

# 为软件工程师揭开 FPGA的神秘面纱

作者: Glenn Steiner  
技术营销高级经理  
Xilinx, Inc  
[glenn.steiner@xilinx.com](mailto:glenn.steiner@xilinx.com)

Dan Isaacs  
技术营销总监  
Xilinx, Inc.  
[dan.isaacs@xilinx.com](mailto:dan.isaacs@xilinx.com)



# 本文件介绍了有关如何为 FPGA 嵌入式处理器开发软件的一些实用技巧。

随着产品设计复杂性的增加，需要使用集成组件（如特定于应用的标准产品 (ASSP)）来满足设计上的要求。几年前，工程师会针对处理器、存储器和外设选择单独的组件，然后通过分立逻辑器件将这些元素拼合到一起。后来，他们会搜索 ASSP 处理系统目录，试图找出最相称的组合，以满足系统要求。当他们需要其他逻辑或外设时，通常会将某个 FPGA 与某个 ASSP 配对来完成解决方案。的确，调查表明 FPGA 目前在全部嵌入式系统中的使用比例占 50% 到 70%。

在过去的几年中，FPGA 的尺寸有了增加，在单个器件中提供了足够的空间来容纳完整的处理器和逻辑系统。现在，软件工程师需要为 FPGA 内部的处理器开发和调试代码。在有些情况下，他们害怕这么做。不过，通过掌握 FPGA 基础知识并了解如何创建和调试用于 FPGA 嵌入式处理器的代码可以让他们坦然面对。

## FPGA 是什么？

现场可编程门阵列 (field-programmable gate array, FPGA) 是一种集成电路，其中包含的逻辑可以在制造后进行配置和连接（即“现场模式”）。在这方面，过去工程师会从产品目录中购买各种逻辑器件，然后通过印刷电路板上的连接将它们组装成一个逻辑设计，而现在的硬件设计人员可以在单个器件中实施完整的设计。形式最简单的 FPGA 中包括：

- 由与、或、非及其它许多逻辑函数组成的可配置逻辑块
- 使逻辑块可以连接到一起的可配置互连功能
- I/O 接口

利用这些元素，用户可以创建任意逻辑设计。

硬件工程师通常用 HDL（一般是 Verilog 或

VHDL) 编写代码, 然后将设计汇编到一个对象文件中, 该文件可加载到器件中供执行。在表面上, HDL 程序可能看起来很像 C 语言之类的高级语言。例如, 用下面以 Verilog 编写的 8 位计数器的实现为例 (由 [www.asic-world.com](http://www.asic-world.com) 提供)。其中可以看到许多目前的高级语言中的构造:

## FPGA 的技术优势

由于不使用 ASIC 以及没有与其关联的高额模板费用, FPGA 是实现数据处理元素最灵活且高性价比的方法。FPGA 由于采用了灵活的架构, 能够允许硬件设计人员实现同时包含并行和串行元素的处理系统。这样一来, 设计人员可以针对性能和延迟对系统进行优化。通常, 与使用通用处理器可获得的性能相比, 这些数据处理系统获得的性能级别更高, 但成本却更低。

```
//-----
// Design Name : up_counter
// File Name   : up_counter.v
// Function    : Up counter
// Coder      : Deepak
//-----
module up_counter (
  out      , // Output of the counter
  enable   , // enable for counter
  clk      , // clock Input
  reset    // reset Input
);
//-----Output Ports-----
  output [7:0] out;
//-----Input Ports-----
  input enable, clk, reset;
//-----Internal Variables-----
  reg [7:0] out;
//-----Code Starts Here-----
always @(posedge clk)
if (reset) begin
  out <= 8'b0 ;
end
else if (enable) begin
  out <= out + 1;
end
endmodule
```

您可以将外部微处理器耦合到 FPGA 中的某个数据处理系统上, 然后用它进行控制。但是, 在 FPGA 中嵌入一个处理器可以提供几项优势。一个内部处理器可以极大地减少处理器与数据处理系统之间的延迟, 能够消除大量处理器周期。处理器与数据处理系统之间的通信信道可以是 32 位或更高, 拥有更多的寻址和控制线缆。对于一个外部处理器, 这些增加的线缆可能需要更大的封装来容纳处理器和 FPGA, 从

