IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION

LUCIO DEVELOPMENT LLC,	§	
	§	
Plaintiff,	§	Case No:
	§	
vs.	§	PATENT CASE
	§	
XILINX, INC.,	§	
	§	
Defendant.	§	
	§	

COMPLAINT

Plaintiff Lucio Development LLC ("Plaintiff" or "Lucio") files this Complaint against Xilinx, Inc. ("Defendant" or "Xilinx") for infringement of United States Patent No. 7,069,546 (hereinafter "the '546 Patent").

PARTIES AND JURISDICTION

1. This is an action for patent infringement under Title 35 of the United States Code. Plaintiff is seeking injunctive relief as well as damages.

2. Jurisdiction is proper in this Court pursuant to 28 U.S.C. §§ 1331 (Federal Question) and 1338(a) (Patents) because this is a civil action for patent infringement arising under the United States patent statutes.

3. Plaintiff is a Texas limited liability company with its office address at 555 Republic Dr., Suite 200, Plano, Texas 75074.

4. On information and belief, Defendant is a Delaware corporation with its principal place of business located at 5801 Tennyson Parkway, Suite 460, Plano, Texas 75024. Xilinx can be served via its registered agent for service of process, CT Corporation System,

1999 Bryan Street, Suite 900, Dallas, Texas 75201.

5. This Court has personal jurisdiction over Defendant because Defendant has committed, and continues to commit, acts of infringement in this District, has conducted business in this District, and/or has engaged in continuous and systematic activities in this District.

6. On information and belief, Defendant's instrumentalities that are alleged herein to infringe were and continue to be used, imported, offered for sale, and/or sold in this District.

VENUE

7. Venue is proper in this District pursuant to 28 U.S.C. §1400(b) because acts of infringement are occurring in this District and Defendant has a regular and established place of business in this District. For instance, on information and belief, Defendant has a regular and established place of business at 5801 Tennyson Parkway, Suite 460, Plano, Texas 75024.

COUNT I
(INFRINGEMENT OF UNITED STATES PATENT NO. 7,069,546)

8. Plaintiff incorporates paragraphs 1 through 7 herein by reference.

9. This cause of action arises under the patent laws of the United States and, in particular, under 35 U.S.C. §§ 271, *et seq*.

10. Plaintiff is the owner by assignment of the '546 Patent with sole rights to enforce the '546 Patent and sue infringers.

11. A copy of the '546 Patent, titled "Generic Framework for Embedded Software Development," is attached hereto as Exhibit A.

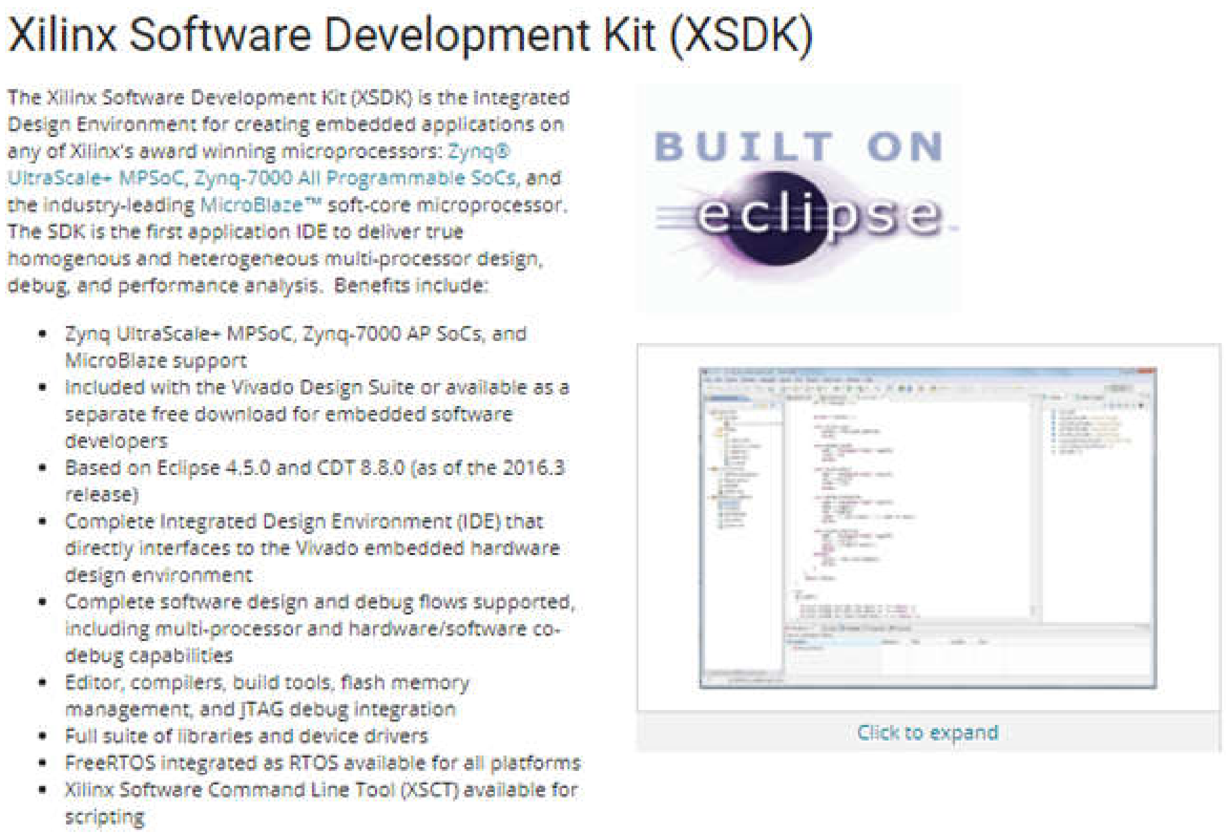
12. The '546 Patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code.

13. On information and belief, Defendant has infringed and continues to infringe

one or more claims, including at least Claim 1, of the ‘546 Patent by making, using, importing, selling, and/or offering for sale a software platform for embedded software development, which is covered by at least Claim 1 of the ‘546 Patent. Defendant has infringed and continues to infringe the ‘546 Patent directly in violation of 35 U.S.C. § 271.

