IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
AUSTIN DIVISION

LUCIO DEVELOPMENT LLC,	§	
	§	
Plaintiff,	§	Case No: 1:17-cv-1150
	§	
vs.	§	PATENT CASE
	§	
CYPRESS SEMICONDUCTOR CORPORATION	§	
	§	
Defendant.	§	
	§	

COMPLAINT

Plaintiff Lucio Development LLC ("Plaintiff" or "Lucio") files this Complaint against Cypress Semiconductor Corporation ("Defendant" or "Cypress") 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 a place of business at 198 Champion Ct., San Jose, CA 95134-1709. Defendant may be served with process in this judicial district by serving its registered agent for service of process:

Corporation Service Company, 2711 Centerville Rd., Suite 400, Wilmington, DE 19808.

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 5204 E. Ben White Blvd., Austin, TX 78741 and at 9442 N. Capital of Texas Hwy, Austin, TX 78759.

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 PSoC platform, PSoC Creator, and any similar products ("Product"), which infringe at least Claim 1 of the '546 Patent.

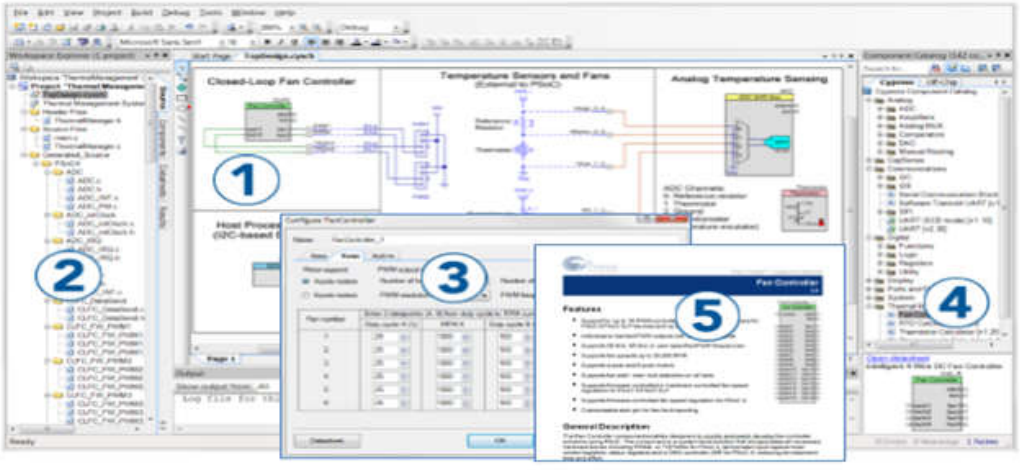
15. The Product is a framework that is configured to create embedded software for multiple hardware modules (e.g., versions of a microcontroller, such as ARM Cortex-Mo or Cortex Mo+). Defendant and/or its customers use the Product (e.g., PSoC and/or WICED) to produce embedded software. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

PSoC Creator

PSoC Creator is a free Windows-based Integrated Development Environment (IDE). It enables concurrent hardware and firmware design of PSoC 3, PSoC 4, and PSoC 5LP systems. See Figure 1 – with PSoC Creator, you can:

1. Drag and drop Components to build your hardware system design
2. Co-design your application firmware with the PSoC hardware
3. Configure Components with config tools
4. Explore the library of 100+ Components
5. Review Component datasheets

Figure 1. PSoC Creator Features



Source: http://www.cypress.com/file/46106/download

USB-Serial Software Development Kit

Last Updated: Jun 02, 2016

Cypress delivers the complete library and driver stack for USB-Serial Bridge Controller devices, in order to easily integrate USB interface into any embedded application. The Software Development Kit (SDK) comes with configuration tool (Windows only), drivers (Windows only), libraries and application examples.

In CDC mode (CDC device class) the device will come up as Virtual COM Port (VCP) device. Native APIs can be used to access the device in CDC mode for all the OS. In Vendor mode, the device enumerates as a USB device and is accessed using the Cypress provided library (for all OS).

For Linux and OS-X, there are no installation steps necessary to use products with USB ports powered by Cypress' USB-Serial products. Linux and OS-X does not need separate driver or library in CDC device class operation. Please use native Serial communication API's for accessing the CDC mode device. There is no special driver needed for Vendor mode device operation in Linux and OS-X. Please refer to API documentation for accessing device using Cypress provided library.

