目录

第 1 章 Philips 单片机 P89V51RD2 简介……1
1.1 概述……1
1.2 特性……1
1.3 引脚图……2
第 2 章 ISP 与 IAP 简介……3
2.1 名词解释……3
2.2 传统的烧录方式……3
2.3 ISP 烧录方式……3
2.4 IAP 烧录方式……3
第 3 章 P89V51RD2 的 IAP……4
3.1 P89V51RD2 的 Flash 存储结构……4
3.2 数据手册上提供的 IAP 操作……4
3.2.1 IAP 子程序介绍……4
3.2.2 在汇编语言里如何调用 IAP 子程序……4
3.2.3 IAP 子程序功能……5
3.3 如何在 C51 中调用 IAP 子程序……7
3.3.1 参数传递的困难……7
3.3.2 在 C51 源程序中直接嵌入汇编指令的方法……7
3.3.3 函数绝对地址定位……7
3.3.4 举例……8
第 4 章 IAP 底层操作研究……9
4.1 IAP 操作的疑问……9
4.2 对 V05 版本 ISP BOOT 程序的反汇编研究……9
4.3 与 Flash 操作相关的 SFR 寄存器……15
第 5 章 热复位时不影响片内 RAM 的 BOOT 程序……17
5.1 热复位时的片内数据 RAM……17
5.2 从引导 Flash 转移到用户 Flash 的方法……17
5.3 从用户 Flash 转移到引导 Flash 的方法……17
5.4 热复位时不影响片内 RAM 的 BOOT 程序……17
第 6 章 如何在 Keil C51 中更好地使用 P89V51RD2……19
6.1 简单的用法……19
6.2 如何将 P89V51RD2 添加到 Keil C51……19
6.3 重新定制头文件……20
第 7 章 在引导 Flash 中如何实现 IAP……26
7.1 联合整型的用法……26
7.1 IAP 头文件源代码……27
第 8 章 定制自己的 ISP 程序……30
8.1 定制 ISP 程序的功能要求……30
8.2 Intel HEX 格式说明……30
8.3 一个分析 Intel HEX 记录的简单程序……30
8.4 定制的简易 ISP 程序的源代码……32

第 9 章 IAP 远程升级演示实例……..…39
9.1 远程升级实例连接原理……..……39
9.2 终端电路原理……..……40
9.3 流程分析……..…41
9.4 参考资料……..……45
9.5 文档信息……..……45
9.5.1 版本声明……..……45
9.5.2 文档版本……..……45

第1章 Philips 单片机 P89V51RD2 简介

1.1 概述

P89V51RD2 是一款带有 64KB 程序 Flash 和 1024B 数据 RAM 的 80C51 微控制器。

P89V51RD2 有一个关键的特性就是其具有 X2 模式。在实际应用中，设计工程师可以让芯片工作于传统 80C51 的 12 时钟模式（12 个时钟周期合 1 个机器周期），或者选择 X2 模式（6 个时钟周期合 1 个机器周期），在相同时钟频率下获得双倍的吞吐量。另一种从这种特性获益的方法是降低一半的时钟频率仍然可以保持相同的性能，这可以显著地降低 EMI（电磁干扰）。

Flash 程序存储器支持并行烧录和串行在系统烧录（ISP）。并行烧录模式提供高速的分组烧录，节省烧录开销和上市时间。ISP 允许在软件的控制下对最终产品进行重复烧录。

P89V51RD2 也可以在应用烧录（IAP），这允许 Flash 程序存储器重新配置，甚至是在运行当中。

1.2 特性

- 80C51 核心处理单元；
- 5V 工作电压，0～40MHz；
- 64KB 片内 Flash 程序存储器，带 ISP（在系统烧录）和 IAP（在应用烧录）；
- 支持 12 时钟（默认）和 6 时钟模式；
- SPI（串行外围接口）总线和增强型 UART；
- PCA（可编程计数阵列），带 PWM 和捕获/比较功能；
- 4 个 8 位 I/O 口，带 3 个高电流 P1 口引脚（每个 16mA）；
- 可编程看门狗定时器（WDT）；
- 8 个中断源，4 个优先级；
- 两个 DPTR 寄存器；
- 低 EMI 模式（ALE 禁止）；
- TTL 和 CMOS 逻辑电平兼容；
- 掉电保护；
- 低功耗模式：外部中断可唤醒的掉电模式和空闲模式。

