EXHIBIT 6

CONTAINS CONFIDENTIAL BUSINESS INFORMATION SUBJECT TO PROTECTIVE ORDER

IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF WISCONSIN

APPLE INC., and NEXT SOFTWARE, INC. (f/k/a NeXT COMPUTER, INC.), Plaintiffs and Counterclaim-Defendants, v. MOTOROLA, INC. and MOTOROLA MOBILITY, INC. Defendants and Counterclaim-Plaintiffs	Case No. 10-CV-662 (BBC)

INITIAL EXPERT REPORT OF DR. NATHANIEL POLISH REGARDING INFRINGEMENT OF U.S. PATENT NUMBER 6,343,263

TABLE OF CONTENTS

Page

I. INTRODUCTION 1

I. BACKGROUND AND QUALIFICATIONS 1

II. SUMMARY OF OPINIONS 3

III. MATERIALS REVIEWED 3

IV. U.S. PATENT NO. 6,343,263 4

A. Description and Background of the ’263 Patent 4

B. Asserted Claims of the ’263 Patent 10

V. ACCUSED PRODUCTS 11

VI. LEGAL STANDARDS 14

A. Infringement 15

VII. PERSONS OF ORDINARY SKILL IN THE ART 20

VIII. RELEVANT LEGAL PRINCIPLES AND GUIDANCE 20

A. Proposed Constructions 20

IX. INFRINGEMENT OF THE ’263 PATENT: OVERVIEW 22

A. Realtime 22

1. Types of Realtime Systems 22

2. Realtime “Streaming” 24

B. OpenMAX IL API 25

X. INFRINGEMENT OF THE ’263 PATENT BY MOTOROLA’S ANDROID PHONES 28

A. Overview 28

B. Infringement of Claim 1 of the ’263 Patent 30

1. "A signal processing system for providing a plurality of realtime services to and from a number of independent client applications and devices, said system comprising:" 31

2. "(a) a subsystem comprising a host central processing unit (CPU) operating in accordance with at least one application program and a device handler program, said subsystem further comprising an adapter subsystem interoperating with said host CPU and said device;" 32

a. "subsystem comprising a host central processing unit (CPU)" 32

b. "operating in accordance with at least one application program and a device handler program" 33

c. "said subsystem further comprising an adapter subsystem interoperating with said host CPU and said device" 38

3. "(b) a realtime signal processing subsystem for performing a plurality of data transforms comprising a plurality of realtime signal processing operations; and" 39

a. for performing a plurality of data transforms comprising a plurality of realtime signal processing operations; and" 40

4. "(c) at least one realtime application program interface (API) coupled between the subsystem and the realtime signal processing subsystem to allow the subsystem to interoperate with said realtime services." 41

C. Infringement of Claim 2 of the '263 Patent 43

1. "The signal processing system as set forth in claim 1, wherein said signal processing system receives and transmits a plurality of datatypes over a plurality of different wide area networks (WANs)." 43

D. Induced And Contributory Infringement 44

XI. RESERVATION OF THE RIGHT TO SUPPLEMENT 45

XII. CONCLUSION 45

I. INTRODUCTION

1. I have been retained by counsel for Apple, Inc. in this Action to provide analysis, expert opinions, and testimony regarding the issue of infringement of U.S. Patent No. 6,343,263 (the "'263 patent"), including a discussion of the '263 patent and the design and operation of Motorola Mobility, Inc.'s and Motorola Solutions, Inc.'s (collectively, "Motorola") phones and other computing devices running versions of the Android operating system as installed on these Motorola devices ("Accused Products"), which are accused of infringing the '263 patent. At any hearing or trial in this matter I may also provide a tutorial on the design and operation of mobile phone operating systems, and other aspects of the relevant technologies.

2. At trial, I may be asked to testify about the claims and disclosures contained in the '263 patent, including the specification and the figures, as well as relevant portions of the file history. In addition, I may testify regarding the operation of the embodiments of the claimed invention that are described in the '263 patent. I may also be asked to testify regarding any prior art that was cited in the course of prosecution of the '263 patent. I may also testify regarding the general state of the art in the field of computer-based realtime signal processing at the time the invention claimed in the '263 patent was developed.

3. I am being compensated for my time expended in connection with this case at the rate of $500 per hour, plus reimbursement of any expenses I incur. I have no financial stake in this litigation, and my compensation is not contingent upon the outcome of this litigation.

I. BACKGROUND AND QUALIFICATIONS

4. I expect to testify regarding my background, qualifications, and experience relevant to the issues in this litigation. I have a Ph.D. in Computer Science from Columbia University. I hold the following four degrees from Columbia, spanning the years 1980 to 1993:

- Ph.D. in Computer Science, May 1993, Thesis: *Mixed Distance Measures for the Optimization of Concatenative Vocabularies in Speech Synthesi*s;
- M.Phil. in Computer Science, December 1989;
- M.S. in Computer Science, December 1987;
- B.A. in Physics, Columbia College, May 1984.

5. For over twenty-five years, I have run a computer technology development firm that I co-founded, called Daedalus Technology Group. My primary business activity is the development of computer-related products. This activity involves understanding the business objectives of customers, designing products to suit their needs, and supervising the building, testing, and deployment of these products. I develop hardware and software as well as supervise others who do so.

6. Also, from time to time I found other companies in order to pursue particular product opportunities. I develop and ultimately sell these companies. Most of my business activity, however, is as a consulting product developer. From time to time I have also served as an expert witness on computer and software related cases. I am a named inventor on seven United States patents, and am a member of several professional societies, including the IEEE and ACM.

7. I have extensive experience in several areas relevant to this case. From the mid-1980s to the mid-1990s I and my company were engaged in a number of research and development projects related to realtime processing, DSPs, signal processing, speech and audio coding and video serving. I used several early TI DSP chips as part of my dissertation research in the late 1980s. I have developed a number of systems having a variety of realtime aspects and requirements. These included signal processing systems as well as media serving systems.

8. These are similar technologies to those discussed in the ’263 patent. As part of my work on these projects, I was involved in all aspects of designing and developing these systems. I was and am very familiar with the technology.

9. Attached to this report as Appendix C is a copy of my curriculum vitae, which includes a list of the matters in which I have served as an expert witness.

II. SUMMARY OF OPINIONS

10. Based on my analysis of accused Motorola products, the documentation and source code produced by Motorola, as well as other documents and/or source code that I have reviewed, and my understanding of the legal standards and the materials considered, I have formed the following opinions regarding the ’263 patent:

- The Accused Products, which include all Motorola Android phones and tablet devices running versions 1.5 to 3.1 of the Android operating system (as identified in Table 1 below), infringe at least claims 1 and 2 of the ’263 patent.

11. I expect to testify at trial concerning these opinions, the subject matter, disclosure, technology, claim construction and validity of claims 1 and 2 of the ’263 patent, based on my review and study of that patents, relevant prior art, and my experience, knowledge of the field, and education. I also expect to rebut any opinions I disagree with that are provided by Motorola’s expert(s) with respect to the opinions I express in this Report.

12. My analyses relating to infringement of the Accused Products are discussed below.

III. MATERIALS REVIEWED

13. In forming the opinions set forth in this report, I considered and relied upon my education, background, and experience. I also reviewed the ’263 patent and its prosecution history (including the prior art cited therein), as well as the other documents or reference

materials cited or listed in this report. In addition, I have evaluated representative samples of the Accused Products. Exhibit A includes a list of the documents I have reviewed. I reserve the right to rely upon any additional information or materials that may be provided to me or that are relied upon by any of Motorola's experts or witnesses, if called to testify or to give additional opinions regarding this matter.

14. I understand that discovery is ongoing in this matter. I reserve the right to supplement my opinions in this report should additional information become available to me.

IV. U.S. PATENT NO. 6,343,263

A. Description and Background of the '263 Patent

15. The '263 patent, entitled "Real-Time Signal Processing System For Serially Transmitted Data," was filed on August 2, 1994, and issued on January 29, 2002. The named inventors on the '263 patent are James B. Nichols and John Lynch, and the patent was assigned to Apple Computer, Inc. I understand that the '263 patent is still assigned to Apple.

16. The '263 patent provides: "The present invention is directed to the transmission of data to and from a computer, and more particularly to a system for performing real-time signal processing of data that is serially transmitted to and from a computer." ('263 patent at col. 1:5-8.) In particular, at the time of the invention: "Various devices are known for transmitting data between a computer and a remote site via wide-area telecommunications networks. One of the most widely used devices of this type is the modem, which enables data to be transmitted to and from a computer over a wide-area analog telephone network." (*Id.* at col. 1:10-15.)

17. Examples of applications that processed data received from over a modem include facsimile or video conferencing applications. ('263 patent at col. 1:40, 2:39-40.)

18. The patent explains that: "[t]o enhance the performance of modems, a digital signal processor ("DSP") has been incorporated into its structure. In this arrangement, the

modem software was designed to cooperate with the DSP to provide data thereto for processing prior to transmission or after reception over the telephone line." ('263 patent at col. 1:45-50.)

19. A DSP is a processor used to accelerate certain kinds of computations, for example, computations used for processing signals such as an audio signal from a microphone. DSPs were first introduced in the early 1980s with, for example, the TMS320 from Texas Instruments.

20. I had first-hand experience with DSPs in the early 1990s. For example, I used DSPs during that timeframe to perform realtime processing of speech signals.

21. The patent explains "[t]here are three possible implementations of the DSP, respectively identified as hard, soft and native." ('263 patent at col. 6:46-48.) In the hard implementation, the DSP is fixed in a piece of hardware and cannot be reprogrammed or updated without changing the chip. (*Id.* at col. 6:48-52.) In the soft implementation, the DSP is still resident as a separate piece of hardware but is programmable and can be updated as desired. (*Id.* at col. 6:59-64.) "In the native implementation of the DSP, the processor does not reside in a separate piece of hardware. Rather, it's [sic] functions are carried out by the CPU of the host computer." (*Id.* at col. 7:5-7.)