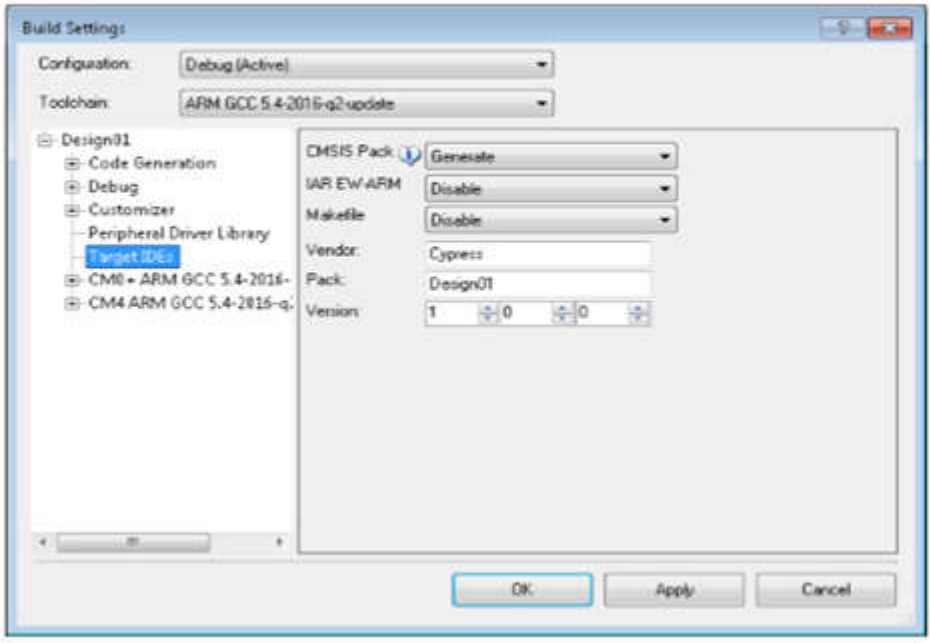
For Android, the user must have root access to use the Cypress library.

Source: http://www.cypress.com/documentation/software-and-drivers/usb-serial-software-development-kit

16. The Product provides one or more generic application handler programs (e.g., PSoC provides a CMSIS Pack containing CMSIS Hardware Abstraction Layer (CMSIS) containing programs, functions and data structures which are common and uniform across all supported ARM Cortex Mo or Cortex Mo+ processors). The generic programs comprise computer program code for performing generic application functions common to multiple types of hardware modules used in a communication environment (e.g., the generic code provides common and generic functions to multiple hardware modules (versions of a microcontroller, such as ARM Cortex-Mo or Cortex Mo+)). Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

Target IDEs Build Settings

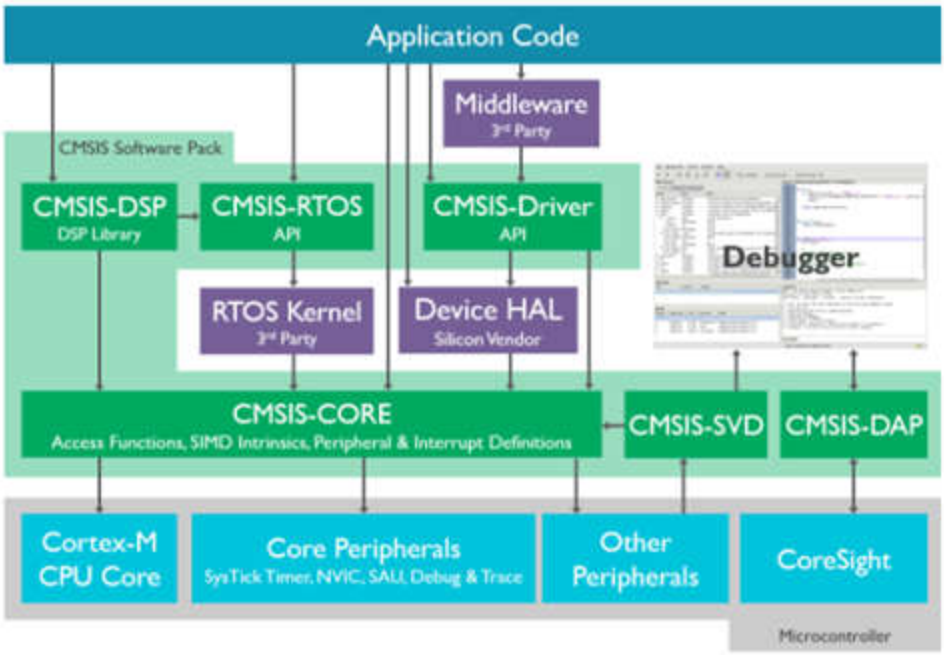
The Target IDEs section of the Build Settings dialog applies to **PSoC 6 devices only**. Use the pull-down menus to select one or more IDEs for which to generate files. You will then use these files to further develop the PSoC Creator design in those selected IDEs. For more information, refer to Integrating into 3rd Party IDEs.



