideration in implementing such a system.