22. In order to achieve the desired performance from a DSP-accelerated system, an engineer needed to write low-level code specifically for the DSP. The process was complicated and time consuming. An engineer who designed a system for a DSP would find that his software architecture lost essentially all hardware independence. The code had to run on a very specific configuration of hardware. As the '263 patent explains, "Further in this regard, the modem control software had to be designed to work with the specific DSP incorporated into the

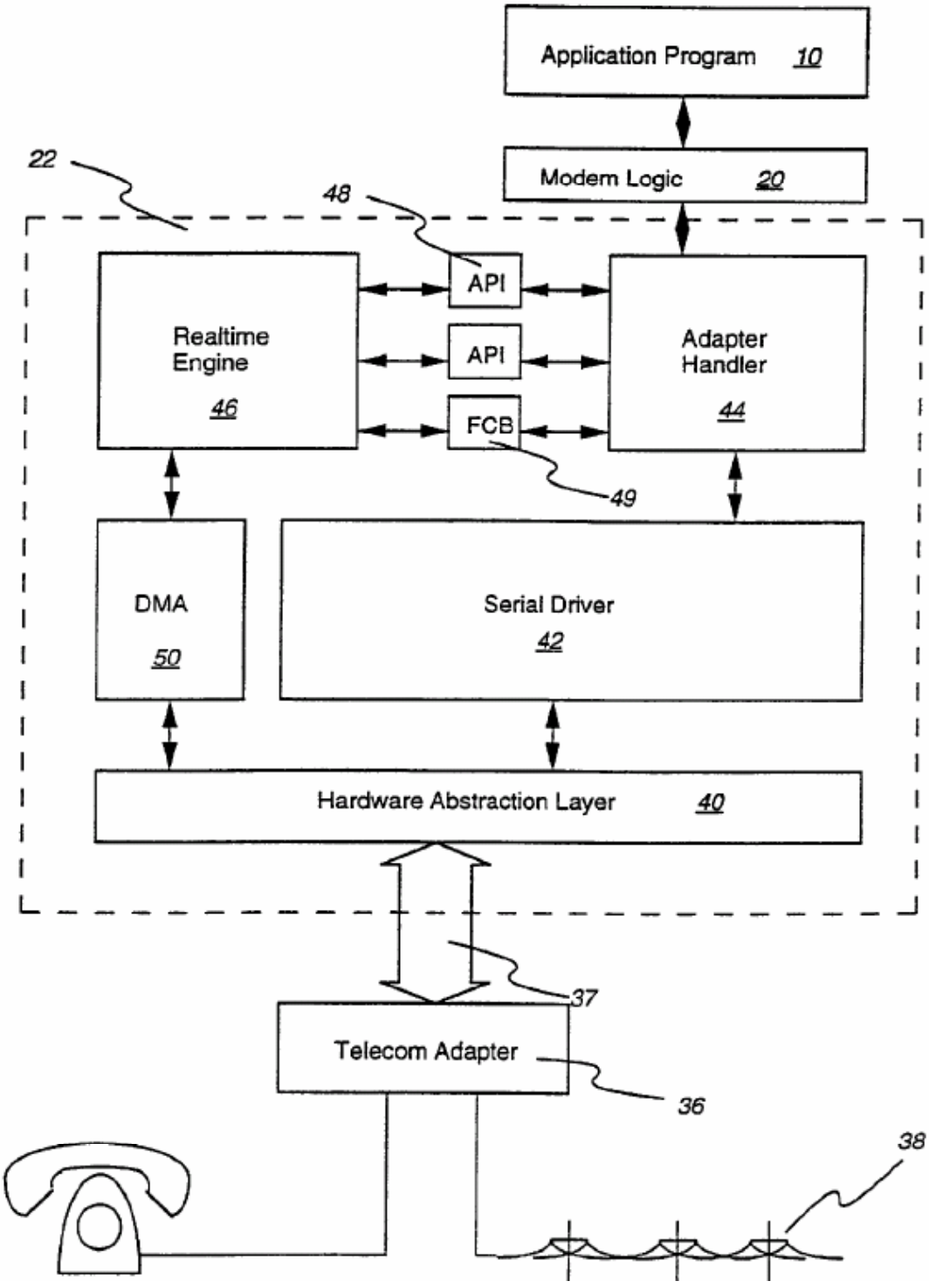
computer. If a different DSP was to be used, the modem control software had to be reprogrammed to work with the new DSP." ('263 patent, at col. 1:63-67.)

23. Additionally, since DSPs had to be communicated with in certain specific ways, there was little flexibility in how an engineer might approach problem-solving with the DSP. Per the patent: "While the addition of the DSP provided increased capabilities in terms of the speed at which the data could be transmitted over a telephone network and the ease with which the modem could be configured, it was still limited in the types of data that could be processed." ('263 patent, at col. 1:50-54.) If tomorrow there was a new audio file format, computer manufacturers would be forced to update each of the DSP-manufacturer-specific code bases. Thus, the patent explains that "it is desirable to provide a signal processing system that is not limited to one specific DSP, but rather one that can operate with any of a variety of different types of signal processors." (*Id.* at col. 2:23-25.)

24. The '263 patent aimed to remedy these problems. At a high level, the '263 patent is concerned with providing realtime APIs through which applications can access the DSP.

25. Per Claim 1 and Figure 2 of the patent, the signal processing system includes two major subsystems, a CPU subsystem (10, 20, 44) and a realtime signal processing subsystem (46). Coupled between the subsystems is a realtime API (48) that allows applications running on the CPU to request realtime services, such as video image processing of data received from over a wide-area network ("WAN"):

FIGURE 2



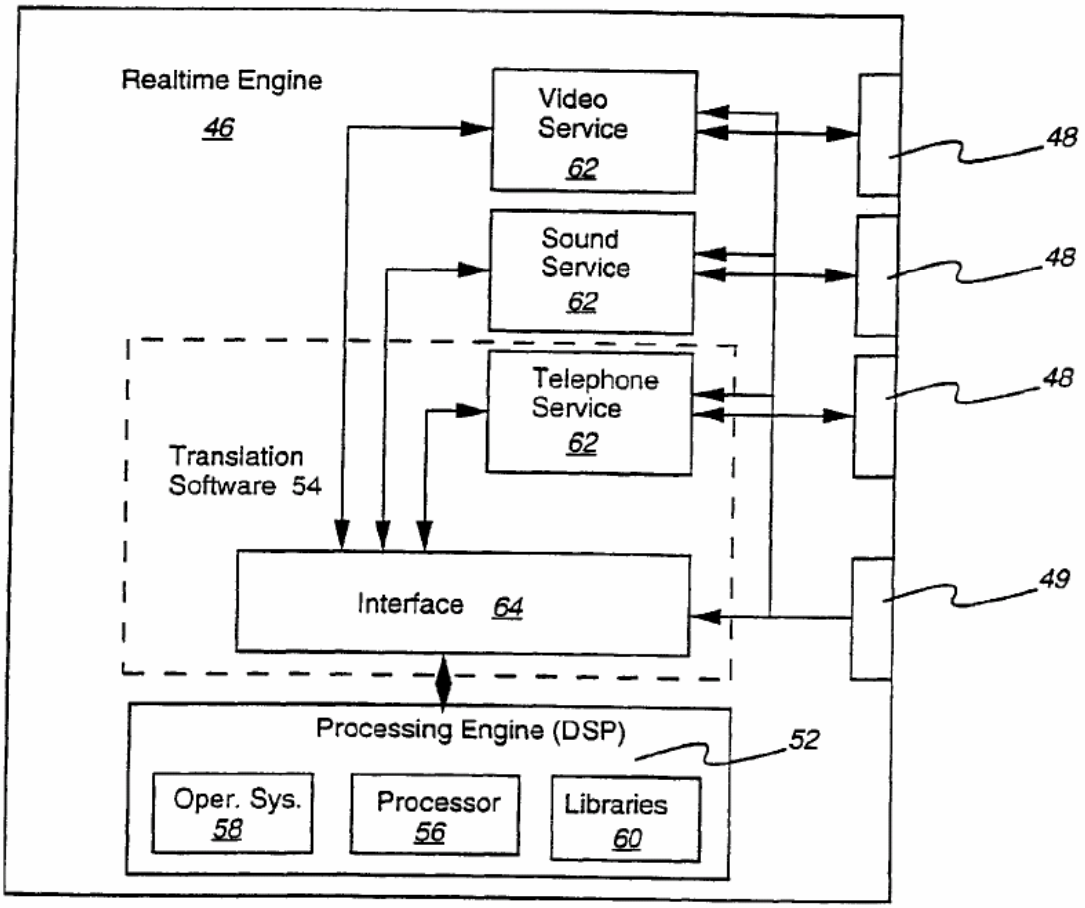
26. The CPU subsystem includes, for example, a host central processing unit (CPU) operating in accordance with at least one application program and a device handler program and

an adapter subsystem that interoperates with the host CPU and the device. Example applications include facsimile or video conferencing applications. (*See* '263 patent at col. 1:40; 2:39-40.) One of ordinary skill in the art would have recognized that the CPU subsystem could include other programs such as driver software.

27. The '263 patent explains that, as illustrated in Figure 2, there can be a number of realtime APIs 48. (*See* '263 patent, at col. 6:25-26.) In particular, each realtime API "represents services for a particular class of functionality. For example, one interface may relate to the operation of the engine as a virtual telephone, another interface can be associated with a virtual sound device, e.g. stereo, and a third interface can pertain to a virtual video device." (*Id.* at col. 6:27-32.) Each realtime API instructs the realtime signal processing subsystem "to carry out the necessary transforms which relate to the function of the virtual device being implemented, e.g. text-to-speech conversion, video image processing, etc." (*Id.* at col. 6:33-38.)

28. Figure 3 of the patent illustrates the architecture of the realtime signal processing subsystem in further detail. ('263 patent, at col. 6:39-40.)

FIGURE 3



29. Per the ’263 patent specification, the generic service providers receive commands from the realtime APIs 48 “to perform the transforms that are required in the operation of the virtual device being implemented.” (’263 patent, at col. 7:16-20.) The generic service providers 62 are labeled as being generic because they are independent of the actual hardware that is used as the DSP. (*Id.* at 7:20-23.) In other words, the realtime communications modules implement the realtime APIs in a DSP-independent manner.

30. The patent explains that “separate generic service providers can be employed for the different virtual devices implemented.” (’263 patent, at col. 7:34-36.) For example, one

service provider could be employed to provide the services of a virtual telephone. (*Id.* at col. 7:36-38.) Another service provider could be used for sound applications, and a third service provider could be used for video applications. (*Id.* at col. 7:44-46.)

31. These generic service providers 68 are made independent by the interface 64. (*See* '263 patent, at col. 7:23-33.) This translation interface program essentially functions as an additional layer of abstraction that virtualizes the DSP, i.e., the realtime communications module is aware of the existence of the DSP but does not need to know if it is hard, soft or native. (*Id.*)

B. Asserted Claims of the '263 Patent

32. I understand that Apple is asserting claims 1 and 2 of the '263 patent. These claims recite:

> 1. A signal processing system for providing a plurality of realtime services to and from a number of independent client applications and devices, said system comprising:
>
> (a) a subsystem comprising a host central processing unit (CPU) operating in accordance with at least one application program and a device handler program, said subsystem further comprising an adapter subsystem interoperating with said host CPU and said device;
>
> (b) a realtime signal processing subsystem for performing a plurality of data transforms comprising a plurality of realtime signal processing operations; and
>
> (c) at least one realtime application program interface (API) coupled between the subsystem and the realtime signal processing subsystem to allow the subsystem to interoperate with said realtime services.
>
> 2. The signal processing system as set forth in claim 1, wherein said signal processing system receives and transmits a plurality of datatypes over a plurality of different wide area networks (WANs).

