ACCELERATING DATACENTER WITH FPGA

Uniquest

Agenda

- Market Trends : Data Explosion, Software Defined X
- What is FPGA? Why FPGA for Data Center?
- Use Examples
- Acceleration Stack for Intel® Xeon® CPU with FPGAs

Market Trends : **Data Explosion**

BY 2020

The average internet user will generate
~1.5 GB OF TRAFFIC PER DAY

Smart hospitals will be generating over
3,000 GB PER DAY

Self driving cars will be generating over
4,000 GB PER DAY… EACH

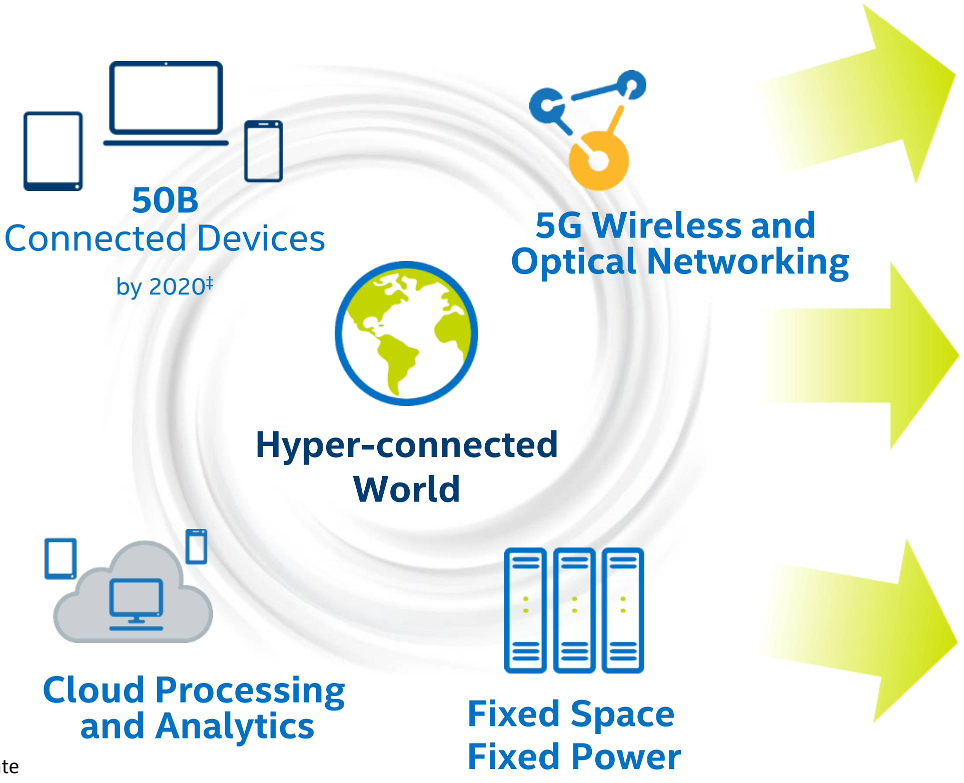
A connected plane will be generating over
40,000 GB PER DAY

A connected factory will be generating over
1,000,000 GB PER DAY

RADAR	~10-100 KB	PER SECOND
SONAR	~10-100 KB	PER SECOND
GPS	~50 KB	PER SECOND
LIDAR	~10-70 MB	PER SECOND
CAMERAS	~20-40 MB	PER SECOND
1 CAR	5 EXAFLOPS	PER HOUR

All numbers are approximated
http://www.cisco.com/c/en/us/solutions/service-provider/vni-network-traffic-forecast/infographic.html
http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html
https://datafloq.com/read/self-driving-cars-create-2-petabytes-data-annually/172
http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html
http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html

Market Trends : **Data Explosion**



50B
Connected Devices
by 2020[‡]

5G Wireless and Optical Networking

Hyper-connected World

Cloud Processing and Analytics

Fixed Space Fixed Power

Market Examples

- Financial
- Genomics
- Government
- Enterprise
- Cloud

Infrastructure

- Network
- Storage
- Compute

Applications

- Security
- Transcode
- Video processing and analytics
- Artificial Intelligence
- Packet processing

[‡] Intel Estimate

Market Trends : **Software Defined X**

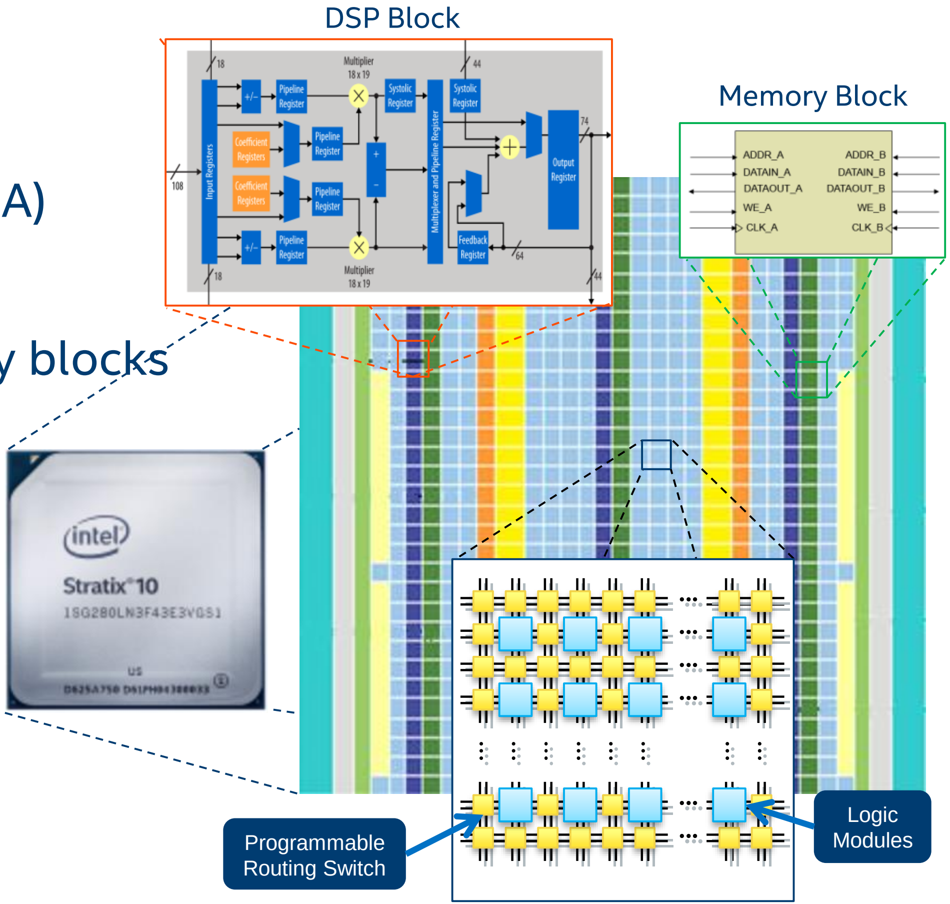
Dedicated Infrastructures
Physical Appliances
Network/Dedicated Servers

