PSDSOFT EXPRESS用户手册
PSD系列设计软件工具

目　　录

绪论

PSDsoft EXPRESS 是PSD系列器件的设计软件。这个新的设计工具可以让你使用简单的点击（point-and-click）环境方便的设计PSD。

这是PSDsoft的用户手册。下一节解释如何安装PSDsoft EXPRESS。尽管安装看起来是微不足道的过程，但还是强烈建议你要仔细地遵循指令，因为许多问题都是由不正确的安装引起的。第3和第4节介绍了PSDsoft EXPRESS环境，设计流程和菜单。第5节指导你一步步实践完整的设计过程。后面一节解释了核心细节设计过程的每一步骤。附录部分解释了整个手册中使用的一些术语。

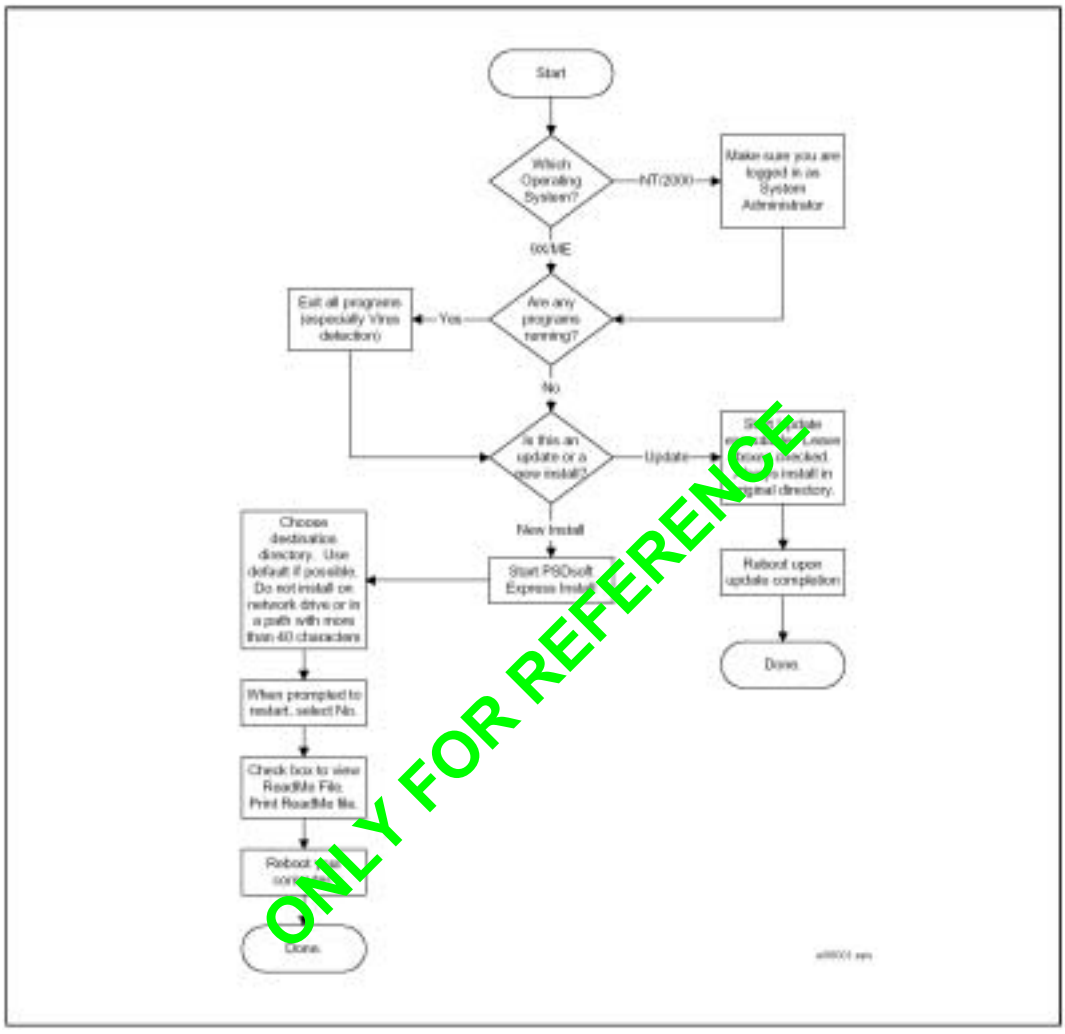
重要注释

第5节中有其他章节中没有的内容，因此所有的用户都应阅读这一节。该用户手册主要是解释软件的功能性方面的问题。因此，你需要下载使用设备的相关数据资料，从而得到一些重要信息。

PSDSOFT EXPRESS 安装

为了更快捷的安装或更新PSDsoftEXPRESS，请使用下面流程表：

图1 安装流程表



下面的章节（sub-sections）包括更详细的信息。

准备你的系统

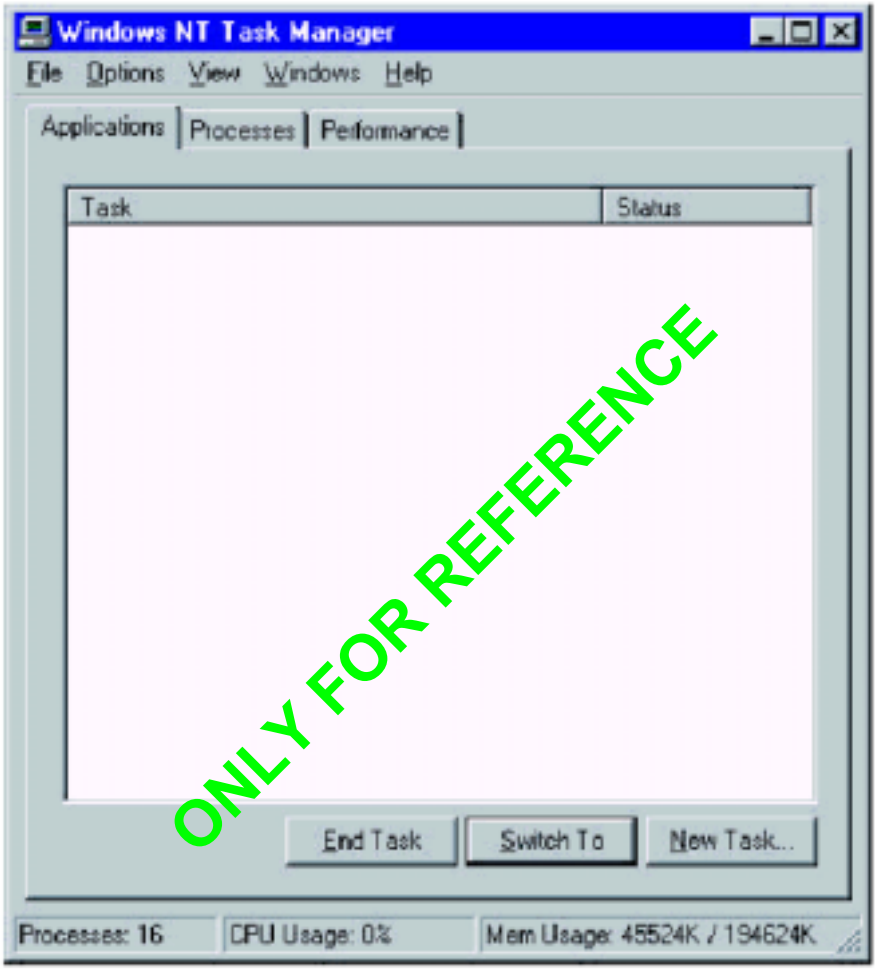
无论你要安装PSDsoftEXPRESS或更新老版本软件，你必须首先为安装或更新而准备你的系统。要做这些，请仔细遵循下面的步骤。如果你不按照这些步骤，你的安装有可能出现不正常。

确定操作系统

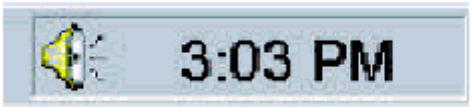
你首先需要确定你正在使用的操作系统。如果你不确定，打开“Explorer”窗口然后到帮助→关于。如果你正在使用的是Windows NT or 2000，你必须注册为系统管理员。

关闭所有的程序

你必须确认在安装软件之前所有的程序（特别是病毒检察）都退出。有两个地方检察运行程序：“任务管理器”和“系统盘”。任务管理器可以通过同时按下CTRL + ALT + DEL键出现。在有些操作系统中，你必须点击**任务管理器...** 按钮。确认“应用”栏已选定。然后选择任何运行程序再点击**结束任务**。在你退出所有的任务后，你的任务管理器应该如下所示：



下一步更重要的是关闭或停止系统盘中的所有项目，它们通常位于你的任务栏右下角。这通常是在后台运行的程序而且常常不出现在任务管理器中。在后台运行的程序可以影响软件的安装。病毒检察程序特别容易产生安装错误。要退出或关闭系统盘中的这些文件，右击每个程序选择恰当的动作，比如“退出”或“关闭”。当结束了所有的程序后，你的系统盘应该基本上已空并且应该像下面：

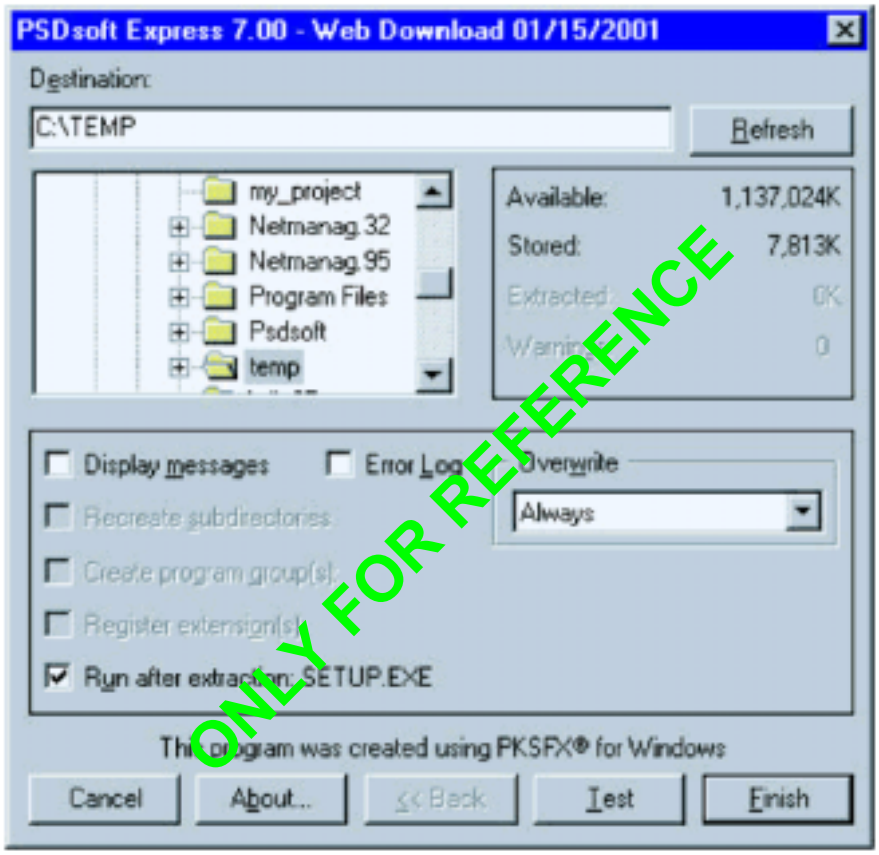


安装 PSDsoftEXPRESS

如果你是第一次安装PSDsoft EXPRESS或你使用过未安装的程序并且希望重新安装该程序，应仔细遵循下列提示。如果你是更新PSDsoft EXPRESS，就跳到“更新”章节。

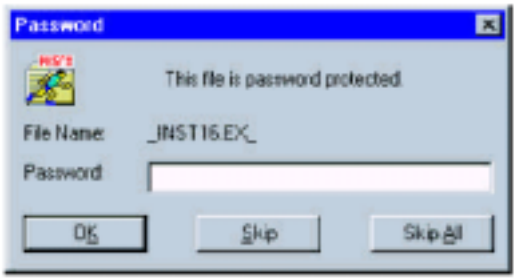
新安装(从下载的文件)

如果你从网站上下载该软件，你首先需要解压出可执行文件。第二步，用你的Window浏览器， 引导到你刚刚解开的可执行文件的位置并双击来开始解压过程。你将看到下面的窗口：



保留默认值的设置并单击**结束**。

或许你将被要求输入密码，它是经电子邮件收到的。如果你没有收到密码，请联系ask.psd@st.com。



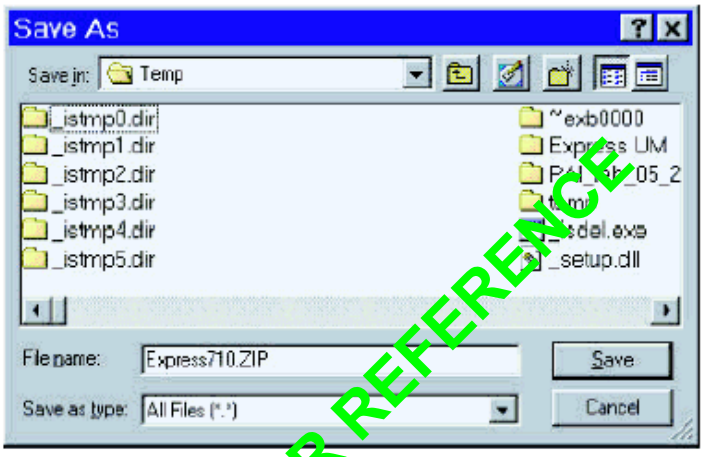
输入你的密码并点击**OK**。安装过程将自动开始。在安装过程完成时强烈推荐你阅读readme.txt文件。

新安装(从 CD ROM)

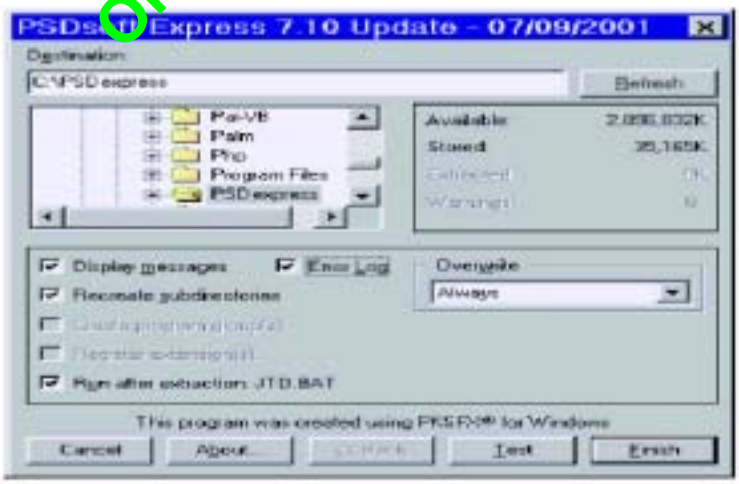
如果你是从CD安装，一般当CD被放进驱动器，安装程序将自动开始。如果没有安装，打开Window Explorer，选择CDROM驱动，双击安装（Setup）程序并按照屏幕上出现的提示进行。强烈推荐你在安装完成时阅读readme.txt 文件。

更新 PSDsoftEXPRESS

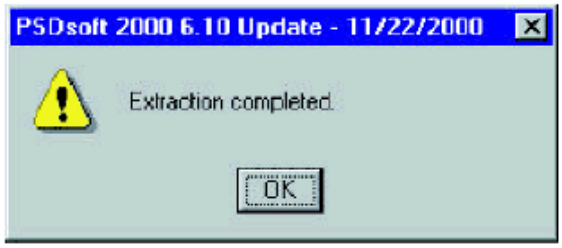
PSDsoft EXPRESS被周期性的更新以改善性能，增加新特色，支持新部件等等。因此，你应该定期检查www.st.com/psd来更新。当你单击更新链接www.st.com/psd/html/express_download.html.，会得到一个与下面相似的窗口：



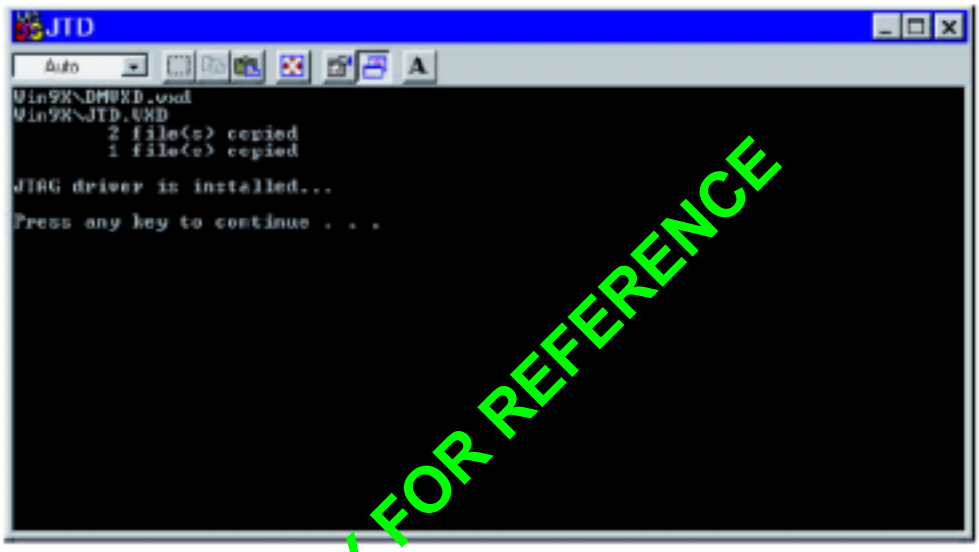
为.ZIP文件选择临时位置并点击保存。在.zip文件中有一个自解压的可执行文件的原因是有些公司有阻止下载可执行文件的防火墙。下一步，双击该自解压的可执行文件开始自解压过程。你将会看到下面的窗口：



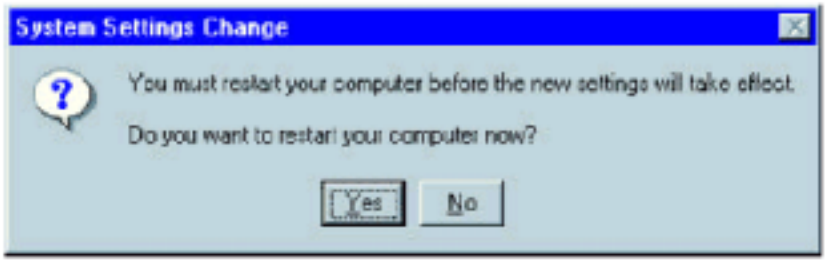
确认你已检查了所有的选框，如上所示，确认你选择了安装的软件目录并单击**完成**。
你应得到自解压完成的提示框。



你也应该看到出现另一个窗口指示JTAG驱动器被安装。



最后，提示你重新启动你的计算机。在你的计算机重新启动后，进入你安装的PSDsoftEXPRESS的目录并阅读README以了解更多的操作。

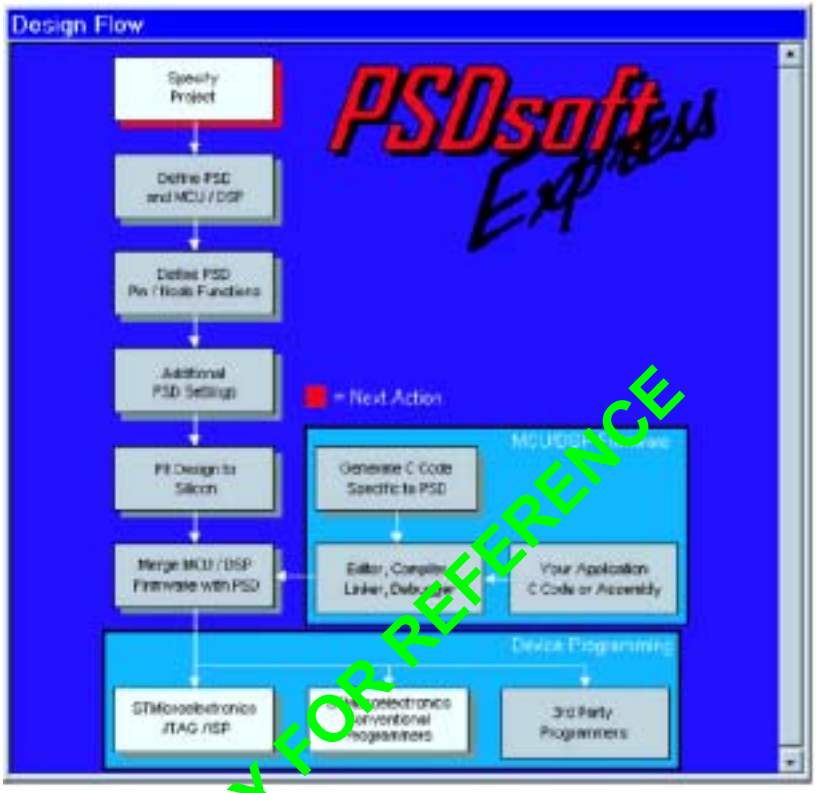


注：在你启动PSDsoftEXPRESS之前，确定你的屏幕的分辨率设定为至少800 x 600象素或更高，否则使用该软件会很困难。

PSDSOFT EXPRESS 设计流程

这一节讲述如何使用PSDsoft设计流程。使用设计流程，只要单击包含设计动作的选框。

注：根据选择的器件和工程的参数，你可能看不到下面所示EXPRESS设计流程的所有选择。在屏幕分辨率更低的计算机上你在窗口的右手边也可以看到一个滚动条。你可能需要下拉滚动来看所有的选择：



上面所示的是PSDsoft EXPRESS的设计流程。当选框为灰的阴影时，表示直到你完成了前一个步骤该选框中的过程才能被调用。红色阴影表示下一个动作可以被执行。在任何时候两个可被执行的动作选框是“STMicroelectronics JTAG/ISP”和“STMicroelectronics Conventional Programmers”。“产生C代码”选框在你完成了“定义PSD和MCU/DSP”后就可应用，除非你用的是一个EPROM的PSD，此时它不可用。

非 PSDsoftEXPRESS 动作

有一些动作PSDsoftEXPRESS不支持，但它是你选定的MCU/DSP设计过程的一部分。这些选框如设计流程中所示，且一直为灰色的并且没有阴影。一直为灰色的三个选框为：

*“Your Application C Code or Assembly”—这是你的特定应用的源代码和库，在PSDsoftEXPRESS外部产生。

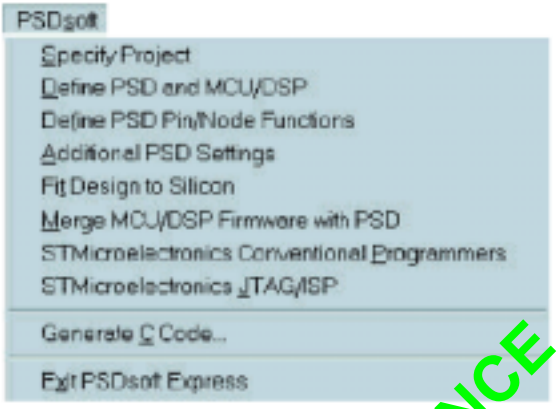
*“Editor, Compiler, Linker, Debugger”—这是你的MCU/DSP固件开发环境，它也不是PSDsoftEXPRESS的一部分。在这个环境中，你应该生成由PSDsoftEXPRESS产生的C代码与你的应用固件结合。

*“3rd Party Programmers”—如果你希望使用第3方编程器来对你的PSD部件编程，需要PSDsoftEXPRESS建立的.obj文件而且软件由第3方商家供应。ST资格认证的第3方编程器的完整目录，请检查网址www.st.com/psd。

菜单操作

尽管你用设计流程可以做一个完全的设计，但菜单上的一些特点你应该清楚。

开始“PSDsoft”菜单，如果我们看一下选项，我们会发现它们和设计流程（除了“退出PSDsoftEXPRESS”外）中一些可用项相对应。



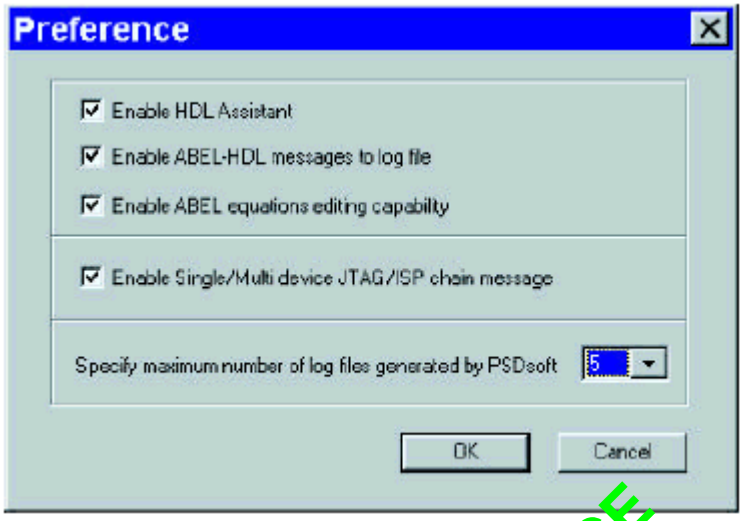
如果我们看看下一个菜单—“工程”，我们看下面：

其中“建立”，“打开”，“保存为”，“删除”和“关闭”的意思很明显，其他功能如下：

* “清除” 删掉PSDsoftEXPRESS产生的文件，该文件对描述你的工程无关紧要。当你有一个比较满意的工程设计数据文件（.obj）时你可能希望运行该功能。

* “关于”…给出你的工程的主要细节，并且有一些比PSDsoftEXPRESS主窗口右下边的信息更详细的信息。

* “参数选择”允许你开启（enable）或关闭PSDsoftEXPRESS的可选特点 。根据PSD的选择，下面所示一些选项可能不可用。



参数选择的意思是什么？

* “允许HDL助手：”当这个选框被选中时，无论什么时候设计流中**编辑/添加逻辑语句**选框被单击时，会弹出一个窗口。该选项允许你检查ABEL关键字和语法并看到一些简单的逻辑执行。见“关于HDL助手”这一节。

