EXHIBIT 19

Exhibit G – U.S. Patent No. 6,275,983

Motorola directly and/or indirectly infringes at least claims 1, 7, 16, and 22 of the ’983 patent, either literally or through the doctrine of equivalents. Motorola’s infringing products include mobile devices such as smartphones and tablet computers, including but not limited to the: Atrix, Bravo, Cliq, Cliq XT, Cliq 2, Charm, Defy, Devour, BackFlip, Devour, Droid, Droid 2, Droid 2 Global, Droid X, Droid Pro, Flipout, Flipside, i1, Xoom, (collectively, “the ’983 Accused Products”).[1]

For the purposes of this analysis, Plaintiffs will examine a representative mobile device, Motorola’s Droid X, which is shipped operates with the Android 2.1 Platform. All other ’983 Accused Products meet the limitations of the asserted claims on the same bases as indicated for the Droid X, unless otherwise stated.

In addition to Motorola’s direct infringement of the claims of the ’983 patent through its development, testing, manufacture and use of its devices, Motorola also indirectly infringes claims 7 and 16 of the ’983 patent. Manufacturers, retailers, distributors, end-users and others in the distribution channel of the ’983 Accused Products directly infringe these claims by using, selling, offering for sale, and/or importing these devices into the United States. Motorola contributes to and induces the infringement of asserted claims 7 and 16 through its promotion and provision of intentional marketing, sale and/or technical support of the ’983 Accused Products and associated specialized components in the United States, and through the intentional design, marketing, manufacture, sale, and/or technical support of the ’983 Accused Products abroad to induce direct infringement in the United States. Motorola supplies ’983 Accused Products and actively encourages the use, sale, offer for sale, and importation of the same in the United States through the promotion and provision of marketing literature and user guides, which induces and results in direct infringement. *See, e.g.*, Motorola Droid X User Guide (WI-Apple0034078-34145). Upon information and belief, Motorola has known or should have known that these actions would cause direct infringement of the ’983 patent and did so with specific intent to encourage direct infringement. Additionally, the ’983 Accused Products have no substantial non-infringing uses.

These infringement contentions are preliminary and based only on publicly available information as to the ’983 Accused Products. Motorola has not yet provided discovery as to its accused products and in addition Plaintiff's investigation of Motorola's infringement is ongoing. Based on discovery and Plaintiff's continued investigations Plaintiff reserves the right to amend these contentions to identify additional bases for infringement and additional accused products, including products that Motorola may introduce in the future. Accordingly, Plaintiff reserves its right to amend these contentions as discovery and its investigation proceeds. Also, these disclosures are made based on information ascertained to date, and Plaintiff expressly reserves the right to

[1] Motorola has announced additional smartphones including XRT and Titanium which may also infringe the ’983 Patent. Apple reserves the right to supplement this analysis and this list of accused products as discovery into these newly announced products progresses.

modify or amend the disclosures contained herein based on the Court's claim constructions or to reflect additional information that becomes available to Plaintiff.

U.S. Patent No. 6,275,983	Infringement Contentions
1. A computer system, comprising:	The '983 Accused Products have a processor and are capable of executing numerous computer applications and are, accordingly, computer systems. • For example, the Motorola Droid X includes a Texas Instruments OMAP3630 processor and is capable of executing applications such as a web browser, email client, or telephone dialing application. *See, e.g.*, **Exh. G-1** [Motorola Droid X Specification].
computer hardware for performing native system services	The '983 Accused Products have hardware for performing native system services. • For example, the Motorola Droid X includes a Texas Instruments OMAP3630 processor that is capable of executing the Linux operating system kernel, which provides native system services to Android programs. *See, e.g.*, **Exh. G-1** [Motorola Droid X Specification]; **Exh. G-2** [Android Dev Site, "What is Android?"].
a procedural operating system, having a native interface, for controlling the computer hardware to perform the native system services	The '983 Accused Products have a procedural operating system, having a native interface, for controlling the computer hardware to perform the native system services. • For example, the '983 Accused Products execute a procedural Linux operating system kernel to control the products' hardware to provide native system services: "Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack." *See* **Exh. G-2** [Android Dev Site, "What is Android?"].
object oriented methods requiring native system services;	The '983 Accused Products have object oriented methods requiring native system services. • For example, applications on the '983 Accused Products are written using the Java programming language and run on the Dalvik Virtual Machine, a custom virtual machine designed for embedded use, which runs on top of a Linux kernel. Android provides many object oriented methods written in Java that manage core system