而抬高了系统成本。或者, 也可以使用 PCI Express® (PCIe®) 来减少引脚的数量。而不幸的是, 由于这是相对较新的接口, 并非所有处理器和 FPGA 都支持 PCIe。虽然 PCIe 具有与生俱来的高性能, 但它是串行接口, 因此会增加处理器与数据处理系统之间的延迟。

在 FPGA 中同时实现处理器与数据处理元素可以减少元件数量, 缩小电路板空间, 有时还能降低功耗。这可能产生一个成本显著降低的解决方案。FPGA 中可提供固化的处理器实现 (如 ARM® Cortex™-A9 处理器) 或软核实现 (如 Xilinx® MicroBlaze™ 处理器)。基于 FPGA 的处理器还可根据应用的要求进行配置。基于 FPGA 的系统支持系统级的调优: 它可以灵活地选择将计算功能在处理器或 FPGA 逻辑中实现。

## 实现技术

FPGA 嵌入式处理系统的实现方法有很多种, 但一般可将其分成三类: 从头组装系统, 使用向导将其组合到一起, 或者通过修改现有设计实现。

FPGA 工具可支持您从头组装一个处理系统, 方法是: 从列表选择一个必需的 IP, 然后通过总线和线缆连接该 IP。这样的组装虽然有效, 但可能会耗费时间。

为了加快速度, FPGA 工具还允许通过向导快速组装微处理器系统。通过使用下拉列表或复选框, 可以轻松指定目标部件以及所需的处理器和外设。图 1 展示了向导启动的介绍窗口以及向导生成的最终系统。类似地, 也可以使用 MATLAB® 软件之类的工具, 快速地将数据处理系统与用于控制的处理器总线接口组装起来。然后, 只需匹配总线接口, 就可将处理器与数据处理系统连接起来。

嵌入式处理系统的第三种实现方法是修改现有的参考设计, 或将其添加到现有的硬处理器系统上。FPGA 参考设计和硬处理器系统会继续演变, 其中许多会变得更加侧重于市场。在许多情况下, 设计均非常全面, 无需硬件设计人员增加任何额外的组件。软件设计人员通常会找到完整的驱动程序, 以及为这些参考设计预置的操作系统。

