

可编程单片机外围器件PSD8XXF系列**应用笔记052****PSD813FH-80C31设计举例****一、引言**

本文介绍基于闪速存储器 (FLASH-based) 的嵌入式微控制器设计, 其特点是使用了WSI公司的闪速可编程系统器件 (Programmable System Device—PSD) 和Intel公司的80C31。这两种芯片结合在一起用作嵌入式处理器内核, 这是一种适用于许多应用的极好的配合。从闪速存储器下载并执行代码的有关问题被简化。

80C31BH-1N是非常廉价的无ROM微控制器单元 (MCU), 具有所有与器件-技术保障有关的优点, 而器件-技术保障问题在工业界中总是存在的。WSI PSD813FH器件是多芯片模块 (multi-chip module—MCM)。它是新型PSD系列中第一种器件, 所有此类器件均是以闪速存储器为基础 (FLASH-based) 并且系统内可编程 (In-System-Programmable—ISP)。这种MCM和低生产成本的单片闪速PSD系列器件是引脚对引脚 (pin-for-pin) 向上兼容的, 在单片器件可供使用之前, 它一直可用于开发。单片器件具有低功耗的特点, 对于袖珍式、低功耗、低成本的应用, 这种方案可成为十分理想的MCU内核。

典型的带有闪速存储器的MCU设计包含MCU、主闪速存储器以及实现ISP下载至主闪速存储器的引导PROM或SRAM。对于把SRAM用于ISP的系统, 必须首先把闪速编程算法下载到SRAM, 然后在ISP期间内MCU从SRAM执行。任何所发生的电源中断或系统闪变将损坏系统。因此对于要求高系统可靠性的应用, 引导PROM是必需的。但是, 引导PROM增加了系统的成本, 一旦投入使用难于更新。FLASH PSD针对这些所关心的问题, 把所有必需的元件组合在一起, 这些元件使MCU易于在线下载主闪速存储器和引导存储器。(在单片器件中引导存储器ISP可供使用)。

这个MCM设计中, 利用MCU的UART通道和位于PSD内的OTP引导代码, 可在系统内从主计算机下载固件并将其编程到MCM闪速存储器中。此引导代码必须在PSD插入电路之前先用WSI MagicPro III器件编程器编程。引导代码必须包含所有访问闪速存储器以及处理UART从主计算机下载协议的MCU子程序。

当单片器件可供使用时, 所有固件 (包括引导存储器和主存储器二者) 以及PSD配置可以用IEEE 1149.1 JTAG口在系统内 (in-system) 下载到全空白的器件。主存储器和引导存储器二者仍可经过UART下载。

你可以一步一步了解此设计并把重点放在系统设计至器件编程的范围内。此设计的结构为易于转至单片闪速PSD813FX器件打下了基础。

二、MCM结构

PSD813FH是在单片PSD813FX器件可供使用之前用于开发设计的中间产品。MCM由三个连接到多层基片的管芯组成, 这三个管芯是: PSD6xx、4兆位闪速存储器以及地址锁存器。

※ PSD6xx管芯包括实现单片器件功能子集的所有必需功能 (PLD、引导OTP PROM、SRAM、I/O端口、页寄存器)。

※ 使用4兆位闪速存储器管芯, 但是只有1兆位可供使用 (在“访问MCM闪速存储器管芯”一节中说明)。

※ 锁存器用于解决对闪速存储器芯片地址和数据线的多路复用。

MCM为52引脚PLCC封装, 其引脚分配和单片PSD813FX器件相同, 表1表示MCM和单片器件之间的主要差别。图1是表示三个管芯和外部引脚之间主要连接的MCM的方框图。

表1 MCM和单片WSI闪速PSD器件之间的主要差别

特 性	PSD813FH-MCM	PSD813F1-单片	PAD813F2-单片
非易失性主存储器	128K字节闪速-ISP	128K字节闪速-ISP	128K字节闪速-ISP
非易失性另一存储器(引导或其它用途)	32K字节OTP PROM-Non-ISP	32K字节EEPROM-ISP (字节擦除/写)	32K字节闪速-ISP(区 段擦除/字节写)
SRAM	512字节-Batt BU (电池备用)	2K字节-Batt BU (电 池备用)	2K字节-Batt BU (电 池备用)
JTAG ISP	No	Yes(存储器, PLD, 配 置)	Yes(存储器, PLD, 配 置)
页寄存器	4位	8位	8位
输出MicroCells(宏单元)	12 (7个控制闪速存储器)	16	16
输入MicroCells(宏单元)	23 (7个不能作通用)	24	24
同时存储器映射功能	放置保持器方案(place holder scheme)。PLD, 页寄 存器	Yes-PLD, 页寄存器, 运行时间(Run-time) 寄存器	Yes-PLD, 页寄存器, 运行时间(Run-time) 寄存器

注释：其它三种单片PSD813F器件将可供使用：

- * PSD813F3 (无另一非易失性存储器但有主闪速存储器和SRAM)
- * PSD813F4 (除了无SRAM外, 与F2相同)
- * PSD813F5 (除了无SRAM外, 与F3相同)

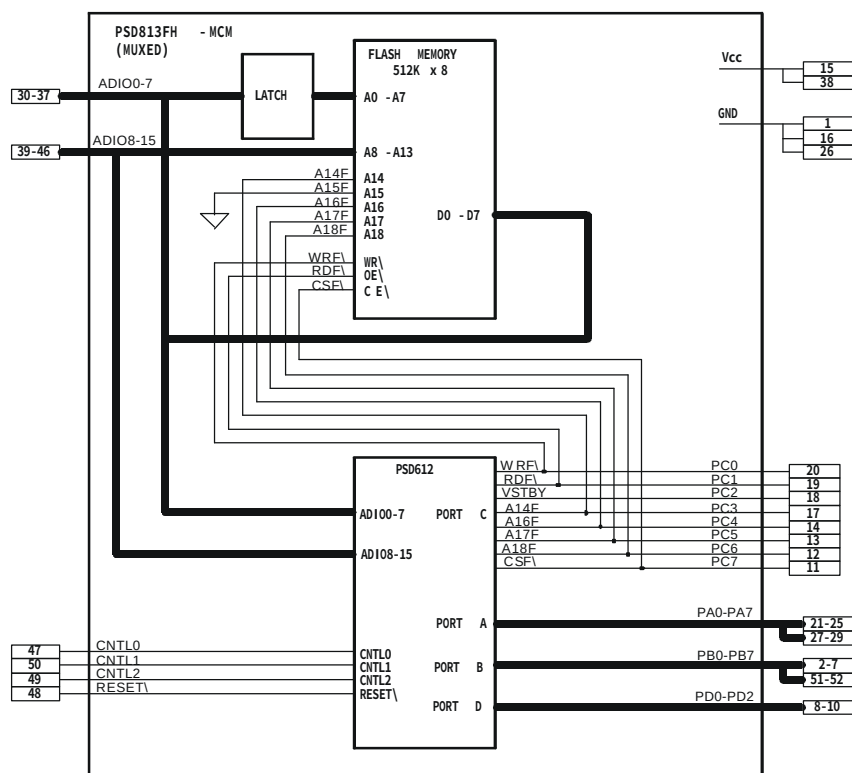


图1 PSD813FH方框图

MCU总线接口

MCM保持类似于PSD6xx的“无缝连接逻辑(No Glue Logic)”MCU接口。MCM支持宽范围的有多路复用和非多路复用数据总线的8位微控制器。虽然本文把重点放在有多路复用数据总线的MCU,但是另一种MCM器件PSD813FN被设计为有非多路复用数据总线的MCU。图2表示非多路复用的MCM配置。注意,地址锁存器被禁止且MCU数据线被连接至PSD6xx管芯的端口A。