U.S. Patent No. 6,275,983	Infringement Contentions
	services and thus require native system services. • For example, Android provides the Thread.java class which offers threading system services. *See, e.g.*, **Exh. G-3** [Thread.java], **Exh. G-2** [Android Dev Site, "What is Android?"].
procedural program logic code, responsive to invocations of the object-oriented methods during runtime, for causing the procedural operating system to control the computer hardware to perform the required native system services	The '983 Accused Products have procedural program logic code, responsive to invocations of the object-oriented methods during runtime, for causing the procedural operating system to control the computer hardware to perform the required native system services. • For example, the '983 Accused Products include procedural program code for causing the operating system to perform native threading system services, in response to calls from the object-oriented thread methods. For example, Android applications access the native threading system services by accessing procedural program logic code that is responsive to the application's call to an object-oriented method. • For example, the setPriority() method in Thread.java calls setPriority() in VMThread.java, which calls the native setPriority() in java_lang_VMThread.c, which calls dvmChangeThreadPriority in Thread.c, which calls setpriority.s. *See, e.g.*, **Exh. G-3** [Thread.java]; **Exh. G-4** [VMThread.java]; **Exh. G-5 [**java_lang_VMThread.c]; **Exh. G-6** [Thread.c]; **Exh. G-9** [setpriority.s]. • For example, an object instantiated from the RecentCallsListActivity.java class, which forms part of the Contact Manager, creates a new thread before setting its priority. *See e.g.*, **Exh. G-22** [RecentCallsListActivity.java].
executable program memory associated with the computer hardware for runtime execution of the procedural operating system, invocations of the object-oriented methods and related portions of the procedural program logic code	The '983 Accused Products have executable program memory associated with the computer hardware for runtime execution of the procedural operating system, invocations of the object-oriented methods and related portions of the procedural program logic code. • For example, the Motorola Droid X includes 512 MB of RAM and 8GB of FLASH that is associated with a Texas Instruments OMAP3630 processor. The processor is capable of runtime execution of the procedural Linux operating system kernel, as well as the Java object-oriented methods and related portions of the procedural program logic code. *See, e.g.*, **Exh. G-1** [Motorola Droid X Specification], **Exh. G-13** [Droid X Specs].

U.S. Patent No. 6,275,983	Infringement Contentions
means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory; and	The ’983 Accused Products have means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable memory. • For example, in the ’983 Accused Products, the means for means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory include the Dalvik Virtual Machine. S*ee, e.g.*, **Exh. G-7** [Android Dev Site, “ClassLoader”]. • For example, when a method from an object-oriented class is called, the Dalvik Virtual Machine checks to determine whether the class has been loaded. The ClassLoader.loadClass() method is one example of how the Dalvik Virtual Machine achieves this. *Id.* “protected Class<?> loadClass (String className, boolean resolve) Loads the class with the specified name, optionally linking it after loading. The following steps are performed: 1. Call findLoadedClass(String) to determine if the requested class has already been loaded. 2. If the class has not yet been loaded: Invoke this method on the parent class loader. 3. If the class has still not been loaded: Call findClass(String) to find the class.” *Id.* “protected final Class<?> findLoadedClass (String className) Returns the class with the specified name if it has already been loaded by the virtual machine or null if it has not yet been loaded.” *Id.* • For example, loadClass() in ClassLoader.c calls loadClass() in java_lang_VMClassLoader, which calls dvmFindClassByName in InternalNative.c. *See* **Exh. G-19** [ClassLoader.java], **Exh. G-20 [**java_lang_VMClassLoader.c], **Exh. G-23** [InternalNative.c].