前面提及的前两种方法都是创建处理器系统的有效方法。但是, 第三种方法从现有的经过验证的参考设计入手, 因此可以极大地减少硬件和软件工程师的开发时间。

## 澄清误解

在工程设计界已经出现了一些有关在 FPGA 中开发处理器代码很困难的误解。我们希望能澄清这些误解。

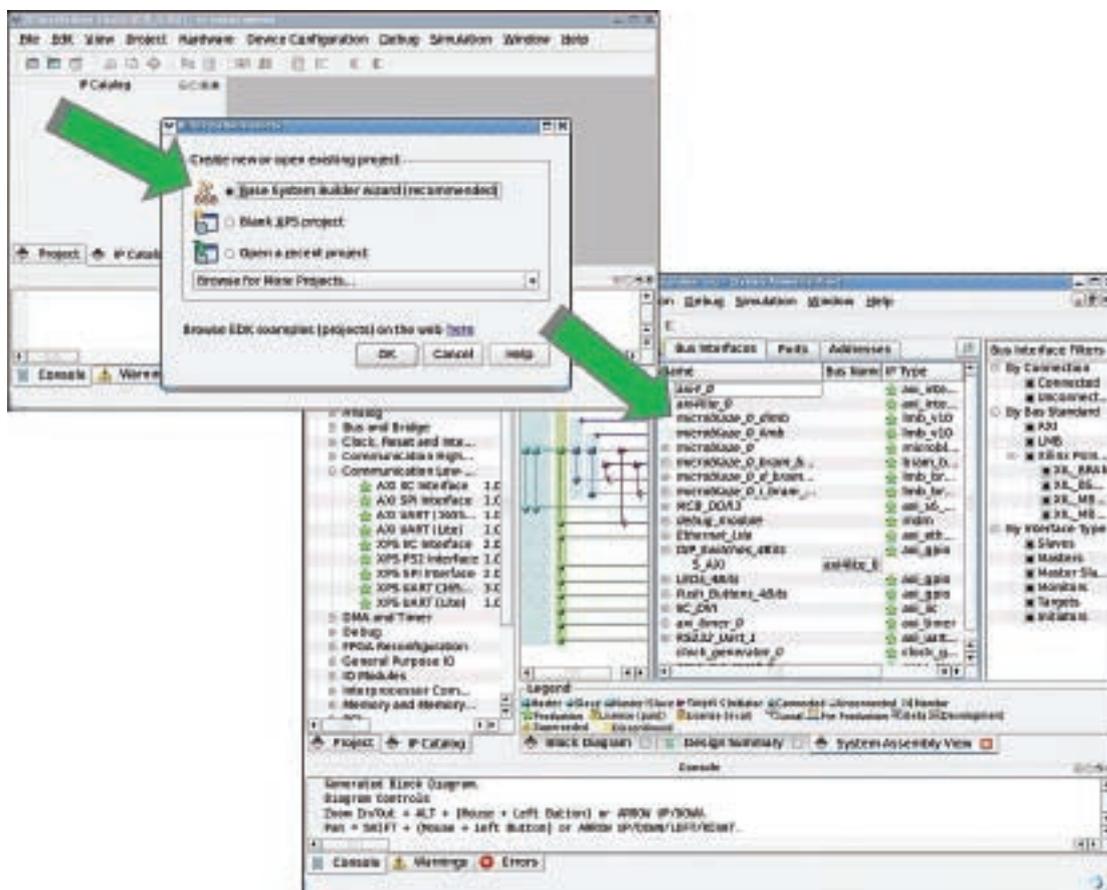


图 1 向导启动屏幕和完成的系统

**误解：**在 FPGA 中为处理器编写代码很困难。

**事实：**大多数 FPGA 嵌入式处理开发都是在现代化软件开发环境中用 C 或 C++ 完成的。

许多 FPGA 供应商现在都支持使用 Eclipse 进行软件开发。Eclipse 是一种支持插件的灵活软件开发环境，提供文本编辑器、编译器、链接程序、调试器、跟踪模块和代码管理等功能。

作为一个开放的环境，Eclipse 拥有一个庞大的开发人员社区，不断地增加新功能。例如，如果程序员不喜欢提供的编辑器，他们可以安装一个更符合其需求的编辑器。图 2 展示了 Eclipse 代码编辑器和“hello world”程序。

**误解：**FPGA 没有像 ASSP 那样的处理器系统。

**事实：**现在已经有预置的 FPGA 软嵌入式处理器设计，也有具备像 ASSP 那样的外设集的硬处理器设计。

包含软处理器和硬处理器的 FPGA 增加了一个额外的功能。FPGA 嵌入式软处理器参考设计整合了 32 位 RISC 处理

器、内存接口以及符合行业标准的外设。此类处理器的灵活性允许用户用逻辑换取额外的性能功能，如增加支持最新操作系统的 MMU。众多的 FPGA 选择使用户可以选择某个处理器配置、外设、数据处理逻辑和逻辑性能级别来满足其系统要求。预置的类似于 ASSP 的参考设计使得软件设计人员可以立刻开始编写代码，不一定需要硬件工程师先实现一个处理器系统。在许多情况下，预置的设计会满足嵌入式处理器系统的要求，从而无需硬件工程师来进行进一步的处理器系统设计。在少数情况下，硬件工程师拥有一个出色的平台，可以在上面增加外设以及连接定制的硬件加速器。