14. Defendant, sells, offers to sell, and/or uses embedded software development packages including, without limitation, the Xilinx Software Development Kit (XSDK), the SDSoC Development Environment, and any similar products (“Product”), which infringe at least Claim 1 of the ‘546 Patent.

15. The Product is a framework that is configured to create embedded software for multiple hardware modules (e.g., versions of a microprocessor, such as the Zyng and/or MicroBlaze). Xilinx and/or its customers use the Product to produce embedded software. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.



Xilinx Software Development Kit (XSDK)

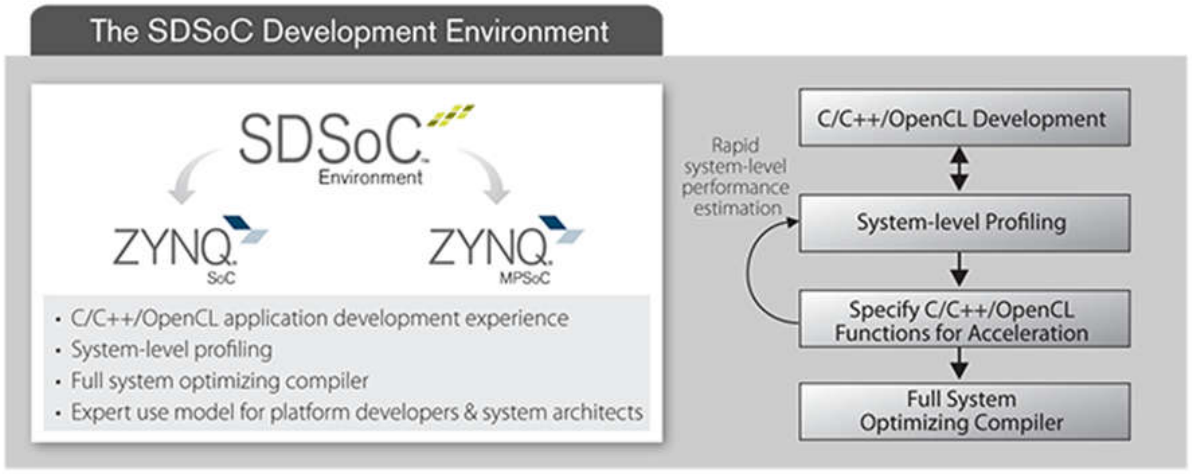
The Xilinx Software Development Kit (XSDK) is the integrated Design Environment for creating embedded applications on any of Xilinx's award winning microprocessors: Zynq® UltraScale+ MPSoC, Zynq-7000 All Programmable SoCs, and the industry-leading MicroBlaze™ soft-core microprocessor. The SDK is the first application IDE to deliver true homogenous and heterogeneous multi-processor design, debug, and performance analysis. Benefits include:

- Zynq UltraScale+ MPSoC, Zynq-7000 AP SoCs, and MicroBlaze support
- Included with the Vivado Design Suite or available as a separate free download for embedded software developers
- Based on Eclipse 4.5.0 and CDT 8.8.0 (as of the 2016.3 release)
- Complete Integrated Design Environment (IDE) that directly interfaces to the Vivado embedded hardware design environment
- Complete software design and debug flows supported, including multi-processor and hardware/software co-debug capabilities
- Editor, compilers, build tools, flash memory management, and JTAG debug integration
- Full suite of libraries and device drivers
- FreeRTOS integrated as RTOS available for all platforms
- Xilinx Software Command Line Tool (XSCT) available for scripting

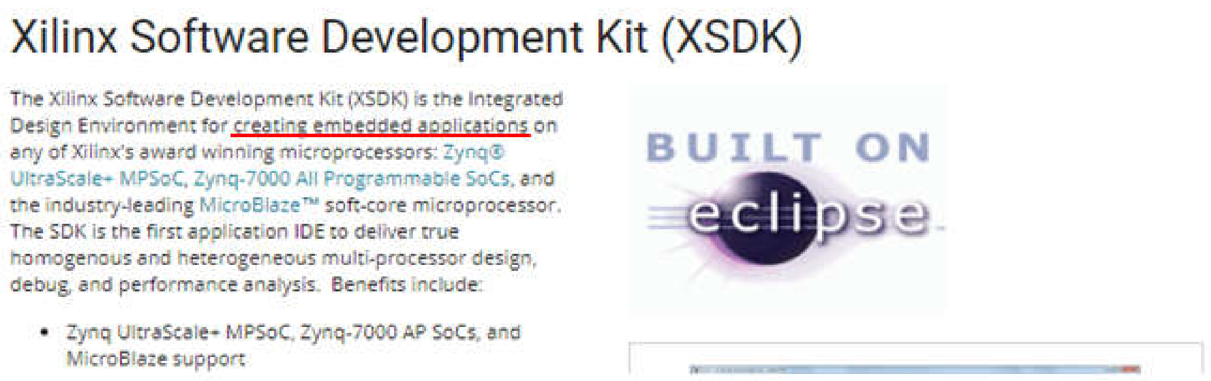
BUILT ON eclipse

Click to expand

Source: https://www.xilinx.com/products/design-tools/embedded-software/sdk.html


The SDSoC Development Environment
SDSoC Environment
ZYNQ SoC
ZYNQ MPSoC
• C/C++/OpenCL application development experience
• System-level profiling
• Full system optimizing compiler
• Expert use model for platform developers & system architects
Rapid system-level performance estimation
C/C++/OpenCL Development
System-level Profiling
Specify C/C++/OpenCL Functions for Acceleration
Full System Optimizing Compiler

Source: https://www.xilinx.com/products/design-tools/software-zone/sdsoc.html#overview

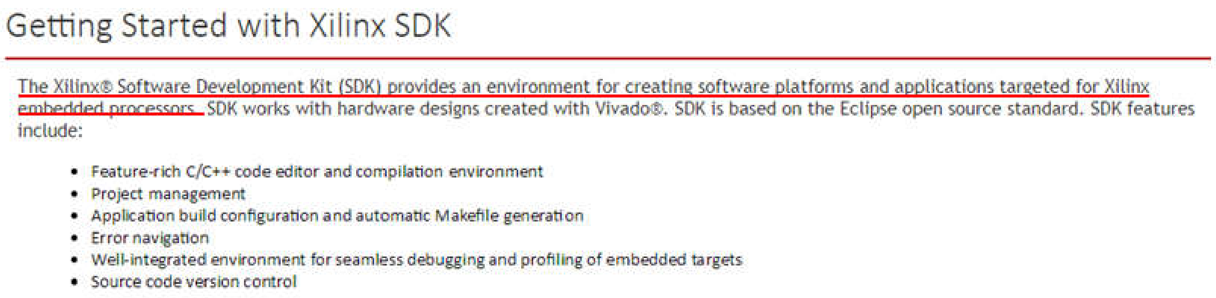


Xilinx Software Development Kit (XSDK)

