

**IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
MIDLAND/ODESSA DIVISION**

NEURAL AI, LLC,

Plaintiff,

v.

NVIDIA CORPORATION,

Defendant.

Case No. 7:24-cv-00221-ADA-DTG

JURY TRIAL DEMANDED

**DEFENDANT NVIDIA CORPORATION'S
OPENING CLAIM CONSTRUCTION BRIEF**

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	1
II. BACKGROUND	2
III. TERMS FOR CONSTRUCTION	3
A. “accelerator” (’867 patent, cl. 16; ’438 patent, cls. 1, 12, 44)	5
1. The Claimed “Accelerator” Is Separate from the CPU	5
2. The Claimed “Accelerator” Is Hardware Based	7
B. “accelerator controller” (’867 patent, cls. 16, 17, 19) / “controller” (’438 patent, cls. 1, 7–10, 21, 40, 42, 44, 50–53)	8
C. “partition” (’867 patent, cls. 16, 18; ’438 patent, cls. 26, 27, 40, 43, 49; ’461 patent, cls. 21, 30)	10
D. “bank” (’438 patent, cls. 6, 20)	13
E. “swapping the first pointer with the second pointer” (’867 patent, cl. 16)	15
F. “input data at first rate; and . . . sequence of computations at a second rate” (’438 patent, cls. 3, 14, 46)	19
1. “input data at a first rate”	19
2. “sequence of computations at a second rate”	20
G. Order of Steps (’867 patent, cl. 16; ’438 patent, cls. 12, 21)	20
1. Claim 16 of the ’867 Patent	21
2. Claims 12 and 21 of the ’438 Patent	22
IV. CONCLUSION	24

TABLE OF AUTHORITIES

	<u>Page(s)</u>
Cases	
<i>Abbott Lab'ys v. Sandoz, Inc.</i> , 544 F.3d 1341 (Fed. Cir. 2008).....	19
<i>Altiris, Inc. v. Symantec Corp.</i> , 318 F.3d 1363 (Fed. Cir. 2003).....	22
<i>Axonics, Inc. v. Medtronic, Inc.</i> , 75 F.4th 1374 (Fed. Cir. 2023)	4
<i>Becton, Dickinson & Co. v. Tyco Healthcare Grp., LP</i> , 616 F.3d 1249 (Fed. Cir. 2010).....	5
<i>Bid for Position, LLC v. AOL, LLC</i> , 601 F.3d 1311 (Fed. Cir. 2010).....	15
<i>Computer Docking Station Corp. v. Dell, Inc.</i> , 519 F.3d 1366 (Fed. Cir. 2008).....	19
<i>Edwards Lifesciences LLC v. Cook Inc.</i> , 582 F.3d 1322 (Fed. Cir. 2009).....	15
<i>Fenner Invs., Ltd. v. Cellco P'ship</i> , 778 F.3d 1320 (Fed. Cir. 2015).....	19
<i>Garrity Power Servs. LLC v. Samsung Elecs. Co.</i> , No. 2:20-cv-00269-JRG, 2021 WL 3403747 (E.D. Tex. Aug. 4, 2021)	4
<i>Gemalto S.A. v. HTC Corp.</i> , 754 F.3d 1364 (Fed. Cir. 2014).....	8
<i>Interactive Gift Exp., Inc. v. Compuserve Inc.</i> , 256 F.3d 1323 (Fed. Cir. 2001).....	21
<i>Kaneka Corp. v. Xiamen Kingdomway Group Co.</i> , 790 F.3d 1298 (Fed. Cir. 2015).....	23
<i>KEYnetik, Inc. v. Samsung Elecs. Co.</i> , 837 F. App'x 786 (Fed. Cir. 2020)	6, 10
<i>Mantech Env't Corp. v. Hudson Env't Servs., Inc.</i> , 152 F.3d 1368 (Fed. Cir. 1998).....	21, 25
<i>Multiform Dessicants, Inc. v. Medzam Ltd.</i> , 133 F.3d 1473 (Fed. Cir. 1998).....	19
<i>Netcraft Corp. v. eBay, Inc.</i> , 549 F.3d 1394 (Fed. Cir. 2008).....	15

<i>Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.</i> , 868 F.3d 1013 (Fed. Cir. 2017).....	4
<i>Nystrom v. TREX Co.</i> , 424 F.3d 1136 (Fed. Cir. 2005).....	15
<i>O2 Micro Int’l Ltd. v. Beyond Innovation Tech. Co.</i> , 521 F.3d 1351 (Fed. Cir. 2008).....	13
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005).....	19
<i>Praxair, Inc. v. ATMI, Inc.</i> , 543 F.3d 1306 (Fed. Cir. 2008).....	6, 10
<i>Regeneron Pharms., Inc. v. Mylan Pharms. Inc.</i> , 130 F.4th 1372 (Fed. Cir. 2025)	5
<i>Regents of Univ. of Minnesota v. AGA Med. Corp.</i> , 717 F.3d 929 (Fed. Cir. 2013).....	9
<i>Sequoia Tech., LLC v. Dell, Inc.</i> , 66 F.4th 1317 (Fed. Cir. 2023)	6
<i>Springs Window Fashions L.P. v. Novo Indus., L.P.</i> , 323 F.3d 989 (Fed. Cir. 2003).....	18
<i>Wasica Fin. GmbH v. Cont’l Auto. Sys., Inc.</i> , 853 F.3d 1272 (Fed. Cir. 2017).....	13, 14

TABLE OF EXHIBITS

Exhibit	Document
Exhibit 1	Excerpts of Chambers Dictionary of Science and Technology (“Chambers Dict.”)
Exhibit 2	Excerpts of Authoritative Dictionary of IEEE Standards Terms 2000 (“IEEE Dict.”)
Exhibit 3	Excerpts of Newton’s Telecom Dictionary, 22d ed. (“Newton’s Telecom Dict.”)
Exhibit 4	Excerpts of Microsoft Computer Dictionary 2002 (“Microsoft Dict.”)
Exhibit 5	Excerpts of IBM Dictionary of Computing 1994
Exhibit 6	Excerpts of ’867 Patent Infringement Contentions (“’867 Inf. Cont.”)
Exhibit 7	Excerpts of Dictionary of Computer Terms, 4th ed. (“Dict. of Computer Terms”)
Exhibit 8	Excerpts of Wiley Electrical and Electronics Engineering Dictionary (“Wiley Dict.”)
Exhibit 9	Excerpts of Comprehensive Dictionary of Electrical Engineering, 2d ed.
Exhibit 10	Excerpts of McGraw-Hill Dictionary of Scientific and Technical Terms, 6th ed. (“McGraw-Hill Dict.”)
Exhibit 11	Excerpts of ’867 File History (“Remarks”)
Exhibit 12	Excerpts of ’438 Patent Infringement Contentions (“’438 Inf. Cont.”)

TABLE OF ABBREVIATIONS

Term	Meaning
'438 Patent	U.S. Patent No. RE48,438
'461 Patent	U.S. Patent No. RE49,461
'867 Patent	U.S. Patent No. 8,648,867
Asserted Patents	'867, '438, and '461 Patents
Neural AI or Plaintiff	Plaintiff Neural AI, LLC
NVIDIA or Defendant	Defendant NVIDIA Corporation

I. INTRODUCTION

Plaintiff Neural AI, LLC (“Plaintiff” or “Neural AI”) asserts three patents directed to systems and methods for performing *general-purpose* computing (such as numerical simulations or simulating artificial neural networks) on traditionally *special-purpose* graphics processing units (“GPUs”). Defendant NVIDIA Corporation (“Defendant” or “NVIDIA”) proposes the Court explain the plain and ordinary meaning, as used in the asserted patents, of a handful of terms that go to the heart of Plaintiff’s infringement allegations. Plaintiff resists these constructions, simply taking the position that nothing in the asserted patents—even the most technical terms—requires construction. For the terms below, however, absent an explanation of what the plain and ordinary meaning is, a jury is likely to be confused as to the scope of the claims given the technical nature of computer terminology involved. For example, the term “accelerator” may have general connotations to a lay person, but it has a specific meaning to a person of ordinary skill in the context of the asserted patents, which is not immediately apparent to a lay person. Another technical term, “accelerator controller,” has a specific meaning in the context of the asserted patents such that a construction is necessary. Additionally, for two terms—“partition” and “bank”—Plaintiff’s infringement contentions impermissibly expand their scope beyond that allowed by the intrinsic record, such that proceeding without clarifying the terms’ ordinary meaning does not resolve the parties’ dispute. Despite the technical subject matter of the asserted patents, Plaintiff believes that a jury will understand the meaning of all of these terms without assistance; to the contrary, NVIDIA contends that construction is necessary to explain the plain and ordinary meaning of these terms in the context of these patents, as well as to resolve disputes between the parties as to claim scope.

NVIDIA therefore respectfully requests the Court adopt NVIDIA’s proposed constructions to ensure that the claims are interpreted in a manner consistent with the specifications and well-established understanding in the field.

II. BACKGROUND

Plaintiff accuses NVIDIA of infringing three related patents: U.S. Patent Nos. 8,648,867 (“the ’867 patent”), RE48,438 (“the ’438 patent”), and RE49,461 (“the ’461 patent”) (collectively, “the Asserted Patents”), each titled “Graphic Processor Based Accelerator System and Method.” “GPUs,” which are found in many computers and video game consoles, are “highly parallel processors.” Unlike CPUs, which excel at handling tasks sequentially, GPUs are optimized for parallel processing. ’867 patent at 1:13–17. Those parallel-processing capabilities can be used for graphics or for general-purpose computations, such as simulations. Using GPUs for general-purpose computing has been referred to as “general-purpose computing on GPUs,” or “GPGPU.”

The Asserted Patents acknowledge that “manufacturers of GPUs have included general purpose programmability into the GPU architecture leading to the increased popularity of using GPUs for highly parallelizable and computationally expensive algorithms outside of the computer graphics domain.” *Id.* at 1:21–25. The patents assert, however, “general purpose GPU (GPGPU) applications are not able to achieve optimal performance” because “[t]here is overhead for graphics-related features and algorithms that are not necessary for these non-video applications.” *Id.* at 1:25–30.

To address these alleged drawbacks, the Asserted Patents describe an “accelerator” (Figure 2, below) separate from the CPU that can offload functions from the CPU. The “accelerator” is defined in the Asserted Patents has “one or more graphics processing units” (GPU 240), “two or more associated memory banks that are logically or physically partitioned” (shader memory bank 210 and texture memory bank 250), and “a specialized controller” (controller 220).