1.3 引脚图

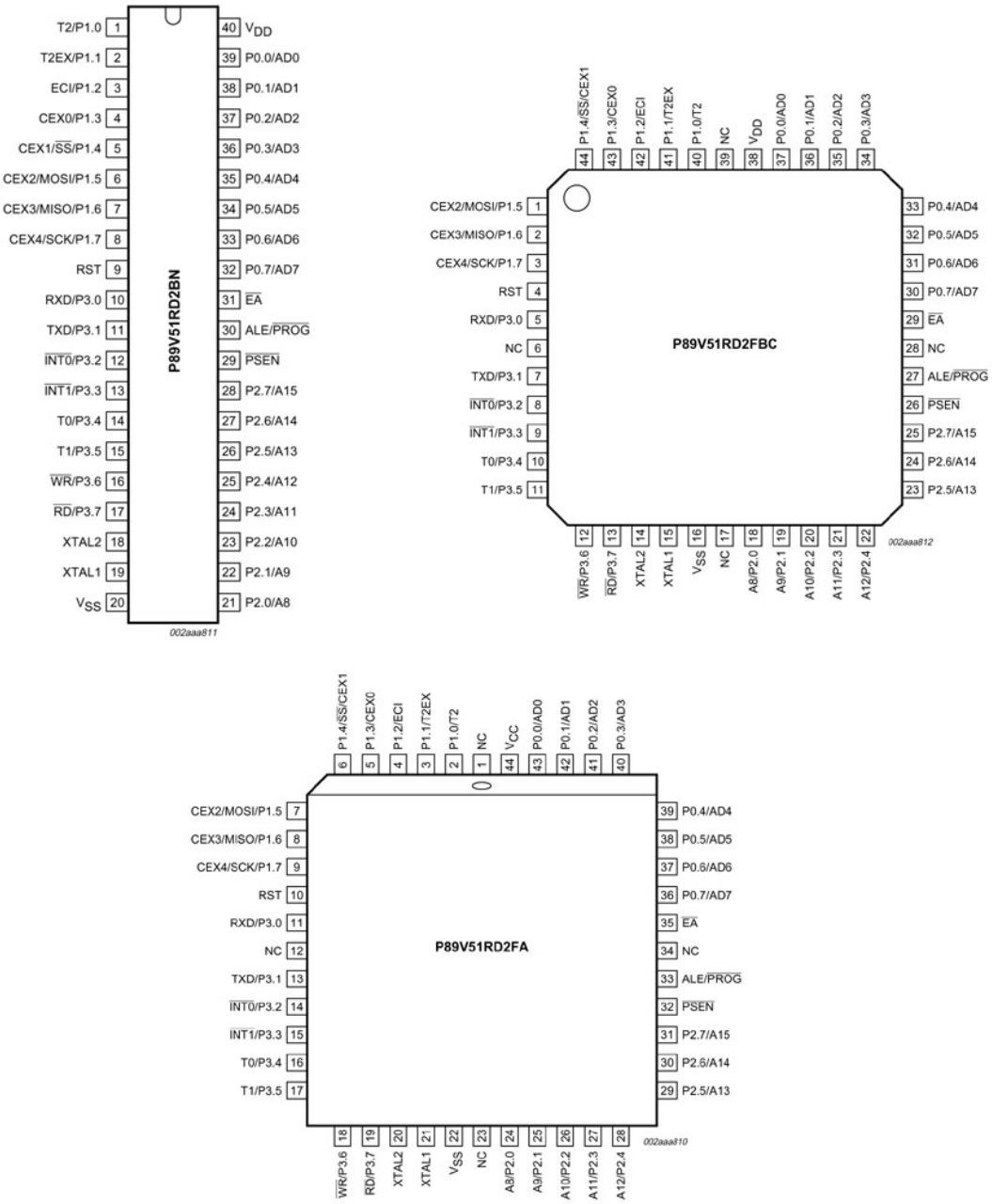


图 1.1.1 P89V51RD2 引脚图

第2章 ISP 与 IAP 简介

2.1 名词解释

ISP 是“In System Programming”的缩写，意为“在系统烧录”。

IAP 是“In Application Programming”的缩写，意为“在应用烧录”。

2.2 传统的烧录方式

对单片机的程序烧录传统上是采用并行烧录器。这种方法要求先把芯片放在烧录器的夹座上进行烧录，然后取下芯片再安装到电路板上。如果要进行反复烧录，则往往要在电路板上为芯片配一个 DIP 或 PLCC 的底座。显然，这要占用比较大的面积，而且插拔也不是特别方便。对于贴片封装，比如 QFP，焊到电路板上之后，如果发现程序有误则应当重新烧录，但是想取下来时却发现并不是件容易的事情，往往只得报废。