The Xilinx Software Development Kit (XSDK) is the Integrated Design Environment for creating embedded applications on any of Xilinx's award winning microprocessors: Zynq® UltraScale+ MPSoC, Zynq-7000 All Programmable SoCs, and the industry-leading MicroBlaze™ soft-core microprocessor. The SDK is the first application IDE to deliver true homogenous and heterogeneous multi-processor design, debug, and performance analysis. Benefits include:

- Zynq UltraScale+ MPSoC, Zynq-7000 AP SoCs, and MicroBlaze support

Source: https://www.xilinx.com/products/design-tools/embedded-software/sdk.html

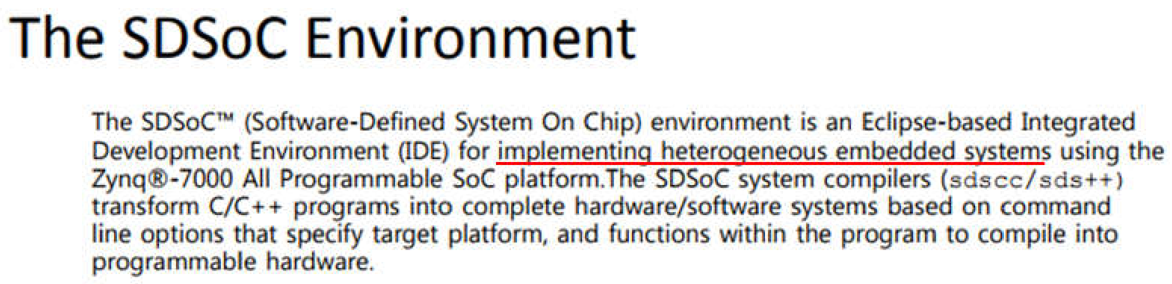


Getting Started with Xilinx SDK

The Xilinx® Software Development Kit (SDK) provides an environment for creating software platforms and applications targeted for Xilinx embedded processors. SDK works with hardware designs created with Vivado®. SDK is based on the Eclipse open source standard. SDK features include:

- Feature-rich C/C++ code editor and compilation environment
- Project management
- Application build configuration and automatic Makefile generation
- Error navigation
- Well-integrated environment for seamless debugging and profiling of embedded targets
- Source code version control

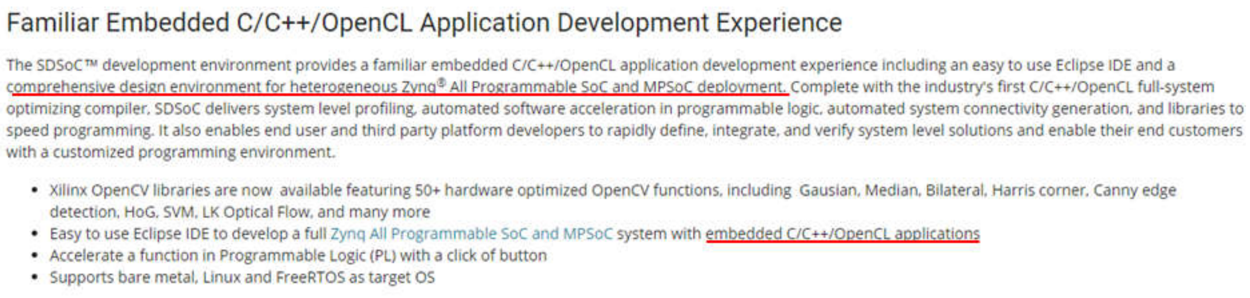
Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_1/SDK_Doc/index.html



The SDSoC Environment

The SDSoC™ (Software-Defined System On Chip) environment is an Eclipse-based Integrated Development Environment (IDE) for implementing heterogeneous embedded systems using the Zynq®-7000 All Programmable SoC platform.The SDSoC system compilers (sdscc/sds++) transform C/C++ programs into complete hardware/software systems based on command line options that specify target platform, and functions within the program to compile into programmable hardware.

Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_2/ug1027-intro-to-sdsoc.pdf



Familiar Embedded C/C++/OpenCL Application Development Experience

The SDSoC™ development environment provides a familiar embedded C/C++/OpenCL application development experience including an easy to use Eclipse IDE and a comprehensive design environment for heterogeneous Zynq® All Programmable SoC and MPSoC deployment. Complete with the industry's first C/C++/OpenCL full-system optimizing compiler, SDSoC delivers system level profiling, automated software acceleration in programmable logic, automated system connectivity generation, and libraries to speed programming. It also enables end user and third party platform developers to rapidly define, integrate, and verify system level solutions and enable their end customers with a customized programming environment.

- Xilinx OpenCV libraries are now available featuring 50+ hardware optimized OpenCV functions, including Gausian, Median, Bilateral, Harris corner, Canny edge detection, HoG, SVM, LK Optical Flow, and many more
- Easy to use Eclipse IDE to develop a full Zynq All Programmable SoC and MPSoC system with embedded C/C++/OpenCL applications
- Accelerate a function in Programmable Logic (PL) with a click of button
- Supports bare metal, Linux and FreeRTOS as target OS

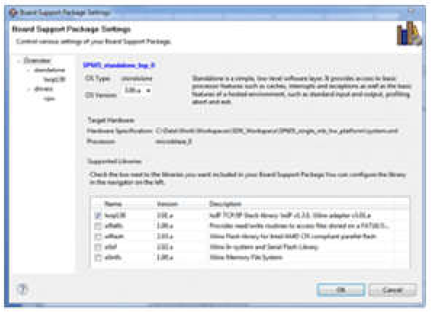
Source: https://www.xilinx.com/products/design-tools/software-zone/sdsoc.html#overview

