Blackfin在线培训课程

课程单元：VisualDSP++®工具入门
主讲人：Nicole Wright

第一章：简介
第1a节：课程说明
第1b节：CROSSCORE产品系列
第2章：配置ADSP-BF537 EZ-KIT Lite评估板
第2a节：硬件设置
第2b节：EZ-KIT会话设置
第3章：VisualDSP++工具
第3a节：创建项目
第3b节：绘图特性
第4章：利用芯片特性提升性能
第4a节：利用高速缓存
第4b节：利用L1内存
第4c节：利用电压调节器
第5章：创建一个以太网应用
第5a节：利用项目向导
第5b节：运行应用

第一章：简介

第1a节：课程说明

大家好，欢迎参加“Visual DSP++®工具入门”培训课程。我是工具小组的工程师，我的名字叫Nicole Wright，今天将由我为大家讲解这个单元。

通过这个单元，你将了解Visual DSP++工具的概况。在举例演示中，我们将以ADSP-BF537 EZ-KIT Lite®作为目标板。我们要使用Visual DSP++ 4.0工具，介绍一些适用于应用分析和优化的小窍门，以及这些工具的相关信息。

在这个单元，我们将提供一些关于Blackfin ® CROSSCORE ®产品系列的信息；演示如何设置硬件、配置工具，以便使用EZ-KIT Lite评估板。然后，我们将快速演示Visual DSP++工具的使用方法。我们将演示项目创建过程和如何使用绘图特性——一个非常有用的调试特性。然后，我们将利用芯片提供的特性，执行性能优化。

具体而言，我们将使用高速缓存、L1内存和电压调节器。最后，我们将利用这些工具的特性，创建一个以太网应用。为此，我已经设置了一个模板。

第1b节：CROSSCORE产品系列

Blackfin CROSSCORE产品系列由三个组件构成，这个是软件组件，即Visual DSP++开发套件，其中包含了所有的代码生成工具、设备驱动程序和系统服务库、一个名为“VDK”的RTOS（实时多任务操作系统）内核、代码向导、闪存编程器、绘图特性及先进的代码生成和调试工具，最后还有一个自动API。CROSSCORE产品系列的第二个组件，即另一套工具，就是EZ-KIT Lite评估板和子卡。我们提供了适用于BlackFin 533、535、537和561的EZ-KIT Lite评估板，以及用于扩展评估板功能的多种子卡，包括音频视频卡、高端音频卡、USB-LAN EZ-Extender扩展卡和FPGA EZ-Extender扩展卡。最后一个组件是JTAG仿真器。这个仿真器可以支持USB 2.0、USB 1.1和PCI端口以及后台遥测。此外，我要指出的是，我们的所有产品都可以免费享受技术支持，用户无需支付维护费。

第2章：配置ADSP-BF537 EZ-KIT Lite评估板

第2a节：硬件设置

接下来，我将演示如何安装EZ-Kit评估板。

首先，插入电源接头。电源接好后，评估板上的一些指示灯将会闪烁。然后，就要插入USB电缆。在这个演示中，USB电缆已经插入了PC的背板；评估板通过USB电缆直接连接至PC。连接完毕后，电脑屏幕将出现一条提示消息，报告发现了新硬件。你只需要在硬件设置向导界面单击“next（下一步）”按钮，然后，EZ-KIT设备驱动程序就会自动加载到系统中。好了，硬件设置已经完成。下面，我将演示如何配置工具，以便在这个目标板上使用这些工具。

