

单片机原理实验指导书

054 系 高 杨 曾 荣

南京航空航天大学

2003 年 9 月

目 录

一.	Insight 仿真器简介及操作说明	1
二.	Insight 仿真器操作指南	1
三.	Medwin 菜单命令	7
四.	Medwin 集成开发环境运算符功能表	10
五.	实验一 数据传送	12
六.	实验二 多字节加法	13
七.	实验三 多字节乘法	15
八.	实验四 无符号数找最大数	19
九.	实验五 数据排序	20
十.	实验六 接口查询方式	23
十一.	实验七 中断方式	25
十二.	实验八 定时器	27

Insight ME-52A 单片仿真器开发系统简介及操作说明

Insight 系列仿真器是南京万利电子有限公司自主研制的 80C51 系列单片机的开发装置，它具有较强的在线仿真及模拟仿真功能，并完全支持 MCS-51 汇编及 C 语言编程，其结构紧凑，集成度高，抗干扰性能强，使用灵活，易于嵌入控制，且具有较高的性能价格比，是 80C51 系列单片机理想的开发工具。

Insight 仿真器的主要性能指标：

1. 仿真 CPU 类型：80C31 80C32 8XC51 8XC52
2. 时钟设置：提供 12M 晶振及 24M 晶振可选
3. 仿真 ROM：64K
4. 仿真 RAM：64K
5. 片内 RAM：128 个字节 RAM；+128 个字节特殊功能寄存器
6. 使用标准+5V 直流电源

操作步骤：

开机：

1. 打开微机主机电源，进入 Windows 操作系统。
2. 接通仿真器+5V 专用外接电源及实验板+5V 电源

关机：

结束实验时，必须关断仿真器与微机电源。

务必遵守实验室安全操作规程，爱护仪器设备，认真进行实验。

操作指南：

设置 MedWin

第一步：启动 MedWin

MedWin 为系统执行文件，其集编辑，编译/汇编，在线及模拟调试为一体。

1. 如果已经连接仿真器，屏幕上出现 MedWin 的初始画面，进入启动 MedWin 的第三步
2. 如果没有连接仿真器，屏幕上出现端口选择画面，进入选择 MedWin 的第二步。

第二步：选择进入在线仿真或模拟仿真

1. 点击 MedWin 后, 仿真器进入正常工作状态, 点击对话框中[仿真器]按钮, 进入在线仿真状态。
2. 点击 MedWin 后, 如果确认需要进行模拟仿真, 点击[模拟仿真]按钮, 进入模拟仿真状态。

经以上步骤, 屏幕上出现 MedWin 集成开发环境的系统主菜单画面。

第三步: 设置工作目录

我们建议用户根据需要选择合适的工作目录, 更改 MedWin 集成开发环境工作目录的步骤, 必须在 Windows 环境下进行:

1. 打开 Windows 的资源管理器
2. 选择我的电脑—本地磁盘。
3. 单击鼠标右键, 选择新建文件夹, 命名为 Workdir / stud31xx/

当建立 Workdir 文件夹后, 在 MedWin 集成开发环境中, 点击当前工作目录菜单, 打开文件夹 (即 目录) Workdir/stud33xx/后, 工作目录设置完成。

第四步: MedWin 集成开发环境初始默认编译/汇编/连接环境为使用万利电子自主知识产权的宏汇编和连接器, 使用默认设置, 汇编连接信息和错误关联均为中文提示。

当选择用户设置编译/汇编/连接环境时, 必须注意选择合适的路径和程序。如:

1. 编译器路径为 C:\C51\BIN\C51.exe
2. 汇编器路径为 C:\C51\BIN\A51.exe
3. 连接器路径为 C:\C51\BIN\L51.exe

系统头文件、库文件的扩展名为:

1. 编译器系统头文件路径为 C:\C51\INC
2. 编译器系统库文件路径为 C:\C51\LIB

MedWin 集成开发环境初始默认程序的扩展名为:

1. C 源程序文件的扩展名为 .C
2. 汇编源程序文件的扩展名为 .ASM

MedWin 开发入门

MedWin 集成开发环境提供了两种方式开发用户应用程序:

1. 不使用 MedWin 集成开发环境项目管理方式——对源程序文件直接进行汇编/连接, 兼容传统开发习惯。
2. 使用 MedWin 集成开发环境项目管理方式——可进行多模块、混合语言编程的联合开发, 同样也适合单模块程序开发。