16. The Product includes a generic application handler program including drivers, libraries, Board Support Packages (BSP) and a Hardware Abstraction Layer (HAL) that

provides multiple generic application programming interfaces (APIs). The generic code provides common and generic functions to multiple hardware modules (e.g., versions of a microprocessor, such as Zynq and/or MicroBlaze). For example, XSDK and SDSoC both provide a hardware abstraction layer (HAL) with common functions related register IO, exception and cache – which are common and uniform across all supported MicroBlaze and Cortex A-9 processors. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

Custom Design Aware

Xilinx SDK understands the custom embedded hardware design that has been defined in the Vivado Design Suite. Based on this design, several key parameters are auto-configured, including memory maps, peripheral register settings, tools and library paths, compiler options, JTAG and flash memory settings, debugger connections, and Linux and bare-metal Board Support Packages (BSPs). This custom design-aware pre-configuration, combined with the auto-generation of critical system software, ensures that software development can progress rapidly with a minimal learning curve.



Board Support Package Settings

Control various settings of your Board Support Package.

OS Type: standalone

Target Hardware

Supported Libraries

Name	Version	Description
lwip130	[illegible]	lwIP TCP/IP Stack library [illegible]
xilffs	[illegible]	Provides read/write routines to access files stored on a FAT16/3...
xilflash	[illegible]	Xilinx Flash library [illegible]
xilisf	[illegible]	Xilinx In-system and Serial Flash Library
xilmfs	[illegible]	Xilinx Memory File System

Drivers and Libraries

XSDK includes user-customizable drivers for all supported Xilinx hardware IPs, POSIX compliant kernel library and networking and file handling libraries. These libraries and drivers can scale for the custom-design based on feature needs, memory requirements and hardware capabilities.

Software Profiling

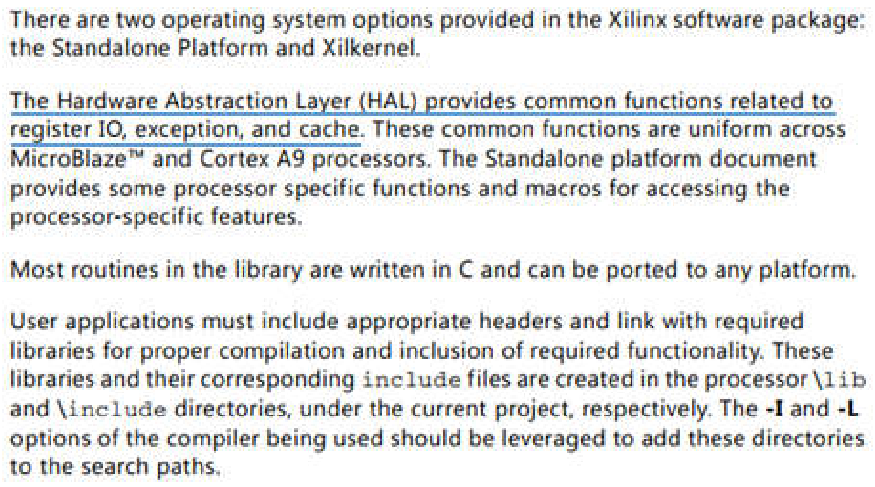
XSDK includes profiling tools that help to identify bottle necks in your code occurring due to the interaction of functions that are executed within the programmable logic, and on the processor. Supports hierarchical profiling - allowing the user to view which called functions, or which calling functions are affecting processor performance the most.

Source: https://www.xilinx.com/products/design-tools/embedded-software/sdk.html

Table 1-1: **OS and Libraries Document Collection Contents**

Document Name	Summary
LibXil Standard C Libraries	Describes the software libraries available for the embedded processors.
Standalone (v.5.2)	Describes the Standalone platform, a single-threaded, simple operating system (OS) platform that provides the lowest layer of software modules used to access processor-specific functions. Some typical functions offered by the Standalone platform include setting up the interrupts and exceptions systems, configuring caches, and other hardware specific functions. The Hardware Abstraction Layer (HAL) is described in this document.
Xilkernel (v6.2)	Describes the Xilkernel, a simple embedded processor kernel that can be customized to a large degree for a given system. Xilkernel has the key features of an embedded kernel such as multi-tasking, priority-driven preemptive scheduling, inter-process communication, synchronization facilities, and interrupt handling. Xilkernel is small, modular, user-customizable, and can be used in different system configurations. Applications link statically with the kernel to form a single executable.

Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_3/oslib_rm.pdf

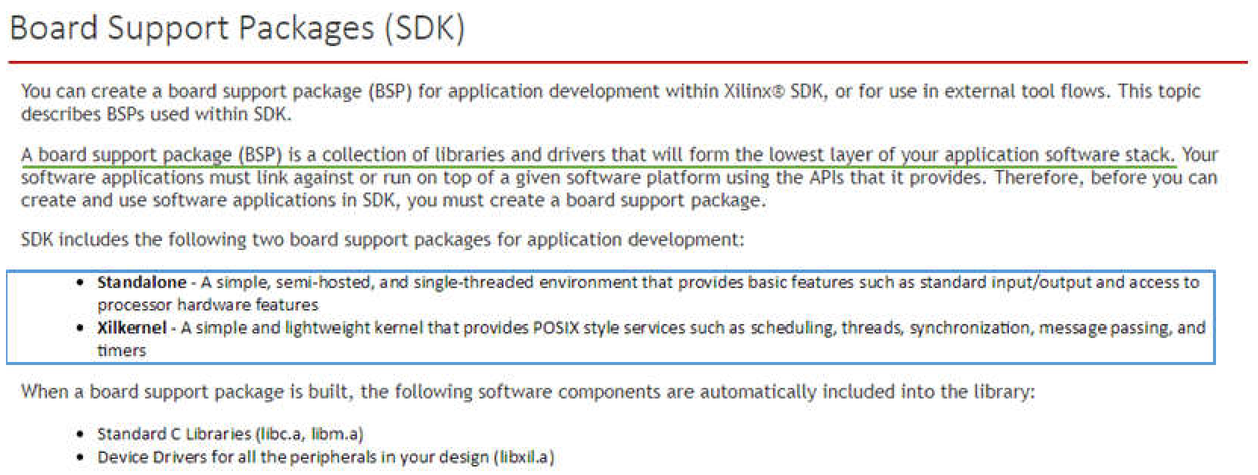

There are two operating system options provided in the Xilinx software package: the Standalone Platform and Xilkernel.