第2b节：EZ-KIT会话设置

现在，我将演示如何从“开始”菜单，调用“Analog Devices”菜单。在“Analog Devices”、“Visual DSP++ 4.0”菜单项下，有四个菜单选项，分别是“Maintain this installation（维护本次安装）”、“Visual DSP++ Configurator（Visual DSP++配置器）”、以及一个文档选项和环境选项。我将演示“维护本次安装”选项的功能。单击这个选项，屏幕将出现一个用户界面，通过其中一个选项，可以链接至Analog Devices公司网站。建议大家在工具安装完毕后，登录Analog Devices公司网站，查看是否有可用的更新程序。更新程序下载完毕后，将自动返回这个安装维护对话框；这时，请选择运行已下载的更新程序。在本例中，我已经完成了这部分操作。下面，我们要启动工具。我将演示如何配置工具，以便用于EZ-KIT目标板。单击“Visual DSP++ environment（Visual DSP++环境）”选项，屏幕出现工具用户界面，要求用户选择一个新的会话。在这个会话中，我们将Blackfin Emulators/ Z-KIT Lite

选作目标板。这里提供了多个EZ-KIT Lites，我们选择的是采用调试代理的537 EZ-KIT Lite。你可以自定义会话名称，今天我们将使用默认名称。现在，工具已启动。这里显示出，我们要使用的目标板已经创建完毕，即这个通过调试代理的EZ-KIT Lite评估板。这里有一个项目窗口，这边是一个反汇编窗口。现在，我们已经完成了硬件设置、项目配置以及配置工具，以使用这个目标板。

第3章：VisualDSP++工具

第3a节：创建项目

接下来，我将快速演示一遍如何使用Visual DSP++工具。首先，我要创建一个项目。不过，我们只编写一个简单的小程序，一个大家早已熟知的“Hello World”程序。你也可以根据需要，调节这些窗口的大小，以便看得更清楚。现在，我在任务栏上单击“File（文件）”，选中“New（新建）”，选择创建一个新项目，屏幕上出现了一个项目向导。在这个界面上，用户可以创建新项目。在这里，你可以在这些目录中选择一个。我打算将我的项目放在“demo”目录下，并将我的项目命名为“hello”。我选择创建一个标准应用，然后单击“Next（下一步）”。很好，提示消息说这个目录不存在，于是，我选择创建这个目录。现在，我要选择处理器类型，也就是说，我创建的项目将在哪个目标处理器上运行。我们选择的目标板采用了537处理器，因此，我们将使用这个目标处理器。接下来，单击“Next（下一步）”。此时此刻，我们不需要添加开始代码，所以忽略这个选项，直接进入“完成”界面。这里列出了刚才做出的所有选择，以便用户检查创建的项目是否符合自己的需要。文件名是“ello.dpj”，项目文件的文件扩展名是“.dpj”；目录是“demo\hello”目录；这是一个标准应用；处理器类型是我们刚才选择的；最后将生成一个可执行文件。好了，单击“Finish（完成）”按钮。你看，这里已经创建了几个文件夹。这些文件夹可以存放你创建的源文件、连接器文件和头文件。今天，我们只创建一个简单的C代码文件。因此，单击“File（文件）”，选中“New（新建）”，选择创建一个新文件。我们只要编写一段“#include”程序，其中将包含我们的标准IO库，然后，编写“printf”函数。这是一个主函数，现在输入主函数代码，也就是“printf”和典型的“Hello World”。然后结束这个函数，保存文件。请注意，“保存的文件”将存入项目目录下，这也是我们想要的。接下来，我们将这个文件命名为“hello.c”，然后保存项目。现在，我们的项目已经创建好了，但是新文件还没添加到项目中。因此，我们要继续将这个文件添加到项目中，单击“Project（项目）”、“Add to Project（添加至项目）”，选择添加文件。同时，请注意，通过这个项目菜单，我们也可以创建项目，选择项目选项、设置项目、更新相关性、导出或新建文件、构建一批文件和设置配置等。此外，如果你使用了源代码控制，那么，你还可以在这里设置源代码控制。我们要添加的文件是“hello.c”。添加完毕后，你会发现，在源文件的左侧将出现一个“+”符号，表明这个文件夹里有一个文件。现在，我们展开这个文件夹，hello.c文件就在这里。接下来，我要构建这个文件，单击“Project（项目）”、“Build Project（构建项目）”。好了，屏幕下方的构建窗口中显示了状态信息，表明构建的进展情况。构建顺利完成后，就要将文件加载到目标板上。对于这个项