* “允许ABEL-HDL消息到日志文件：”打印错误和警告消息送到日志文件，该日志通常是位于屏幕底部的窗口打开。

* “允许编辑ABEL方程式”选中这个选框使**编辑/添加逻辑语句**选框出现在设计流程中。见编辑/添加逻辑语句（仅CPLD部件）。

* “允许单个/多个设备JTAG/ISP链接消息：”当这个选框被选中时，PSDsoft EXPRESS 将一直提示你设计流程中的**STMicroelectronics JTAG/ISP**选框被选中时你的JTAG链接多少设备。如果你有一个或多个设备并在前面作了这个选择，你可以通过不选定这个选框来关闭该提示。

* “PSDsoft产生日志文件的最大值”选择器可用来设定日志文件（.plg）最大值并保存在你的工程目录一到十之间。当到达最大值时，产生的下一个日志文件将取代最老的日志文件。

在“工程”之后，有一个“文件”菜单。

文件菜单选项工作如下：

* 新建…：产生一个新的文本文件。
* 打开…：打开一个已存在的文本文件来编辑。
* 保存，保存为…，打印，打印预览，和打印设置都是标准的窗口选项。

下一个菜单—“报告”有下面一些条目：

该菜单可用来浏览和打印下面的报告：

* 设备配置 — 基本显示了你的PSD是如何被配置的
* 设计助手概述 — 概述了引脚和方程式是如何在设计助手中配置的。如果你在任何设计助手屏幕上单击“**View**”按钮会看到同样的文件。
* “Fitter”— 显示你的设计如何映射到PSD芯片。
* 存储器映射 — 提供了你的存储器的映射图，包括内部和外部片选方程式。
* 地址转换 — 显示你的MCU/DSP存储器图转换到实际PSD地址并且MCU/PSD固件如何写入PSD存储器。

“观看日志文件”可以被选中或不选中。当被选中时，每个动作和与它相关联的错误消息（如果有）将出现在一个日志文件窗口中，这窗口通常在屏幕底部。更多资料见日志文件这一节。

“eWindow”菜单简单易懂，不需要说明。

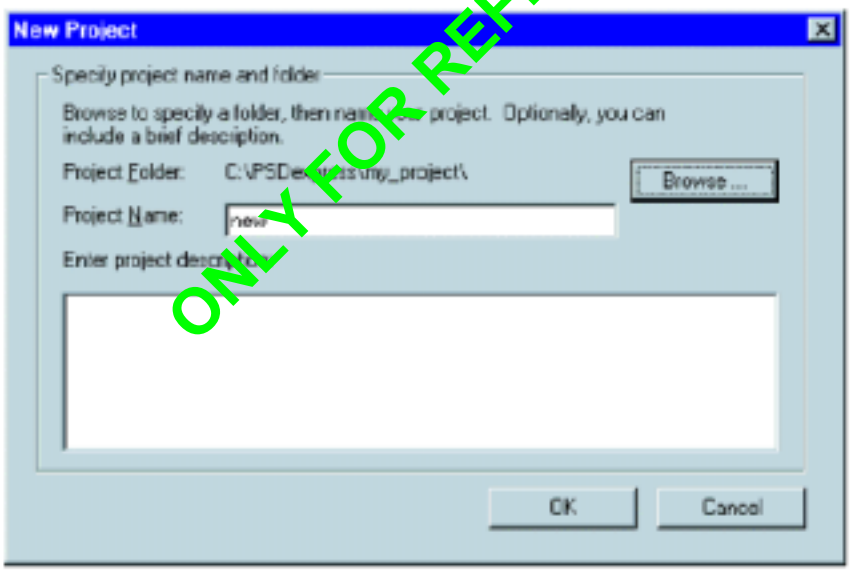
你的第一个工程

在这一节，你将学习PSDsoft EXPRESS完整的设计过程。在这一节后面的几节中的设计步骤更详细些。但是这一节的有些内容是其他任何一节中都没有的：

当你第一次双击PSDsoft EXPRESS图标，会看到下面的对话框：

选择“建立新工程”并单击**OK**。

你会看到屏幕为：



选择一个恰当的工程名并用**浏览...**按钮定位。如果需要可以输入工程说明。

选择和配置你的PSD与微控制器

单击**OK**，你会看到下面对话框：

你第一次进入该工程，对话框自动提示你下一个动作。

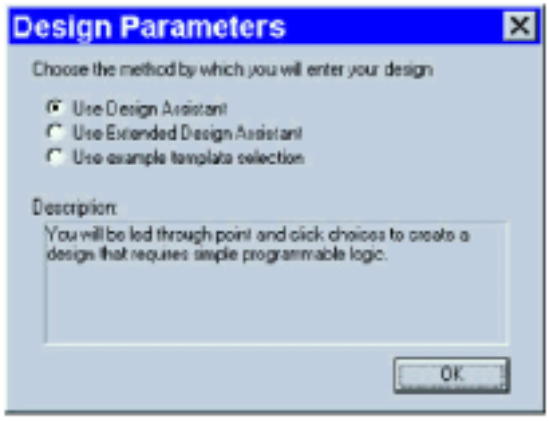
1. 从列表中选择你的微控制器（MCU）或数字信号发生器（DSP）制造商和MCU/DSP类型。如果你的器件在选择表中没有，尽可能的选择一个相兼容的。如果你的MCU/DSP不在表中且没有相兼容的，必须选择“〈其他〉”。如果MCU/DSP制造商和类型在表中，一些或所有的控制信号将自动为你选择。

2. 选择PSD系列，器件和你想用的封装。如果你不确定哪一个PSD最适合你的应用，单击**向导**按钮并根据向导步骤进行。

3. 在这一步，你定义MCU/DSP和PSD连接的方式。即8位或16位总线模式，复用或非复用总线等等。根据你的MCU/DSP选择，这些条目可能已经被选择。

如果你需要更多关于如何使用该屏幕的信息，请到MCU/DSP和PSD选择这一节。

重要提示：本文中许多屏幕上的信息依据不同的MCU/DSP而不同，特别依据你选择的PSD系列。一旦你完成了这一屏，单击**OK**到下一屏：



注：如果你选择的MCU/DSP没有相关联的模板，你将看不到“Use example template selection”选项并且如果你选择一个PSD没有CPLD，你将看不到“Use Extended Design Assistant”选项。选择设计助手允许你做一般的设计，设计中将提示MCU/DSP的PSD引脚的预定义及配置。

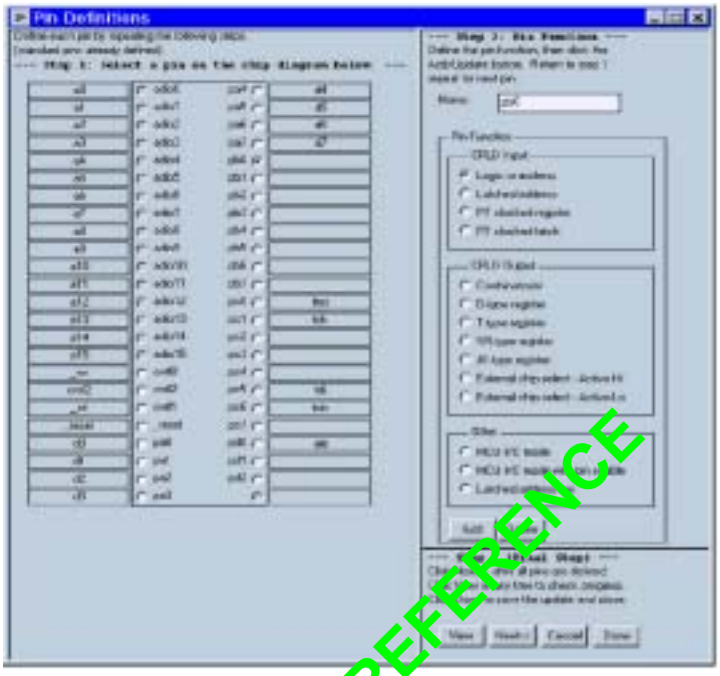
扩充的设计助手将提供HDL实例，并帮助有更复杂逻辑需求的用户，比如状态机和比较器。

选择实例模板给你一个设计实例，它基于你选择的特定MCU/DSP。包括推荐的存储器映射和I/O实例，并且在大多数情况下只需要较少的附加装置。

如果你希望使用设计助手或扩充的设计助手，请继续往下阅读。如果你喜欢使用一个实例模板，到使用实例模板这一节。

引脚定义

在你第一次进入到你的工程或者如果你在“设计流程”窗口中单击**定义PSD引脚/结点功能**，下面一屏会自动出现。



在这一屏，你将定义PSD的引脚功能。在这过程中，你将选择一个引脚然后定义其功能。其步骤如下：

1. 选择你想定义的引脚。

2. 可用的功能表依据你选择的引脚可能会变化。选择合适的功能，输入满意的名字再单击**添加**。如果你在之前已定义了该引脚的功能，“添加”就变成了“更新”。要得到引脚功能含义的简单说明，见引脚定义这一节（第8节）。

3. 你结束了所有的引脚定义后，选择**下一步>>**。如果你想浏览一下你的进程，选择**查看**。如果你仅改变引脚功能，不需要对片选方程式做任何改变，选择**Done**。

JTAG用户注意：如果你想在非复用模式下使用PSD的JTAG端口，要确认如上所示的JTAG引脚名。更多信息见引脚定义这一节。

注：

* 你可以调整引脚定义窗口大小并且你调整的新的尺寸将在后面的使用中被记住。

* 如果你想改变先前定义的引脚功能，采用下面的步骤：

1. 选择老引脚分配并单击**删除**按钮。

2. 选择先前被定义的新引脚，其功能将被指配。

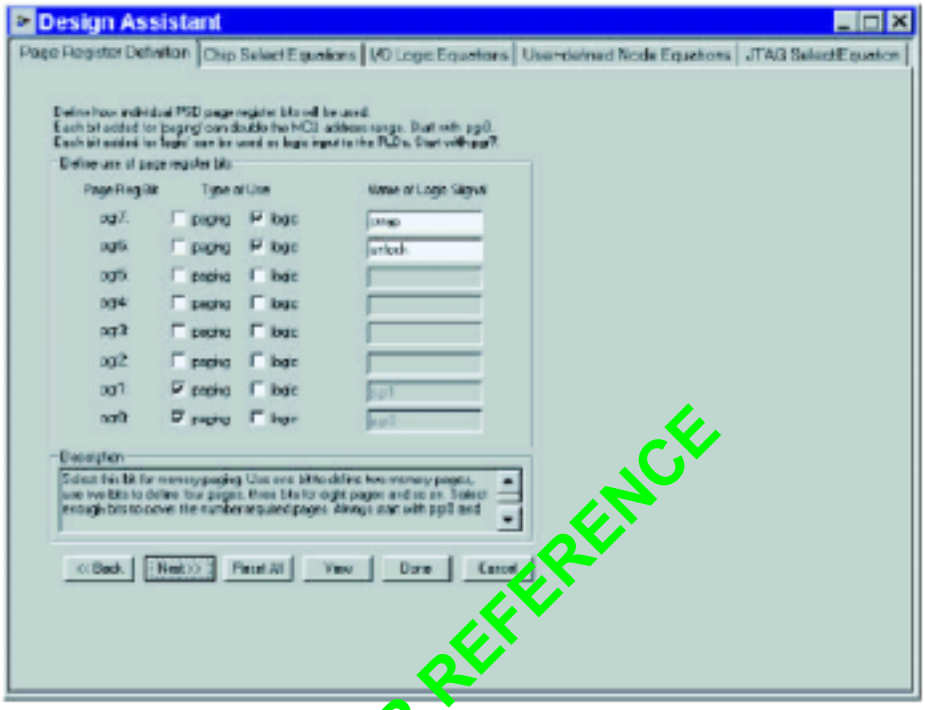
3. 分配引脚以前被使用的功能。

4. 给该引脚起名要与老引脚一致。

例如，看一下上面屏幕收集到的信息，“ram_cs1”定义在引脚pa0上作为一个“外部片选为高”。现在你可以选择pa0单击**删除**按钮，选择一个不同的引脚（pb0），选择“外部片选为高”，起名“ram-cs1”，再单击**添加**。你的“ram_cs1”的老方程式将仍有效。

页寄存器配置

从“引脚定义”窗口单击**下一步 >>**，会看到“页寄存器定义”窗口：



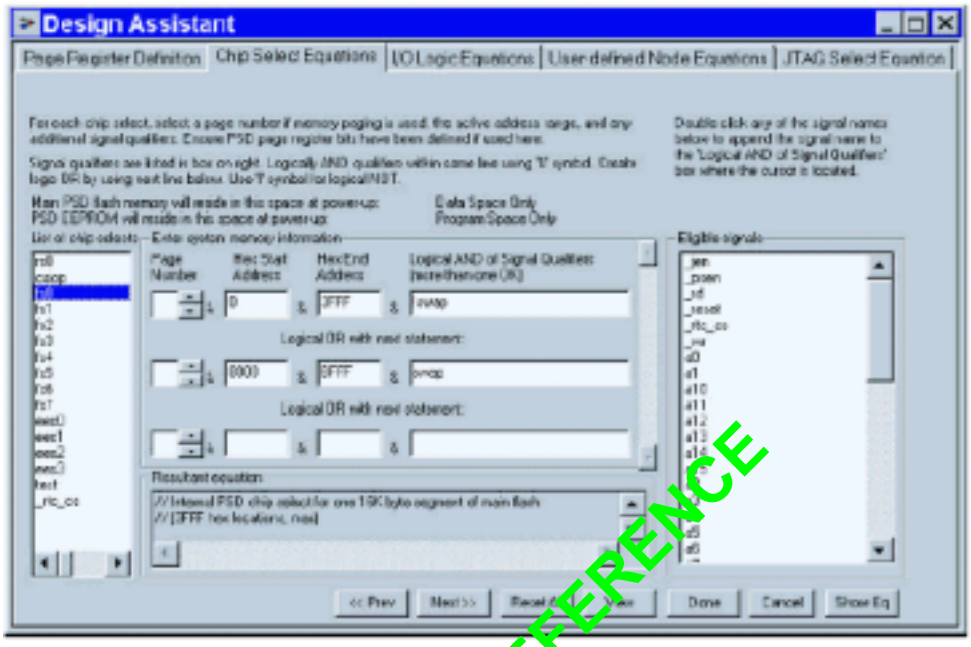
页寄存器在寻址空间有限的MCUs/DSPs中的应用将是非常理想的。它可以用来扩充物理地址空间或用作PLDs的逻辑输入。如果你想在你的设计中使用任何PSD页寄存器位，在操作之前它们必须在这儿定义。注：单击**<< 向后**使你返回到“引脚定义”屏。

想知道更多的关于页寄存器的信息，请看页寄存器节和什么是分页节。

重要提示：如果你选择一个带64K字节存储扇区（现在只有PSD835和PSD935）的PSD和限定地址总空间为64K字节的MCU，很有可能需要建立一个可用来控制存储器页被称作“fa15”的结点。该位常用来在64K字节存储扇区的一半（32K字节）之间选择。把两个十六进制文件分配到单个存储扇区见分段。

芯片选择方程式/存储器映射定义

单击**下一步>>** (或单击“芯片选择方程式”图标)，引导你到“芯片选择方程式”屏：



你在这个窗口定义内部和外部片选方程式。要做这些需采取下面的步骤：

1. 在最左边栏目中选择合适的片选段。

2. 选择片选有效（如果可应用）的页数值。然后输入片选的十六进制地址范围。你可以通过把它们放在下一栏来逻辑与（AND）这些信号。你也可以通过在连续行上输入参数来逻辑或（OR）这些信号。

3. 重复上面的步骤直到你定义了所有的片选方程式。

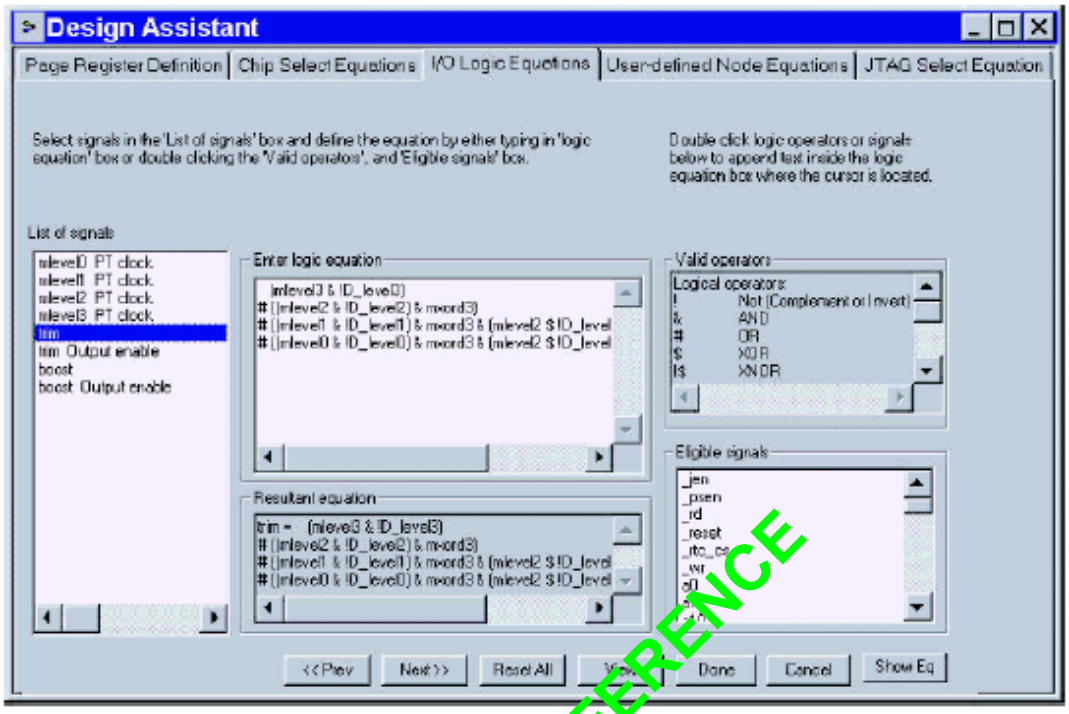
例如，如果你希望内部Flash存储器选择片0（fs0）从0H到3FFFH有效并且“SWAP”位为低，或从8000H到BFFFH有效并且“SWAP”位高，你要象上面一样输入信息。

注：要从合格的信号表中添加一个标记，双击表中你希望添加的名字然后它就会为你添加。与此同时在同一行上将添加“&”符号，这样标记就添加上了。

更多的信息见芯片选择方程式节（第10节）。如果你有一个非CPLD部件，你将看不到其他的图标来定义方程式类型，你可以单击**Done** 并跳到附加PSD设置。如果你准备到下一步，点击**NEXT >>**。如果你有一个CPLD部件而且不能确定应该使用什么方程式，那么在“芯片选择，I/O逻辑和结点方程式之间有什么差别？”（第19节）这一节有更多的帮助。

I/O 逻辑方程式（仅 CPLD 部件）

单击**下一步>>** (或点击“I/O逻辑方程式”图标)，你会看到与下面相似的一屏：



用这一屏给PSD中的I/O定义逻辑方程式。如果你需要在方程式中使用用户定义结点，首先要单击**下一步>>**到下一屏或点击“用户定义结点方程式”图标并单击该窗口中**定义结点...** 按钮。

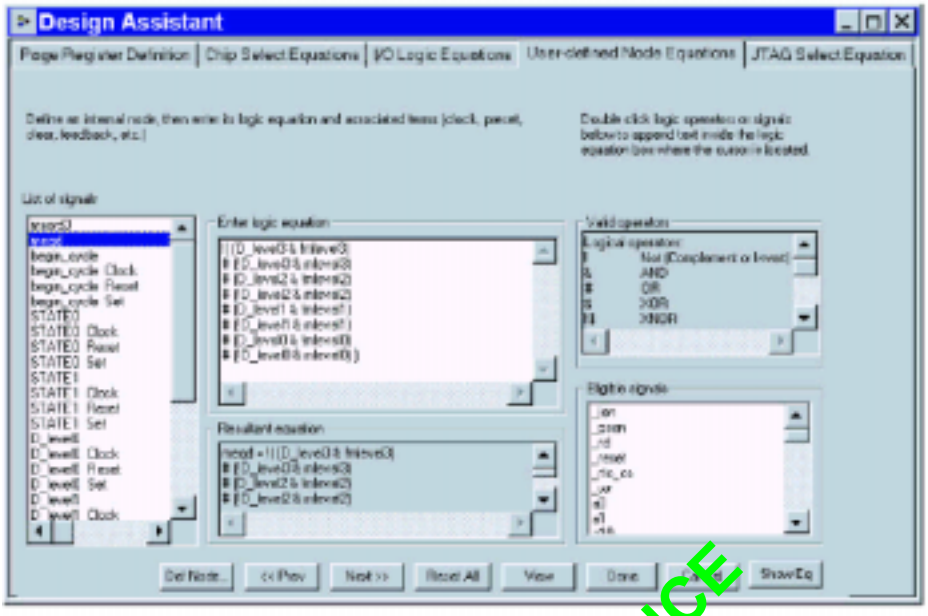
定义一个I/O信号，该信号要先从“信号列表”选框（在最左边）中点击信号来定义。然后你可以在“输入逻辑方程式”选框中键入方程式，或通过双击“信号”列表来输入。你也可以输入逻辑操作符或者双击右边的“有效操作符”列表中的操作符来选定。用括号来表达优先权。一旦输入方程式结束，你可以单击**显示方程式**按钮然后结果就会呈现在“结果方程式”选框中。 (如果你继续定义另一个信号，然后再回到原来地方，也会看到结果方程式。）到**Report->Fitter**菜单中可以浏览到最后表格中的方程式。

注：不需要键入等号。事实上，如果你输入等号标记会产生错误。

例如，标记“trim”在前面已经被定义为一个组合CPLD输出引脚。要定义那个信号，它已在所示的列表被突出，然后所示的方程式被输入。结果方程式在底部显示。

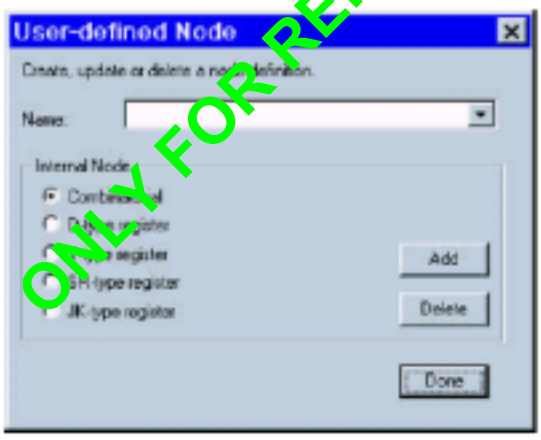
用户定义结点方程式(仅 CPLD 部件)

单击**下一步>>** (或单击“用户定义结点方程式”图标)，你会看到屏幕与下面相似：



这一屏在用法上与“I/O逻辑方程式”的屏幕很相似。

要定义一个新结点或删除或更改一个前面定义的结点，单击**定义结点...** 按钮。你将看到下面窗口：



要定义一个新结点，在“名字：”选框键入想要的名字并选择类型（组合逻辑或D，T，SR或JK触发器）再单击添加。要变更或删除一个已存在的结点，从“名字：”选框下拉列表中选择该结点然后选择一个新功能单击**更新**或简单地单击**删除**去掉该结点。当你完成了结点定义后，单击**Done**。

定义一个内部结点的功能性，先从“信号列表”选框中左击该信号选定要定义的结点。然后你可以开始在“输入逻辑方程式”选框中的逻辑方程式中输入方程式，或通过双击“信号”列表来输入方程式。你也可以输入逻辑操作符或双击右边的“有效操作符”列表中的操作符来选定，这样它们就会被输入。用括号来表达优先权。一旦你完成了输入方程式就可以单击**显示方程式**按钮，结果会显示在“结果方程式”选框。（如果你继续定义另一个信号，然后返回到原位置也会看到结果方程式）到**报告->Fitter**菜单中可以浏览到最后表格中的方程式。

注：不需要键入等号。事实上，如果你输入等号标记会产生错误。

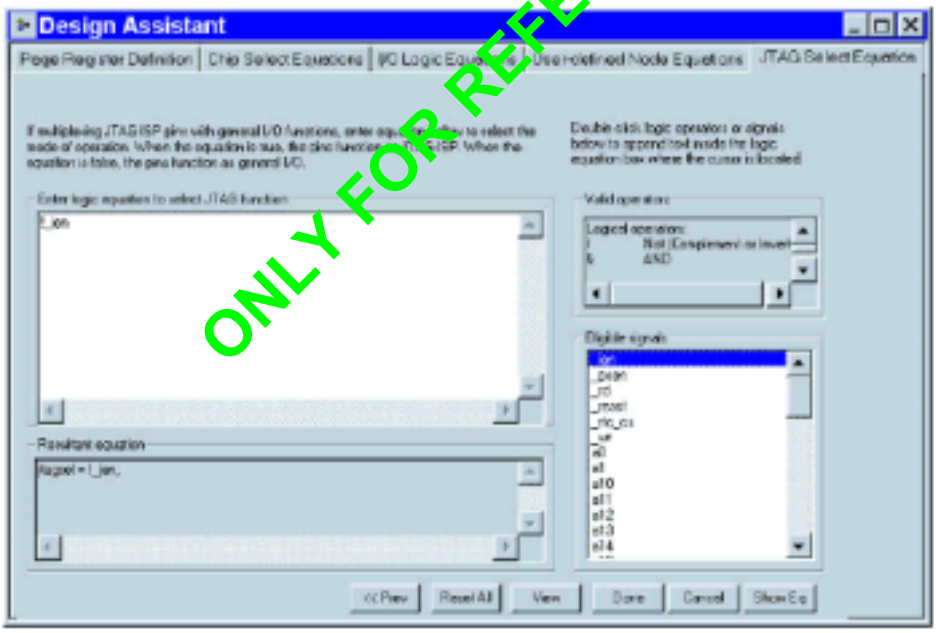
JTAG 选择方程式(仅带复用 JTAG 端口的部件)