U.S. Patent No. 6,275,983	Infringement Contentions
	• For example, findLoadedClass() in ClassLoader.c calls findLoadedClass() in java_lang_VMClassLoader.c, which calls dvmLookupClass() in Class.c. *See* **Exh. G-19** [ClassLoader.java], **Exh. G-20 [**java_lang_VMClassLoader.c], **Exh. G-21** [Class.c].
a runtime loader, responsive to the determinations, to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods.	The '983 Accused Products have a runtime loader, responsive to the determinations, to selectively load the required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods. • For example, if the Dalvik Virtual Machine determines during runtime that requested object-oriented methods are not in the executable program memory prior to invocation, it will selectively load the required methods into memory. The ClassLoader.loadClass() method is one example of how the Dalvik Virtual Machine achieves this. *See, e.g.*, **Exh. G-7** [Android Dev Site, "ClassLoader"]. "protected Class<?> loadClass (String className, boolean resolve) Loads the class with the specified name, optionally linking it after loading. The following steps are performed: 1. Call findLoadedClass(String) to determine if the requested class has already been loaded. 2. If the class has not yet been loaded: Invoke this method on the parent class loader. 3. If the class has still not been loaded: Call findClass(String) to find the class." *Id.* "protected final Class<?> findLoadedClass (String className) Returns the class with the specified name if it has already been loaded by the virtual machine or null if it has not yet been loaded." *Id.* • For example, loadClass() in ClassLoader.c calls loadClass() in java_lang_VMClassLoader, which calls dvmFindClassByName in InternalNative.c. *See* **Exh. G-19** [ClassLoader.java], **Exh. G-20 [**java_lang_VMClassLoader.c], **Exh. G-23**

U.S. Patent No. 6,275,983	Infringement Contentions
	[InternalNative.c]. • For example, findLoadedClass() in ClassLoader.c calls findLoadedClass() in java_lang_VMClassLoader.c, which calls dvmLookupClass() in Class.c. *See* **Exh. G-19** [ClassLoader.java], **Exh. G-20 [**java_lang_VMClassLoader.c], **Exh. G-21** [Class.c].
7. A method for operating a computer system, comprising the steps of: executing a procedural operating system on computer hardware, the procedural operating system including a native interface, responsive to procedural function calls, for providing native system services	The '983 Accused Products perform a method for operating a computer system comprising the steps of: executing a procedural operating system on computer hardware, the procedural operating system including a native interface, responsive to procedural function calls, for providing native system services. • For example, the '983 Accused Products execute a procedural Linux operating system kernel to control the products' hardware to provide native system services such as thread and process management: • "Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack." *See* **Exh. G-2** [Android Dev Site, "What is Android?"].
issuing calls during runtime, compatible with the native interface, to provide the native system services in response to invocations of object-oriented methods requiring such native system services.	The '983 Accused Products perform the step of issuing calls during runtime, compatible with the native interface, to provide the native system services in response to invocations of object-oriented methods requiring such native system services. • For example, Android applications access the native threading system services by accessing procedural program logic code that is responsive to the application's call to an object-oriented method. • For example, the setPriority() method in Thread.java calls setPriority() in VMThread.java, which calls the native setPriority() in java_lang_VMThread.c, which calls dvmChangeThreadPriority in Thread.c, which calls setpriority.s. *See, e.g.*, **Exh. G-3** [Thread.java]; **Exh. G-4** [VMThread.java]; **Exh. G-5 [**java_lang_VMThread.c]; **Exh. G-6** [Thread.c], **Exh. G-9** [setpriority.s]. • For example, an object instantiated from the RecentCallsListActivity.java class, which forms part of the Contact Manager, creates a new thread before setting its priority. *See*