2.3 ISP 烧录方式

ISP 烧录方式较好地解决了上述问题。它提供了一个串行的烧录接口，不必取下芯片就可以进行烧录，非常方便，尤其是对于贴片封装的芯片。

2.4 IAP 烧录方式

但是 ISP 烧录方式也有其不足之处，每次烧录都必须要和计算机（或专用烧录设备）连机才行。有些用户希望产品在实际应用期间能够通过某种远程通信方式自动地更新程序内容，显然 ISP 已经无法满足这样的需求了，必须要找到新的办法。新出现的 IAP 烧录方式为程序的自我更新提供了有效手段。单片机内部的 Flash 存储器保存有用户的程序代码，这些代码在正常运行期间是不能被修改的。但是有了 IAP，用户程序就能够根据需要（满足某种条件）自行修改部分甚至全部程序代码。新的程序代码可能是程序在运行过程中自动生成的，也可能来自于远程设备，而后者更普遍。

第3章 P89V51RD2 的 IAP

3.1 P89V51RD2 的 Flash 存储结构

P89V51RD2 片内实际上有两块 Flash 存储器，一块是用户 Flash，有 64KB；另一块是引导 Flash，有 8KB；两者合计为 72KB。64KB 的用户 Flash 分为 512 个扇区，每个扇区 128 字节，在 IAP 功能里可以按扇区擦除用户 Flash，具体细节在后续章节里会有详细说明。引导 Flash 的地址范围是 0000H～1FFFH，与用户 Flash 地址有重叠，但实际上位于另外的物理空间，不会与用户 Flash 产生冲突。

引导 Flash 在芯片出厂时保存有 Philips 公司提供的 ISP 程序（内含 IAP）。用户可以使用像 FlashMagic 这样的软件以 ISP 的方式烧录用户程序。在硬件复位时（上电复位和外部复位），程序首先从引导 Flash 的 0000H 处执行程序。如果不需要烧录任何程序，则 ISP 引导程序会自动退出，并转到用户 Flash 的 0000H 处继续执行。P89V51RD2 有软件复位（包括内部看门狗复位）功能，复位时直接从用户 Flash 的 0000H 处执行，而引导 Flash 不会被激活。如果有必要，用户自己也可以定制引导程序。编写引导程序与编写正常的应用程序没什么两样，也能够使用全部指令集、访问全部寄存器（含 SFR）和读写全部 RAM。但有一点，引导程序最终还是要转到用户 Flash 的，具体的转移方法在后续的章节中会有详细说明。正在执行的用户程序是无法直接转到引导 Flash 的，只能通过 IAP 子程序调用或者硬件复位。

3.2 数据手册上提供的 IAP 操作

3.2.1 IAP 子程序介绍

Philips 公司针对 P89V51RD2 的 IAP 功能提供了封装好的子程序，用户只要用正确的方法调用此子程序就能够实现 IAP 操作。IAP 子程序的入口地址被称为“PGM_MTP”，约定用寄存器 R1、ACC、DPTR 等来传递参数（具体的操作在后面会有详细解释）。

应当注意的是，用户程序调用 IAP 子程序的调用点必须位于地址 2000H 以后，否则无法正确执行。这是因为在调用操作期间，8KB 的引导 Flash 会映射到程序存储器空间的 0000H～1FFFH，而原来用户 Flash 的 0000H～1FFFH 这段空间会被屏蔽掉。另外的注意事项是必须关闭所有中断（可以通过执行“CLR EA”指令来实现）。这是因为中断向量的入口地址也在前 8KB 空间，如果不关闭中断，一旦在 IAP 子程序执行期间发生了中断，程序就会转到引导 Flash 的相应中断向量入口处执行程序，然而那里可能并不包含您所期望的中断服务例程，接下来将导致不可预料的结果。

3.2.2 在汇编语言里如何调用 IAP 子程序

IAP 子程序的入口地址是 1FF0H，被称为“PGM_MTP”。在汇编程序里，首先要用 EQU 伪指令定义 IAP 子程序入口地址 PGM_MTP，即“PGM_MTP EQU 1FF0H”。调用 IAP 子程序的调用点必须位于 2000H 之后。参数传递：工作寄存器 R1 保存 IAP 子程序的功能号，ACC 和 DPTR 是入口参数，返回值保存在 ACC 中。调用 IAP 子程序之前应当关闭总中断。此外还要激活引导 Flash，方法是执行“ANL 0B1H,#0FCH”指令，其中“0B1H”是有关 Flash 操作的 SFR 寄存器，后面的章节中会后详细解释。最后执行“LCALL PGM_MTP”指令实现具体的 IAP 功能。调用完毕后，别忘了执行“ORL 0B1H,#01H”指令，以恢复被屏蔽的用户代码 0000H～1FFFH，同时引导 Flash 被隐藏。下面以读取用户 Flash 数据功能为例（全部功能将在下一小节中给出），来说明 IAP 子程序的调用是如何具体实现的：

