

**United States Court of Appeals
for the Federal Circuit**

GEMALTO S.A.,
Plaintiff-Appellant,

v.

**HTC CORPORATION, HTC AMERICA, INC.,
EXEDEA, INC., GOOGLE, INC., MOTOROLA
MOBILITY, LLC (also known as Motorola Mobility,
Inc.), SAMSUNG ELECTRONICS CO., LTD. AND
SAMSUNG TELECOMMUNICATIONS AMERICA,
LLC,**
Defendants-Appellees.

2013-1397

Appeal from the United States District Court for the
Eastern District of Texas in No. 10-CV-0561, Judge
Leonard Davis.

Decided: June 19, 2014

JOHN M. WHEALAN, of Chevy Chase, Maryland, ar-
gued for plaintiff-appellant. With him on the brief were
DIRK D. THOMAS, McKool Smith, P.C., of Washington, DC;
ROBERT A. COTE, of New York, New York; and JOEL L.
THOLLANDER, of Austin, Texas.

DAVID A. PERLSON, Quinn Emanuel Urquhart & Sullivan, LLP, of San Francisco, California, argued for defendants-appellees. With him on the brief were CHARLES K. VERHOEVEN and ANTONIO R. SISTOS. Of counsel were KRISTIN J. MADIGAN, of San Francisco, California, JOSEPH MILOWIC, III, and ROBERT B. WILSON, of New York, New York.

Before NEWMAN, RADER,* and DYK, *Circuit Judges*.

DYK, *Circuit Judge*.

Gemalto S.A. (“Gemalto”) is the owner of U.S. Patent Nos. 6,308,317 (“the ’317 patent”), 7,117,485 (“the ’485 patent”), and 7,818,727 (“the ’727 patent”). Gemalto sued HTC Corporation, HTC America, Inc., Exedea, Inc., Google, Inc., Motorola Mobility, LLC, Samsung Electronics Co., Ltd., and Samsung Telecommunications America, LLC (collectively, “defendants”) in the Eastern District of Texas, alleging infringement of various claims of the three patents. The district court construed the asserted claims and granted summary judgment of non-infringement, concluding that the accused products did not infringe literally or under the doctrine of equivalents. On appeal, Gemalto challenges the district court’s claim construction and its grant of summary judgment of non-infringement under the doctrine of equivalents. We affirm.

BACKGROUND

The ’317 patent, the ’485 patent, and the ’727 patent derive priority from the same provisional patent applica-

* Randall R. Rader vacated the position of Chief Judge on May 30, 2014.

tion,¹ and share the same named inventors, specification, and title: “Using a High Level Programming Language with a Microcontroller.”² The patented technology is designed to allow resource-constrained devices, including microcontrollers, to run software applications (or programs) written in high level programming languages, such as Java.

Before Gemalto’s invention, microprocessor-based personal computers could run Java applications. At the time of Gemalto’s invention, these computers used processors that required substantial amounts of memory, which was located on chips separate from the chip containing the processor (referred to as off-chip memory). However, microcontroller-based devices, such as integrated circuit cards (or smart cards), had substantially less memory, using memory located on the same chip as the processor. These devices did not require external memory to function but were constrained by the amount of space on the chip (or integrated circuit) used for memory. At the time of Gemalto’s invention, there were no Java implementations for microcontroller-based smart cards or integrated circuit cards. Due to the disparity between the constraints of the devices and the demands of the applications, “[f]itting Java technology inside smart cards was like playing golf in a telephone booth.” Joint Appendix (“J.A.”) 643 (quoting Sun Microsystems CEO Scott McNealy) (internal quotation marks omitted).

According to Gemalto, its invention enabled resource-constrained devices to run applications written in high level programming languages (such as Java) by minimiz-

¹ Application Serial No. 60/029,057, filed on October 25, 1996.