目，工具中已经设置了选项偏好，并默认设置为在构建完毕后执行加载。另外，你可能会注意到，我们当前是在蓝色显示的“printf”语句上，这就表示程序计数器目前是在这个位置。你还会看见一个箭头标记，也表明了这一点。这儿还有一个红色的按钮，表明这里有一个自动设置的断点。这也是你可以自定义的用户偏好。好了，现在项目构建已经完成，下面就来运行这个项目，看看会出现什么情况。我要单击“Run（运行）”图标，注意观察，你会发现在控制台窗口中显示了“Hello World”。到此为止，我们创建了一个项目，并运行了这个项目，大家已经初步了解了创建一个应用的流程。

第3b节：绘图特性

现在，我要关闭这个项目。接下来，我将演示绘图特性。关闭所有的源代码窗口，从头开始。

打开项目，这是保存已安装工具的默认目录：program files\Analog Devices，所有东西都在这儿。我们在绘图中要使用的项目是一个可以执行排序的简单的C项目。要使用绘图特性，首先必须构建项目。单击“Project（项目）”、“Build Project（构建项目）”。现在，项目已经构建并完成加载。接下来，我们要做的就是配置绘图菜单。在“View（视图）”菜单中，选中“debug windows（调试窗口）”，这里有反汇编、跟踪、局部和表达式等选项，这些都是为典型的调试窗口提供的选项，当然还有绘图选项。现在，我们创建一个新的绘图窗口，我们选择线图，你可以看见，这个下拉菜单中提供了多种不同的绘图类型。我们将这个绘图命名为“bubble（冒泡）”，因为其数据来自冒泡排序算法。我们要调用数据集“data set 1”，查看Blackfin内存的相关数据。我们要使用的地址是程序中阵列的地址，也就是程序中的这个“out_b”，刚刚输入的就是这个地址。将计数设置为128，因此，跨距将是阵列中的128个元素，这也是内存空间的数值；然后将“stride”设置为1。我们希望数据类型是整数，因此，在这里选择整数值。在这里，一个关键的步骤是必须单击这个“Add（添加）”按钮，添加数据集，然后再退出对话框。现在，数据集已经添加完毕，接着单击“OK（完成）”按钮。现在，你可以看见绘图已经完成。此时，我们的阵列中什么都没有，它已被初始化为零。在开始执行冒泡排序之前，我要设置一个断点，这样，在完成冒泡排序的线图之前，会先出现一个随机阵列函数的线图。在随机阵列中，数据是随机排列的，然后，对其进行冒泡排序，显示出有序的数据。创建断点的方法有多种，可以单击当前所在的代码行，然后在相应的位置双击装订线，然后，断点就生成了。或者，你可以利用图标，生成断点。现在，我们要运行这个断点，在“Debug（调试）”菜单上，选择“Run（运行）”。现在你看，数据已经改变，全都是随机数据，我们的变量全在这里。现在我们还没进行排序，只是刚刚开始运行程序。接下来，我要继续，结束运行，最后，将显示有序的变量。再一次，在“Debug（调试）”菜单上，选择“Run（运行）”。现在，你可以看见这个冒泡排序线图，数据已经被排序。通过这种便捷的方法，用户可以查看内存的相关数据。这样，我就不必实际打印，或者在内存窗口中，手动查看每一项参