程序清单 3.1 在汇编语言里调用 IAP 子程序的方法

```
;实现读取用户 Flash 数据的 IAP 子程序
;这段程序必须安排在 2000H 以后
;(R6,R7)=地址
;R5=读取到的数据
UserFlashRead:
    MOV     R1, #02H        ;设定 IAP 功能号
    MOV     DPH, R6         ;设定地址高字节
    MOV     DPL, R7         ;设定地址低字节
    CLR     EA              ;关闭总中断
    ANL     0B1H, #0FCH     ;激活引导 Flash
    LCALL   PGM_MTP         ;调用 IAP 子程序
    ORL     0B1H, #01H      ;关闭引导 Flash
    SETB    EA              ;重新打开总中断
    MOV     R5, ACC         ;保存读到的数据
    RET                     ;返回
```

应当指出的是，所谓程序的调用点必须位于 2000H 之后的说法，更精确地是指“ANL 0B1H,#0FCH”这条指令的存储位置。因为只有执行了这条指令，引导 Flash 才开始映射到程序存储器空间的 0000H～1FFFH。

3.2.3 IAP 子程序功能

在 Philips 提供的 P89V51RD2 数据手册里详细地给出了 IAP 子程序的各项功能列表。下面就给出比较完整的总结：

■ 读取 ID（Read ID）

输入参数：

R1＝00H

DPH＝00H

DPL＝00H：读取厂商 ID

DPL＝01H：读取器件 ID

DPL＝02H：读取 ISP 版本号

DPL＝03H：读取 IAP 版本号

返回参数：

ACC＝所需参数（ID 或版本号）

说明：

就通常的操作来讲，这一功能基本上不被使用。

■ 擦除第 0 块（Erase Block 0）

输入参数：

R1＝01H

返回参数：

ACC＝00H：通过

ACC!=00H：失败

说明：

第 0 块指的就是用户 Flash。通常，请勿执行此功能，否则用户 Flash 将被全部擦除，从而导致彻底死机。

■ 烧录用户代码（Program User Code）

输入参数：

R1＝02H

DPH＝存储器地址高字节

DPL＝存储器地址低字节

ACC＝要烧录的字节

返回参数：

ACC＝00H：通过

ACC!=00H：失败

说明：

每次只能烧录一个字节。烧录之前必须保证该存储单元已经被擦除过了。一个字节烧录完毕后会自动进行校验，校验是否成功通过 ACC 来反映。

■ 读用户代码（Read User Code）

输入参数：

R1＝03H

DPH＝存储器地址高字节

DPL＝存储器地址低字节

返回参数：

ACC＝器件数据

说明：

每次只能读取一个字节。实际上完全可以用“MOVC A,@A+DPTR”指令来代替此功能。

■ 烧录加密位和双倍时钟位（Program Security Bit,Double Clock）

输入参数：

R1＝05H

DPL＝01H：烧录加密位

DPL＝05H：烧录双倍时钟位

返回参数：

ACC＝00H：通过

ACC!=00H：失败

说明：

如果加密位或双倍时钟位未被烧录过，则可以执行此 IAP 功能。一旦烧录，就无法再恢复（擦除），除非使用商业并行烧录器。

■ 读取加密位和双倍时钟位（Read Security Bit,Double Clock）

输入参数：

R1＝07H（原文是“ACC＝07H”，显然有误）

返回参数：

ACC.0＝双倍时钟位（DBL_CLK）：0－12 时钟模式，1－6 时钟模式

ACC.2＝加密位（SB）：0－未加密，1－已加密

ACC.4＝序列号是否匹配（S/N-match）

ACC 的其它位都是 0

说明：

序列号是否匹配（S/N-match）是属于 ISP 的功能，可以防止被解密。

■ 扇区擦除（Sector Earse）

输入参数：

R1＝08H

DPTR＝扇区首地址

返回参数：

ACC＝00H：通过

ACC!=00H：失败

说明：

64KB 的用户 Flash 分为 512 个扇区，编号 0～511，每个扇区 128 字节。扇区首地址＝扇区号×128。

3.3 如何在 C51 中调用 IAP 子程序

3.3.1 参数传递的困难

Philips 提供的 IAP 子程序接口，参数传递是采用寄存器 ACC、R1、DPTR 来实现的。在 C51 中这些寄存器都是系统级的，用通常的手段无法直接访问。通常的解决方法是直接嵌入汇编指令或调用汇编语言子程序。