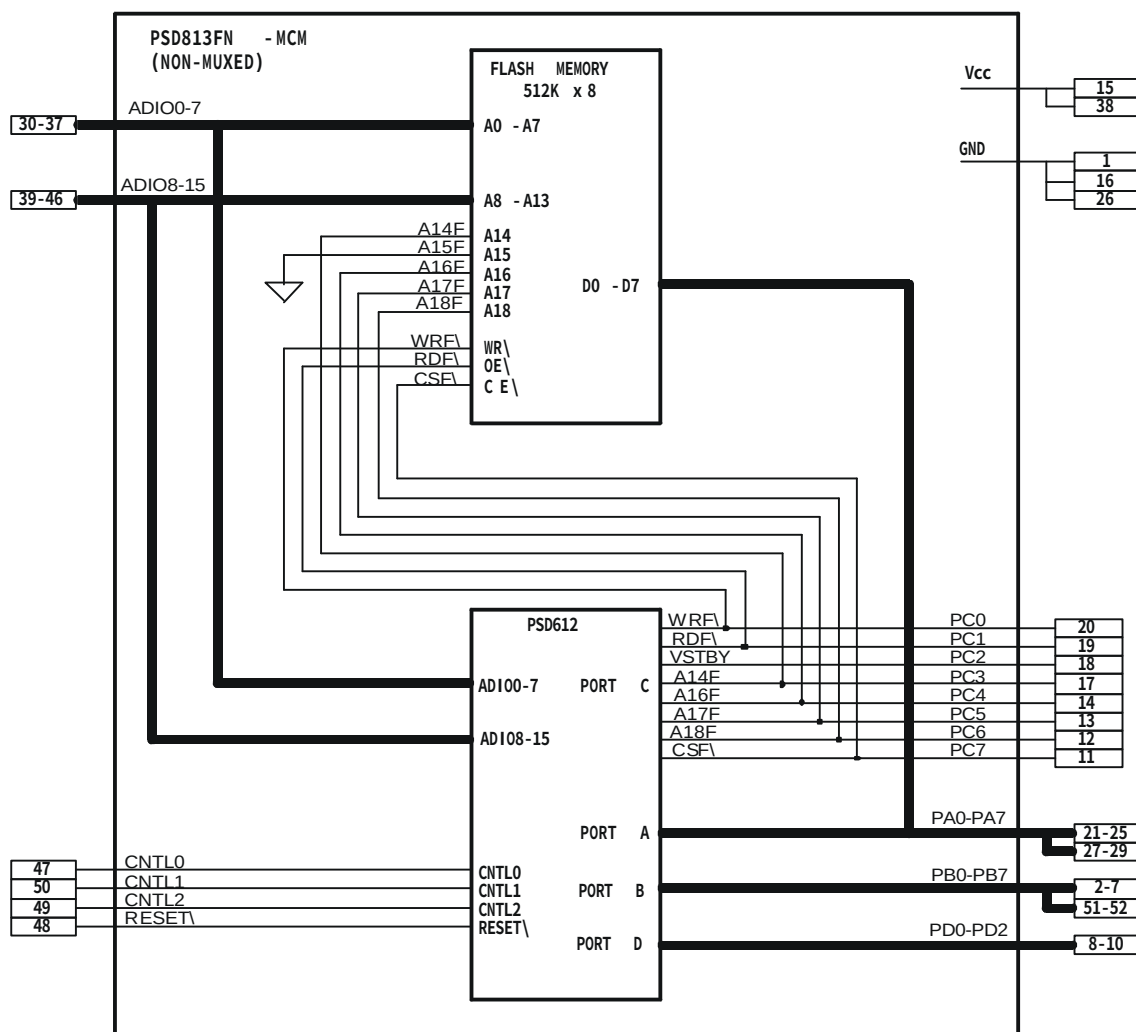


图2 PSD813FN方框图

三、设计举例说明

本设计的目标是使用MCM FLASH（闪速）PSD创建支持ISP基于闪速存储器的嵌入式MCU内核。此设计的方法提供了至单片闪速PSD的方便的升级途径。通过创建支持同时存储器访问的存储器映射结构以及保留某些引脚供JTAG使用，可以达到升级的途径。在ISP期间内，为了在执行来自另一非易失性存储器的代码的同时，写系统闪速存储器，单片器件中同时存储器访问是必需的。同样地，可以在执行来自自主闪速存储器的代码的同时，写另一非易失性存储器。

该系统的电路示于图3。系统的要求列在下面。

- * 128K字节闪速存储器（在系统内可编程，ISP）
 - 通过RS-232可下载固件
 - 利用PSD器件的资源可以非常简单地实现下载期间内存管理
 - 单片器件内升级至固件和芯片配置的JTAG下载
- * 16K字节OTP引导ROM
 - 单片器件内升级至32K字节EEPROM
- * 768字节SRAM（MCU中256字节，PSD中512字节）
 - 单片器件内升级至2.25K字节
- * 支持外围设备和存储器
 - 接口至LCD、协议芯片（Protocol Chip）、ADC、DAC、存储器
 - 16个多路复用（demuxed）地址位

- 8位数据总线
- 4个可编程芯片选择
- * 来自8031的8个I/O
- 定时器输入
- 通用I/O
- * 来自PSD的备用I/O
 - 如所示在此MCM设计中5个备用I/O
 - 采用单片器件多达25个备用I/O
- (如果未使用外围设备支持或JTAG)
- * I/O的特点包括PSD Micro<=>Cell (宏单元) 逻辑
- * UART端口
- * 2个定时器/计数器
- * 2个外部中断
- * 16MHz Xtal(晶振)
- * 所有这些包含在两种封装内 (44引脚和52引脚PLCC)

注意, 如果不需要外部器件, 那么端口A和B的引脚可随意作通用引脚用。在转变到单片PSD器件之后, 端口C的引脚可选择性地用于IEEE 1149.1 JTAG接口。

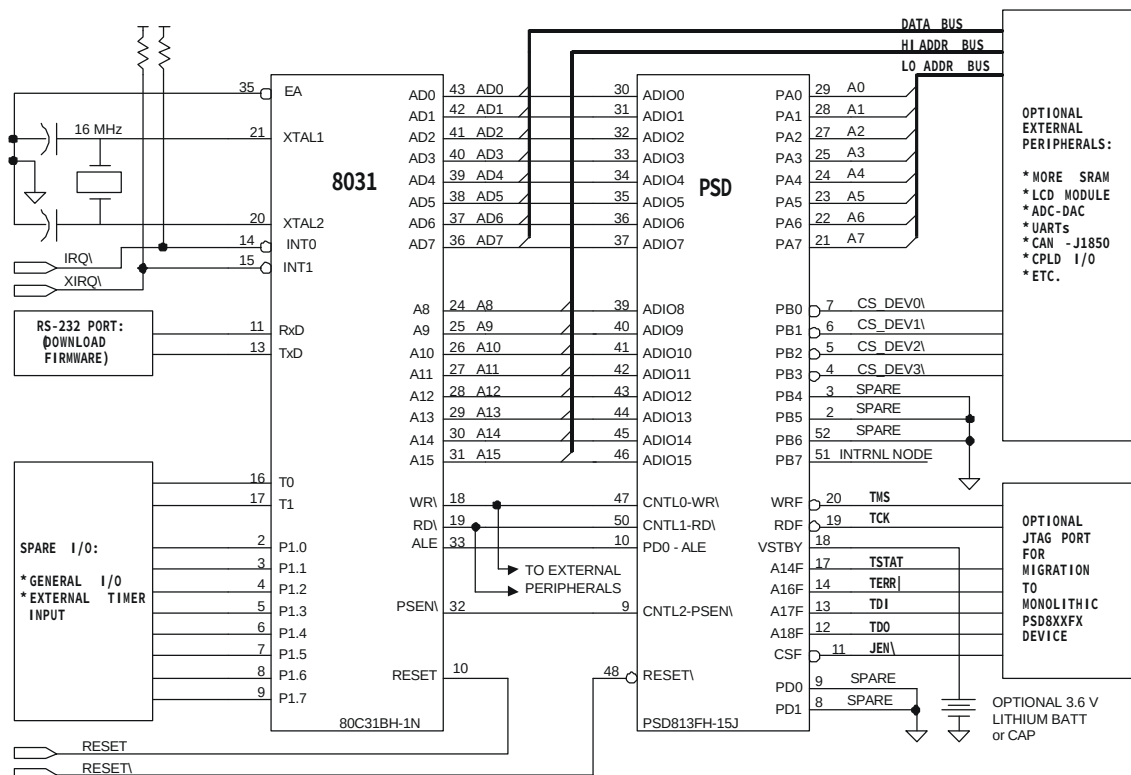


图3 电路图

四、系统存储器映射

系统存储器映射示于图4和图5。标号FSx和CSB00Tx是MCM器件中内部存储器块的名字。FSx=主闪存存储器块 (每个16K字节), CSB00Tx=OTP引导ROM块 (每个8K字节)。

有两种基本的工作方式, 一种是引导/下载方式, 另一种是正常工作。图4代表上电 (引导) 时的存储器映象图。系统将从MCM PSD OTP引导ROM自举, 然后用8031 UART实现主闪存存储器的下载 (如果需要的话)。在这一点上, 所有主闪存存储器被组合在8031“数据”和“程序”空间, 因而它可以被当作数据处理。其次, 图5代表闪存存储器被编程和/或有效化 (validated) 之后的存储器映象图。上电期间系统由此引导的引导ROM现在用包含应用矢量和代码的闪存存储器来代替。两个映象图之间的转换在8031的控制下通过设置PSD

内的“SWAP（交换）”位来实现。在“SWAP”位被设置后，所有主闪存存储器仅保留在8031“程序”空间内，不在“数据”空间中。

在本设计中，由于系统包含多于8031能线性寻址的存储器，所以使用了分页（paging）。MCM内PSD6xx管芯通过使用页寄存器实现分页，页寄存器可由8031访问。因为使用了分页，所以不管MCU从哪一页执行，对于必须是可访问的固件子程序，需要公共存储器区域。此公共区驻留在程序空间每一个存储器页的低半部内（图5）且应当包含处理初始化、中断、实现页切换以及驱动实际器件的子程序。在所有时候保持关键的数据空间项目（item）可用也是重要的。例如，在此设计中，PSD控制寄存器、I/O以及用于堆栈和全局变量的系统SRAM在任何存储器页上可供使用（图4和图5，系统RAM和I/O）。