Flexible Cloud Infrastructure
Commercial Off the Shelf (COTS) Servers

Uniform
Scalable
Programmable(**Reconfigurable**)

FPGA Overview

- Field Programmable Gate Array (FPGA)
  - Millions of logic elements
  - Thousands of embedded memory blocks
  - Thousands of DSP blocks
  - Programmable routing
  - High speed transceivers
  - Various built-in hardened IP
- Used to create **Custom Hardware!**



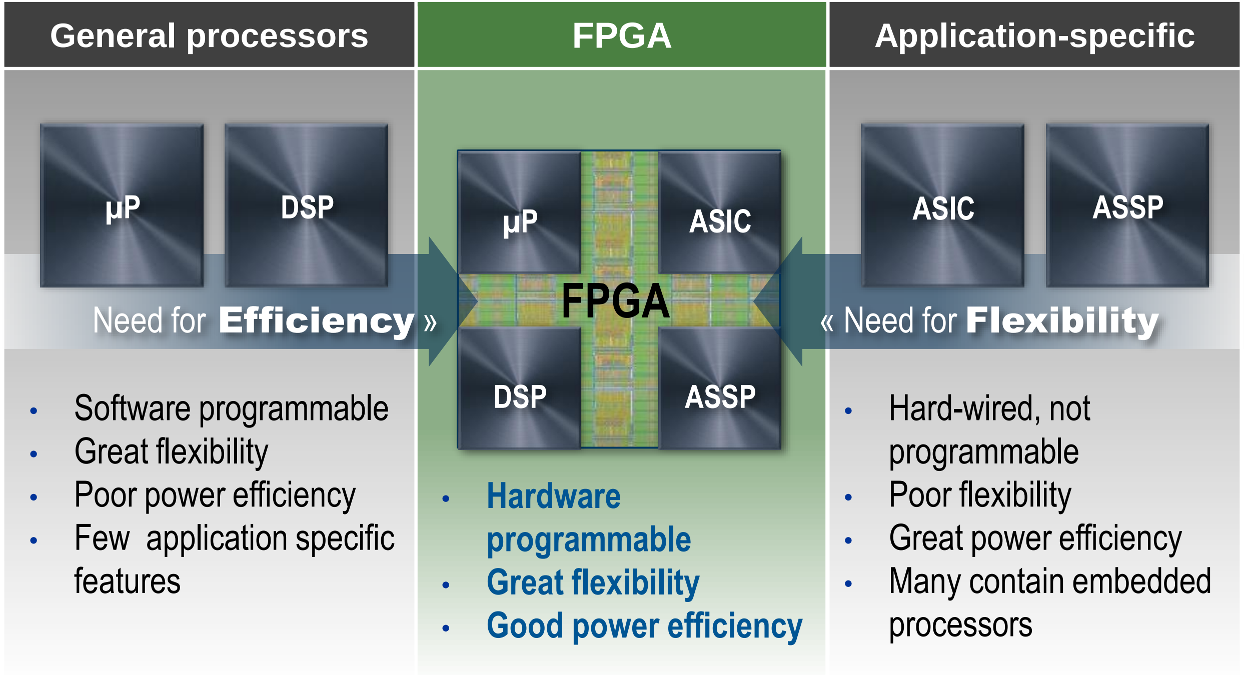
DSP Block

Memory Block

Programmable Routing Switch

Logic Modules

Advantages of Custom Hardware with FPGAs



General processors

- Software programmable
- Great flexibility
- Poor power efficiency
- Few application specific features

FPGA

- **Hardware programmable**
- **Great flexibility**
- **Good power efficiency**

Application-specific

- Hard-wired, not programmable
- Poor flexibility
- Great power efficiency
- Many contain embedded processors

*ARM is a registered trademark of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

FPGA Custom Hardware

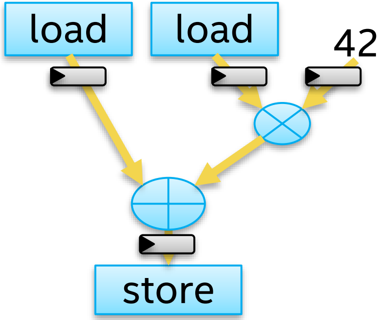
Custom Datapath on the FPGA Matches Your Algorithm!

- Creates typically very deeply pipelined version of a kernel
  - Huge number of operations simultaneously inflight
- Data can more easily be localized on chip

High-level code

```
Mem[100] += 42 * Mem[101]
```

Custom datapath



load

load

42

store

Build exactly what you need:

Operations

Data widths

Memory size & configuration

Efficiency:

Throughput / Latency / Power

FPGA Enabled Performance and Agility

Workload Optimization: ensure Xeon cores serve their highest value processing

Workload 1

Workload 2

Workload N

Efficient Performance: improve performance/watt

Real-Time: high bandwidth connectivity and low-latency parallel processing

Milliseconds

Developer Advantage: code re-use across Intel FPGA data center products

Acceleration Stack for Intel® Xeon® CPU with FPGAs

Intel Environment | Code re-use | IP Libraries

The Intel® Xeon® processor with FPGA acceleration can reduce TCO and solve new problems

Acceleration types in a Data Center

- Application Acceleration : Part of the application domain
  - Artificial Intelligence, Video Transcoding, HPC, …
- Infrastructure Acceleration : Part of the data center infrastructure
  - Virtual Switching, Software defined Networking, compression, cryptography, packet processing, …

Use Examples

- Microsoft
  - SmartNIC
- GATK
- SQL Acceleration
- Public Cloud Service
  - AWS F1 Instance
  - Alibaba
  - NIMBIX

Why FPGAs As Accelerators?