不使用项目管理方式开发应用程序

不使用 MedWin 集成开发环境项目管理方式, 只能对单模块方式下的应用程序开

发, 具有很大的局限性, 以下是不使用 MedWin 集成开发环境项目管理方式开发应用程序的步骤。

第一步: 关闭当前项目文件

命令: [项目管理][关闭当前项目]

不使用 MedWin 集成开发环境项目管理方式开发应用程序, 用户必须关闭已经打开的项目, 此时 MedWin 集成开发环境关闭界面上所有的窗口, 应为当打开项目文件后, MedWin 集成开发环境默认所有编译/汇编、产生代码的过程都是对项目或项目所包含的文件进行的。

第二步: 在文件菜单下打开文件(应用程序)

命令: [文件][新建]或[打开]

单模块方式下的文件调试可以按照以下方式新建或打开文件

1. 点击[文件][新建], 输入文件名和扩展名, 新建文件。
2. 点击[文件][打开], 选择文件检取框中的文件将其打开。

第三步: 编译/汇编

命令: [项目管理][编译/汇编]

MedWin 集成开发环境根据文件的扩展名, 自动对当前激活的文件选择调用外部编译器或汇编器。

执行[项目管理][编译/汇编]命令后产生的结果显示在消息框中。

第四步: 错误信息关联

文件经过编译/汇编后的结果显示于消息窗口, 消息窗口可由热键 Ctrl+9 激活, 当出现错误后, 消息窗口中的错误信息自动与文件关联。在消息窗口中错误之处双击鼠标左键或键入回车, 即可将错误与文件关联:(仅限 C 语言编程)

第五步: 产生代码并装入入仿真器调试

命令: [项目管理][产生代码]或[项目管理][产生代码并装入]

代码装入仿真器后, 此时 MedWin 集成开发环境进入调试状态。MedWin 集成开发环境中被激活的文件的左侧出现一系列小圆点, 表示程序的有效行, 即此行存在相应的代码, 并且在 LJMP MAIN 行的左侧出现黄色的箭头, 表示当前的程序计数器 PC。

产生代码或产生代码并装入命令对经过编译/汇编无误后产生的 OBJ 文件进行连接产生用于下载的代码, 此命令自动地对修改过的程序进行编译或汇编, 对没有修改过的程序将越过编译或汇编过程, 然后连接所有的 OBJ、LIB 文件, 再装载代码到仿真器, 完成调试程序所需要的准备工作。

第六步: 调试应用程序。

经过第五步操作，为 MedWin 集成开发环境提供了所有准备工作，可以全速运行程序。

第七步：输出代码文件

输出后缀为 LEX 格式文件，及后缀为 BIN 格式文件，装入仿真器。

输出后缀为 LST、ASM 格式文件可打印输出。

使用项目管理方式开发应用程序

使用 MedWin 集成开发环境项目管理方式开发程序，适用于各种方式下的单模块和多模块方式应用程序。以下是使用集成开发环境项目管理方式开发应用程序的步骤。

1. 建立一个新的项目进入第一步——新建项目文件
2. 打开已经存在的项目文件进入第三步——打开项目文件

第一步：新建项目文件

命令：[项目管理][新建项目文件]

项目名称

MedWin 集成开发环境的项目管理器是按项目名称管理的，项目管理器内的项目名称不可以相同。在项目名称输入栏内，用户必须输入项目名称。并且项目名不得超过 8 个字符，不能使用汉字及“?”、“-”、“/”等符号。此时项目名称将等同于一个子目录。

如果源文件与项目文件不在同一目录时，将会出现不能进行源文件调试的情况发生。

添加文件

选择添加文件，表示在项目建立后会自动打开文件检索框，供用户选择文件添加到项目管理器中。

调试信息

建立新的项目后，系统默认为产生调试信息，能够进行原文件调试。

第二步：添加文件

命令：[项目管理][添加文件项]

添加文件命令用于将各类文件添加至项目管理器，[新建项目文件]后集成环境会自动打开添加文件对话框，提示选择。请根据：

1. 如果文件存在，在文件检取中选择并打开文件添加。
 2. 如果文件不存在，必须输入文件名（包括扩展名）建立新文件。
- 新建文件后，键入命令 [项目管理][添加文件项]，将新建文件添加到项目中。

第三步：打开项目文件

命令：[项目管理][打开项目文件]

对于已经存在的项目文件，可以直接打开项目文件进入第四步操作。

第四步：编辑、编译/汇编/连接

命令：[项目管理][编译/汇编]