U.S. Patent No. 6,275,983	Infringement Contentions
	e.g., **Exh. G-22** [RecentCallsListActivity.java].
determining during runtime if object-oriented methods to be invoked during runtime execution are present in executable program memory associated with the computer hardware; and	The '983 Accused Products perform the step of determining during runtime if object-oriented methods to be invoked during runtime execution are present in the executable program memory associated with the computer hardware. • For example, when a method from an object-oriented class is called, the Dalvik Virtual Machine checks to determine whether the class has been loaded. The ClassLoader.loadClass() method is one example of how the Dalvik Virtual Machine may perform this step. *See, e.g.*, **Exh. G-7** [Android Dev Site, "ClassLoader"]. "protected Class<?> loadClass (String className, boolean resolve) Loads the class with the specified name, optionally linking it after loading. The following steps are performed: 1. Call findLoadedClass(String) to determine if the requested class has already been loaded. 2. If the class has not yet been loaded: Invoke this method on the parent class loader. 3. If the class has still not been loaded: Call findClass(String) to find the class." *Id.* "protected final Class<?> findLoadedClass (String className) Returns the class with the specified name if it has already been loaded by the virtual machine or null if it has not yet been loaded." *Id.* • For example, loadClass() in ClassLoader.c calls loadClass() in java_lang_VMClassLoader, which calls dvmFindClassByName in InternalNative.c. *See* **Exh. G-19** [ClassLoader.java], **Exh. G-20 [**java_lang_VMClassLoader.c], **Exh. G-23** [InternalNative.c]. • For example, findLoadedClass() in ClassLoader.c calls findLoadedClass() in java_lang_VMClassLoader.c, which calls dvmLookupClass() in Class.c. *See* **Exh. G-19**,

U.S. Patent No. 6,275,983	Infringement Contentions
	[ClassLoader.java], **Exh. G-20 [**java_lang_VMClassLoader.c], **Exh. G-21** [Class.c].
selectively loading the object-oriented methods into the executable program memory during runtime before invocation thereof, if not yet loaded.	The '983 Accused Products perform the step of selectively loading the objectoriented methods into the executable program memory during runtime before invocation thereof, if not yet loaded. • For example, if the Dalvik Virtual Machine determines during runtime that requested object-oriented methods are not in the executable program memory prior to invocation, it will selectively load the required methods into memory. The ClassLoader.loadClass() method is one example of how the Dalvik Virtual Machine may perform this step. *See, e.g.*, **Exh. G-7** [Android Dev Site, "ClassLoader"]. "protected Class<?> loadClass (String className, boolean resolve) Loads the class with the specified name, optionally linking it after loading. The following steps are performed: 1. Call findLoadedClass(String) to determine if the requested class has already been loaded. 2. If the class has not yet been loaded: Invoke this method on the parent class loader. 3. If the class has still not been loaded: Call findClass(String) to find the class." *Id.* "protected final Class<?> findLoadedClass (String className) Returns the class with the specified name if it has already been loaded by the virtual machine or null if it has not yet been loaded." *Id.* • For example, loadClass() in ClassLoader.c calls loadClass() in java_lang_VMClassLoader, which calls dvmFindClassByName in InternalNative.c. *See* **Exh. G-19** [ClassLoader.java], **Exh. G-20 [**java_lang_VMClassLoader.c], **Exh. G-23** [InternalNative.c]. • For example, findLoadedClass() in ClassLoader.c calls findLoadedClass() in

U.S. Patent No. 6,275,983	Infringement Contentions
	java_lang_VMClassLoader.c, which calls dvmLookupClass() in Class.c. *See* **Exh. G-19** [ClassLoader.java], **Exh. G-20 [**java_lang_VMClassLoader.c], **Exh. G-21** [Class.c].
16. A method for operating a computer system including a memory, comprising the steps of:	The '983 Accused Products perform a method for operating a computer system including a memory. The '983 Accused Products have a processor and are capable of executing numerous computer applications and are, accordingly, computer systems. • For example, the Motorola Droid X includes a Texas Instruments OMAP3630 processor and is capable of executing applications such as a web browser, email client, or telephone dialing application. *See, e.g.*, **Exh. G-1** [Motorola Droid X Specification]. The '983 Accused Products have a memory. • For example, the Motorola Droid X includes 512 MB of RAM and 8GB of FLASH that is associated with a Texas Instruments OMAP3630 processor. *See, e.g.*, **Exh. G-1** [Motorola Droid X Specification], **Exh. G-13** [Droid X Specs].
storing in the memory a library of procedural program logic code;	The '983 Accused products perform the step of: storing in the memory a library of procedural program logic code. • For example, the '983 Accused Products include procedural program code for causing the operating system to perform native threading system services, in response to calls from the object-oriented thread methods. For example, Android applications access the native threading system services by accessing procedural program logic code that is responsive to the application's call to an object-oriented method. • For example, the setPriority() method in Thread.java calls setPriority() in VMThread.java, which calls the native setPriority() in java_lang_VMThread.c, which calls dvmChangeThreadPriority in Thread.c, which calls setpriority.s. *See, e.g.*, **Exh. G-3** [Thread.java]; **Exh. G-4** [VMThread.java]; **Exh. G-5** [java_lang_VMThread.c]; **Exh. G-6** [Thread.c], **Exh. G-9** [setpriority.s]. • For example, an object instantiated from the RecentCallsListActivity.java class, which forms part of the Contact Manager, creates a new thread before setting its