Id. at 2:6–17. This arrangement allegedly improves performance because “[t]he controller handles most of the primitive operations needed to set up and control GPU computation.” *Id.* at 2:17–19. “As a result, the CPU is freed from this function and is dedicated to other tasks.” *Id.* at 2:19–20.

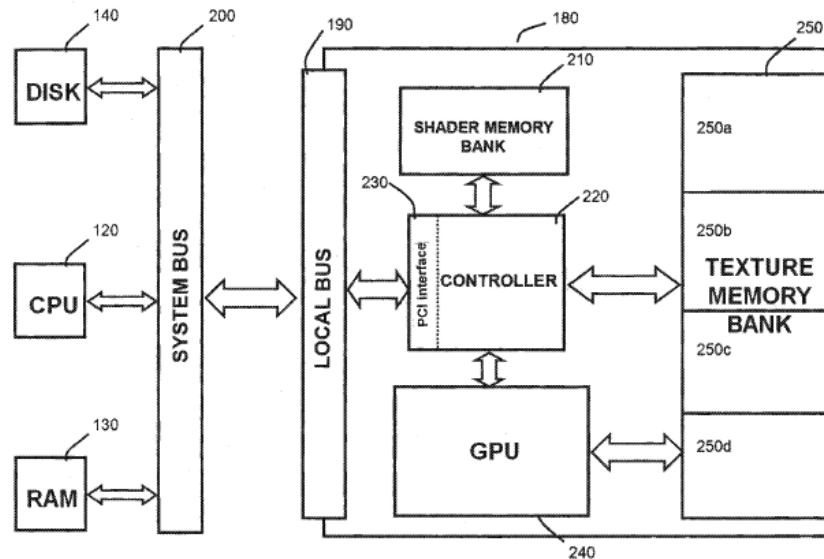


FIG. 2

The three Asserted Patents purport to claim different aspects and applications of the specific architecture described above. The '867 patent claims “[a] method for performing a numerical simulation” on an “accelerator” in which the output of one cycle of the computation is used as an input to the next cycle of computation ('867 patent at 16:21–40), and the '438 and '461 patents claim systems and methods for simulating an artificial neural network on a GPU ('438 patent at 14:51–21:9; '461 patent at 16:41–17:3).

III. TERMS FOR CONSTRUCTION

NVIDIA requests that the Court adopt each of the constructions below:

- “Accelerator” and “accelerator controller” are technical terms with a specific meaning in the context of the purported invention in the Asserted Patents. The Court should explain the meaning of these technical terms, in the context of these patents, to assist the jury understand these terms.

- For “partition” and “bank”—in addition to being technical terms with specific meanings in the context of the Asserted Patents—Plaintiff’s infringement contentions demonstrate a disagreement between the parties regarding their scope. The Court should construe these terms to prevent Plaintiff from continuing to stretch them beyond the meaning compelled by the intrinsic record of the patents.
- For “swapping the first pointer with the second pointer,” the Court should construe the term to account for the applicant’s arguments during prosecution that limited the scope of the limitation.
- Like the “accelerator” terms, the “rate” terms are technical terms, and the jury would benefit from an explanation of their plain and ordinary meaning.
- Finally, the Court should construe the asserted method claims to require the specific order of steps dictated by the claims.

In its Disclosure of Extrinsic Evidence, Plaintiff argues that the “plain and ordinary meaning” of each term is appropriate because NVIDIA did not propose these constructions in *inter partes* review (“IPR”) petitions on the Asserted Patents. That theory has been rejected by other courts because the Patent Trial and Appeal Board (“PTAB”) construes terms only when necessary for an IPR proceeding—that is, an IPR petition need not “describe all possible or reasonable claim constructions.” *Garrity Power Servs. LLC v. Samsung Elecs. Co.*, No. 2:20-cv-00269-JRG, 2021 WL 3403747, at *11 (E.D. Tex. Aug. 4, 2021) (declining to “preclude[] an accused infringer from proposing a construction in litigation” due to “the absence of a proposed construction in IPR proceedings.”); *cf. Axonics, Inc. v. Medtronic, Inc.*, 75 F.4th 1374, 1380–81 (Fed. Cir. 2023); *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (explaining that construction is required only for disputed terms, and only as necessary to resolve the controversy). The terms proposed for construction here are necessary for *this litigation*.

A. “accelerator” (’867 patent, cl. 16; ’438 patent, cls. 1, 12, 44)

NVIDIA’s Proposed Construction	Plaintiff’s Proposed Construction
hardware separate from the central processing unit	No construction necessary

The claims and specification of the ’867 and ’438 patents dictate that the recited “accelerator” of the patents is (1) separate from the central processing unit, and (2) hardware. Providing an explanation of the meaning of “accelerator”—of which a lay juror may have a general understanding in other contexts—is necessary to enable the jury to understand the plain and ordinary meaning of this term in the context of the patents, which states that this feature is the entire purpose of the alleged invention.

1. The Claimed “Accelerator” Is Separate from the CPU

The intrinsic evidence—and, indeed, the very purpose of the Asserted Patents—requires that the “accelerator” be separate from the CPU.

Intrinsic Record: Because the “accelerator” of the asserted claims is listed separately from the CPU, “the clear implication of the claim language is that those elements are distinct components of the patented invention.” *Becton, Dickinson & Co. v. Tyco Healthcare Grp., LP*, 616 F.3d 1249, 1254 (Fed. Cir. 2010) (internal quotation marks omitted). For example, claim 16 of the ’867 patent recites “an accelerator, operably coupled to the central processing unit.” ’867 patent at 14:21–22; *see also* ’438 patent at 14:56–57 (same), 16:3–4 (“an accelerator operably coupled to the CPU and the main memory”). There is no evidence in the claims or specification of the patents to overcome the presumption of separateness under *Becton*. *See Regeneron Pharms., Inc. v. Mylan Pharms. Inc.*, 130 F.4th 1372, 1380–81 (Fed. Cir. 2025).

Throughout the specification, the accelerator is separate from the CPU. *See* ’867 patent at 4:31–33. The specification explains, for example, that “[t]he CPU 120 and the system memory 130 . . . are preferably independent of the expansion card 180,” which is an “accelerator.” *Id.* at

4:8–12. Figure 2 depicts the accelerator/expansion card (180) separated from the CPU via local bus 190. *Id.* at 4:51–58, Fig. 2. Notably, the accelerator is separate from the CPU in *all* embodiments of the Asserted Patents. *Praxair, Inc. v. ATMI, Inc.*, 543 F.3d 1306, 1324 (Fed. Cir. 2008) (“The claims of the patent must be read in light of the specification’s consistent emphasis on this fundamental feature of the invention.”).

The stated purpose and alleged advantages of the patents also support NVIDIA’s proposed construction. As explained in Section II, the patents state that the purpose of the accelerator is to allow the CPU to remain “dedicated to other tasks,” *i.e.*, not be involved in the accelerated computations on the accelerator. ’867 patent at 2:19–21. The patents reiterate at the end, in a summary of the “potential advantages” of the “GPU accelerator system,” that there are “[l]imited computations on the CPU 120.” *Id.* at 13:32. Thus, the “accelerator” being separate from the CPU is critical to the patents’ functionality, as it enables the offloading of computational tasks from the CPU to the accelerator. “To find otherwise would eviscerate the stated purpose of the claimed invention.” *KEYnetik, Inc. v. Samsung Elecs. Co.*, 837 F. App’x 786, 792 (Fed. Cir. 2020); *see also Sequoia Tech., LLC v. Dell, Inc.*, 66 F.4th 1317, 1326 (Fed. Cir. 2023) (“A patent’s express purpose of the invention informs the proper construction of claim terms.”) (cleaned up).

Extrinsic Record: The extrinsic evidence is consistent with the intrinsic record. For example, the Chambers Dictionary of Science and Technology confirms that an “accelerator”—just as it is used in the examples of the Asserted Patents—is “[a] special circuit board that is placed within a computer to speed up some aspect of its operation”—*i.e.*, it is separate from the CPU. Ex. 1 at 7. None of the extrinsic evidence supplied by Plaintiff contradicts this understanding of “accelerator” in the specific context of the Asserted Patents. *See, e.g.*, IEEE Dict. (Ex. 2) at 4 (“[a] circuit or device that accelerates some unit in a computer”).

2. The Claimed “Accelerator” Is Hardware

Intrinsic Record: The claimed “accelerator” is referenced by the claims and the specification of the ’867 and ’438 patents only as a hardware-based component. For example, in the asserted system claims, the “accelerator” comprises specific hardware sub-components, including a “graphics processing unit” and “accelerator memory.” ’438 patent at 14:54–16:4 (cl. 1), 19:60–20:7 (cl. 44). Because the “accelerator” in the asserted method claims “receives” input data, those claims also require a hardware-based accelerator (otherwise, it could not “receive” data).

In the Summary of the Invention, the “solution” proposed by the inventors is a “hardware implementation [on] an expansion card . . . having (a) one or more graphics processing units, (b) two or more associated memory banks that are logically or physically partitioned, (c) a specialized controller, and (d) a local bus”—*i.e.*, all hardware. ’867 patent at 2:6–17. A similar configuration is depicted in Figure 2, where “the GPU accelerator is implemented as an expansion card 180.” *Id.* at 4:31–43, 4:51–54. The specification does not disclose a *software*-based accelerator; rather, the GPU accelerator of all embodiments of the Asserted Patents is hardware based, with physical hardware subcomponents such as a GPU and memory banks. *See id.* at 3:59–4:7, 4:31–58, 5:17–19.

Extrinsic Record: The extrinsic evidence further supports NVIDIA’s construction. Indeed, the definitions of “accelerator” identified by both NVIDIA and Plaintiff emphasize an accelerator as a physical element. *E.g.*, Chambers Dict. (Ex. 1) at 7 (“[a] special circuit board”); Newton’s Telecom Dict. (Ex. 3) at 78 (“[a] board added onto a personal computer’s main board”); Microsoft Dict. (Ex. 4) at 13 (“In hardware, a device . . .”); IEEE Dict. (Ex. 2) at 4 (“[a] circuit or device”).

**B. “accelerator controller” (’867 patent, cls. 16, 17, 19) /
“controller” (’438 patent, cls. 1, 7–10, 21, 40, 42, 44, 50–53)**

NVIDIA’s Proposed Construction	Plaintiff’s Proposed Construction
specialized hardware and/or software located on the accelerator	No construction necessary

“Accelerator controller” is a technical term with a specific meaning in the computer context that a lay person would not understand, and thus should be construed to provide guidance to the jury. NVIDIA’s construction should be adopted because it comports with the intrinsic record—*i.e.*, that the “accelerator controller” of the ’867 patent and the “controller” of the ’438 patent is specialized hardware and/or software located on the accelerator. *Gemalto S.A. v. HTC Corp.*, 754 F.3d 1364, 1369 (Fed. Cir. 2014).

