XJFlash

Übersicht

XJFlash ist eine fortschrittliche und innovative Methode für die In-System-Programmierung (ISP) von Flash-Komponenten über JTAG. Mit XJFlash können Sie bis zu 50x schnellere Flash-Programmiergeschwindigkeiten erreichen als dies mit herkömmlichen Boundary-Scan-Techniken möglich ist.

Automatisch generierte, kundenspezifische Lösungen

Mit XJFlash können Sie automatisch benutzerdefinierte Programmierlösungen für die an FPGAs angeschlossenen Flash-Komponenten auf Ihrem Board generieren.

Die Funktionsfähigkeiten der FPGAs werden genutzt, um die schnellstmöglichen Programmiergeschwindigkeiten zu bieten. XJFlash generiert automatisch ein benutzerdefiniertes Design für jede FPGA/Flash-Kombination, so dass Sie die besten Programmierzeiten erreichen, während Sie keinerlei FPGA-Entwicklung erbringen müssen.*

Ob Sie SPI, QSPI oder parallele NOR-Schnittstellen für den Flash verwenden, verbunden mit einem FPGA von Altera, Xilinx, Microsemi oder Lattice: XJFlash bietet Ihnen eine Programmierlösung, die für Ihre Leiterplatte optimiert ist.

**Bei der Konfiguration von XJFlash wird eine lizenzierte Version der entsprechenden Tools des FPGA-Herstellers benötigt. Freie Versionen reichen für viele Geräte aus.*

XJFlash Zeitsteuerung (Beispiel)

Theoretische Mindestzeit für konventionellen Boundary-Scan: *35 Min.*
Gesamte XJFlash-Laufzeit: *10,5 bis 32,6 Sek.*

XJFlash durchläuft jedes Mal automatisch vier Stufen, wenn eine Flash-Komponente programmiert wird:

Initialisierung – Das mit dem Flash verbundene FPGA wird mit dem für die Zielleiterplatte benötigten XJFlash-Bild konfiguriert. Beispielzeit: 2.1 Sek.

Löschung – Der Flash kann gelöscht werden, indem einer von zwei Algorithmen angewandt wird. Die Basis-Löschung löscht einfach alle Blöcke innerhalb eines definierten Bereichs (dies kann der ganze Flash oder nur der Platz, der für das zu programmierende Bild benötigt wird, sein). Die intelligentere Löschung wird die Tatsache nutzen, dass es schneller ist, den Flash zu lesen, als ihn komplett zu löschen. So wird von jeder Adresse ausgehend gelesen und nur dann gelöscht, wenn auch wirklich Daten gefunden werden. Dieser Schritt kann übersprungen werden, wenn bekannt ist, dass der Flash immer unbeschrieben ist, bevor er programmiert wird.

Beispielzeit – intelligente Löschung aktiviert: 0,9 Sek. bei bereits gelöschter Komponente, auf 23 Sek. bei voll programmierter Komponente (begrenzt durch die Löschzeit der Komponente).

Programmierung – Daten aus den Zielbildern werden über den JTAG-Port in das FPGA gestreamt. Das FPGA programmiert diese Daten dann in den/die angeschlossenen Flash(s). Mehrere Dateien können bei definierten Abweichungen spezifiziert und programmiert werden. Dieser Schritt kann ausgelassen werden, wenn lediglich eine Überprüfung erforderlich ist.

Beispielzeit: 6.2 Sek. (begrenzt durch die Programmiergeschwindigkeit der Komponente).

Verifizierung – Prüft jedes Byte im Flash gegen die angegebene(n) Datei(en), um sicherzustellen, dass es keine Datenbitfehler gibt. Dieser Schritt kann ausgelassen werden, wenn nur eine Programmierung oder Löschung erforderlich ist.

Beispielzeit: 1,8 Sek. mit TCK bei 10 MHz, reduziert auf 1,3 Sek. mit TCK bei 20 MHz.

Diese Beispielzeiten sind für eine Spartan 6 XC6SLX9-Programmierung einer pseudozufälligen 2 Mbyte-Datendatei in die FPGA SPI-Konfiguration PROM vorgesehen.

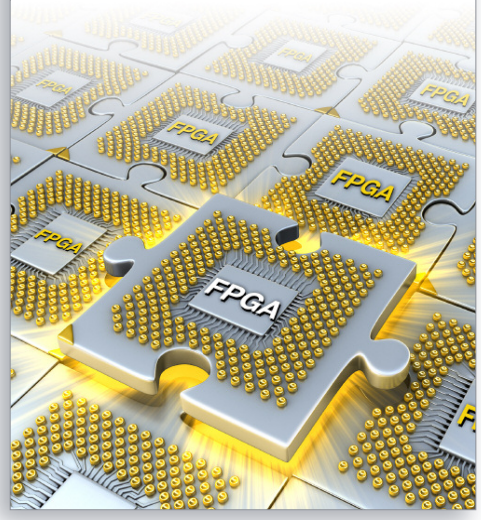
Wesentliche Vorteile

- Reduziert Flash-Programmierzeiten
- Flash-Unterstützung von SPI, QSPI, parallelen NOR-Schnittstellen
- Unterstützung für NAND-Flashgeräte auf Anfrage verfügbar
- Verkürzt Entwicklungszyklen
- Kein zusätzliches Equipment notwendig
- Kann für schnelle Firmware-Upgrades verwendet werden
- Keine FPGA-Entwicklung erforderlich

