Intel® Open FPGA Stack (Intel® OFS) is a scalable, source-accessible hardware and software infrastructure provided with universal verification methodology (UVM) environment that addresses the challenges associated with designing FPGA-based acceleration platform solutions. Intel OFS enables software, hardware, and application developers to:

- Use standard interfaces and application programming interfaces (APIs) to accelerate workload development and enable code reuse
- Port existing workloads to the accelerator functional unit (AFU) region to proliferate across Intel OFS-based platforms
- Utilize workloads provided by Intel and third parties directly or further customize existing workloads
- Deploy bare metal, virtualized, or containerized applications with data center class management for network functions virtualization (NFV), SmartNIC, virtual radio access network (vRAN), financial services industry (FSI), and more
- Build application-specific FPGA interface managers (FIMs) with modular and composable code allowing a ‘take and tailor’ FPGA design approach
- Leverage open-sourced and upstreamed software code delivered via Git Repositories with native support from leading software vendors
Use the source-accessible, modular infrastructure to quickly create application specific FIMs tailored to the differentiated needs of various boards and target workloads.

Leverage the Open Programmable Acceleration Engine (OPAE) software development kit, libraries, and APIs along with upstreamed, open-source kernel drivers that target the FIM and accelerate integration into common application frameworks.

Tap into a proven infrastructure and growing ecosystem to achieve greater return on investment and proliferation of workloads across a growing number of Intel OFS-based platforms.

Board Developers

Software Developers

Application Developers

What’s Inside

Hardware

AFU Region

What it is: The AFU is the algorithm or processing unit that provides FPGA-based acceleration and/or CPU offload.

Why it’s important: Workload performance is improved by offloading a computational operation for an application from the CPU to the FPGA’s AFU region. The AFU region includes an optional Partial Reconfiguration (PR) region for dynamic configuration. Providing a designated area with specified insertion points for users to customize their application allows for cleaner, easier to manage register transfer level (RTL) design.

FIM

What it is: Reference FIMs provide platform management, functionality, clocks, resets, and standard interfaces to the host and AFU.

Why it’s important: These easy-to-use reference FIMs are provided as working examples to help users develop their own custom platforms. They are designed to be modular and scalable, so customers can ‘take and tailor’ the design blocks relevant to their application.

Board Management Controller

What it is: The Board Management Controller (BMC) is responsible for controlling, monitoring, and granting access to board features.

Why it’s important: The BMC provides root of trust and authentication to verify that bitstreams come from a trusted source. It interfaces to the server management controller using industry standard protocols.

Software

Upstreamed, Open-Source Kernel Drivers

What it is: The Intel OFS kernel drivers are the first layer in the FPGA software stack.

Why it’s important: This layer decomposes implemented functionality into sets of individual device features, providing a clean and extensible framework for the creation and integration of additional functionalities and their features. Kernel drivers are upstreamed to the Linux kernel (kernel.org) to enable native support for Intel OFS by third-party open-source software vendors.

OPAE Libraries and APIs

What it is: The OPAE C libraries are lightweight, user-space libraries that provide abstraction for FPGA resources in a compute environment.

Why it’s important: This library abstracts hardware and operating system (OS) specific details, exposing the underlying FPGA resources as a set of features accessible from within software running on the host. The unified C API supports different FPGA integration and deployment models to enable portability across OS’s and bare metal, hypervisor, or containerized use models.

Example Applications

What it’s important: Intel OFS provides example FIM designs that users can leverage as a starting point for custom designs. These designs target specific programmable acceleration cards (PACs), Infrastructure Processing Units (IPUs), or development kits.

UVM Test Framework

What it is: The UVM test framework provides a verification environment for the FIM and new features.

Why it’s important: UVM delivers a modular, reusable, and scalable testbench structure by providing an API framework and unit test cases that can be deployed across multiple projects.

OpenCL™ Shim

What it is: A collection of hardware and software components enabling OpenCL™ workload support.

Why it’s important: OpenCL abstracts the FPGA hardware and design flow, enabling users to design FPGA custom hardware with a C-based software language.

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos.
Open Source Methodology
All Intel OFS hardware and software code has been developed using an open source development methodology, resulting in code organized and delivered through git repositories. We refer to our model as being source-accessible because users are granted access to source code in the git repositories. Technical documentation is co-located with the code itself.

Intel will allow users to submit code contributions back from users who wish to share with the Intel OFS community. This will enable the Intel OFS infrastructure to grow and be enriched more quickly.

Intel FPGA-Based Acceleration Platforms
Intel FPGA-based Infrastructure Processing Units (IPUs) and PACs are production-qualified, acceleration hardware solutions designed for telecommunications, cloud services, and other data center environments. The Intel® FPGA PAC D5005 and Silicom FPGA IPU N5010 are the first Intel-OFS enabled platforms using the Intel® Stratix® 10 FPGA, with more Intel OFS-enabled platforms to come.