Source: http://www.cypress.com/file/137441/download

Starting from CMSIS-CORE, a vendor-independent hardware abstraction layer for Cortex-M processors, CMSIS has since expanded into areas such as software component management and reference debugger interfaces. Creation of software is a major cost factor in the embedded industry. Standardizing the software interfaces across all Cortex-M silicon vendor products, especially when creating new projects or migrating existing software to a new device, means significant cost reductions.

CMSIS is defined in close cooperation with various silicon and software vendors and provides a common approach to interface to peripherals, real-time operating systems, and middleware components. It simplifies software reuse, reducing the learning curve for new microcontroller developers and cutting the time-to market for devices.



Source: https://developer.arm.com/embedded/cmsis

17. The Product includes generating specific application handler code to associate the generic functions with the specific functions at a device driver for at least one of the types of hardware modules. For example, in addition to the generic drivers and CMSIS Pack, PSoC also includes specific application handler code that is specific to the application and specific to particular processor families. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

Generated Files (PSoC 3, PSoC 4, PSoC 5LP)

Upon a successful build, PSoC Creator generates various files that become a part of your design. These files are listed in the Workspace Explorer under the **Source** tab. These files are specific to the device family (PSoC 3, PSoC 4, or PSoC 5LP) and the selected compiler. The following lists and describes the files generated from a build.

File(s)	Description
cy_boot (Refer also to the System Reference Guide.)	
CyBootAsmKeil.a51 (PSoC 3) CyBootAsmIar.s (PSoC 4/5LP IAR) CyBootAsmGnu.s (PSoC 4/5LP GCC) CyBootAsmRv.s (PSoC 4/5LP MDK)	Provides device- and toolchain-specific assembly implementations for startup and time critical routines.
CyDmac.c /.h	The software API for using the DMA Controller.
CyFlash.c /.h	The software API for writing to flash.
CyLib.c /.h	The software APIs for power management, string/character routines, memory manipulation, as well as enabling/disabling selected portions of the PSoC device.
cypins.h	Contains the function prototypes and constants used for port/pin access and control.
cyPm.c/.h	Provides the function definitions for the power management API.
CySpc.c /.h	The software API for writing to the System Performance Controller.
cytypes.h	Provides macros and defines to allow code to be written tool chain and processor agnostic.
cyutils.c	Implements low-level utility functions used to provide tool chain/processor agnostic functions. Exposed in cytypes.h.
cymem.a51 (PSoC 3)	Specialized memory routines for Keil boot-up.
KeilStart.a51 (PSoC 3)	Bootup code for PSoC 3 chips using Keil tools.
Cm3Start.c/Cm0Start.c/Cm0pStart.c	Startup code for the ARM CM3/CM0/CM0+.

Source: http://www.cypress.com/file/355131/download

File(s)	Description
PSoC3_8051.h /.inc (PSoC 3)	8051 register definitions for the PSoC 3 architecture.
cm3gcc.ld/cm0gcc.ld	Linker script for the GCC toolchain
Cm3RealView.scat/Cm0RealView.scat	Scatter file for the RealView & MDK toolchains
core_cm0.h or core_cm3.h core_cmFunc.h and core_cmInstr.h	CMSIS standard libraries for Cortex-M series of processors: core_cm0.h for PSoC 4/core_cm3.h for PSoC 5LP. Both files included for PSoC 4/PRoC BLE and PSoC 5LP.
core_cm0_psoc4.h or core_cm3_psoc5.h	PSoC 4/PRoC BLE or PSoC 5LP specific interrupt information for CMSIS libraries.
General	
cydevice_trm.h	Defines all of the addresses in the configuration space of the device. These addresses do not contain any context information related to instances drawn in your design(s). You should not need to use any of these addresses directly.
cydevice.h (PSoC 3/PSoC 5LP)	Deprecated version of cydevice_trm.h.
cydevicekeil_trm.inc (PSoC 3) cydevicegnu_trm.inc (PSoC 5LP / PSoC 4 GCC) cydevicerv_trm.inc (PSoC 5LP / PSoC 4 MDK) cydeviciar_trm.inc (PSoC 5LP / PSoC 4 IAR)	Defines all of the addresses in the configuration space of the device for the specific toolchain. These addresses do not contain any context information related to instances drawn in your design(s). You should not need to use any of these addresses directly.
cydevicekeil.inc (PSoC 3) cydevicegnu.inc (PSoC 5LP GCC) cydevicerv.inc (PSoC 5LP Real View)	Deprecated version of cydevicekeil_trm.inc, cydevicegnu_trm.inc, or cydevicerv_trm.inc.
cyfitter.h	Defines all of the instance specific addresses calculated by the Code Generation step. This file is mainly intended for use by instance API implementations, although advanced users may find interesting information in this file.
cyfitter_cfg.c	Implements the methods and logic necessary to configure the device before main. You should not need to use anything implemented in this file.
cyfitter_cfg.h	Contains definitions used by the boot firmware to configure the device before main. You should not need to use anything defined in this file.