U.S. Patent No. 6,275,983	Infringement Contentions
	priority. *See e.g.*, **Exh. G-22** [RecentCallsListActivity.java].
said library including first procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform first type native system services;	The '983 Accused products perform the step of storing in the memory a library of procedural program logic code, said library including first procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform first type native system services. • For example, the '983 Accused Products include procedural program code for causing the operating system to perform native threading system services, in response to calls from the object-oriented thread methods. For example, Android applications access the native threading system services by accessing procedural program logic code that is responsive to the application's call to an object-oriented method. • For example, the setPriority() method in Thread.java calls setPriority() in VMThread.java, which calls the native setPriority() in java_lang_VMThread.c, which calls dvmChangeThreadPriority in Thread.c, which calls setpriority.s. (*See, e.g.*, **Exh. G-3** [Thread.java]; **Exh. G-4** [VMThread.java]; **Exh. G-5** [java_lang_VMThread.c]; **Exh. G-6** [Thread.c], **Exh. G-9** [setpriority.s].) • For example, an object instantiated from the RecentCallsListActivity.java class, which forms part of the Contact Manager, creates a new thread before setting its priority. *See e.g.*, **Exh. G-22** [RecentCallsListActivity.java].
said library including second procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform second type native system services different from said first type;	The '983 Accused products perform the step of storing in the memory a library of procedural program logic code, said library including second procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform second type native system services different from said first type. • For example, the '983 Accused Products include procedural program code for causing the operating system to perform native process system services, in response to calls from the object-oriented process methods. For example, the Process class includes the method killProcess(), which is used to control a task by "kill[ing] the process with a given P*ID*." The killProcess() method invokes the sendSignal() method, which is a native method that accesses services provided by the underlying Linux kernel. *See* **Exh. G-18 [Android Dev Site, "Process"].**" The Process class also defines

U.S. Patent No. 6,275,983	Infringement Contentions
	statements that, for example, invoke the killProcess() method or the sendSignal() method. *Id.* • For example, the ActivityManagerService calls the killProcess() method in Process.java, which calls the native sendSignal(), which is mapped to android_os_Process_sendSignal() in android_util_Process.cpp, which calls kills.S. *See* **Exh. G-14** [ActivityManagerService.java], **Exh. G-15** [Process.java], **Exh. G-16** [android_util_Process.cpp], **Exh. G-17** [kill.S].
executing a procedural operating system in the memory, the procedural operating system including a native interface responsive to procedural function calls, for providing native system services;	The '983 Accused products perform the step of: executing a procedural operating system in the memory, the procedural operating system including a native interface responsive to procedural function calls, for providing native system services. • For example, the '983 Accused Products execute a procedural Linux operating system kernel to control the products' hardware to provide native system services: "Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack." *See* **Exh. G-2** [Android Dev Site, "What is Android?]. • For example, the '983 Accused Products include libraries for thread and process services. *See e.g.*, **Exh. G-6**, [thread.c], **Exh. G-9** [setpriority.s], **Exh. G-16**. [android_util_process.cpp], **Exh. G-17** [kill.S]. • For example, an object instantiated from the RecentCallsListActivity.java class, which forms part of the Contact Manager, creates a new thread before setting its priority. *See e.g.*, **Exh. G-22** [RecentCallsListActivity.java].
running an object-oriented program in a task address space of the memory, the program including an object-oriented method requiring the second type native system services;	The '983 Accused products perform the step of: running an object-oriented program in a task address space of the memory, the program including an object-oriented method requiring the second type native system services. • For example, the Process class includes the method killProcess(), which is used to control a task by "kill[ing] the process with a given P*ID*." This is invoked by, for example, the Activity Manager. *See* **Exh. G-14** [ActivityManagerService.java]. The killProcess()