² For simplicity, all references to the specification are to the ’317 patent.

ing the computing resources that applications consumed during storage and execution. The asserted claims are directed to applications that are converted from a high level programming language into another format that is suitable for resource-constrained computing devices. The application, in converted form, is stored in the memory of the chip containing the embedded processor that executes the application. However, the processor cannot run the converted application directly and requires an interpreter (or virtual machine) to translate the converted application into instructions that the processor can execute. The interpreter is also stored in on-chip memory. This is important because, to run a Java application, both the application and the interpreter must fit within the constraints of the platform for the purposes of storage and execution.

In October 2010, Gemalto sued the defendants for infringement, alleging that the defendants' smartphones infringe when they run the Android operating system and Java applications (converted using the Android software development kit). The defendants contended that the accused smartphones do not infringe because they are not resource-constrained devices, but rather rely on off-chip memory to run Java applications, similar to prior art personal computers.

Before summary judgment, Gemalto narrowed the number of asserted claims to the 6 claims asserted on appeal: claims 1, 4, and 5 of the '317 patent; claims 38 and 39 of the '485 patent; and claim 3 of the '727 patent.³ These claims recite either an "integrated circuit card" or a "programmable device" that includes a processor and a

³ Gemalto previously asserted 28 claims across the three patents. The 22 claims not asserted on appeal were not before the district court at summary judgment.

“memory” storing a converted application and an interpreter. Claim 1 of the ’317 patent, which is representative of the “integrated circuit card” claims, reads:

1. An integrated circuit card for use with a terminal, comprising:

a communicator configured to communicate with the terminal;

a memory storing:

an application derived from a program written in a high level programming language format wherein the application is derived from a program written in a high level programming language format by first compiling the program into a compiled form and then converting the compiled form into a converted form, the converting step including at least one step selected from a group consisting of

recording all jumps and their destinations in the original byte codes;

converting specific byte codes into equivalent generic byte codes or vice-versa;

modifying byte code operands from references using identifying strings to references using unique identifiers; and

renumbering byte codes in a compiled format to equivalent byte codes in a format suitable for interpretation; and

an interpreter operable to interpret such an application derived from a program written in a high level programming language format; and

a processor coupled to the memory, the processor configured to use the interpreter to interpret the

application for execution and to use the communicator to communicate with the terminal.

'317 patent col. 19 ll. 38–67. Claim 3 of the '727 patent, the only asserted claim directed to a “programmable device,” reads:

3. A programmable device comprising:

a *memory*, and

a processor;

the memory comprising:

an interpreter; and

at least one application loaded in the memory to be interpreted by the interpreter, wherein the at least one application is generated by a programming environment comprising:

a) a compiler for compiling application source programs written in high level language source code form into a compiled form, and

b) a converter for post processing the compiled form into *a minimized form suitable for interpretation within the set of resource constraints by the interpreter.*

'727 patent col. 19 ll. 29–43 (emphases added). Both the integrated circuit card and programmable device claims require the recited “memory” to store a converted application and an interpreter. The asserted programmable device claim expressly requires the application to be converted “into a minimized form suitable for interpretation within the set of resource constraints by the interpreter.” '727 patent col. 19 ll. 41–43.

The district court construed the recited “memory” as “all program memory”—*i.e.*, “sufficient memory to run the Java code [or other high level programming language] in

accordance with the patentee’s invention.” J.A. 26 (footnote omitted). In other words, the court’s construction required all application memory to be stored on the same chip as the processor. The court construed “resource constraints” to mean “insufficient memory to run the compiled application source program in an unconverted form.” J.A. 40–41. The district court construed “integrated circuit card” to mean “a card containing a single semiconductor substrate [*i.e.*, a chip] having a central processing unit and all program memory.” J.A. 31. The court construed “programmable device” as “a single semiconductor substrate integrating electronic circuit components that includes a central processing unit and all program memory making it suitable for use as an embedded system.” J.A. 33.

After the court construed the claims, the defendants moved for summary judgment of non-infringement. Gemalto did not dispute that the “accused devices do not contain ‘all program memory’ on a single semiconductor substrate because they require ‘off chip’ memory to run the accused . . . Android applications,” J.A. 11, but argued that requiring off-chip memory did not preclude infringement. The court concluded that the accused devices did not literally infringe because applications were stored in off-chip memory—*i.e.*, on a chip other than the chip containing the processor. The court also rejected Gemalto’s theory that the accused devices infringed under the doctrine of equivalents when they temporarily loaded program instructions into on-chip cache memory prior to execution. The court concluded that temporary storage in cache memory was not substantially the same as permanent memory storage.