Source: http://www.cypress.com/file/355131/download

18. The Product generates specific application handler code and defines a specific element in the specific code to be handled by one of the generic application functions for that hardware module. For example**,** PSoC generates system-specific application handler code by defining a specific element such as functions and data structure that connect to or extend the generic application functions. When specific functions are written for handling defined specific elements, the specific functions must be registered. PSoC 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.

File(s)	Description
PSoC3_8051.h /.inc (PSoC 3)	8051 register definitions for the PSoC 3 architecture.
cm3gcc.ld/cm0gcc.ld	Linker script for the GCC toolchain
Cm3RealView.scat/Cm0RealView.scat	Scatter file for the RealView & MDK toolchains
core_cm0.h or core_cm3.h core_cmFunc.h and core_cmInstr.h	CMSIS standard libraries for Cortex-M series of processors: core_cm0.h for PSoC 4/core_cm3.h for PSoC 5LP. Both files included for PSoC 4/PRoC BLE and PSoC 5LP.
core_cm0_psoc4.h or core_cm3_psoc5.h	PSoC 4/PRoC BLE or PSoC 5LP specific interrupt information for CMSIS libraries.
General	
cydevice_trm.h	Defines all of the addresses in the configuration space of the device. These addresses do not contain any context information related to instances drawn in your design(s). You should not need to use any of these addresses directly.
cydevice.h (PSoC 3/PSoC 5LP)	Deprecated version of cydevice_trm.h.
cydevicekeil_trm.inc (PSoC 3) cydevicegnu_trm.inc (PSoC 5LP / PSoC 4 GCC) cydevicerv_trm.inc (PSoC 5LP / PSoC 4 MDK) cydeviciar_trm.inc (PSoC 5LP / PSoC 4 IAR)	Defines all of the addresses in the configuration space of the device for the specific toolchain. These addresses do not contain any context information related to instances drawn in your design(s). You should not need to use any of these addresses directly.
cydevicekeil.inc (PSoC 3) cydevicegnu.inc (PSoC 5LP GCC) cydevicerv.inc (PSoC 5LP Real View)	Deprecated version of cydevicekeil_trm.inc, cydevicegnu_trm.inc, or cydevicerv_trm.inc.
cyfitter.h	Defines all of the instance specific addresses calculated by the Code Generation step. This file is mainly intended for use by instance API implementations, although advanced users may find interesting information in this file.
cyfitter_cfg.c	Implements the methods and logic necessary to configure the device before main. You should not need to use anything implemented in this file.
cyfitter_cfg.h	Contains definitions used by the boot firmware to configure the device before main. You should not need to use anything defined in this file.

Source: http://www.cypress.com/file/355131/download

Generated Files (FM0+)

Upon a successful build, PSoC Creator generates various files that become a part of your design. These files are listed in the Workspace Explorer under the **Source** tab. These files are specific to the device series (FM0+) and the selected compiler. The following lists and describes the files generated from a build.

File(s)	Description
pdl	
All files in this folder are copied from the installed peripheral driver library (PDL). See Options Dialog.	
General Files	
cydisabledsheets.h	File of defines for disabled schematic pages.
cyfitter.h	Defines all of the instance specific addresses calculated by the Code Generation step. This file is mainly intended for use by instance API implementations, although advanced users may find interesting information in this file.
cymetadata.c	This file defines all extra memory spaces that need to be included. This file is automatically generated by PSoC Creator.
pdl_user.h	User settings header file for Peripheral Driver Library.