U.S. Patent No. 6,275,983	Infringement Contentions
	method invokes the sendSignal() method, which is a native method that accesses services provided by the underlying Linux kernel. S*ee* **Exh. G-18** [Android Dev Site, "Process."]. The Process class also defines statements that, for example, invoke the killProcess() method or the sendSignal() method. *Id.* These statements are insertable by programmers into object-oriented applications to access services to reference and control a task.
determining during runtime whether said second type procedural program logic code is available in said task address space; and	The '983 Accused products perform the step of: determining during runtime whether said second type procedural program logic code is available in said task address space. • For example, in the '983 Accused Products, the dvmLoadNativeCode() function in Native.c checks whether a given library is loaded. *See e.g.*, **Exh. G-12** [Native.c].
loading said second type procedural program logic code into said task address space during runtime.	The '983 Accused products perform the step of: loading said second type procedural program logic code into said task address space during runtime. • For example, the load() and loadLibrary() methods in Runtime.java, call the nativeLoad() function in java_land_Runtime.c, which calls the dvmLoadNativeCode() function in Native.c. *See e.g.*, **Exh. G-10** [Runtime.java], **Exh. G-11 [java**_lang_Runtime.c], **Exh. G-12** [Native.c].
22. A computer system including an executable program memory, comprising:	The '983 Accused Products have a processor and are capable of executing numerous computer applications and are, accordingly, computer systems. • For example, the Motorola Droid X includes 512 MB of RAM and 8GB of FLASH that is associated with a Texas Instruments OMAP3630 processor. The processor is capable of runtime execution of the procedural Linux operating system kernel, as well as the Java object-oriented methods and related portions of the procedural program logic code. *See, e.g.*, **Exh. G-1** [Motorola Droid X Specification], **Exh. G-13** [Droid X Specs].
a library of procedural program logic code in the computer system which is responsive to invocations of	The '983 Accused Products have an executable program memory comprising a library of procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform native system services.

U.S. Patent No. 6,275,983	Infringement Contentions
object-oriented methods, for causing a procedural operating system to control the computer system to perform native system services;	• For example, the '983 Accused Products include procedural program code for causing the operating system to perform native threading system services, in response to calls from the object-oriented thread methods. For example, Android applications access the native threading system services by accessing procedural program logic code that is responsive to the application's call to an object-oriented method. • For example, the setPriority() method in Thread.java calls setPriority() in VMThread.java, which calls the native setPriority() in java_lang_VMThread.c, which calls dvmChangeThreadPriority in Thread.c, which calls setpriority.s. *See, e.g.*, **Exh. G-3** [Thread.java]; **Exh. G-4** [VMThread.java]; **Exh. G-5** [java_lang_VMThread.c]; **Exh. G-6** [Thread.c], **Exh. G-9** [setpriority.s]. • For example, an object instantiated from the RecentCallsListActivity.java class, which forms part of the Contact Manager, creates a new thread before setting its priority. *See e.g.*, **Exh. G-22** [RecentCallsListActivity.java].
a procedural operating system in the computer system, the procedural operating system including a native interface responsive to procedural function calls, for providing native system services;	The '983 Accused Products have a procedural operating system, having a native interface responsive to procedural function calls, for providing native system services. • For example, the '983 Accused Products execute a procedural Linux operating system kernel to control the products' hardware to provide native system services: "Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack." *See* **Exh. G-2** [Android Dev Site, "What is Android?]. • For example, the setPriority() method in Thread.java calls setPriority() in VMThread.java, which calls the native setPriority() in java_lang_VMThread.c, which calls dvmChangeThreadPriority in Thread.c, which calls setpriority.s. *See, e.g.*, **Exh. G-3** [Thread.java]; **Exh. G-4** [VMThread.java]; **Exh. G-5** [java_lang_VMThread.c]; **Exh. G-6** [Thread.c], **Exh. G-9** [setpriority.s]. • For example, an object instantiated from the RecentCallsListActivity.java class, which forms part of the Contact Manager, creates a new thread before setting its priority. *See e.g.*, **Exh. G-22** [RecentCallsListActivity.java].