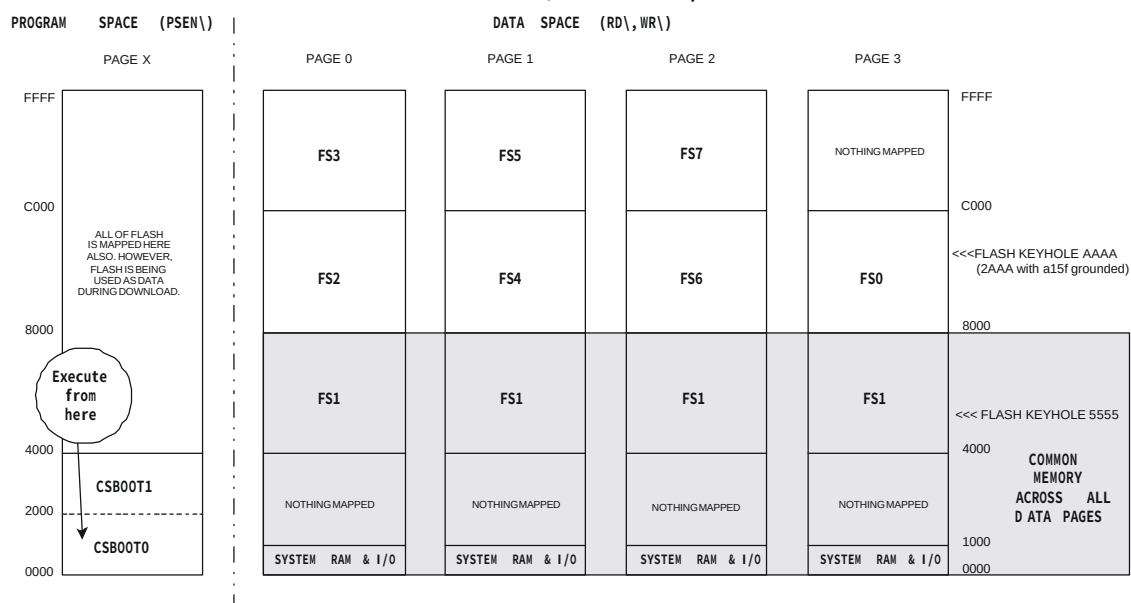


图4 存储器映象-引导/下载方式（SWAP位=0）

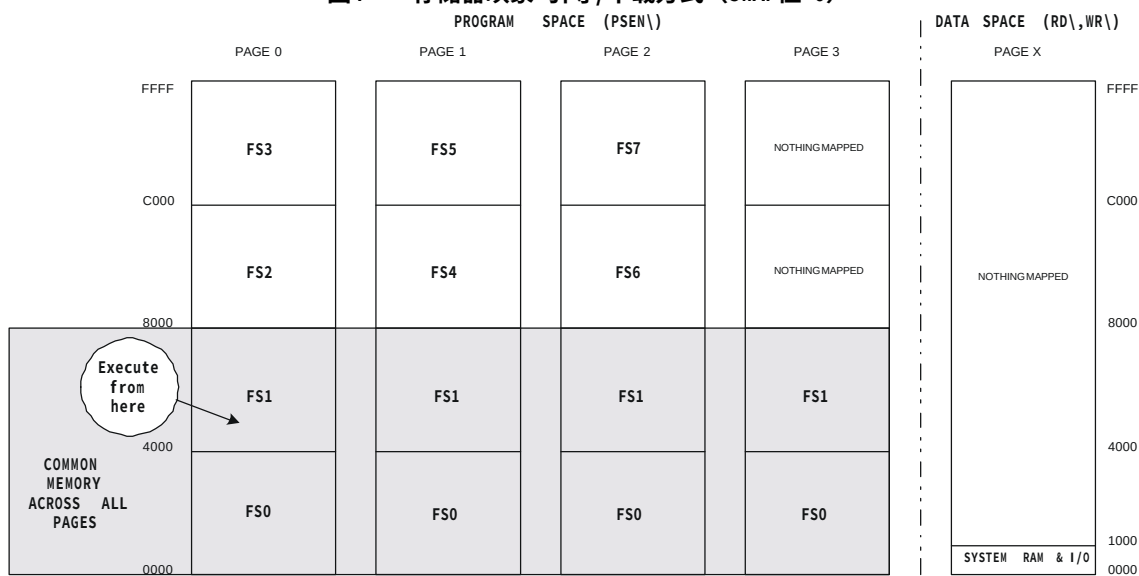


图5 存储器映象-正常工作（SWAP位=1）

以下是自举（boot-up）序列的综述：

1. 在刚好复位之后，ROM段CSB00T0和CSB00T1中的引导代码应当检查以决定闪存存储器是否需要编程

P&S武汉力源电子股份有限公司

地址：湖北武汉市卓刀泉路15号

信箱：武汉市70020信箱

电话：(86) (027) 87493500~87493506

P&S网网址：http://www.p8s.com

邮编：430079

传真：(86) (027) 87491166, 87493493

(通过校验和测试)并检查在UART通道上是否有来自主计算机的下载请求。此时主闪速存储器在组合数据空间和存储空间内(图4)。

2. 如果需要下载,引导代码通过擦除闪速存储器、用来自UART通道的数据对它编程、并用校验和对装载结果进行校验的方法执行下载(图4)。其后,执行从CSB00T0/CSB00T1转移到闪速存储器段FS1。8031从FS1执行在PSD内设置“SWAP”位,这将移去CSB00T0/CSB00T1并代之闪速存储器段FS0(图5)。设置“SWAP”位也把全部主闪速存储器移至程序空间。正常应用代码现在有效。

3. 如果没有下载要求(闪速存储器校验和为0K,无主机下载要求),那么执行将从CSB00T0/CSB00T1跳到FS1。8031从FS1开始,设置“SWAP”位,它把主闪速存储器置于程序空间内,并开始正常工作。

注释1:一旦“SWAP”位被设置,8031将访问跨越三个存储器页的128K字节的闪速存储器(见图5)。

注释2:PSD上电伴随页寄存器=0。

注释3:当此MCM设计转为单片闪速PSD设计时,存在另一种下载可能性。单片器件(PSD813F1/F2/F4)也允许引导代码的ISP。请参见本文末尾“转换问题(Conversion Issues)”一节。

现在让我们看看闪速存储器页内分隔的代码(Partitioning code)。由于在本设计中OTP引导ROM只用于自举(boot-up)和ISP,所以MCU最终将从闪速存储器执行。如图5所示,让我们假设在闪速存储器内有128K字节的程序空间。128K字节的代码将驻留在4个区域内,32K字节在公共区域内(FS0、FS1可从任何页访问),32K字节在页0(FS2、FS3),32K字节在页1(FS4、FS5),32K字节在页2(FS6、FS7)。

应当记住。如果8031在执行时从未离开过页0,那么它可以访问64K字节的闪速存储器以及所有的SRAM和I/O。但是,如果根据页0上半部的调用,8031的执行跳到页1或2上的闪速存储器,那么必须注意保留再次返回到页0的路径。如果至页1和2的调用来自页0低半部(公共区)内的子程序,那么从调用返回没有问题。

当把闪速存储器内的代码放置在页0、1或2的上半部时,软件设计师必须把任务分到逻辑组(logical group)。这些组应当不需要频繁地访问其它页上的代码(大多数软件可以用这种方式分开并且是良好的模块化设计的结果)。因为系统SRAM在任何页上可供使用,所以驻留在不同页上的固件子程序可以利用全局变量或堆栈传送数据。设计师可创建页切换算法以便在位于不同页上的任务之间跳转,有许多途径可实现分页方法。一种方法包括使用所有程序任务的地址表和页号(page numbers),这些程序任务可以从页到页被调用。这个表和这些算法必须驻留在闪速存储器内,闪速存储器驻留在公共区内。这提供了非常清楚的分页解决方案,它可以通过使用高级编译器(Keil的编译器直接支持此功能)而实现。使用这种方法的唯一代价是从页至页切换时所需的总开销。

当这种MCM设置转为单片PSD设计时,类似的存储器映射结构可以被保持。用户可以使用PSD813F1器件并用EEPROM代替这种MCM中存在的OTP PROM。32K字节的EEPROM可以分为16K字节的引导存储器和16K字节的数据存储器(用于产品配置、校准等等)。有许多可能性。

除了主闪速存储器的ISP之外,单片PSD器件也支持MCU引导存储器的ISP。用于引导存储器ISP的存储器管理技术非常适合于此MCM设计的存储器映射方案。引导存储器ISP将在有关单片PSD器件的应用笔记中单独讨论。另外,也可参见本文末尾“转换问题”一节。

五、访问MCM闪速存储器管芯

单片闪速PSD器件(PSD813F1/F2/F3/F4/F5)具有128K字节闪速主系统存储器,它们在8个段内可供使用,每个段包含16K字节。标准PSD结构为每个段提供单独的芯片选择方程从而提供更大的灵活性,在需要分页时特别有用。PSD813FH MCM器件通过使用4兆位(512K字节)的闪速存储器管芯和来自PSD6xx管芯的控制信号模拟这种分段。其结果是,MCM器件在8个16K字节段内使128K字节的闪速存储器可供MCU使用,正好和单片器件一样。

如图1所示,闪速存储器管芯的控制信号和某些地址线从PSD6xx管芯的端口C产生。它们列于表2。

P&S武汉力源电子股份有限公司

地址:湖北武汉市卓刀泉路15号

信箱:武汉市70020信箱

电话:(86)(027)87493500~87493506

P&S网网址: <http://www.p8s.com>

邮编:430079

传真:(86)(027)87491166,87493493

表2 来自PSD6xx管芯的MCM闪速存储器管芯控制信号

控制信号	引脚（端口）分配
WRF\	20(PC0)
RDF\	19(PC1)
A14F	17(PC3)
A16F	14(PC4)
A17F	13(PC5)
A18F	12(PC6)
CSF\	11(PC7)

注意：在图1中到闪速存储器管芯的地址线A15F始终接地。这允许送到闪速存储器管芯的三根高地址线（A18F、A17F、A16F）把闪速存储器分为8个段，每个段包含16K字节，正象单片闪速PSD器件一样。送至闪速存储器管芯的低14根地址线（A13-A0）对特定16K字节段的每一个字节寻址。

A14的特别处理需要某些说明。如MCM PSD813FN/FH数据手册所述，对闪速存储器管芯的某些访问（擦除、写入、校验等等）需要调用闪速存储器管芯内部嵌入的算法。当MCU写特定命令至闪速存储器管芯上“关键单元（keyhole）”地址0x2AAA和0x5555时，这些算法被调用（注意，由于A15F接地，混用（aliased）地址0xAAAA和0xD5555也可访问嵌入算法）。这些地址要求在闪速存储器管芯上地址线A14F有效。因此，在MCU访问内部闪速存储器算法时，PSD6xx管芯将把MCU地址线A14传送到闪速存储器管芯地址线A14F。当MCU象读ROM一样读闪速存储器管芯时，PSD6xx管芯将把地址线A14F接地。

注意：单片闪速PSD器件嵌入算法“关键单元（keyhole）”地址驻留在0xX555和0xXAAA从而具有更大的映射灵活性（X=不关心）。

让我们查看设计中用于控制闪速存储器的一些PSDabel方程。完整的PSDabel源文件请参见附录A。

PSD具有4个页寄存器位，pgro..pgr3，它们都可供MCM内部的译码逻辑和可编程逻辑使用。这个页寄存器可以由8031访问且在运行时间（run-time）期间内可以被改变。在本设计中因为只使用4个存储器页，所以只需要两个页位（page bit），另外两位留作其它用途。

以下说明页位是怎样被使用的：

page=[pgr1, pgr0]； “PSD页定义

A14F这样被实现：

pass_a14=pgr2； “PSD页寄存器位用于控制闪速存储器算法访问
a14f=a14 & pass_a14；

在访问闪速存储器算法之前，信号“pass_a14”由MCU设置。在闪速存储器被当作ROM使用之前，该位被MCU清除。

“SWAP”位用下述方法实现：

swap=pgr3； “PSD页寄存器用于改变系统存储器映射

上电或PSD复位时此位的缺省状态为零。当此位为零时，系统存储器如图4所示那样安排，供自举和下载（如果需要的话）用。其后，8031将设置SWAP=1以便把系统存储器映象转为如图5所示，供正常工作之用。