Intrinsic Record: As explained in Section II, the inventors of the patents sought to improve existing GPGPU applications that were “not able to achieve optimal performance” due to excessive “overhead.” ’867 patent at 1:21–30. For example, according to the inventors, prior art systems of GPU acceleration required “extensive interactions with the CPU.” *Id.* at 13:32–39. The “solution” proposed by the inventors was to configure the accelerator (“expansion card”) to include “a specialized controller” separate from the CPU, thus “free[ing]” the CPU from control operations and allowing it to be “dedicated to other tasks.” *Id.* at 2:6–20, 2:36–40 (“This accelerator comprises at least one graphics processing unit, accelerator memory for the graphics processing unit, and an accelerator controller”); *see also id.* at 13:32–39 (identifying the first “potential advantage[]” of the claimed GPU accelerator as “[l]imited computations on the CPU”).

While the specification explains that in one embodiment the “controller can be partially or even fully implemented in software,” even in this embodiment the software-based controller is located and running *on the accelerator*—not on the CPU. *Id.* at 4:65–67. For example, Figure 2 illustrates controller 220 (blue) on expansion card 180 (the “accelerator”) (red). *Id.* at 4:51–67

(“The GPU accelerator further preferably comprises one specifically designed accelerator controller.”); *see also id.* at 2:55–60, 5:53–59, 7:57–61, 8:50–57, 13:50–54 (“general purpose controller 220 located on the expansion card”). There is no embodiment in which the controller is *not* located on the accelerator. Although the Federal Circuit has warned against reading in limitations from embodiments into the claims, the court has also held that where “only one” embodiment is described—especially where, as here, that embodiment is a key feature of the “invention”—that embodiment informs the scope of the claims. *See Regents of Univ. of Minnesota v. AGA Med. Corp.*, 717 F.3d 929, 936 (Fed. Cir. 2013) (limiting claims to devices formed by affixing two physically separate disks because “attaching two disks together is not just the preferred embodiment of the invention; it is the *only* one described”) (internal quotation marks omitted).

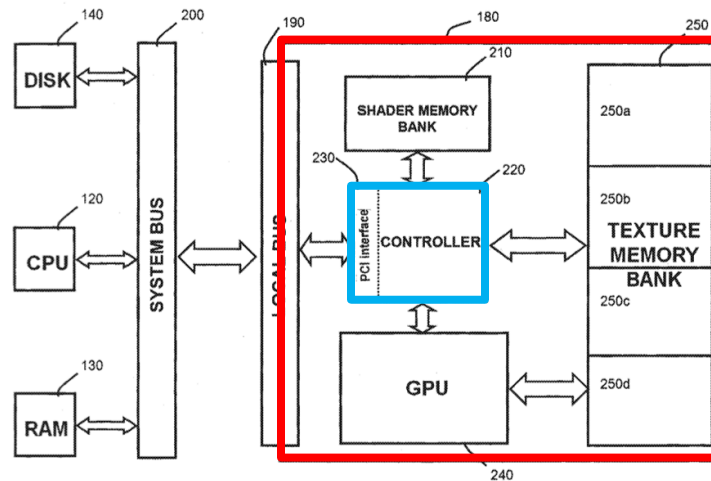


FIG. 2

By locating the controller on the accelerator (as opposed to on the CPU), the system “frees the CPU 120 to execute other applications and allows the expansion card to run at its full capacity without being slowed down by extensive interactions with the CPU 120.” *Id.* at 13:36–39. The claims of the Asserted Patents “must be read in light of the specification’s consistent emphasis on this fundamental feature of the invention.” *Praxair*, 543 F.3d at 1324. To construe the term

otherwise “would eviscerate the stated purpose of the claimed invention.” *KEYnetik*, 837 F. App’x at 792.

Extrinsic Record: As with “accelerator,” none of the extrinsic evidence changes the ordinary meaning of “accelerator controller” or “controller” in the context of the Asserted Patents. For example, Plaintiff’s definition in the IBM Dictionary of Computing states that a controller “coordinates and controls the operation of one or more input/output devices” (Ex. 5 at 145), but it does not bear on the allegedly inventive idea in the patents of placing the controller on the accelerator itself to free up the CPU. *See supra*.

C. “partition” (’867 patent, cls. 16, 18; ’438 patent, cls. 26, 27, 40, 43, 49; ’461 patent, cls. 21, 30)

NVIDIA’s Proposed Construction	Plaintiff’s Proposed Construction
a separated subdivision of memory designed to hold specified data	No construction necessary

The term “partition” requires construction not only because it is a technical term but also because there is a dispute between the parties about the scope of the ordinary meaning of this term in these patents (as reflected in Plaintiff’s infringement contentions). “[P]artition” has a specific and well-understood technical meaning: “a separated subdivision of memory designed to hold specified data.” Plaintiff’s infringement contentions, by contrast, appear to take the position that the data copied to any “destination memory address” on a device is necessarily stored within a “partition” on the device. *See, e.g.*, ’867 Inf. Cont. (Ex. 6) at 45–54. This position is contrary to the meaning of “partition” in the patents. Thus, there is a dispute about claim meaning and claim scope, and claim construction is necessary. NVIDIA’s construction aligns with what a person of ordinary skill in the art would have understood at the time of the invention and should be adopted.

Intrinsic Record: The claim language explicitly states the partitions are subdivisions of memory designed to hold specific data. The claims of the ’867 patent recite, for example, that the

first partition is for storing “first input data” and the second partition is for storing “first output data.” ’867 patent cl. 16 (“transferring . . . the first input data into a first partition” and “storing . . . the first output data into a second partition”); *see also id.* at cls. 1, 13 (“the texture memory is partitioned into a first partition to store the input data, a second partition to store internal variables . . .”). The claims further recite the existence of third and fourth partitions for storing “internal variables” and “data used as input at a particular computation cycle of the numerical simulation.” ’867 patent cl. 18 (“a third partition to store internal variables, and a fourth partition to store data used as input at a particular computation cycle of the numerical simulation.”); *see also id.* at cls. 5, 13. These claims demonstrate that partitions are structured subdivisions of memory with designated storage functions, rather than ad hoc for individual variables. *See also* ’438 patent cls. 26, 27, 28, 40, 43, 49; ’461 patent cls. 21, 30.

The shared specification demonstrates why NVIDIA’s construction is correct. According to the specification, the purpose of partitions is to “separate the memory” based on when and how the data in each memory partition is to be used. *See* ’867 patent at 5:17–52, 5:60–6:31. For example, the specification describes four partitions, each “designed to hold” specified data: “The first partition 250*a* is designed to hold the external input data patterns. The second partition 250*b* is designed to hold the data textures representing internal variables. The third partition 250*c* is designed to hold the data textures used as input at a particular computation step on the GPU 240. The fourth partition 250*d* holds the data textures used to accommodate the output of a particular computational step on the GPU 240.” *Id.* at 5:39–48.

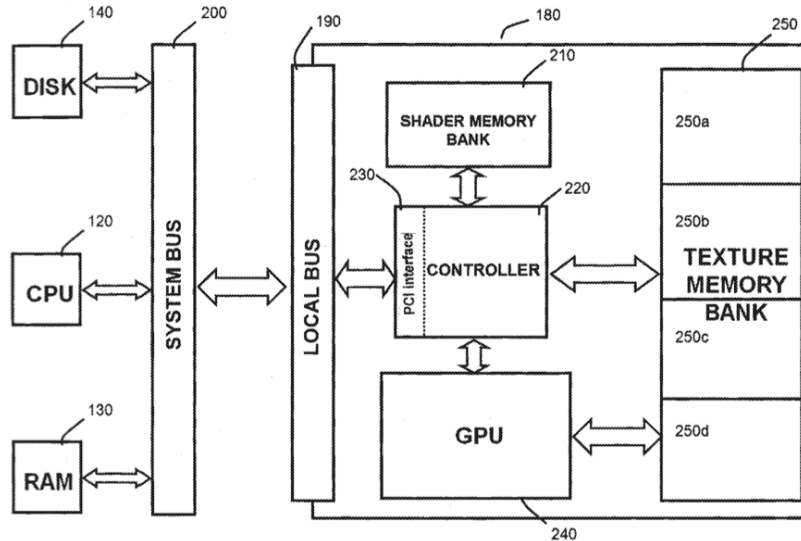


FIG. 2

Id. at Fig. 2. Each of the partitions (250a, 250b, 250c, and 250d) is a separate subdivision of texture memory bank 250, which itself is a partition of GPU associated memory. *Id.* at 5:14–52. Thus, use of “partition” throughout the specifications of the Asserted Patents demonstrates that a partition must be “a separated subdivision of memory designed to hold specified data.”

Extrinsic Record: The extrinsic evidence confirms NVIDIA’s construction. For example, the technical dictionary definitions of “partition” identified by NVIDIA emphasize that partitions are (1) separated subdivisions of memory that (2) hold specified data. *E.g.*, Chambers Dict. (Ex. 1) at 873 (“To subdivide the available storage space on a hard-disk drive. Each section may be used to hold programs and data for different operating systems or to provide additional logical drives.”); Dict. of Computer Terms (Ex. 7) at 232 (“a part of a HARD DISK that is treated by the computer as if it were a separate disk drive”); IEEE Dict. (Ex. 2) at 797 (“A portion of a storage medium that is set aside for some special purpose”); Wiley Dict. (Ex. 8) at 555 (“The dividing of computer memory into fixed portions”). Plaintiff’s extrinsic evidence is consistent. *See, e.g.*, Microsoft Dict. (Ex. 4) at 392 (“partition n. 1. A logically distinct portion of memory or a storage device that functions as though it were a physically separate unit.”).

Infringement Contentions: Plaintiff’s infringement contentions demonstrate that the parties disagree as to the claims’ meaning and scope, and that Plaintiff’s understanding differs from what a POSITA would have understood at the time of the invention. Adopting no construction for “partition,” as Plaintiff proposes, will invite continued dispute over the claims’ scope and confuse the jury. *See O2 Micro Int’l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1361 (Fed. Cir. 2008) (“A determination that a claim term ‘needs no construction’ or has the ‘plain and ordinary meaning’ may be inadequate when a term has more than one ‘ordinary’ meaning or when reliance on a term’s ‘ordinary’ meaning does not resolve the parties’ dispute.”).

Given the dispute over claim meaning and scope, the Court should construe the term. NVIDIA’s construction should be adopted because it is consistent with the intrinsic and extrinsic record, and provides clarity to the technical term “partition” for the jury.