V. ACCUSED PRODUCTS

33. Motorola designs and sells mobile devices, including phones and tablet computers, that run the Android operating system. These devices also incorporate touchscreens to allow user input. At this time, I understand that the mobile devices that Motorola makes, uses, sells, or offers to sell in the United States, or imports into the United States that run the Android operating system are the Motorola Atrix, Backflip, Bravo, Charm, Citrus, Cliq/Dext, Cliq 2, Cliq XT/Quench, Defy, Devour, Droid, Droid 2, Droid 2 Global, Droid Bionic, Droid Pro, Droid X, Droid X2, Droid 3, Flipout, Flipside, i1, Titanium, Xoom, and XPRT (the "Accused Products"). *See, e.g.*, Respondent Motorola Mobility, Inc's 1st Suppl. Resp. to Complainant Apple Inc.'s 1st Set of Interrog., 337-TA-750, at 22; Respondent Motorola Mobility, Inc's Resp. to Complainant Apple Inc.'s 2d Set of Interrog., 337-TA-750, at 15; Respondent Motorola Mobility, Inc.'s 2d Suppl. Resp. to Complainant Apple's 1st Set of Interrog., 337-TA-750, at 20-21.

34. A table of the Accused Products based on information produced by Motorola as well as public information provided by Motorola at http://developer.motorola.com/products/ is provided in TABLE 1 below:[1]

[1] *See, e.g.*, WI-Apple0034146 (Atrix Tech Specs); WI-Apple1604158 (Atrix Dev Specs); WI-Apple0034176 (Backflip Tech Specs); WI-Apple0147729 (Backflip Dev Specs); WI-Apple0147565 (Bionic Dev Specs); WI-Apple0034213 (Bravo Tech Specs); WI-Apple0147718 (Bravo Dev Specs); WI-Apple0034203 (Charm Tech Specs); WI-Apple0147332 (Charm Dev Specs); WI-Apple0034181 (Citrus Tech Specs); WI-Apple0147224 (Citrus Dev Specs); WI-Apple0034191 (Cliq2 Tech Specs); WI-Apple0147231 (Cliq2 Dev Specs); WI-Apple0034208 (Cliq XT Tech Specs); WI-Apple0147649 (Cliq XT Dev Spec); WI-Apple0147573 (Cliq Dev Specs); WI-Apple0034152 (Defy Tech Specs); WI-Apple0147325 (Defy Dev Specs); WI-Apple0034157 (Devour Tech Specs); WI-Apple0147651 (Devour Dev Specs); WI-Apple0034186 (Droid Tech Specs); WI-Apple0147502 (Droid Dev Specs); WI-Apple0034192 (Droid 2 Tech Specs); WI-Apple0147222 (Droid 2 Dev Specs); WI-Apple0034162 (Droid 2 Global Tech Specs); WI-Apple0147506 (Droid 2 Global Dev Specs) WI-Apple1604162 (Droid 3 Tech Specs); WI-Apple1604161 (Droid 3 Dev Specs); WI-Apple0034170 (Droid Pro Tech Specs); WI-Apple0147220 (Droid Pro Dev Specs); WI-Apple0034198 (Droid X Tech Specs); WI-Apple0147323 (Droid X Tech Specs); WI-Apple1604237 (Droid X2 Tech Spec); WI-Apple1604235 (Droid X2 Dev Specs); WI-Apple0034182 (Flipout Tech Specs); WI-

Product (Code Name)	Product Type	Operating System (Reviewed)	Touch Screen	Processor	Memory (RAM)	Memory (FLASH)
Atrix (Olympus)	Smartphone	Android v2.2 (2.2.2-OLYFR_U4_1.8.3)	Capacitive	NVIDIA Tegra 2 AP20H	1 GB	2 GB
Backflip (Motus)	Smartphone	Android v2.1 (2.1-update1-MOTUS_U3_0.139.0)	Capacitive	Qualcomm MSM7200A	256 MB	512 MB
Bionic (Targa)	Smartphone	Android v2.3.3 (2.3.3-5.5.1-c71610)	Capacitive	NVIDIA Tegra 2 AP20H Dual Core	512 MB	2 GB
Bravo (Kobe)	Smartphone	Android v2.1 (2.1-update1-KOBE_U3_5.35.0)	Capacitive	TI OMAP3610	512 MB	2 GB
Charm (Ciena)	Smartphone	Android v2.1 (2.1-update1-A3013_X_00.58.22P)	Capacitive	TI OMAP3410	512 MB	512 MB
Citrus (Basil)	Smartphone	Android v.2.1 (2.1-update1-BASIL_U3_03.90.7)	Capacitive	Qualcomm MSM7625	256 MB	512 MB
Cliq (Morrison)	Smartphone	Android v.1.5 (Blur_Version.1.0.11.MB200.T-Mobile.en.US) (Blur_Version.1.4.8.MB200.T-Mobile.en.US)	Capacitive	Qualcomm MSM7200A	256 MB	512 MB
Cliq 2 (Begonia)	Smartphone	Android v2.2 (BGN_1.0.23)	Capacitive	TI OMAP3620-1000	512 MB	1 GB
Cliq XT (Zeppelin)	Smartphone	Android v1.5 (MB501_0.12.11_1.3.27)	Capacitive	Qualcomm MSM7200A	256 MB	512 MB
Defy (Jordan)	Smartphone	Android v2.1 (JORDN_U3_6.36.0)	Capacitive	TI OMAP3620-1000	512 MB	2 GB
Devour (Calgary)	Smartphone	Android v1.6 (CALAND_X_01.15.10P)	Capacitive	Qualcomm MSM7627	256 MB	512 MB
Droid (Sholes)	Smartphone	Android v2.1 (MILER_X1_01.22.0) (MILER_X1_01.26.1)	Capacitive	TI OMAP3430	256 MB	512 MB
Droid 2 (Droid2)	Smartphone	Android v2.2 (DROID_X6_2.3.20)	Capacitive	TI OMAP3620-1000	512 MB	8 GB

Apple0147716 (Flipout Dev Specs); WI-Apple0034214 (Flipside Tech Specs); WI-Apple0147330 (Flipside Dev Specs); WI-Apple0034180 (i1 Tech Specs); WI-Apple0147229 (i1 Dev Specs); WI-Apple1604306 (Milestone Tech Specs); WI-Apple1604304 (Milestone Dev Specs); WI-Apple1604367 (Photon Tech Specs); WI-Apple1604365 (Photon Dev Specs);WI-Apple1604527 (Titanium Tech Specs); WI-Apple1604525 (Titanium Dev Specs); WI-Apple1604611 (Triumph Tech Specs); WI-Apple1604609 (Triumph Dev Specs); WI-Apple0147327 (Xoom Tech Specs); WI-Apple0147504 (Xoom Dev Specs); WI-Apple1604687 (XPRT Tech Specs); WI-Apple1604684 (XPRT Dev Specs) (collectively, "List of Accused Products Specifications")

Product (Code Name)	Product Type	Operating System (Reviewed)	Touch Screen	Processor	Memory (RAM)	Memory (FLASH)
Droid 2 Global (DroidWE)	Smartphone	Android v2.2.1 (DROID2WE_X6_2.5.12)	Capacitive	TI OMAP3630-1200	512 MB	8 GB
Droid 3	Smartphone	Android v2.3.4	Capacitive	TI OMAP4430-1000	512 MB	1.5 GB
Droid Pro (Venus)	Smartphone	Android v2.2.1 (VNUS2_X1_02.26.6)	Capacitive	TI OMAP3620-1000	512 MB	2 GB
Droid X (Shadow)	Smartphone	Android 2.2.1 Android v2.3.3 (SHADOW_X6_2.3.34) (DROIDX_SHADOW_06-23-11)	Capacitive	TI OMAP3630-1000	512 MB	8 GB
DroidX2 (Daytona)	Smartphone	Android v2.2.2 (4.4.1A-274_DTN-14.8)	Capacitive	NVIDIA Tegra 2 AP20H Dual Core	512 MB	2 GB
Flipout (Ruth)	Smartphone	Android v2.1 (RUTH_U3_00.28.4)	Capacitive	TI OMAP3410	512 MB	512 MB
Flipside (Sage)	Smartphone	Android v2.1 (SAGE_U3_10.23.20)	Capacitive	TI OMAP3410	512 MB	512 MB
i1 (OpusOne)	Smartphone	Android v1.5 (OPUS_ONE_AP_RBE.03.00_MS_ARM)	Capacitive	Freescale Zeus 2.0 ARM1136	256 MB	512 MB
Milestone	Smartphone	Android v2.2	Capacitive	TI OMAP3430	256 MB	512 MB
Photon	Smartphone	Android v2.3.3	Capacitive	NVIDIA Tegra 2 AP20H Dual Core	1 GB	3 GB
Spice (Sesame)	Smartphone	Android v2.1 (SESLA_U3_01.71.0)	Capacitive	Qualcomm MSM7225	256 MB	512 MB
Titanium	Smartphone	Android v2.1	Capacitive	Freescale Zeus 2.0 ARM1136	256 MB	512 MB
Triumph	Smartphone	Android v2.2	Capacitive	Qualcomm MSM8655	512 MB	2 GB
Xoom (Everest) (Hubble)	Tablet	Android v3.0.1 (MR0) Android v3.1 (XOOM_HUBBLE_WIFI_06-23-11)	Capacitive	NVIDIA Tegra 2 T20	1 GB	32 GB
XPRT	Smartphone	Android v2.2	Capacitive	TI OMAP3620-1000	512 MB	2 GB

35. I note that Motorola has not produced source code for every version of the Android OS currently available for use on each of the Motorola Accused Products. For example, I understand that certain phones have received OTA ("over the air") upgrades, such as the Atrix to version 2.3.4. OTA upgrades may be relevant to infringement to the extent that certain

functionality is added, however, in my opinion, an OTA upgrade that purportedly deleted functionality previously present on a phone would not defeat infringement, because infringement—as I understand it—can be established based on the devices as sold. I reserve the right to supplement my analysis for additional source code and other documentation as it is made available by Motorola or Google. Based on my review of the source code, documents, and materials described herein and in Appendix A, it is my opinion that any differences between the source code present on each of the different Accused Products are not material to my infringement analysis and conclusions.