CLOUD
DATA CENTER

FPGAs Maximize ROI

- ✓ One Architecture **efficiently** implements many workloads
- ✓ Application flexibility
- ✓ Reconfiguration in µs
- ✓ Power efficient

Modern Data Center Facts

- ❑ 3-5 year Life Cycle
- ❑ High CAPEX and OPEX
- ❑ Requirement to support rapid scale-out
- ❑ Flexibility to adapt to rapidly changing workloads

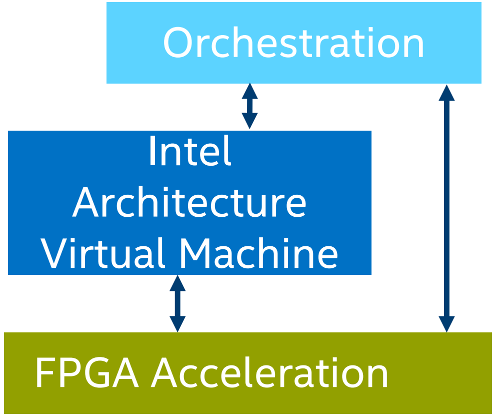
Alternative Accelerators

	GPU	Network Processor	FPGA
Throughput	✓ High	✓ High	✓ High
Latency	High	✓ Low	✓ Low
Power	High	✓ Low	✓ Low
General computing	✓ Yes	No	✓ Yes

FPGA provides optimal networking & compute combination

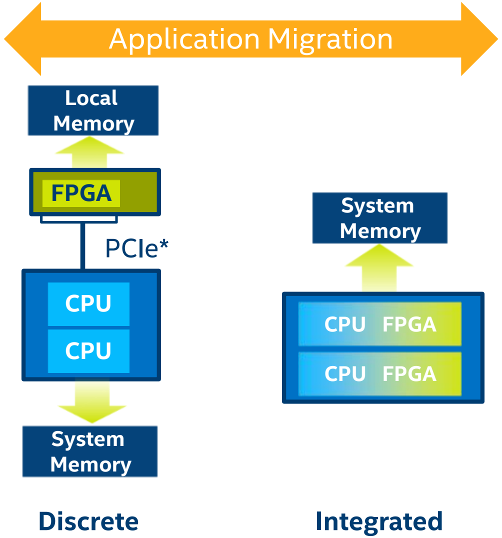
Why **Intel® FPGAs** in the Data Center?

Extended Intel® Architecture



Intel® Solutions and Development

Form-factor Choice



And a mix of Integrated & Discrete

FPGA Acceleration Enabled with the Extended Intel® Architecture

Acceleration Stack for Intel® Xeon® CPU with FPGAs

Dynamically Allocate Intel® FPGAs for Workload Optimization	**Rack-Level Solutions**	intel Rack Scale Design openstack
Simplified Application Development	**User Applications**	Deep Learning, Networking, Genomics, etc.
Leverage Common Frameworks	**Industry Standard SW Frameworks**	Caffe theano gatk
Fast-Track Your Performance	**Acceleration Libraries**	LZ4, Snappy, etc. zlib DPDK
Workload Optimization with Less Effort	**Intel Developer Tools** *(Intel Parallel Studio XE, Intel FPGA SDK for OpenCL™, Intel Quartus® Prime)*	Parallel Studio XE OpenCL Quartus Prime Design Suite
Common Developer Interface for Intel FPGA Data Center Products	**Acceleration Environment** *(Intel Acceleration Engine with OPAE Technology, FPGA Interface Manager (FIM))*	

intel XEON PLATINUM inside

Intel® Hardware

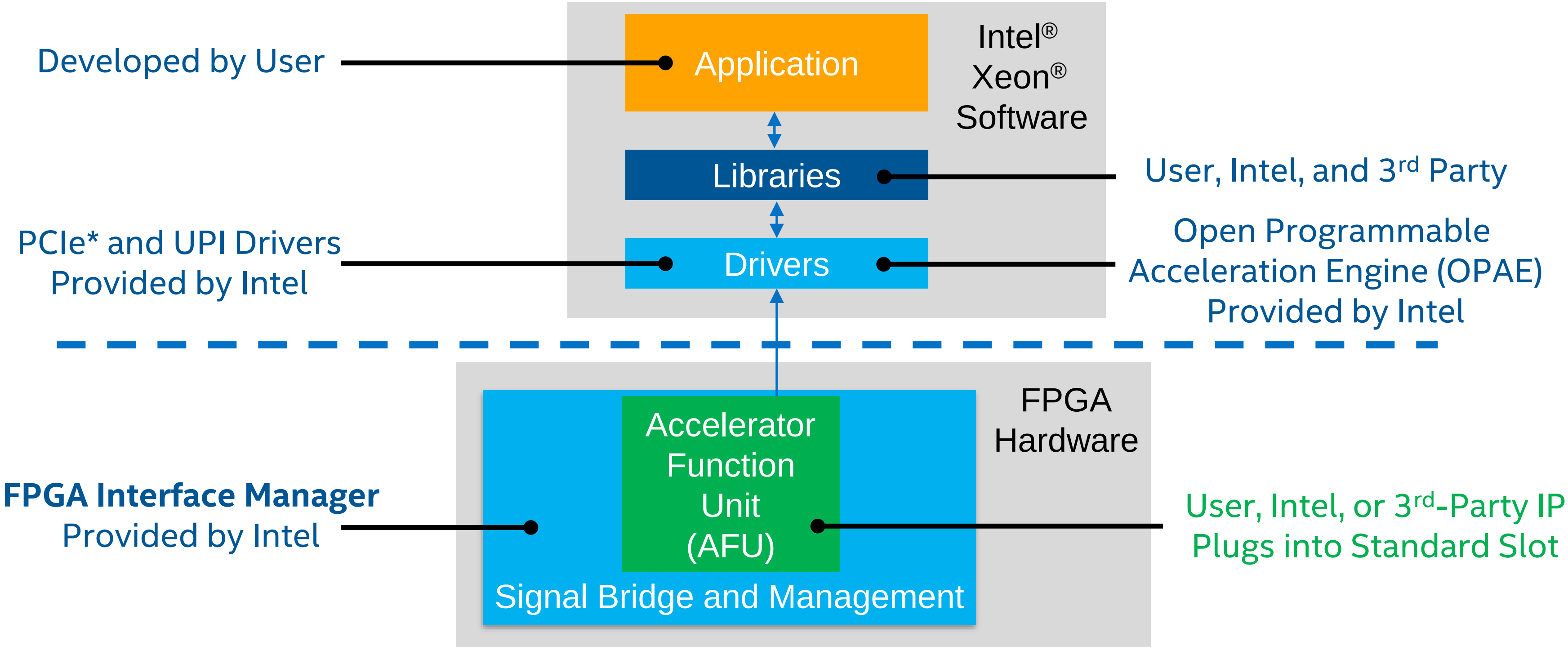
intel ARRIA inside

intel STRATIX inside

Intel® delivers a system-optimized solution stack for your data center workloads

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos
Some names pending final approval and may change in the future
Logos and names provided for illustrative purposes only. Current availability may be different.

