

## Обзор

XJFlash — это программное приложение для внутрисистемного программирования (In-System Programming – ISP) flash памяти через JTAG. XJFlash позволяет ускорить процесс программирования до 50 раз по сравнению со стандартным программированием через boundary scan.

### Индивидуальная “прошивка” — Генерируется автоматически

XJFlash автоматически генерирует “прошивку” ПЛИС, к которой подключена flash память, которую требуется запрограммировать.

Для максимально возможного ускорения программирования задействуются все имеющиеся возможности конкретной использованной ПЛИС. Конфигурация ПЛИС генерируется индивидуально в каждом конкретном случае.

Также анализируется конкретное сочетание ПЛИС/Память и оптимальная конфигурация ПЛИС генерируется под данную пару.\*

Вне зависимости от типа flash памяти – SPI, QSPI или parallel NOR, и вне зависимости от использованной ПЛИС – Altera, Xilinx, Microsemi или Lattice – XJFlash сгенерирует максимально оптимальную для конкретной печатной платы конфигурацию ПЛИС.

\* В процессе конфигурирования ПЛИС приложению XJFlash требуется лицензионная версия программных приложений от производителей ПЛИС. Как правило, существуют бесплатные версии требуемых приложений.

### Пример затрат времени на процесс программирования

Минимум при использовании boundary scan: 35 мин.

Через XJFlash: от 10.5 сек. до 32.6 сек.

до 50-ти раз быстрее



Каждый раз во время программирования XJFlash автоматически выполняет следующую последовательность действий:

**■ Инициализация** — XJFlash загружает в ПЛИС, подключенную к flash памяти, двоичный образ конфигурации ПЛИС. Данный этап может длиться, например, 2.1 сек.

**■ Стирание** — flash память может быть очищена при помощи одного из двух алгоритмов. Базовый алгоритм просто удаляет данные из всех блоков в заданном диапазоне (это может быть как вся имеющаяся память, так и точный объём, необходимый под загрузку образа). Интеллектуальный алгоритм стирания данных учитывает тот факт, что предварительное чтение данных перед их стиранием ускоряет процесс стирания; в связи с этим, производится последовательное чтение данных, а стирание начинается, если найдены определённые данные. Стирание можно не выполнять, если точно известно, что перед записью flash память будет всегда пустой.

Интеллектуальный алгоритм может длиться, например, от 0.9 сек., в случае, когда память уже пуста, до 23 сек., в случае, когда требуется стирание данных.

**■ Программирование** — данные образа для записи в память передаются в ПЛИС через JTAG порт. Затем сама ПЛИС записывает данные образа в подключенную flash память. Можно записать сразу несколько образов с заданными смещениями. Данный этап можно пропустить если требуется только проверка целостности данных памяти.

Данный этап может длиться, например, 6.2 сек. и ограничен только скоростью записи данных в flash память.

**■ Проверка** — выполняется побайтовое сравнение данных flash памяти с данными переданного образа. Данный этап можно пропустить, если требуется только программирование или только стирание.

Данный этап может длиться, например, 1.8 сек. при частоте TCK 10 MHz, или 1.3 сек. при частоте TCK 20 MHz.

Примеры временных затрат приведены для записи псевдо-случайных данных в объёме 2 MByte в SPI configuration PROM, подключенную к ПЛИС Spartan 6 XC6SLX9.

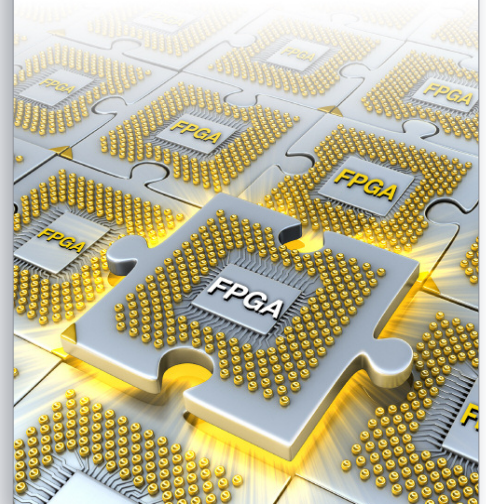
## Выгоды

- Ускорение программирования flash памяти
- Поддержка SPI, QSPI, parallel NOR
- Работа с NAND flash — по запросу
- Ускорение проектирования всего прибора
- Не требуется дополнительное оборудование
- Применимо при обновлении “прошивки”
- Не требуется разработка проекта ПЛИС