如前所述，在此设计中，PSD管芯通过创建用于高地址线和主芯片选择的信号控制闪速存储器管芯。但是，单片闪速存储器PSD用8个内部芯片选择控制其主闪速存储器，这8个芯片选择信号是fs7…fs0。即使MCM不把这些芯片选择用于硅片，仍需为fs7…fs0书写方程。PSDsoft Address Translator(地址翻译器)实用程序（在后面说明）将使用这些有关fs7…fs0的方程，使MCU固件文件的内容传递到PSD闪速存储器地址。以下是用于此MCM设计的方程：

```
fs0 = ((address>=^h8000) & (address<=^hBFFF) & (page==3)&!swap)
      # ((address>=^h0000) & (address<=^h3FFF) & (page == X)&swap) ;
fs1 = (address>=^h4000) & (address<=^h7FFF) & (page == X) ;
```

```

fs2 = (address>=^h8000) & (address<=^hBFFF) & (page==0) ;
fs3 = (address>=^hC000) & (address<=^hFFFF) & (page==0) ;
fs4 = (address>=^h8000) & (address<=^hBFFF) & (page==1) ;
fs5 = (address>=^hC000) & (address<=^hFFFF) & (page==1) ;
fs6 = (address>=^h8000) & (address<=^hBFFF) & (page==2) ;
fs7 = (address>=^hC000) & (address<=^hFFFF) & (page==2) ;

```

注意：当把MCM设计转为单片闪速PSD设计时，这些有关主闪速存储器的方程（fs7…fs0）将被使用，不需要有关a18f、a17f、a16f、a14f以及fcs\的方程。

现在来观察PSD6xx管芯是怎样产生高地址线（a18f、a17f、a16f）的。表3表示实行所需的真值表。请参见图4和图5中存储器映象图。

表3 闪速存储器管芯段真值表

A18F	A17F	A16F	闪速存储器段 芯片选择	8031地址范围（十六进制）	PSD页 寄存器	SWAP位
0	0	0	fs0	C000 FFFF或0000 3FFF	3或X	0或1
0	0	1	fs1	4000 7FFF	X	X
0	1	0	fs2	8000 BFFF	0	X
0	1	1	fs3	C000 FFFF	0	X
1	0	0	fs4	8000 BFFF	1	X
1	0	1	fs5	C000 FFFF	1	X
1	1	0	fs6	8000 BFFF	2	X
1	1	1	fs7	C000 FFFF	2	X

以下是实现表3的MCM PSDabel方程。

```

a16f = (address>=^h4000) & (address<=^h7FFF) & (page= =X) "fs1
# (address>=^hC000) & (address<=^hFFFF) & (page= =0)"fs3
# (address>=^hC000) & (address<=^hFFFF) & (page= =1)"fs5
# (address>=^hC000) & (address<=^hFFFF) & (page= =2)"fs7
a17f = (address>=^h8000) & (address<=^hBFFF) & (page= =0) "fs2
# (address>=^hC000) & (address<=^hFFFF) & (page= =0)"fs3
# (address>=^h8000) & (address<=^hBFFF) & (page= =2)"fs6
# (address>=^hC000) & (address<=^hFFFF) & (page= =2)"fs7
a18f = (address>=^h8000) & (address<=^hBFFF) & (page= =1) "fs4
# (address>=^hC000) & (address<=^hFFFF) & (page= =1)"fs5
# (address>=^h8000) & (address<=^hBFFF) & (page= =2)"fs6
# (address>=^hC000) & (address<=^hFFFF) & (page= =2)"fs7

```

其次，闪速存储器管芯芯片选择信号（CSF\）用下述方法实现：

```

!csf = (address>=^h8000) & (address<=^hBFFF) & (page= =3)&! swap "fs0
# (address>=^h0000) & (address<=^h3FFF) & (page= =X)&swap "fs0
# (address>=^h4000) & (address<=^h7FFF) & (page= =X) "fs1
# (address>=^h8000) & (address<=^hBFFF) & (page= =0) "fs2
# (address>=^hC000) & (address<=^hFFFF) & (page= =0) "fs3
# (address>=^h8000) & (address<=^hBFFF) & (page= =1) "fs4
# (address>=^hC000) & (address<=^hFFFF) & (page= =1) "fs5
# (address>=^h8000) & (address<=^hBFFF) & (page= =2) "fs6
# (address>=^hC000) & (address<=^hFFFF) & (page= =2) "fs7

```

闪速存储器嵌入算法“关键单元（keyhole）”地址0x2AAA和0x5555必须被包括在CSF\有效范围内。因为闪速存储器管芯地址线a15F永久接地，所以在混用（aliased）地址0xAAAA和0xD555“关键单元”地址也分别有效。

最后，闪速存储器管芯读和写信号由PSD6xx管芯使用这些方程来产生。注意，在组合空间和程序空间之

间移动主闪速存储器时，SWAP位是有帮助的。

```
! wrf=(!wr) ;           "闪速存储器写输入
! rdf=((!rd#!psen) &!swap "为下载把闪速存储器放置在组合空间
#( ! psen&swap) ;       "为正常工作把闪速存储器放置在程序空间
```

六、访问OTP引导ROM

在PSD6xx管芯上MCM包含32K字节的OTP ROM，它们可用于引导、ISP下载以及闪速存储器访问子程序。这个OTP ROM被分为4个8K字节的段，每一个段具有它自己的芯片选择方程。对于此设计，为了使8031可用的主闪速存储器数量为最大，只使用了两个段（总共16K字节）。由于单片闪速PSD比MCM闪速PSD器件具有更为灵活的存储器映射能力，所以这种折衷方案对单片闪速PSD没有影响。以下是PSDabel方程：

```
csboot0 = ((address>=^h0000) & (address <=^h1FFF) & (page= =X)&!swap) ;
csboot1 = ((address>=^h2000) & (address<=^h3FFF) & (page= =X)&!swap) ;
```

当此MCM设计被移植到单片闪速PSD器件时，MCM OTP引导ROM将变为替代的ISP非易失性存储器。单片PSD813F1器件将提供32K字节的EEPROM，PSD813F2/F4将提供32K字节的闪速存储器。这允许从主机更新引导代码。引导存储器的ISP被包括在单独的有关单片PSD器件的应用笔记中。也可参见本文末尾“转换问题(Conversion Issues)”一节。

七、访问PSD SRAM

虽然此MCM器件内SRAM为512字节，但是因为单片闪速PSD（除了PSD813F4/F5之外）具有2K字节，所以这里2K字节的地址空间被译码。被译码的SRAM空间开始于紧接在8031内部RAM末尾之后的地址且在任何页上均有效。

```
rs0_node = (address>=^h0100) & (address<=^h08FF) & (page= =X) ;
rs0 = rs0_node ;
```

注释：此地址范围包括rs0方程中的反馈项。由于MCM的内部结构，rs0的解必须包括连接到端口引脚的结点。此结点rs0_node被声明在端口B的引脚pb7。当此MCM设计被转换为单片闪速PSD时，此中间结点不需要。rs0可直接使用，而pb7可供I/O使用。

八、访问I/O

以下是本设计与I/O有关的PSDabel方程。在任何存储器页上所有I/O均有效。

PSD内部控制空间（256字节）

```
csiop = (address>=^h0900) & (address<=^h09FF) & (page= =X) ;
```

通用外部芯片选择。这些可用于外围设备或存储器（LCD模块，ADC，DAC，UART，CAN，更多的SRAM等等）。

```
cs-dev0 = (address>=^h0A00) & (address<=^h0AFF) & (page= =X) ;      "256字节
cs-dev1 = (address>=^h0B00) & (address<=^h0BFF) & (page= =X) ;      "256字节
cs-dev2 = (address>=^h0C00) & (address<=^h0DFF) & (page= =X) ;      "256字节
cs-dev3 = (address>=^h0E00) & (address<=^h0FFF) & (page= =X) ;      "512字节
```

在此设计中，端口A用于“锁存地址输出”方式。这意味着端口A引脚对外部外围设备或存储器提供8个去多路复用的低位地址线。如果外部外围设备或存储器不需要这些芯片选择，那么与这些信号有关的引脚以及整个端口A可用作通用I/O。

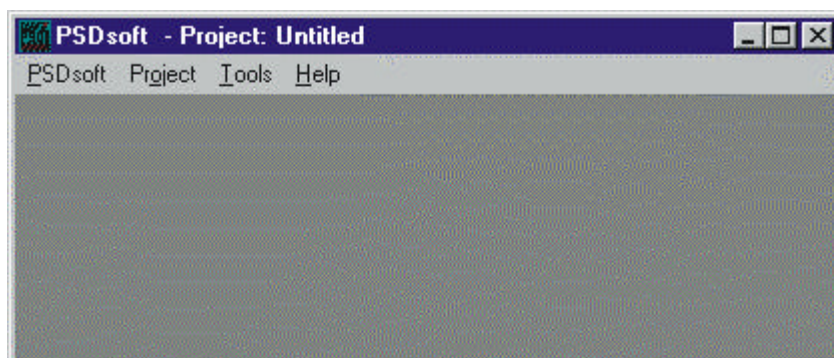
九、使用PSDsoft设计工具的MCM配置

PSDsoft是一套在Windows环境下运行的软件工具，它可用于：

- * 执行设计输入
- * 产生设计的最小化逻辑实现

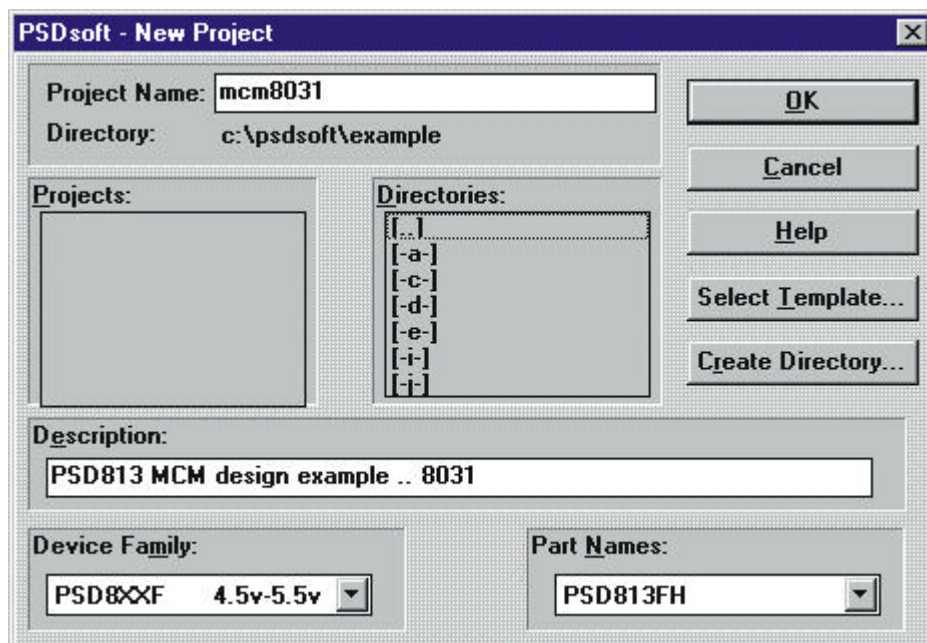
- * 配置PSD器件的选项
- * 在PSD器件内装配设计
- * 把PSD器件的配置与MCU固件相融合
- * 编程PSD器件