Intel® Xeon® with FPGA **Virtualization Framework**

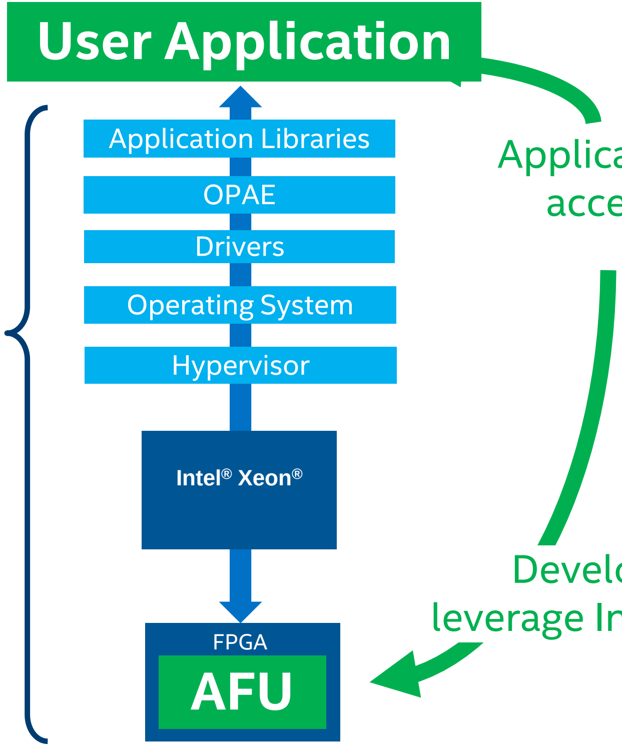

Developed by User
Application
Intel® Xeon® Software
Libraries
User, Intel, and 3rd Party
PCIe* and UPI Drivers Provided by Intel
Drivers
Open Programmable Acceleration Engine (OPAE) Provided by Intel
FPGA Hardware
Accelerator Function Unit (AFU)
FPGA Interface Manager Provided by Intel
User, Intel, or 3rd-Party IP Plugs into Standard Slot
Signal Bridge and Management

