AMD PENSANDO™ SOFTWARE-IN-SILICON DEVELOPMENT KIT (SSDK)

November 2023

The AMD Pensando™ Software-in-Silicon Development Kit (SSDK) enables the development of software for AMD Pensando data processing units (DPUs). The SSDK provides a complete container-based development environment on x86 systems. The software developed can run on a physical DPU, a DPU simulator, and an x86 host.

The SSDK allows for the development of data plane, management plane, and control plane functions, including DPU fast path, DPU slow path, security offloads, PCIe® emulation, and CPU complex applications.

Key features include:

- Simplified setup that quickly installs the development environment.
- Extensive examples and reference pipelines to simplify getting started with P4 development.
- Readily available containerized infrastructure to build, test, and debug code, before cross-compiling the same code to the DPU.
- Compiled and ready-to-integrate system software capable of supporting secure boot for DPU and reference pipelines.
- Extensive documentation that helps navigate the development environment.

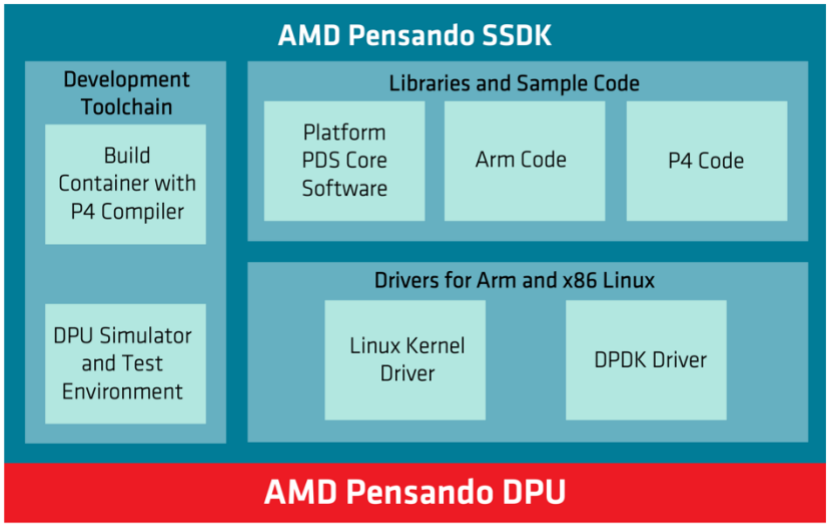
The SSDK includes:

- Linaro Arm® cross-compile toolchain
- $P4_{16}$ compiler
- DPU simulator
- Debugging tools and libraries
- DPDK driver
- P4 compiler auto-generated P4 table management CRUD APIs
- Comprehensive documentation
- Reference pipelines

A rich set of reference pipelines provides working sample code to demonstrate the performance, security, and stateful and stateless services supported in the P4 engine, including support for session, flow, routing, and security policy at scale.

AMD PENSANDO SOFTWARE-IN-SILICON DEVELOPMENT KIT (SSDK)

- Development toolchain for DPU and simulator
- Compiler
- Libraries and code in P4$_{16}$, C, and C++
- Drivers for Arm and x86 systems (both Linux kernel and DPDK)
- Rich set of reference pipelines for easy development
- Documentation
- Debug tools and logs



Overview

The Software-in-Silicon Development Kit (SSDK) is an environment that enables the development of software for AMD Pensando DPUs.

Code can be written in P4$_{16}$ to execute in a DPU's fast path match-processing units (MPUs) and can be written in C and C++ for its Arm core complex. The DPU's built-in function accelerators can also be leveraged.

Instead of developing your own pipeline from scratch, you can use the SSDK for programmatic control of the DPU with a production-ready and customizable pipeline and P4 libraries via the runtime gRPC API.

Simulator Characteristics

The simulator can validate, accelerate development, and facilitate debugging issues in a virtualized environment, without uploading the image onto the actual hardware. The ability to validate code can be useful when integrating the SSDK and simulator into CI/CD-based development and workflows.

The simulator is machine-register accurate, allowing any code built for the simulator to be cross-compiled to run on the actual hardware.[1]

AMD Pensando DPU Unique Capabilities

The SSDK leverages various AMD Pensando DPU capabilities and allows deployment with minimal dependencies on host or DPU x86/Arm resources. Many DPU features can be developed in $P4_{16}$ to execute in the fast path, which provides greater scale, CPS and PPS, low latency, and jitter. This also allows for multiple service functions to execute concurrently on the same DPU.

Capability	Enables
Shared memory across the DPU (Up to 64 GB shared memory on the DSC card)	• With local caching for each stage, any sized table can be managed • No need to break up the number of stages based on table size
All memory is read/write capable for both fast path (P4) and slow path (C and C++)	• Enables development of stateful services, new flows, or control/management requirements • Networking, security, storage offloads, and any infrastructure or I/O offload can be built into fast path
Flow High Availability	• Support for flow high availability (HA) - Showcases Flow HA (sync-up) in the data plane across DPU peers. • To provide high availability against device failures, each DPU has another DPU paired with it. Together, the pair provides active/standby HA for all flows. The active/standby role is per-flow, i.e., when both DPUs are functional, DPU1 will be 'active' for some flows while DPU2 will be 'active' for others. When a device fails, the surviving device becomes active for all flows.
Graceful Upgrade	• Support for DPU software upgrade without resetting the DPU or its PCIe interface.