让我们依次实现上述各项。首先，通过单击其图标调用PSDsoft (v4.2x)，你可以看到如下图所示的屏幕：

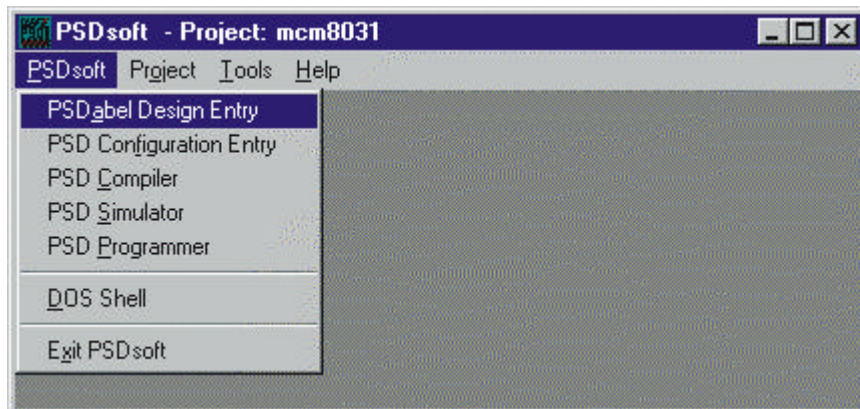


现在进入如下图所示的Project(项目)窗口并选择‘New Project(新项目)’。对于此设计，如图所示令项目名为‘mcm8031’。使用指向保存此设计的所需目录的路径，填充说明域 (description field)，如图所示选择正确的器件 (device) 和元件 (part) 名，最后单击‘Select Template(选择模板)’并选择‘8031 General(8031通用)’模板。单击OK以开始设计输入 (design entry)。

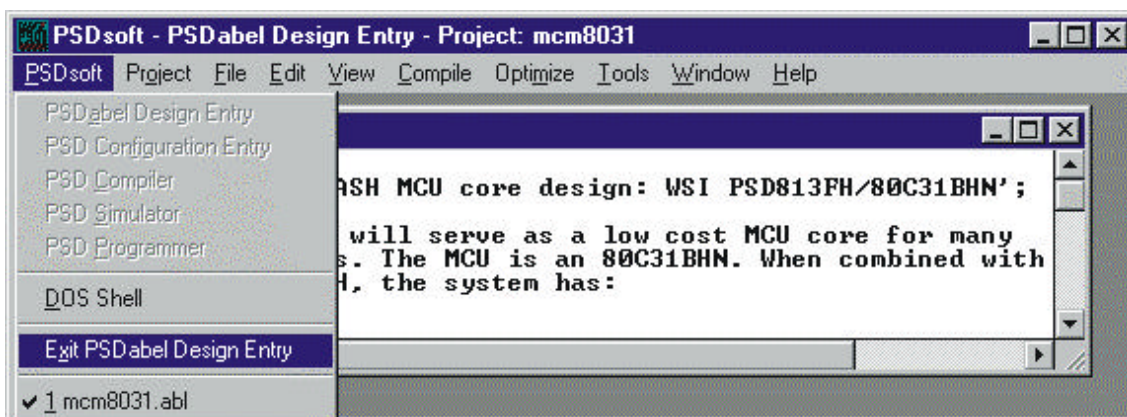
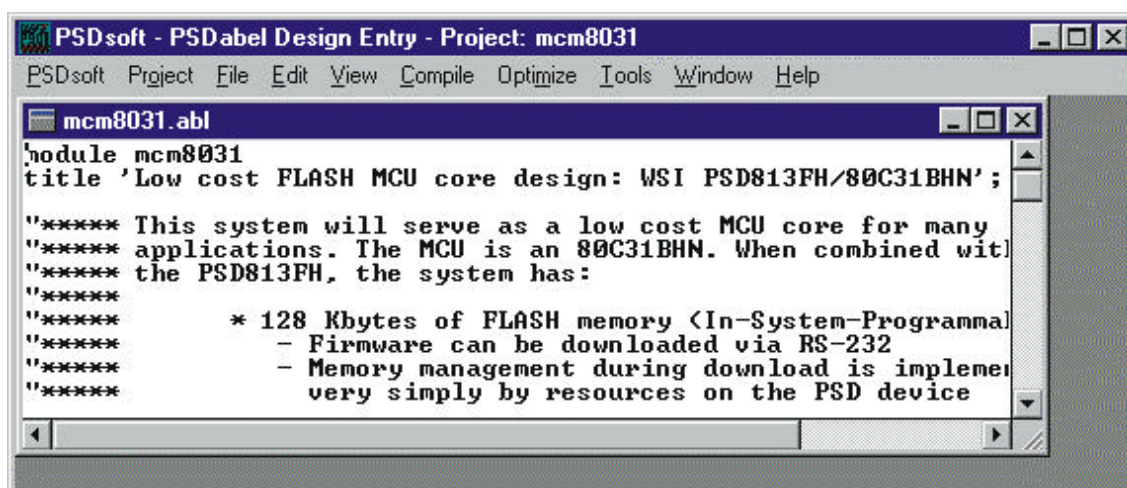
为了输入设计，通过调用如下图所示的编辑器对模板进行编辑。



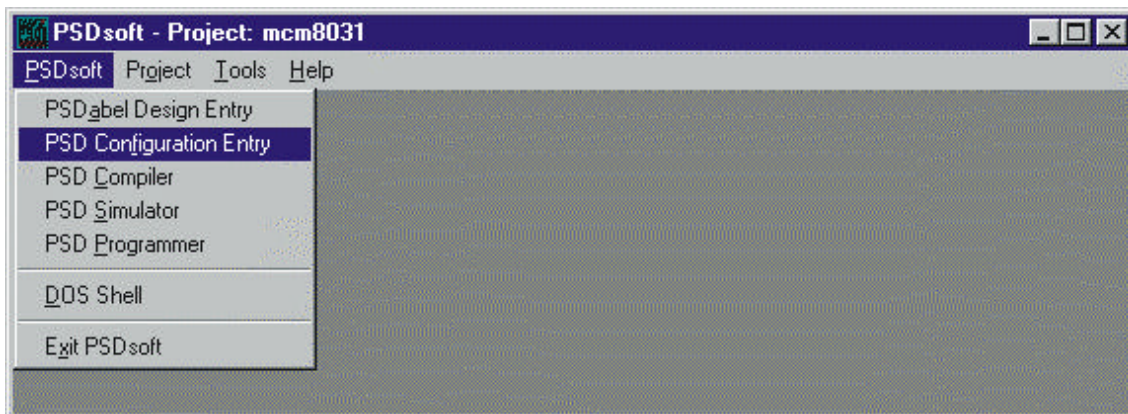
你可以编辑模板文件以便与附录A中mcm8031.ab1文件相符，或者你可以在apphelp@wsiusa.com地址与WSI公司联系以得到如何下载文件的细节。下图为设计输入编辑器的屏幕显示。



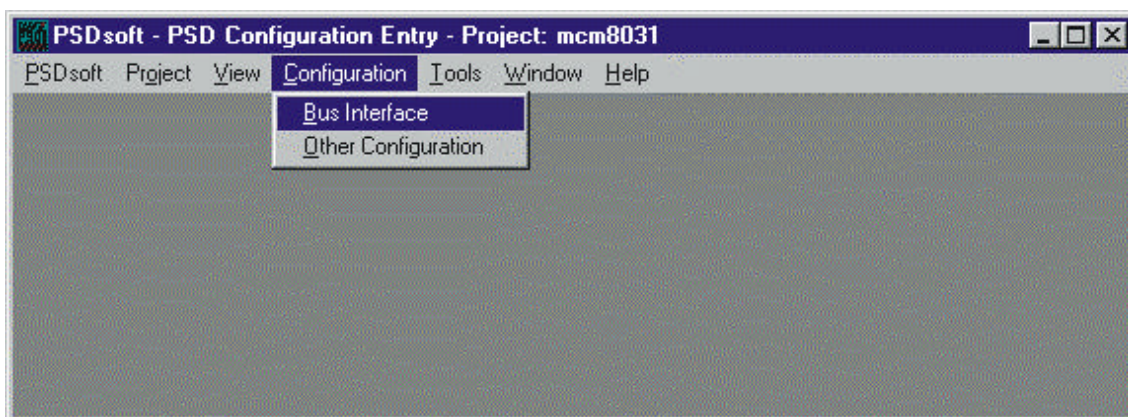
在编辑完成之后单击‘Compile(编译)’，然后再次单击子菜单上的‘Compile’。在编译成功之后，如下图所示，退出此设计输入实用程序