**误解：**使用 FPGA 中的处理器调试代码很困难。

**事实：**FPGA 嵌入式处理器的软件调试跟非嵌入式处理器的调试一样方便。调试器支持下载代码、运行程序、在源代码和对象代码级别单步执行、设置断点、检查存储器和寄存器。另外，还有其他工具可用于对代码进行特性分析和跟踪。

**误解：**不支持我喜爱的操作系统。

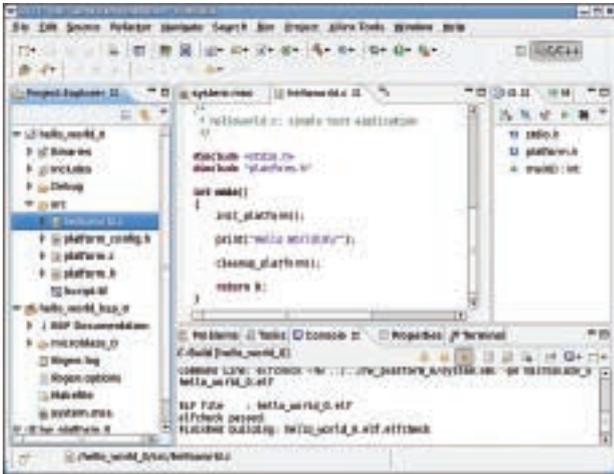


图2 包含代码的 Eclipse 集成开发环境

**事实：**大多数常见的嵌入式处理器都支持最常见的操作系统，并且这个名单还在不断增加中。Xilinx MicroBlaze 支持 Linux、ThreadX、MicroC/OS-II 和 eCos 等操作系统。

**误解：**没有驱动程序。

**事实：**FPGA 嵌入式处理器有一个很大的外设库，其中包含了驱动程序。表 1 是一个有代表性的名单，其中列出了 FPGA 软处理器可用的一些软外设。所有这些设备都存在相应的驱动程序。

**误解：**硬件工程师必须先建立它之后我才能编写代码。

**事实：**可以使用经过测试的预置处理器系统设计，因此可以立即进行软件开发。

这些预置的类似 ASSP 的处理器系统中包含处理器、内存控制器和内存、闪存控制器以及 UART、GPIO 和以太网接口等外设。这些系统在交付时都带有参考软件设计实例，其中包括对 Linux 引导的展示。

**误解：**无法使用 FPGA 嵌入式处理器进行特性分析或跟踪。

**事实：**特性分析和跟踪工具也是可以使用的。开发人员可以通过特性分析了解处理器在各个函数上所用的时间，以及对任何指定函数的调用次数。

**误解：**FPGA 软件开发工具价格太贵了。

**事实：**ASSP 和 FPGA 供应商为其嵌入式软件开发功能提供的定价在 200 美元到 500 美元之间。此外，许多供应商还提供试用版本以及免费或功能受限的版本，另外还有打折的评估套件。

## 创建和调试代码

FPGA 嵌入式处理器系统的软件开发流程遵循一些常规步骤：

1. 创建软件开发工作区并导入硬件平台。
2. 创建软件项目和板支持包。
3. 创建软件。
4. 运行并调试软件项目。
5. 可选步骤：对软件项目进行特性分析。

步骤 3、4 和 5 是大多数开发人员都熟悉的。有些开发人员可能不熟悉步骤 1 和 2，但这些步骤都很明确。我们将采用 Eclipse 开发环境作为例子，更具体地了解每个步骤。

### 创建工作区并导入硬件平台

启动 Eclipse 以后，系统会提示用户要使用的工作区。工作区就是一个目录路径，用于存储项目文件。接着，用户指定硬件平台（设计）。硬件开发工具会自动生成此文件。该文件对处理器系统进行描述，其中包括存储器接口和外设以及存储器映射。该文件是硬件开发工具的输出内容，硬件工程师通常会向软件开发人员提供此文件。指定以后，就会导入硬件平台，此步骤也宣告完成。

### 创建软件项目和板级支持包

板级支持包 (board support package, BSP) 包含供软件应用使用的库和驱动程序。软件项目是软件应用源和设置。

在针对 Xilinx 项目定制的 Eclipse 版本中，可以选择“File（文件）”→“New（新建）”→“Xilinx C Project（Xilinx C 项目）”。对于 Xilinx C 项目，Eclipse 会自动创建 Makefiles，将源文件编译成对象文件，并将这些对象文件链接到一个可执行文件中。在此步骤中，用户可以定义“Project Name（项目名称）”，通过提供在步骤 1 中创建的硬件平台名称将其与硬件平台关联，然后指定项目名称。

接下来，系统会确认 BSP 的生成情况，并根据定义的硬件平台和操作系统自动加载适用的驱动程序。随后会编译这些驱动程序，生成 BSP。

### 创建软件

此时可以导入软件实例，或者从头创建代码。保存代码时，Eclipse 会自动编译并链接代码，并报告任何出现的编译器或链接程序错误。

### 运行并调试软件项目

使用 FPGA 时，在执行代码之前必须完成一个步骤：

通信	外设	系统
PCI	通用 I/O	调试模块
PCI Express	UART 16550	中断控制器
USB	Delta/Sigma ADC、DAC	计时器/PWM
I2C	外部外设控制器	时基/看门狗定时器
SPI	ADC	邮箱
Gigabit Ethernet MAC	<b>存储器</b>	互斥对象
Ethernet MAC Lite	多端口内存控制器 (SDRAM、DDR、DDR2、DDR3)	
CAN	外部内存控制器 (Flash、SRAM)	时钟发生器
FlexRay	CF 卡	系统复位块
MOST	片上存储器	中央 DMA