MedWin 集成开发环境根据文件的扩展名，自动对激活的文件选择调用外部编译器或汇编器：

第五步：错误信息关联

文件经过编译/汇编后的结果显示于消息窗口，出现错误后错误信息与文件关联。在消息窗口中错误之处双击鼠标左键或键入回车，即可将错误与文件关联：（仅限 C 编程）

第六步：产生代码并装入仿真器

命令：[项目管理][产生代码]

命令：[项目管理][产生代码并装入]

1. 使用[产生代码]命令，项目管理器会自动判别文件是否需要重新编译/汇编，提高调试效率。
2. 使用[产生代码并装入]命令，项目管理器会自动判别文件是否需要重新编译/汇编，并将连接产生的代码下载到仿真器，提供调试运行。

第七步：调试应用程序。

经过第五步操作，MedWin 集成开发环境提供了所有准备工作，可以全速运行程序。在文件窗口的左侧出现一系列小圆点，表示程序的有效行，键入 F7 跟踪运行键，程序运行到 Main 函数的第一条语句，并在该行的左侧出现黄色箭头，表示当前的程序计数器 PC。到此 MedWin 集成开发环境提供了调试应用程序的所有准备工作

第八步：输出代码文件

1. 输出 Intel HEX 格式文件。在[项目管理]输出 Intel HEX 后，其自动添加扩展名 HEX。
2. 输出 Binary 格式文件。在[项目管理]输出 Binary 文件后。其自动添加扩展名 BIN。

MedWin 断点功能

MedWin 集成开发环境配合 Insight 系列仿真器，具有强大的断点功能。可以在编辑和调试状态下设置或清除断点，并在退出时自动保存断点信息。

MedWin 断点功能

断点标记:

MedWin 集成开发环境的断点标记, 根据系统状态和所处窗口类型, 标记如下:

1. 在编辑状态下, 源程序文本窗口内的白色箭头——预设断点, 退出时保存,
2. 在调试状态下, 源程序文本窗口内的白色箭头——无效断点, 退出时保存
3. 在调试状态下, 源程序文本窗口内的黄色箭头——有效地址断点, 退出时保

存

4. 在调试状态下, 源程序文本窗口内的蓝绿色箭头——临时断点, 退出时不保

存

5. 在反汇编本窗口内设置的断点, 标记为红色光带——地址断点, 退出时不保

存

断点颜色变化规律:

根据系统状态和所处窗口类型, Medwin 集成开发环境的断点颜色变化规律如下:

编辑状态下, 在源程序窗口内设置断点, 标记为白色; 再次设置断点, 白色断点标记被清除。

调试状态下, 在源程序窗口内设置断点, 标记为黄色; 再次设置断点, 黄色断点标记被清除。

调试状态下, 在源程序窗口内蓝绿色断点处设置断点, 标记为黄色; 再次设置断点, 黄色断点标记被清除。

调试状态下, 在与源程序断点关联的反汇编窗口红色光带处设置断点, 在反汇编窗口内清除断点, 同时将源程序断点标记为白色; 再次设置断点, 反汇编窗口标记为红色光带, 源程序的白色断点再次标记为黄色。

源程序文本上设置断点**编辑状态设置断点**

当用户在编写源程序的过程中, 根据调试需求, 可以直接在源程序文本的任何地方设置或清除断点。如果设置了断点, 将以白色箭头标记在源程序文本左侧的灰色状态栏内, 断点属性为预设断点。对于在源程序文本上设置的断点, 系统退出时全部自动保存。

调试状态设置断点

在调试状态下, 可以直接在源程序文本的有效行上设置断点(当文本左侧的灰色状态栏内出现小圆点时, 表示当前程序为有效行)。设置后, 断点以黄色箭头标记在文本左侧的灰色状态栏内; 如果没有出现小圆点, 则不可以设置断点。在编辑状态下设置的断点, 经过编译/汇编, 如果预设的断点编译后是有效行, 则断点标记为黄色; 如果不是有效行, 断点标记为白色。对于在源程序文本上设置的断点, 系统退出时全部自动保存。

反汇编窗口内设置断点

断点在反汇编窗口内以红色光带表示, 可以任意设置地址断点, 如果设置的断点与源程序有效行地址关联, 则与源程序相关联的地址处设置断点, 在源程序窗口内出

现蓝绿色箭头。

如果在源程序断点相关联的反汇编窗口红色光带处设置断点，将清除窗口内的断点光带，同时将源程序断点标记为白色；再次设置断点，反汇编窗口标记为红色光带，源程序的白色断点再次标记为黄色。