如果你有一个带复用JTAG端口的器件并且没有看到这个图标，那么将把该引脚给JTAG专用（默认）。换句话说，该图标出现的唯一条件是器件有复用JTAG端口，而该端口的JTAG引脚已定义了附加功能。

该屏常用来允许/禁止JTAG端口。这个I/O引脚依次连到JTAG操作的PSD FlashLINK™电缆。然后在JTAG操作时，电缆将驱动引脚拉低，使能（enable）JTAG端口。FlashLINK电缆用它的/JEN信号去做这个工作。在正常（非ISP）操作时，该引脚将保持高电平，并且JTAG端口将被禁止而且其他的I/O可以被连到JTAG端口。于是，如果你想与其他I/O共享JTAG端口，你需要定义引脚。一旦你定义了引脚，然后用这一屏来给引脚（该引脚已定义到称为“jtagsel”的内部结点）赋值，这实际上控制着（打开或关闭）JTAG端口。

例如，你定义一个端口引脚为“逻辑或地址”CPLD输入并命名为pin _jen。你应该把电路板上_jen连到/JEN，信号从FlashLINK电缆输出。然后双击非符号（!）把_jen分配给“jtagsel”结点（在“JTAG选择方程式”屏），再在信号列表的右边双击_jen符号。你也可以简单键入方程式。单击**显示方程式**按钮看结果方程式，即“jtagsel = !_jen;”。这个方程式主要说明，当指定_jen的引脚为逻辑0时，JTAG端口将被允许（enable）。

由于JTAG端口的应用容易混淆，因此强烈推荐阅读 “JTAG-ISP Information for Flash PSDs”应用笔记和数据手册中的JTAG部分。



编辑/添加逻辑语句（仅 CPLD 器件）

如果你使用设计助手时有复杂逻辑设计需求，单击设计流程中的**编辑/添加逻辑语句**选框。注：如果你在设计流程中看不到这个选框，到**工程->属性**菜单选中“允许ABEL方程式编辑能力”选框。如果你想编辑或添加HDL逻辑语句（如这一节中所建议的），推荐你下载应用笔记：“Flash PSD，CPLD入门”。它讨论了如何为不同的目的使用PLDs。如果你不想使用这个选项可以条到下一节。

非常重要：如果你想在ABEL设计文件中添加任何声明或方程式，必须放在指定区域。任何声明或方程式放在指定区域外将**丢失**！ 声明就是引脚和结点的定义，以及你想写方程式的符号组。在方程式这一

段之前出现的任何条目都被看作是一个声明。要ABEL语言的全部指引，见PSDabel-HDL参考（从www.st.com/psd上软件中心可下载）或用HDL助手可看到实例。

两个指定区域如下所示：

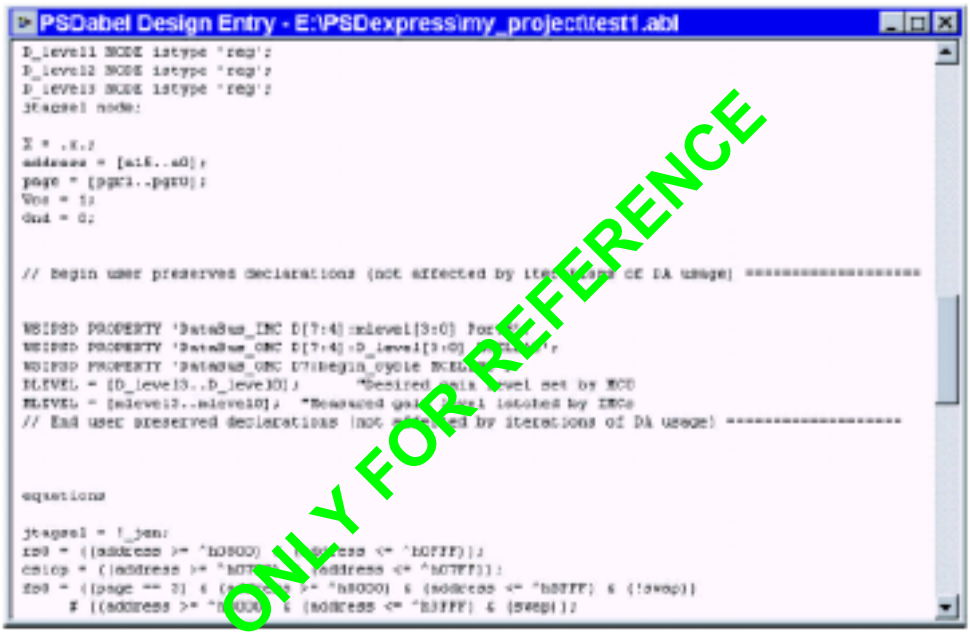
声明：

```
// Begin user preserved declarations (not affected by iterations of DA usage)
// End user preserved declarations (not affected by iterations of DA usage)
```

方程式：

```
// Begin user preserved equations (not affected by iterations of DA usage)
// End user preserved equations (not affected by iterations of DA usage)
```

一个带声明样例的ABEL文件例子如下所示。注意当ABEL文件被打开时菜单会改变。



要更多的ABEL语法帮助，额外的可用菜单，以及何时用ABEL与何时用设计助手，见编辑/添加ABEL方程式节（第十一节）。

关于模拟器 PSDsoft Express有一个仅模拟PLD方程式的模拟器。为调用PSDsoft Express模拟器，必须有方程式指定部分的有效测试向量。要知道测试向量语法，见ABEL方程式节。要运行模拟器，ABEL文件打开时到**报告->模拟结果**。

关于HDL助手无论何时如果在**工程->参数选择**菜单中选中了“允许HDL助手”，你在设计流程中单击**编辑/添加逻辑语句**选框，HDL助手应该自动打开。HDL助手有三个基本用途：

* 列出被每一个PSD设备保留的名字：引脚、结点和PSD中寄存器(在PSD中有特殊功用)保留名字。
* ABEL语言用法模板：显示ABEL语言的所有基本单元。
* 普通逻辑模板：显示如何实现普通逻辑功能和PSD指定的声明和功能。

有两个方法使用HDL助手中的代码：

1. 从左边列表中选择想要的模板或保留的名字。然后选中右边的代码把它粘贴到正确的指定区域。

对大多数普通逻辑模板，最好做两次复制和粘贴动作。换句话说，你要将一部分粘贴到声明节而从其中剪切方程式再粘贴到方程式节。

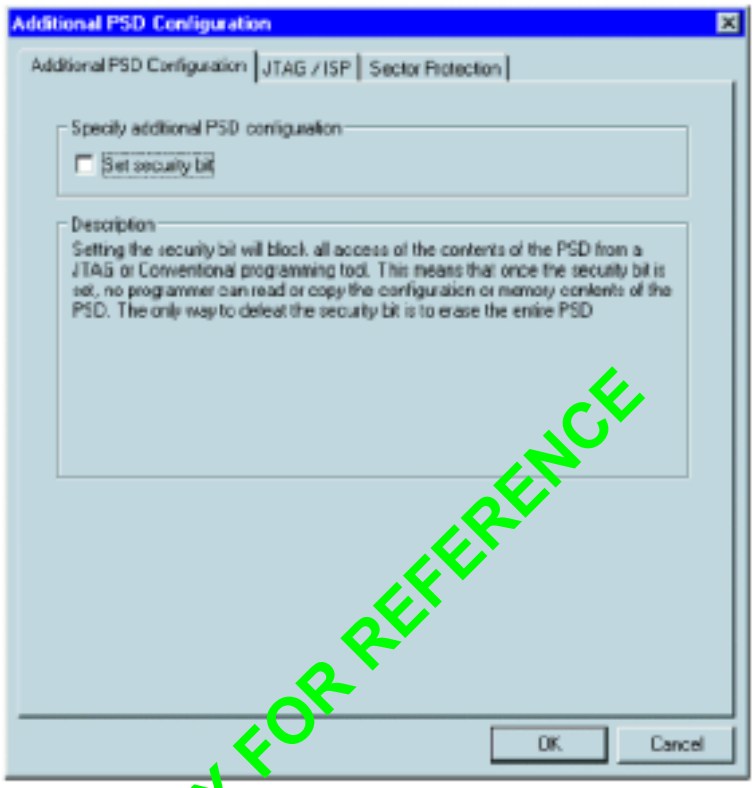
2. 把指针放在ABEL文件的一个指定位置。然后从列表左边选择想要的模板或保留名字右击，从下拉菜单中选择**使用**。你需要再次剪切指定区域的部分代码粘贴到其他地方。

如果你想定义自己的模板，右击“用户定义逻辑模板”然后选择从下拉菜单中选择**新建**。

删除和**重命名**只能在用户定义逻辑模板中使用。

附加PSD设置

设计流程中下一步是**附加PSD设置**。在设计流程中单击**附加PSD设置**按钮，见会看到如下屏幕显示（除非你的PSD部件没有Flash存储器，这种情况更多的信息见附加PSD设置节（第12节））：



在这个对话框中可以完成三个功能：

1. 设定加密位 — 阻止所有通过JTAG或传统的程序烧写器对PSD存储器内容的访问。就是说加密位设定了后，程序烧写器不能阅读或复制PSD的配置或存储器的内容。擦除加密位的唯一方法是完全删除PSD。

2. 指定用户模式 — 允许你输入32位代码，它可用作不同的功能。单击“JTAG/ISP”图标阅读说明框，你可以了解更多的细节。

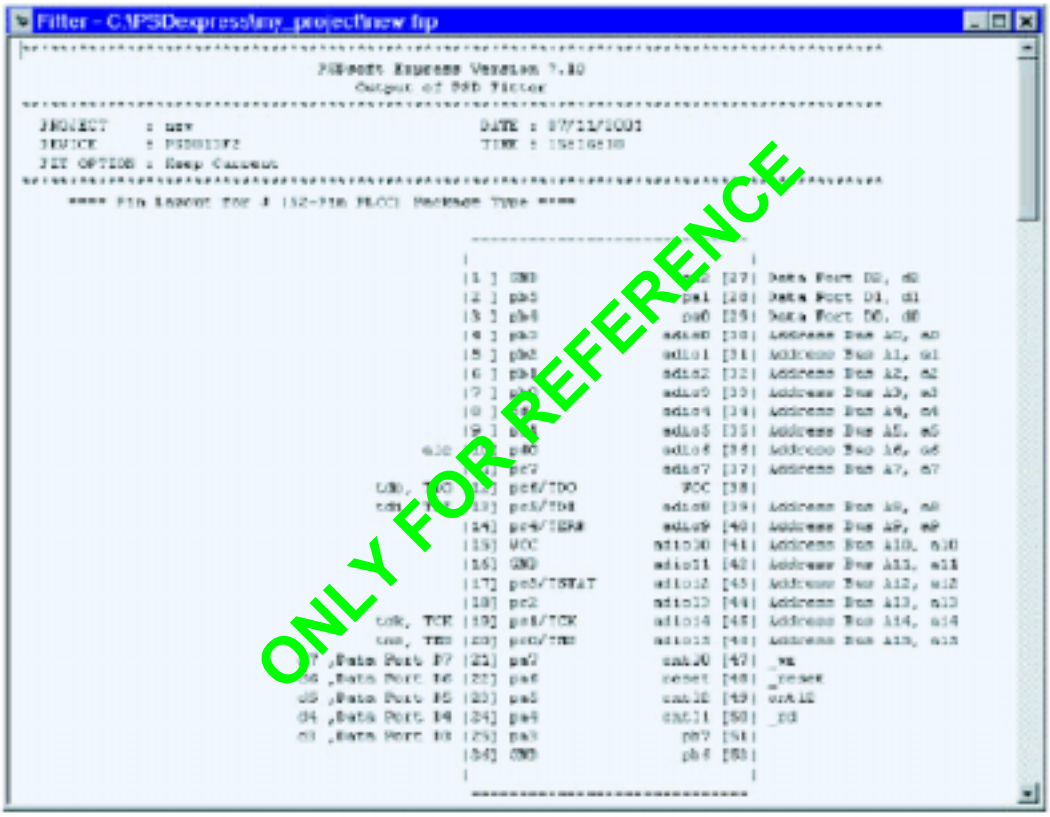
3. 设置内部存储器扇区保护 — 允许写PSD的各个存储器扇区来防止突然的数据丢失。MCU/DSP在运行时不能改变这些设置。只有程烧写器能更改这些设置。

单击**OK**你会再次看到设计流程。更多的信息见附加PSD设置节（第12节）。下一步，我们要尝试把设计装载到芯片中。

把设计安装到芯片

PSDsoft Express 装配程序把你的PLD设计映射到PSD芯片。该装配程序自动的编译你放入指定区域的ABEL代码并返回错误报告。如果没有错误，将产生一个部分程序数据文件（.obj）。(要完成.obj文件，必须完成“合并MCU/DSP固件与PSD”过程。)

要调用程序，单击设计流中的**装配设计到芯片**选框。你会注意到PSDsoft Express 完成了一些你的设计到芯片的装配。如果你在这一步中有错误，你应该返回并根据给定错误编辑你的设计。如果你分辨不出是什么原因造成的错误，请与apps.psd@st.com 联系如果你的设计正确会看到屏幕如下类似（注：你要看到这个文件必须到**Report->Fitter**菜单。）：

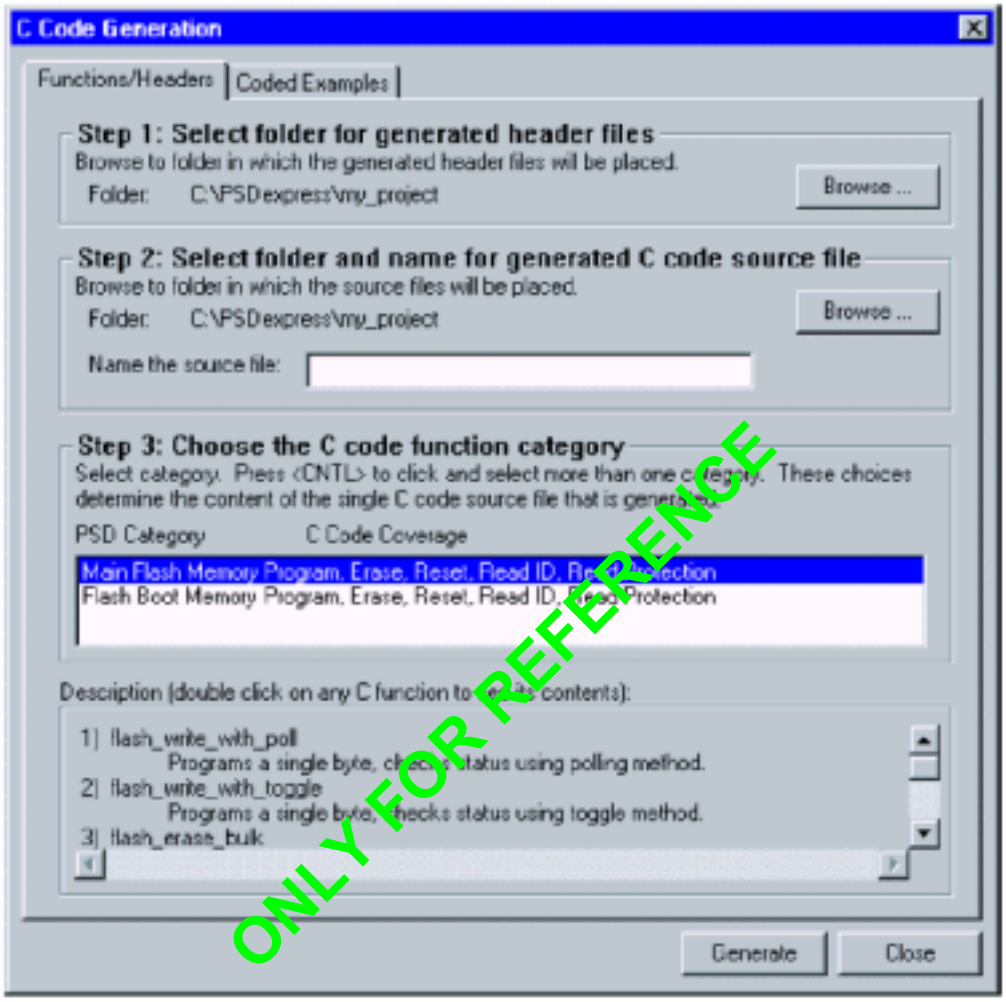


这一屏在看资源分配与剩余以及软件如何翻译你的方程式和设计入口方面很有用。更多信息见装配设计到芯片节（第13节）。

在你完成建立用于对PSD编程的.obj文件之前，可能你想知道PSDsoft Express怎样帮助你产生有用的C代码。如果你的PSD无Flash 存储器或不想用PSDsoft Express产生C代码，可以在建立文件（用来对PSD编程）— 程序数据文件时，跳过该子节（5.12节）。

产生C代码(仅Flash/EEPROM PSDs)

在设计流中单击**产生C代码**得到下面屏幕：



使用PSDsoft Express的这个功能可以产生用C语言写的C代码实例，它可以完成以下功能：

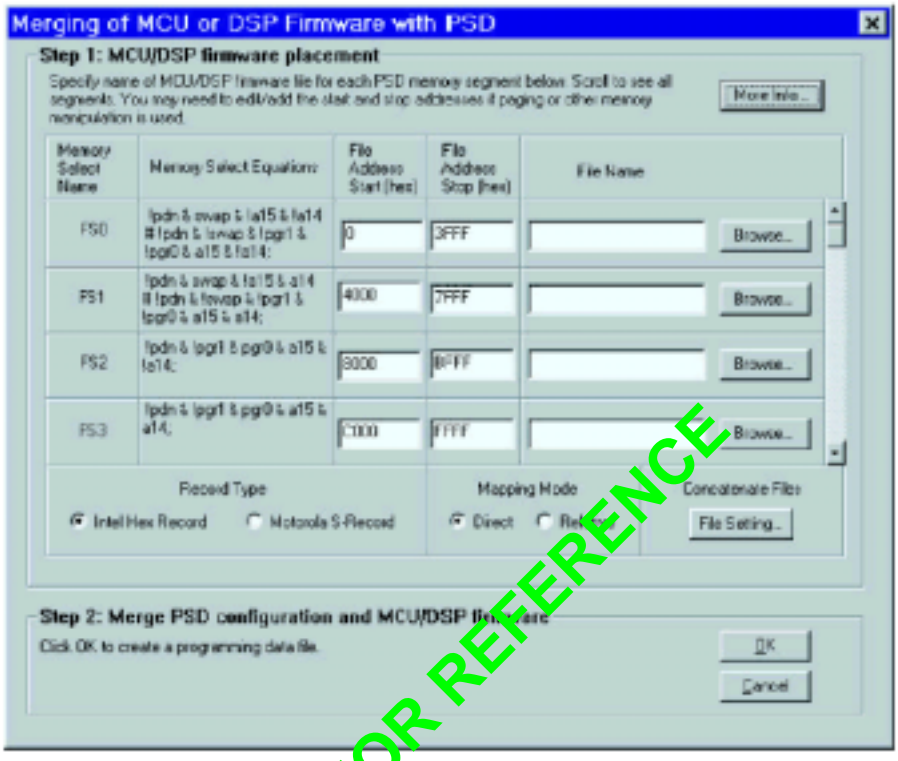
* 编程和擦除PSDs的非易失性存储器（NVMs）。
* PSDs NVM(s) 复位到 “读模式”。
* 读Flash存储器标识符字节和读保护状况。

当你准备用PSDsoft Express生成所需的文件时，单击**产生**按钮 。一旦产生按钮被点击，日志文件(.plg)将列出产生的文件。注：对“Functions/Headers”图标，你可以双击“说明”选框中任何一个功能来浏览包含在选择功能中的C代码列表。你也可以单击“Coded Examples”图标，使用PSD开发工具箱来浏览完整的实例。更多的信息和产生文件列表见产生C代码节（第14节）。

可以了，我们现在准备好产生用于PSD编程的程序数据文件(.obj)的剩余部分了。

建立文件（用来PSD对编程）— 程序数据文件

在设计流程中点击**合并MCU/DSP固件与PSD**按钮。你会看到与下面类似的窗口：



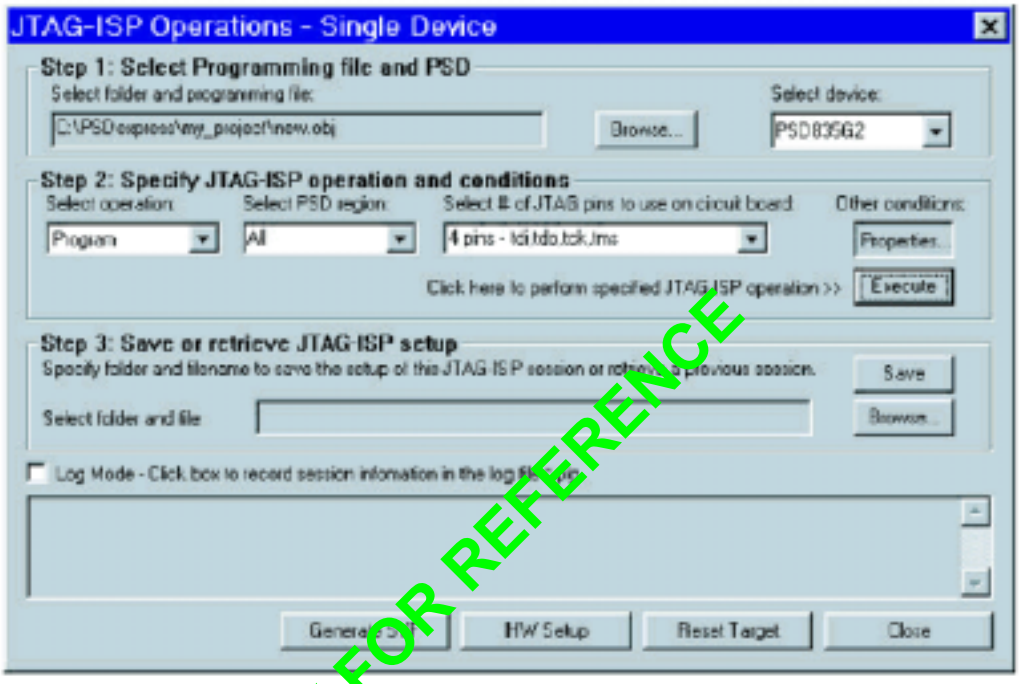
注：如果你在设计中使用了页寄存器可能会收到警告。基本上，该警告是告诉你PSDsoft Express检查到片选方程式没有包含正常MCU/DSP地址信号。该警告是提醒你确认MCU/DSP编译器/连接器。如果他们不自动出现，你可能不得不手工填入文件开始/结束地址。同样，为了分页设计，必需核对开始/结束地址的正确性。

此时你应该准备Intel Hex Record 或 Motorola S-Record 文件。对PSD每一个存储器扇区，点击**浏览**按钮并且定位你准备在该扇区烧写的文件。注意一个文件可能跨几个扇区。记住，需要时用下拉来添入所有的文件名。根据你的编译器/连接器输出的“Intel Hex Record”或“Motorola S-Record”选择合适的类型。下一步，选择希望的映射模式。“直接”映射表示十六进制文件和PSD存储器片段位置一一对应。“相联”映射允许你指定不同的物理地址（基于你的固件文件的MCU/DSP输出），在PSD中选择存储器扇区的指定方程式。就是说，十六进制文件将放在指定PSD扇区的底部。大多数用户选择“直接”，当你完成后，单击**OK**，然后程序数据文件就产生了。程序数据文件包括你的PSD设计和你的MCU/DSP固件。更多细节见合并MCU/DSP固件与PSD。

你现在准备好了用JTAG或常规程序设计器来对PSD编程。在下面两个子节中两种方法都有介绍。

用JTAG对PSD编程（仅基于Flash存储器的PSDs）

单击设计流程底部的“STMicroelectronics JTAG/ISP”选框，会问你“你的电路板的JTAG链有多少个设备？”如果你的设计仅有一个PSD设备并且没有其他JTAG兼容设备，选择“一个”。如果你多于一个PSD/或JTAG兼容设备，浏览JTAG章节（第16节）。如果你选择了“一个”，“JTAG-ISP操作—单个设备”对话框将打开：



对PSD编程应采取下面的步骤：

1. 在选框中确定希望的程序数据文件（.obj）。单击“浏览”按钮定位想要的文件。然后确定选择的芯片与你希望编程的芯片一致。

2. 指定JTAG操作PSD的区域以及你将对PSD编程使用的引脚数（4或6）。点击“特性”按钮设置PSD引脚在JTAG操作期间应如何工作。为PSD的核查JTAG特性并且输入用户代码与先前输入（附加设置项）的比较。单击**执行**按钮，选定的JTAG操作就开始了。

3. 保存你的JTAG配置。如果你以前保存过一个文件，你可以象第一步那样打开它而且你先前的所有设定都将被恢复。

如果你在JTAG操作期间遇到一些问题，可以去PSM网址看JTAG FAQ节。你也可以下载JTAG应用笔记：“JTAG-ISP Information for Flash PSDs”。 你还可以单击“日志模式”选框来保存日志文件然后把文件送到apps.psd@st.com分析。

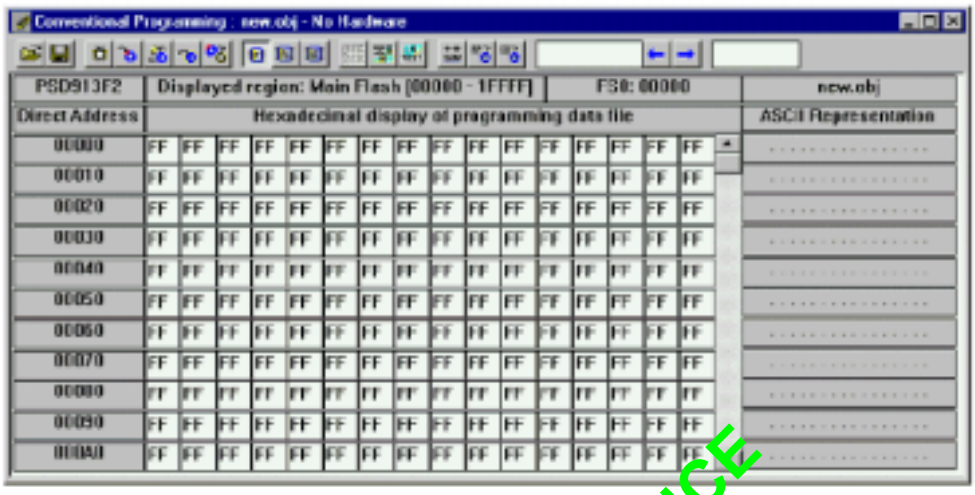
更多细节见JTAG编程章节。

注：可以在“Conventional Programming”浏览.obj文件，而不管你是否有标准烧写器。更多细节见PSD标准烧写器（第17节）。

如果你完成了融合MCU/DSP固件（与一个有效的十六进制或S-Record文件）和JTAG-ISP步骤并且在步骤3保存了你的设置，下面的菜单中将有一个新的按钮变得可以使用。