36. I reserve the right to supplement this report to accuse additional products in the event that Motorola is making, using, selling or offering for sale in the United States or importing into the United States other mobile devices running the Android operating system that it has not identified during discovery.

37. I understand that there may be disputes between the parties about additional discovery to be provided. I understand that discovery in this case is continuing, and I will consider additional facts and material produced through discovery to determine whether such additional material has an impact on my opinions. I may amend or supplement this Report as necessary based on such additional information. I reserve the right to supplement my report if I receive additional information, including information about the Accused Products or their operation.

VI. LEGAL STANDARDS

38. I am not a legal expert and offer no legal opinions. However, I have been informed by counsel of the various legal standards that apply to the pertinent technical issues, and I have applied those standards in arriving at my conclusions expressed in this Report.

A. Infringement

39. I understand that to determine whether there is infringement of a patent: (1) the claims of the patent must be construed; and (2) the properly construed claims must then be compared with the Accused Products.

40. Where the construction of the asserted claims of the ’263 patent have not been proposed, I have interpreted the claims as one of ordinary skill in the art would have at the time the patent was filed.

41. As the second step in the infringement analysis, I understand that the properly construed claim must be compared to the Accused Products. I understand that an accused product may infringe a claim either literally or equivalently. If a product does not literally satisfy an element of the claim, it can still infringe under the doctrine of equivalents if it is insubstantially different from the claimed element.

42. I understand that one test for determining equivalence is to determine whether the differences between the claimed limitation and the accused product are insubstantial. I understand that another test for determining equivalence is to examine whether the accused product performs substantially the same function in substantially the same way to achieve substantially the same result as the claimed element.

43. My understanding is that a showing of inducement or contributory infringement requires proof both of direct infringement by the direct infringer as well as additional requirements to show inducement and contributory infringement.

44. I understand that to be liable as a contributory infringer, a party must import into the United States or offer to sell or sell in the United States a component of a patented machine or apparatus constituting a material part of the invention, and must know that the component is

especially made or adapted for use in infringement of the patent and is not a staple article suitable for substantial non-infringing uses.

45. I understand that whoever actively induces infringement of a patent is liable as an infringer. I understand that in order to be liable for inducement there must be direct infringement of the asserted claim. Induced infringement also requires that the infringer knew or should have known that his actions would induce actual infringement, and that the alleged infringer had specific and actual intent to cause the actions that would result in direct infringement. Although I offer my opinion as to whether certain conduct directly infringes the asserted claims of the '263 patent, I do not offer any opinion as to Motorola's knowledge or intent.

46. I understand that if the body of the claim describes a structurally complete invention such that deletion of the preamble phrase does not affect the structure or steps of the claimed invention, the preamble is generally not limiting unless there is clear reliance on the preamble during prosecution to distinguish the claimed invention from the prior art.

47. I understand that there may be disputes between the parties about additional discovery to be provided. I understand that discovery in this case is continuing, and I will consider additional facts and material produced through discovery to determine whether such additional material has an impact on my opinions. I may amend or supplement this Report as necessary based on such additional information. I reserve the right to supplement my report if I receive additional information, including information about the Accused Products or their operation. Specifically, I understand that whoever induces infringement of a patent is also liable as an infringer. I understand that in order to be liable for inducement there must be direct infringement of the asserted claim, as well as knowledge that the induced acts constitute patent infringement. I understand that contributory infringement is found where a party offers for sale,

or sells in the United States, or imports into the United States a component of a patented machine or apparatus, or offers for sale or sells in the United States, or imports into the United States a patented machine or apparatus (among other things) for use in practicing a patented method, knowing the component or apparatus to be specifically made to be used to infringe the patent and where the apparatus is not a staple article suitable for substantial non-infringing use.

48. I understand that Apple asserts that Motorola's Accused Products, which run the Android operating system, infringe the '263 patent. I have reviewed the Accused Products and documentation relating to those devices, portions of Android source code obtained from public sources, portions of Motorola's source code for the Accused Products, deposition testimony of certain Motorola employees, and interrogatory responses. I have also relied on my personal knowledge and experience in the field of computer science, as well as my review of the '263 patent, its prosecution history and cited references, and materials produced by and exchanged between the private parties. Other documents or materials that I have reviewed are listed in Appendix A. I conclude that Apple's position is correct and that the Accused Products practice every limitation of claims 1 and 2 of the '263 patent, both literally and under the doctrine of equivalents. My analysis below applies to the Accused Products.

49. I understand that Motorola and Apple exchanged proposed constructions of certain terms of the '263 patent, but those proposed constructions are still before the Court. My analysis below considers Motorola's proposed constructions and Apple's proposed constructions that were exchanged between the parties. I also understand that certain terms from the asserted claims of '263 patent were construed in ITC Investigation No. 710, particularly in the Initial Determination by Administrative Law Judge Carl C. Charneski. *See, e.g.*, Initial Determination

337-TA-710, pp. 23-44; WI-Apple1604823-844. More particularly, Administrative Law Judge Charneski construed the following terms as set forth below:

- The claim term "realtime" was construed to mean "within the defined upper bounded time limit." *Id.* at 26; WI-Apple1604826.
- The claim term "realtime signal processing subsystem" was construed to carry a plain and ordinary meaning. *Id.* at 32; WI-Apple1604832.
- The claim term "realtime API" was construed to mean "API that allows realtime interaction between two or more subsystems." *Id.* at 33; WI-Apple1604833.
- The claim term "device handler" was construed to mean "software associated with an interface device that sets up data flow paths, and also presents data and commands to a realtime processing subsystem." *Id.* at 42; WI-Apple1604842.

50. I adopt the constructions set forth in Administrative Law Judge Charneski's Initial Determination for the '263 patent.

51. [redacted] I reserve the right to supplement this report in the event that new issues regarding claim construction are raised in connection with the '263 patent.

52. In developing my opinions, I have reviewed the source code produced for inspection by Motorola. I reviewed the source code builds for each Accused Product produced by Motorola, which included the specific version of the Android operating system used on the particular phones. *See, e.g.*, MOTO-662-SOURCE 00339-643.

53. It is my opinion, based on my review of Motorola's source code, the publicly-available Google Android platform code, documents describing the Accused Products and the

code, and the testimony of witnesses, that there are no material differences in how the different Accused Products operate for purposes of my infringement analysis. I have reviewed the corresponding code in all versions of the Android operating system produced by Motorola, and this code reflects how the infringing realtime services on the Accused Motorola Android Products works in all of the Accused Products. I have printed out certain source code relevant to the functionality accused of infringing the '263 patent. Although I cannot attach it to this report because of Protective Order restrictions, it is identified as follows: MOTO-662-SOURCE 00328-643.

54. Further, due to restrictions on the number of pages of source code that can be printed, I have only printed code for a subset of the versions of Android used in the Accused Products. I have not come across any material differences between the realtime services in the code I have printed and the corresponding functionality in the other Accused Products.

55. Similarly, in developing my opinions, I have reviewed public Android platform source code provided by Google, and have compared it to the functionality in the confidential source code produced by Motorola for the Accused Products. In the course of my study, I have reviewed sections of code from Google's Android code base versions 1.5, 1.6, 2.1, 2.2, 2.2.1, 2.2.2, and 2.3.3, all accessible from http://android.git.kernel.org, or from http://source.android.com/, which is a website that is maintained by the Android Open Source Project (AOSP). *See* http://source.android.com/about/index.html [Android Developer site] at WI-Apple1684898 \0011\INDEX008.HTM. I reserve the right to use portions of this code beyond those cited specifically in my report. It is my understanding that Motorola receives initial builds of the Google Android source code from Google, and then may modify the initial build to customize it for use in Motorola products. *See* Steimle Dep. at 11:19-12:20; 13:4-24;

Boldt Dep. at 85:11-86:24. Further, based on my review of the public Android code and the Motorola-produced code, I did not identify any material differences between the public source code and the Motorola code with respect to my infringement analysis.

56. Additionally, in developing my opinions, I have reviewed and relied upon a number of Google Android Developer Documents that were produced by Motorola. *See, e.g.*, MOTO-APPLE-0002584118; MOTO-APPLE-0000335047; MOTO-APPLE-0000369211; MOTO-APPLE-0000404496, (and other materials listed in my Materials Considered).

VII. PERSONS OF ORDINARY SKILL IN THE ART

57. I may offer testimony regarding the level of ordinary skill in the art of the ’263 patent, and specifically the qualifications of such a person at the time of the inventions of the ’263 patent. In my opinion and as found by Administrative Law Judge Charneski a person of ordinary skill in the art for the ’263 patent would have a bachelor of science degree in computer science, or the equivalent, and at least two to three years experience in signal processing systems. *See* Initial Determination at WI-Apple1604825.

58. I meet the aforementioned criteria and consider myself a person to have at least ordinary skill in the art pertaining to the ’263 patent and I was such a person as of the date of invention of the ’263 patent.

VIII. RELEVANT LEGAL PRINCIPLES AND GUIDANCE

59. I am not a legal expert and offer no legal opinions. However, I have been informed by counsel of the various legal standards that apply to the pertinent technical issues, and I have applied those standards in arriving at my conclusions expressed in this Report.

A. Proposed Constructions

60. I understand that in a prior case, ITC investigation 337-TA-710, claim construction issues were extensively considered. The issues in the present case are the same as

they relate to claim construction. I am also aware that neither party in the present case has submitted terms for construction. As such I am adopting the interpretations of the claims that were used by Apple and adopted by the ITC in the 337-TA-710 investigation. I also adopt and incorporate by this reference, Administrative Law Judge Charneski's reasoning and opinions for construing the claim terms of the '263 patent as set forth at pages 23-44 of the public version of the Initial Determination at WI-Apple1604823-844. For all other terms, as mentioned above, I have applied the plain and ordinary meaning of those terms as they would have been understood by a person of ordinary skill in the art at the time of the invention. I also understand that claim construction is a matter of law and will be determined by the Judge in this Action. Should the Court adopt different constructions, I reserve the right to supplement my opinions in accordance with the court's constructions. I understand that the meaning of the term "realtime" is important to an understanding of the subject matter of the '263 patent. Administrative Law Judge Charneski construed "realtime" to mean "within a defined upper bounded time limit" in the ITC investigation. *See* Initial Determination at WI-Apple1604832. A person of ordinary skill in the art would, therefore, understand "realtime" to mean that somewhere in the system, there is a time value that is defined for some period of time that is used to bound the processing speed of the system. In the example of the Accused Products, this upper bounded time limit is the rate at which the media (audio or video) is sequenced out to the speaker or screen. This rate will changed depending on the specific media involved. Different "realtime" applications would have allowed (and still do allow) for different time limits for processing.