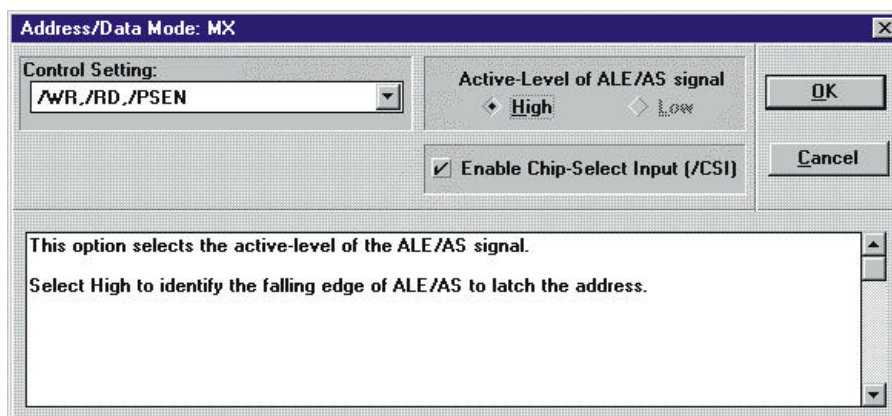
现在，如下图所示，进入Configuration(配置)菜单。



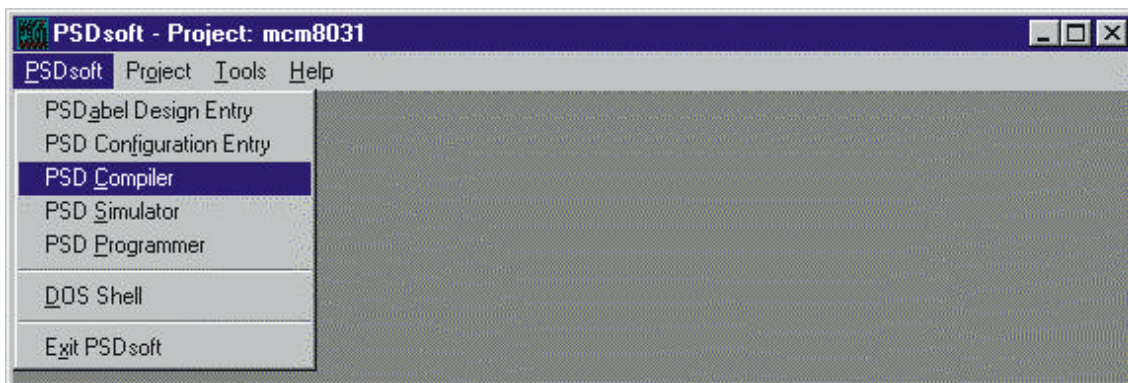
如图所示，选择‘Bus Interface(总线接口)’。



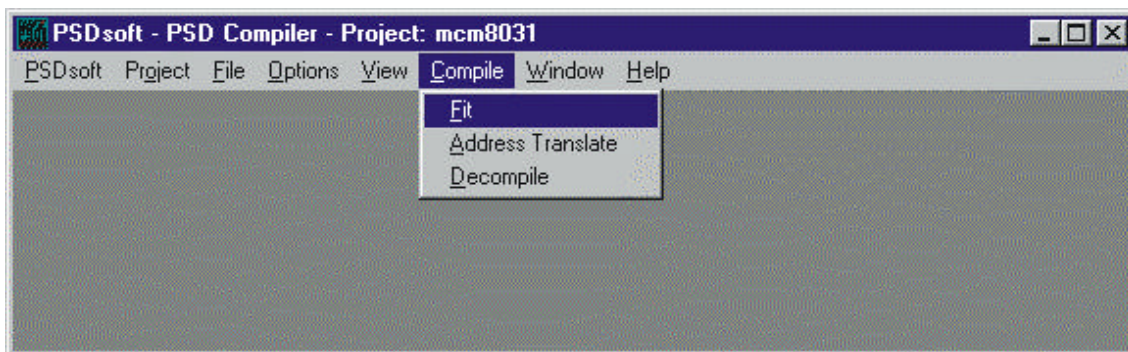
如下图所示进行选择以便为多路复用8031 MCU配置接口



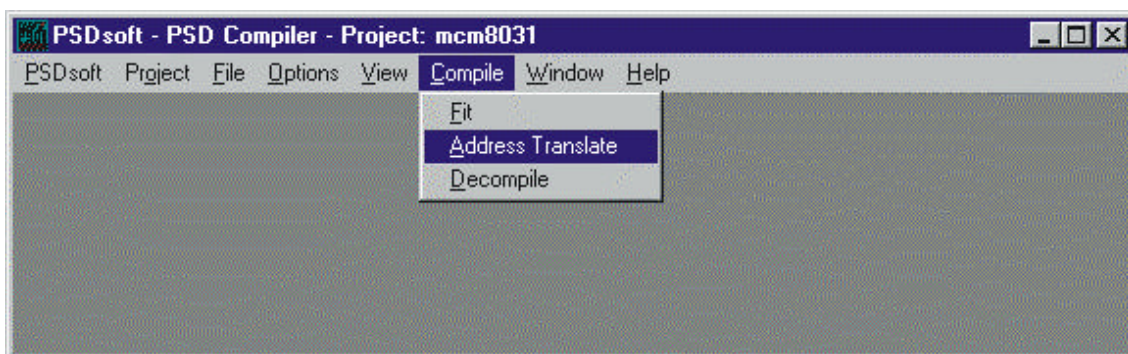
现在退出配置阶段进入编译和装配设计的第二阶段。



如下图所示，调用装配（fitting）实用程序。



装配成功之后，使用‘Address Translator(地址翻译器)’把你的MCU编译器/连接器输出与PSD的配置和逻辑一起组合到一个文件中，此文件可以被编程到PSD器件内。这个文件是mcm8031.obj。调用Address Translator(地址翻译器)如下图所示。



现在，在适当位置键入你的MCU编译器/连接器输出的文件名。在此例中，使用两个文件名，第一个文件text_mcm.hex是驻留在闪速存储器页0、1、2和公共区的系统代码。第二个文件boot_mcm.hex是驻留在OTP引导ROM区用以处理自举、RS-232协议以及闪速存储器访问的代码。

因为驻留在闪速存储器内的代码是分页的，所以你必须把分页逻辑包含在‘File Address Start(文件地址起始)’和‘File Address Stop(文件地址终止)’域内。例如，因为PSD页1的闪速存储器内的代码可以

驻留在与PSD页2的闪速存储器内的代码相同的物理MCU地址,而你的编译器/连接器的输出不能把这些代码段放置在同样的物理文件地址,所以你必须键入正确的文件地址,该地址反映了你的编译器/连接器输出位于何处。请参见《PSDsoft用户手册》。

注释1. 包含系统代码文件text_mcm.hex是可选项,这是因为可能要求在系统内(in-system)而不是在MagicPro编程器上把系统代码装入闪速存储器。

注释2. 你可以通过单击合适的选项(choice)来选择你想要输入到此PSD设计中的MCU编译器格式类型(Intel Hex或Motorola S-Record格式)。

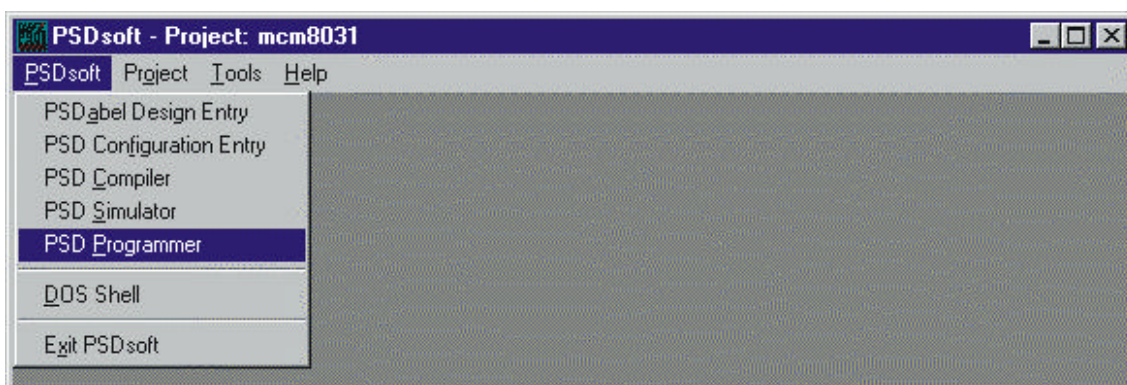
Memory Select Name	Memory Select Equations	File Address Start	File Address Stop	File Name
FS0	FS0 = !a15 & !a14 & !a13 & pgr3 # !a15 & a14 & !a13 & pgr0 & pgr1 & !pgr3;	0000	13FFF	mcm8031.hex
FS1	FS1 = !a15 & !a14 & a13;	4000	7FFF	mcm8031.hex
FS2	FS2 = !a15 & a14 & !a13 & !pgr0 & !pgr1;	8000	BFFE	mcm8031.hex
FS3	FS3 = !a15 & a14 & a13 & !pgr0 & !pgr1;	C000	FFFE	mcm8031.hex

Record Type: ☒ Intel Hex Record ☐ Motorola S-Record

Mapping Mode: ☒ Direct ☐ Relative

OK Cancel Help

在地址翻译(转换)完成的基础上,产生mcm8031.obj文件供MCM编程用。如图所示退出此实用程序并调用编程实用程序。文件mcm8031.obj自动被选择供编程用。