D. “bank” (’438 patent, cls. 6, 20)

NVIDIA’s Proposed Construction	Plaintiff’s Proposed Construction
partition	No construction necessary

The term “bank” requires construction both because it is a technical term and because there is a dispute between the parties about the scope of the term’s ordinary meaning. The dispute between the parties is the same as that for “partition”—Plaintiff’s infringement contentions stretch “bank” beyond its ordinary meaning. ’438 Inf. Cont. (Ex. 12) at 118–30. Because the claims and specification of the ’438 patent use “partition” and “bank” interchangeably, “bank,” as used in the ’438 patent, should be construed to have the same meaning as “partition.” *Wasica Fin. GmbH v. Cont’l Auto. Sys., Inc.*, 853 F.3d 1272, 1282 (Fed. Cir. 2017). NVIDIA’s construction is necessary to resolve the parties’ dispute on this term.

Intrinsic Record: The “specification and claims” of the ’438 patent “use the words ‘[bank]’ and ‘[partition]’ interchangeably;” “[t]his drafting choice equates the two terms for claim

construction purposes.” *Id.* Specifically, as shown below, the patent claims employ “bank” and “partition” identically to recite separate subdivisions of memory designed to hold specified data.

Compare ’438 patent, cl. 6 to ’438 patent, cls. 26 and 27.

'438 Patent, Claim 66	'438 Patent, Claims 26 & 27
6. The computer system of claim 1, wherein accelerator memory comprises: a first memory bank to store parameters common to all of the computations in the sequence of computations; and a second memory bank to store data specific to at least one computation in the sequence of computations.	26. The method of claim 21, further comprising: storing, in a first memory partition of the memory, parameters common to all of the computations in the sequence of computations. 27. The method of claim 26, further comprising: storing, in a second memory partition of the memory, data specific to the first computation in the sequence of computations.

The ’438 patent specification, similarly, refers to the two elements identified by reference numbers 210 and 250 interchangeably as “[t]wo memory banks 210 and 250” and “two partitions 210 250.” ’438 patent at 5:13–52.

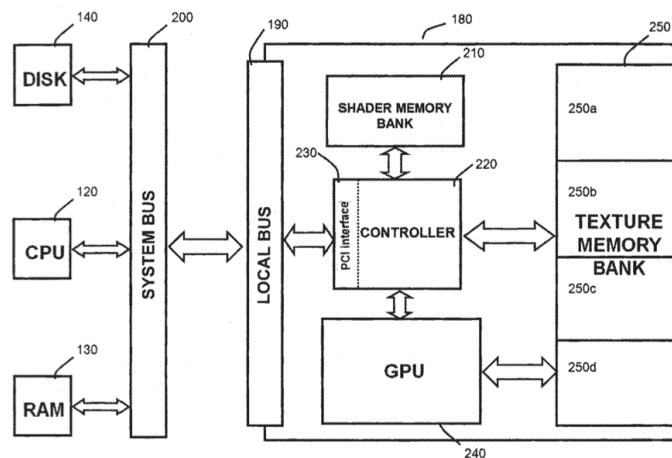


FIG. 2

Id., Fig. 2. In consecutive paragraphs of the specification, the ’438 patent alternates between “partition” and “bank” to refer to the same element. *Id.* at 5:13–52 (“memory banks 210 and 250 . . . partitions 210 250 . . . memory bank 210 . . . memory bank 250”). *See Bid for Position, LLC*

v. AOL, LLC, 601 F.3d 1311, 1317 (Fed. Cir. 2010) (construing “bid” and “value of the bid” identically because the claim used the terms interchangeably); *Edwards Lifesciences LLC v. Cook Inc.*, 582 F.3d 1322, 1329 (Fed. Cir. 2009) (“The interchangeable use of the two terms is akin to a definition equating the two.”)

Although it is a canon of construction that “[w]hen different words or phrases are used in separate claims, a difference in meaning is presumed,” the Federal Circuit has counseled that “simply noting the difference in the use of claim language does not end the matter.” *Nystrom v. TREX Co.*, 424 F.3d 1136, 1143 (Fed. Cir. 2005). Instead, “[d]ifferent terms or phrases in separate claims may be construed to cover the same subject matter where the written description and prosecution history indicate that such a reading of the terms or phrases is proper.” *Id.* Thus, claim differentiation here does not overcome the clear import of the specification. *See Netcraft Corp. v. eBay, Inc.*, 549 F.3d 1394, 1400 n.1 (Fed. Cir. 2008).

Extrinsic Record: Several technical dictionaries confirm that a “bank,” like a “partition,” is a separated subdivision of memory designed to hold specified data. For example, the Comprehensive Dictionary of Electrical Engineering defines “memory bank” as “a *subdivision of memory* that can be accessed independently of (and often in parallel with) other memory banks.” Ex. 9 at 433 (emphasis added); *see also* McGraw-Hill (Ex. 10) at 1308 (defining “memory bank” as “[a] physical section of computer memory . . .”).

E. “swapping the first pointer with the second pointer” (’867 patent, cl. 16)

NVIDIA’s Proposed Construction	Plaintiff’s Proposed Construction
swapping, by the accelerator controller, the first pointer with the second pointer	No construction necessary

During prosecution, the applicant repeatedly argued that this “swapping” step is performed by the accelerator controller in an effort to overcome prior art. That disavowal is consistent with

the other intrinsic evidence. The Court should therefore construe this term consistent with the clear meaning given to it during the prosecution history.

The specification describes various embodiments in which pointers to memory are “swapped.” In all of these embodiments, the step of swapping pointers is performed by the accelerator controller. For example, Fig. 4 is a “detailed flow diagram illustrating” an embodiment of the claimed system and method. ’867 patent at 11:23–42. The “three parallel substreams of execution” shown in FIG. 4—“Computational Substream 403, Data Output Substream 402, and Data Input Substream 404”—are “executed by the [accelerator] controller 220.” *Id.* at 11:43–12:52. One way the claimed method allegedly reduces computational cost is by swapping “pointers for input and output textures after each time step,” allowing the output of one time step to serve as “input during the next time step,” instead of copying the output value over the old input value. *Id.* Flow element 485 of Fig. 4 provides literal support for the limitation in question: “Swap input and output texture pointers.” *Id.* at 11:43–12:52. Flow element 485 occurs as part of the Computational Substream 403, which is executed by the accelerator controller. Thus, in the specification, the “swapping” step is performed by the accelerator controller.

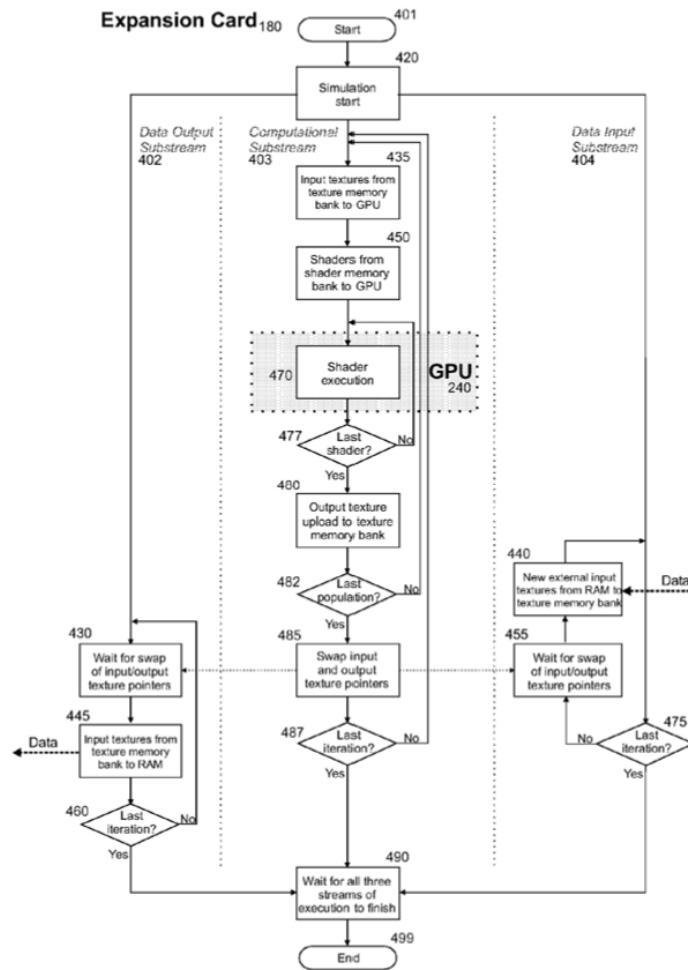


FIG 4

Id., Fig. 4.

During prosecution, the applicant limited the definition of “swapping” in the patent to overcome prior art through a clear disavowal of scope. As explained below, the applicant amended the claims during prosecution to recite the “swapping” step and explicitly stated that it was performed by the accelerator controller. Remarks (Ex. 11) at 10–15. Indeed, the applicant’s basis for distinguishing the prior art was the claimed “accelerator controller” and its recited functionality, including swapping pointers. *See id.*

Specifically, in response to an obviousness rejection based on two prior art patent publications, the applicant argued not only that “the cited references fail to teach or suggest at least

the ‘accelerator controller’” itself but also that the applied prior art references did not “disclose or suggest an accelerator controller . . . to swap the first pointer and the second pointer at the conclusion of the second computational cycle such that the second output data becomes an input for a third computational cycle of the plurality of computational cycles.” *Id.* at 11–13. According to the applicant, “at best the disclosure of [the applied prior art reference] suggests an element that simply moves data into and out of a video card,” but “[t]here is no disclosure whatsoever in [the reference] of an accelerator controller that swaps respective pointers referencing a portion of the input data and the output data . . . as recited in claim 1.” *Id.* The applicant also stated that application claim 17 (which issued as independent claim 16), like application claim 1, had been amended to recite “swapping the first pointer with the second pointer,” and was therefore “patentably distinct from” the applied prior art reference for the same reasons claim 1 was. *Id.* A patentee must “be held to what he declares during the prosecution of his patent.” *Springs Window Fashions L.P. v. Novo Indus., L.P.*, 323 F.3d 989, 995–96 (Fed. Cir. 2003) (construing claims based on disclaimer to exclude a single plate because patentee “explicitly stated during prosecution that his claims differed from a single plate”). At a minimum, these statements are relevant to determining “the scope of the actual invention that is disclosed, described, and patented.” *Fenner Invs., Ltd. v. Cellco P’ship*, 778 F.3d 1320, 1323 (Fed. Cir. 2015) (cleaned up); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1317 (Fed. Cir. 2005) (en banc).