Simplifies the use of FPGAs in virtualized cloud environments

Interfacing with the Software Stack

Intel®-developed
virtualization framework

Qualified with ecosystem partners



User Application

Application Libraries

OPAE

Drivers

Operating System

Hypervisor

Intel® Xeon®

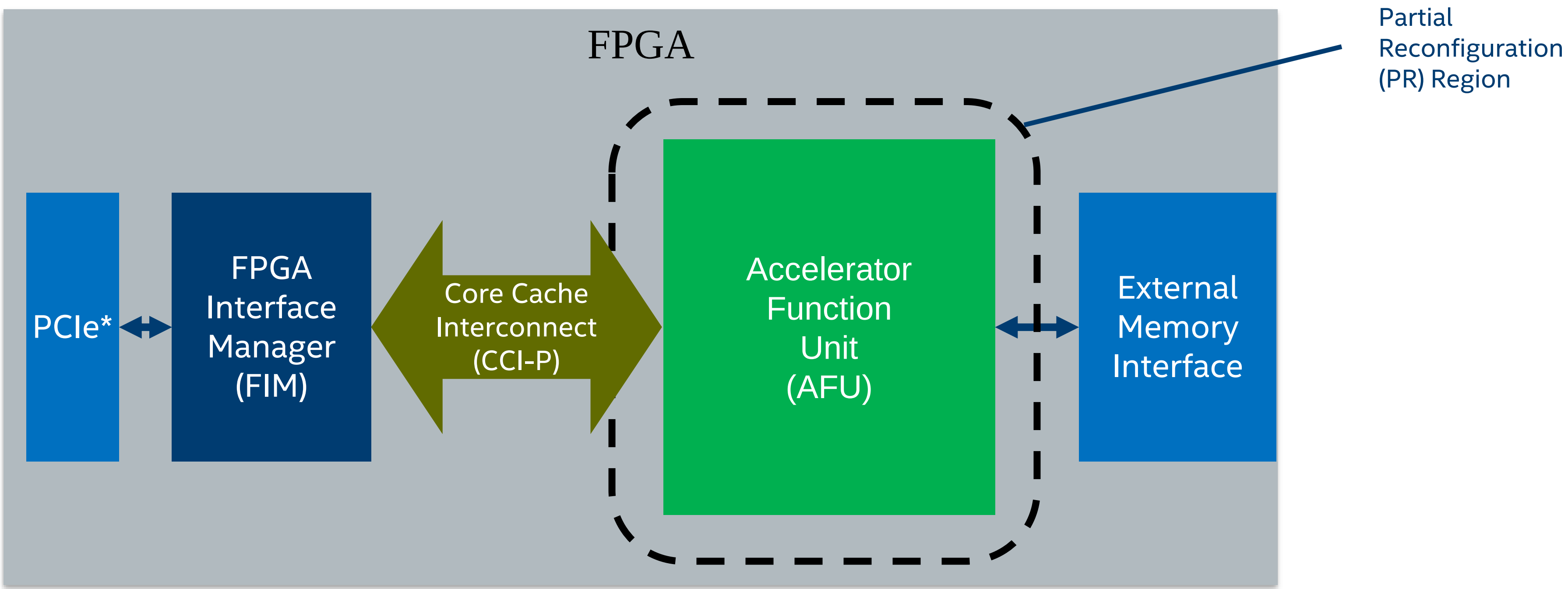
FPGA

AFU

Application developer focuses on
accelerating core workloads

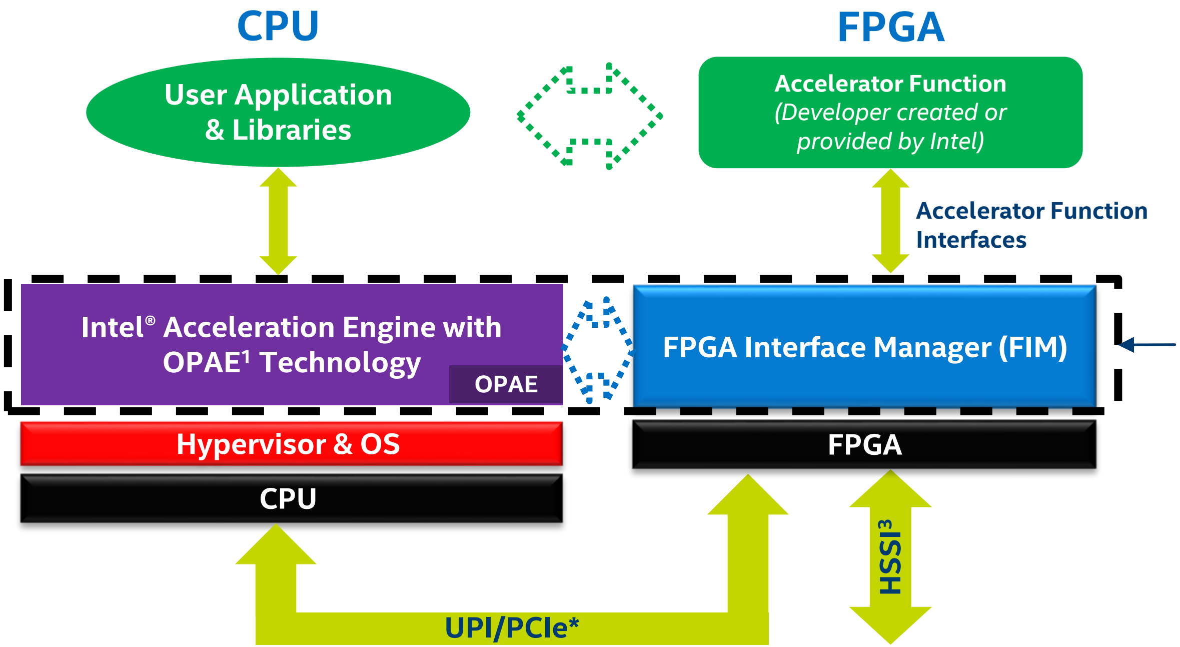
Develop own accelerators or
leverage Intel or 3rd party accelerators

FPGA Components


FPGA
PCIe*
FPGA Interface Manager (FIM)
Core Cache Interconnect (CCI-P)
Accelerator Function Unit (AFU)
External Memory Interface
Partial Reconfiguration (PR) Region

Acceleration Environment

Common Developer Interface for Intel® FPGA Data Center Products


CPU
FPGA
User Application & Libraries
Accelerator Function
(Developer created or provided by Intel)
Accelerator Function Interfaces
Intel® Acceleration Engine with OPAE¹ Technology
OPAE
FPGA Interface Manager (FIM)
Hypervisor & OS
FPGA
CPU
HSSI³
UPI/PCIe*

Optimized and simplified hardware and software APIs provided by Intel®

[1] OPAE = Open Programmable Acceleration Engine

[2] UPI = Intel® Ultra Path Interconnect

[3] HSSI = High Speed Serial Interface

Open Programmable Acceleration Engine (OPAE)

Consistent API across product generations and platforms

- Abstraction for hardware specific FPGA resource details

Designed for minimal software overhead and latency

- Lightweight user-space library *(libfpga)*

Open ecosystem for industry and developer community

- License: FPGA API (BSD), FPGA driver (GPLv2)

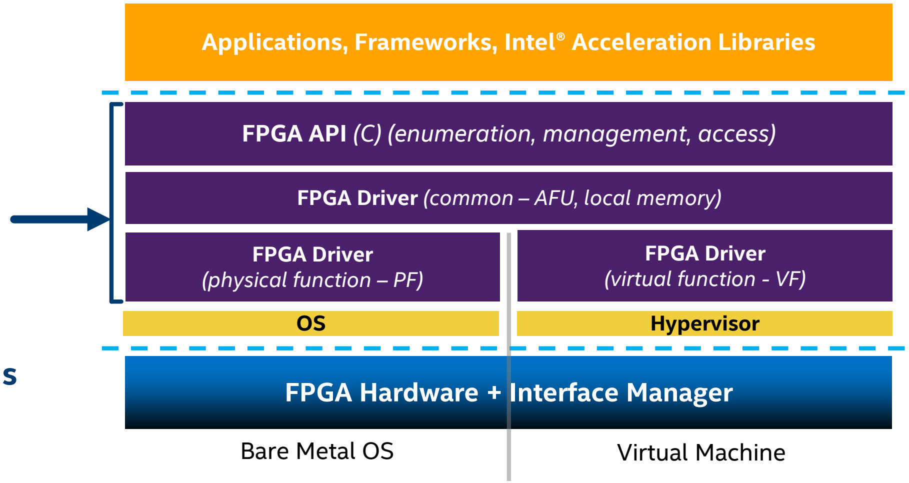
FPGA driver being upstreamed into Linux kernel

Supports both virtual machines and bare metal platforms

Faster development and debugging of Accelerator Functions with the included AFU Simulation Environment (ASE)

Includes guides, command-line utilities and sample code

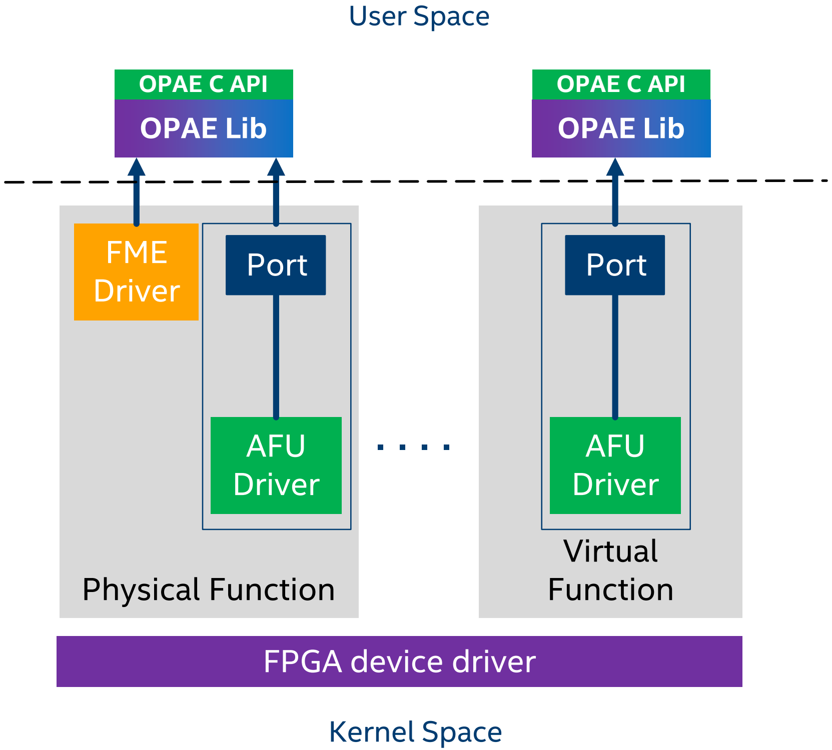
Simplified FPGA Programming Model for Application Developers