IX. INFRINGEMENT OF THE '263 PATENT: OVERVIEW

A. Realtime

1. Types of Realtime Systems

61. "A *realtime* system is a system that must satisfy explicit (bounded) response-time constraints or risk severe consequences, including failure." Phillip A Laplante, Real-Time Systems Design and Analysis: An Engineer's Handbook at 10 (IEEE Computer Society Press, 1st ed. 1993); WI-Apple0610122-476. "A *failed* system is a system which cannot satisfy one or more of the requirements laid out in the formal system specification." WI-Apple0610150. In a realtime system, if a defined performance deadline is missed then the value of continued performance seriously degrades. By contrast, in a non-realtime system quick performance may be desirable, but performance is not subject to any particular deadline.

62. A *hard realtime* system is one where "failure to meet response time constraints leads to system failure." WI-Apple0610150. A classic example of a hard realtime system is missile defense. Performance is subject to a hard deadline — defined by the time at which the incoming missile will hit its target. The missile defense system must perform certain calculations in order to successfully shoot down the missile. If the system fails to complete those calculations before the deadline, then continued performance is of no value — the missile has already hit.

63. A *soft realtime* system is one where "performance is degraded but not destroyed by failure to meet response time constraints." WI-Apple0610150. An example of a soft realtime system is one that processes streaming media such as audio or video. With streaming media, there is an implicit playback rate. If a deadline for delivering the next frame of audio is missed then a detectible artifact such as a click, pop, or silence may result. Such missed deadlines cause degradation in playback, but the value of continued playback is not destroyed.

64. Streaming media may be categorized as a "firm" realtime system: one with "hard deadlines where some low probability of missing a deadline can be tolerated." WI-Apple0610150. The deadline for delivering a frame of audio is a hard deadline, but missing the occasional frame doesn't lead to complete system failure; it can be tolerated as long as the system is generally reliable.

65. The *reliability* of a realtime system can be measured by "the probability that [it] will operate without failure for a specified period of time." WI-Apple0610381. A *deterministic* realtime system, for instance, allows perfect reliability — it can guarantee that it will meet every deadline (short of serious hardware failure). WI-Apple0610152 An example of a "deterministic" realtime system is a pacemaker. A pacemaker has consistent performance regardless of the input signals. By contrast, a missile defense system, while a hard realtime system, is not necessarily "reliable," and is certainly not "deterministic," because it cannot guarantee that it will shoot down every missile.

66. Related to reliability, fault tolerance "is the ability of the system to continue to function in the presence of hardware or software failures. In real-time systems, fault tolerance includes design choices that transform hard real-time deadlines into soft ones. … *Temporal* fault tolerance involves techniques that allow for tolerating missed deadlines." WI-Apple0610393.

67. Reliability is an important goal in any realtime system. WI-Apple0610152 ("the *goal* is to predict how a system will behave in all possible circumstances") (emphasis added). Indeed some hard realtime systems, such as pacemakers, have little utility if they are non-deterministic, because superior *deterministic* alternatives are widely available.

68. But while reliability is an important *goal* in a realtime system, no particular degree of reliability is *required* in order for a system to be considered realtime. The suggestion

that a realtime system must guarantee timely performance is negated by the very concept of *fault-tolerant* realtime systems. *See, e.g.*, WI-Apple0610393-403. A fault-tolerant realtime system knows that it cannot guarantee performance in all cases, but nonetheless aims to tolerate faults. *Id.* Indeed determinism is impossible to guarantee for any but the simplest realtime systems. As Laplante's Real-Time Systems treatise explains, "cache, pipelines, and DMA (all designed to improve average real-time performance) destroy determinism and thus make prediction of real-time performance essentially impossible." *See* Real-Time Systems 2nd ed. at WI-Apple0579511. The essence of a realtime system is that late performance has severe consequences — not that late performance is guaranteed never to occur. *See* Real-Time Systems 1st ed. at WI-Apple0610149.

2. Realtime "Streaming"

69. In the context of realtime processing of data streams, a realtime system must process each "frame" of streaming data within a short period of time. Specifically, suppose that a mobile phone user wishes to stream music from the Internet, so clicks "play" on an online music file. The first frame of music in that file is not subject to any particular processing deadline, but it should received, processed, and output within some reasonably responsive time period. Every subsequent frame of music, however, is subject to a deadline: it must be processed and ready for playback no later than the moment the immediately preceding frame finishes playback.

70. Every realtime system must have some latency between when the user presses play and when playback begins. In part, this delay is a necessary implication of physics; a user's command can only travel at a finite speed (the speed of light). In addition, some small delay is often built into a realtime system to smoothly handle network jitters.

71. A "progressive download" system is one where playback begins after a portion of the file has downloaded. A progressive download system may be realtime, if playback begins

after a short latency period and keeps going until the file ends; or non-realtime, if playback begins after a substantial portion of the file has downloaded in order to mask a download or processing speed that is too slow, too unreliable, or both. For example, in the case of video streaming services like YouTube, the system is realtime only if the network bandwidth and processing power allows for the file to be downloaded and processed reliably faster than the video playback rate. In the realtime case, the video will start after an initial short latency, and keep playing without pause until the video ends. Thus in certain circumstances whether a system is realtime can depend on real world performance metrics.

B. OpenMAX IL API

72. Motorola's Accused Products utilize the OpenMAX Integrated Layer (or "IL") API. *See*, *e.g.*, MOTO-662-SOURCE 0481-520. The OpenMAX IL components are used to create DSP independent processing components. This allows systems built using OpenMAX IL flexibility in what DSP hardware they use. According to the online description of OpenMAX IL: "The principal goal of the IL is to give codecs a degree of system abstraction using a specialized arsenal of features, honed to combat the problem of portability among many vastly different media systems." (OpenMAX - The Standard for Media Library).[2] Per OpenMAX documentation: "The OpenMAX IL API strives to give media components portability across an array of platforms. The interface abstracts the hardware and software architecture in the system." OpenMAX Integration Layer Application Programming Interface Specification Version 1.1.2, at WI-Apple0576950. This document further states: "The OpenMAX IL API gives applications and media frameworks the ability to interface with multimedia codecs and supporting components (i.e., sources and sinks) in a unified manner. … Without a standardized interface of

[2] *Available at* http://www.khronos.org/openmax/.

this nature, component vendors have little alternative than to write to proprietary or closed interfaces to integrate into mobile devices." *Id*. at WI-Apple0576950-51. Moreover:

> One of the goals of OpenMAX IL is hardware independence provided by the IL layer to the layers above it. The goal of hardware independence can be achieved by specifying the following requirements regarding resource management:
>
> - An IL client (e.g., a multimedia plug-in that is typically part of a software platform) should not need to know the details of an IL implementation or which resource an IL component is using.
> - In case of resource conflicts, an IL client should be able to rely on consistent component behavior across IL implementations and hardware platforms.
> - An IL client should not have to interface directly with a hardware vendor-specific resource manager for two reasons.
>   - This method violates the goal of hardware independence.
>   - This method adds considerable re-work to the IL client, which has an impact on the re-usability of the IL client on multiple hardware platforms.

Id. at WI-Apple0576978.

73. OpenMAX IL consists of two parts, an OMX Core and individual OMX Components. *See, e.g.*, WI-Apple0576953. The OMX Core enumerates codecs available on a phone. *Id.* OpenMAX components implement codecs. WI-Apple0577080-135. A given phone can have multiple Cores provided by different manufacturers. WI-Apple1695883-905.

74. At a high level, clients of the OpenMAX IL API (*e.g.*, components or nodes that need to use audio or video decoding services) ask the OMX Core for a particular OMX Component by providing a name. For example, a client can ask the OMX Core for the Qualcomm video decoder for the AVC file format by asking for a component with the name, "OMX.qcom.video.decoder.avc." The client then sends commands to this Component using the OpenMAX IL API, such as the following:

3.1.1.1 OMX_COMMANDTYPE

Table 3-1 represents the possible commands that an IL client can send to an OpenMAX IL component. Since commands are non-blocking, the OpenMAX IL component generates a command completion event via a callback function when the command has completed. Callbacks are defined in a dedicated structure; see section 3.1.2.8.

Table 3-1: OpenMAX IL Commands

Field Name	Description
`OMX_CommandStateSet`	Change the component state
`OMX_CommandFlush`	Flush the queue(s) of buffers on a port of a component
`OMX_CommandPortDisable`	Disable a port on a component
`OMX_CommandPortEnable`	Enable a port on a component
`OMX_CommandMarkBuffer`	Mark a buffer and specify which other component will raise the event mark received

OpenMAX Integration Layer Application Programming Interface Specification at WI-Apple0576986.

75. Certain commands contain further sub-enumerations. For example, the `OMX_CommandStateSet` contains the following state enumerations:

This section describes component states. An IL client commands a component to change states via the `OMX_SendCommand` function using the `OMX_CommandStateSet` command.

Table 3-3 represents the states of an OpenMAX IL component.

Table 3-3: OpenMAX IL Component States

Field Name	Description	Static Resources Allocated	Location of buffer
OMX_StateInvalid	Component is corrupt or has encountered an error from which it cannot recover.	Unknown	Unknown
OMX_StateLoaded	Component has been loaded but has no resources allocated.	No	Not available
OMX_StateIdle	Component has all resources but has not transferred any buffers or begun processing data.	Yes	Supplier only
OMX_StateExecuting	Component is transferring buffers and is processing data (if data is available).	Yes	Supplier or non-supplier
OMX_StatePause	Component data processing has been paused but may be resumed from the point it was paused.	Yes	Supplier or non-supplier
OMX_StateWaitFor Resources	Component is waiting for a resource to become available.	No	Not available

Id. at WI-Apple0576988.