用 PSD 标准烧写器对 PSD 编程

要到标准烧写器窗口，单击**STMicroelectronics标准烧写器**按钮。



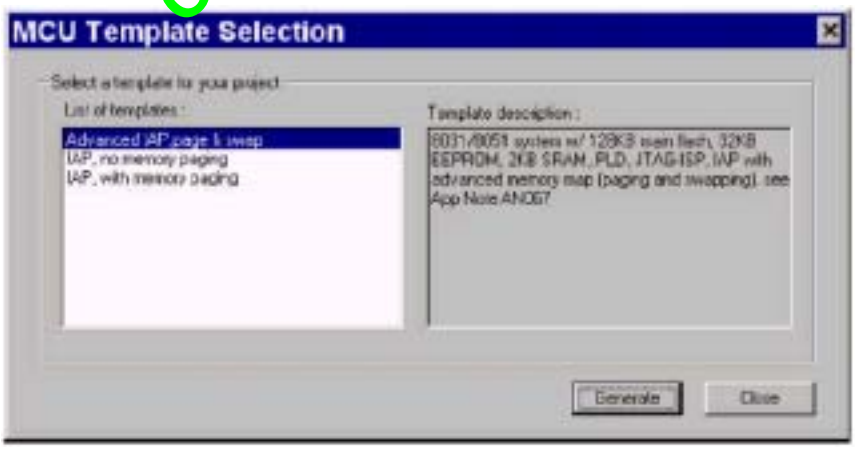
用各种工具栏来执行想要的功能。如果你不清楚某个按钮是干什么的，让鼠标指针在该按钮上停留，然后将出现一个“工具提示”。

注：用“标准烧写器”窗口，你可以浏览和编辑程序数据文件（.obj)。

更多细节见PSD标准烧写器章节（第17节）。

使用实例模板

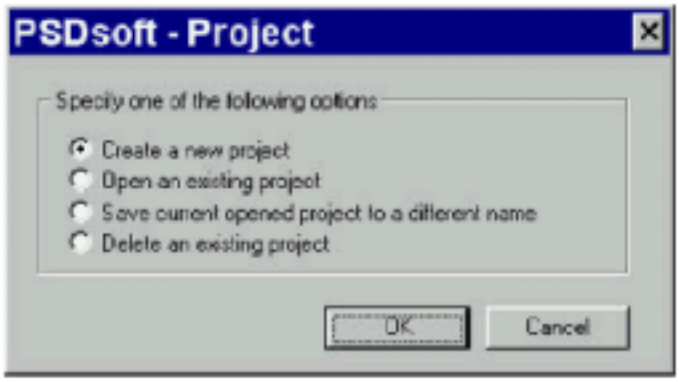
如果你在“设计参数”窗口（前面显示）选择了使用实例模板，根据你选择的MCU/DSP将出现与下面所示类似的一屏。每一个可用模板有一个与之相联系的简短说明。选择与你的终端系统匹配最接近的模板。



如果你不想使用模板，单击**关闭**你将返回到“设计参数”屏。否则，一旦你已选择，单击**产生**就会到引脚定义屏。在引脚定义，页寄存器定义（如果你选择了带分页的模板），以及芯片选择方程式屏将填入一些信息。在单击**Done**之前，看一下上面提到的每一屏，看看模板产生信息以及做一些想要的修改。

指定工程

在设计流中单击“指定工程”，将出现下面窗口：

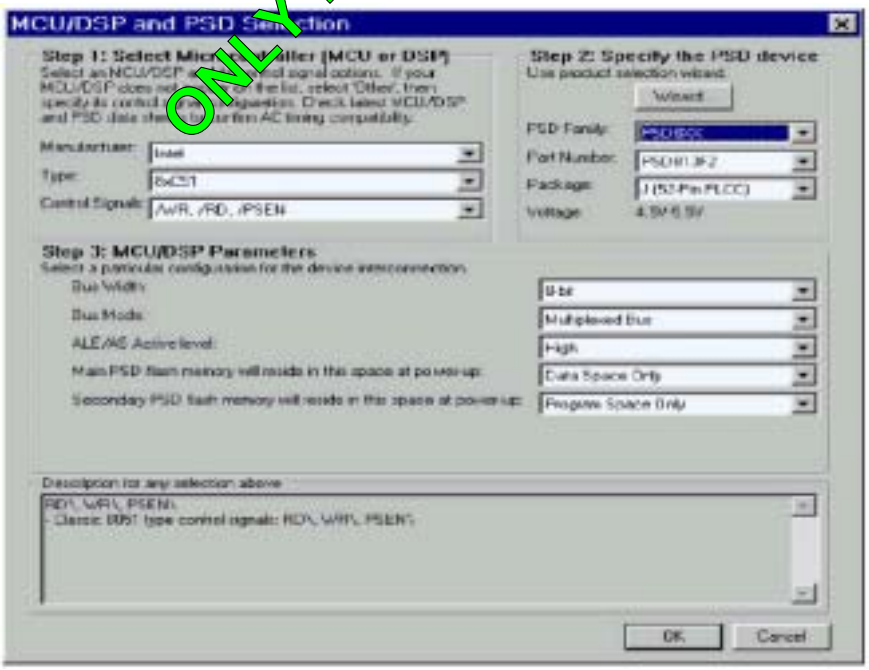


你在里面有四个选择：

* 建立一个新工程 —使用这个选项建立PSD每个设计的不同工程。
* 打开一个已存在的工程
* 保存现在打开的工程为不同的名字—在工程后面注释时很有用。这样你想检查以前的某些工作是如何做的，就有一个参考了。
* 删除一个已存在的工程 —仅当你希望完全清除一个工程以及所有与之相关的文件时使用该选项。

注： 如果你只希望删除重建工程时不需要的文件，可以从菜单中选择工程->清除。

MCU/DSP 与 PSD 选择



步骤1: 选择微控制器 (MCU/DSP)

在该步骤中，你将选择PSD要使用的微控制器(MCU)或数字信号处理器(DSP) 。首先，从列表中选择制造商。如果你想使用器件的制造商不在列表中，可以选择在不同的制造商中找一个相似的MCU/DSP。如果你不能够定位想要的MCU/DSP，把制造商设定为“<其他>” 并且你将自己选择信号。根据选定的MCU/DSP，控制信号将自动为你选择，除非有多个可能的组合。如果有多于一个的可能性，从下拉列表中选一个合适的。

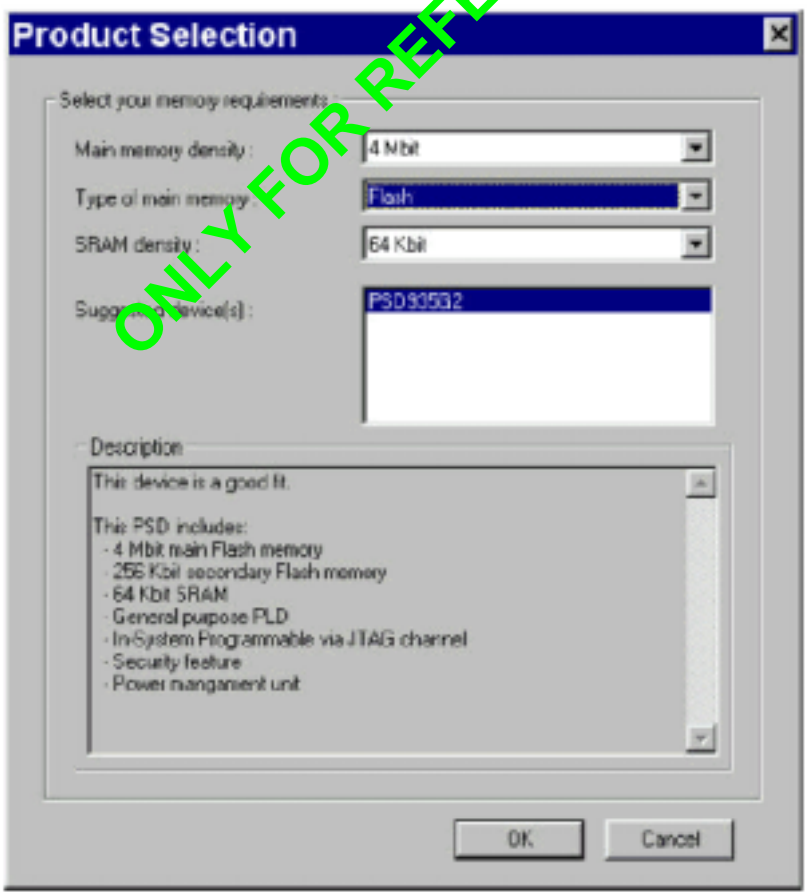
MCU/DSP 是什么？如果你不确定微控制器是什么或干什么，见微控制器系统笔记：http://www2.tcd.ie/Engineering/Courses/BAI/JS_Subjects/3D1/Notes/index.html。

如果你的MCU/DSP没有被列表。 如果你的微控制器(MCU)或数字信号处理器(DSP)不在可用的MCUs/DSPs列表中并且你可能想在PSDsoft Express的将来修订本看到，发邮件给：ask.psd@st.com。也可以用这个email来提供你想在将来的PSDs中看到的附加特性的反馈。

步骤 2: 指定PSD设备

如果你不确定PSD是什么，可访问PSM网址：www.st.com/psd。如果你清楚的知道你想使用的是那一种PSD，就选择系列，部件号和封装。如果你确定PSD时需要一点帮助，只须在“步骤2”中简单的单击**向导...** 按钮。

你会看到下面一屏：



在这一屏选择你的系统需要的“主”存储器容量，用“主存储器” 这一术语是因为大多数PSDs有一个第二存储器用于在应用编程（IAP）。选择了主存储器和SRAM后，你将看到与你的选择相匹配的一系列设备。

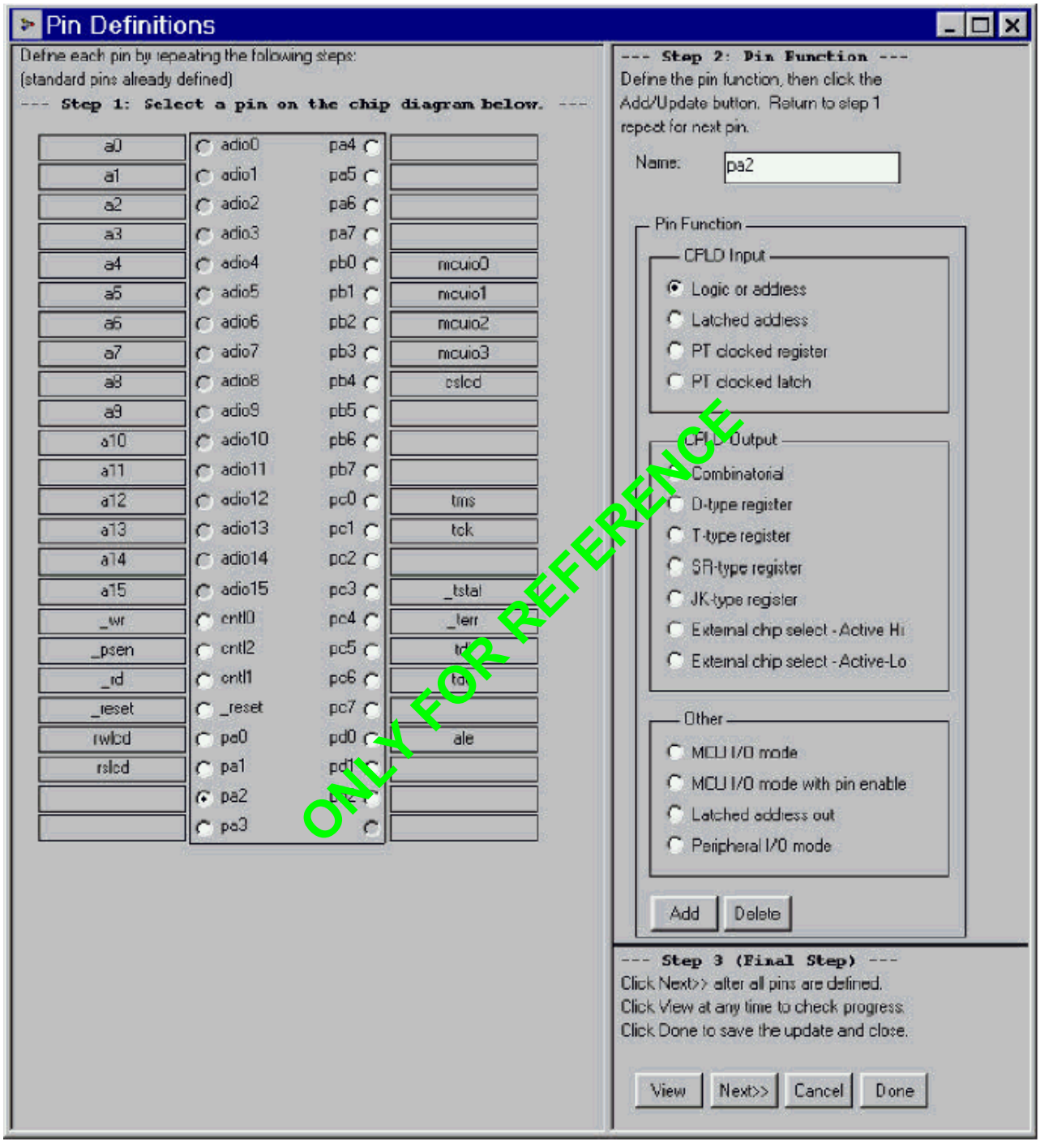
步骤 3: MCU/DSP 参数

在最后一步，定义MCU/DSP与PSD相互连接的方式。即8-位或16-位总线模式，复用或非复用总线等等。根据你的MCU/DSP选择，这些项可能已经为你选定。否则根据需要进行合适的选择。

分离的程序与数据空间

有些MCUs，比如80C51，P51XA，和80C251通过控制一个称为程序空间允许（PSEN）或类似的信号来支持分离的程序与数据空间。这也许是“MCU/DSP和PSD选择”屏的最复杂部件，因为它问你哪里安置主存储器和（可选）第二存储器 。把一个存储器安置在“仅程序空间”意味着MCU通过PSEN读存储器。这就是说，它只考虑代码存储器。与之相对的“仅数据空间”也是这样。在这种情况下，MCU将只能用RD信号读存储器。 但是，如果你选择“程序与数据空间”（结合空间），则PSEN或RD信号都可以访问存储器。在结合空间模式下，你将不得不为任何重叠存储器明确地添加控制信号到内部PSD片选方程式。记住这个选择将指定在电源加点和复位时PSD如何动作。MCU在运行时通过写PSD的VM寄存器（PSD的CSIOP寄存器的一部分）来修改这些选择。通常，在电源加电和复位时第二存储器安置在“仅程序空间”而主Flash存储器安置在“仅数据空间”。它允许MCU从第二存储器导入然后是程序和/或主Flash存储器的合法内容。之后，MCU可以通过写VM寄存器把主Flash存储器置入“仅程序空间”。见PSM站点的应用笔记。 还有各种各样的开发板的用法说明手册（比如DK900），以及更多实例。

引脚定义



步骤 1

ADIO0到ADIO15常常只留作地址（非复用总线）或地址与数据复用总线。其他类型的I/O不能连到这些引脚并且引脚名不能被改变，因此这些选择只灰色的。如果你使用的MCU/DSP为非复用总线，各种端口将自动地为数据总线保留。CNTL0 和 CNTL1也将为MCU/DSP的控制信号（比如/RD和WR）保留。

根据你的MCU/DSP选择，CNTL2可能也被用作控制信号（如/PSEN）。但是，如果你的MCU/DSP仅使用两个控制信号为存储器访问，CNTL2可被用作到PLD的输入。_RESET 也是一个保留引脚。控制和复位引脚只有一个可能的功能，但如果需要引脚名可以改变。其他端口引脚可以留作特殊MCU/DSP控制信号。根据你的MCU/DSP选择，这些通常将为你选定。例如，你能从上面抓到的屏幕所看到的，PD0留作ALE信号。

对于可能有多功能的引脚，如果你想知道一个引脚的隐藏功能，只须在步骤1中单击各个引脚，其可用功能将根据选择的引脚列出来。一般，端口引脚不必为了特定功能而作为一类使用。该规则的一个例外是用JTAG端口作为JTAG专用，但是如果你设定一个JTAG信号的作为专用，其他引脚功能将自动为你设定。PSD JTAG端口的更多信息见“JTAG-ISP Information for Flash PSDs”应用笔记。其他的例外是外围I/O模式中将需要整个端口。

引脚功能简短说明

下面是引脚功能的简短说明。注：PSD系列的功能性变化，则这些选择中有一些可能不可用。更多细节见数据手册与应用笔记。

(C)PLD 输入

逻辑或地址是最常用的输入类型，它可以在PLD中用作解码和其他逻辑功能。从MCU/DSP的附加地址输入可在这儿使用。

锁存地址 —该引脚的信号输入将闭琐ALE或AS信号。这在MCU/DSP为多路复用时的附加地址输入很有用。

PT 时钟寄存器 —该引脚将被连到IMC D触发器，该触发器可在PLD中使用另一信号定时。

PT 时钟锁存 —该引脚将被连到IMC透明锁存器，该锁存器可以在PLD中使用另一信号锁存。

(C)PLD 输出

组合 — 基于组合逻辑方程式的输出。

D-类型触发器 — 从 OMC D-类型触发器输出。

T-类型触发器 —从 OMC T-类型触发器输出。

SR-类型触发器 — 从 OMC SR-类型触发器输出t。

JK-类型触发器 — 从 OMC JK-类型触发器输出。

外部芯片高电平选择 — 基于解码地址和控制信号的高电平输出。

外部芯片低电平选择 — 基于解码地址和控制信号的低电平输出。

其他

MCU I/O 模式 —运行时允许使用MCU/DSP数据总线直接访问PSD的端口寄存器。

MCU I/O 模式带引脚允许 —除了一个由CPLD驱动的输出允许信号可用来控制输出外，其他同上面一样。

锁存地址输出 —用ALE或AS闭锁的复用地址位可以在一些端口引脚上输出。查看数据手册，看哪一个地址位将在哪一个引脚输出。

外围设备 I/O 模式 —在该模式中整个端口作为你的MCU/DSP的三状、双向数据缓冲器。

专用JTAG —设置这个选项将设定这个引脚，也会设定其他专用JTAG 引脚。这些引脚成为专用引脚并且不能与普通I/O功能复用。

SRAM 备用电压输入 —选择这个选项需要一个电池或可选电源连到这个引脚，在电源掉电时用来保护SRAM中的内容。

Rdy/Bsy 输出 —指示PSD非易失性存储器的写/擦除状态

备用电源指示器—系统到电后指示SRAM从备用电源输入引脚（上面有说明）提取电流。

公共时钟输入 —时钟输入到宏单元，自动掉电单位计数器，和PLD与门阵列。

PSD 片选输入 —该信号可以把PSD存储器置为备用模式（功耗非常低）。这是一个低电平有效信号。

注：PSD211R， (Z)PSD3XX ， (Z)PSD4XX，和 (Z)PSD5XX引脚可能有些功能这儿没有列出来。此时，详细情况请查数据手册。

带 JTAG ISP 功能 PSD 部件的特色注解

大多数JTAG用户把带JTAG功能的端口作为专用端口，即不与其他I/O复用JTAG引脚。如果你是这样的应用，可以跳过这一段。但是如果你想与其他I/O复用JTAG引脚，可以采取下面的动作：

1. 由于JTAG引脚是默认为JTAG，首先你需要把那些引脚上的功能改变为需要的I/O。例如，如果你希望把一个引脚在JTAG操作时作为TMS功能，而在非-JTAG（正常）操作时作为片选（称作CS0），你要选择“tms”(引脚定义屏中的引脚pc0)并把它重命名为“CS0”，选择其功能为“外部芯片选择”然后单击**更新**按钮。

2. 用同样的方法继续定义剩下的JTAG引脚。

3. 在JTAG操作时定义一个引脚为JTAG引脚使能（例如，JTAG选择方程式（第5.7节）中所示的_jen，（在JTAG端口复用的情况下））。任何PLD输入引脚都可以。为与我们的文献保持一致，建议“JTAG Enable”的名字为“_jen”。该引脚功能在CPLD输入中应该被设定为“逻辑或地址”。 这个引脚在后面将用来访问PSD的特殊寄存器以打开与关闭JTAG端口。见PSDsoft Express用户手册中的JTAG选择方程式节。

4.关于JTAG更多细节记得阅读数据资料和“JTAG-ISP Information for Flash PSDs”应用笔记：

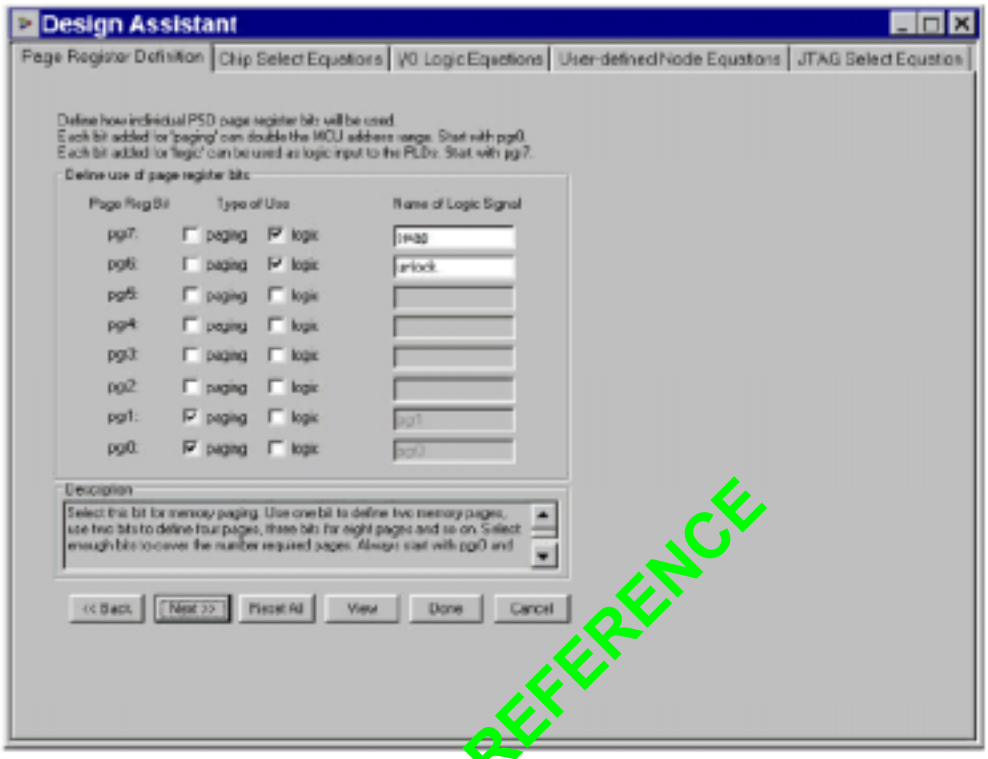
步骤 2

当你第一次定义端口引脚时，在“步骤2”中输入你想分配给该引脚的名字（最多31个字符）到“名字”选框中，然后从列表中分配引脚功能并单击**添加**。如果你想改变引脚的功能或名字，就输入新名字或功能再单击**更新**按钮（它是**添加**按钮）。如果你想删除一个引脚功能而不想分配新功能，只需单击**删除**按钮。注意如果你删除一个引脚功能或在第一步中没有定义它，其默认值是“PLD输入”。于是任何未使用的引脚应该接到Vcc (100K的上拉电阻应该足够了)或接地以避免CMOS PSD吸取过多的电流。

步骤 3

任何时候，如果你想检查设计进展，你可以单击**View**按钮。单击**下一步>>**把你带到“页寄存器定义”（除了没有页寄存器的PSDs外）。如果你不希望保存你做的任何修改而且你不想继续使用设计助手，可以单击**取消**。如果你只想改变引脚定义，单击**Done**。

页寄存器配置



如果你不确定什么是分页或如果你的设计需要分页，见“什么是分页”。

页寄存器是通过CSIOP空间访问的寄存器之一。页寄存器可被MCU/DSP直接读或写。页寄存器的输出被连到PSD的PLDs.