Applications, Frameworks, Intel® Acceleration Libraries
FPGA API (C) (enumeration, management, access)
FPGA Driver (common – AFU, local memory)
FPGA Driver (physical function – PF)
FPGA Driver (virtual function - VF)
OS
Hypervisor
FPGA Hardware + Interface Manager
Bare Metal OS
Virtual Machine

Start developing for Intel FPGAs with OPAE today: http://01.org/OPAE

FPGA Driver Architecture



FME: FPGA Management Engine Driver

- Static circuits for power/thermal management, reconfiguration, debugging, error reporting, performance counters, etc.

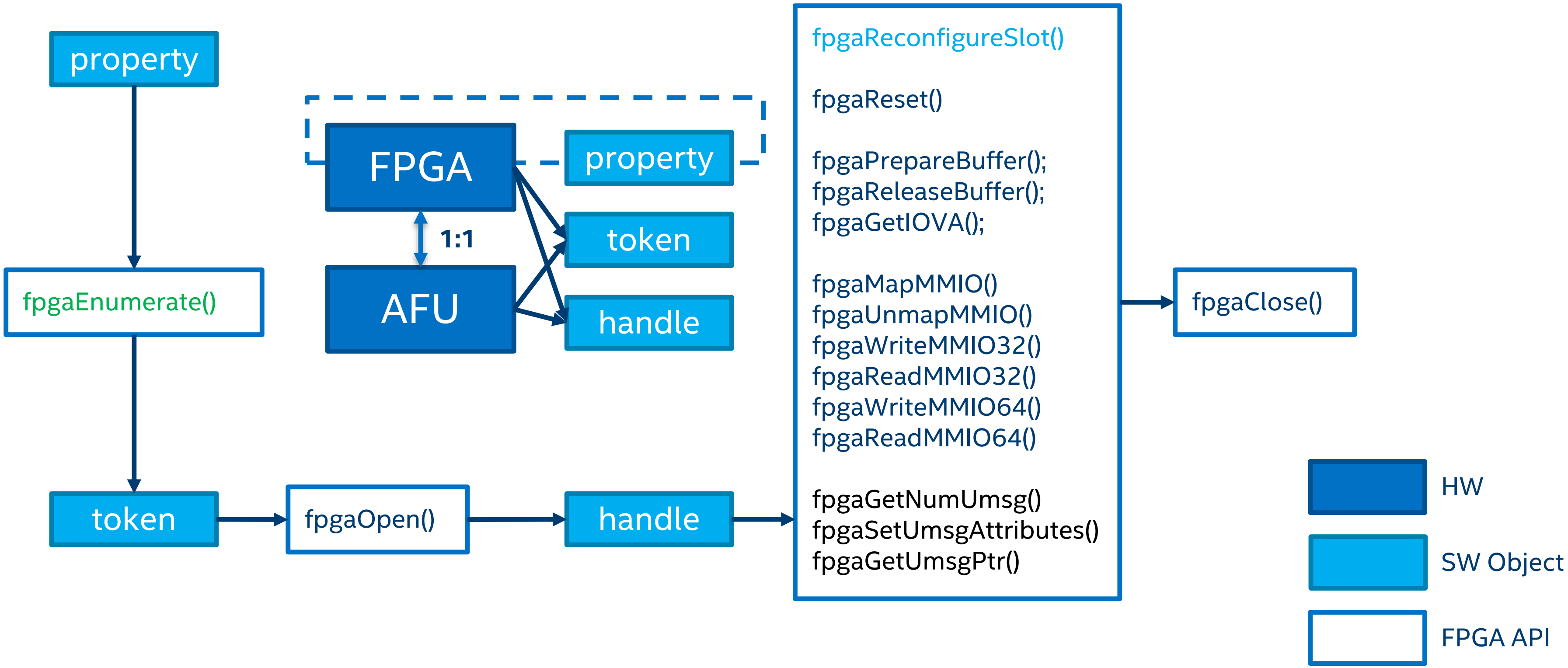
AFU: Accelerator Function Unit Driver

- Reconfigurable circuits for application specific functions.
- Exposes a 256KB region as control registers.
- User process can share memory buffers with AFU.

Port:

- Interface between the static and the reconfigurable regions
- Each port can attach an AFU. There may be multiple ports.
- A port can be assigned to a VM and expose the AFU.

OPAE FPGA API – Enumerate, Manage & Access



fpgaReconfigureSlot()

