XFLASH Command-Line Manual
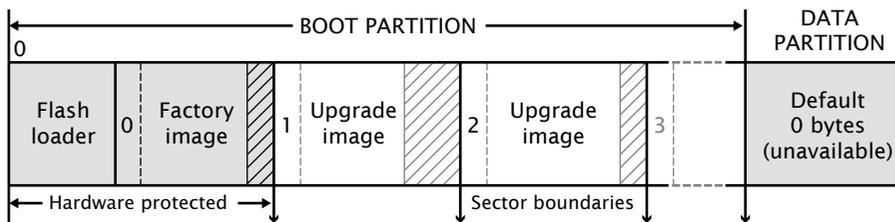
XFLASH creates binary files in the xCORE flash format, as illustrated in the diagram below. It can also program these files onto flash devices used to boot XMOS systems.

Figure 1: Flash format diagram

1 Overall Options

The following options are used to specify the program images and data that makes up the binary and its layout. Padding is inserted when required to ensure that images are aligned on sector boundaries.

```
xe-file [size]
--factory xe-file [size]
```

> Specifies *xe-file* as the factory image. If *size* is specified, padding is inserted to make the space between the start of this image and the next image at least the specified size. The default unit of *size* is "bytes;" the size can be postfixed with `k` to specify a unit of kilobytes.
>
> At most one factory image may by specified.

```
--upgrade id xe-file [size]
```

> Specifies *xe-file* as an upgrade image with version *id*. Each version number must be a unique number greater than 0. If *size* is specified, padding is inserted to make the space between the start of this image and the next image at least the specified size. The default unit of size is "bytes;" the size can be postfixed with `k` to specify a unit of kilobytes.

Multiple upgrade images are inserted into the boot partition in the order specified on the command line.

If no factory image is specified, a single upgrade image may be specified and written to a file with the option `-o`.

`--factory-version version`

Specifies version as the tools release master version that was used to create the factory image. Accepted values are: 13.0, 13.1, 13.2 and 14.0. This option need only be specified when `--upgrade` is provided but `--factory` is not. This option will ensure that the produced flash upgrade image is of the correct format for the installed factory image.

`--boot-partition-size n`

Specifies the size of the boot partition to be *n* bytes. If left unspecified, the default size used is the total size of the flash device. *n* must be greater than or equal to the minimum size required to store the boot loader, factory image and any upgrade images. XFlash will round up the actual boot partition size to the next sector boundary in flash memory.

`--data file`   Specifies the contents of *file* to be written to the data partition.

`--loader file`

Specifies custom flash loader functions in *file* (see XM-000949-PC). The file may be either an object (`.o`) or archive (`.a`).

By default, the xCORE flash loader loads the image with the highest version number that validates against its CRC.

`--verbose`   Prints additional information about the program when loaded onto the target system.

`--help`   Prints a description of the supported command line options.

`--version`   Displays the version number and copyrights.

2   Target Options

The following options are used to specify which flash device the binary is to be programmed on. The type of flash device used determines the values for the SPI divider, sector size and memory capacity.

`--list-devices`
`-l`   Prints an enumerated list of all JTAG adapters connected to the PC and the devices on each JTAG chain, in the form:

```
ID        Name        Adapter ID        Devices

--        ----        ----------        -------
```

The adapters are ordered by their serial numbers.

`--id ID`   Specifies the adapter connected to the target hardware.

XFLASH connects to the target platform and determines the type of flash device connected to it.

`--adapter-id ADAPTER-SERIAL-NUMBER`
Specifies the serial number of the adapter connected to the target hardware. XFLASH connects to the target hardware and determines the type of flash device connected to it.

`--jtag-speed n`
Sets the divider for the JTAG clock to *n*. The corresponding JTAG clock speed is $6/(n+1)$MHz. The default value of the divider for the JTAG clock is 0, representing 6MHz.

`--spi-spec file`
Enables support for the flash device specified in *file* (see XM-000941-PC).