页寄存器使用

页寄存器有三个用法：

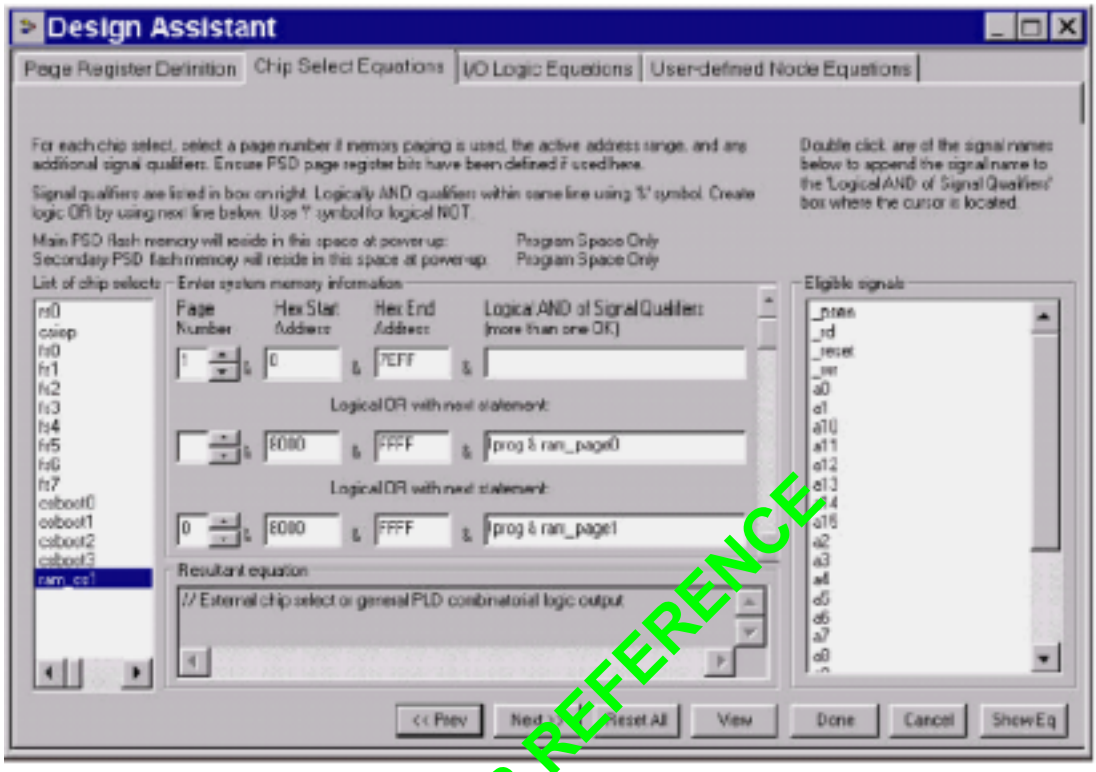
* 为你的地址解码逻辑分页。
* 为通用逻辑输入到PLDs。记住，页寄存器用MCU/DSP写。
* 作为一个可被MCU/DSP访问的通用存储寄存器。

如果你想把页寄存器用作分页和逻辑，分页位应该从pgr0开始顺序向上，而逻辑位应该从pgr7开始顺序向下。上一屏所示的八个页寄存器位的四个将被用作设计：pgr0 和 pgr1被用作分页 (将可到$2^2 = 4$页).。Pgr7被用作逻辑输入（称为“Swap”），pgr6 为逻辑输入（称为“Unlock”）。

移动

当你对页寄存器配置满意时可以单击**下一步>>**或“芯片选择方程式”图标，然后你会被带到可以定义内部和外部片选方程式的窗口。单击**全部复位，**清除所有在屏幕上的输入（在确认之后）。单击**View**检查到现在为止的进展情况。单击**<< Back**回到“引脚定义”屏。如果你不对片选方程式做任何改变，但是想保存目前的进度，可以返回“设计流程”窗口单击**Done**。如果你不想保存在这个期间所做的改变，单击取消。

芯片选择方程式/存储器映射定义



这个屏幕用来定义你的系统关于PSD以及任何外部片选的存储器映射。如果你用一个实例模板，这些片选中有一些已被填入，此时你只须简单的做些必要的调整来与你的需要相配。

下面的步骤是一个循环过程：

1. 在左边栏目中选中你想定义的片选。

2. 如果你在页寄存器定义屏中定义了一些页并且你想为选中的片选分页，在其栏目中输入一个页数字或用箭头选中它。如果没有分页，则不需要任何动作。

3. 在片选选定有效情况下，在下两个栏目中用十六进制格式输入开始和结束地址。

4. 在“logical AND of Signal Qualifiers”栏目，你可以从最右边的信号列表中输入任何信号（你希望它包含在最终的方程式中）。

5. 如果你有附加信号到逻辑与，在信号名后输入“&”。非信号使用符号“!”。

6. 如果片选信号也需要多个乘积项（逻辑或），重复2到5的步骤。单击**显示方程式**按钮看一下“结果方程式”选框中的结果。依据需要重复步骤。

7. 重复步骤1到6直到左边栏目中所有的片选都被定义完。提示：你开始该过程之前先画出系统级存储器映射图是非常有益处的。同样，从数据手册中会发现每一个PSDs内部片应该有多大以及存储器扇区重叠的规则（优先方案）。

重要提示：内部片选方程式正常时不需要与MCU/DSP控制信号相与，因为这在芯片级已自动被考虑，而且这样做会造成延时问题。唯一例外是在内部存储器被置为组合空间模式与重叠时（选择PSD与MCU/DSP），需要MCU/DSP控制信号允许的外部片选必须由用户添加。

看一下上面屏幕的例子，我们可以看到：

– 信号被定义为“ram_cs1”
– 信号在第1页从0h 到 7EFFh有效“或”
– 信号从8000h到FFFFh “与”!prog“与”ram_page0有效“或”
– 信号在第0页从8000h到FFFFh“与”!prog“与”ram_page1有效。

这很容易使人糊涂，因为使用了两种形式的分页：PSD的内部分页方案和称为“ram_page0”和“ram_page1”的逻辑输入。关于分页的更多信息见附录B.4。

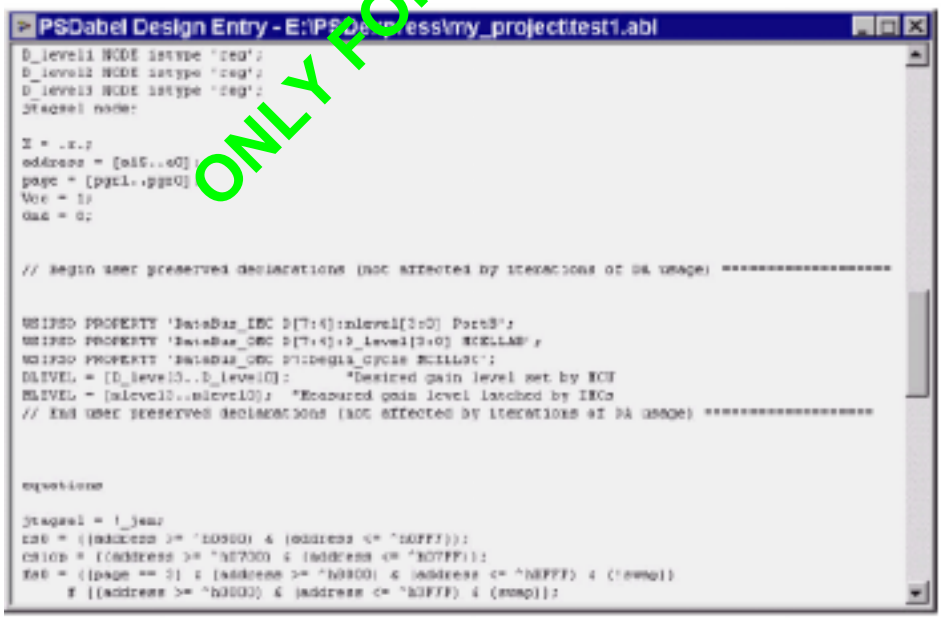
最终地址解码读：

```
ram_cs1 = ((address >= 0h) AND (address <= 7EFFh) AND (Page == 0)) OR (((address
>= 8000h)
AND (address <= FFFFh)) AND ((!prog AND ram_page0) OR (!prog AND ram_page1)))
```

你完成了输入片选方程式后单击**View**来检查结果。如果你想返回到页寄存器屏，单击**<<前一步**按钮。如果你想重新输入信号的定义，选中想要的信号然后单击**全部复位**。如果你对结果满意，单击**Done**，它将做初步信息检查和语法检查确保你没有输入任何无效的方程式。

编辑/添加ABEL方程式

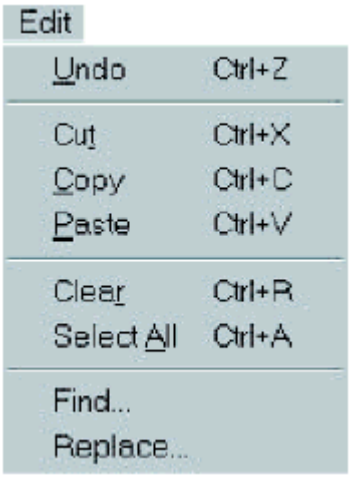
该文件包含下面的信息：
– ABEL文件打开时菜单项目可用
– PSD指定的PSDabel-HDL关键字和语法
– 测试向量语法
– 什么时候使用设计助手以及什么时候使用ABEL



手动编辑ABEL文件时，所有指定部分的声明与方程式（包括测试向量）必须被保留。

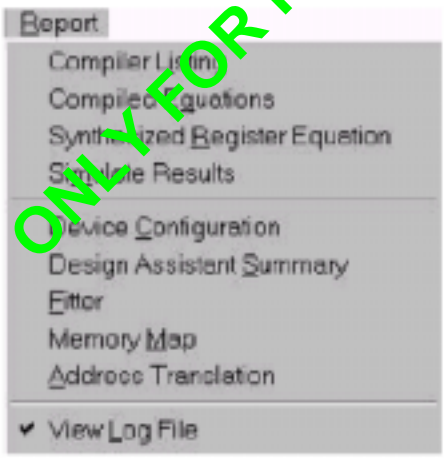
ABEL 文件打开时可用菜单项目

编辑菜单



仅当ABEL文件打开时编辑菜单才出现，他们在一个典型文本编辑器中做同样的功能。

报告菜单

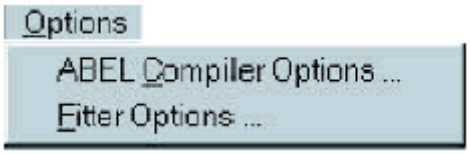


当ABEL文件打开时，在报告菜单中有附加菜单项目可用：

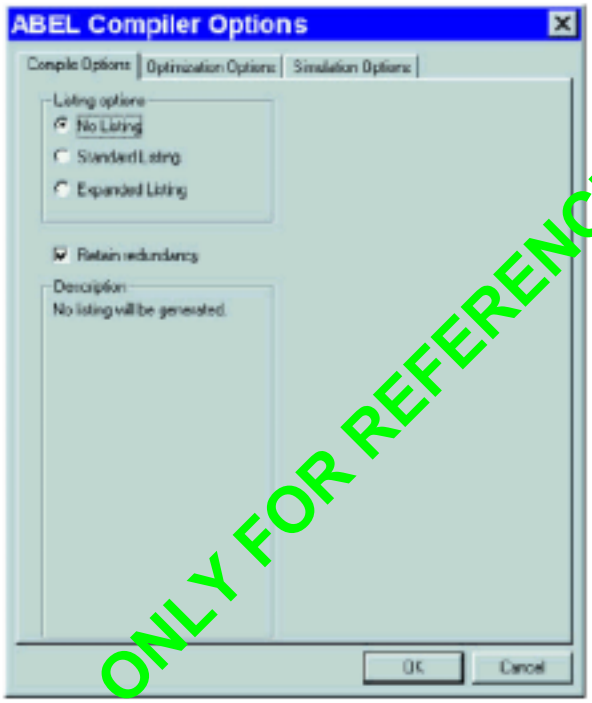
* 编译器列表编译你的最新版本设计并显示带行数的编辑列表。
* 编译方程式编译和优化你的最新版本设计文件然后显示优化后的编译方程式的输出。
* 合成的寄存器方程式编译，优化并综合SR，JK，T和D寄存器。完成后你应该可以看到结果。
* 模拟结果执行基于你的ABEL文件中写的测试向量，而显示结果基于你的模拟选项。

选项菜单

选项菜单仅当ABEL文件打开时出现。Fitter选项在Fitter时覆盖。



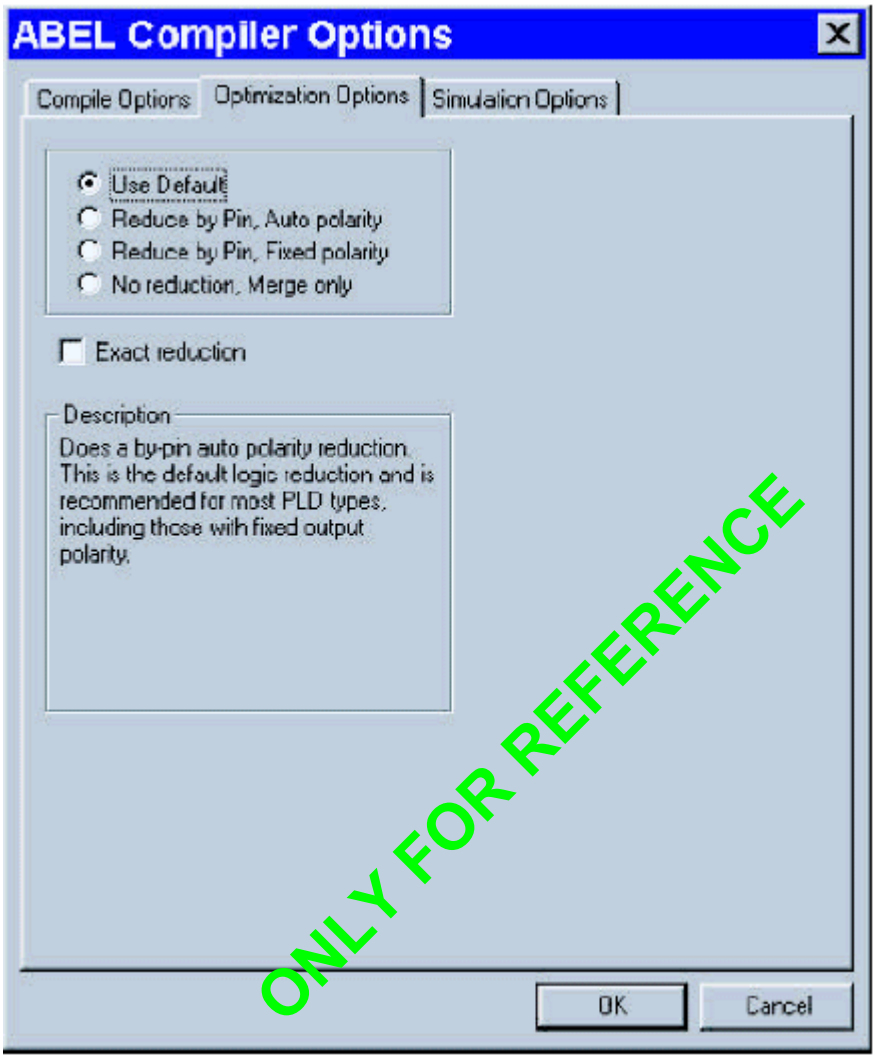
当ABEL编译器选项被选定后，“ABEL 编译器选项”屏出现：



编译选项

* 无列表意味着编译过程没有永久文件要建立。这是一个默认设置而且大多数用户不需要去改变它。
* 标准列表产生包含数字化的源文件行和错误消息列表（如果有的话）。
* 扩展列表产生包含数字化源文件行，扩展宏单元，指令，和错误消息（如果有的话）。
* 保留冗余：选定这个选框禁止编译时间微量减少，比如 A # !A = 1。如果你希望保留多余的乘积项（防止危险）就选定这个选框。注意，这些减少也是自动进行的，除非“不减少，仅融合”选项在下一屏被选定优化。

在对话框顶部单击“优化选项”图标，你会看到：



优化选项

* Use Default 选择此项将自动的减少极性。这是默认的逻辑减，推荐用户使用此项。

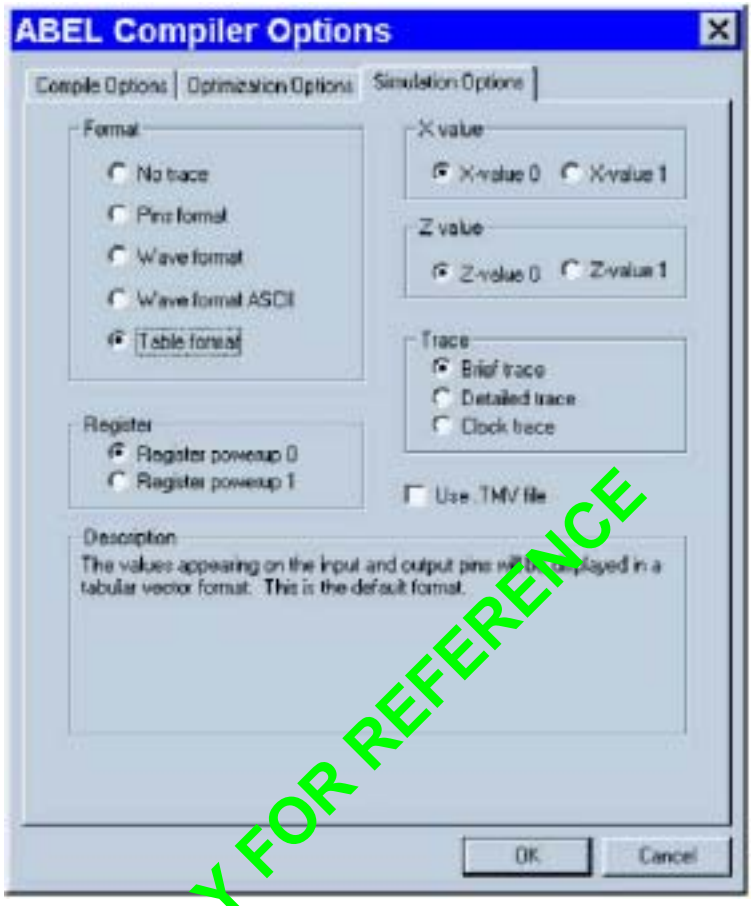
* Reduce by Pin, Auto polarity 此选项与Use Default一样。

* Reduce by Pin, Fixed polarity 此选项将减少逻辑，这样每个信号将有最小数量的乘积项。在可编程极性输出部件中很有用。

No reduction,Merge only(不减少，仅合并)减少逻辑，这样每个信号将有最小数量乘积项，保持信号的极性。当需要把输出信号加到一个指定极性时应该使用这个。

Exact reduction(精确减少)调用Espresso的准确最小选项。这个选项将比标准Espresso产生更完全的逻辑最小化。

单击“仿真选项”会看到下面界面：



仿真选项

FORMAT（格式）选项将影响你的仿真如何被显示：

* No trace（无跟踪） 只显示错误（如果有的话）。无仿真输出产生。

* Pins Format (引脚格式) 将显示器件的每个引脚的测试向量。每一个测试向量的输入和输出引脚上的值将被显示。

* Wave format（波形格式）在仿真过程期间显示每一个指定器件上的输出电平。用每个测试向量的逻辑高和逻辑低表示测试向量的输出级别。这个波形格式通常可以看到但不能被打印。

* Wave formatASCII 此种格式除了可以打印外与上面一样。

* Table fortmat(表格式)除了用H，L和Z代替相对应的逻辑高，逻辑低和高阻抗外与波形格式很相象。这是系统默认格式。

寄存器选项包括：

* Register powerup 0(寄存器上电为0)在仿真开始之前所有的寄存器置为 0 (LO)。

* Register powerup 1（寄存器上电为1）在仿真开始前所有寄存器为置为 1 (HI) 。

X和Z值选项被用在测试向量为“不关心”以及为高阻抗值时，但在仿真期间必须给定一个值的时候。于是，任何时候遇到X为LO时，将X value设定为“X-value 0”。

Trace(跟踪)选项如下：

* Brief trace(简单跟踪)为每一个寄存器的时钟周期或整个系统的输出产生一个仿真结果报告。

* Detailed tacve(详细跟踪)为每一个逻辑产生一个仿真结果报告。这个选项在调试复杂的逻辑电路时非常有用。

* Clock trace(时钟跟踪)当时钟为LO，HI，和再次为LO为每一个向量的寄存器的值产生一个仿真报告。这个选项在为调试异步电路的宏单元跟踪时很有用。

Use .TMV此选项将强制仿真 .tmv文件中的向量。

PSDabel-HDL关键字和语法

这一节祥述PSD的关键字和语法。更多的详细信息参见“Flash PSD CPLD Primer”应用笔记。

允许访问PSD-specific功能模块的关键字为WSIPSD PROPERTY 。

它可以用于连接Databus_IMC和databus_OMC以访问PSD的输入宏单元（IMCs）和输出宏单元（OMCs）。

语法—IMCs

```
WSIPSD PROPERTY 'DataBus_IMC D[s1:s2]: signal[x1:x2] portname'
or
WSIPSD PROPERTY 'DataBus_IMC D s1 ,D s2 ,D s3 ...: x1,x2,x3... portname'
```

语法—OMCs

```
WSIPSD PROPERTY 'DataBus_OMC D[s1: s2]: signal[x1:x2] portname'
or
WSIPSD PROPERTY 'DataBus_OMC D s1 ,D s2 ,D s3 ...: x1,x2,x3... portname'
```

这里

- signal [x1:x2] 是用户自定义的信号名以及下标为X1：X2的信号。x1与x2都为整数值。
- D [s1:s2]是包含的数据总线的下标范围。 is1和s2为整数值，它表示数据总线位数。数据总线名必须为“D”。
- s1:s2 对于8位微控制器，其范围小于或等于8。对于16微控制器，其范围小于或等于16。
- DataBus_IMC有效的端口名为PortA/B/C/D。
- 根据其结构，DataBus_OMC有效的端口名为PortA/B/C/D 或 PortAB/BC/C。

至于应该用哪一个请看数据资料。

例子

```
WSIPSD PROPERTY 'DataBus_IMC D[7:4]:mybus[3:0] PortC'
WSIPSD PROPERTY 'DataBus_OMC D[7:4]:mybus[3:0] PortC'
```

目的

DataBus_IMC/DataBus_OMC特性可以让系统把用户定义的信号分配到相应的端口/微单元（当被指定时），并与指定的数据总线位相联系。它们之间是一对一的映射。当端口名没有被指定时，系统将把用户定义的信号分配给设备上可用的资源，在那里指定的数据总线位可以被访问。这在强加一个特定的数据位命令时非常有用，比如当为计数器建立寄存器或逻辑寄存器移位时。

错误检测

* 当指定的数据总线位不能访问（8位或16位），将产生下面的错误信息：
  “In mode, D[m1:m2] bit(s) is/are not accessible”（“在该模式中，D[m1:m2]位不能访问”）
* 当指定的端口名与已分配的相冲突，将产生下面的错误信息：
  Conflicting port name assignment in DataBus_IMC/DataBus_OMC property”.
  （“DataBus_IMC/DataBus_OMC中端口名分配冲突”。）
* 数据总线位总数和信号位总数不相等时，将产生下面的信息：
  “Inconsistency in number of bits in DataBus_IMC/ DataBus_OMC property”.
  （“DataBus_IMC/ DataBus_OMC数据位冲突”。）

更多实例

例1：在端口B上分配信号到IMCs

```
WSIPSD PROPERTY ’DataBus_IMC D[7:0]:key[7:0] PortB’
```

注意：

■ 如果没有预先定义管脚和节点，key7和key0将分配到Port B因此管脚的分配可能是多余的，并且可能会引起错误。

■ 这些信号只能分配到节点，管脚可以用作其它用途

■ 默认情况下，IMC配置为透明输入

Example 2: 分配信号到OMCs

```
WSIPSD PROPERTY ’DataBus_OMC D[7:4]:sreg[3:0] MCELLAB’;
```

■ PSDabel 文件中要求有 “sreg”的子信号名称定义，优先于PROPERTY描述

■ sreg3 to sreg0是MCELLAB的保留节点

如果希望将OMCS分配到特殊的管脚，可以用以下的语法

```
sreg3..sreg0 pin; or
sreg3..sreg0 pin 24, 23, 22, 21;
```

注意：

如果想把OMCS指派到特定节点，要确保节点与OMCs 所属端口匹配

要想知道如何使用IMCs 和OMCs各种功能，可以在参见“Flash PSD CPLD Primer”应用笔记

测试向量语法

测试向量就是用来测试PLD编码的一种方法。以下规则可以用来测试向量：

- 被测试的信号必须在ABEL文件的 “Declarations section” (定义部分)出现过
- Test_Vectors为开始向量测试
- 向量中的所有输入和输出信号都必须包含在括号中，输入和输出用“->”符号分开。看如下例子：
- 放在括号中的输入应用逗号分开
- 每个测试向量用逗号分开，都包含在括号中，输入和输出用“->”符号分开，并以分号结束
- 测试向量个数和组数无限制
- 测试向量应该放在 “Designated Preserved” (定义区)的位置

- 左边的输入包括逻辑-0和逻辑-1，“don’t cares”，高阻抗，以及时钟值。
  输出都是期望值，可以是逻辑-0，逻辑-1，“don’t cares”。
- 输入和输出可以是二进制，八进制，十六进制和十进制。以下例子显示二进制和十六进制。

下面的例子种可以看到2进制和16进制的输入和2进制的输出。

Example:

```
// Begin user preserved equations (not affected by iterations of DA usage)
Test_Vectors
// Test the state machine, trim, and boost signals
([clkin, reset, begin_cycle, MLEVEL, DLEVEL] ->
[Start_Conv, Intrn, STATE1, STATE0, Trim, Boost])
[ X, 0, X, ^h3, ^h4 ] -> [ X, X, X, X, 0, 1 ];"system in reset
[ 0, 1, X, ^h4, ^h4 ] -> [ 0, 1, 0, 0, 0, 0 ];"system out of reset
[ C, 1, 1, ^h5, ^h4 ] -> [ 1, 1, 0, 1, 1, 0 ];
[ C, 1, 1, ^h5, ^h4 ] -> [ 0, 1, 1, 0, 1, 0 ];
[ C, 1, 1, ^h5, ^h4 ] -> [ 0, 0, 1, 1, 1, 0 ];
[ C, 1, 1, ^h4, ^h4 ] -> [ 0, 1, 0, 0, 0, 0 ];
[ C, 1, 0, ^h4, ^h4 ] -> [ 0, 1, 0, 0, 0, 0 ];
// End user preserved equations (not affected by iterations of DA usage)
```

Note that the X and C that appear above in the test vectors would have to be defined in the Declarations section of the ABEL file as follows:

```
C = .C.;
X = .X.;
Z = .Z.;
```