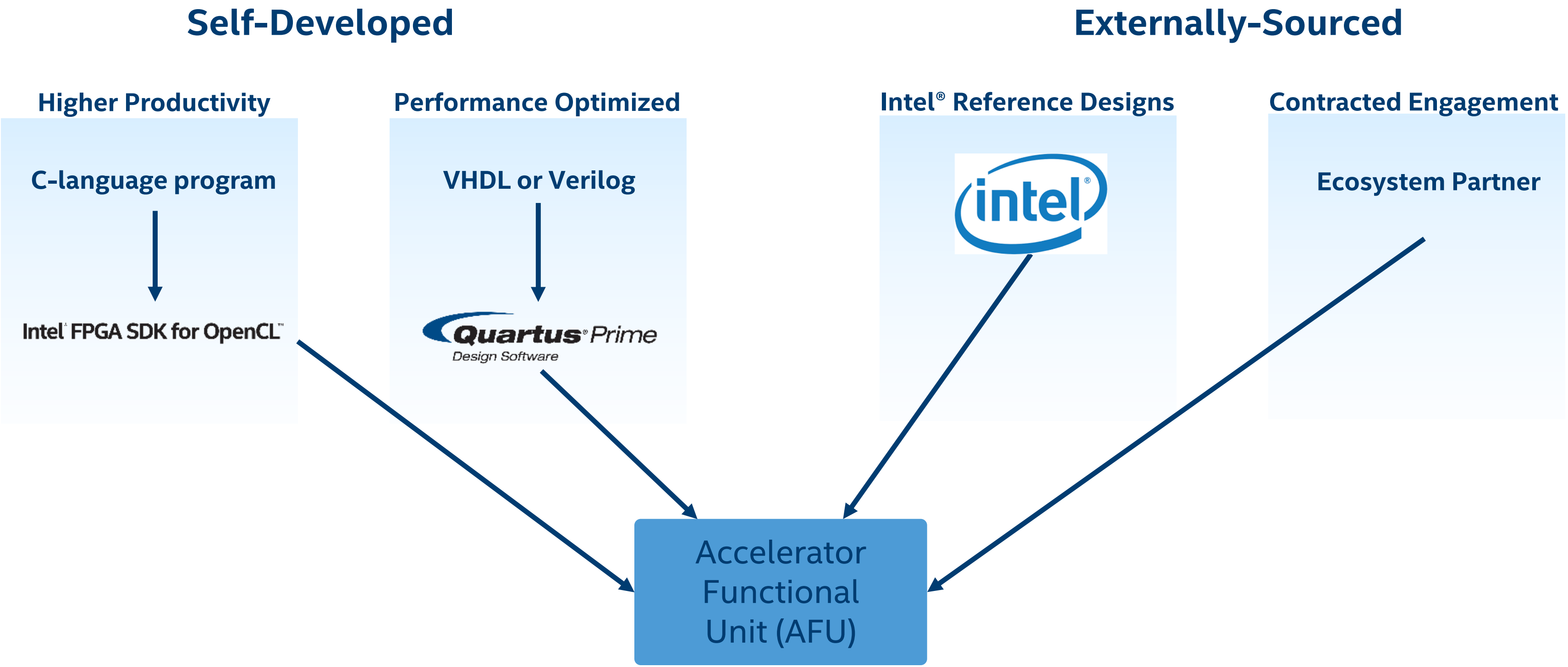
fpgaReset()

fpgaPrepareBuffer();
fpgaReleaseBuffer();
fpgaGetIOVA();

fpgaMapMMIO()
fpgaUnmapMMIO()
fpgaWriteMMIO32()
fpgaReadMMIO32()
fpgaWriteMMIO64()
fpgaReadMMIO64()

fpgaGetNumUmsg()
fpgaSetUmsgAttributes()
fpgaGetUmsgPtr()

How can FPGA accelerators be **created**?



Self-Developed

Higher Productivity

C-language program

Intel® FPGA SDK for OpenCL™

Performance Optimized

VHDL or Verilog

Quartus® Prime Design Software

Externally-Sourced

Intel® Reference Designs

intel®

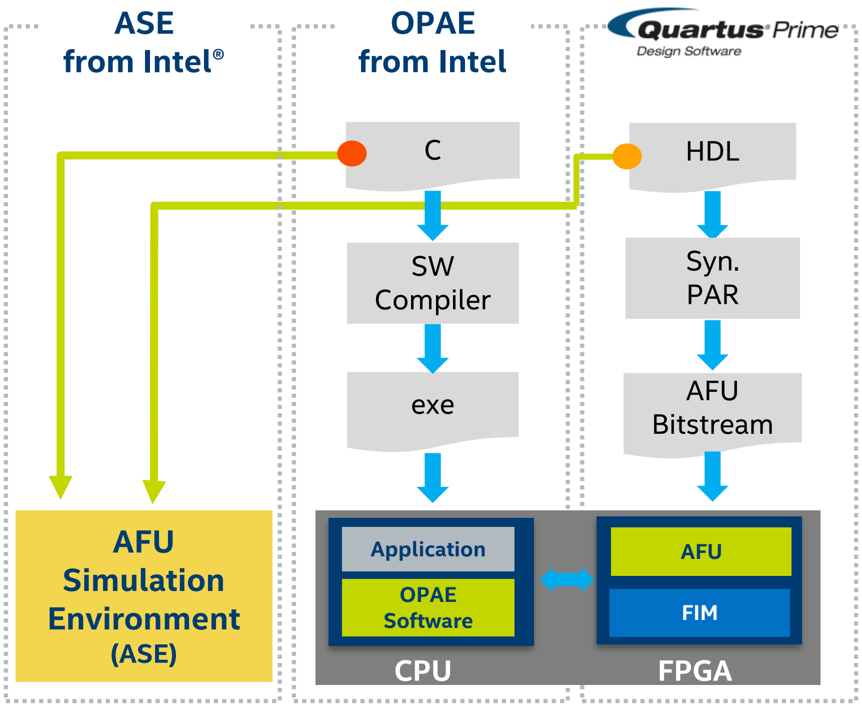
Contracted Engagement

Ecosystem Partner

Accelerator Functional Unit (AFU)

Two Development Approaches

HDL Programming



ASE
from Intel®

OPAE
from Intel

Quartus® Prime
Design Software

C

SW Compiler

exe

HDL

Syn. PAR

AFU Bitstream

AFU Simulation Environment (ASE)