76. As an example usage, on a Motorola device having a Qualcomm video decoder, if a client of the OpenMAX IL API wants to decode an AVC file, they would first ask the OMX Core for a handle to the component with the name "OMX.qcom.video.decoder.avc." The client could then ask the OMX Component to begin decoding by invoking `OMX_SencCommand(handle, CommandStateSet, OMX_StateExecuting)`. This use case can be visually depicted, in Figure 1, as follows:

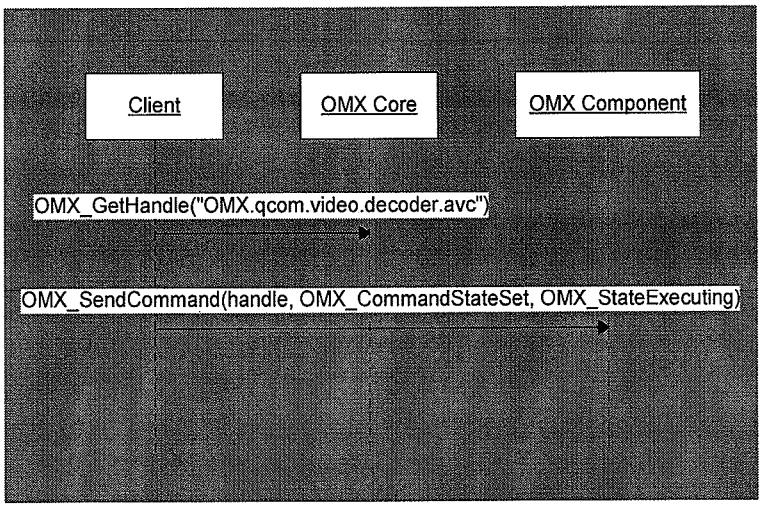


Figure 1: *See, e.g.*, OpenMax Call Sequences at WI-Apple0593216-239.

X. INFRINGEMENT OF THE '263 PATENT BY MOTOROLA'S ANDROID PHONES[3]

A. Overview

77. At a high level, multimedia applications on the Accused Products use Android's Media Server for eventual output to the relevant "Flinger" – the SurfaceFlinger for display or the AudioFlinger for audio. Noisy Androids Mastering the Android Media Framework at WI-Apple0575181-236. "Flinger" is a word often used in Android-related documentation to refer to

[3] For each of the claims and elements discussed below, to the extent I have not specifically addressed a particular model of the Accused Product, it is my opinion, based on the totality of materials I have reviewed and the similarities among the models and the source code for them, that my analysis applies to any models not specifically addressed.

resources that present media to the user. *Id.* The distinction is made between the SurfaceFlinger and the AudioFlinger to distinguish between display media and audio media. *Id.* The use of the multimedia applications on the Accused Products is illustrated below in Figure 1. An example of such an application includes the YouTube application, which is installed on each of the Accused Products. [4]

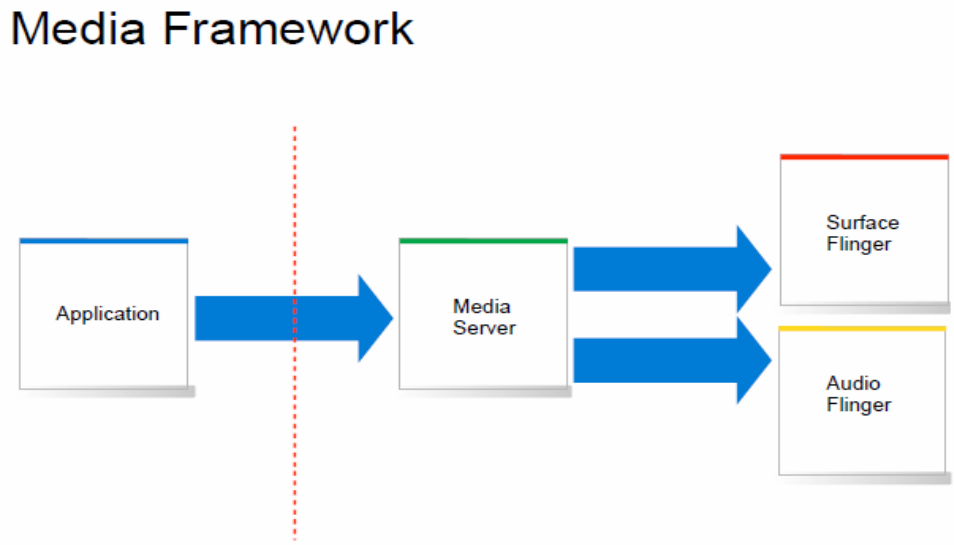


Figure 2: Mastering the Android Media Framework at WI-Apple 0575186.

[4] User Guides and Manuals for the Motorola Accused Products, including but not limited to WI-Apple0147233 (Atrix User Guides); WI-Apple0033229 (Backflip User Guides); WI-Apple0033363 (Bravo User Guides); WI-Apple0032798 (Charm User Guides); WI-Apple0033492 (Citrus User Guides); WI-Apple0033554(Cliq 2 User Guides), WI-Apple0033880(CliqXT User Guides); WI-Apple0033295 (Defy User Guides); WI-Apple0032930 (Devour User Guides); WI-Apple0032860 (Droid 2 User Guides); WI-Apple0147575 (Droid 2 Global User Guides); WI-Apple1604165 (Droid 3 User Guides); WI-Apple0034020 (Droid User Guides); WI-Apple0033950 (Droid Pro User Guides); WI-Apple1604240 (Droid X2 User Guides); WI-Apple0034078 (Droid X User Guides); WI-Apple0033624 (Flipout User Guides); WI-Apple0033688 (Flipside User Guides); WI-Apple0033429 (i1 User Guides); WI-Apple1604309 (Milestone User Guides); WI-Apple1604370 (Photon User Guides); WI-Apple1604531 (Titanium User Guides); WI-Apple1604616 (Triumph User Guides); WI-Apple0147653 (Xoom User Guides); WI-Apple1604692 (XPRT User Guides) (collectively, "List of Accused Products User Guides").

78. Depending on the file type, the Media Server utilizes different virtual media players. Examples of such virtual media player include PVPlayer and the Stagefright Player. Figure 2 below illustrates, at a high level, how Media Server uses such media players. In Figure 2, PVPlayer is part of OpenCORE.

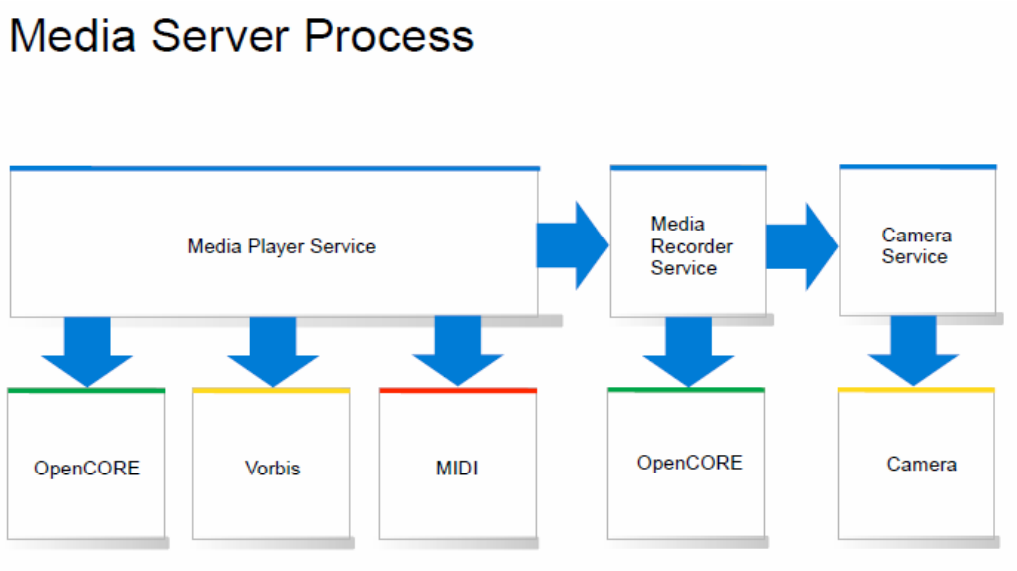


Figure 2: Mastering the Android Media Framework at WI-Apple0575188.

79. The system selects the appropriate player based on the file extension specified by the URL the user is trying to play. In order to handle the particular media files, PVPlayer and Stagefright Player set up data flow paths using nodes. These nodes then call the OpenMAX IL API for processing services. The details of this are addressed below.

B. Infringement of Claim 1 of the '263 Patent

80. As an exemplary Motorola Accused Product running the Android OS, I examined a Motorola Droid 2 device, which runs version 2.2 of the Android OS. I also examined a Motorola Xoom device, which runs version 3.0 of the Android OS. Portions of my analysis discuss the Motorola Droid 2 running version 2.2 of the Android OS for ease of reference. My focus on the Droid 2 is not an express or implicit exclusion of any other Motorola product or

Android OS version. The implementation of the relevant source code in the Android OS on the Motorola Droid 2 is, for all relevant purposes, substantially the same as the implementation in the other Motorola Accused Products. I have reviewed both Motorola produced code and Motorola open source code for the Accused Products beyond the excerpts that I printed, and these excerpts are representative of the realtime services functionality in any of the Accused Products, described below, and are intended to be exemplary. I have not come across any material differences between the realtime services functionality in the code I have printed and the corresponding functionality in the other Accused Products.[5]

1. "A signal processing system for providing a plurality of realtime services to and from a number of independent client applications and devices, said system comprising:"

81. It is my understanding that Motorola has not argued that the preamble of claim 1 is limiting. In my opinion, the preamble of claim 1 is not limiting because the claim body defines a structurally complete invention and because the applicants did not rely on the preamble during prosecution to distinguish the claimed invention from the prior art.

82. To the extent that this preamble is found to be a limitation, it is my opinion is that the Accused Products practice claim 1 even if limited by its preamble. For the reasons discussed below, the Accused Products include a signal processing system for providing a plurality of

[5] I also note that the consistency between the various generations of the Android OS is expected, because new versions of the code will generally need to maintain compatibility with earlier versions in order to maintain compatibility between the core system applications, new applications, and newer and older versions of the Android OS. This same consistency is also expected between the public versions of the Android OS and Motorola's proprietary versions thereof, and I have indeed observed that consistency in my source code review. In particular, while Motorola may seek to add functionality to differentiate its products from the Android products sold by other manufacturers, such changes would be made with a mind to maintain compatibility between the core system applications, new applications, and the same newer and older versions of the Android OS. In other words, Motorola would be unlikely to change its version of Android in such a manner to make it incompatible with the expected functionality of the Android system as a whole.

realtime services to and from a number of independent client applications and devices. *See supra* Table 1.

83. To the extent Motorola contends that the Accused Products do not "provid[e] a plurality of realtime services," it is my opinion that Motorola nonetheless infringes under the doctrine of equivalents because the claimed and systems of the Accused Products are insubstantially different. As described at length elsewhere in this report, to the extent that the Accused Products could be argued as not "realtime," they are designed to be imperceptibly different from a "realtime" system, with processing capacity exceeding or approaching the rate of the multimedia stream, such that they provide seamless presentation of multimedia data to the end-user. In that way, the Accused Products perform substantially the same function, in substantially the same way, to achieve substantially the same result, as a "realtime" system.