The Hardware Abstraction Layer (HAL) provides common functions related to register IO, exception, and cache. These common functions are uniform across MicroBlaze™ and Cortex A9 processors. The Standalone platform document provides some processor specific functions and macros for accessing the processor-specific features.

Most routines in the library are written in C and can be ported to any platform.

User applications must include appropriate headers and link with required libraries for proper compilation and inclusion of required functionality. These libraries and their corresponding `include` files are created in the processor `\lib` and `\include` directories, under the current project, respectively. The **-I** and **-L** options of the compiler being used should be leveraged to add these directories to the search paths.

Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_3/oslib_rm.pdf


Board Support Packages (SDK)

You can create a board support package (BSP) for application development within Xilinx® SDK, or for use in external tool flows. This topic describes BSPs used within SDK.

A board support package (BSP) is a collection of libraries and drivers that will form the lowest layer of your application software stack. Your software applications must link against or run on top of a given software platform using the APIs that it provides. Therefore, before you can create and use software applications in SDK, you must create a board support package.

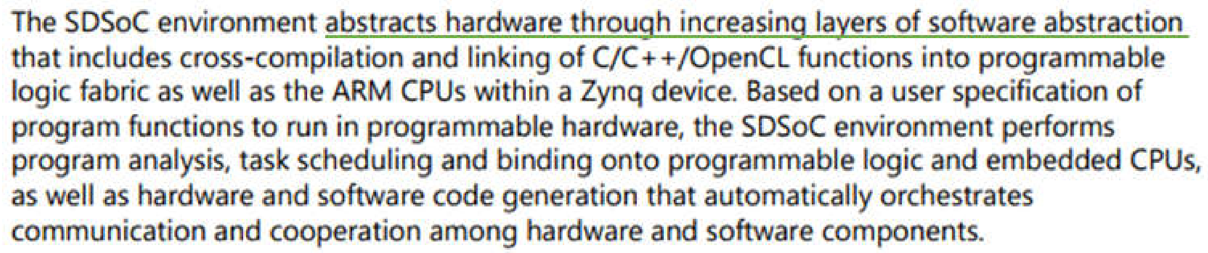
SDK includes the following two board support packages for application development:

- **Standalone** - A simple, semi-hosted, and single-threaded environment that provides basic features such as standard input/output and access to processor hardware features
- **Xilkernel** - A simple and lightweight kernel that provides POSIX style services such as scheduling, threads, synchronization, message passing, and timers

When a board support package is built, the following software components are automatically included into the library:

- Standard C Libraries (libc.a, libm.a)
- Device Drivers for all the peripherals in your design (libxil.a)

Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_1/SDK_Doc/index.html


The SDSoC environment abstracts hardware through increasing layers of software abstraction that includes cross-compilation and linking of C/C++/OpenCL functions into programmable logic fabric as well as the ARM CPUs within a Zynq device. Based on a user specification of program functions to run in programmable hardware, the SDSoC environment performs program analysis, task scheduling and binding onto programmable logic and embedded CPUs, as well as hardware and software code generation that automatically orchestrates communication and cooperation among hardware and software components.

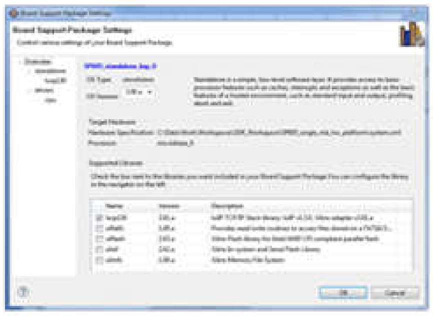
Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_2/ug1028-sdsoc-intro-tutorial.pdf

17. In addition to the generic drivers and HAL, the Product (e.g., XSDK and/or SDSoC) also includes processor-specific application handler code with specific functions of a device driver for at least one of the types of the hardware modules (e.g., the processor-specific application handler code is specific to particular Zynq and/or MicroBlaze microprocessors).

Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

Custom Design Aware

Xilinx SDK understands the custom embedded hardware design that has been defined in the Vivado Design Suite. Based on this design, several key parameters are auto-configured, including memory maps, peripheral register settings, tools and library paths, compiler options, JTAG and flash memory settings, debugger connections, and Linux and bare-metal Board Support Packages (BSPs). This custom design-aware pre-configuration, combined with the auto-generation of critical system software, ensures that software development can progress rapidly with a minimal learning curve.



Drivers and Libraries

XSDK includes user-customizable drivers for all supported Xilinx hardware IPs, POSIX compliant kernel library and networking and file handling libraries. These libraries and drivers can scale for the custom-design based on feature needs, memory requirements and hardware capabilities.