3.3.2 在 C51 源程序中直接嵌入汇编指令的方法

在 C51 中直接嵌入汇编代码是解决 IAP 子程序参数传递的一种比较简单的方案。下面就以 Keil C51 为例，简要介绍一下具体是如何操作的：

- 在 C 文件中用“#pragma asm”嵌入汇编指令

  #pragma asm

  ;在这里添加汇编指令

  #pragma endasm

- 编译环境设置

  在“Project”窗口中包含有汇编代码的 C 文件上点右键，选择“Options for ...”，然后在弹出的对话框里点击右边的“Generate Assembler SRC File”和“Assemble SRC File”项，使检查框由灰色变成黑色（有效）状态。

- 添加必要的库文件

  根据选择的编译模式，把相应的库文件加入工程中，该文件必须作为工程的最后一个文件。通常就是 Small 模式，把目录“Keil\C51\Lib”下的“C51S.Lib”文件复制到工程文件夹下，并添加到工程中。

- 编译，生成目标代码

3.3.3 函数绝对地址定位

还有一个问题，调用 IAP 子程序时，调用点必须位于 2000H 以后，在 C51 中如何实现呢？通常，编写 C51 程序时，程序员并不关心某个函数位于什么样的绝对地址。但是 Philips 已经规定 IAP 子程序的调用点不能在 2000H 之前，所以相关函数的绝对地址定位是必须的。在上一节中，我们注意到，所有调用 IAP 功能的程序都有一个公共部分，即：

```
CLR     EA          ;关闭总中断
ANL     0B1H, #0FCH ;激活引导 Flash
LCALL   PGM_MTP     ;调用 IAP 子程序
ORL     0B1H, #01H  ;关闭引导 Flash
SETB    EA          ;重新打开总中断
```

我们可以把这一段单独写成一个子函数，并采用特殊的办法把它定位在 2000H 以后。需要指出的是：调用此子函数的程序还是可以位于 2000H 之前的。该函数的源程序代码如下：

程序清单 3.2　在 C51 中调用 IAP 子程序的公共函数

```
/*
函数：CallIAP()
功能：调用 IAP 子程序
*/
void CallIAP()
{
    #pragma asm
    CLR     EA
    ANL     0xB1, #0xFC
    LCALL   0x1FF0
    ORL     0xB1, #0x01
    SETB    EA
    #pragma endasm
}
```

显然，把此函数定位在 64KB 用户 Flash 的最后比较合适。具体方法是，打开“Option for target”选项，在“BL51 Locate”选项卡里的 Code 栏填入：

?PR?CallIAP?CFileName(0xFF80)

其中 CallIAP 是函数名，CFileName 是函数 CallIAP 所在的 C 文件的文件名（不要带扩展名），0xFF80 是函数定位地址（正好是最后一个扇区的首地址）。

3.3.4　举例

以下函数可以实现扇区擦除功能：

程序清单 3.3　C51 中的 IAP 扇区擦除函数

```
/*
函数：SectorErase()
功能：扇区擦除
参数：
    sector：扇区编号，取值 0～510
说明：
    最后一个扇区（511）保存有公共函数 CallIAP()，不允许被擦除
*/
void SectorErase(unsigned int sector)
{
    unsigned int addr;
    if ( sector >= 511 ) return;
    addr = sector * 128;
    DPH = (unsigned char)(addr >> 8);
    DPL = (unsigned char)(addr & 0x00FF);
    #pragma asm
    MOV     R1, #0x08
    #pragma endasm
    CallIAP();
```

```
}
```

这篇文档仅提供了全文的前三章，如果您需要整篇文章及所有代码，请与当地的 ZLG 销售部门联系。

3.4 参考资料

《P89V51RD2 数据手册》，www.zlgmcu.com；
《ZNE-100T 说明书》，www.zlgmcu.com；
《跨网关通信中路由器的设置》，www.zlgmcu.com。

3.5 文档信息

3.5.1 版权声明

本文档版权归 **广州周立功单片机发展有限公司** 所有，未经书面许可不得随意修改本文档中相关内容；如有违反，本公司保留对其提起法律诉讼的权利。本公司保留在不通知读者的情况下，修改文档相关内容的权利。对于使用中所出现的任何效果，本公司不承担任何责任。

3.5.2 文档版本

表 3.1 文档版本

版本号	发布日期	说明
V1.0	2005-9-28	原始版本