

## 概述

XJFlash 是一款通过JTAG在线烧写闪存器件的高级、革新方法。使用XJFlash，您可以达到与传统边界扫描技术可能性相比将近50倍更快的闪存编写速度。

## 定制的方案 -- 自动生成

XJFlash允许您自动生成面向于您板卡上连接到FPGA的闪存器件的定制编程方案。

FPGA的功能被利用来提供尽可能最快的编写速度。对于每个FPGA/闪存组合，XJFlash自动生成一个

定制的设计，以助您达到最佳的编程时间，而不需要您做任何FPGA开发。\*

无论您是用连接到Altera、Xilinx、Microsemi或者Lattice公司FPGA的SPI、QSPI或者平行式NOR闪存--XJFlash都会给您提供一个优化您板卡的编程方案。

\*在设置XJFlash的过程中，需要一个相关FPGA厂商工具的授权证书。对于大部分器件，免费的版本就可以。

## XJFlash 计时实例

最低理论边界扫描时间：35分钟  
XJFlash运行总时间：10.5秒到32.6秒



每当一个闪存被编写的时候，XJFlash都会自动单步调试四个阶段。

■ **初始化** -- 用目标板卡所需的XJFlash image来配置连接到FPGA的闪存。实例时间：2.1秒

■ **擦除** -- 闪存可以用两种算法之一来擦除。基本擦除只会简单地界定范围内所有的模块擦除（这个可能是整个闪存，或者只是需要用来编写image的空间）。更智能的擦除基于闪存读取比擦除更快的事实；正因如此，它读取每一个地址并且只当有数据被发现的时候才开始擦除。如果知道闪存存在编写之前总是空白的，这一步可以被跳过。实例时间 -- 智能擦除被启动时：对于一个已经擦除的器件需0.9秒，到对于一个完全编好的器件需23秒（受限于器件的擦除时间）

■ **写入** -- 来自于目标image的数据通过JTAG端口被流式传输进FPGA。FPGA接着把这些数据编写进相连的闪存。多个文件可以被指明并且可以在定义的偏移处被编写。如果只需要验证，这步可以跳过。实例时间：6.2秒（受限于器件的编写时间）

■ **验证** -- 验证对着指定的文件检查闪存内的每一个字节，以确保没有数据字节错误。如果只需要编写或者擦除，这步可以跳过。实例时间：对于10MHz的TCK的1.8秒，减少至对于20MHz的TCK的1.3秒。

这些实例时间是为一个Spartan 6 XC6SLX9把2MB的伪随机数据文件编写进FPGA的SPI配置PROM所提供的。

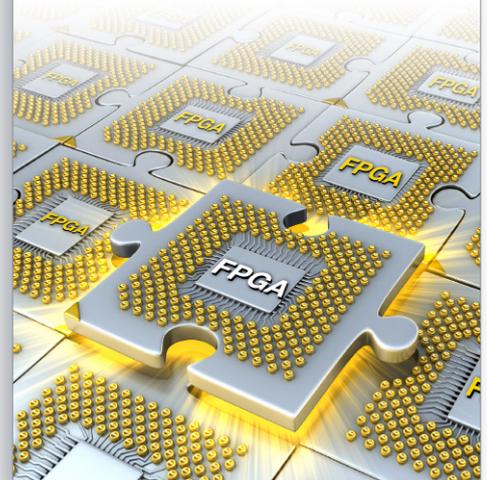
## 主要优点

- 节省闪存编写时间
- 支持SPI、QSPI、平行式NOR闪存器件
- 如有需求，可提供NAND闪存支持
- 缩短开发周期
- 无须其它设备
- 可用来快速固件升级
- 无须FPGA开发