Gemalto appealed. We have jurisdiction pursuant to 28 U.S.C. § 1295(a)(1). Claim construction is a question of law that we review *de novo*. *Lighting Ballast Control LLC v. Philips Elecs. N.A. Corp.*, 744 F.3d 1272, 76–77 (Fed.

Cir. 2014) (en banc); *Cybor Corp. v. FAS Techs., Inc.*, 138 F.3d 1448, 1454–55 (Fed. Cir. 1998) (en banc). We review a district court’s grant of summary judgment of noninfringement without deference. *Bd. of Regents of the Univ. of Tex. Sys. v. BENQ Am. Corp.*, 533 F.3d 1362, 1367 (Fed. Cir. 2008).

DISCUSSION

I. Claim Construction

A. “Integrated Circuit Card” Claims

Our claim construction methodology is described in *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–19 (Fed. Cir. 2005) (en banc). Gemalto first challenges the district court’s construction of the “memory” limitation in the integrated circuit card claims to require all program memory on a single semiconductor substrate (or chip). Gemalto does not dispute that an integrated circuit card is “a card containing a single semiconductor substrate having a central processing unit and memory,” J.A. 31, but contends that the card should be able to access applications stored in off-chip memory, *i.e.*, memory located on chips other than the chip containing the processor. The claim language does not support Gemalto’s theory. The claim language requires a “memory” that stores an “application” and a “processor” that is coupled to the memory. ’317 patent col. 19 ll. 42, 43, 64. The specification demonstrates that the entire purpose of the invention was to enable the application to be stored within the memory on the chip of the integrated circuit card.

Integrated circuit cards, such as smart cards, are small, portable devices. Before Gemalto’s invention, applications for integrated circuit cards were typically written in low level programming languages to conserve memory. With the expansion of digital networking, there was a “need to load new applications on the card

However, typically, this [wa]s not practical . . . using low level languages.” ’317 patent col. 3 ll. 47–48. The purpose of Gemalto’s invention was to make it possible to load applications written in high level languages, such as Java, into the memory of a resource-constrained integrated circuit card.

Gemalto does not dispute that integrated circuit cards are one-chip devices, but argues that these cards can access applications stored in external memory sources, such as other chips, in accordance with its invention. The specification, however, identifies configurations in which a processor runs Java applications stored in off-chip memory and ascribes them to the prior art—*i.e.*, microprocessor-based personal computers. The specification describes “[c]onventional platforms that support Java” as “microprocessor-based computers,” which have “a central processing unit that requires certain external components (*e.g.*, memory, input controls and output controls) to function properly.” ’317 patent col. 1 ll. 55–56; 62–66.

Gemalto’s invention was not directed to these conventional prior art platforms for Java. Instead, it focused on resource-constrained computing platforms, such as integrated circuit cards and microcontrollers, which were previously unable to run Java applications. “In contrast to the microprocessor, a microcontroller includes a central processing unit, memory and other functional elements, all on a single semiconductor substrate, or integrated circuit (*e.g.*, a ‘chip’).” ’317 patent col. 2 ll. 2–5. According to the specification, the crucial difference between prior art microprocessor-based computers and microcontroller-based devices is that “[i]n a microcontroller, the amount of each kind of memory available is constrained by the amount of space on the integrated circuit,” ’317 patent col. 2 ll. 14–16, while “[a] microprocessor system . . . is not constrained by what will fit on a single integrated circuit

device.” *Id.* ll. 26–31. The specification demonstrates that external memory storage was a defining feature of prior art Java technology, and that the patented invention was designed to eliminate the need for such external storage. Of course, “the claims cannot be of broader scope than the invention that is set forth in the specification.” *On Demand Mach. Corp. v. Ingram Indus., Inc.*, 442 F.3d 1331, 1340 (Fed. Cir. 2006).