Unterstützte FPGAs

- **Altera**
  Arria GX, Arria II GX, Arria II GZ, Arria V, Arria V GZ, 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
- **Lattice**
  MachXO2, LatticeECP3, LatticeXP2
- **Microsemi**
  IGLOO2, ProASIC3, ProASIC3E, ProASIC3L, SmartFusion2
- **Xilinx**
  Artix-7, Kintex-7, Kintex UltraScale, Spartan-3, Spartan-3A, Spartan-3E, Spartan-6, Virtex-II, Virtex-II Pro, Virtex-4, Virtex-5, Virtex-6, Virtex-7, Virtex UltraScale, Zynq-7000, Zynq UltraScale+

Diese Liste wird kontinuierlich erweitert, deswegen kontaktieren Sie uns bitte für die aktuellsten Details.

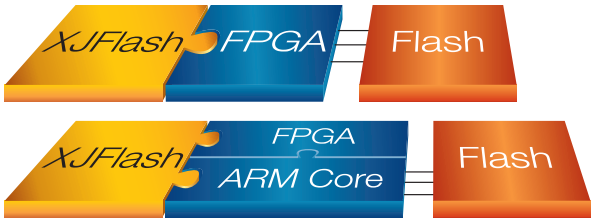


XJFlash

Kann ich XJFlash verwenden?

Um XJFlash anwenden zu können, müssen alle Daten, Adress- und Steuersignale auf dem(n) Flash(s) mit einem FPGA auf der Zielleiterplatte verbunden sein. Dies kann eine PROM-Konfiguration oder eine Flash-Komponente sein, die mit einem beliebigen I/O-Pin verbunden ist. Diese Verbindungen können direkt oder indirekt sein, zugewiesen oder gemeinsam genutzt werden:

DIREKTE VERBINDUNGEN – JA

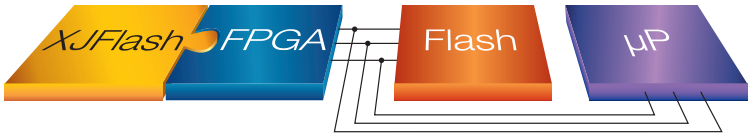


Der Flash ist direkt mit dem FPGA verbunden.

INDIREKTE VERBINDUNGEN – JA

1) Der Flash wird über einen Pufferspeicher mit dem FPGA verbunden. **2)** Einige der Adresssignale werden mit den Datensignalen geteilt und über einen Latch verbunden. **3)** Zudem gibt es andere konfigurierbare Komponenten, wie z.B. ein CPLD, zwischen Flash und FPGA.

GETEILTE VERBINDUNG – JA



Der Flash ist mit dem FPGA in einem der oben beschriebenen Modi verbunden, aber diese Verbindungen werden mit einer anderen Komponente (wie z.B. einem Prozessor) geteilt.

KEINE VERBINDUNG – JA (mit Anpassungen im Design)

Wenn Ihr Design ein FPGA beinhaltet, aber der Flash nicht mit einer der beschriebenen Konfigurationen angeschlossen ist, kann eine Verwendung von Ersatzpins auf dem FPGA möglich sein, um Verbindungen zum Flash herzustellen. Diese Verbindungen werden nicht im Missionsmodus des Boards verwendet, sondern erlauben Ihnen, XJFlash zu verwenden, um schnelles Programmieren des Flashs durchzuführen. Wenn Ihr FPGA ein Slave-Gerät auf der gleichen Adresse/Datenbus wie der Flash ist, wird dies nicht viele zusätzliche Signale erfordern.

KEINE FPGA – Nicht direkt

Leider ist die Anwendung von XJFlash nicht möglich, wenn kein FPGA zur Verfügung steht, aber es kann unter Umständen möglich sein, die schnelle Flash-Programmierung mit dem Debug-Interface auf einem Prozessor durchzuführen – bitte kontaktieren Sie uns, um zu sehen, ob Sie diesen Ansatz auf Ihrem Board realisieren können.

Test-Integration

XJFlash ist voll kompatibel mit dem restlichen XJTAG-Entwicklungssystem. Alle XJFlash-Programmierungen können als Teil eines Boundary-Scan-Testprojekts in XJRunner ausgeführt werden.

Konfigurierbare Flash-Programmierung

Es spielt keine Rolle, ob Sie eine einzelne oder mehrere, in Serie verbundenen, Flash-Komponenten programmieren, den Adressraum oder parallel dazu den Datenbus erweitern: Sie können mit XJFlash Ihre Programmieraktivitäten beschleunigen.

Kundenspezifische Entwicklung

XJFlash kann auch für eigenständige Programmieranforderungen verwendet werden, einschließlich direkten Zugriffs auf I²C- und SPI-Busse oder benutzerdefinierte Protokolle wie Microchips ICSP.

Die benötigten Anschlüsse müssen nicht von einem FPGA auf der Zielplatine kommen. Wenn die Protokollsignale auf einem Header auf dieser Leiterplatte zur Verfügung stehen, sollte es möglich sein, XJFlash zu verwenden, um eine schnelle Programmierung als Teil einer XJTAG-Lösung zu erreichen.

Bitte kontaktieren Sie uns für weitere Informationen über diesen Service.

Vertriebspartner / Technologiepartner

www.xjtag.com/partners