MedWin 菜单命令

文件(F)

新建 (N)

新建文件，在输入文件名时必须输入文件的扩展名。如果是程序文件，汇编语言扩展名必须为(.asm)，C语言扩展名必须为(.c)

打开 (C)

打开用户程序文件，可以在文件检取框中选择，也可以在文件检取框中直接输入文件名，当文件名不存在时，系统默认为新建文件。

打开项目文件 (P)

打开 MedWin 集成开发环境项目文件，项目文件的扩展名为 mpf。打开后，根据需要可以将打开的项目文件添加到项目管理器中。

保存 (S) Ctrl+S

保存当前激活的文件。

另存 (A)

将当前激活的文件另存为指定的文件。

退坝 (X)

退出 MedWin 集成开发环境。

编辑(E)

撤消 (U)

撤消当前的操作。Ctrl+Z

剪切 (T)

将选择的块剪切到剪贴板。Ctrl+X

复制 (C)

将选择的块复制到剪贴板。Ctrl+C

粘贴 (P)

将剪贴板粘贴到文件。Ctrl+P

选择整个文档 (L)

将整个文档作为块。Ctrl+A

查找 (F)

在文件中查找字符串。Ctrl+F

替换 (E)

替换匹配的字符串。Ctrl+H

定位 (G)

定位到指定行或地址。Ctrl+G

定位到前一个错误 V)

将编译 / 汇编发生的错误与源程序关联, 并定位到前一个错误的位置。

定位到后一个错误 N)

将编译 / 汇编发生的错误与源程序关联, 并定位到后一个错误的位置。

查看(V)**寄存器 (R)**

寄存器窗口, 显示 80c51 内部基本的寄存器 RO—R7, A, B, DPH, DP 乙, SP 和 PSw, 以 16 进制方式显示字节寄存器的内容, 以位的方式显示 PSW 的内容。

特殊功能寄存器 (S)

当前被选择的 CPU 所包含的特殊功能寄存器窗口, 以 16 进制方式显示字节寄存器的内容。

反汇编窗口 (C)

反汇编窗口, 将程序代码区的内容以反汇编方式及源程序方式显示。反汇编窗口同时支持行汇编方式输入或修改程序代码。

观察窗口第一观察组 (1) Ctrl+1

为了方便用户避免多次添加和删除需要观察的变量, 设置的第一组观察窗口。

观察窗口第二观察组 2) ctrl+2

为了方便用户避免多次添加和删除需要观察的变量, 设置的第二组观察窗口。

同理第三观察窗口第四观察窗口) Ctrl+3、Ctrl+4

数据区 Data

片内 RAM 和 SPR 区域, 被直接寻址访问的数据区。

数据区 Code

程序代码空间。

数据区 xData

外部数据空间。

数据区 Bit

片内的位寄存器区域 0—127 与片内地址为 20H—2FH 的 RAM 对应, 128—255 与 SFR 相对应。

数据区 PData

页面存储方式下的外部数据空间, 使用 MOVx @ri, a 或 MOVXa, @ri 指令。

调试 (R)

切换到调试态, 如果已经打开了项目文件, 则进行产生代码并装入操作。

终止调试 (D) CtrI+D

切换到编辑态。

全速运行 (R) F9

全速运行, 调试态有效。

跟踪 (T) F7

跟踪运行程序, 在反汇编窗口下执行一条指令, 如果当前是调用指令, 则进入所调用的子程序; 如果在源程序窗口下, 执行当前文本下的一条语句, 如果是调用则进入所调用的子程序。调试态有效。

单步 (S) F8

单步运行程序。反汇编窗口下如是调用指令, 则越过所调用的子程序; 源程序窗口下, 如果是调用语句, 则越过所调用的子程序。调试态有效。

运行到光标处 (G) F4

程序全速运行到光标处, 调试态及源程序文本或反汇编窗口有效。

交互跟踪 (A) Ait+F7

如果当前激活的是程序文本窗口, 执行反汇编窗口的跟踪指令; 如果当前激活的是反汇编窗口, 执行源程序文本窗口的跟踪指令, 调试态有效。

执行到 (E)

执行到设定的地址, 调试态有效。

自动单步 (M)

自动以跟踪的方式运行程序, 调试态有效。

程序复位 (P) CtrI+F2

复位仿真器, 调试态有效。

外围部件(S)