数。这是一个非常直观的指示器，可用于确保数据符合自己的要求。现在，我们已快速演示了如何使用Visual DSP++工具和绘图特性。

第4章：利用芯片特性提升性能

第4a节：利用高速缓存

接下来，我将演示如何利用芯片提供的特性，提升应用性能。我要关闭这个应用，关闭所有的源代码窗口，再关闭冒泡排序图。你也可以按照自己的需要调节窗口。我们要打开第一个项目，依次单击“Open（打开）、“Project（项目）”。这个项目也是我专为这个课程而提前创建好的，保存在工具包中，待会我将向大家演示。这只是一个简单的排序项目。接下来，我要演示如何利用统计监测器，查看项目的运行情况。现在，单击“Tools（工具）”、“Statistical Profiling（统计监测器）”，然后选中“Profile（监测）”，好了。我使用这个程序是因为它未经优化。所有代码都在外接存储器上运行，所以，我们才有时间讨论当前的运行情况。现在，构建并加载应用，构建窗口中显示了一些信息，正在加载，加载完毕。这里，我想顺便告诉大家，在控制台窗口中，单击鼠标右键可以清除控制台窗口中的内容。在控制台窗口中已经有许多信息，所以如果在查看运行结果之前清除这些信息，会更加一目了然。接下来，运行这个项目。统计监测器将以每秒数百次的速率轮询芯片，以监视应用的运行状况。因此，在直方图中，你可以看见，大多数时间都用于执行冒泡排序运算，因为这种排序的速度比快速排序算法慢得多，这也是意料之中的。此外，你会发现运行速度很慢，这是因为尚未进行优化。在屏幕底部的这个位置，你可以查看当前正的运行状态。同样地，所有代码都在外接存储器上运行，这也会降低速度，我们是故意设置成这样的。另外，当统计监测器轮询芯片时，它不会侵入程序，它是在后台运行的，无需添加特殊代码。

现在，应用运行完毕。你看，运行这个未经优化的应用的时间是38秒，消耗了超过900亿个周期。从直方图中你可以看出，执行冒泡排序运算耗用了大部分周期，近70％。执行快速排序运算也占用了一部分时间。

接下来，继续执行演示。我要关闭一些窗口，以便大家可以更清楚地观看。下面，我们要启用高速缓存，就是片上的保留内存。Blackfin处理器允许用户将指令保存到这个保留内存中。我们将通过“Project Options（项目选项）”进行设置。你会发现，在这个滚动菜单中，有许多选项，可以设置编译器、汇编器、连接器等等。现在，我们要设置开始代码，更改高速缓存和内存保护设置。设置指令高速缓存非常容易，只要单击这里，选择“instruction cache（指令高速缓存）”选项，然后单击“OK（完成）”，就可以了。这个设置会使我们的项目在保留内存中运行，速度将远远快于刚才那种在外接存储器中运行时的情况。在修改设置时，必须牢记，一定要重新构建项目。现在，我要重新构建项目，使这些操作生效。这一次，我们并未进行任何编程，完全是在GUI上完成的更改。直方图现已清空，因为没有执行任何运行。好了，再来看看刚才的数据，第一次运行用了38秒。现在，运行新的项目，这一次性能应该会显著提升。在直方图中，你会看见，相比于快速排序，冒泡排序

仍然消耗了多得多的时间。在下面这个输出窗口中，你可以看见，运行这个程序只用了3秒，并且仅消耗了70亿个周期，比第一次快多了。

第4b节：利用L1内存

除了允许用户使用高速缓存，芯片还允许用户将部分代码保存到片上或L1内存中。所以，接下来我们要关闭指令高速缓存，然后将冒泡排序函数映射到L1内存中。现在，我开始演示。要关闭指令高速缓存，我们就要返回“Project Options（项目选项）”，将高速缓存和内存保护设置，改回当初的“RAM with no memory protection（RAM未设内存保护）”选项，然后单击“OK（完成）”。要将排序数据映射到L1内存中，我们需要对C代码进行一些微小的改动。由于我打算将冒泡排序函数映射至L1内存，因此，我要在冒泡排序说明之前，添加一个语句。这样修改，然后将原来函数中的“segment”，修改为“section”，也就是改成section语句。现在，这里依次是“section”、“L1 code”、“bubble sort”。这行代码的意思是将冒泡排序函数映射至L1内存。我把这个窗口放大一点，以便大家看得清楚。我们必须再次构建这个项目，因为通过关闭指令高速缓存，我们已经更改了项目，而且我们还更改了源代码。因此，我要重新构建项目，完成项目连接和加载。屏幕上自动出现了这个反汇编窗口，其中显示了当前的运行位置。我要关闭这个窗口，因为我知道正在运行什么。现在，我要运行我们的项目，你会发现一些有趣的现象。由于我们将冒泡排序函数映射至L1内存，因此，无需往来于片上内存和外接存储器之间，从而缩短了运行时间。而快速排序算法并未在这个快速内存中运行，因此花了更长时间。你还会发现，运行这个项目的周期反而增多了，耗用了300多亿个周期，实际运行应用的时间也增加了。你可以根据应用的情况，随需组合利用指令高速缓存和L1内存。未经优化时，运行时间是38秒；利用指令高速缓存，运行时间只要3秒；而利用L1内存，运行时间则增至13秒。