到REPROT菜单，选择Simulate Resulte，仿真将被调用.至于仿真的结果和输出格式将按照仿真选项的配置操作，具体操作在前面介绍过。

何时使用设计助手，何时使用ABEL

大多数情况用设计助手应该是可以完成整个设计的，而不必到ABEL。但是有些时候不可避免地要用到ABEL，包括以下情况：

■定义一组信号，例如: `Group = [signalA, signalB, SignalC];` or `Group = [Signal3..Signal0];`

■在语言中用到组的时候，例如，如果你先前定义了A和B是分别是2个4位的输入组，而需要C是一个4位的输出组，必须用ABEL写这个语句: `C = A + B;`

■在定位输入/输出宏单元到MCU/DSP数据总线将用到WSIPSD PROPERTY语句（前面曾经介绍过）。

■当需要完成复杂的逻辑功能，要求从其他的逻辑单元返回项时，例如机器状态、移位寄存器、计数器还是用ABEL语言有把握一些 ，尽管他们也可以用设计助手来完成。

附加 PSD 设置

设置安全位。在这个选项里，可以设置PSD的安全位。如果安全位被设置，所有编程工具将不能访问PSD的内容。也就是说，任何上载PSD内容（包含存储器和配置）的操作都会是无效的。只有选择整片擦除操作才能擦除安全位。注意：编程窗口下的校验和计算是无效的。

使能 EPROM/SRAM 低功耗模式（仅适用 PSD3XX/211 系列）。

选择该选项可以使EPROM和SRAM在没有进行访问操作的时候进入低功耗模式。

JTAG/ISP（仅适用内部有 FLASH 的 PSD）

在界面下方的Description(描述)中介绍了如何使用，这个操作是与JTAG ISP有关的。

扇区保护（仅适用内部有 FLASH 的 PSD）

在这个界面中可以将PSD需要保护的扇区设置为写保护，此方法可以防止数据的意外丢失。动态的写保护要求使用页面寄存器。网站上有相应的应用笔记例子。

FITTING THE DESIGN TO SILICON

这段介绍了与 fitter 相关的详细I信息，包括：

- Fitter 报告
- 改变FITTER选项
- Fitter运行时怎样处理
- 当FITTER运行后有错误时怎样处理

Fitter 报告

fitter report file (.frp)包括大量非常有用的信息，包括：

- “ Pin Definitions”界面中的信号以及相关的管脚号
- 有关PSD的配置信息
- 地址和数据总线的分配
- 输入和输出宏单元分配（何处可用）
- 乘积（Product）项，I/O，端口使用节点的情况
- 最后所有的简化逻辑方程

这些信息在决定所用资源时是非常有用。另外一方面，你也可以知道还有那些资源可供使用。

点击**Fit Design to Silicon**，如果FIT成功将自动打开FITTERH报告。如果报告没有弹出来，或者在菜单中无效，你可能在fit过程中存在错误，检查你的逻辑文件（.plg）是有错误。

Changing the Fitter Options（改变FITTER选项）

改变FITTER选项的唯一办法是打开ABEL文件，你可以看到**option**菜单，如果你还不清楚怎样打开ABEL文件，看Edit/Add Logic Statements。要改变Fitter 选项，打开**Options->Fitter Options**你会看到以下界面：

Pin Assignment(管脚分配)有以下几个作用：

- Keep Current（保持当前设置）:允许fitter按照Pin Definitions 界面中的定义以及PSDabel中Preserved Decleration中规定的管脚/节点分配
- Try: 允许fitter尝试按照Pin Definitions 界面中的定义以及PSDabel中Preserved Decleration中规定的管脚/节点分配。
- Ignore: 允许fitter忽略所有Pin Definitions 界面中的定义以及PSDabel中Preserved Decleration中规定的管脚/节点分配。

你如果在PSD资源分配上遇到问题，用“Ignore” 运行Fitter是最好的方法，它可以帮你找到可能的fit。然后使用“Keep Current”选框，用户就可以使用返回的Fitter报告中重新分配的管脚定义。

选择“Enable Product Term Expansion”允许Fitter执行乘积（product），建议用户选择该选项。

选择“Perform Register Synthesis”允许Fitter将ABEL设计文件中的SR，JK，T，和D寄存器合成为D-and T-型触发器

Fitter运行时怎样处理

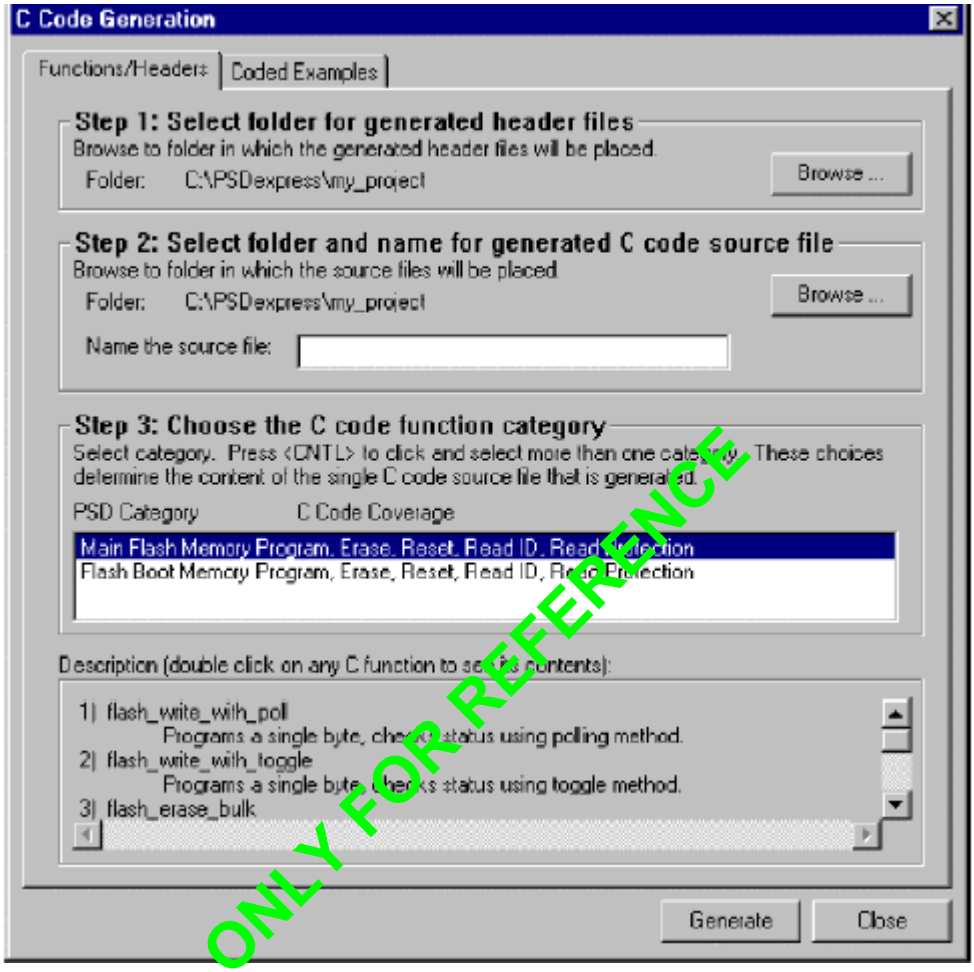
如果Fitter运行超过几分钟，说明你的设计方案可能有错误。有时，通过改变Fitter设计就可以很简单的解决这个问题。通常最好的方法是进入“Pin Assignments”在**Options->Fitter Options**中选择“Ignore”。推荐使用Design Assistant(设计助手)而尽量不要用ABEL文件。

运行Fitter后出现错误怎样处理

如果你一直使用的是Design Assistant(设计助手)，而没有进入手工进行配置的 ABEL 文件，那么在FIT过程中出现错误的可能性就会很小。但是你必须使用手工ABEL文件描述并且出现了错误，通常可以根据提示信息，很容易找到错误。如果出错信息不明显或你还是不能解决问题，请通过EMAIL与apps.psd@st.com联系.

GENERATE C CODE (仅适用于FLASH/EEPROM PSDS)

Functions/Headers



通过该界面的操作可以产生ANSI-C语言的函数和头文件。这些函数可用于PSD的基本操作（写、删除等），还可以帮助你按PSD编码要求整合到你的整个系统设计中。

Step 1: 选择用于创建头文件的文件夹。头文件是通过PSDsoft Express基于C文件产生的。既然在你自己的C文件中将用到这些文件，所以最好这些文件都放在一个文件夹内。点击BROWSE选择放置头文件的文件夹，然后点击OPEN

Step 2: 选择用于产生C代码源文件的文件夹和文件名。首先为将产生的C文件赋一个文件名并点击BROWSE按钮，找到文件夹并点击OPEN

Step 3: 选择C代码函数类型。选择将产生的函数的种例。如果想选择多个选项，先按下CTRL键不放在进行点选。如果想知道产生代码的意思，可以在“Description” 选框中双击函数文件名。

完成这几步后，点击**Generate按扭，**你会看到源文件产生成功的信息以及一些产生的文件。有三个文件将被放进指定文件夹。例如，如果你使用的是PSD4135，那么将产生以下3文件：

<your_specified_name>.c—ANSI-C source for all of the selected functions

psd4135g2.h—ANSI-C—header file to define particular PSD registers

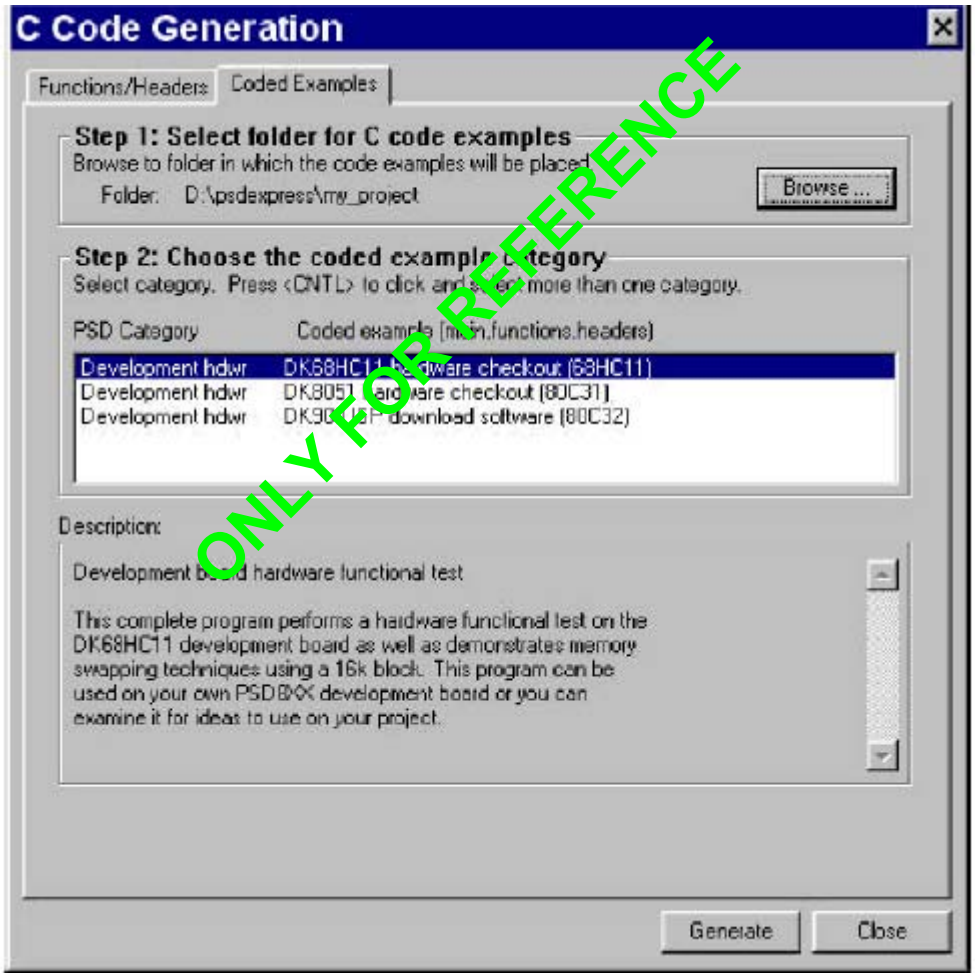
map4135g2.h—ANSI-C—header file to define locations of system memory elements (Main/Secondary Flash memory and PSD registers).

注意：不必给新产生的两个头文件重新命名。因为这些文件在产生的C函数源文件中已经给定文件名。如果你想给新产生的新头文件重新命名，必须保证产生的C函数源文件与头文件名匹配。

现在可以把新产生的三个文件整合到你的编译器环境中。PSD4135g2.h文件包含对<your_specified_name>.C中各个C函数的 #define描述。编辑psd4135g2.h并移动各个C函数#define描述中的注释分隔符（//），你可以把这些C函数与剩下的C源代码编译。

现在你可以点击“Coded Examples”会出现下面的界面，如果不需要可以点击**Close按钮**。

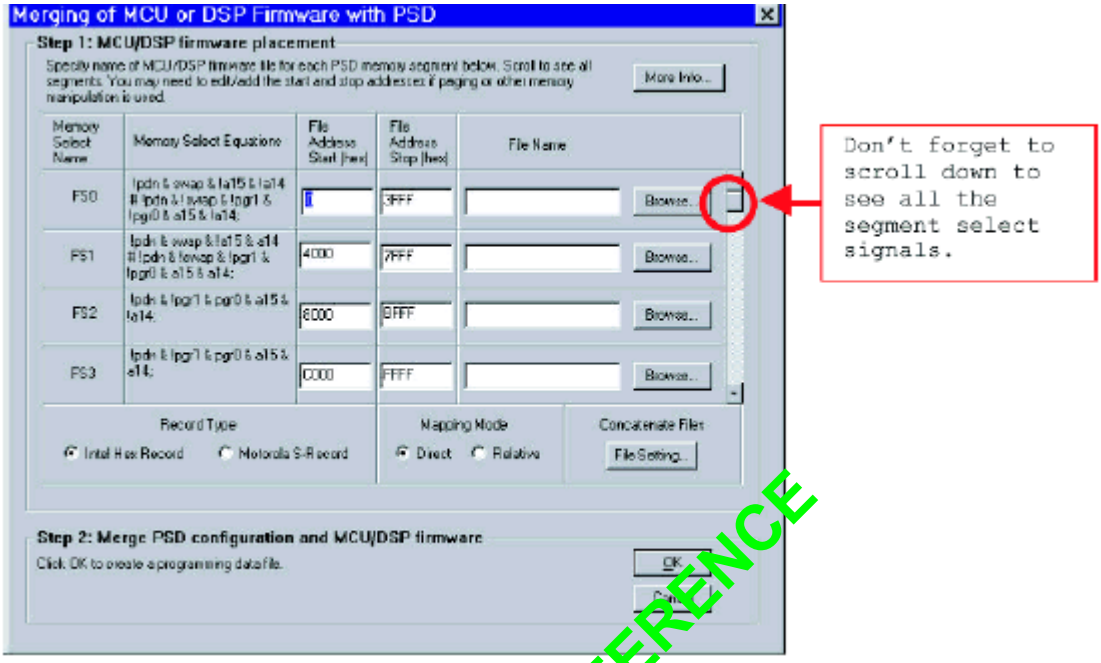
Coded Examples



该界面可以产生PSD开发板的代码。即使你不购买开发板，你也可以通过该文件了解各代码是怎样组合的，比如：UART码，等。

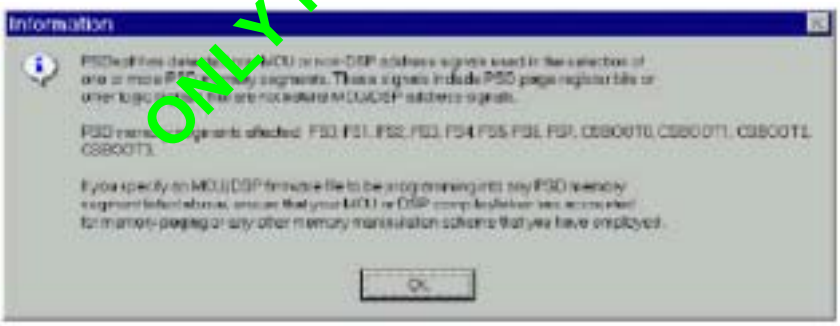
首先在STEP ONE中点击BROWSE按钮，选择希望存放例子文件的文件夹。选择与你最终产品最接近的开发套件，点击Generate，会有一个提示成功的消息框出现，点击CLOSE按钮。

MERGE MCU/DSP FIRMWARE WITH PSD

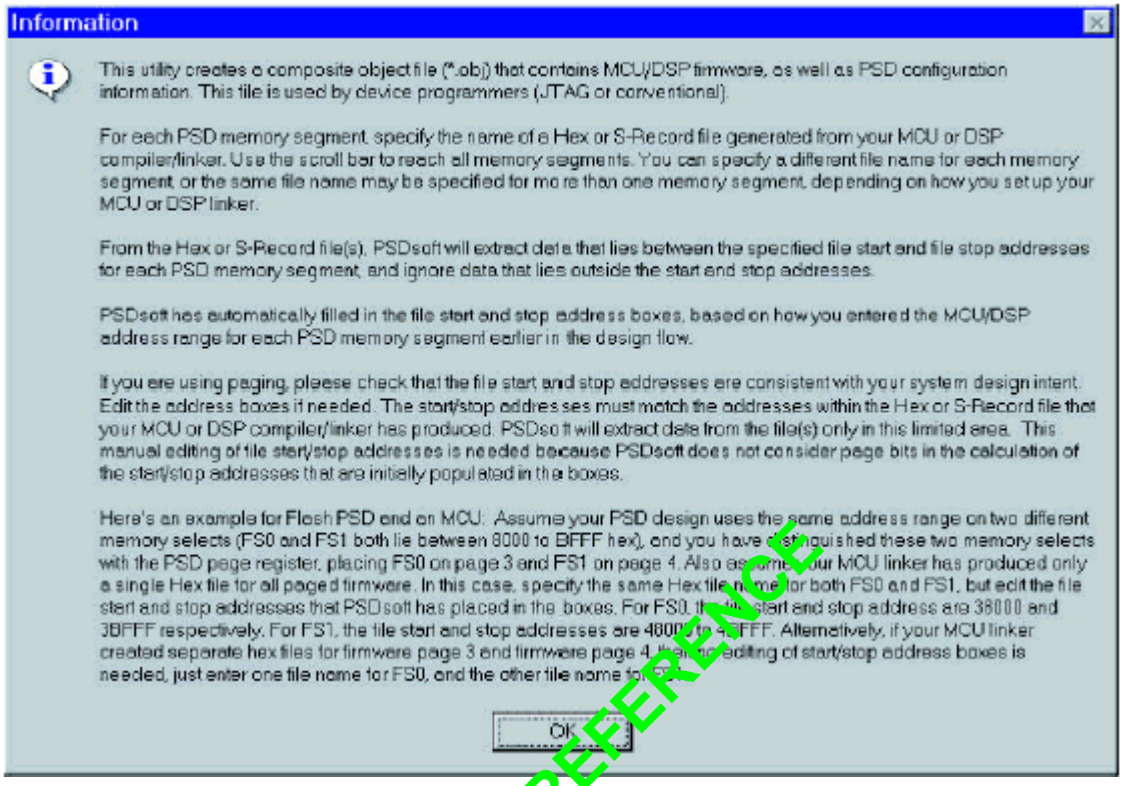


该界面是PSDsoft Express的主要操作部分，在这里将你的编程文件（.obj）和PSD的最终配置整合到一起。仔细进行每一步操作，注意警告和错误信息。

注意：当点击“Merge MCU/DSP Firmware”时，可能会出现以下警告信息。



大体上，这个警告是提示你 PSDsoft Express 检测到了片选等式包含了一些其他的 MCU/DSP 地址信号。提醒你确定在 MCU/DSP 编译/连接过程中考虑到这些。如果 File start/stop 地址没有自动出现，你可能还需要手工添上。同样，对于页设计的情况，也要检验 Start/Stop 地址的有效性，因为软件内部检测不包含页位。如果你想知道更多有关 Merge 单元的信息，在界面顶部点击 More Info…按钮，你将会看到以下界面



第一步

在开始这步之前，必须有一个有效Intel Hex or Motorola S-Record文件或者在编译连接环境下基于源代码生成的文件。也就是用MCUs/DSPs固件文件.obj文件，这些文件后来将下载到PSDs非易失性存储器。产生记录文件有三种方法：

以前在片选方程中定义的PSD内部存储器每段都产生一个记录文件。

存储器几段共创造一个文件名。

第三种方法就是以上两种的任意组合。

现在我们来看一下“Merging of MCU/DSP Firmware with PSD”屏，第一栏表示所选存储器段（segment-select）名（FS表示主闪存，CSBOOT表示次闪存，EES表示EEPROPM，ES表示EPROM）。注意用滚动条来看内部PSD存储器各段。下面一栏表示你为各段输入的方程式。中间两栏表示文件开始和结束地址。要注意的是如果你在方程中使用了非地址逻辑（如分页），PSD将会出现警报，可能不会填写开始结束地址，也有可能会填写。在最后一栏点击BROWSE按钮，选择固件文件，烧写到PSD非易失性存储器。

例如，在屏幕中可以看到Flash存储器段和三个Hex文件。FS0 从0h 到3FFFh有效，FS1 从4000h到7FFFh有效，“my-hex.hex”文件包含这两部分的代码。FS2 和FS3 从C000h到FFFFh都有效（但在不同页），因此含有不同的Hex文件。在代码指定到相同的MCU/DSP物理地址时，不同的Hex文件通常都会用到，但在不同页。因此你必须设置Page Register去寻址。

一旦填上所有的文件名后，通过连接程序选择与输出匹配的相应记录类型(Intel Hex or Motorola S)。

然后选择期望映射模式。“Direct”映射包含MCU/DSP地址和选择内部存储器各部分的地址之间的一一对应。“Relative”映射可以让你指定不同的物理地址（基于固件文件的MCU/DSP输出），而不是PSD内选

择存储器各段的方程。大多数用户都选择“Direct”。

第二步

完成第一步后，如果想让PSDsoft Express完成.obj文件创造则点击**OK**，否则点击**CANCEL**

后台运行

以下操作都是在后台完成

- !pdn（低有效的内部掉电信号)自动包含在PSDs段选（segment-select）方程中，这样就有自动电源关闭特点。
- 如果你选择使用D端口的管脚2 (pd2)作为片选输入(_csi)，那么它将自动包含在段选方程中。
- 如果你选择使用Motorola S-Record作为记录类型，在创造.obj文件之前，PSDsoft会将其转换成Hex记录格式。并把文件保存下来，不过扩展名为.Hex。因此你的Motorola S-Record文件扩展名不可以是.hex。
- 点击OK，进行地址翻译。在这个过程中，记固件将会烧写到相应的PSD非易失性存储器段。在PSD中这些存储器段都有一个地址，与“Chip Select Equations”窗口中定义的不同。基于PSDs 自己内部地址，地址翻译器将创造.obj 文件。如果想知道地址是怎样翻译的，选择**Report->Address Translation**.选择下面的提示会有更多有用的信息。

常见报警和错误信息

你可能会遇到的一些报警和错误信息。如果有信息不清楚，或者没有写出错信息的地址可以发电子邮件apps.psd@st.com

提示

要检验PSDsoft Express是否按照你所希望的正确解释固件文件，以及将你的代码放在PSDs内部存储器段何处。要做这些，在设计流程中点击STMicroelectronics Conventional Programmers。确保你的.obj文件在顶部窗口中。在工具栏点击相应的图标，选择PSD中你想要看的存储器。记住在“Direct Address”栏中看到的地址仅仅是PSD知道的地址。

以下屏展示了所选存储器的代码。没有数据和代码的将显示FFs (blank). “Conventional Programming”更多的信息，看Conventional Programming部分。

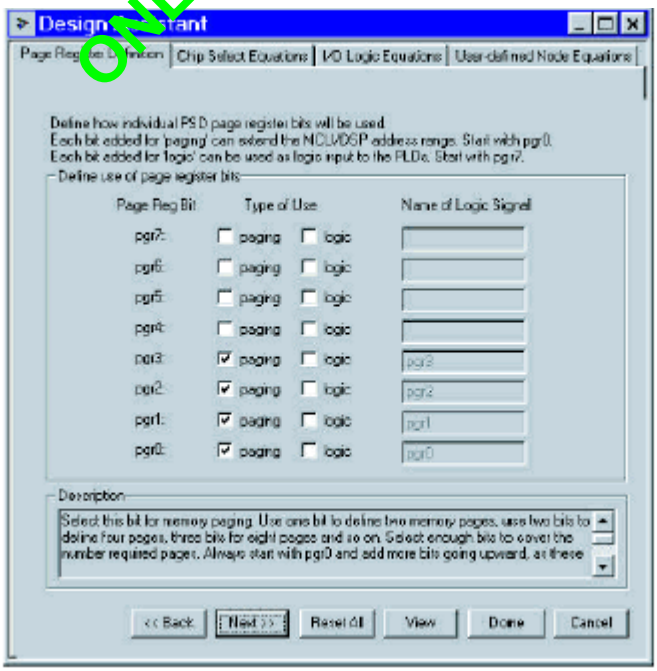
将两个Hex文件分配到单存储器段

为了配合某些MCU/PSD，将两个Hex文件分配到单一存储器上是必需的。如用PSD835 或PSD935（每个都有64K Flash存储器段）和64K地址空间的MCU配合。本质上，每个Flash存储器段都要分成两个32K页。PSDSOFT通过关键字“fa15” 将存储器段分割。为了保持简单，下面的例子说明如何设置“fa15”以至于当页寄存器的第一位置高时(pgr0)，“fa15”也位 “真”。