Gemalto’s interpretation of the term “memory” also conflicts with the prosecution history. During initial prosecution and reexamination of the ’317 patent, Gemalto repeatedly distinguished invalidating prior art by emphasizing the novelty of squeezing a Java application onto the memory of an integrated circuit card. Initially, the United States Patent and Trademark Office (“PTO”) rejected all claims in the ’317 patent as anticipated and obvious in view of U.S. Patent No. 5,923,884 (“Peyret”), titled “System and Method for Loading Applications onto a Smart Card,” as well as other prior art references. In response, the patentee emphasized the novelty of fitting the application onto an integrated circuit card, arguing that “Peyret does not deal with a solution of how to squeeze Java or another high level language onto a smart card.”⁴ J.A. 1162. The applicant explained that

[m]aking it possible to run programs written in a high level language on a smart card was not obvious prior to Applicants’ invention *To put Java (or any other high level language) on an integrated circuit card* is anything but obvious. At the time of the invention, the typical Java Virtual

⁴ The specification uses the terms integrated circuit card and smart card interchangeably. *See, e.g.*, ’317 patent col. 2 ll. 37–38 (“microcontrollers frequently are used in integrated circuit cards, such as smart cards”).

Machine required over 1 MB of memory. Any person of ordinary skill would realize that *to squeeze such an interpreter into an integrated circuit card (such as a smart card) is anything but an obvious task.*”

J.A. 1164 (emphases added). Based on the applicant’s argument, the application subsequently issued as the ’317 patent.

During reexamination of the ’317 patent, the patentee again argued for patentability based on the difficulty of fitting a Java application into the memory of a resource-constrained device. Gemalto argued that “providing Java technology onto smart cards would be very challenging due to the size constraints of smart cards as contrasted to the minimum requirements of Java.” J.A. 1199. Because Java “applications typically compile into files much larger than the pragmatically available space in a smart card[,] . . . the problem of fitting them onto a smart card would be rather challenging. Also, the intermediate language interpreter, *e.g.*, the JVM, would also have to fit on the card.” J.A. 1201.

In obtaining and sustaining the ’317 patent, Gemalto emphasized the significance of fitting the application and interpreter onto the memory of an integrated circuit card. If the card could simply access external memory sources, there would be no need to “squeeze” the application or its interpreter onto the card. The integrated circuit card could simply access the application and interpreter from an external memory source without being constrained by the resources available on the integrated circuit card itself.

However, Gemalto contends that claim differentiation supports its theory that the integrated circuit card claims should be construed to cover off-chip memory. That contention rests on claim 4 of the ’317 patent, which recites

“[t]he integrated circuit card of claim 1 wherein at least a portion of the memory is located in the processor.” ’317 patent col. 20 ll. 6–7. According to this theory, by requiring a “portion of the memory” to be in the processor, claim 4 indicates that some program memory can be stored on separate chips. But the language of claim 4 only allows some memory to be outside the processor, not outside the chip. It implies that the memory recited in other claims may be located on the same chip as the processor, but not within the processor itself. Claim differentiation based on claim 4 only indicates that none of the memory recited in other claims needs to be located “in the processor”; it says nothing about the location of the memory with respect to the chip containing the processor. Thus, claim differentiation does not support Gemalto’s argument or overcome the conclusion that the claims require the processor to be “coupled to the memory,” *id.* col. 19 l. 64, *i.e.*, on the same chip, storing the application and interpreter necessary for the integrated circuit card to run the application.