U.S. Patent No. 6,275,983	Infringement Contentions
an object-oriented program in the executable program memory, the program including an object-oriented method requiring native system services;	The ’983 Accused Products have an object-oriented program in the executable program memory, the program including an object-oriented method requiring native system services. • For example, applications on the ’983 Accused Products are written using the Java programming language and run on the Dalvik Virtual Machine, a custom virtual machine designed for embedded use, which runs on top of a Linux kernel. Android provides many object oriented methods written in Java that manage core system services and thus require native system services. *See e.g.*, **Exh. G-2** [Android Dev Site, “What is Android?]. • For example, Android provides the Thread.java class which offers threading system services. *See, e.g.*, **Exh. G-3** [Thread.java].
a processor in the computer system for determining during runtime whether procedural program logic code is available in the executable program memory to provide said required native system services;	The ’983 Accused Products have a processor for determining during runtime whether procedural program logic code is available in the executable program memory, to provide required native system services. • For example, the Motorola Droid X includes a Texas Instruments OMAP3630 processor and is capable of executing applications such as a web browser, email client, or telephone dialing application. *See, e.g.*, **Exh. G-1** [Motorola Droid X Specification]. • For example, in the ’983 Accused Products, the dvmLoadNativeCode() function in Native.c checks whether a given library is loaded. Upon information and belief, the Android libraries which are loaded using this function include code for performing native system services. *See e.g.*, **Exh. G-12** [Native.c].
said processor loading procedural program logic code from said library into the executable program memory during runtime to provide said required native system services;	The processor in the ’983 Accused Products loads procedural program logic code from a library into the executable program memory during runtime to provide required native system services. • For example, in the ’983 Accused Products, the means for selectively loading related portions of the procedural program logic code into the executable program memory upon runtime loading of the selected object-oriented methods include the code for loading shared libraries, such as the Java Native Interface (JNI).

U.S. Patent No. 6,275,983	Infringement Contentions
	• For example, load() and loadLibrary() methods in Runtime.java, call the nativeLoad() function in java_land_Runtime.c which calls the dvmLoadNativeCode() function in Native.c. *See e.g.,* **Exh. G-10** [Runtime.java], **Exh. G-11 [java**_lang_Runtime.c], **Exh. G-12** [Native.c]. Upon information and belief, the Android libraries which are loaded by this function include code for performing native system services.
said processor invoking said object-oriented method of said object-oriented program during runtime; and	The processor in the '983 Accused Products invokes an object-oriented method of an object-oriented program during runtime. • For example, applications on the '983 Accused Products are written using the Java programming language and run on the Dalvik Virtual Machine, a custom virtual machine designed for embedded use, which runs on top of a Linux kernel. Android provides many object oriented methods written in Java that manage core system services and thus require native system services. *See e.g.*, **Exh. G-2** [Android Dev Site, "What is Android?]. • For example, Android provides the Thread.java class which offers threading system services. *See*, *e.g.*, **Exh. G-3** [Thread.java].
said loaded procedural program logic code responding to said invoking to cause said procedural operating system to control the computer system to perform said required native system services.	The '983 Accused Products contain procedural program logic code, responsive to invocations of the object-oriented methods during runtime, for causing the procedural operating system to control the computer hardware to perform the required native system services. • For example, the '983 Accused Products include procedural program code for causing the operating system to perform native threading system services, in response to calls from the object-oriented thread methods. For example, Android applications access the native threading system services by accessing procedural program logic code that is responsive to the application's call to an object-oriented method. • For example, the setPriority() method in Thread.java calls setPriority() in VMThread.java, which calls the native setPriority() in java_lang_VMThread.c, which calls dvmChangeThreadPriority in Thread.c, which calls setpriority.s. *See*, *e.g.*, **Exh. G-3** [Thread.java]; **Exh. G-4** [VMThread.java]; **Exh. G-5 [**java_lang_VMThread.c]; **Exh. G-6** [Thread.c], **Exh. G-9** [setpriority.s]. • For example, an object instantiated from the RecentCallsListActivity.java class, which

U.S. Patent No. 6,275,983	Infringement Contentions
	forms part of the Contact Manager, creates a new thread before setting its priority. *See e.g.*, **Exh. G-22** [RecentCallsListActivity.java].