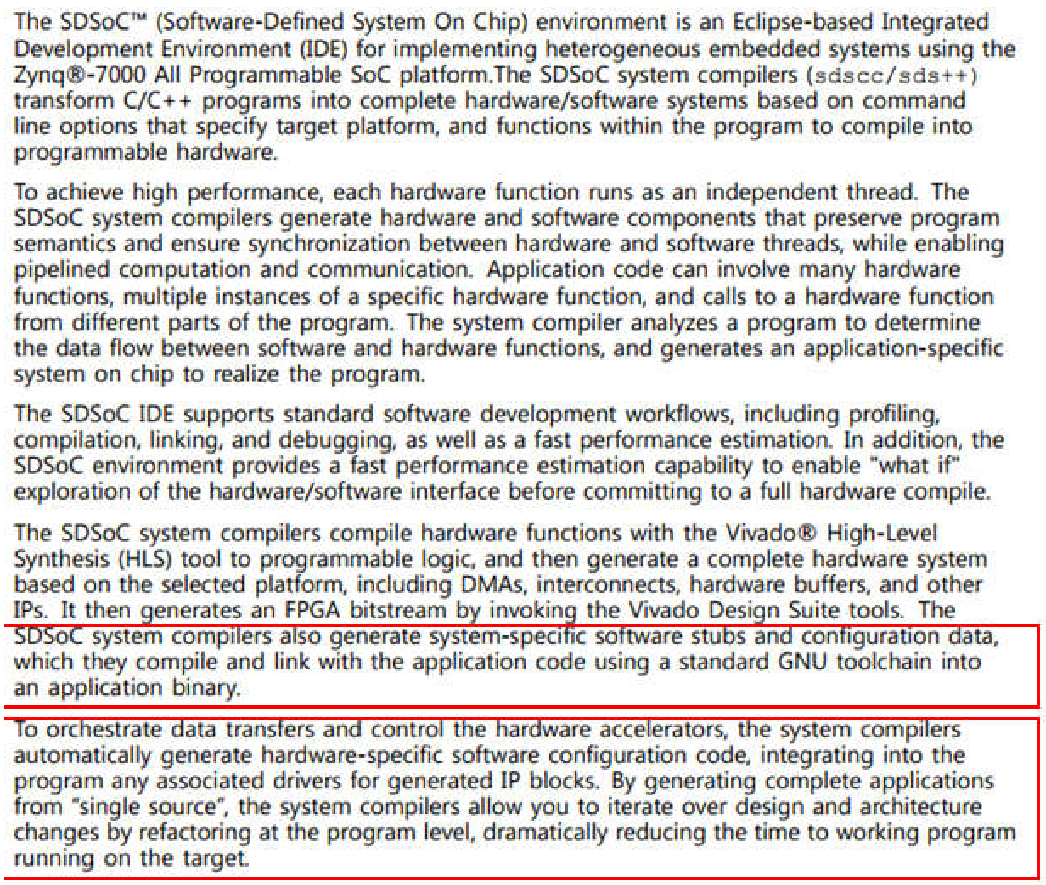
Software Profiling

XSDK includes profiling tools that help to identify bottle necks in your code occurring due to the interaction of functions that are executed within the programmable logic, and on the processor. Supports hierarchical profiling - allowing the user to view which called functions, or which calling functions are affecting processor performance the most.

Source: https://www.xilinx.com/products/design-tools/embedded-software/sdk.html

The Hardware Abstraction Layer (HAL) provides common functions related to register IO, exception, and cache. These common functions are uniform across MicroBlaze™ and Cortex A9 processors. The Standalone platform document provides some processor specific functions and macros for accessing the processor-specific features.

Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_3/oslib_rm.pdf


The SDSoC™ (Software-Defined System On Chip) environment is an Eclipse-based Integrated Development Environment (IDE) for implementing heterogeneous embedded systems using the Zynq®-7000 All Programmable SoC platform.The SDSoC system compilers (`sdscc/sds++`) transform C/C++ programs into complete hardware/software systems based on command line options that specify target platform, and functions within the program to compile into programmable hardware.

To achieve high performance, each hardware function runs as an independent thread. The SDSoC system compilers generate hardware and software components that preserve program semantics and ensure synchronization between hardware and software threads, while enabling pipelined computation and communication. Application code can involve many hardware functions, multiple instances of a specific hardware function, and calls to a hardware function from different parts of the program. The system compiler analyzes a program to determine the data flow between software and hardware functions, and generates an application-specific system on chip to realize the program.

The SDSoC IDE supports standard software development workflows, including profiling, compilation, linking, and debugging, as well as a fast performance estimation. In addition, the SDSoC environment provides a fast performance estimation capability to enable "what if" exploration of the hardware/software interface before committing to a full hardware compile.

The SDSoC system compilers compile hardware functions with the Vivado® High-Level Synthesis (HLS) tool to programmable logic, and then generate a complete hardware system based on the selected platform, including DMAs, interconnects, hardware buffers, and other IPs. It then generates an FPGA bitstream by invoking the Vivado Design Suite tools. The SDSoC system compilers also generate system-specific software stubs and configuration data, which they compile and link with the application code using a standard GNU toolchain into an application binary.

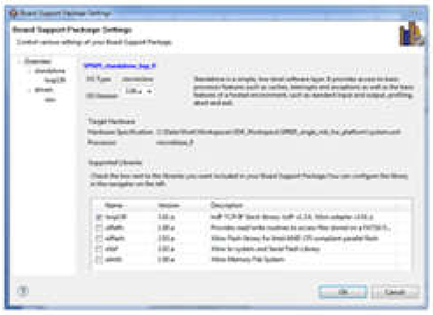
To orchestrate data transfers and control the hardware accelerators, the system compilers automatically generate hardware-specific software configuration code, integrating into the program any associated drivers for generated IP blocks. By generating complete applications from "single source", the system compilers allow you to iterate over design and architecture changes by refactoring at the program level, dramatically reducing the time to working program running on the target.

Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_2/ug1028-sdsoc-intro-tutorial.pdf

18. The Product (e.g., XSDK and SDSoC) generates system-specific application handler code by defining a specific element such as data structures and functions that are to be handled by one or more generic application functions in the Board Support Packages (BSP), HAL and/or generic drivers. When specific functions are written for handling defined specific elements, the specific functions must be registered. The BSP, HAL and/or specific code accordingly contains data structures that register and embed the required functions. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

Custom Design Aware

Xilinx SDK understands the custom embedded hardware design that has been defined in the Vivado Design Suite. Based on this design, several key parameters are auto-configured, including memory maps, peripheral register settings, tools and library paths, compiler options, JTAG and flash memory settings, debugger connections, and Linux and bare-metal Board Support Packages (BSPs). This custom design-aware pre-configuration, combined with the auto-generation of critical system software, ensures that software development can progress rapidly with a minimal learning curve.



Board Support Package Settings

Drivers and Libraries

XSDK includes user-customizable drivers for all supported Xilinx hardware IPs, POSIX compliant kernel library and networking and file handling libraries. These libraries and drivers can scale for the custom-design based on feature needs, memory requirements and hardware capabilities.