表 1 软处理器外设清单范例

为 FPGA 编程。在 Eclipse 中，只需选择“Tools (工具)” → “Program FPGA (FPGA 编程)”。此步骤将获取硬件工程师已创建的硬件设计，并将其下载到 FPGA 上。该作业完成后，就可以选择要构建的软件类型了。“Debug (调试)”会禁用代码优化并插入调试符号，而“Release (发布)”则会启用代码优化。若要进行特性分析，可使用 -pg 编译选项。

最后，通过选择“Run (运行)”并定义运行配置的类型和编译器选项，可以运行代码。如果选择了“Release (发布)”，则处理器将立即开始执行代码。否则，处理器将执行一些引导指令，并将在源代码的第一行处停止，Eclipse 中将出现调试视图。

调试视图会显示源代码或对象代码、寄存器、存储器和变量。可以在源代码或对象代码级别单步执行代码，并可以设置代码执行的断点。

### 对软件项目进行特性分析

如果您需要，可以在此时对代码进行特性分析，查看函数调用的次数，并可查看用在任何指定函数上的时间百分比。图 3 展示了包含特性分析器结果的 Eclipse 特性分析视图实例。

### FPGA 的优势

在成本、能耗、尺寸和整体系统效率方面，FPGA 嵌入式处理器正在成为主要的设计选择。好消息是，软件工程师不需要将 FPGA 嵌入式处理器看得很神秘，或

者不再比外部处理器更难以编程。

FPGA 供应商提供的是符合行业标准的开发环境（如 Eclipse），成本上有竞争力，并针对 FPGA 嵌入式处理进行了定制。在这些环境中，用户可以创建、编译、链接和下载代码，并可像其以前对待外部处理器一样，以相同的方式调试其设计。借助预置的处理器参考设计，软件工程师可以立即开始编写代码并进行测试，不一定需要硬件工程师提供最终设计。最后，FPGA 嵌入式处理器拥有丰富的 IP 库、驱动程序和操作系统支持。

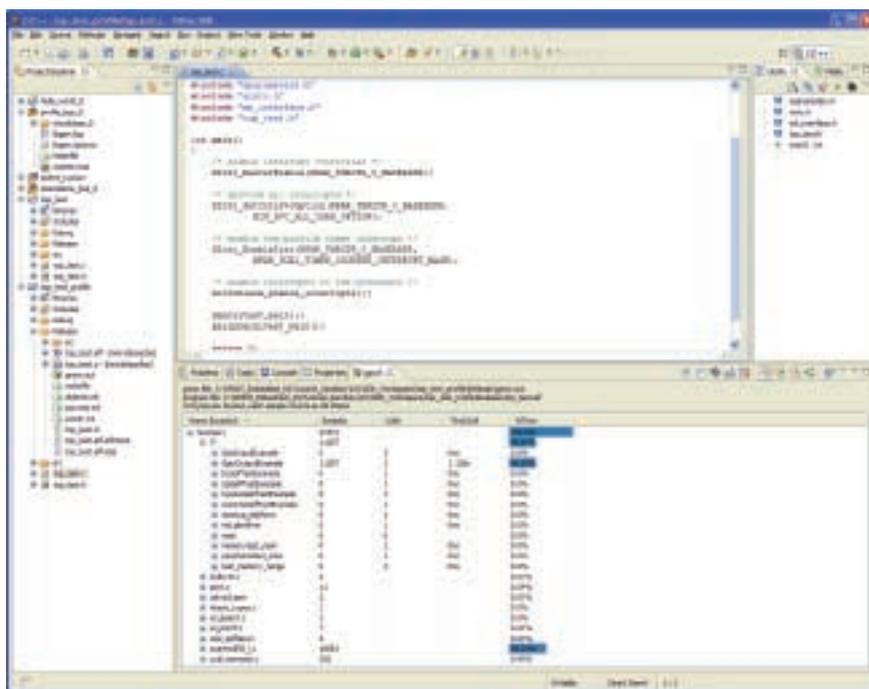


图 3 Eclipse 特性分析视图