中断

中断状态窗口, 包括 INTO, INTI, To, T1, T2 和 UART 中断状态以及优先级和允许设置。设置或清除相应的标志, 可以改变中断的状态, 也可以通过相应的值, 作为中断初始化的编程。

MedWin 集成开发环境的观察窗口, 由观察窗口 1 至观察窗口 4 和变量窗口组成, 方便用户在调试不

同函数 / 模块时观察变量, 变量窗口动态显示相应作用域内的局部变量。当退出 MedWin 集成开发环境后,

系统保存当前所有的观察变量。

Medwin 集成开发环境, 支持 FraMk 帅 / Kei[所有数据类型的观察, 必须注意区分大小写!

标识符

MedWin 集成开发环境对于表达式所用的标识符定义如下:

1. 由英文字母 A—Z, a—z, 数字 0—9 和下划线 B—曲成
2. 第一个字符必须是字母或下划线, 其余任意组合
3. 最大长度 32 个字符
4. 不得与保留字相同

MedWin 集成开发环境对于表达式支持的运算符见下表, 使用符合汇编和 C 语言通常习惯.

(32)

运算符	功能	类型	表达式举例
HIGH , Low	取字类型的高低字节	数值	LOW(65536.1000)
	取成员	数值	"ptr, ' ptr.. 10
巴	取地址	数值	8Lpoint
G	逻辑非	逻辑	[B00[
NOT. .	按位取反	数值	. A
., /	乘, 除	数值	A' B
MOD, %	取模	数值	AMOD10
十. .	加, 减	数值	A + B
SHL, <<	左移	逻辑	A << 2
SHR, >>	右移	逻辑	B >> 4
LT, <	小于	逻辑	A < B
LE. <:	小于或等于	逻辑	A <= B
GT. >	大于	逻辑	A > 8
GE. >=	大于或等于	逻辑	A >: B
EQ. : =	等于	逻辑	AEOB
NE. 卜	不等于	逻辑	ANE B
AND, 8C	与	数值	AAND B
XOR. “	异或	数值	AXORB
ORB, [	或	数值	AORB
[]	数组下标	数值	Item[1 四
	重复	数值	Data(20h).. 5
[DATA	间址访问的片内 RAM 区	数值	[Data(0010h)
DATA	直接寻址访问的片内数据区	数值	Data(OOE0h)
x □ ATA	片外数据区	数值	xdata(32768)
CODE	代码区	数值	code(0030h)

PDATA	外部页面数据区	数值	Pdata(OOPFh)
B[T	位寄存器区	数值	Bit(o)

实验一 数据传送

一. 实验目的

1. 通过实验掌握仿真器的使用方法。
2. 通过实验了解数据传送类指令的常用方法。
3. 熟悉 8031 内部 RAM, 和外部 RAM 之间的数据传送指令, 从而了解 MCS—51 存储空间的分布以及内外数据存储器的特点。

二. 实验设备:

微机 1 台, 80C51 仿真器 1 台, +5 V 稳压电源 1 台

三. 实验内容

将 8031 内部 RAM ^{40 4A}30H—3FH 单元分别置初值 ⁰⁻¹⁰A0—AFH, 然后将 30H—3FH 单元内容分别传送到片外 RAM 的 A800H—A80FH 单元, 再将片外 A800H—A80FH 单元内容传送到片内 50H—5FH 单元中。

四. 参考程序

地址	目的代码	源程序
		ORG 0000H
0000H	21 00	AJMP MAIN
		ORG 0100H
0100H	78 30	MAIN: MOV R0, #30H
0102H	7A 10	MOV R2, #10H
0104H	74 A0	MOV A, #A0H
0106H	F6	A1: MOV @R0, A
0107H	08	INC R0
0108H	04	INC A
0109H	DA FB	DJNZ R2, A1
010BH	78 30	MOV R0, #30H
010DH	90 A8 00	MOV DPTR, #A800H
0100H	7A 10	MOV R2, #10H
0112H	E6	A2: MOV A, @R0
0113H	F0	MOVX @DPTR, A
0114H	08	INC R0
0115H	A3	INC DPTR

0116H	DA FA		DJNZ	R2, A2
0118H	78 50		MOV	R0, #50H
011AH	90 A800		MOV	DPTR, #A800H
011DH	7A 10		MOV	R2, #10H
011FH	E0	A3:	MOVB	A, @DPTR
0120H	F6		MOV	@R0, A
0121H	08		INC	R0
0122H	A3		INC	DPTR
0123H	DA FA		DJNZ	R2, A3
0125H	02 01 25		LJMP	\$
			END	