Additionally, the applicant explicitly stated that “the accelerator controller swaps the first pointer and the second pointer” in a summary of the claimed subject matter. *Id.* at 10. While the applicant qualifies this summary as being “provided solely for the convenience of the Examiner” and “not intended to characterize the scope of any claim in any way,” the “totality of the prosecution history” still requires a finding that the accelerator controller performs the “swapping” step. *Computer Docking Station Corp. v. Dell, Inc.*, 519 F.3d 1366, 1379 (Fed. Cir. 2008) (finding

disclaimer where “the sum of the patentees’ statements during prosecution would lead a competitor to believe that the patentee had disavowed” claim coverage). This term should therefore be construed accordingly.

**F. “input data at first rate; and . . . sequence of computations at a second rate”
(’438 patent, cls. 3, 14, 46)**

NVIDIA’s Proposed Construction	Plaintiff’s Proposed Construction
input data at an amount of data received per unit time; and . . . sequence of computations at a number of computations per unit time	No construction necessary

Three claims of the ’438 patent recite two different “rate[s]”: (1) “input data at a first rate,” and (2) “sequence of computations at a second rate.” While the patentee uses these “rates” in their ordinary senses, construction is necessary to ensure that the jury applies the correct claim meaning. *See Abbott Lab’s v. Sandoz, Inc.*, 544 F.3d 1341, 1360 (Fed. Cir. 2008) (citing *Multiform Dessicants, Inc. v. Medzam Ltd.*, 133 F.3d 1473, 1477 (Fed. Cir. 1998), for the proposition that “claims are construed as an aid to the decision-maker, by restating the claims in non-technical terms”).

1. “input data at a first rate”

The intrinsic record establishes that the patentee used the term “first rate” in its ordinary, quantitative sense—*i.e.*, to describe how much data is transferred to the CPU or GPU over a specific amount of time. For example, in describing the “local bus” that transfers data from the CPU to the accelerator, the ’438 patent states that “the local bus has maximal peak data transfer rates of 4 GB/s (PCI Express 16) or 2 GB/s (PCI-X 2.0).” ’438 patent at 4:34–36; *see also id.* at 4:31–33 (system bus bandwidth from 3.2 GB/s to 15 GB/s). Thus, in the specification’s examples, the rate of data input (*e.g.*, into or from the CPU) is expressed as an amount of data received (*e.g.*, in gigabytes) per unit time (*e.g.*, in seconds).

Interpreting “input data at a first rate” as “input data at an amount of data received per unit time” also aligns with the ordinary meaning of an “input rate” in the art of computer architecture, where the term denotes a quantity per time dimension—such as megabytes or gigabytes per second—used to evaluate bus bandwidth, memory bandwidth, or data-ingest capacity. Nothing in the claims, specification, or prosecution history suggests that the inventors intended to redefine “rate” to mean anything other than this conventional throughput measure. By construing the phrase to capture the clear notion of data transfer throughput, the Court will assist the jury in understanding the rate of input data in these claims.

2. “sequence of computations at a second rate”

Although the specification does not provide any examples of performing a “sequence of computations at a second rate,” *see generally* ’438 patent, the ordinary meaning of this phrase is that a number of computations are performed in every unit of time. This interpretation aligns with the common understanding of the term “rate” in technical and scientific fields, where it typically denotes a quantity measured against a specific time interval, with the “quantity” here being the number of computations. By defining “sequence of computations at a second rate” as “sequence of computations at a number of computations per unit time,” the Court will ensure clarity and precision in the claim language, which is crucial for the jury to accurately assess issues of infringement and validity.

G. Order of Steps (’867 patent, cl. 16; ’438 patent, cls. 12, 21)

NVIDIA’s Proposed Construction	Plaintiff’s Proposed Construction
steps must be performed in the order listed	No construction necessary

Three asserted claims are method claims that implicitly require an order, with each step building on the last. These claims should be construed to require that they be performed in the order written. *Interactive Gift Exp., Inc. v. Compuserve Inc.*, 256 F.3d 1323, 1342 (Fed. Cir. 2001).

1. Claim 16 of the '867 Patent

For this claim, the “sequential nature of the claim steps is apparent from the plain meaning of the claim language, and nothing in the written description suggests otherwise.” *Mantech Env't Corp. v. Hudson Env't Servs., Inc.*, 152 F.3d 1368, 1376 (Fed. Cir. 1998). The “sequential nature” of this claim is summarized in the chart below.

'867 Patent, Claim 16 Steps	Relation to Other Step(s)
receiving, by an accelerator, first input data from the central processing unit	
transferring, by an accelerator controller, the first input data into a first partition, referenced by first pointer, of an accelerator memory before a first computational cycle of the numerical simulation	The prior step transfers first input data to the accelerator; this step operates on first input data in the accelerator (<i>e.g.</i> , transferring it into a first partition).
performing, by at least one graphics processing unit during the first computational cycle , at least one calculation on the first portion of the input data as to generate first output data	The prior step occurs <i>before</i> a first computational cycle; this step occurs <i>during</i> the first computational cycle.
storing, by the accelerator controller, the first output data into a second partition, referenced by a second pointer , of the accelerator memory	The prior step <i>generates</i> first output data; this step <i>stores</i> first output data.
swapping the first pointer with the second pointer at the end of the first computational cycle, such that the first output data becomes an input for a second computational cycle of the numerical simulation	The prior step <i>introduces</i> a second partition referenced by a second pointer; this step <i>operates</i> on the second pointer (<i>e.g.</i> , by swapping it with the first).

Thus, as shown above, “as a matter of logic [and] grammar,” these steps “must be performed in the order written.” *Altiris, Inc. v. Symantec Corp.*, 318 F.3d 1363, 1369 (Fed. Cir. 2003).

This ordering of steps is confirmed by the specification. In the patent’s embodiments, the relevant steps of data input and computations proceed as follows (*see generally* '867 patent at Figs. 3–4):

1. The CPU transfers data to expansion card 180 (*i.e.*, the accelerator) (*id.* at 8:6–10);

2. Expansion card 180 (*i.e.*, the accelerator) stores the data in a partition in texture memory bank 250 (*id.* at 8:10–11, 8:17–20);
3. Computation in the GPU of the accelerator begins, generating output data (textures) stored in “the output portion of the texture memory bank” (*i.e.*, a second partition) (*id.* at 8:50–53, 11:51–60); and
4. Pointers for input and output textures are swapped “after each time step” (*id.* at 11:38–42).

Because the plain language of the claims implies an order and nothing in the specification compels otherwise, claim 16 should be construed to require the steps occur in the order listed.

2. Claims 12 and 21 of the '438 Patent

Like claim 16 of the '867 patent, claims 12 and 21 of the '438 patent require a specific order of steps. The order of steps is dictated both by the plain language of the claims and the physical realities of these steps. *See Kaneka Corp. v. Xiamen Kingdomway Group Co.*, 790 F.3d 1298, 1306 (Fed. Cir. 2015).

'438 Patent, Claim 12 Steps	Relation to Other Step(s)
(A) performing, by the GPU, the sequence of computations on a first portion of the input data so as to generate a first portion of the output data, the first portion of the output data representing an output of a neuron in a first layer of the artificial neural network . . . [and] accumulating a first value for the output variable in the first texture during a first time step	
(B) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the input data from the main memory to the accelerator via the bus	While part of this step can occur “in parallel” with step (A), the <i>second</i> portion of input data is transferred after a <i>first</i> portion of input data.

'438 Patent, Claim 12 Steps	Relation to Other Step(s)
(C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus, the second portion of the output data representing an output of a neuron in a second layer in the artificial neural network	While part of this step can occur “in parallel” with step (A), the <i>second</i> portion of output data is transferred after a <i>first</i> portion of output data. As a physical reality, an output of a neuron in a <i>second</i> layer of an artificial neural network occurs after an output of a neuron in a <i>first</i> layer.
(D) performing, by the GPU, the sequence of computations on the second portion of the input data , wherein performing the sequence of computations on the second portion of the input data comprises (i) accumulating a second value for the output variable in the second texture during a second time step and (ii) making the first value of the output variable in the first texture accessible to other computational elements in the plurality of computational elements during the second time step	Computations on the <i>second</i> portion of input data cannot occur until after the data is transferred, as in step (B). The <i>second</i> time step of step (D) is necessarily after the <i>first</i> time step of step (A).

'438 Patent, Claim 21 Steps	Relation to Other Step(s)
receiving . . . first input data acquired from an external system in real time	
initializing, by a controller operably coupled to a graphics processing unit (GPU), textures and shaders in a memory operably coupled to the GPU	This step must occur before any computations, though it does not necessarily have to occur after first input data is received in the first step.
transferring the first input data received by the CPU to the memory operably coupled to the GPU	The first step <i>receives</i> input data at the CPU; this step <i>transfers</i> the input data from the CPU to memory coupled to the GPU.
performing, by the graphics processing unit (GPU), a first computation in the sequence of computations on the first input data based on the textures and shaders to generate first output data . . .	The prior step <i>transfers</i> first input data to the GPU; this step <i>operates</i> on first input data in the GPU. This step must also expressly occur <i>after</i> initialization of textures and shaders in the second step.
storing . . . the first input data and the first output data	The prior step <i>generates</i> output data; this step <i>stores</i> output data.

'438 Patent, Claim 21 Steps	Relation to Other Step(s)
transferring second input data acquired from the external system in real time into the memory operably coupled to the GPU after the GPU starts the first computation and before the GPU starts a second computation of the sequence of computations . . .	This step must occur <i>after</i> the GPU starts computations, and it operates on “second input data” acquired in real-time, which comes <i>after</i> first input data.

Nothing in the specification contracts the order of steps dictated by claims 12 and 21. In connection with Figure 4, the '438 patent shows the sequence (as read from the top of the chart to the bottom) of (1) inputting textures (data) (*e.g.*, step 435); executing shaders on first input data (*e.g.*, step 470); (3) outputting textures (resulting data) (*e.g.*, step 480); (4) inputting additional textures (data) (*e.g.*, steps 440, 445); (5) executing shaders on newly inputted data (*e.g.*, step 470, *after* the determination of “Last iteration?” is “No”). *See* '438 patent at 11:23–13:28, Fig. 4. While portions of these steps can occur in parallel (as recited in claim 12), the sequence of steps shown in Figure 4 confirm that certain steps must occur (or at least begin) before others as a practical reality.

Thus, because the “sequential nature of the claim steps is apparent from the plain meaning of the claim language and nothing in the written description suggests otherwise,” these claims should be construed to require the order explained above. *Mantech Env't*, 152 F.3d at 1376.

IV. CONCLUSION

For the foregoing reasons, NVIDIA respectfully requests that the Court adopt NVIDIA's proposed constructions.