Application

OPAE Software

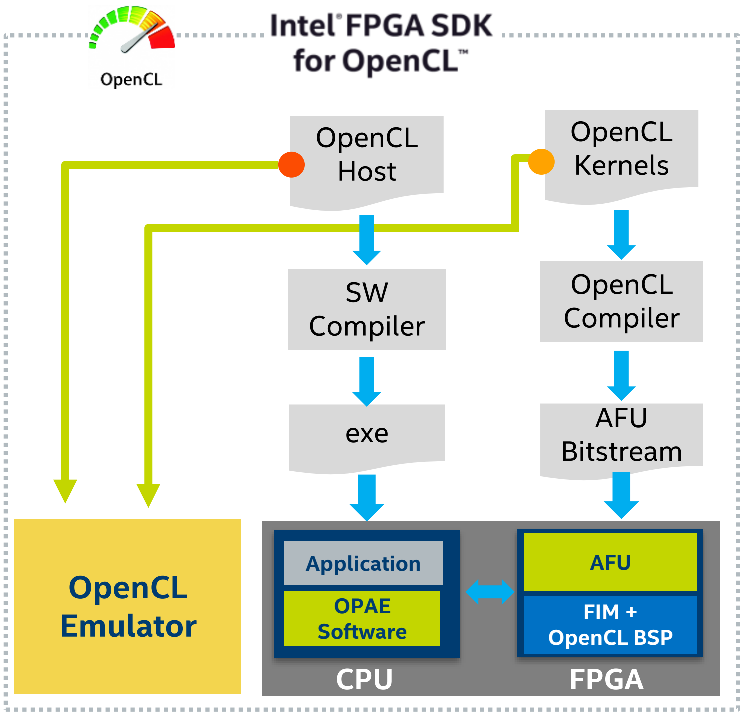
CPU

AFU

FIM

FPGA

OpenCL* Programming



OpenCL

Intel® FPGA SDK for OpenCL™

OpenCL Host

SW Compiler

exe

OpenCL Kernels

OpenCL Compiler

AFU Bitstream

OpenCL Emulator

Application

OPAE Software

CPU

AFU

FIM + OpenCL BSP

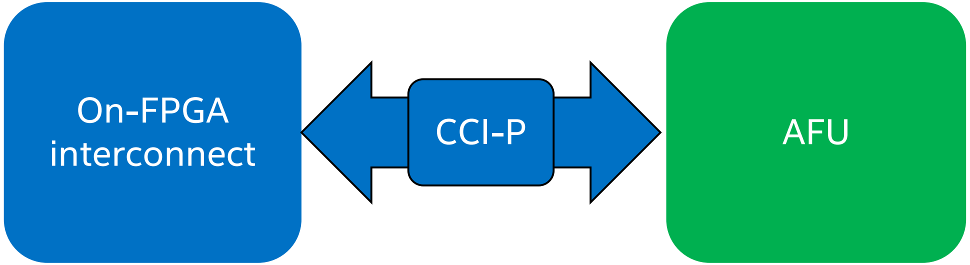
FPGA

OpenCL™ Flow

- Usage no different from traditional OpenCL™ flow
  - C based development and optimization flow to create AFUs and Host Application
  - Standard OpenCL FPGA application using the Intel® FPGA SDK for OpenCL
    - FPGA OpenCL debug and profiling tools supported
  - More information on using [OpenCL with FPGAs]()
- The Acceleration Stack abstracted away from user
  - OPAE part of the Host Run-Time
    - Host does not need to interact with OPAE SW directly
  - OpenCL BSP part of the FPGA Interface Manager
    - Kernel Avalon Interface translated to CCI-P by the BSP

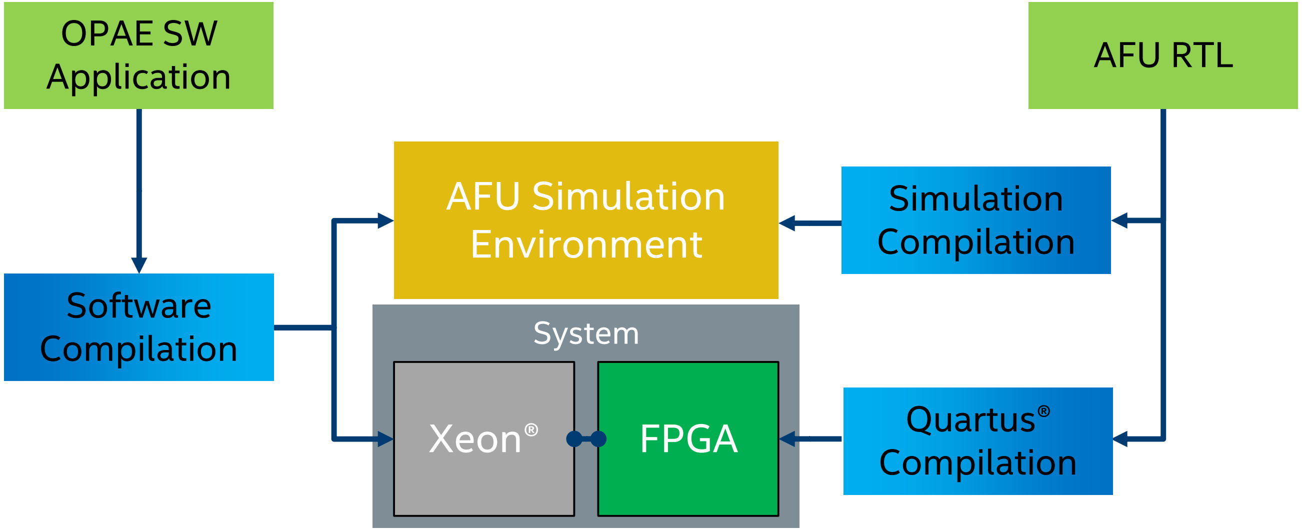
To learn more about using OpenCL with FPGAs, visit [Intel FPGA Customer Training]() page

RTL AFU



- Develop RTL AFU with standard FPGA development tools
- Interface with the acceleration stack through Core Cache Interconnect (CCI-P)
  - Provides a base platform memory interface
    - Simple request/response interface (Simple Read/Write)
    - Physical addresses
    - No order guarantees
  - These minimal requirements satisfy major classes of algorithms, e.g.:
    - Double buffered kernels that read from and write to different buffers
    - Streaming kernels that read from one memory-mapped FIFO and write to another

RTL Flow



- AFU Simulation Environment (ASE) enables seamless portability to real HW
  - Allows fast verification of OPAE software together with AFU RTL without HW
    - SW Application loads ASE library and connects to RTL simulation
  - For execution on HW, application loads Runtime library and RTL is compiled by Intel® Quartus into FPGA bitstream

Intel® Programmable Acceleration Card with Intel Arria® 10 GX FPGA

Intel's 1st versatile FPGA PCIe acceleration card that offers inline & look-aside acceleration for workloads requiring up to 45W

1st acceleration card to offer the Acceleration Stack for Intel Xeon CPU with FPGAs enabling broader FPGA adoption in data center

[Intel Programmable Acceleration Card with Intel Arria 10 GX FPGA]()

Summary

- Acceleration Stack for Intel® Xeon® CPU with FPGAs
  - Robust collection of software, firmware, and tools
  - Makes it easy to develop and deploy Intel FPGAs in the data center
  - Supports both RTL and OpenCL™ development flows
  - Intel FPGA Acceleration Hub
- Follow-on trainings
  - RTL Development and Acceleration with the Acceleration Stack for Intel® Xeon® CPU and FPGAs
  - Intel FPGA OpenCL Trainings
- References
  - Various quick start and development guides associated with the Acceleration Stack