2. "(a) a subsystem comprising a host central processing unit (CPU) operating in accordance with at least one application program and a device handler program, said subsystem further comprising an adapter subsystem interoperating with said host CPU and said device;"

a. "subsystem comprising a host central processing unit (CPU)"

84. The Accused Motorola Android Products each include a subsystem comprising a host central processing unit ("CPU"). As described in publicly available literature and summarized in Table 1, each of the Accused Products include a CPU. *See, e.g.*, *supra* Table 1; List of Accused Products Developers Specifications at *supra* n.1. For example, the Droid 2 uses a TI OMAP3630-1200 CPU. *See, e.g., id.*; Droid 2 Developer Specification at WI-Apple0147222.

85. To the extent Motorola contends that the Motorola Accused Android Products do not include "a subsystem comprising a host central processing unit (CPU)," Motorola nonetheless infringes under the doctrine of equivalents because the claimed and accused systems

are insubstantially different. For example, the Accused Products include one or more processors that are primarily responsible for the execution of application programs. In this respect, these processor(s) in the Accused Products perform substantially the same function, in substantially the same way, to achieve substantially the same result as a "host CPU."

b. "operating in accordance with at least one application program and a device handler program"

86. A person of ordinary skill in the art would understand "device handler" to mean software associated with an interface device that sets up data flow paths, and also presents data and commands to a realtime processing subsystem. As shown above, this construction is identical to Administrative Law Judge Charneski's construction of "device handler" in ITC Investigation No. 710. *See, e.g.*, Initial Determination at WI-Apple1604823-44.

87. The CPU in the Accused Motorola Android Products operates in accordance with at least one application program and a device handler program.

88. I understand the parties disagree about the meaning of "device handler." Apple has proposed that "device handler" means "software associated with an interface device that sets up data flow paths, and also presents data and commands from the data managers to a real-time data processing engine." *See* June 3, 2011 Apple's Proposed Claim Construction. Apple's construction is identical to the construction adopted by Administrative Law Judge Charneski's construction in ITC Investigation No. 710. *See e.g.*; WI-Apple1604842. Motorola has proposed that "device handler" means "software associated with an interface device that sets up data paths and also presents data and commands to a realtime signal processing subsystem." *See* June 3, 2011 Motorola Proposed Constructions. I do not believe that there is any meaningful difference between these two proposed constructions and it is my opinion that Motorola infringes under both constructions. Should the court adopt a different construction, my opinion may not

materially change depending on the construction adopted and information that may be obtained during ongoing discovery, and I reserve the right to supplement my opinions accordingly.

89. Each of the Accused Products includes one or more application programs installed by default that make use of various types of media files. The Accused Products include application programs such as the YouTube application, a music player, and the browser each of which make use of the OpenMax IL API and operate in accordance with the CPU. All accused products have the YouTube application and music player installed by default. *See*, *e.g.*, List of Accused Products Developers Specifications at *supra* n.1.

90. In the Accused Products, the PVPlayer implementation is a device handler program. Some of the Accused Products (specifically, those with Android version 2.2 and later) also have a Stagefright Player that can use the OpenMAX IL API. The Stagefright Player implementation is also a device handler program.

91. The PVPlayer implementation in the Accused Products consists of the PVPlayer, the PlayerDriver and the PVPlayerEngine – collectively a PVPlayer implementation. [At least a portion of the source code for these is found in playerdriver.h, playerdriver.cpp, pv_player_engine.cpp] Additional code related to the media player is found in mediaplayer.cpp. I have specifically printed some of this code but have examined all of it as part of my investigation.

92. The PVPlayer implementation exists in each of the Accused Products. That is, each of the Accused Products includes OpenCORE. OpenCORE within Android includes the PacketVideo Player Engine.

93. The PVPlayer implementation is associated with an interface device such as the cellular or Wi-Fi antenna. The PVPlayer implementation also sets up data flow paths which, for

example, are a sequences of nodes that go from the interface device (*e.g.*, antenna, hard drive, or Flash) through coders and decoders, all the way to the relevant output device (or Flinger) such as the screen or speaker. The PVPlayer inherits from MediaPlayerInterface [playerdriver.h]. As a result, the PVPlayer has two methods for setting up the datasource in the datapath: (1) setDataSource(int fd, int64_t offset, int64_t length) which is used for local file playback and (2) setDataSource(const char *url), which is used for playing URLs. These methods are specific to a source device (*e.g.*, antenna, hard drive, or Flash). The local file (hard drive or Flash memory) playback method is associated with the local hard drive (or Flash), and the URL playback method is associated with playback from over a network interface (*e.g.*, antenna).

94. While creating the source node (the first node in the path), the code within pv player engine.cpp sets the source, sink, and type of the noted and then issues the appropriate call to create the node. These nodes are referred to as PVMF nodes. Examples of such nodes include those defined in the files pvmf_omx_videodec_node.h and pvmf_omx_audiodec_node.h. These nodes, for example perform video or audio decoding.

95. Specifically, when the source for playback is a URL, the PVPlayerDataSourceURL's data source format type is PVMF_MIME_DATA_SOURCE_HTTP_URL. [playerdriver.cpp] If the source is a local file, the source format type is PVMF_MIME_FORMAT_UNKNOWN. The PVPlayerNodeRegistry creates a PVMFDownloadManagerNode node. In contrast if the source is a file, the PVPlayerNodeRegistry creates the relevant parser node, for example a PVMFMP3FFParserNode for mp3 files. The PVPlayer implementation loads different source nodes depending on the source device. Moreover, the PVMFMP3FFParserNode will be used for both URLs and local files, but the code performs different kinds of processing depending on whether playback is local

or not. [pvmf_mp3ffparser_node.cpp] Thus the PVPlayer has device specific code, that is, it has "software associated with [] interface device[s]."

96. The PacketVideo Player Engine utilizes MIO components that are "customized for a specific target platform and hardware" Media IO Developer's Guide OpenCORE at WI-Apple0593246. Per PacketVideo's documentation: "In the PV multimedia framework (PVMF) architecture, the Media I/O (MIO) component is a data sink or source at either the beginning or end of the datapath for media data" *Id.*

97. As such, each of the Accused Products has "a subsystem comprising a host central processing unit (CPU) operating in accordance with at least one application program and a device handler program."

98. The Stagefright player implementation consists of the StagefrightPlayer and the AwesomePlayer. [StagefrightPlayer.cpp and Awesomeplayer.cpp] The Accused Motorola Android products running Android 2.2 and above use the Stagefright player implementation and it is my opinion that the Stagefright player implementation also constitutes a device handler program. I have reviewed the Motorola produced Android code for Android versions 2.2 and above and found the Stagefright player implementation in each of them including, at least, the two files listed in this paragraph and OMXCODEC.cpp.

99. The Stagefright player implementation is associated with an interface device such as the cellular or WiFi antenna. Similar to the PVPlayer implementation, the Stagefright player implementation also inherits from the MediaPlayerInterface [StagefrightPlayer.h] As a result, the Stagefright player implementation also contains separate setDataSource() methods" setDataSource(init fd, int64_t offset, int64_t length), which is used for local file playback, and setDataSource(const char *url), which is used for playing URLs. These methods are specific to a

source device. For example, the local file playback method uses a FileSource object to access local files. [awesomeplayer.cpp] In contrast, the URL playback method uses an HTTPDataSource onject, which is specific to playback from over a network interface.

100. In similar fashion to the PVPlayer implementation, the Stagefright player implementation is such that a higher level module – the Stagefright player (akin to Media Player) – calls to another "player" – the Awesome Player (akin to Player Driver in that each has knowledge of whether the realtime service is for local or URL). Awesome Player, like PVPlayer sets up data flow paths. Awesome Player then calls OMX Codec which serves the same purpose as the audio and video PVMF Nodes in PVPlayer. Accordingly, the Stagefright player implementation is also a device handler program and for this additional reason, Accused Products running Android version 2.2 or above meet this limitation.

101. There is little to no available documentation of the functioning of the Stagefright and Awesome players. My analysis is from a detailed examination of the code provided.

102. To the extent Motorola contends that the Accused Products do not include a subsystem or host CPU "operating in accordance with at least one application program," it is my opinion that Motorola nonetheless infringes under the doctrine of equivalents because the claimed and accused systems are insubstantially different. For example, the Accused Products include one or more software modules that provide the functionality typical of an application program and that execute on the CPU in the Accused Products.

103. To the extent Motorola contends that the Accused Products do not include a "device handler program," it is my opinion that Motorola nonetheless infringes under the doctrine of equivalents because the claimed and accused systems are insubstantially different. To the extent that the accused "device handler program" is not "specific to a device," it is performs

substantially the same function, in substantially the same way, to achieve substantially the same result as a structure that is "specific to a device" because it is "associated with an interface device," as I explained elsewhere, and it interacts with the device to achieve the result of presenting information to the realtime signal processing subsystem for processing. To the extent that Motorola asserts that the accused "device handler program" does not "set up data flow paths," it is insubstantially different from a structure that does, because it sets up at least a substantial portion of the data flow path, such that data is presented to the realtime signal processing subsystem. To the extent that Motorola asserts that the accused "device handler program" does not "present[] data and commands to a realtime processing system," it is insubstantially different from a structure that does, because it presents information to the realtime processing subsystem for processing, along with other information about the information to be processed.

104. To the extent Motorola contends that the Motorola Accused Android Products do not include a subsystem or host CPU "operating in accordance with ... a device handler program," it is my opinion that Motorola nonetheless infringes under the doctrine of equivalents, for the reasons explained above with respect to the term "device handler program."

c. "said subsystem further comprising an adapter subsystem interoperating with said host CPU and said device"

105. In my opinion, the Accused Products' subsystem further comprises an adapter subsystem interoperating with said host CPU and said device. The Accused Products include an adapter subsystem that interoperates with the ARM processor of the CPU (*e.g.*, the TI OMAP3620-1000) and devices. By way of one example, the Droid 2 includes an SDIO interface and driver software that interoperates with the ARM processor and the WiFi modem unit. *See*, *e.g.*, TI OMAP 3X Product Bulletin at WI-Apple0149673-75; Android Open Source: WiFi at

WI-Apple0149679-81; *supra* Table 1. By way of another example, Droid 2 also includes a USB interface and multichannel buffered serial port (McBSP) interface and driver software that interoperates with the ARM processor and the 3G/4G modem unit. *See, e.g.*, TI OMAP 3X Product Bulletin at WI-Apple0149673-75; *supra* Table 1.