Dated: May 20, 2025

Respectfully submitted,

/s/ L. Kieran Kieckhefer

L. Kieran Kieckhefer (*pro hac vice*)
Jaysen S. Chung (*pro hac vice*)
GIBSON, DUNN & CRUTCHER LLP
One Embarcadero Center
San Francisco, CA 94111-3715
(415) 393-8200
kkieckhefer@gibsondunn.com
jschung@gibsondunn.com

Brian A. Rosenthal (*pro hac vice*)
Ahmed ElDessouki (*pro hac vice*)
Vivian Lu (*pro hac vice*)
GIBSON, DUNN & CRUTCHER LLP
200 Park Avenue
New York, NY 10166-0193
(212) 351-4000
broenthal@gibsondunn.com
aeldessouki@gibsondunn.com
vlu@gibsondunn.com

Brian M. Buroker (*pro hac vice*)
GIBSON, DUNN & CRUTCHER LLP
1700 M Street, NW
Washington, DC 20036-4504
(202) 955-8541
bburoker@gibsondunn.com

Casey J. McCracken (*pro hac vice*)
GIBSON, DUNN & CRUTCHER LLP
3161 Michelson Drive
Irvine, CA 92612-4412
(949) 451-3932
cmccracken@gibsondunn.com

Nathan R. Curtis (*pro hac vice*)
GIBSON DUNN & CRUTCHER LLP
2001 Ross Avenue
Dallas, TX 75201-2923
(214) 698-3423
ncurtis@gibsondunn.com

Lillian J. Mao (*pro hac vice*)
GIBSON DUNN & CRUTCHER LLP
310 University Avenue
Palo Alto, CA 94301-1744
(650) 849-5307
lmao@gibsondunn.com

Barry K. Shelton (TX State Bar No. 24055029)
SHELTON COBURN LLP
311 RR 620 S, Suite 205
Austin, TX 78734-4775
(512) 263-2165
bshelton@sheltoncoburn.com

Attorneys for Defendant NVIDIA Corporation

CERTIFICATE OF SERVICE

I hereby certify that all counsel of record are being served with a copy of the foregoing sealed documents via electronic mail and publicly filed documents via the Court's CM/ECF system on May 20, 2025.

/s/ L. Kieran Kieckhefer
L. Kieran Kieckhefer

Exhibit 1

Chambers Dictionary of Science and Technology

General Editor
John Lackie

Chambers

Neural AI, LLC, Exhibit 2010

Page 2010 - 35

IPR2025-00609, NVIDIA Corporation v. Neural AI, LLC

CHAMBERS

An imprint of Chambers Harrap Publishers Ltd
7 Hopetoun Crescent
Edinburgh
EH7 4AY

This edition first published by Chambers Harrap Publishers Ltd 2007

© Chambers Harrap Publishers Ltd 2007

Previous edition published in 1999. Previously published in 1995 as *Larousse Dictionary of Science and Technology*. First published as *Chambers's Technical Dictionary* in 1940 by W&R Chambers Ltd (revised 1958, 1971, 1974 and 1984).

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying or otherwise, without the prior permission of the publisher.

A CIP catalogue record for this book is available from the British Library.

ISBN-13: 978 0550 100719

Text design by Ken Wilson
Typeset by Macmillan India Ltd
Printed in Great Britain by Clays Ltd, St Ives plc

ac- (*Chem*) Abbrev indicating substitution in the alicyclic ring.

acacia (*For*) See MYRTLE.

acacia gum (*Chem*) See GUM ARABIC.

acalculia (*Psych*) Inability to make simple mathematical calculations. Also *dyscalculia*.

acamprostate (*Pharmacol*) A drug thought to be beneficial in maintaining abstinence in alcohol-dependent patients.

Acanthamoeba (*Med*) A protozoon which can survive in tapwater and may cause KERATITIS if contact lenses are washed in tapwater.

acanthite (*Min*) An ore of silver, Ag_2S , crystallizing in the monoclinic system. Cf ARGENTITE.

acantho- (*Genrl*) Prefix from Gk *akantha*, spine, thorn.

Acanthocephala (*BioSci*) A phylum of elongate worms with a rounded body and a protrusible proboscis, furnished with recurved hooks. There is no mouth or alimentary canal. The young stages are parasitic in various Arthropoda, the adults in fish and aquatic birds and mammals. Thorny-headed worms.

acanthoma (*Med*) A tumour of epidermal cells.

acanthosis nigricans (*Med*) A rare disease characterized by pigmentation and warty growths on the skin, often associated with cancer of the stomach or uterus.

acariasis (*Vet*) Contagious skin disease caused by mites (*acari*).

Acarina (*BioSci*) An order of small Arachnida, with globular, undivided body. The immature stages (hexapod larvae) have six legs. The order is a large worldwide group, occupying all types of habitat, and of great economic importance. Many are ectoparasitic. Mites and ticks.

acarophily (*BioSci*) A symbiotic association between plants and mites. Also *acarophytism*.

acatamnesia (*Psych*) The inability to comprehend data (objects, language, etc) presented to the senses.

acaulescent (*BioSci*) Having a short stem.

acauline (*BioSci*) Stemless or nearly so. Also *acaulose*.

ac balancer (*ElecEng*) An arrangement of transformers or reactors used to equalize the voltages between the wires of a multiple-wire system. Also *static balancer*.

ac bias (*Electronics*) A high-frequency signal applied to a magnetic tape recording head along with the signal to be recorded. This stabilizes magnetic saturation and improves frequency response, at the same time reducing noise and distortion. The bias signal frequency has to be many times the highest recording frequency.

AC-boundary layer (*Acous, Aero*) See STOKES LAYER.

accelerated ageing test (*ElecEng*) A stability test for cables using twice normal working voltage. It is claimed this gives quick results that correlate with service records.

accelerated fatigue test (*Eng*) Test which applies a cyclic loading schedule, which can be of varying frequency and/or amplitude, to a machine or component simulating its loading in service, but at a higher rate, to determine its safe fatigue life before it is reached in service. See panel on FATIGUE.

accelerated freeze-drying (*FoodSci*) FREEZE-DRYING in which the final sublimation under vacuum is accelerated by heat. Abbrev AFD.

accelerate-stop distance (*Aero*) The total distance, under specified conditions, in which an aircraft can be brought to rest after accelerating to CRITICAL SPEED for an engine failure at take-off.

accelerating chain (*Electronics*) The section of an electron beam tube or system, eg cathode-ray tube or electron microscope, in which electrons are accelerated by voltages on accelerating electrodes. Also used in particle accelerators.

accelerating electrode (*Electronics*) An electrode in a thermionic valve or cathode-ray tube maintained at a high positive potential with respect to the electron source. It accelerates electrons in their flight to the anode but does not collect a high proportion of them.

accelerating machine (*Electronics*) See ACCELERATOR.

accelerating potential (*Electronics*) The potential applied to an electrode to accelerate electrons from a cathode.

acceleration (*Phys*) The rate of change of velocity, expressed in metres (or feet) per second squared. It is a vector quantity and has both magnitude and direction.

acceleration due to gravity (*Phys*) Acceleration with which a body would fall freely under the action of gravity in a vacuum. This varies according to the distance from the Earth's centre, but the internationally adopted value is $9.806\,65\text{ m s}^{-2}$ or 32.1740 ft s^{-2} . Abbrev *g*. See HELMERT'S FORMULA.

acceleration error (*Aero*) The error in an airborne magnetic compass due to manoeuvring; caused by the vertical component of the Earth's magnetic field when the centre of gravity of the magnetic element is displaced from normal.

acceleration stress (*Space*) The influence of acceleration (or deceleration) on certain physiological parameters of the human body. Humans can withstand transverse accelerations better than longitudinal ones, which have a profound effect on the cardiovascular system. The degree of tolerance also depends on the magnitude and duration of the acceleration.

acceleration tolerance (*Space*) The maximum acceleration force that an astronaut can withstand before 'blacking out' or otherwise losing control.

accelerator (*Aero*) A device, similar to a CATAPULT, but generally mounted below deck level, for assisting the acceleration of aircraft flying off aircraft carriers. Land versions have been tried experimentally.

accelerator (*Autos*) A pedal connected to the carburettor throttle valve of a motor vehicle, or to the fuel injection control.

accelerator (*BioSci*) Any muscle or nerve that increases rate of action.

accelerator (*Build*) A hardener or catalyst mixed with synthetic resins in TWO PACK MATERIALS to speed up the hardening rate.

accelerator (*Chem*) (1) A substance which increases the speed of a chemical reaction. See CATALYSIS. (2) A substance which increases the efficient action of an enzyme. (3) Any substance effecting acceleration of the vulcanization process of rubber. The principal types are aldehyde derivatives of Schiff's bases: butyraldehyde-butylidene-aniline, di-orthotolyl-guanidine, diphenyl-guanidine, benzthiazyl disulphide, tetramethyl-thiuran disulphide and zinc dimethyl-dithiocarbamate.

accelerator (*CivEng*) Any substance mixed with cement concrete for the purpose of hastening hardening.

accelerator (*Electronics*) Machine used to accelerate charged particles to very high energies. See BETATRON, CYCLOTRON, LINEAR ACCELERATOR, SYNCHROCYCLOTRON, SYNCHROTRON.

accelerator (*ICT*) A special circuit board that is placed within a computer to speed up some aspect of its operation.

accelerator (*ImageTech*) A chemical used to increase the rate of development; eg sodium carbonate or borax.

accelerator pump (*Autos*) A small cylinder and piston fitted to some types of carburettor, and connected to the throttle so as to provide a momentarily enriched mixture when the engine is accelerated.

accelerometer (*Acous, Electronics*) A transducer used to provide a signal proportional to the rate of acceleration of a vibrating; or other body, usually employing the piezo-electric principle. See PICK-UP HEAD.

accelerometer (*Aero*) An instrument, carried in aircraft, guided missiles and spacecraft, for measuring acceleration in a specific direction. Main types are *indicating*, *maximum-reading*, *recording* (graphical) and *counting* (digital, totalling all accelerations above a set value). See IMPACT ACCELEROMETER, VERTICAL-GUST RECORDER.

acceptable daily intake (*FoodSci*) Estimated daily intake in mg kg^{-1} of a substance (eg additive, residue) regarded as having no obvious harmful effect to humans during their lifespan.

partial veil

873

Pascal's triangle

partial veil (*BioSci*) In some basidiomycete fruiting bodies, eg mushrooms, a membrane joining the edge of the cap to the stalk, rupturing to leave an annulus. Cf **UNIVERSAL VEIL**.

particle (*MinExt*) A single piece of solid material, usually defined (when small) by its mesh, or size passing through a specified size of sieve.

particle (*Phys*) (1) A useful concept of a small body which has a finite mass but negligible dimensions so that it has no MOMENT OF INERTIA about its CENTRE OF MASS. (2) A volume of air or fluid which has dimensions very small compared with the wavelength of a propagated sound wave, but large compared with molecular dimensions.

particle accelerator (*Electronics*) See **ACCELERATOR**.

particle exchange (*Phys*) The interaction between fundamental particles by the exchange of another fundamental particle. See **GAUGE BOSON**.

particle-induced X-ray emission (*Phys*) X-rays emitted and studied when a sample has been bombarded with protons or heavy ions. Abbrev **PIXE**. See panel on **TRACE ELEMENT ANALYSIS**.

particle mean size (*PowderTech*) The dimension of a hypothetical particle such that, if a powder were wholly composed of such particles, such a powder would have the same value as the actual powder in respect of some stated property. See **PARTICLE SIZE**.

particle physics (*Phys*) The study of the properties of fundamental particles and of fundamental interactions. Also *high-energy physics*.

particle porosity (*PowderTech*) The ratio of the volume of open pores to the total volume of the particle. See **POROSITY**.

particle scattering (*Phys*) See **SCATTERING**.

particle size (*Geol*) The general dimensions of grains in a rock, esp a sediment. Many definitions have been used. One of the more common is the **WENTWORTH SCALE** (see table). Field definitions are a little less precise. 'If the grains can be distinguished then it is at least silt grade; if it doesn't feel gritty on the teeth, then it is clay.' Also *grain size*.