[1] Execution timing will vary between simulator and actual hardware.

Multiple Services in Parallel without Degradation	• SDN • Security (IPsec, Firewall, and ACL) • Storage Offload • Encrypt/Decrypt • NAT • Encap/Decap • SLB

144 Programmable MPU Stages	• Shared memory with local caching for all stages
	  ○ All pipeline tables (SRAM and TCAM) are accessible in all stages, and any tables placed in HBM/DRAM are accessible to all pipelines
	• Large or small tables with no impact on memory allocation
	• Entire memory is open for read and write access
	  ○ No requirement for register memory per stage
	• Each MPU stage runs to completion
	  ○ No need to split stages for complex programs
	• 5 Programmable pipelines
	  ○ Ingress
	  ○ Egress
	  ○ RxDMA
	  ○ TxDMA
	  ○ SxDMA
	• Stateful and stateless processing at scale
	  ○ No requirement to offload to Arm processor for stateful service delivery

Reference Pipeline Examples

The SSDK reference pipelines are short but fully functional pipeline examples that showcase table management, Ingress, and Egress packet flows and contain P4, C, and C++ code samples for Arm interactions and packet manipulation for pipeline processing. The reference pipelines with the generated P4 PDS (Pensando Distributed Services) table APIs and the libraries and toolchain for development and debugging. These examples can be used for prototyping and as starting points for developing new pipelines and simplifying adding custom code specific to the developer's application needs.

Hello World

Basic P4 pipeline to introduce the SSDK. Demonstrates maximum performance (PPS - packets per second): packets are received and sent out, showcasing NACL redirect.

SDN Policy Offload

Showcases a ready-to-deploy pipeline providing a P4 library to accelerate LPM, ACL, and flow/session aging in P4 MPU. A large-scale LPM and ACL leveraging DDR. Ready-to-deploy code to integrate your P4 code with the AMD-provided P4 library. Implements flow table based forwarding along with security policy, IPv4/IPv6 route table configuration from DP_App, and lookup by P4+ programs. The pipeline uses all five pipeline modules in the data path (P4I, P4E, P4+RXDMA, P4+TXDMA and SXDMA).

- Flow offload with a 64 million connection tracking in P4.
- LPM priority-based route (1M) lookup in P4+ with opaque result (for rewrites including NAT, VXLAN, and VLAN).
- Route lookups in the P4+ data path.
    - Route APIs that support insert and delete operations.
- Hardware aging (idle and connection track) in P4+.
- IPv4/IPv6 support for data path (NSG and routing).
- Policy-based security action in P4+ based on 5 tuples.
- Policy lookups in the P4+ data path.
    - Security Policy APIs that support insert and delete operations.

SAI

Enables integrating an SSDK-based pipeline into a SONiC environment running on a DPU. Includes SAI interface for traditional routing/switching constructs.

The SAI reference pipeline implements all the base SAI functionality needed on a DPU, including:

- SAI Switch creation and initialization
- Interface management
- Host I/F and Traps support
- Underlay routing

- Implements the LibSAI interface, and can generate a `Libsai.so` shared library to be linked into SONiC. It also provides a reference P4 pipeline for TCAM-based underlay routing in P4. SRAM-based policer in P4.

Flow Offload

A reference pipeline for flow offload, including DDR-based bandwidth policers (one million) in P4. Showcases large-scale tables with flow lookup. Highlights the performance and capability of using large scale tables, leveraging the stage cache and DDR memory for table lookups, as well as demonstrating how large tables can scale in terms of flow entries. A single pipeline using P4I only. Also includes session statistics.

IPsec GW

Showcases IPsec transport and tunnel model implementation in a P4 pipeline between de-parser and packet buffer (PB). Both encryption and decryption take place inline without the need to offload to a crypto engine, providing the best performance of IPsec in the data plane.

Classic RTR

Showcases large-scale LPM (one million routes) stateless forwarding in P4, including IPv4/IPv6 support, inline IPsec support (transport/tunnel mode) based on LPM result, IPv4 fragmentation, and reassembly in P4/P4+ to achieve high throughput when fragmentation/reassembly is needed with IPsec or other tunnel encapsulation.

Classic Host Offload

Showcases classic NIC forwarding performance. Verifies the DPDK Tx/Rx into/from Arm core or host CPU core. Includes TSO, partial checksum, complete checksum offloads, and receive side scaling.

Access the SSDK

To sign up for the SSDK, please visit amd.com/pensando#SSDK.

Hardware and Support

For information about ordering Distributed Services Cards based on the AMD Pensando 2nd generation ("Elba") DPU, and the appropriate support for your deployment, please contact your AMD sales representative, referencing the Distributed Services Card part number listed below.

Part Number / SKU	Description
DSC2-2Q200-32R32F64P-S	DSC2-200 (Card) - 2 x 200Gbps (QFP56) FHHL

In addition to the PCIe card-based form factor, DPU-only deployment options are also available; check with your AMD sales representative for further information.

Additional Resources

- SSDK documentation
  - User guides for four sample reference pipelines, detailing the P4 program code, and including a set of ready-to-deploy precompiled libraries and other ready-to-use code
- AMD Pensando 2nd Generation ("Elba") DPU Product Brief
- AMD Pensando DSC2-200 Product Brief

AMD
together we advance_networking

Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

amd.com/pensando

PPB22002