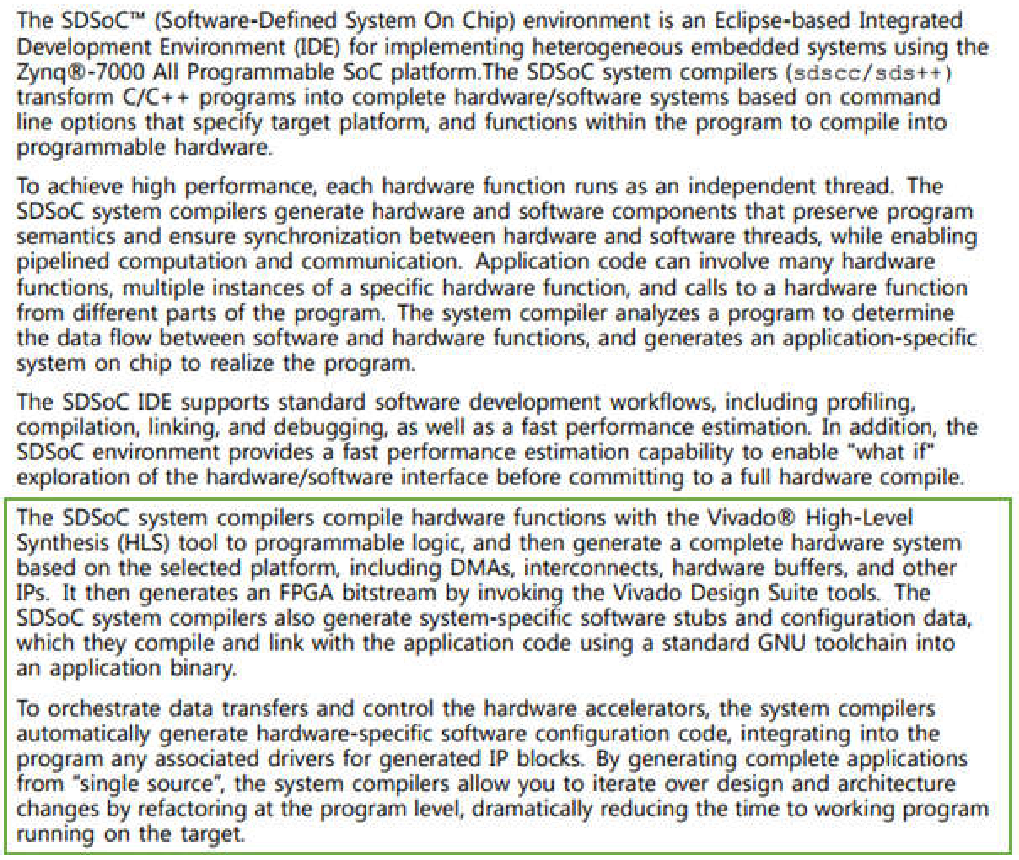
Software Profiling

XSDK includes profiling tools that help to identify bottle necks in your code occurring due to the interaction of functions that are executed within the programmable logic, and on the processor. Supports hierarchical profiling - allowing the user to view which called functions, or which calling functions are affecting processor performance the most.

Source: https://www.xilinx.com/products/design-tools/embedded-software/sdk.html

The Hardware Abstraction Layer (HAL) provides common functions related to register IO, exception, and cache. These common functions are uniform across MicroBlaze™ and Cortex A9 processors. The Standalone platform document provides some processor specific functions and macros for accessing the processor-specific features.

Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_3/oslib_rm.pdf


The SDSoC™ (Software-Defined System On Chip) environment is an Eclipse-based Integrated Development Environment (IDE) for implementing heterogeneous embedded systems using the Zynq®-7000 All Programmable SoC platform.The SDSoC system compilers (`sdscc/sds++`) transform C/C++ programs into complete hardware/software systems based on command line options that specify target platform, and functions within the program to compile into programmable hardware.

To achieve high performance, each hardware function runs as an independent thread. The SDSoC system compilers generate hardware and software components that preserve program semantics and ensure synchronization between hardware and software threads, while enabling pipelined computation and communication. Application code can involve many hardware functions, multiple instances of a specific hardware function, and calls to a hardware function from different parts of the program. The system compiler analyzes a program to determine the data flow between software and hardware functions, and generates an application-specific system on chip to realize the program.

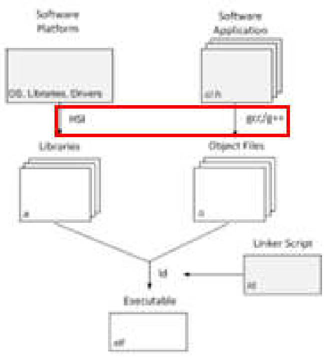
The SDSoC IDE supports standard software development workflows, including profiling, compilation, linking, and debugging, as well as a fast performance estimation. In addition, the SDSoC environment provides a fast performance estimation capability to enable "what if" exploration of the hardware/software interface before committing to a full hardware compile.

The SDSoC system compilers compile hardware functions with the Vivado® High-Level Synthesis (HLS) tool to programmable logic, and then generate a complete hardware system based on the selected platform, including DMAs, interconnects, hardware buffers, and other IPs. It then generates an FPGA bitstream by invoking the Vivado Design Suite tools. The SDSoC system compilers also generate system-specific software stubs and configuration data, which they compile and link with the application code using a standard GNU toolchain into an application binary.

To orchestrate data transfers and control the hardware accelerators, the system compilers automatically generate hardware-specific software configuration code, integrating into the program any associated drivers for generated IP blocks. By generating complete applications from "single source", the system compilers allow you to iterate over design and architecture changes by refactoring at the program level, dramatically reducing the time to working program running on the target.

Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_2/ug1028-sdsoc-intro-tutorial.pdf

19. When a specific application is needed for a particular hardware, the generic functions and the specific functions are compiled together to yield a machine readable code. Xilinx and/or its customers compile the generic functions and the specific functions using XSDK, SDSoc or any other IDE supported by Xilinx. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.



The first step in developing a software application is to create a Board Support Packages (SDK) that will be used by the application. Then, you can create an application project.

To build an executable for this application, SDK automatically performs the following actions. Configuration options can also be provided for these steps.