Dimension	Name
greater than 256 mm	boulder
64–256 mm	cobble
4–64 mm	pebble
2–4 mm	gravel
$\frac{1}{16}$ –2 mm	sand
$\frac{1}{256}$ – $\frac{1}{16}$ mm	silt
less than $\frac{1}{256}$ mm	clay

particle size (*PowderTech*) The magnitude of some physical dimension of a particle of powder. When the particle is a sphere, it is possible to define the size uniquely by a unit of length. For a non-spherical particle, it is not possible to define a size without specifying the method of measurement. See **PARTICLE MEAN SIZE**.

particle size analysis (*PowderTech*) The study of methods for determining the physical structure of powdered materials.

particle velocity (*Acous*) In a progressive or standing sound wave, the alternating velocity of the particles of the medium, taken as either the maximum or rms velocity.

particular average (*Ships*) Loss or damage to marine property whose cost is borne by the owner (unless insured against it).

particular integral (*MathSci*) Generally, a solution of a differential equation formed by assigning values to the arbitrary constants in the complete primitive. Non-singular solution of a differential equation containing no arbitrary constants.

particulates (*MinExt*) Microscopic airborne material such as sand and volcanic ash but also industrial dust from power stations and other manufacturing processes.

parting bead (*Build*) A bead fixed to the cased frame of a double-hung window to separate the inner and outer sashes.

parting of bullion (*Eng*) See **INQUARTATION**.

parting sand (*Eng*) Dry sand sprinkled on the parting face of a mould to prevent adhesion of the two surfaces at the joint when the cope is rammed.

parting slip (*Build*) A thin lath of wood or zinc which keeps the sash weights apart within the cased frame of a double-hung window. Also *mid-feather*, *wagtail*.

partite (*BioSci*) Split almost to the base.

partition (*ICT*) (1) To subdivide the available storage space on a HARD-DISK DRIVE. Each section may be used to hold programs and data for different OPERATING SYSTEMS or to provide additional LOGICAL DRIVES. (2) To subdivide the WORKING STORE such that several APPLICATION PROGRAMS may be stored simultaneously and each be executed for a short period of time in a system using a MULTI-PROGRAMMING OPERATING SYSTEM.

partition (*MathSci*) (1) The division of a set into a collection of disjoint subsets so that their union is the whole set; membership of the same subset is then an EQUIVALENCE RELATION. (2) A decomposition of a positive number n into the sum of positive integers $n = a_1 + a_2 + \dots + a_r$, where $a_1 \geq a_2 \geq \dots \geq a_r$; eg the partitions of 4 are 4, 3+1, 2+2, 2+1+1 and 1+1+1+1.

partition chromatography (*Chem*) See **CHROMATOGRAPHY**.

partition coefficient (*Chem*) The ratio of the equilibrium concentrations of a substance dissolved in two immiscible solvents. If no chemical interaction occurs, it is independent of the actual values of the concentrations. Also *distribution coefficient*.

partition noise (*Electronics*) That arising when electrons are abstracted from a stream by a number of successive electrodes, as in a travelling-wave tube.

partition plate (*Build*) The upper horizontal member of a wooden partition, capping the studding and providing support for joists etc. Also *head piece*.

partridge disease (*Vet*) A popular term for infection of the gut of partridges by the nematode worm *Trichostrongylus tenuis*.

parts rationalization (*Eng*) Design method which seeks to minimize the number of parts in a specific product, thereby saving assembly costs and material, eg in replacing many metal parts with a single injection moulding.

parturient (*Med*) Of or pertaining to parturition.

parturient fever (*Vet*) Also *parturient eclampsia*, *parturient paresis*. See **MILK FEVER**.

parturition (*BioSci*) In viviparous animals, the act of bringing forth young.

paruresis (*Med*) Difficulty in urinating in the presence of other people.

parvifoliate (*BioSci*) Having leaves that are small in relation to the size of the stem.

parvovirus (*Vet*) See **CANINE PARVOVIRUS INFECTION**, **PORCINE PARVOVIRUS INFECTION**.

Parylene (*Plastics*) **TN** for POLYXYLYLENE film.

PAS (*BioSci*) (1) Abbrev for *periodic acid-Schiff* reagent. (2) See **PARA(4)-AMINOSALICYLIC ACID**.

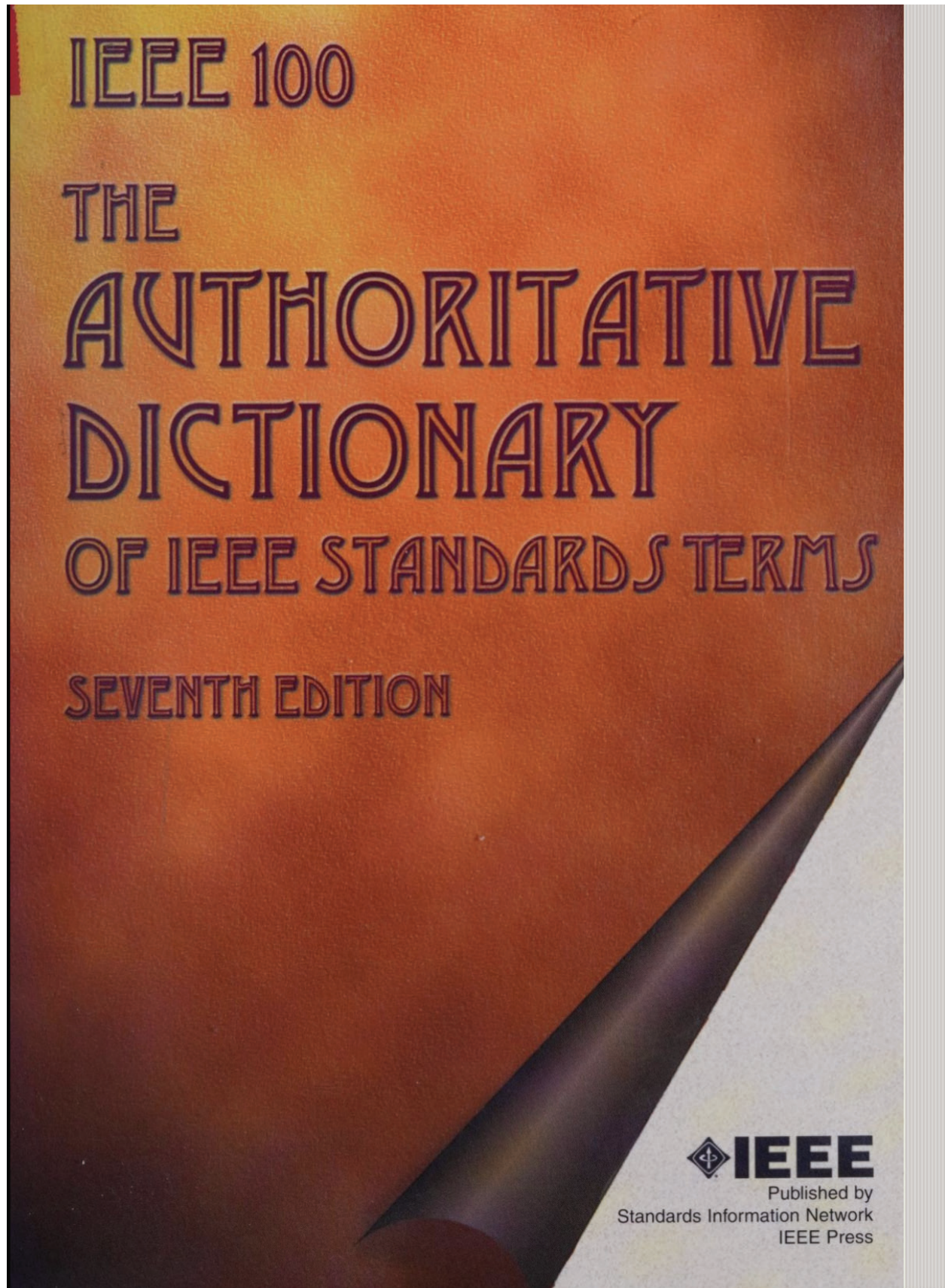
PASCAL (*ICT*) Programming language designed to encourage structured programming.