五. 实验步骤:

1. 输入程序并调试
2. 分别用单步及跟踪方法调试程序,
3. 用查看方法检查 RAM 30H---3FH, XCODE 区 A800H—A80FH, 50H—5FH 各单元内容。
4. 试编写一程序, 将 0---10 送入 40---4A 单元。

实验二 多字节加法

一. 实验目的:

1. 通过实验掌握多字节二进制加法的程序设计方法
2. 了解十进制调整指令 DAA 的转换依据

二. 实验设备:

微机 1 台, 80C51 仿真器 1 台, +5 V 稳压电源 1 台

三. 实验内容:

三字节加法, 被加数置于 30H---32H 单元

加数置于 33H---35H 单元

结果置于 50H---52H 单元

四. 参考程序:

地址	目的程序	源程序
		ORG 0000H
0000H	21 00	AJMP MAIN
		ORG 0100H
0100H	E5 30	MAIN: MOV A, 30H
0102H	25 33	ADD A, 33H
0104H	D4	DAA
0105H	F5 50	MOV MOV 50H, A
0107H	E5 31	MOV A, 31H
0109H	35 34	ADDC A, 34H
010BH	D4	DAA
010CH	F5 51	MOV 51H, A
010EH	E5 32	MOV A, 32H
0110H	35 35	ADDC A, 35H
0112H	D4	DAA
0113H	F5 52	MOV 52H, A
0115H	E4	CLR A
0016H	34 00	ADDC A, #00H
0018H	F5 53	MOV 53H, A
001AH	80 FE	SJMP \$
		EDN

五. 实验步骤:

1. 输入程序并调试
2. 分别用单步及跟踪方法调试程序,
3. 试求 $362418+382615=?$

分别置被加数与加数

$$\begin{array}{r} 32H \quad 31H \quad 30H \quad + \quad 35H \quad 34H \quad 33H \\ 36 \quad 24 \quad 18 \quad + \quad 38 \quad 26 \quad 15 \\ \hline \end{array}$$
 结果置于 50H---52H 单元 (52H)=74 (51H)=50 (50H)=(33)

4. 观察 DAA 指令前后 A 的变化, 从而了解 DAA 指令的作用

六. 思考题:

1. 自编一加法程序, 从而达到熟悉指令的目的。

如: $1+2+ \dots +10$

实验二 多字节乘法

一. 实验目的:

1. 通过被乘数为三字节, 乘数为一字节的乘法和被乘数、乘数均为二字节的乘法程序, 掌握计算机内部进行乘法的原理和法则。
2. 掌握二进制数的运算规律。

二. 实验设备:

486 以上微机 1 台、80C51 仿真器 1 台, +5V 稳压电源 1 台

三. 实验内容:

1. 三字节与一字节的乘法

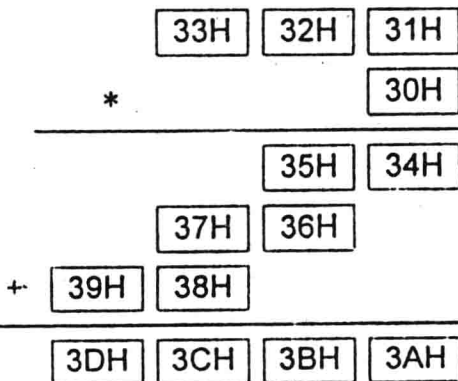
被乘数放在 30 单元, 被乘数放在 31H---33H 单元, (从低到高字节), 乘积放在 3AH---3DH 单元 (从低到高字节)。

例如: $44 \ 2211 \times 11 = ?$

(30H)=11 (31H)=22 (32H)=33 (33H)=44

算法:

21 09 15
x 10



(30H) * (31H) 部份积:

低字节置于 34H, 高字节置于 35H

(30H) * (32H) 部份积:

低字节置于 36H, 高字节置于 37H

30H) * (33H) 部份积:

低字节置于 38H, 高字节置于 39H

积置于 3AH---3DH 单元中

2. 两个双字节数相乘, (移位相加法)

$R2R3 \times R6R7 = R4R5R6R7$

被乘数和乘数分别放入 R2 R3 * R6 R7, 结果放 R4 R5 R6 R7 中

四. 参考程序 1