如果需要的话,可参见《PSDsoft用户手册》以便配置MagicPro编程器。把空白MCM放在适配器上并如下图所示对它编程。

P&S武汉力源电子股份有限公司

地址:湖北武汉市卓刀泉路15号

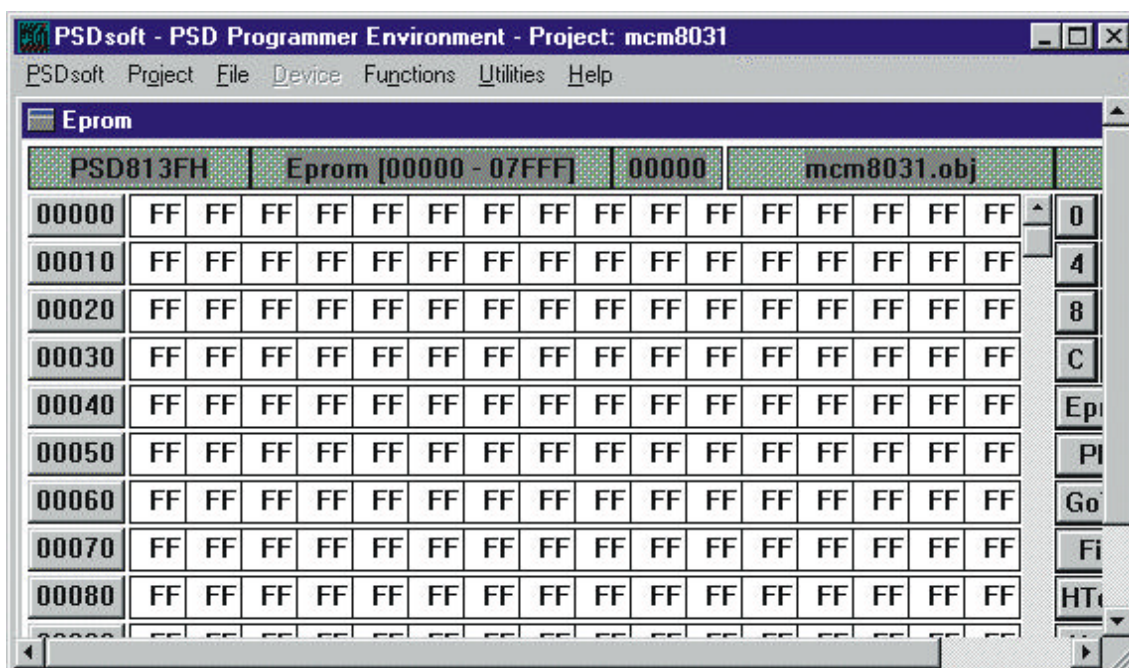
信箱:武汉市70020信箱

电话:(86) (027) 87493500~87493506

P&S网网址: <http://www.p8s.com>

邮编:430079

传真:(86) (027) 87491166, 87493493



十、MCU固件文件格式

因为可以在MCU控制下在系统内（in-system）对闪速存储器编程，所以你可以选择从PSDsoft产生的mcm8031.obj文件或从MCU编译器/连接器产生的mcm_test.hex文件提供可下载的固件。如果你使用mcm8031.obj文件，那么它的地址结构如表4所示

表4 *.obj文件结构

块	起始地址	终止地址
EPROM块0 (CSB00T0)	0x00000	0x01FFF
EPROM块1 (CSB00T1)	0x02000	0x03FFF
EPROM块2 (CSB00T2)	0x04000	0x05FFF
EPROM块3 (CSB00T3)	0x06000	0x07FFF
PLD熔丝图 (fusemap)	0x40000	0x40AEF
ACR配置	0x41000	0x4104F
闪速存储器块0 (FCS0)	0x50000	0x53FFF
闪速存储器块1 (FCS1)	0x54000	0x57FFF
闪速存储器块2 (FCS2)	0x58000	0x5BFFF
闪速存储器块3 (FCS3)	0x5C000	0x61FFF
闪速存储器块4 (FCS4)	0x62000	0x65FFF
闪速存储器块5 (FCS5)	0x66000	0x69FFF
闪速存储器块6 (FCS6)	0x6A000	0x6DFFF
闪速存储器块7 (FCS7)	0x6E000	0x71FFF

起始/停止地址是*.obj文件的地址，与系统中实际的EPROM或闪速存储器地址或*.ab1文件中定义地址无关。使用位于0x50000和0x71FFF之间的部分*.obj文件，放到PSD的闪速存储器中。

如果你选择使用MCU编译器/连接器产生的mcm_test.hex文件，有关其文件结构请参见编译器/连接器文档。

十一、转换问题——MCM至单片闪速PSD

当这个MCM设计转为单片闪速PSD器件时，有三个主要修改区域。

P&S武汉力源电子股份有限公司

地址：湖北武汉市卓刀泉路15号

信箱：武汉市70020信箱

电话：(86) (027) 87493500~87493506

P&S网网址：http://www.p8s.com

邮编：430079

传真：(86) (027) 87491166, 87493493

1. 将具有可重新编程的NVM引导区（PSD813F1的EEPROM或PSD813F2/F4的闪存存储器）代替OTP引导ROM存储器。你可以从引导区域执行同时擦除/写主闪存存储器，反之亦然。这允许引导代码的ISP更新并如图6-9所示改变此MCM例子的存储器映像图。

2. 端口C引脚可以有选择地具有IEEE 1149.1 JTAG功能。这将允许通过JTAG串行连接对整个PSD编程。PSD可以完全空白且仍用此方法编程。可编程区域包括所有存储器、PLD以及PSD配置。引脚是多功能的，当JTAG通道未使用时可用作正常I/O。这些JTAG特性不影响此MCM设计的存储器映像。WSI公司可提供低成本的JTAG电缆、FlashLink，以便实现从PC或并行端口对闪存PSD编程。

3. 可用的输出宏单元（Micro $\leftrightarrow$ Cells）从5个变至16个。这不影响此MCM设计的存储器映像。

让我们看看上述第1条是怎样装入系统存储器映像的。假设需要以PC作为主机通过RS-232通道（代替JTAG）把固件下载到这个嵌入式系统。这些下载动作最初可对主闪存存储器编程，在主闪存存储器一旦被编程之后可对其进行更新，或在利用器件编程器或JTAG连接首次对引导代码编程之后把它更新。

有6种有效自举的情况，上电（复位）时它们必须由系统加以处理。上电时的缺省状态使主闪存存储器在数据空间内，EEPROM在程序空间内。请参见图6-9的存储器映像图。

a. RS-232电缆未连接，主闪存存储器有效。

动作——系统从ees0/ees1引导，求闪存存储器校验和，检查UART有无挂起的主闪存存储器的主机下载要求（图6）；设置PSD的位以便把主闪存存储器放入程序空间（图7）；设置PSD的位，交换ees0/ees1和FS0（图8）；设置PSD的位以便把EEPROM放入数据空间（图9）。现在系统处于正常工作方式。接着8031检查UART有无引导存储器的主机下载请求，然后从主闪存存储器开始应用程序的执行。

b. RS-232电缆已连接，主闪存存储器有效，没有来自主机的下载要求。

动作——与上面“a”所述相同。

c. RS-232电缆已连接，主闪存存储器有效，主机请求主闪存存储器的d/I。

动作——系统从ees0/ees1引导，求闪存存储器校验和，检查UART有无挂起的主闪存存储器的主机下载请求，用来自UART的数据对主存储器编程，求闪存存储器校验和（图6）；设置PSD的位以便把主闪存存储器放入程序空间（图7）；设置PSD的位，交换ees0/ees1和FS0（图8）；设置PSD的位以便把EEPROM放入数据空间（图9）。现在系统处于正常工作方式。接着8031检查UART有无引导存储器的主机下载请求，然后从主闪存存储器开始应用程序的执行。

d. RS-232电缆未连接，主闪存存储器空白或无效。

动作——系统从ees0/ees1引导，求闪存存储器校验和，检查UART有无挂起的主闪存存储器的主机下载请求，一直等待直至UART通信出现为止（图6）。

e. RS-232电缆已连接，主闪存存储器空白或无效。

动作——系统从ees0/ees1引导，求闪存存储器校验和，检查UART有无挂起的主闪存存储器的主机下载请求，用来自UART的数据对主存储器编程，求闪存存储器校验和（图6）；设置PSD的位以便把主闪存存储器放入程序空间（图7）；设置PSD的位，交换ees0/ees1和FS0（图8）；设置PSD的位以便把EEPROM放入数据空间（图9）。现在系统处于正常工作方式。接着8031检查UART有无引导存储器的主机下载要求，然后从主闪存存储器开始应用程序的执行。

f. RS-232电缆已连接，主闪存存储器有效，系统请求引导存储器的d/I

动作——系统从ees0/ees1引导，求闪存存储器校验和，检查UART有无挂起的主闪存存储器的主机下载请求（图6）；设置PSD的位以便把主闪存存储器放入程序空间（图7）；设置PSD的位，交换ees0/ees1和FS0（图8）；设置PSD的位以便把EEPROM放入数据空间（图9）。现在系统处于正常工作方式。接着8031检查UART有无引导存储器的主机下载请求，设置PSD的位以便把EEPROM引导区解锁，用来自UART的数据对EEPROM引导

存储器编程，求EEPROM引导区的检验和，然后从主闪存存储器开始应用程序的执行。

对于这些主机RS-232下载选项的任一项，假设正常引导（ees0/ees1）区最初是由器件编程器在PSD安装在电路板之前编程或由JTAG接口在PSD为在线的情况下编程。

转换这个MCM设计包括对PSDabel源文件的一些编辑。以下是使用PSD813F1实现图6-9存储器映象图的一些例子：

删除PASS_A14F定义（页寄存器位pgr2）。

把SWAP位定义变至：

swap=pgr7；(was pgr3)

更新闪存存储器芯片选择方程如下：

```
fs0 = (address>=^h9000) & (address<=^hBFFF) & (page= =X) ;
fs1 = (address>=^h0000) & (address<=^h3FFF) & (page= =0) ;
fs2 = (address>=^h4000) & (address<=^h7FFF) & (page= =0) ;
fs3 = (address>=^h0000) & (address<=^h3FFF) & (page= =1) ;
fs4 = (address>=^h4000) & (address<=^h7FFF) & (page= =1) ;
fs5 = (address>=^h0000) & (address<=^h3FFF) & (page= =2) ;
fs6 = (address>=^h4000) & (address<=^h7FFF) & (page= =2) ;
fs7 = ((address>=^h4000) & (address<=^h7FFF) & (page=3)&!swap) # ((address>=^hC000)
      & (address<= ^hFFFF) & (page= =X)&swap) ;
```