第4c节：利用电压调节器

芯片优化的另一个途径就是通过IDDE或编程，设置芯片的工作电压和频率。因此，下面我将利用一个应用，向大家演示如何实现这一点。我已经准备好了一个项目，现在只要打开就可以了，这也是一个排序项目。将这个主窗口设置为浮动。现在关闭这个反汇编窗口，腾出更多屏幕空间，然后打开这个排序项目窗口。代码构建过程需要一些时间，因此在向大家解释代码之前，我们要构建这个应用。单击“Rebuild All（全部重建）”图标。对于Blackfin 537处理器，用户可以利用系统服务库，设置电压和频率。电压设置完毕后，系统服务库会将频率自动设置为芯片在该电压下的最高频率。在这个演示中，我们将分别采用4级电压设置，也就是我们目前显示的这个阵列。当电压设置为0.85时，系统服务库自动将芯片频率设置为该电压下的最高频率。运行这个项目需要一点时间，我们将以三种不同的速度运行这个项目。下面我将解释调用代码行以及作用。这里将显示电压设置、运行时间和运行所用的周期数量。我们设置了4级电压，也就是说要分别采用这4级电压，运行这个项目。向下滚动可以看见，我们通过调用系统服务库，完成了这个设置。这个调用代码是“adi_ power_init”。如需了解关于这个代码的更多信息，请使用帮助系统，

参阅关于如何使用调用代码的特定信息。在这个演示中，我已经完成了这个调用设置，一会儿这里就会显示运行结果。当电压设置为0.85时，运行这个简单的排序项目使用了30秒。请注意，在这个演示中，周期数量很重要。因为当我们更改芯片的电压和频率时，运行周期数量将保持不变，发生变化的是应用的运行速度。好了，你看，电压只提高了0.10，而运行时间却缩短了一半，并且周期数保持不变。屏幕底部的这个运行指示器显示了当前的运行状态。接下来，运行另一个应用，采用另一种设置；这会出现新的情况。为“ez_kit_init”设置命令对；“ez_kit_init”也是一个系统服务库特性，很有用。好了，应用运行完毕。提醒大家，应用运行完毕时，将位于反汇编窗口中的“lib_prog_term”位置；状态屏幕底部的停止消息也表明应用已运行完毕。现在，快速浏览一下运行结果。这些数据证明了我们刚才的讨论，随着电压的增高，频率均设置为该电压下的最高频率，运行程序所用的时间随之相应地减少，而运行周期则保持不变。

在这个部分的演示中，我们解释了对程序进行优化的三种方法：可以利用指令高速缓存、L1内存、以及设置芯片的电压和频率。取决于用户对应用的运行要求，用户既可以选择其中一种快速优化方法，也可以组合利用这些方法，完全由用户自己决定。好了，现在大家已经了解了要快速实现应用优化，需要进行哪些操作。