The patentee’s arguments regarding the “memory” limitation recited in the integrated circuit card claims of the ’317 patent apply equally to the “memory” limitation recited in the integrated circuit card claims of the ’485 patent. As we held in *Elkay Manufacturing Co. v. EBCO Manufacturing Co.*, 192 F.3d 973, 980 (Fed. Cir. 1999), “[w]hen multiple patents derive from the same initial application, the prosecution history regarding a claim limitation in any patent that has issued applies with equal force to subsequently issued patents that contain the same claim limitation.” *Id.* (citing *Jonsson v. Stanley Works*, 903 F.2d 812, 817–18 (Fed. Cir. 1990)); see also *Andersen Corp. v. Fiber Composites, LLC*, 474 F.3d 1361, 1368–69 (Fed. Cir. 2007) (construing claim language based on statements made during prosecution of parent application regarding similar claim language), *Biovail Corp. Int’l v. Andrx Pharms., Inc.*, 239 F.3d 1297,

1301–02 (Fed. Cir. 2001) (construing disputed limitation based on statements relating to that limitation during prosecution of a patent deriving priority from the same original application as the patent-in-suit). The '485 patent is a continuation of the '317 patent and shares the same title, specification, and provisional application from which both derive their priority date. Further, claim 38 of the '485 patent describes the memory and the application it stores using identical language as claim 1 of the '317 patent. *Compare* '317 patent col. 19 ll. 42–49 *with* '485 patent col. 23 ll. 25–32. Claim 39 depends from claim 38 and incorporates the same memory limitation, adding additional limitations only to the process for converting the application before loading it onto the memory.

In short, we agree with the district court that the memory recited in the integrated circuit card claims should be construed as “all program memory,” which means “sufficient memory to run the Java code [*i.e.*, the application and interpreter] in accordance with the patentee’s invention.”

B. “Programmable Device” Claims

Gemalto also argues that any limitations barring the use of off-chip memory in an integrated circuit card are not relevant to claim 3 of the '727 patent, which is directed to a “programmable device.” Gemalto asserts that the term should be construed broadly to mean “a device that can execute a computer program.” Appellant’s Br. 37. Gemalto suggests that its proposed construction reflects the ordinary meaning of “programmable device” to one of skill in the art. However, the testimony of two of the asserted patents’ inventors (Timothy Wilkinson and Scott Guthery), persons skilled in the relevant art, confirms that the term had no such meaning. When asked what the meaning of “programmable device” is, one inventor stated, “[i]t has no meaning.” J.A. 1074. The other averred, “I

don't remember anybody ever talking about a programmable device particularly. . . . Sounds like a patent term to me," and agreed that even a punch-card computer could be considered a programmable device. J.A. 985. Gemalto does not assert that "programmable device" has any ordinary meaning to a skilled practitioner in the field of computer programming, let alone the particular construction it proposes.

In any event, Gemalto's central problem is that the limitation is not "programmable device" but "memory." Claim 3 explicitly requires a device that has "a memory" and "at least one application loaded in the memory." '727 patent col. 19 ll. 30, 34. The meaning of terms appearing in the '727 patent must be construed in light of the same terms recited in the '317 patent. All three asserted patents are directly related (the '727 patent is a continuation of the '485 patent, which is a continuation of the '317 patent), derive priority from the same provisional application, and share the same disputed claim language. Statements made by the applicant concerning the "memory" limitation in the '317 patent's claims therefore apply with equal force to the "memory" limitation in the '727 patent's claims with the same limitation. *Elkay*, 192 F.3d at 980. Claim 3 of the '727 patent uses identical language to that of claim 65 of the '317 patent, requiring the recited application to be "loaded in the memory to be interpreted by the interpreter" and converted into "a minimized form suitable for interpretation within the set of resource constraints by the interpreter." '727 patent col. 19 ll. 30–43; *see also* '317 patent col. 24 ll. 51–64. The claim requires the application to be loaded in the recited memory and processed into a form that allows the application to be interpreted (and thus executed) within the set of resource constraints. As construed by the district court—and not challenged on appeal—those constraints require the

device to lack sufficient memory if the application had not been so converted.

Here, the meaning of “memory” is made even clearer by the language of the programmable device claim that requires the application loaded in the memory to be converted “into a minimized form suitable for interpretation within the set of *resource constraints* by the interpreter.” ’727 patent col. 19 ll. 42–43 (emphasis added). Gemalto’s arguments to the PTO during prosecution of the ’317 patent directly make clear the meaning of “resource constraints” recited in claim 3 of the ’727 patent. To obtain the ’317 patent, Gemalto emphasized that the conversion of the application was necessary because of the resource constraints on the device on which it was stored. Gemalto explained that

Appellants recognized the difficulty in operating Java (or other high level language) programs within the limited resources of an integrated circuit card or other microcontroller. To solve that problem, Appellants introduced the additional step of converting the compiled form from a Java compiler into a form suitable for interpretation on a specialized interpreter. Peyret does not teach or suggest this additional step.

