

Linux kernel for ARC processors

Other sources of information

Below are some resources where more information can be found on ARC processors and relevant open source projects.

- <https://enbarc.org> - Community portal for open source on ARC. Good place to start to find relevant FOSS projects, toolchain releases, news items and more.
- <https://github.com/foss-for-synopsys-dwc-arc-processors> - Home for all development activities regarding open source projects for ARC processors. Some of the projects are forks of various upstream projects, where "work in progress" is hosted prior to submission to upstream projects. Other projects are developed by Synopsys and made available to community as open source for use on ARC Processors.
- [Official Synopsys ARC Processors website](#) - location, with access to some IP documentation ([Programmer's Reference Manual, AKA PRM for ARC HS processors](#)) and free versions of some commercial tools ([Free nSIM](#) and [MetaWare Light Edition](#)). Please note though, registration is required to access both the documentation and the tools.

Important note on ARC processors configurability

ARC processors are highly configurable and several configurable options are supported in Linux. Some options are transparent to software (i.e cache geometries, some can be detected at runtime and configured and used accordingly, while some need to be explicitly selected or configured in the kernel's configuration utility (AKA "make menuconfig").

However not all configurable options are supported when an ARC processor is to run Linux. SoC design teams should refer to "Appendix E: Configuration for ARC Linux" in the ARC HS Databook for configurability guidelines.

Following these guidelines and selecting valid configuration options up front is critical to help prevent any unwanted issues during SoC bringup and software development in general.

Building the Linux kernel for ARC processors

The process of kernel building for ARC processors is the same as for any other architecture and could be done in 2 ways:

- Cross-compilation: process of compiling for ARC targets on a development host with a different processor architecture (generally x86_64/amd64).
- Native compilation: process of compiling for ARC on a ARC platform (hardware board or a simulator like QEMU) with complete development environment (GNU toolchain, dtc, make etc) installed on the platform.

In both cases, up-to-date GNU toolchain for ARC for the host is needed. Synopsys offers prebuilt toolchain releases which can be used for this purpose, available from:

- Synopsys GNU toolchain releases: <https://github.com/foss-for-synopsys-dwc-arc-processors/toolchain/releases>
- Linux kernel compilers collection: <https://mirrors.edge.kernel.org/pub/tools/crosstool>
- Bootlin's toolchain collection: <https://toolchains.bootlin.com>

Once the toolchain is installed in the system, make sure its "bin" folder is added in your `PATH` environment variable. Then set `ARCH=arc & CROSS_COMPILE=arc-linux` (or whatever matches installed ARC toolchain prefix) and then as usual `make defconfig && make`.

This will produce "vmlinux" file in the root of the kernel source tree usable for loading on the target system via JTAG. If you need to get an image usable with U-Boot bootloader, type `make uImage` and `uImage` will be produced in `arch/arc/boot` folder.