`--spi-div n` Sets the divider for the SPI clock to *n*, producing an SPI clock speed of $100/2n$MHz. By default, if no target is specified, the divider value is set to 3 (16.7MHz).

`--fast-spi-div n`
Sets the divider for the QuadSPI clock used during fast boot to *n*, producing an QuadSPI clock speed of xcore_clock_speed/$2n$MHz. By default the divider value is set to 5 (50MHz with a 500MHz xcore_clock). Supported divide values for fast boot are 3, 4, 5 and 6

`--spi-read-id cmd`
Reads the spi manufacturer's id from the attached device. The *cmd* can be obtained from the spi manufacturer's datasheet. If there is more than one device in a network then all id's will be returned.

`--noinq` Does not run the device inquisitor program, which checks that images are aligned on sector boundaries. If `--noinq` is omitted XFLASH expects to be able to connect to the device via JTAG.

`--force-pll-reset`
Will force an xCORE-200 device to reset when the PLL register is written during the boot process. By default the xCORE-200 device will not reset when the PLL register is written allowing for faster boot times.

`--image-search-page`
Will instruct the second stage bootloader to search flash memory for potential upgrade images at every page boundry within the boot partition. This was the default search mechanism in tools 14.0 and all previous tools versions.

`--image-search-sector`
Will instruct the second stage bootloader to search flash memory for potential upgrade images at every sector boundry within the boot partition. This is the new default search mechanism in tools 14.2.

`--image-search-address address`
Will instruct the second stage bootloader to search flash memory for potential upgrade images at a specified address within the boot partition. This option can

be provided up to a maximum of three times, allowing for 3 seperate upgrade images to be present within flash memory.

3 Security Options

The following options are used in conjunction with the AES Module (see XM-000955-PC).
`--key keyfile`
> Encrypts the images in the boot partition using the keys in *keyfile*.

`--disable-otp`
> Causes the flash loader to disable access to OTP memory after the program is booted. This is default if the option `--key` is used.

`--enable-otp`
> Causes the flash loader to enable access to OTP memory after the program is booted. This is default unless the option `--key` is used.

4 Programming Options

By default, XFLASH programs the generated binary file to the target flash device.
`-o file` Places output in *file*, disabling programming.

> If the target platform is booted from more than one flash device, multiple output files are created, one for each device. The name of each output file is `file`_*node*, where *node* is the value of the `Id` attribute (see XM-000929-PC) of the corresponding node.

`--arm-binary file`
> Places XFLASH output into the reserved flash memory area of the ARM binary *file*.

> The given ARM binary file will not be directly modified, but a copy of the file will be created. If the option `-o` *output file* is given then *output file* will contain the modified ARM binary. If the `-o` option is not supplied then a temporary file will be created and programmed directly onto the attached ARM device.

The following options perform generic read, write and erase operations on the target flash device. A target XN file must be specified, which provides ports used to communicate with the SPI device on the hardware platform.
`--target-file xn-file [node]`
> Specifies *xn-file* as the target platform.

> If *xn-file* specifies more than one flash device, a value for *node* must be specified. This value must correspond to the `Id` attribute (see XM-000929-PC) of the node connected to the target flash device.

`--target platform [node]`
> Specifies a target platform. The platform configuration must be specified in the file `platform.xn`, which is searched for in the paths specified by the `XCC_DEVICE_PATH` environment variable (see XM-000927-PC).

> If *xn-file* specifies more than one flash device, a value for *node* must be specified. This value must correspond to the `Id` attribute (see XM-000929-PC) of the node connected to the target flash device.

`--erase-all`   Erases all memory on the flash device.

`--read-all`   Reads the contents of all memory on the flash device and writes it to a file on the host. Must be used with `-o`.

`--write-all file`
             Writes the bytes in *file* to the flash device.

`--no-reset-on-write`
             Prevents XFLASH from resetting the xCORE after programming the device.

`--no-compression`
             Prevents XFLASH from compressing all binary data.

`--no-compression-validation`
             Prevents XFLASH from validating the decompression algorithm within XSIM.

XMOS®

Copyright © 2016, All Rights Reserved.