106. To the extent Motorola contends that the Motorola Accused Android Products do not include "an adapter subsystem interoperating with said host CPU and said device," it is my opinion that Motorola nonetheless infringes under the doctrine of equivalents because the claimed and accused systems are insubstantially different. For example, the Accused Products include hardware and optionally software that provides an interface to one or more networks, including wide area networks; allows the Accused Products to transmit data via those network(s); and interacts with the CPU and one or more devices.

3. "(b) a realtime signal processing subsystem for performing a plurality of data transforms comprising a plurality of realtime signal processing operations; and"

107. In my opinion, the Accused Products include a "realtime signal processing subsystem for performing a plurality of data transforms comprising a plurality of realtime signal processing operations."

108. Per the PVPlayer documentation: [PVPlayer SDK Developer's Guide include section 6.3 "Faster or slower than real-time"]

109. By way of example, the Droid 2 uses a hardware accelerator, the IVA 2, for performing a plurality of data transforms that comprise a plurality of realtime audio and video processing operations. *See, e.g.*, TI OMAP 3X Product Bulletin at WI-Apple0149673-75; *supra* Table 1. More specifically, PVPlayer sets up dataflow paths that consist of PVMF nodes. One of those nodes, in the case of video, is PVMF_OMX_videodec_node. That node will in turn call, via Open MAX IL, an MPEG decoding OMX component that resides in the hardware

accelerator. Similarly, in the case of audio, PVPlayer sets up a PVMF_OMX_audiodec.mode. That node will in turn call, via OpenMAX IL, an AAC decoding OMX component, for example, that resides in the hardware accelerator. The Stagefright player operates in a substantially similar way.

110. To the extent Motorola contends that the Accused Products do not include "a realtime signal processing subsystem," it is my opinion that Motorola nonetheless infringes under the doctrine of equivalents because the claimed and accused systems are insubstantially different. As described at length elsewhere in this report, to the extent that the accused systems could be argued as not "realtime," they are designed to be imperceptibly different from a "realtime" system, with processing capacity exceeding or approaching the rate of the multimedia stream, such that they provide seamless presentation of multimedia data to the end-user. In that way, the Accused Products perform substantially the same function, in substantially the same way, to achieve substantially the same result, as a "realtime" system.

a. for performing a plurality of data transforms comprising a plurality of realtime signal processing operations; and"

111. A person of ordinary skill in the art would understand the realtime subsystem to perform a "plurality of data transforms comprising a plurality of realtime signal processing operations," because the realtime subsystem executes a plurality of encoding and decoding operations, some DSP hardware-accelerated, that process audio and video signals in realtime.

112. The Accused Products satisfy this element because a plurality of the hardware-accelerated codecs (as described above in Section X.3) are designed to and in fact do perform computations on an isochronous data stream.

113. To the extent Motorola contends that the Accused Products do not include a realtime signal processing subsystem "for performing a plurality of data transforms comprising a

plurality of realtime signal processing operations," it is my opinion that Motorola nonetheless infringes under the doctrine of equivalents because the claimed and accused systems are insubstantially different. As described at length elsewhere in this report, to the extent that the accused systems could be argued as not "realtime," they are designed to be imperceptibly different from a "realtime" system, with processing capacity exceeding or approaching the rate of the multimedia stream, such that they provide seamless presentation of multimedia data to the end-user. In that way, the Accused Products perform substantially the same function, in substantially the same way, to achieve substantially the same result, as a "realtime" system.

4. "(c) at least one realtime application program interface (API) coupled between the subsystem and the realtime signal processing subsystem to allow the subsystem to interoperate with said realtime services."

114. A person of ordinary skill in the art would understand "realtime services" to mean data handling in realtime. I understand that the parties disagree about the meaning of "realtime services." Apple has proposed "realtime services" means "data handling in realtime." June 3, 2011 Apple's Proposed Constructions. Motorola has proposed "realtime services" means "constant bit rate data handling in realtime." June 3, 2011 Motorola's Proposed Constructions. Nevertheless, it is my opinion that Motorola infringes under both constructions. Should the court adopt a different construction, my opinion may not materially change depending on the construction adopted and information that may be obtained during ongoing discovery, and I reserve the right to supplement my opinions accordingly.

115. The PVMF Node API, which is a part of the PVPlayer implementation, is a realtime API in the context of the '263 patent, because it is an "API that allows realtime interaction between two or more subsystems." *See* Initial Determination at WI-Apple1604833. Specifically, it is an interface that abstracts underlying audio/video encoding and decoding functionality. Per PacketVideo's documentation: "There are several ways to integrate a codec

into the PVOpenCORE multimedia framework including as a compressed media I/O component, as a node, and an OpenMAX component integrated into the OpenMAX codecs nodes that are part of the framework. Many codecs, especially those that include hardware acceleration, implement the OpenMAX IL interface making the OpenMAX interface the most straightforward method of integration in those cases." *See, e.g.*, OMX Core Integration Guide at WI-Apple1695883-906. Thus it is my opinion that the PVMF Node API constitutes the realtime API.

116. The PVMF Node API is an API that calls the OpenMAX IL API. As such, the PVMF Node API is coupled between the subsystem and the realtime signal processing subsystem. This arrangement allows the subsystem to interoperate with the realtime services.

117. Similarly, the Stagefright player implementation meets this element. The Stagefright player implementation includes a realtime API as described in this element in the OMX Codec portion of the implementation because it is an API that allows realtime interaction between two or more subsystems – for example, it allows for abstracting audio and video coding and decoding functionality in a manner substantially similar to the PVPlayer's PVMF Node API.

118. To the extent Motorola contends that the Motorola Accused Android Products do not include "at least one realtime application program interface (API) coupled between the subsystem and the realtime signal processing subsystem to allow the subsystem to interoperate with said realtime services," it is my opinion that Motorola nonetheless infringes under the doctrine of equivalents because the claimed and accused systems are insubstantially different. As described at length elsewhere in this report, to the extent that the accused systems could be argued as not "realtime," they are designed to be imperceptibly different from a "realtime" system, with processing capacity exceeding or approaching the rate of the multimedia stream,

such that they provide seamless presentation of multimedia data to the end-user. In that way, the Accused Products perform substantially the same function, in substantially the same way, to achieve substantially the same result, as a "realtime" system.

C. Infringement of Claim 2 of the '263 Patent

1. "The signal processing system as set forth in claim 1, wherein said signal processing system receives and transmits a plurality of datatypes over a plurality of different wide area networks (WANs)."

119. The Accused Products receive and transmit data over a plurality of different wide area networks (WANs), such as cellular and WiFi. *See*, *e.g.*, *supra* Table 1; List of Accused Products Developers Specifications at *supra* n.1.

120. To the extent Motorola contends that transmitting and receiving and receiving data signals over cellular data and Wi-Fi networks does not constitute transmitting and receiving data over "a plurality of different" WANs, I disagree. In my opinion different connection types, such as cellular data or Wi-Fi, comprise a plurality of different WANs.

121. The Accused Motorola Android Products receive and transmit a plurality of different audio and video data types. For example, MPEG4 and H.263 at least are different data types. [perhaps a reference showing the phones use MPEG4 and h.263] Further: [table2 from OpenCORE Multimedia framework capabilities] shows that different media types are supported.

122. To the extent Motorola contends that, in the Motorola Accused Android Products, the signal processing system does not "receive[] and transmit[] a plurality of datatypes over a plurality of different wide area networks (WANs)," it is my opinion that Motorola nonetheless infringes under the doctrine of equivalents because the claimed and accused systems are insubstantially different. To the extent Motorola contends that the various multimedia formats used by the Accused Products are not "datatypes," they would be understood by a person of ordinary skill to be equivalent to "datatypes," as they represent different types of techniques for

encoding and decoding data. To the extent Motorola contends that the various cellular and WLAN standards used by the Accused Products to transmit data are not a "plurality of different WANs," they would be understood by a person of ordinary skill to be equivalent, as they represent distinct, separately defined standards for accessing networks with broad geographic coverage (the Internet, in the case of Wi-Fi/WLAN transmission).

D. Induced And Contributory Infringement

123. **Contributory Infringement:** In my opinion, the sale, offer to sell and importation of the Accused Products is an act of contributory infringement of claims 1 and 2 of the '263 patent because the realtime services functionality component described above, as present in the Accused Products, has no substantial non-infringing use. Specifically, the realtime services functionality component in the Accused Products (which is utilized by pre-installed applications such as the YouTube application, music player application, and browser) is fundamental to the use of realtime services on the Accused Products. Further, the realtime services functionality component described above is a material part of the patented apparatus of claims 1 and 2 that is specifically made and adapted (or designed) for use in an infringing manner. Moreover, the Accused Products are and must be known to be especially adapted for infringement of the '263 patent, as using the realtime services functionality component on the Accused Products, is one of the natural and ordinary purposes for such a device. Therefore, the realtime services functionality component described above in the Accused Products has no substantial non-infringing use other than to be used in an infringing manner.

124. **Inducement:** In my opinion, Motorola actively induces infringement of claims 1 and 2 of the '263 patent by the users of the Accused Products. As discussed above with respect to contributory infringement, the Accused Products as sold are devices that utilize a realtime services functionality component. In addition, the User Guides provided by Motorola to

purchasers of the Accused Products provide instructions on how to use the Accused Products in a manner, such as viewing videos using the installed YouTube application and listening to music using the music player application, that infringes claims 1 and 2 as discussed above. *See, e.g.*, List of Accused Products User Guides at supra n.4.

XI. RESERVATION OF THE RIGHT TO SUPPLEMENT

125. I reserve the right to supplement my report after the patent terms are construed. I further reserve the right to supplement my opinions if or when Respondents clarify their noninfringement positions. Also, I reserve the right to supplement my opinions if or when Respondents provide source code, documents, and other evidence for additional Accused Products. Finally, I reserve the right to create demonstrative exhibits, summary charts, and the like that may be useful in presenting my opinions in further proceedings in this case.

XII. CONCLUSION

126. I understand that there is still time remaining for fact discovery, and that there are current outstanding discovery issues relating to Motorola's and third-party document production. I understand there are also still depositions remaining to be completed. I therefore reserve the right to supplement my report should additional information be produced that is relevant to the infringement and domestic industry issues discussed in this Report.

127. In connection with my anticipated testimony in this action, I may use as exhibits various documents produced in this case that refer or relate to the matters discussed in this report. I have not yet selected the particular exhibits that might be used. In addition, I may create or assist in the creation of certain demonstrative exhibits to assist in the presentation of my testimony and opinions as described herein or to summarize the same or information cited in this report. Again, those exhibits have not yet been created.

Dated: September 15, 2011

Nathaniel Polish