J.A. 1174. The prosecution history of the ’727 patent and those from which it derives priority do not support Gemalto’s theory that a “programmable device” can be any type of computer. Rather, the prosecution history demonstrates that a programmable device must have resource constraints that the patentee argued made squeezing Java applications on them “anything but obvious.” J.A. 1176.

On appeal, Gemalto does not challenge the construction of “resource constraints” as meaning “insufficient memory to run the compiled application source program

in an unconverted form.” J.A. 40–41. While Gemalto admits that the device must meet these limitations, it does not explain how the broad construction it proposes would satisfy them. To be consistent with the “resource constraints” limitation, the device must have insufficient memory to run the application in an unconverted form. Allowing the device to access external off-chip memory would effectively read this express limitation out of the claim.

Gemalto contends that the prosecution history of the ’727 patent supports a broad construction, pointing to a statement by the examiner that “the present claims are merely considered *a broader recitation* of claims 1–86 of the ’317 patent.” Appellant Br. 34 (internal quotation marks omitted). In fact, the examiner made the exact same statement about the claims as originally drafted—at which point they recited a “smart card” rather than a “programmable device.” Further, the examiner also recognized that the “[t]he subject matter claimed in the instant application [for the ’727 patent] is fully disclosed in the [’317] patent and is covered by the [’317] patent.” J.A. 751; *see also* J.A. 1183. The ’317 patent expressly discloses prior art computers—devices capable of executing computer programs—that run Java applications without the claimed invention. The examiner’s statement, standing alone, does not support construing “programmable device” so broadly that it encompasses any device capable of executing program instructions, particularly those within the prior art as defined by the specification of the ’727 patent and shared by all three asserted patents.

We thus agree with the district court that the programmable device of claim 3 of the ’727 patent should be construed as “a single semiconductor substrate integrating electronic circuit components that includes a central

processing unit and all program memory making it suitable for use as an embedded system.” J.A. 33.

II. Infringement

The district court granted summary judgment of non-infringement as to all asserted claims. The court found that the accused devices did not infringe literally because it was undisputed that they “store program instructions off-chip and access those off-chip instructions to run the accused applications.” J.A. 12. Because the devices could not literally infringe, the court found no indirect infringement of claim 3 of the ’727 patent. With respect to infringement, Gemalto does not challenge the court’s findings on literal or indirect infringement under the district court’s construction.

Instead, Gemalto argues that the accused devices infringe the asserted claims, as construed by the district court, under the doctrine of equivalents. A device that does not literally infringe a claim may infringe under the doctrine of equivalents. *Energy Transp. Grp., Inc. v. William Demant Holding A/S*, 697 F.3d 1342, 1352 (Fed. Cir. 2012). We have held that “a patentee may prove that a particular claim element is met under the doctrine of equivalents . . . by showing that ‘the accused product performs substantially the same function in substantially the same way with substantially the same result’ as claimed in the patent.” *Id.* (quoting *Crown Packaging Tech., Inc. v. Rexam Bev. Can Co.*, 559 F.3d 1308, 1312 (Fed. Cir. 2009)); see also *Warner-Jenkinson Co. v. Hilton Davis Chem. Co.*, 520 U.S. 17, 39–40 (1997).

Gemalto contends that the accused devices infringe under the doctrine of equivalents when they temporarily load program instructions from off-chip memory into on-chip cache memory before execution. Because cache memory cannot store applications (or any content) when a device is turned off, the court concluded that cache

memory is substantially different from permanent memory and not equivalent for infringement purposes. Gemalto admits that cache memory does not store program instructions when the accused devices are turned off, but argues that the difference between temporary on-chip storage and permanent on-chip storage is insubstantial or, alternatively, that the significance of this difference raises a genuine question of material fact. Gemalto's theory of infringement by equivalents hinges on its contention that on-chip cache memory is equivalent to on-chip memory permanently storing applications. According to Gemalto, applications are loaded into on-chip cache memory before execution 97% of the time, and the difference between 97% and 100% is insubstantial.