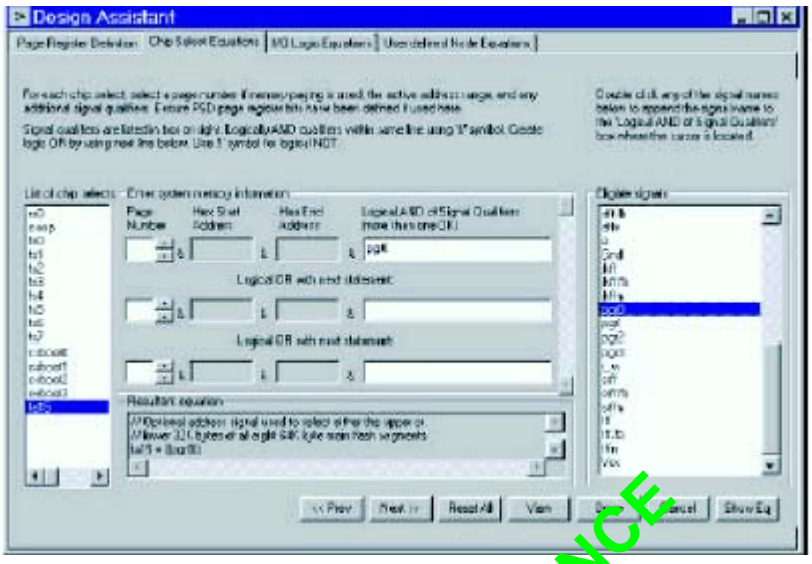
设置页寄存器

如果你想用PSD835/935所有的存储器，而且你有只支持64K地址空间的MCU，你就需要16个32K页面，支持4页位($2^4 = 16$)。

以下的屏幕描述的是怎样通过“Design Assistant”完成分页。



建立 fa15 bit. 以下屏幕演示了如何在页寄存器的第一位为真时，将fa15 bit设为真。

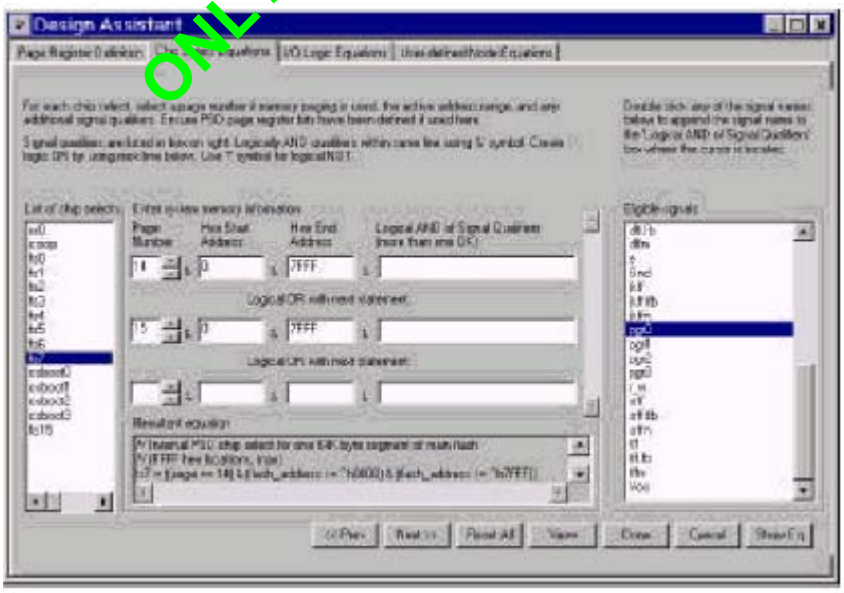


建立Flash存储器方程

找到每个需要分成两个32K逻辑段Flash存储器扇区。然后，当MCU操纵页寄存器时，fa15将自动设置，Flash存储器的正确部分将被访问。下面将演示fs0。

设置融合MCU/PSD固件到PSD

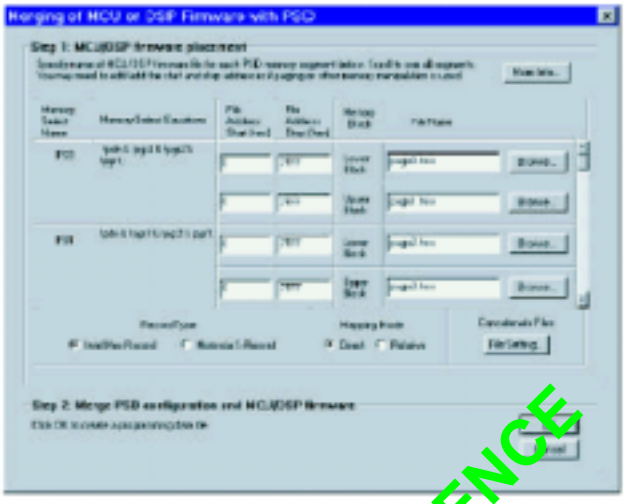
最后一步是正确的设置“Merging of MCU or DSP Firmware with PSD”屏。注意你的连接程序要把代码分成两个32K页面。作为参考，下面将演示fs0和fs1。



注意：Hex文件或许包含指定范围外的数据(0 – 7FFFh in this case)，因为融合工具只从0 – 7FFFh范围内提取数据。

结合Hex或 S-Record文件

PSDsoft提供组合Hex or S-Record files的功能。通过以下的“MCU/DSP Firmware File Concatenation”屏幕可以看到，组合只有简单的几步。



以下是组合两个或更多文件的步骤：

1. 选择文件类型
2. 在Step 2点击**Browse**按钮，搜索第一个文件。这将打开“Open”窗口。
3. 找到你的文件，选择**Open**.
4. 文件将在Step 2 的“Select Folder and File”列出。
5. 点击Add，第一个文件将会在表中列出来
6. 重复Steps 2 到 5直到表中列出你所要的所有文件

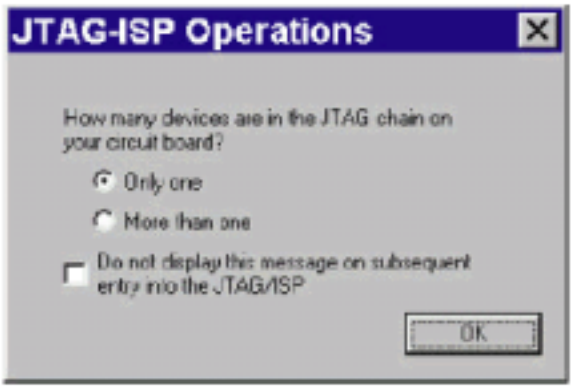
注意：文件将在File Name中按顺序组合。如果你想改变顺序，用Move up and down，如果你想删除某个文件，在窗口中左击文件，然后点击**Delete**.

7. 在Step 3中点击**Browse**，然后，将会出现“Open”窗口。
8. 在“File Name”中键入文件名，点击**Open**. 文件名和位置将会出现在Step 3的“Select Folder and File”中
9. 点击**Concatenate**组合文件，你将在屏幕底部看到状态提示信息

注意：**Save Setup**可以保存屏幕当前状态，**Retrieve Setup** 可以回到上次保存的部分

PSD JTAG/ISP (FLASH-BASED PSDS ONLY)

在设计流程中点击“STMicroelectronics JTAG/ISP”，将出现以下对话框（除非你提前设置过对话框）

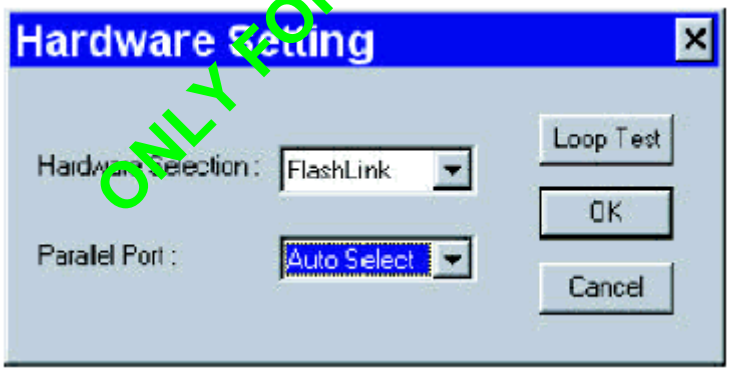


在JTAG 链中选择器件数量，如果器件数量始终为同一值，可以选择“Do not display this message on subsequent entry into the JTAG/ISP”.

注意：可以在**Project->Preferences**菜单中查看信息。如果你还不了解JTAG链，先看JTAG链定义和规则。我们推荐使用JTAG链端口的每个用户都阅读“JTAG-ISP Information for Flash PSDs”应用笔记。如果你以前使用过该窗口，想回到JTAG-ISP设置，可以跳到Save or Retrieve JTAG-ISP Setup.

硬件设置

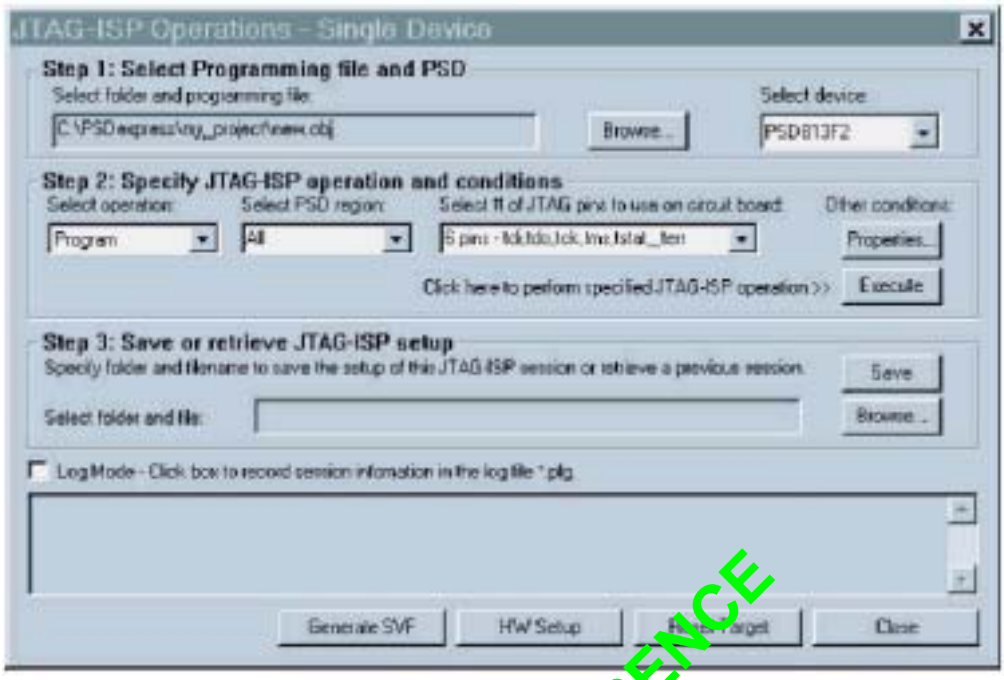
如果你是第一次使用FlashLINK cable，或者是换了计算机或硬件，首先应设置硬件。在窗口底部点击**HW Setup**进行设置，过程中将弹出下面的窗口。



根据计算机的不同，不能确定是否能改变Parallel Port的设置，“Auto Select”在大多数情况下都行，除非你的端口没有正确安装。在FlashLINK cable的任何JTAG操作之前运行Loop Test是不错的选择。运行之前，确保所有loopback connector 连接到FlashLINK、电源打开、接有地线。然后点击**Loop Test**按钮进行测试。你已经进行了JTAG操作，但通过测试并不表示你已经成功。只要在JTAG中安装有一个器件，继续读后面部分。否则跳到JTAG Chaining 部分。

单个PSD器件的JTAG-ISP操作

如果在问到“在你的电路板上有几个JTAG 器件中”选择了“Only one”，你将看到下面的对话框。



Step 1: Select Programming File and PSD

如果你已经打开一个有效的工程文件并且已经创建好了程序数据文件（.obj），那么这个窗口将打开与当前工程对应的OBJ件。如果你没有打开任何工程文件或希望打开另外的.obj文件，那么你可以在Step 1中点击**Browse…按钮，一旦你选定了一个.obj**文件，对应的芯片类型将自动上载。如果由于某些原因，你可在这一步里改变器件类型，如果你是在同一系列选择相似的器件，通常不推荐这样做。

Step 2: Specify JTAG-ISP Operation and Conditions

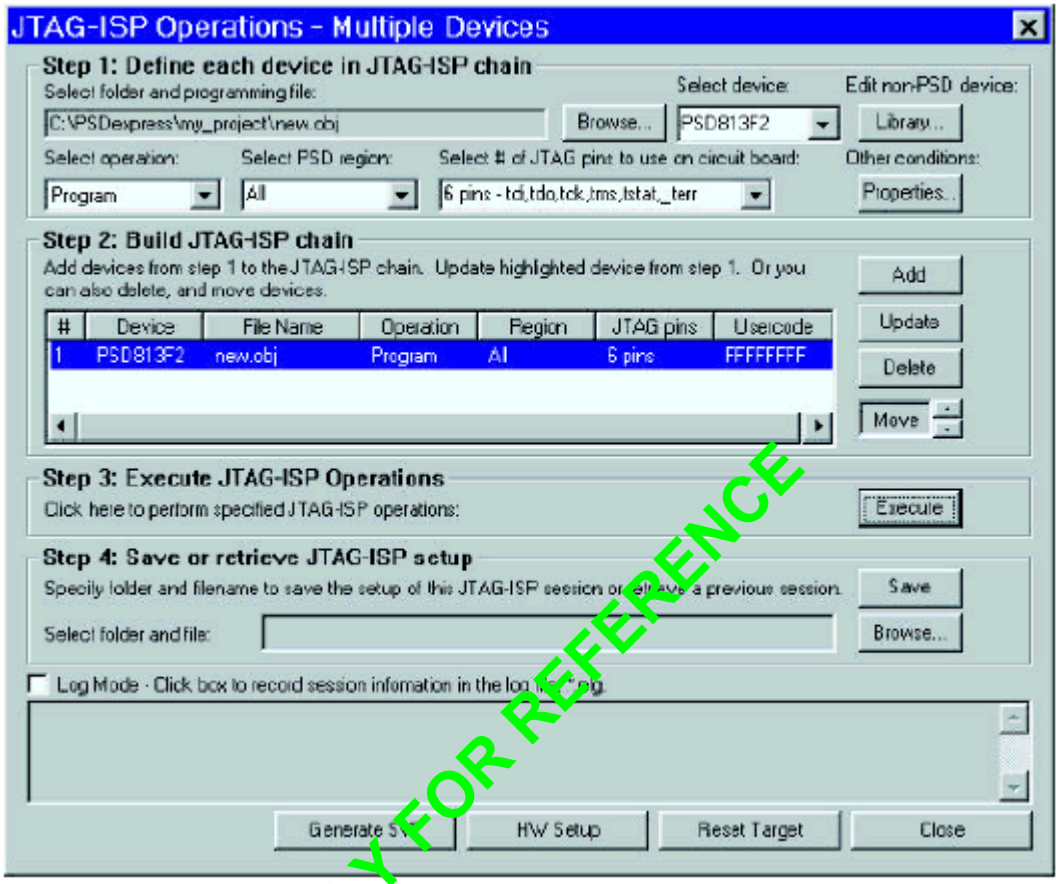
首先从下拉菜单中选择所要求的操作。有以下项可供选择。

- Blank Test—检查PSD是否已经编过程
- Erase—擦除PSD的一个或多个区域
- Program—根据.obj文件，将代码、配置、数据写到PSD的一个或多个部分
- Verify—将用PSD内部编程的实际.obj内容与期望上载到PSD的.obj文件内容比较
- Upload—将PSD内容上载到.obj文件。注意：如果已经设置加密位，上载将是无效操作。删除加密位的唯一方法是执行整片擦除。
- Bypass—将器件以旁路模式放置。既然你只用一个器件，除了迅速检查你的JTAG接口是否工作正常外没有其他作用。

下一步，选择PSD操作区域。注意：如果选择“All”项而且只编程PSD的要求区域，在编程之前，软件会查看.obj文件。例如：如果在Merge MCU/DSP Firmware过程中，你只为fs0 (main Flash memory, segment 0)指定了固件文件，那么就只有PSD的fs0 和PLD部分会被编程。根据前面的选择在Pin Definitions屏幕选择JTAG管脚数。默认值是你在Pin Definitions屏幕指定的管脚数。最后点其**Properties…**按钮，用来设置JTAG过程中PSD的 I/O、查看JTAG属性、检验JTAG用户代码。当你对一切都满意后，点击**Execute**。如果你在执行过程中出现错误信息，查看错误和警告信息。完成第二步后，跳到Save or Retrieve JTAG-ISP Setup。

多个器件的JTAG-ISP操作

如果你在JTAG chain中选择了“More than one”，会看到例似如下窗口：



Step 1: 定义JTAG-ISP链中的每个设备

增加PSD设备

如果你打开其中已包含了obj 文件的工程，该窗口将会打开当前工程中的obj 文件，还会为你选择PSD器件。如果你还没有打开任何工程，想打开其他的obj 文件，在Step 1中点击**Browse…**

一旦你打开obj 文件后，合适的器件将会自动加入。点击**Add** 按钮向链中添加器件。

从下拉菜单中选择你想要的操作，有以下几项可供选择：

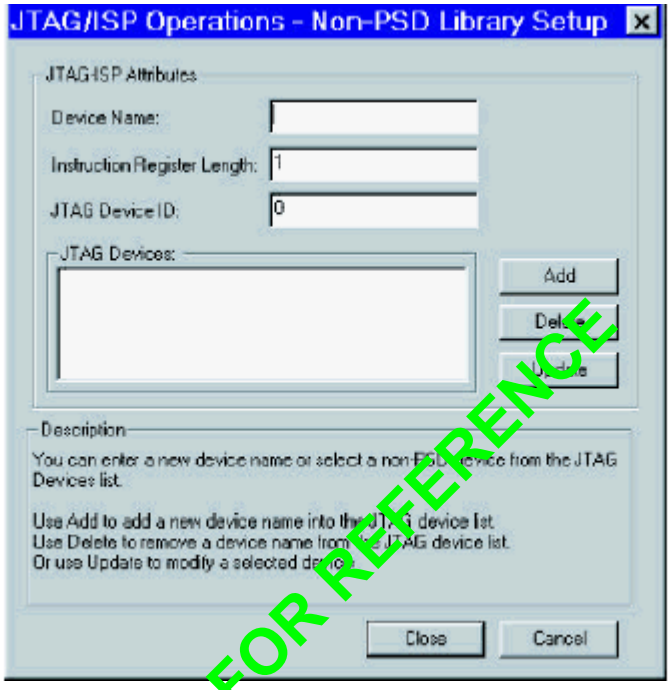
- Blank Test—检查是否PSD的某些区域已经编程
- 擦写—擦掉PSD的一个或多个区域
- 编程—根据.obj文件，将代码、结构、数据写到PSD的一个或多个部分
- 校验—将用PSD编程的.obj实际内容与常见打开的.obj文件内容比较
- 上载—将PSD内容转储到.obj文件。注意：如果加密位已经提前设置，上载会产生不可用文件。删除加密位的唯一方法是全片删除。
- 旁路—将器件以旁路模式放置。

接着，选择PSD区域。如果选择“All”项，而且只要求编程PSD的指定区域，在编程之前，软件会查看.obj文件。例如：如果在Merge MCU/DSP Firmware过程中，你只为fs0 (main Flash memory, segment 0)指定了固件文件，那么就只有PSD的fs0 和PLD部分会被编程。根据前面的选择在Pin Definitions屏幕选择

JTAG管脚数。默认值会反映你在Pin Definitions屏幕指定的管脚数。最后点其**Properties…**按钮，用来设置JTAG过程中PSD的 I/O、查看JTAG属性、检验JTAG用户代码。

Adding Non-PSD Devices

只有以前通过点击**Library**定义后，非Non-PSD器件才可以加到链中。点击后，将会看到以下窗口：



输入所要器件名。该名称将会出现在主屏幕的下拉菜单。接着用十六进制输入指令寄存器长度（Instruction Register Length）和JTAG Device ID。你可能要从厂商获取这些数字。输入器件所有信息后，点击**Add**按钮。在“JTAG Devices”中你可以看到器件。你还可以用该方法添加更多的器件。如果你想从列表中删除器件，点亮器件，点击**Delete**。如果你想改变预先定义器件的信息，点亮器件，改动后，点击**Update** button。添加更新完毕后点击**Close**回到JTAG主窗口。一旦你定义了non-PSD器件，用“Select Device”下拉菜单选中它，点击**Add** button。注意：对于non-PSD器件，唯一有效的操作是Bypass，JTAG管脚数也只能是四。如果你想让PSDsoft Express检验以前输入的JTAG Device ID，点击**Properties** button，然后点击JTAG Device ID.

Step 2: 构造JTAG-ISP链

每次点击**Add** 按钮，器件都会按照选择的先后顺序添加到链中。顺序必与器件在链中的物理顺序一致。如果你不清楚其物理顺序，可以在 ***JTAG-ISP链的定义和规则*** 中查看。如果器件顺序不对，可以点亮链中需要移动的器件，然后点击Move Up或者 Move Down 箭头。如果你想要删除某个器件，先点亮，然后点击**Delete**。如果你想改变操作（OPERATION）、管脚数、器件的属性，先点亮器件，然后选择相应的操作（OPERATION）、管脚数，或者点**Properties…**完成后点击**Update**。注意：Device #1必须是TDI管脚与**FlashLINK cable**相连的板子上的第一个器件。***请参看JTAG-ISP链的定义和规则.***

Important note: 如果你想改变操作（operation）、PSD区域等，必须在Step 2中点亮改动行，而且还必须在Step 1中改动之前。改动完毕，在Step 2中点击**Update**。如果只改动操作（operation）或者属性中的一个，可以右击要改动的行。

Step 3: Execute JTAG-ISP Operations

如果前面两步已经完成，点击**Execute**。如果在执行过程中出现错误，看错误和报警信息。如果你有超过一个以上的操作（operation）或者有一个以上的器件，回到前面的步骤，更新和继续执行，直到完成。

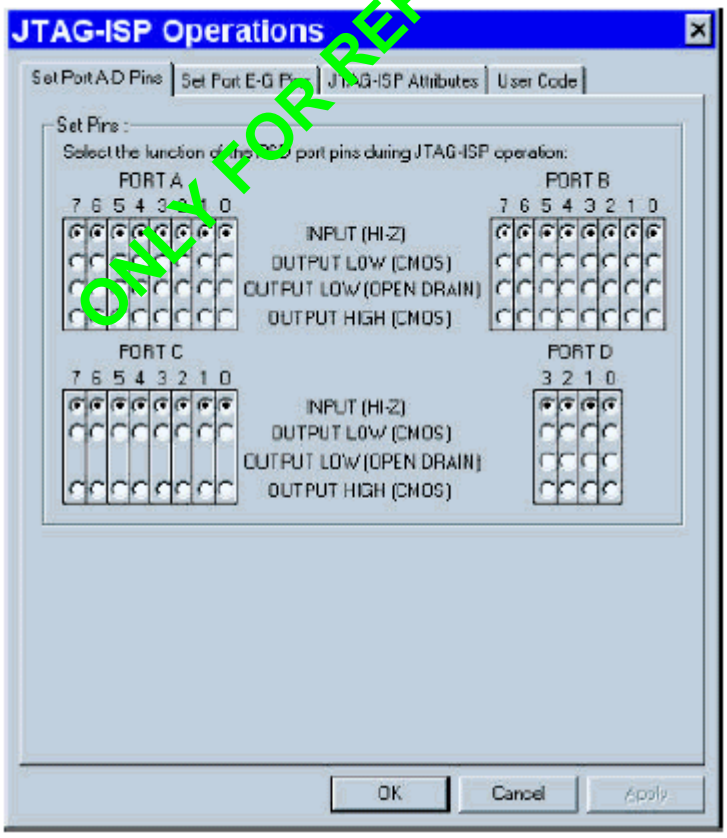
如果这是你第一次为当前工程（project）使用“JTAG-ISP Operations”，点击**Save**按钮，将会出现“Save As”窗口提示你输入文件名。完成后点击**Save**按钮或者点击**Cancel**不保存退出

找回设置

如果你之前为当前项目保存了JTAG设置，点击**Browse…**在”Open”窗口中点**Open**

Properties

选择PSD器件后点**Properties…**将会出现类似以下的窗口。如果你点non-PSD器件的按钮，看JTAG-ISP属性.