## 支持的FPGA

- **Intel (Altera)**  
Arria GX, Arria II GX, Arria II GZ, Arria V, Arria V GZ, Arria 10, Cyclone, Cyclone II, Cyclone III, Cyclone III LS, Cyclone IV E, Cyclone IV GX, Cyclone V, Stratix, Stratix GX, Stratix II, Stratix II GX, Stratix III, Stratix IV, Stratix V, Stratix 10
- **Lattice**  
MachX02, LatticeECP3, LatticeXP2
- **Microchip (Microsemi)**  
IGLOO2, ProASIC3, ProASIC3E, ProASIC3L, SmartFusion2
- **AMD (Xilinx)**  
Artix-7, Artix UltraScale+, Kintex-7, Kintex UltraScale, Kintex UltraScale+, Spartan-3, Spartan-3A, Spartan-3E, Spartan-6, Spartan-7, Versal, Virtex-II, Virtex-II Pro, Virtex-4, Virtex-5, Virtex-6, Virtex-7, Virtex UltraScale, Virtex UltraScale+, Zynq-7000, Zynq UltraScale+

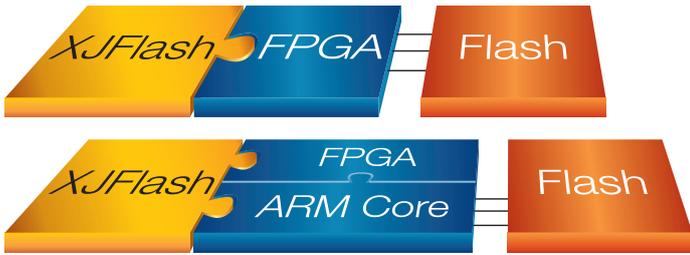
该列表在持续更新，联系我们获取最新情况。



## 我可以使用 XJFlash 吗？

为了使用XJFlash，闪存器件上所有数据、地址和控制信号都必须连接到目标板卡上的FPGA。这可以是配置PROM，或者是连接到通用I/O针脚的闪存器件。这些连接可以是直接、间接、专用的或共享的。

**直接连接--可以**  
闪存直接连接到FPGA。



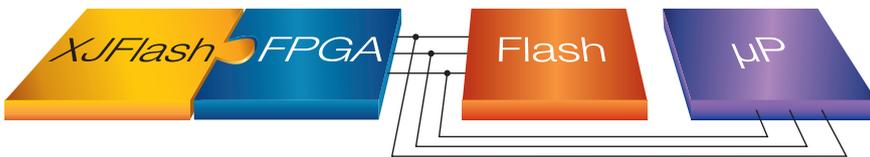
**间接连接--可以**

1. 闪存通过缓存连接到FPGA。
2. 部分地址信号和数据信号共享并且通过latch连接到一起。
3. 存在一个其它可配置的器件，例如闪存和FPGA之间的CPLD。



**共享连接--可以**

闪存和FPGA通过以上描述的模式之一连接，但是这些连接和其它的器件共享（例如一个处理器）。



**无连接--可以（需设计改变）**

如果您的设计包含FPGA而闪存并没有通过以上任何描述的配置之一来连接，那么通过FPGA上闲置的针脚或许可以建立闪存的连接。这些连接不会在板卡的任务模式下使用，但是它们允许您使用XJFlash来执行高速闪存编程。如果您的FPGA是一个与闪存同一个地址、数据总线的从器件，额外的信号或许不是必要的。



**无FPGA--直接不可以**

遗憾的是，如果没有FPGA，XJFlash是不可以使用的，然而通过处理器上的调试接口或许可以执行高速闪存编程--想要知道能否在您的板卡上使用这种方法，请联系我们。或者，如果可以把闪存的连接引到板卡上的连接器，那么通过这个连接器就可以执行XJFlash高速编程。



## 测试集成

XJFlash和XJTAG开发系统其它部分完全兼容。所有XJFlash编程都可以作为XJRunner边界扫描测试项目的一部分来执行。

## 可配置的闪存编程

无论您是否需要编写单一的闪存器件、或者串联的多个器件、或者扩展地址空间、或者并联、或者构造一个更大的数据总线，您都可以使用XJFlash来加速您的编程操作。

## 定制开发

XJFlash还可以应对独立的编程需求来使用，包括直接访问I2C和SPI总线或者定制的协议例如Microchips ICSP。

要求的连接器不需要来自于目标板卡上的FPGA。在板卡连接器上协议信号可用的条件下，可使用作为XJTAG解决方案一部分的XJFlash来达到高速编程。

想要了解这项服务的更多信息，请联系我们。

经销商、技术伙伴