PSoC® Creator™ User Guide, Document # 001-93417 Rev *G 378

CYPRESS

Building a PSoC Creator Project

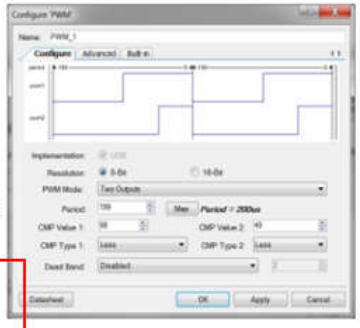
File(s)	Description
project.h	This file includes all of the other header files found in this directory and its sub-directories. It exists for convenience sake, allowing you to include all of the generated headers with just one #include statement.
<device>_rom.icf	Generated IAR linker file for the device. Do not touch.
<device>_rom.id	Generated GCC linker file for the device.

Source: http://www.cypress.com/file/355131/download

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. Cypress and/or its customers compile the generic functions and the specific functions using PSoC and/or any other compiling IDE supported by Cypress Semiconductor. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

PSoC Creator is an Integrated Design Environment (IDE) that enables concurrent hardware and firmware editing, compiling and debugging of PSoC and FM0+ systems. Applications are created using schematic capture and over 150 pre-verified, production-ready peripheral Components.

Components are analog and digital peripherals represented by a symbol that users drag-and-drop into their designs and configure to suit a broad array of application requirements. Each Component in the rich mixed-signal Cypress Component Catalog is configured with a customizer dialog and includes a full set of dynamically generated API libraries. After configuring all the peripherals, firmware can be written, compiled, and debugged within PSoC Creator or exported to leading 3rd party IDEs such as IAR Embedded Workbench®, ARM® Microcontroller Development Kit, and Eclipse™.



PSoC and FM0+ Systems are energy optimized beyond a typical MCU because PSoC Creator optimizes designs to enable only the required functionality. Users can even create custom Components using state machine diagrams or Verilog to further optimize hardware and energy usage.

PSoC Creator is a free Windows-based IDE that includes:

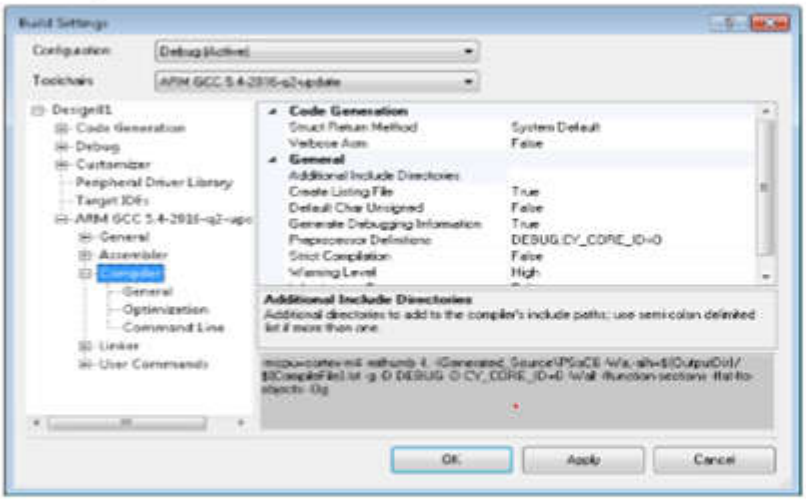
- Hardware design with complete schematic capture and easy-to-use wiring tool
- Over 150 pre-verified, production-ready Components
  - Full communications library including I2C, USB, UART, SPI, CAN, LIN, and Bluetooth Low Energy
  - Digital peripherals with powerful graphical configuration tools
  - Extensive analog signal chain support with amplifiers, filters, ADC and DAC
  - Dynamically generated API libraries
- Free C source code compiler with no code size limitations
- Integrated source editor with inline diagnostics, auto-complete and code snippets
- Built-in debugger

Source: http://www.cypress.com/products/psoc-creator-integrated-design-environment-ide#tabs-0-bottom_side-1

Compiler Build Settings

The Compiler section of the Build Settings dialog is used to control various options that will vary depending on the CPU and compiler.

ARM Options:



Code Generation

- **Struct Return Method** – Specify the method used for returning short structs/methods: system default, register, or memory.
- **Verbose Asm** – Enable extra commentary information in the generated assembly code to make it more readable: true or false.

Source: http://www.cypress.com/file/137441/download

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