A plaintiff must provide "particularized testimony and linking argument to show the equivalents" are insubstantially different. *AquaTex Indus., Inc. v. Techniche Solutions*, 479 F.3d 1320, 1329 (Fed. Cir. 2007); *accord Texas Instruments, Inc. v. Cypress Semiconductor Corp.* 90 F.3d 1558, 1566 (Fed. Cir. 1996). "Generalized testimony as to the overall similarity between the claims and the accused infringer's product or process will not suffice." *Texas Instruments*, 90 F.3d at 1567. These requirements "assure that the fact-finder does not, 'under the guise of applying the doctrine of equivalents, erase a plethora of meaningful structural and functional limitations of the claim on which the public is entitled to rely in avoiding infringement.'" *Id.* (quoting *Pennwalt Corp. v. Durand-Wayland, Inc.*, 833 F.2d 931, 935 (Fed. Cir. 1987) (en banc)). Gemalto has failed to provide particularized testimony and linking argument.

Gemalto provided no testimony asserting that the difference in functionality between cache memory and permanent memory is in fact insubstantial. Gemalto's expert testimony only addresses the difference between having 100% or 97% of program instructions stored on-

chip, not the underlying difference between temporary and permanent storage. The testimony does not address whether cache memory is equivalent to other types of memory that are capable of storing applications after a device is turned off. Absent any testimony suggesting that cache memory is equivalent to permanent memory, no reasonable fact finder could find that the two types of memory function in substantially the same way to achieve substantially similar results.

Gemalto asserts that the “invention works the same whether or not the memory is on the same chip as the CPU, or the program is executed from temporary memory.” Appellant’s Reply Br. 30. These arguments are not directed to the equivalence between the accused products’ functionality and the particular claim limitation reciting a memory storing an application and interpreter as a whole. Rather, Gemalto offers “merely generalized testimony as to overall similarity” between the accused products and the claims, which under well-established case law “cannot support a finding that the differences [a]re ‘insubstantial.’” *Texas Instruments*, 90 F.3d at 1568; *see also AquaTex*, 479 F.3d at 1329 (“[T]he doctrine of equivalents must be applied to individual elements of the invention, not the invention as a whole.” (quoting *Warner-Jenkinson*, 520 U.S. at 29)). Establishing infringement under the doctrine of equivalents requires particularized testimony and linking argument as the equivalence between the claim limitation and the alleged equivalent. *Texas Instruments*, 90 F.3d at 1567. Such testimony is missing here.

Moreover, the cache memory functionality that is the basis for Gemalto’s theory was employed by microprocessor-based systems at the time of the invention. Gemalto has admitted that “microprocessors in 1996 *did* have cache memory,” J.A. 1852, and that these microproces-

sors ran Java applications before the 1996 priority date of the asserted patents. If cache memory were equivalent to the recited memory that stores an application and interpreter, Gemalto's claims would read on microprocessor systems that were widely used prior to its invention. The doctrine of equivalents cannot be applied to encompass the prior art as "this court has consistently limited the doctrine of equivalents to prevent its application to ensnare prior art." *Marquip, Inc. v. Fosber Am., Inc.*, 198 F.3d 1363, 1367 (Fed. Cir. 1999) (citing *Wilson Sporting Goods v. David Geoffrey & Assoc.*, 904 F.2d 677, 683 (Fed. Cir. 1990)). In *Marquip*, we recognized that "[b]ecause prior art limits the exclusive right available to an inventor, it also limits the range of permissible equivalents of a claim." *Id.* Applying the doctrine of equivalents to cover cache memory used in the prior art is not permissible. We agree with the district court that the accused devices do not infringe under the doctrine of equivalents due to their use of cache memory.

CONCLUSION

We affirm the district court's claim construction and summary judgment of no infringement with respect to all asserted claims.

AFFIRMED