## Поддерживаемые ПЛИС

- **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**  
MachX02, 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+

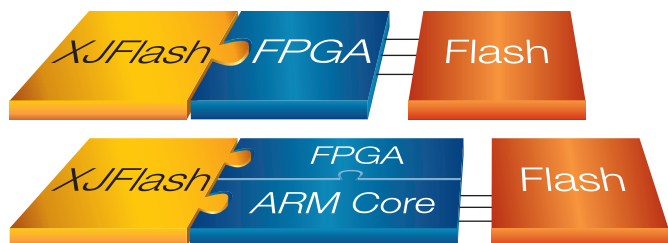
Список постоянно пополняется, свяжитесь с нами, мы вышлем актуальный!



## Применим ли XJFlash в конкретном проекте?

Для того, чтобы XJFlash можно было применить, требуется, чтобы все шины данных и адресов, а также все управляющие сигналы были подключены к ПЛИС. То есть и конфигурационная PROM и любая другая микросхема flash памяти должна быть подключена к контактам ввода/вывода общего назначения (general purpose I/O) ПЛИС. Такое подключение может быть прямым, опосредованным, специализированным или общим:

### ПРЯМОЕ ПОДКЛЮЧЕНИЕ — Применим



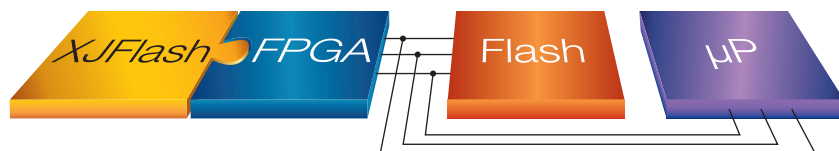
Flash память напрямую подключена к ПЛИС.

### ОПОСРЕДОВАННОЕ ПОДКЛЮЧЕНИЕ — Применим



1. Flash память подключена к ПЛИС через буфер.
2. Некоторые шины адреса используются ещё и как шины данных и подключаются через защёлку.
3. Есть ещё одна конфигурируемая микросхема, например, ещё одна ПЛИС между основной ПЛИС и памятью.

### ОБЩЕЕ ПОДКЛЮЧЕНИЕ — Применим



Flash память подключена к ПЛИС одним из способов, перечисленных выше, но использованные шины одновременно подключены ещё и к другой микросхеме (например, к процессору).

### НЕТ ПОДКЛЮЧЕНИЯ — Применим (требуется внести изменения в проект)



Если проект содержит ПЛИС, но flash память не подключена к этой ПЛИС ни одним из перечисленных выше способов, возможно, получится подключиться к памяти при помощи незадействованных контактов ПЛИС. Эти незадействованные контакты ПЛИС можно подключить к памяти исключительно с целью быстрого программирования через XJFlash. Если ПЛИС на плате работает в режиме slave и подключена к тем же шинам адреса/данных, что и flash память, то в этом случае потребуется провести совсем немного дополнительных цепей.

### ПЛИС ОТСУТСТВУЕТ — Применим обходными путями

К сожалению, если ПЛИС на плате отсутствует, то применить XJFlash, скорее всего, не получится, но есть вероятность, что получится использовать отладочные возможности процессора, если таковой имеется на плате — пожалуйста, свяжитесь с нами, мы рассмотрим проект и ответим на вопрос получится ли ускорить программирование памяти. Другой вариант — если есть возможность вывести контакты flash памяти на какой-нибудь коннектор платы, то можно выполнить быстрое программирование непосредственно через коннектор.



### Как часть процесса тестирования

XJFlash полностью совместим с остальными инструментами системы XJTAG и может выполнять все свои функции как часть процесса тестирования, выполняемого в XJRunner.

### Адаптивность

XJFlash применим для ускорения программирования как одной микросхемы памяти, так и нескольких микросхем, соединённых как последовательно, для увеличения адресного пространства, так и параллельно, для увеличения разрядности шины данных.

### Частный случай

XJFlash применим в том числе и в случае наличия специфических требований, например, прямой доступ к шинам I2C и SPI или использование частных протоколов, таких как Microchips ICSP.

Перечисленные выше шины и интерфейсы не обязательно должны быть подключены к ПЛИС. Простой вывод этих шин и интерфейсов на коннектор платы уже должен позволить XJFlash выполнить быстрое программирование.

Пожалуйста, свяжитесь с нами для обсуждения деталей проекта платы.

Дистрибьютор / Технологический Партнер