

## Overview

XJFlash is an advanced and innovative method for In-System Programming (ISP) of flash devices through JTAG. Using XJFlash you can achieve flash programming speeds up to 50 times faster than those possible using conventional boundary scan techniques.

### Custom Solutions – Automatically Generated

XJFlash allows you to automatically generate customised programming solutions for the flash devices connected to FPGAs on your board.

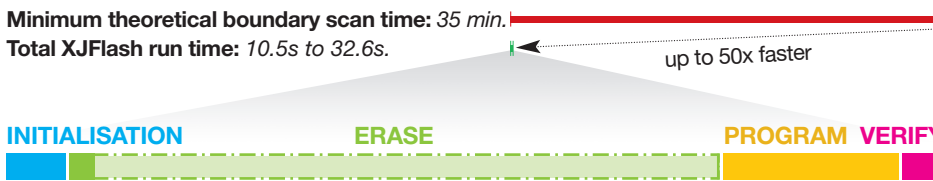
The functional capabilities of the FPGA are harnessed to provide the fastest possible programming speeds. XJFlash automatically generates a custom

design for each FPGA/flash combination, allowing you to achieve the best programming times, whilst not requiring you to do any FPGA development.\*

Whether you are using SPI, QSPI or parallel NOR flash connected to an FPGA from Altera, Xilinx, Microsemi or Lattice – XJFlash will provide you with a programming solution optimised for your board.

\*A licensed version of the relevant FPGA manufacturer's tools will be required during the configuration of XJFlash. Free versions are sufficient for many devices.

### XJFlash Example Timings



XJFlash will automatically step through four stages each time a flash device is programmed:

**Initialisation** – The FPGA connected to the flash is configured with the XJFlash image required for the target board.  
Example time: 2.1 s.

**Erase** – The flash can be erased using one of two algorithms. The basic erase will simply erase all blocks within a defined range (this may be the whole flash or just the space needed for the image to be programmed). The more intelligent erase will use the fact that it is quicker to read the flash than to erase it; as such it reads from each address and only starts erasing if some data is found. This step can be skipped if it is known that the flash will always be blank before it is programmed.

Example time – intelligent erase enabled: 0.9 s with a device already erased, to 23 s with a fully programmed device (limited by erase time of device).

**Program** – Data from the target images is streamed into the FPGA through its JTAG port. The FPGA then programs this data into the connected flash(s). Multiple files can be specified and programmed at defined offsets. This step can be bypassed if only verification is required.

Example time: 6.2 s (limited by the programming speed of the device).

**Verify** – The verification checks every byte in the flash against the specified file(s), ensuring there are no data bit errors. This step can be bypassed if only programming or erasing is required.  
Example time: 1.8 s with TCK at 10 MHz, reducing to 1.3 s with TCK at 20 MHz.

These example times are provided for a Spartan 6 XC6SLX9 programming a 2 MByte pseudo-random data file into the FPGA's SPI configuration PROM.

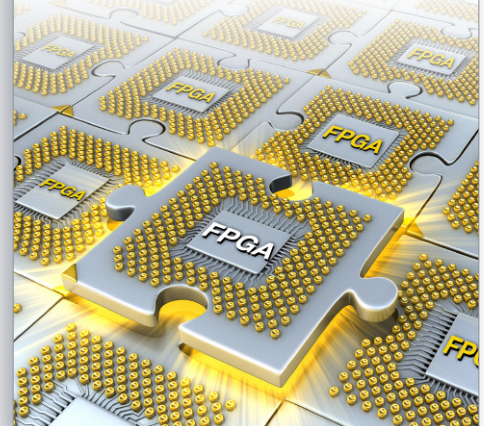
### Key Benefits

- Reduce flash programming times
- SPI, QSPI, parallel NOR flash devices supported
- Support for NAND flash devices available on request
- Shortened development cycles
- No need for additional equipment
- Can be used for fast firmware upgrade
- No FPGA development required

### Supported FPGAs

- **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**  
MachXO2, 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+

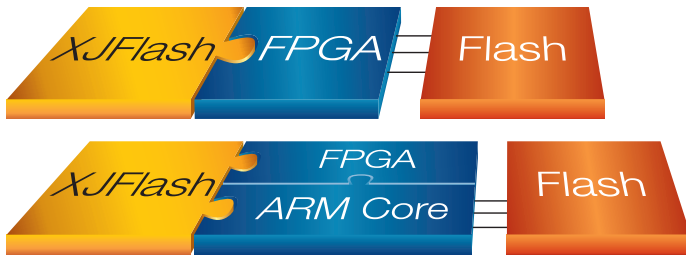
This list is continuously growing, so contact us for the latest details!



## Can I use XJFlash?

In order to use XJFlash, all of the data, address and control signals on the flash device(s) must be connected to an FPGA on the target board. This can be a configuration PROM, or a flash device connected to any general purpose I/O pin. These connections can be direct, indirect, dedicated or shared:

### DIRECT CONNECTIONS – YES



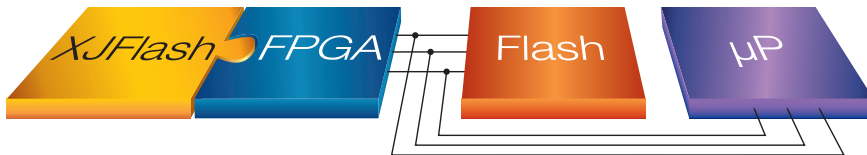
The flash is directly connected to the FPGA.

### INDIRECT CONNECTIONS – YES



1. The flash is connected to the FPGA via a buffer.
2. Some of the address signals are shared with the data signals and connected via a latch.
3. There is another configurable device, such as a CPLD between the flash and the FPGA.

### SHARED CONNECTIONS – YES



The flash is connected to the FPGA in one of the modes described above but these connections are shared with another device (such as a processor).

### NO CONNECTIONS – YES (with design changes)



If your design contains an FPGA but the flash is not connected in any of the configurations described, it may be possible to use spare pins on the FPGA to establish connections to the flash. These connections would not be used in the mission mode of the board but would allow you to use XJFlash to perform fast flash programming. If your FPGA is already connected to the same address/data bus as the flash then this may not require many extra signals.

### NO FPGA – Not directly

Unfortunately, it is not possible to use XJFlash if there is no FPGA, but it may be possible to do fast flash programming using the debug interface on a processor – please contact us see if you can use this approach on your board. Alternatively, if it is possible to bring the connections of the flash to a header on the board, XJFlash fast programming can be performed directly via that connector.



## Test Integration

XJFlash is fully compatible with the rest of the XJTAG development system. All XJFlash programming can be run as part of an XJRunner boundary scan test project.

## Configurable Flash Programming

It doesn't matter whether you need to program a single flash device, or multiple devices that are connected in series, to expand the address space, or in parallel, to make a wider data bus, you can use XJFlash to speed up your programming operations.

## Custom Development

XJFlash can also be used for standalone programming requirements including direct access to I<sup>2</sup>C and SPI busses or custom protocols such as Microchips ICSP.

The required connections do not need to come from an FPGA on the target board. Providing the protocol signals are available on a header on that board, it should be possible to use XJFlash to achieve fast programming as part of an XJTAG solution.

Please contact us for more details of this service.

Distributor / Technology Partner