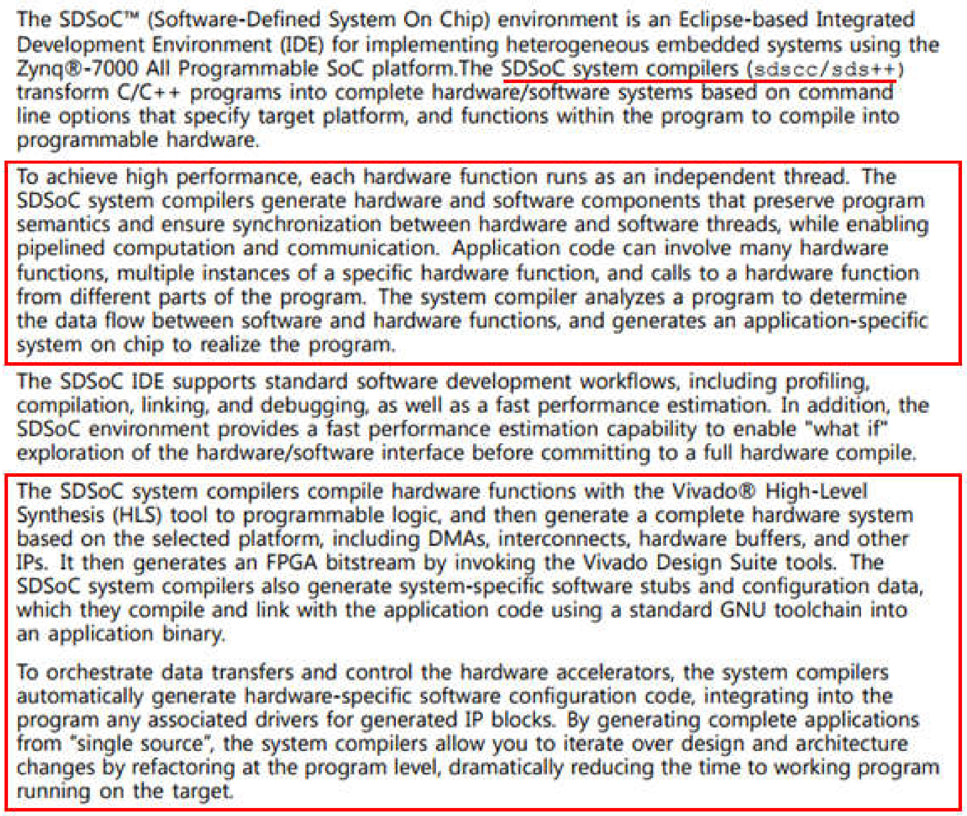
1. SDK builds the board support package. This is sometimes called a software platform.
2. SDK compiles the application software using a platform-specific gcc/g++ compiler.
3. The object files from the application and the Board Support Package are linked together to form the final executable. This step is performed by a linker which takes as input a set of object files and a linker script that specifies where object files should be placed in memory.

Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_1/SDK_Doc/index.html

Full System Optimizing Compiler

- Compiles C/C++/OpenCL applications into a fully functional Zynq SoC and MPSoC system
- Automatic function acceleration in programmable logic generating both the ARM software and FPGA bitstream
- Optimizes the system connectivity and allows rapid system exploration of throughput, latency and area tradeoffs

Source: https://www.xilinx.com/products/design-tools/software-zone/sdsoc.html#overview


The SDSoC™ (Software-Defined System On Chip) environment is an Eclipse-based Integrated Development Environment (IDE) for implementing heterogeneous embedded systems using the Zynq®-7000 All Programmable SoC platform.The SDSoC system compilers (`sdscc/sds++`) transform C/C++ programs into complete hardware/software systems based on command line options that specify target platform, and functions within the program to compile into programmable hardware.

To achieve high performance, each hardware function runs as an independent thread. The SDSoC system compilers generate hardware and software components that preserve program semantics and ensure synchronization between hardware and software threads, while enabling pipelined computation and communication. Application code can involve many hardware functions, multiple instances of a specific hardware function, and calls to a hardware function from different parts of the program. The system compiler analyzes a program to determine the data flow between software and hardware functions, and generates an application-specific system on chip to realize the program.

The SDSoC IDE supports standard software development workflows, including profiling, compilation, linking, and debugging, as well as a fast performance estimation. In addition, the SDSoC environment provides a fast performance estimation capability to enable "what if" exploration of the hardware/software interface before committing to a full hardware compile.

The SDSoC system compilers compile hardware functions with the Vivado® High-Level Synthesis (HLS) tool to programmable logic, and then generate a complete hardware system based on the selected platform, including DMAs, interconnects, hardware buffers, and other IPs. It then generates an FPGA bitstream by invoking the Vivado Design Suite tools. The SDSoC system compilers also generate system-specific software stubs and configuration data, which they compile and link with the application code using a standard GNU toolchain into an application binary.

To orchestrate data transfers and control the hardware accelerators, the system compilers automatically generate hardware-specific software configuration code, integrating into the program any associated drivers for generated IP blocks. By generating complete applications from "single source", the system compilers allow you to iterate over design and architecture changes by refactoring at the program level, dramatically reducing the time to working program running on the target.

Source: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_2/ug1028-sdsoc-intro-tutorial.pdf

20. Defendant's actions complained of herein will continue unless Defendant is enjoined by this court.

21. Defendant's actions complained of herein are causing irreparable harm and monetary damage to Plaintiff and will continue to do so unless and until Defendant is enjoined and restrained by this Court.

22. Plaintiff is in compliance with 35 U.S.C. § 287.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff asks the Court to:

(a) Enter judgment for Plaintiff on this Complaint on all causes of action asserted herein;

(b) Enter an Order enjoining Defendant, its agents, officers, servants, employees, attorneys, and all persons in active concert or participation with Defendant who receive notice of the order from further infringement of United States Patent No. 7,069,546 (or, in the alternative, awarding Plaintiff a running royalty from the time of judgment going forward);

(c) Award Plaintiff damages resulting from Defendant's infringement in accordance with 35 U.S.C. § 284;

(d) Award Plaintiff pre-judgment and post-judgment interest and costs; and

(e) Award Plaintiff such further relief to which the Court finds Plaintiff entitled under law or equity.

Dated: December 11, 2017

Respectfully submitted,

/s/ Jay Johnson

JAY JOHNSON
State Bar No. 24067322
D. BRADLEY KIZZIA
State Bar No. 11547550
KIZZIA JOHNSON, PLLC
1910 Pacific Ave., Suite 13000
Dallas, Texas 75201
(214) 451-0164
Fax: (214) 451-0165
jay@kjpllc.com
bkizzia@kjpllc.com

ATTORNEYS FOR PLAINTIFF

EXHIBIT A