用下列方程代替有关“csbootx”的方程：

```
ees0 = ((address>=^hC000) & (address<=^hDFFF) & (page= =X) &!swap)
      # ((address>=^h4000) & (address<=^h5FFF) & (page= =3) & swap & unlock) ;
ees1 = ((address>=^hE000) & (address<=^hFFFF) & (page= =X)&!swap)
      # ((address>=^h6000) & (address<=^h7FFF) & (page= =3) & swap & unlock) ;
ees2 = ((address>=^h0000) & (address<=^h1FFF) & (page==3) ;
ees3 = ((address>=^h2000) & (address<=^h3FFF) & (page= =3) ;
```

删除有关a18f、a17f、a16f、a14f、fsc、wrf、rdf以及rs0_node的方程。重新编译设计，使用新的MCU代码执行另一次装配（Fit）和地址翻译（Address Translate），并对闪存PSD编程。

请参见即将出版的有关使用单片闪存PSD和ISP问题的应用笔记。

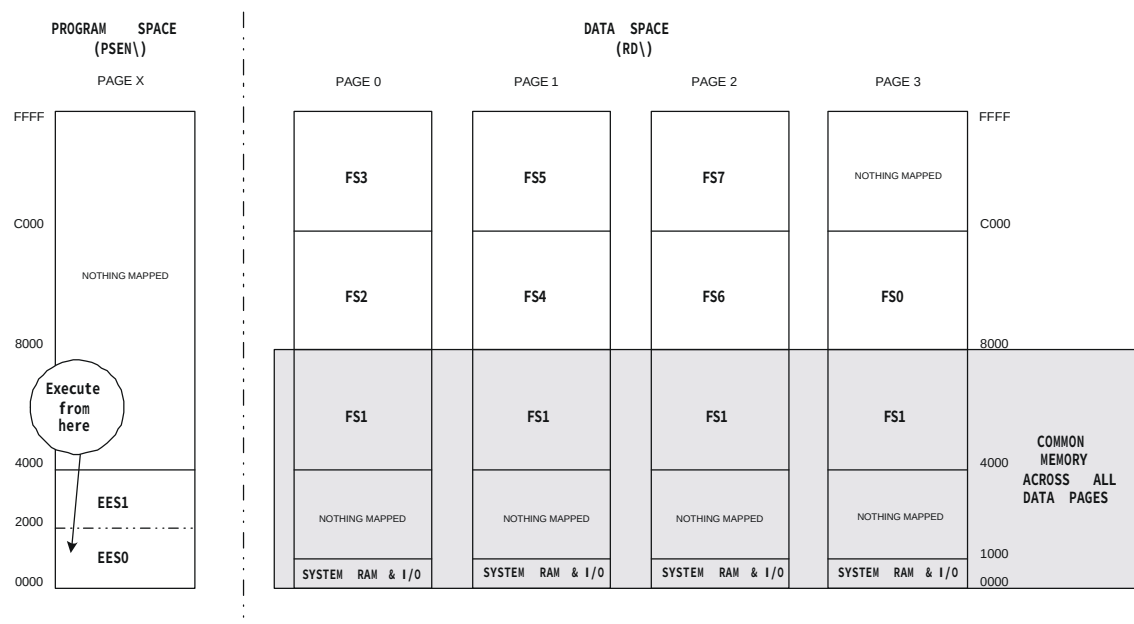


图6 PSD813F1系统存储器映象图，引导/下载

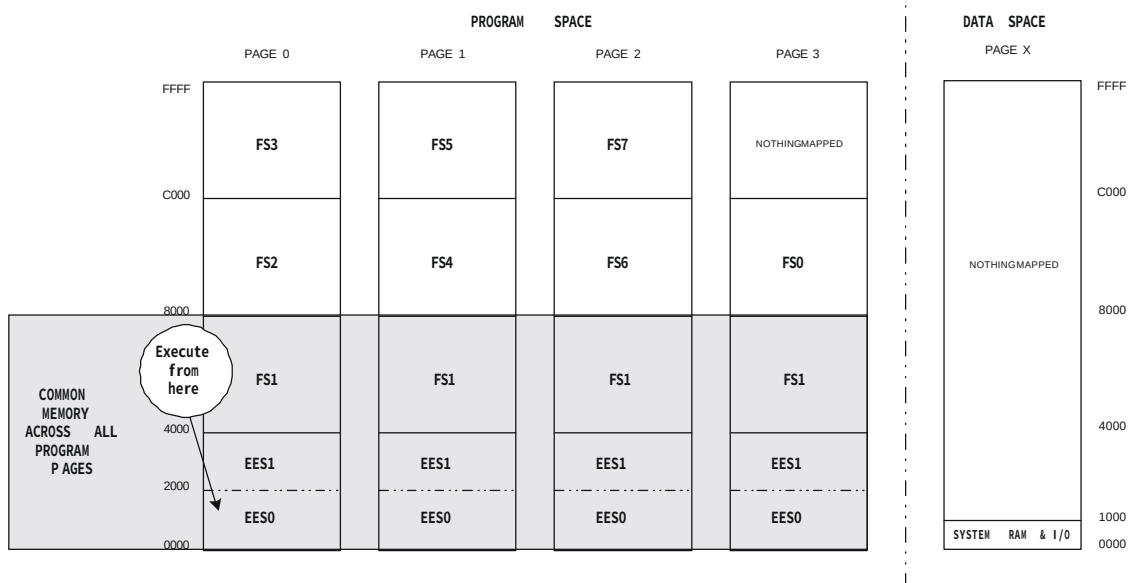


图7 PSD813F1系统存储器映像图，把闪速存储器移至程序空间

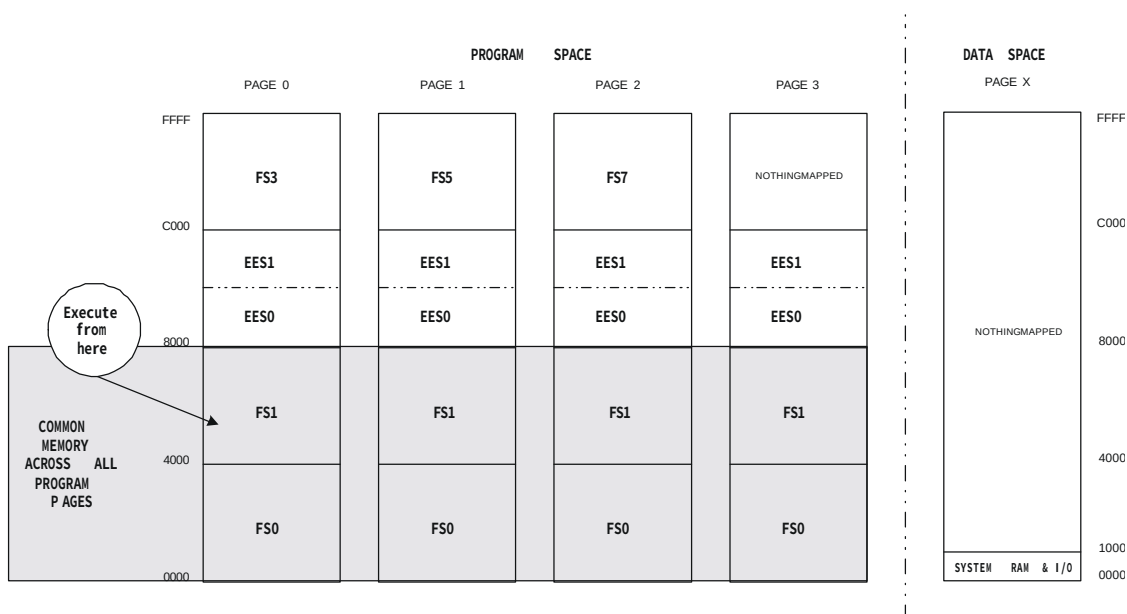


图8 PSD813F1系统存储器映像图，引导EEPROM与闪速存储器段交换

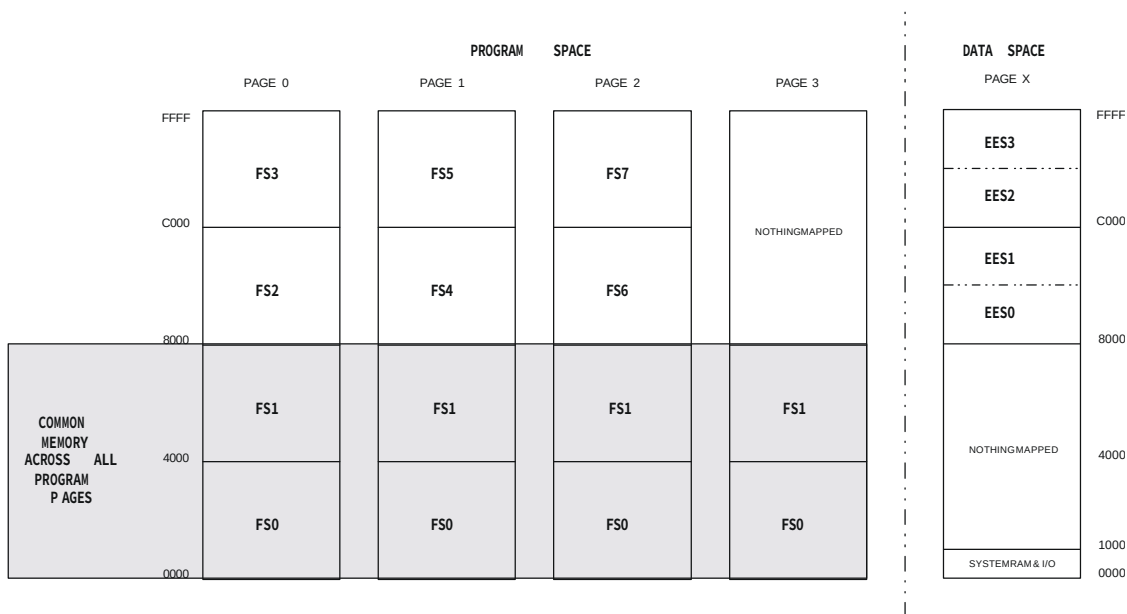


图9 PSD813F1系统存储器映像图，把EEPROM移至数据空间