第5章：创建一个以太网应用

第5a节：利用项目向导

接下来，我将演示如何快速创建一个以太网应用。我们要使用已经创建好的项目向导模板。现在，关闭这个应用；再通过项目向导，创建一个新应用。好了，清空输出窗口。然后，利用项目向导，创建我们的以太网应用。依次单击“File（文件）”、“New（新建）”、“Project（项目）”。这一次，我们要进行不同的设置。我将这个项目命名为“Ethernet（以太网）”，并将其放到我的“demo\Ethernet”目录下，这是自动生成的。这一次，我们不要创建一个标准应用，而是创建一个TCPIP应用。我们要使用轻型互联网协议库，即LWIP，和VDK——用于驱动TCPIP的实时操作系统。单击“next（下一步）”，屏幕出现提示消息，询问我是否要创建这个目录，我选择了“Yes（是）”。然后，选择处理器类型，当然还是Blackfin 537。这一次，也不用设置开始代码。接下来，进入LWIP向导页面，这个轻型互联网协议向导将要求用户选择标准目标板，以便为库创建适当的设备驱动程序和库。好了，设置完毕，进入这个项目概要页面。这里列出了刚才的设置，这里显示出我们更改了项目类型；请注意，虽然项目类型已经改变，但是文件扩展名依然是“.dpj”。此外，你会发现，在创建以太网应用时，模板将自动设置一些源代码和库，以及连接器使用说明文件和面向RTOS的内核文件。此外，模板还将通过Ping操作，从互联网上获取一个以太网地址，以供目标板使用。不过，在此之前，我们要设置好目标板的以太网连接。将这条电缆，接入目标板；这是我们的以太网应用，这是以太网电缆。在这个演示中，这条电缆将连接至PC背面连接的一个路由器，因此，这个演示设置其实就是PC、路由器和目标板。

第5b节：运行应用

现在，构建这个应用，全部更新。在下面的构建窗口中，显示了构建的进展情况。现在正在加载至芯片。接下来，运行代码。这些是关于以太网应用运行状态的消息。我们为此编写了一些printf语句，所以，这里显示说我们在等待建立链接，链接建立完毕，我们得到一个IP地址。这里，我要顺便解释一下我们的线程；你可以看见，这里是printf语句，然后是堆栈和一些“Cliff Notes”。在这里，在IP地址的printf语句之后，是一些关于添加应用代码的位置的说明。这个模板使用起来非常便捷，用户可以利用printf语句轻松查看运行状态，添加的代码，以及执行一些调试，以免最终出现错误。

好了，现在我们有了IP地址。不过，需要进行测试，以确保我们的以太网连接正在正常运行。现在，我要打开一个DOS窗口。然后，利用DOS系统的ping命令，通过以太网向目标板发送数据，并检查命令运行情况。我们将使用控制台窗口中显示的这个IP地址。现在，键入“ping”；这是一个标准的DOS应用，因此，用户应当可以在自己的计算机上执行这个操作。好了，我们进入了我们这个小小网络，你看，这里显示了数据往返信息。也就是说，我们的以太网应用可以畅通运行，只是没有执行打印，我们已经成功完成了任务。

现在，以太网应用演示也完成了。我们来小结一下，在本次课程中，我们首先介绍了我们的CROSSCORE产品系列；然后，演示了如何设置目标板硬件、配置工具以用于目标板；接着，我们快速演示了应用设置流程以及绘图特性；然后，介绍了如何利用芯片提供的特性，优化应用；最后，我们演示了模板提供的以太网功能。

好了，本次课程的内容已讲解完毕。下面，我将介绍一些提供更多相关信息的资源。今天我们使用的所有排序项目都包含在《Getting Started with the ADSP-BF537 EZ_KIT Lite Manual》（《ADSP-BF537 EZ_KIT Lite入门手册》）中。你可以在在线帮助中下载这些工具的用户手册，其中包括关于特定工具的信息，以及如何使用这些工具。你也可以登录模拟器件公司（ADI）网站：www.analog.com，查找关于工具、工具更新程序、应用设计等方面的信息。另外，我们还提供了两个技术支持电子邮件地址，如有关于工具使用的疑问，请发送电子邮件至：processor.tools.support@analog.com。如有关于所用处理器的疑问，请发送电子邮件至：processor.support@analog.com。你也可以单击演示界面下面的“ask a question（我有疑问）”按钮，提出疑问。

谢谢观看！