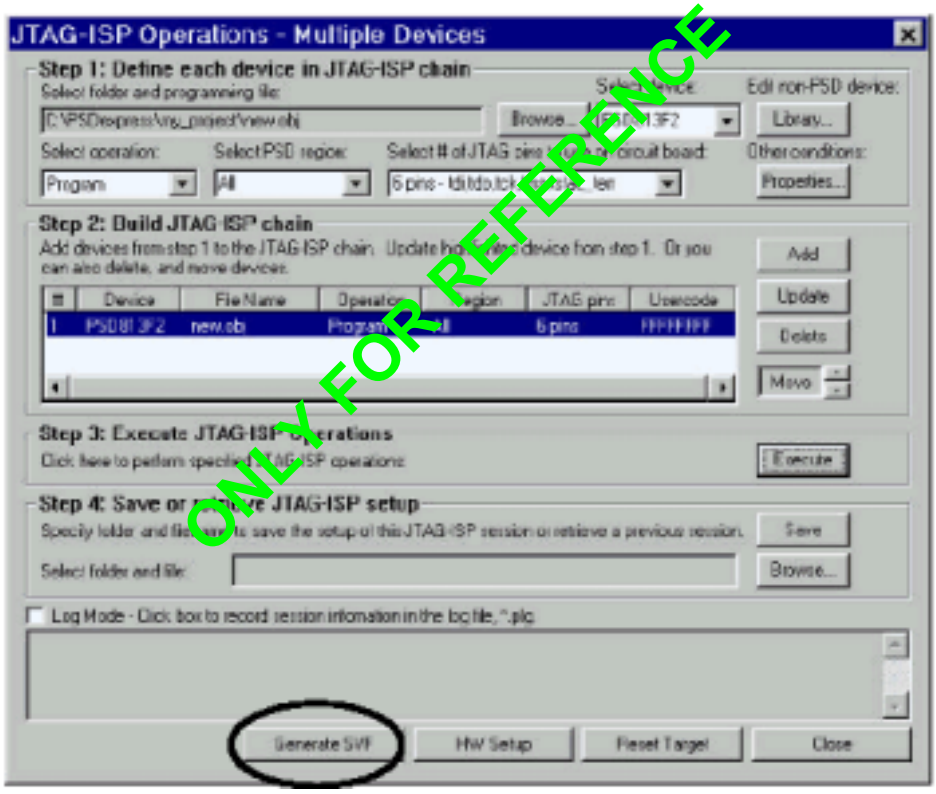


Set Port Pins. 根据你的PSD，在JTAG过程中，最前面的一个或两个tabs专用于设置PSD的 I/O管脚。默认（JTAG除外）为输入，对大多数管脚都比较好。（注意：PSD对non-JTAG I/O没有响应）。然而有时候还是希望为在JTAG过程中，设置为输出管脚。例如，如果你有PSD产生的片选信号，你希望在JTAG过程中该器件片选无效。

JTAG-ISP Attributes. 如果为PSD器件点击“JTAG-ISP Attributes”，你将会看到器件名和指令寄存器长度（Instruction Register length）。这些信息对其他的实际程序非常有用（没有“auto-detect”功能）。

User Code. 基本上，通过点击“User Code”，将为你提供输入IEEE 1149.1 用户代码的空间，用户代码可以与以前在Additional PSD Setting输入的值相比较。

Generating a SVF File 通过点击**Generate SVF**，PSDsoft 可以产生Serial Vector Format文件(.svf)



点击**Generate SVF**后，产生两组文件：过程文件和数据文件。.svf文件是表示通过JTAG端口编程PSD的的必需步骤。如果你想通过你的MCU或者DSP使用该文件，需要解释.svf文件中的低电平命令，并写出相应的代码。数据文件是为PSD更新的代码值。因此文件保存以后，为你的MCU/DSP写的软件流程应该根据JTAG屏幕上的状态，控制四个或六个JTAG管脚。也就是编写用来模拟.svf文件行为的代码。这就需要通过JTAG lines重新编程PSD。这些可以由PC机通过使用数据文件甚至板子上的另外的MCU完成。

Behind the Scenes

无论你什么时候点击**Execute**，在执行期望的操作之前，都会先执行以下操作：

- 并行端口被驱动器接管，由驱动器检查PC机的并行端口与FlashLINK cable之间是否有通信正常。
- 检查User Code和Device ID(s)
- 发送一些信号，并且在几个TCK时钟周期接收到反应信号。

  在进行期望操作之前，以上的每一步都可能会产生错误信号。如果有错误信息，看下面部分。

Error and Warning Messages

如果有错误信息，可采取以下措施：

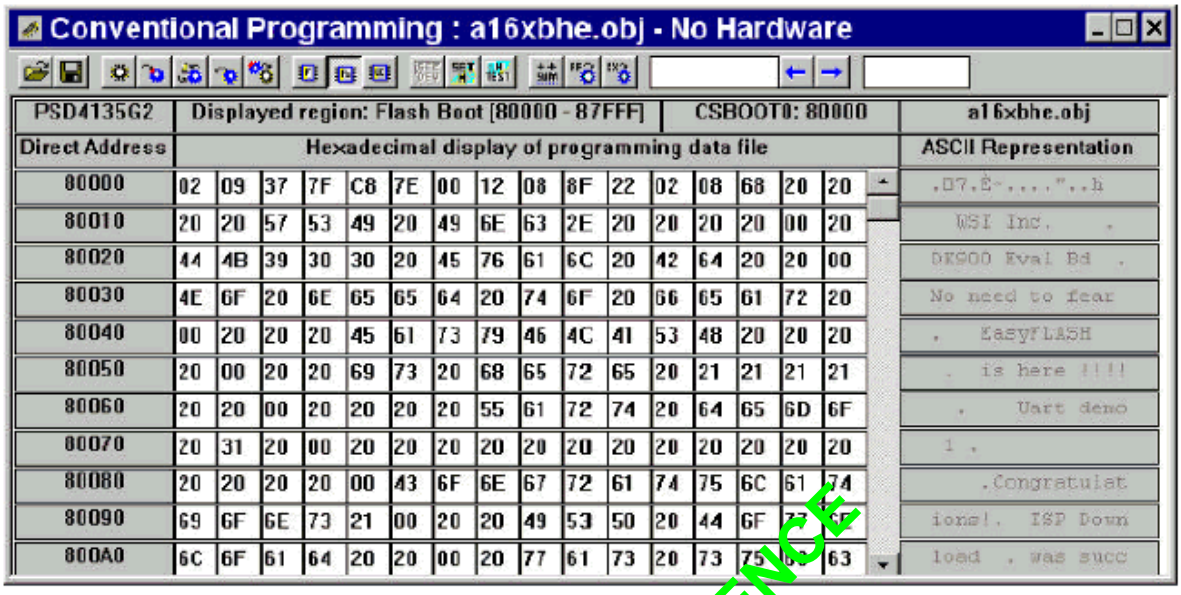
1. 检查错误是否在PSD JTAG FAQ网站上列出
2. 如果错误没有列出，或者你还有其他问题，与apps.psd@st.com联系

提示与线索

批量JTAG 下载程序

你可能正在想这个软件真是太棒了，可以让你的系统启动运行（up-and-running），但是如果你有大量板子想编程并想执行批处理程序怎么办？FLINK程序可以满足要求，而且可以在PSM软件中心免费下载。

PSD 标准编程器



有关这个窗口最重要的事情是：如果你已经打开一个项目并且在项目中创造了有效编程数据文件(.obj)，编程数据文件将会自动加载并显示。从上面屏幕可以看到，工程名为“a16xbhe”，我们视为.obj文件。

工具条

接着，注意有效工具图标，该图标可以演示以下功能（从左到右）。将鼠标放在图标上可以演示其功能

- Open file—如果你没有打开带有有效obj文件的项目，或者你想看其他obj文件时可以点击
- Save file—允许你保存对obj文件的任何修改。这表明obj文件可以在该窗口直接修改。修改时，只需要将指针放在新值的位置和分类中（你将会看到原来值被自动覆盖）。注意，在obj文件中修改任何字都会改变checksum。通常不推荐直接修改obj文件。任何修改都必须原始资料中进行，而且新的记录文件都必须利用compiler/linker环境产生
- Blank Test—检验PSD的某些端口是否已经编程
- Program—将代码或者数据写到PSD
- Verify—将PSD的实际内容与加载的obj文件比较
- Upload—将PSD的内容转存到另外的一个obj文件。注意：如果提前已经设置加密位，上载将产生不可用文件，删除加密位的唯一方法是全片删除。
- Erase (Flash memory/EEPROM only)—删除PSD的一个或多个部分

下面的两个或三个按钮允许你在PSD范围内为各种非易失性存储器查看obj文件内容。注意：F代表主Flash存储器，“Fb”代表次Flash存储器，“UC”代表用户代码，“E”代表EPROM，“EE”代表EEPROM

- Select Device—允许你选择哪个PSD器件与新obj文件一起使用。只能在新obj文件已经打开才能使用（从“File”菜单）。新文件保存后选项变灰。

- Hardware Setup—允许选择编程器硬件和并行端口
- Hardware Test—检查编程器硬件确保连接和工作正常，第一次插入新硬件都应该运行测试。
- Checksum——计算基于通常加载的obj文件的非易失性存储器的检验和。PSDsoft使用简单的二进制加法。因此只有细微差别的两个obj文件有可能得到想同的检验和，但是数据越大，这种可能性就越小。
- Fill blank—PSD内非易失性存储器的默认“un-programmed”值为逻辑高或“1”。点击该按钮将PSD内所有非易失性存储器填充到“1”，这就等效于擦这些存储器。
- Fill value—有时可能需要将某个值或者一些值填写到PSD内非易失性存储器，例如用于调试。点击按钮，弹出一个你可以在其中选择填写哪些存储器、开始和结束地址以及填写值的窗口。
- 工具栏最后两块用于模式搜索，左边一个用于搜索上一个，右边一个用于搜索下一个。

显示

工具栏的下一行包含以下信息：

- 与.obj文件有关的PSD型号
- 你在工具栏中选择的PSD非易失性存储器
- 所选非易失性存储器中当前显示的段
- 当前打开的.obj文件名

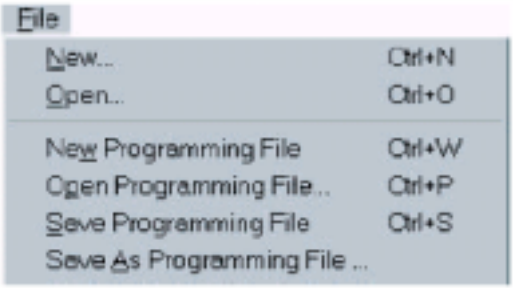
下一栏显示下列信息：

- **Direct Address**地址栏中的地址是内部PSD原始地址，这些地址与当前显示的非易失性存储器有关。如果你想弄清楚直接地址是怎样翻译成MCU/DSP输出的系统地址，就到**Report->Address Translation**.
- 十六进制数据文件逐字显示你的固件
- 最后一栏显示上一列的ASCII码

菜单

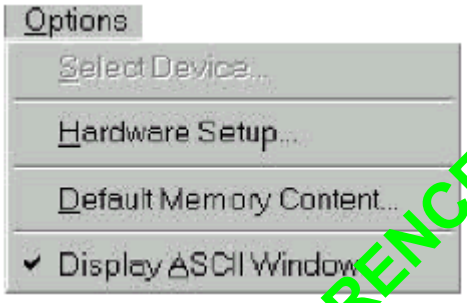
当“Conventional Programming”为活动窗口时，有三个新菜单。下面就看以下这些菜单。

下面的“File”菜单允许你创造、打开和保存编程数据文件。



功能菜单中的功能与上面的类似

下面是“Options”菜单

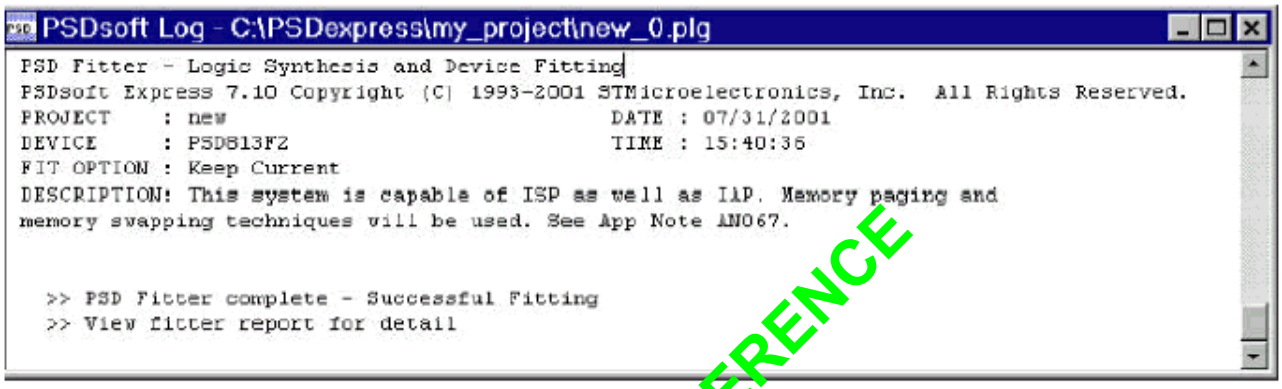


“Select Device…”和“Hardware Setup…”上面已经叙述。“Default Memory Content…”会打开一个新的窗口，你可以在其中用十六进制添上所有非易失存储器。“Display ASCII Window”允许你打开和关闭固件文件的ASCII表示法。

PSDSOFT 日志文件

PSDsoft Express的日志文件记录设计过程中的错误和报警等信息。在**Project->Preference**中你可以规定系统中的日志文件数（最多十个）。文件格式为ProjectName_X ，其中X为0到你在Preferences中指定数中间的任何整数。写入日志文件的最大数后，X重新置0。

就象上面所说的，可以在“Report”菜单中看日志文件。在设计过程中，你随时都可以检查文件，以观察你的设计状态和和可能出现的信息。如果在使用PSDsoft Express过程中出现错误信息，可以将你的日志文件和设计文件一起发送到apps.psd@st.com并附上解释。



定义片选、I/O口逻辑和节点方程的区别

如果你还不能确定I/O或者内部节点该选择什么类型的方程，可以使用以下向导：

- Chip-select equations方程通常解码地址行和可选控制逻辑或者其他的PLD输入。例如，PSD内存储器段的内部段选。
- User-defined nodes通常用于临时存储和PSD内指定的输入输出宏单元。
- I/O Logic是要求输出使能信号。

例如，假设你的设计有以下需求：

- 8-bit可加载的移位寄存器
- LCD模块的片选信号
- 可通过MCU/DSP直接访问的I/O管脚。

设计时，8位移位寄存器的7位被定义为内部节点，节点方程写在设计助手的“User defined Node Equations”部分。移出位在“ Pin Definitions”屏幕中定义为CPLD输出，其方程写在设计助手的the I/O Logic Equations部分。通过将节点和PSD内输出宏单元联系起来，移位寄存器可加载。因此，可以由MCU/DSP通过数据总线使用PSD’s CSIOP寄存器直接访问移位寄存器。LCD芯片必须先在Pin Definitions屏幕定义为“External chipselect – Active Hi/Lo”，然后才能使用设计助手的Chip Select Equations写入其方程，最后通过MCU/ DSP可直接访问的I/O管脚在Pin Definitions屏幕被简单地定义为MCU I/O模式，其它的在运行时完成。

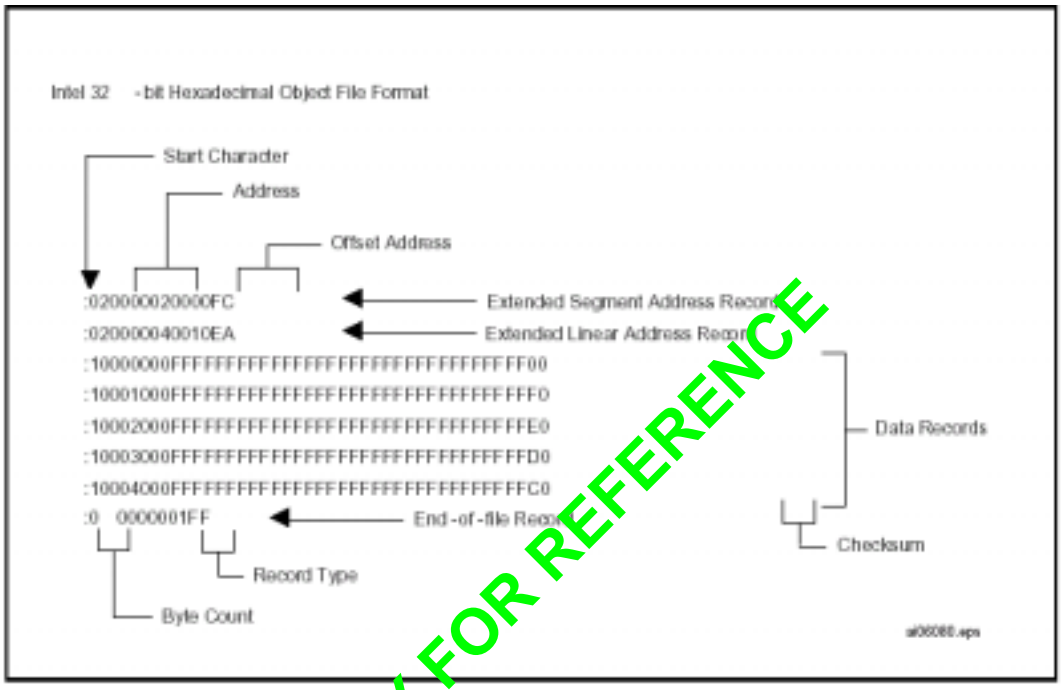
- “Flash PSD CPLD Primer”应用笔记详细解释了PSD的PLD部分和它的许多用途。包括设计时要做什么和运行时会发生的情况。
- “Design Guide for PSDsoft 2000和 PSD4235详细介绍了用PSDsoft Express设计PSD4235G2的实例，是CPLD-based PSD非常好的方法。
- 在PSM站点还有其他PSD 和MCU/DSP应用笔记

APPENDIX

Appendix A—Intel 32-bit十六进制目标文件记录格式

The Intel 32-bit十六进制目标文件记录格式有9-character (4-field)前缀，用来定义记录开始，字节数，登陆地址和记录类型。目标文件还有2-character后缀。.hex后缀表示该格式的样本记录。

Figure 2. Intel 32-bit十六进制目标文件记录格式



以下是四种记录类型：

- **00—Data Record:** 这种记录以冒号开头，然后是第一字节数（十六进制符号），第一数据字节地址和记录类型（等效“00”）。

  接下来是数据字节，数据字节后的检验和（二进制）是记录中前面字节的补充。包括字节数，地址，记录类型和数据字节。

- **01—End Record:** 这种end-of-file记录仍然是以冒号开头，接着是字节数（等效于“00”），地址（等效于“0000”），记录类型（等效于“01”)和检验和，“FF”。
- **02—Extended Segment Address Record:** 绝对目标地址。该记录的地址中必须包含ASCII zeros (hexadecimal 30)。这种记录类型在基地址中占4到19位，它随机出现在目标文件中，并且影响文件中子数据记录的绝对存储器地址。下面的例子讲述了扩展段地址怎样决定字节地址。

问题:

找到下列文件的的一级数据字节地址：

```
:02 0000 04 0010 EA
:02 0000 02 1230 BA
:10 0045 00 55AA FF ..... BC
```

解决:

第一步 找到数据记录的扩展线性地址偏移(0010 in the example)
第二步 找到数据记录的扩展段地址偏移(1230 in the example)
第三步 找到数据记录的地址偏移(0045 in the example)
第四步 按照下面的方法计算数据记录的一级字节的绝对地址
00100000 线性地址偏移,左移（shifted left） 16 bits
+ 12300 段地址偏移, 左移（shifted left）4 bits
+ 0045 数据记录的地址偏移

00112345 32-bit 一级数据字节地址
一级数据字节地址为112345

Note: 使用这种格式通常有特定地址偏移，即使偏移为零。输出翻译时，如果记录大于16，固件会将其强行改为16（十进制）。如果小于16则没有此限制

- **04—扩展线性地址记录：**数据记录目标地址规定为16—31位，将其加到地址偏移中得到绝对目标地址，并且随机出现在目标文件中。在记录地址字段中必须包含ASCII zeros (hexadecimal 30s)。

Appendix B—定义

CPLD 对 Non-CPLD Parts.

在手册中有许多关于CPLD-based PSDs 和 non-CPLD-based PSDs的参考资料。它们的区别也很简单：CPLD-based PSDs有寄存器逻辑（registered logic），而non-CPLD PSDs只有组合逻辑。也就是说CPLD-based PSDs可以执行简单的寄存器功能（registered functions），例如记数，移位寄存器和状态机（state machines）。基本上，(Z)PSD211R, (Z)PSD3XX, PSD9XX, and PSD41XX都是非-CPLD（non-CPLD），其他的PSD有CPLD。

复用总线对非复用总线定义.

微控制器（MCU）和数字信号处理器（DSPs）有两种类型的地址和数据总线：多路复用和非多路复用。区别在于：多路复用类有多路复用总线，地址从MCU/DSP输出，数据用同一物理总线读写，但在不同时间（时分）。对非多路复用，当地址输出时，数据通过分离的总线读写。相比较而言，多路复用管脚较少，而非多路复用总线通常快一些。

何为固件（Firmware）?

固件是从源代码翻译过来的，写(C, C++)成MCU/DSP能够识别的可执行格式。被称之为固件是因为它通常储存在非易失性存储器（EPROM, EEPROM, Flash memory, or battery-backed SRAM)中。因此如果想从PSD 导入，固件中必须有重启向量，中断向量和导入代码。如果你想在运行过程中通过串行端口或其它媒介更新PSD，那么存储器中必须有处理更新的代码。www.st.com/psd站点有PSD 的更多信息。

何为分页?

分页是扩展数字信号处理器（DSP）和微处理器（MCU）物理地址空间的一种方法。MCUs/DSPs

有固定的地址总线的地址位。例如，大多数8位MCUs/DSPs有用于外部器件的16位地址。这就是说，可以访问2^{16}或64K地址空间。这些是目前的标准，但是分页可以帮你解决这些问题。分页允许你创造大的物理地址空间。举例来说，如果你有8位分页寄存器，你就有2^8页或256页。页数就变成物理地址空间的乘数。因此，如果你有16地址位和8页位，你的虚拟存储空间为$2^{16}X2^8$或16M。

我需要分页吗？

要回答这个问题，要看你的MCU/DSP的输出地址位，看MCU/DSP的物理存储空间是否大于或等于MCU/DSP外部所有存储空间。例如，如果你没有使用输出16地址位的MCU/DSP，而且你有两个外部存储器：128 K x 8 Flash和32 K x 8 SRAM，这时候你就需要三分页（在每一分页留有32K自由空间）。因为分页必须是偶数出现，所以需要两页位(2^2 = 4)。注意：有的MCUs/DSPs内部存储器有时用尽某些物理地址空间，在决定是否需要分页时必须考虑这种情况。

何为IAP？

工业中经常用到在系统编程（ISP）。ISP可用于编程逻辑，可编程非易失性存储器。然而，本手册将要讲到在应用编程（IAP）。使用微处理器，ISP和IAP存在非常微妙而且非常重要的区别。对于ISP，给存储器编程时，MCU/DSP不在线。对于IAP，MCU/DSP参与给存储器编程，这对更新固件时必须在线的系统有重要意义。通常，ISP用于生产，而IAP用于现场更新。Flash PSD器件，两者都可以。要注意的是IAP只能编程PSD的存储器部分，而不能给结构和逻辑端口编程。有ISP，整个PSD就可以擦写和编程。

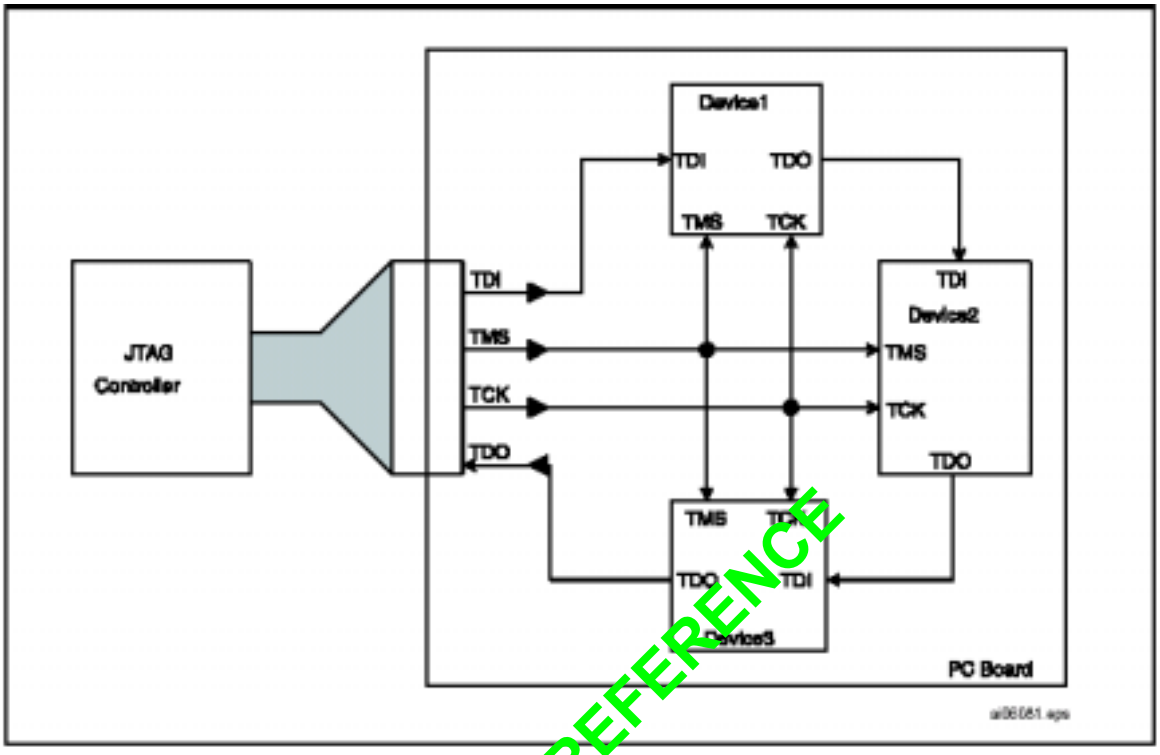
何为CSIOP？

CSIOP (Chip-Select I/O Port)是PSD内部寄存器的集合。你必须在数据手册中给CSIOP分配足够的地址空间。然后在CSIOP空间内访问某个寄存器，将数据手册中的偏移加到你在“Chip Select Equations”指定的基地址上。

Appendix C—JTAG Chain Definition and Rules

JTAG Chain 是在一个链中的两个或多个JTAG兼容（compliant）器件，就像下图所示

Figure 3. JTAG Chain Diagram



链中的每个器件都必须支持基本的JTAG信号：TDI，TDO，TCK，TMS。有些器件是在系统编程（ISP），而不是标准的JTAG端口。这些器件不是IEEE 1149.1兼容（compliant）器件，因此不能用于JTAG链。以下是常规JTAG链的规则：

- 链中只有一个器件可以编程，其它器件必须是“Bypass Mode”
- 不同品牌的芯片要求不同的编程软件
- 在JTAG的任何操作之前必须规定链的顺序，而且软件还必须知道各器件的指令寄存存器长度。给器件编程时，链的顺序非常重要。在编程软件中你必须建立链与板子上的器件的物理顺序一致。对上图，Device1是链中的第一个器件，Device2是第二个，Device3是第三个 。在“JTAG-ISP Information for Flash PSDs”中可以看到PSD中JTAG和JTAG链的更多信息。

Table 1. 文件历史修订

日期	版本	修订简述
	1.0	基于PSDsoft 2000 6.02的原始版本
	2.0	为PSDsoft 2000 7.0修订，附加安装向导
十月一日	2.1	为PSDsoft 2000 7.3修订，在Software Center的PSD站点可以查看软件历史修订

在www.st.com/psd可以查看PSD产品的当前信息

如果你对文件中的问题与什么疑问或建议，发送到apps.psd@st.com （应用支持）

ask.memory@st.com（常规问题）

记得写上姓名，公司，地点，电话和传真