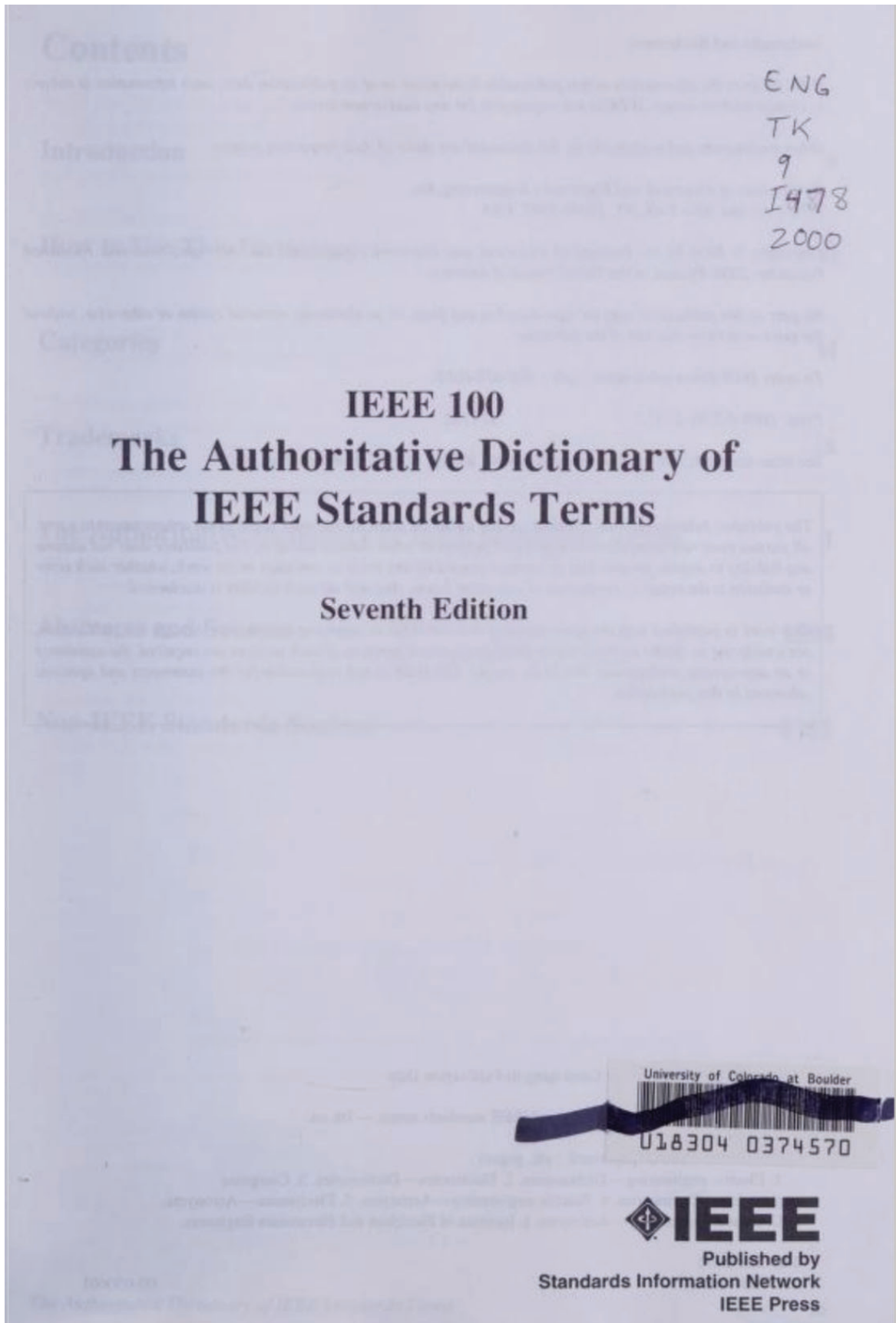
pascal (*Genrl*) The SI derived unit of pressure or stress, equal to 1 newton per square metre. Symbol **Pa**.

Pascal's theorem (*MathSci*) The theorem that the intersections of pairs of opposite sides of a hexagon inscribed in a conic are collinear. Cf **BRIANCHON'S THEOREM**, **PAPPUS'S THEOREM** (2).

Pascal's triangle (*MathSci*) An easily remembered summary of the **BINOMIAL COEFFICIENTS** in the expansion of $(1+x)^n$ for $n=0, 1, 2, 3, \dots$. Starting with a single 1 in the top line, each digit is the sum of the two above and on

Exhibit 2





Trademarks and disclaimers

IEEE believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. IEEE is not responsible for any inadvertent errors.

Other tradenames and trademarks in this document are those of their respective owners.

*The Institute of Electrical and Electronics Engineering, Inc.
3 Park Avenue, New York, NY, 10016-5997, USA*

Copyright © 2000 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Published December 2000. Printed in the United States of America.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

To order IEEE Press publications, call 1-800-678-IEEE.

Print: ISBN 0-7381-2601-2

SP1122

See other standards and standards-related product listings at: <http://standards.ieee.org/>

The publisher believes that the information and guidance given in this work serve as an enhancement to users, all parties must rely upon their own skill and judgement when making use of it. The publisher does not assume any liability to anyone for any loss or damage caused by any error or omission in the work, whether such error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

This work is published with the understanding that the IEEE is supplying information through this publication, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. The IEEE is not responsible for the statements and opinions advanced in this publication.

Library of Congress Cataloging-in-Publication Data

IEEE 100 : the authoritative dictionary of IEEE standards terms.—7th ed.
p. cm.

ISBN 0-7381-2601-2 (paperback : alk. paper)

1. Electric engineering—Dictionaries. 2. Electronics—Dictionaries. 3. Computer engineering—Dictionaries. 4. Electric engineering—Acronyms. 5. Electronics—Acronyms. 6. Computer engineering—Acronyms. I. Institute of Electrical and Electronics Engineers.

TK9 .J28 2000
621.3'03—dc21

00-050601

abstraction

4

accelerometer

operations are realized. The data is "abstract" in the sense that values in the extent, i.e., the concrete values that represent the instances, are any set of values that support the operations and are irrelevant to the user. An abstract data type defines the operations on the data as part of the definition of the data and separates what can be done (interface) from how it is done (realization). (C/SE) 1320.2-1998

abstraction (A) A view of an object that focuses on the information relevant to a particular purpose and ignores the remainder of the information. *See also:* data abstraction. (B) The process of formulating a view as in (A). (C) 610.12-1990

abstract machine (A) (software) A representation of the characteristics of a process or machine. (B) (software) A module that processes inputs as though it were a machine. *See also:* module; process. (C/SE) 729-1983

abstract quantity *See:* mathematico-physical quantity.

abstract symbol A symbol whose meaning and use have not been determined by a general agreement but have to be defined for each application of the symbol. (C) 1084-1986w

Abstract Syntax Notation One (ASN.1) A notation that both enables complicated types to be defined and also enables values of these types to be specified. (C/PA) 1328.2-1993w, 1224.2-1993w, 1327.2-1993w, 1326.2-1993w

ac (alternating current) *See:* alternating current.

AC *See:* acoustic coupler.

ACA *See:* adjacent-channel attenuation.

academic simulation *See:* instructional simulation.

ac analog computer An analog computer in which electrical signals are of the form of carrier signals where the absolute value of a mathematical variable is represented by the amplitude of the carrier and the sign of the mathematical variable is represented by the phase (0 or 180 degrees) of the carrier relative to the computer. (C) 610.10-1994w, 165-1977w

ACAR *See:* aluminum conductor, aluminum alloy reinforced.

ac breakdown voltage (gas tube surge-protective device) The minimum root-mean-square value of sinusoidal voltage at frequencies between 15 Hz and 62 Hz that results in arrester sparkover. (SPD/PE) C62.31-1981s

ac cable (armored cable) A fabricated assembly of insulated conductors in a flexible metallic enclosure. (NESC/NEC) [86]

accelerated aging The application of intensified aging stress or stresses in order to increase the degradation rate above that expected in service. (DEL/RE) 775-1993w

accelerated life test (test, measurement, and diagnostic equipment) A test in which certain factors, such as voltage, temperature, and so forth, are increased or decreased beyond normal operating values to obtain observable deterioration in a reasonable period of time, and thereby afford some measure of the probable life under normal operating conditions or some measure of the durability of the equipment when exposed to the factors being aggravated. (MIL) [2]

accelerated test (evaluation of thermal capability) (thermal classification of electric equipment and electrical insulation) A functional test in which one or more factors of influence are increased in magnitude or frequency of application so as to decrease the time needed for the test. (EI) 1-1986r

accelerating (rotating machinery) The process of running a motor up to speed after breakaway. *See also:* asynchronous machine. (PE) [9]

accelerating device (power system device function numbers) A device that is used to close or to cause the closing of circuits that are used to increase the speed of a machine. (PE/SUB) C37.2-1979s

accelerating electrode An electrode to which a potential is applied to increase the velocity of the electrons or ions in the beam. (NPS) 61-1971w, 398-1972r

accelerating grid *See:* accelerating electrode.

accelerating relay A programming relay whose function is to control the acceleration of rotating electrical equipment. (SWG/PE) C37.100-1992

accelerating time (control) The time in seconds for a change of speed from one specified speed to a higher specified speed while accelerating under specified conditions. *See also:* electric drive. (IA/ICTL/APP/IAC) [69], [60]

accelerating torque (rotating machinery) Difference between the input torque to the rotor (electromagnetic for a motor or mechanical for a generator) and the sum of the load and loss torques; the net torque available for accelerating the rotating parts. *See also:* rotor. (PE) [9]

accelerating voltage (oscilloscopes) The cathode-to-viewing-area voltage applied to a cathode-ray tube for the purpose of accelerating the electron beam. *See also:* oscillograph. (IM/HFIM) [40]

acceleration (electric drive) Operation of raising the motor speed from zero or a low level to a higher level. *See also:* electric drive. (IA/IAC) [60]

acceleration factor The ratio between the times necessary to obtain the same stated proportion of failures in two equal samples under two different sets of stress conditions involving the same failure modes and mechanisms. (R) [29]

acceleration-forced response (automatic control) The total (transient plus steady-state) time response resulting from a sudden increase in the rate of the rate of change of input from zero to some finite value. (PE/EDPG) [3]

acceleration-insensitive drift rate (gyros) The component of systematic drift rate that has no correlation with acceleration. *See also:* systematic drift rate. (AES/GYAC) 528-1994

acceleration, programmed *See:* programmed acceleration.

acceleration-sensitive drift rate (gyros) Those components of systematic drift rates that are correlated with the first power of linear acceleration applied to the gyro case. The relationship of these components of drift rate to acceleration can be stated by means of coefficients having dimensions of angular displacement per unit time per unit acceleration for accelerations along each of the principal axes of the gyro (for example, drift rate caused by mass unbalance). *See also:* systematic drift rate. (AES/GYAC) 528-1994

acceleration space (velocity-modulated tube) The part of the tube following the electron gun in which the emitted electrons are accelerated to reach a determined velocity. *See also:* velocity-modulated tube. (ED) [45], [84]

acceleration-squared-sensitive drift rate (gyros) Those components of systematic drift rates that are correlated with the second power or product of linear accelerations applied to the gyro case. The relationship of these components of drift rate to acceleration can be stated by means of coefficients having dimensions of angular displacement per unit time per unit acceleration squared for accelerations along each of the principal axes of the gyro and angular displacement per unit time per the product of accelerations along combinations of two principal axes of the gyro (for example, drift rate caused by anisotropy). (AES/GYAC) 528-1994

acceleration time The part of access time that is required to bring a storage device, typically a tape or disk drive, to the speed at which data can be read or written. *Synonym:* start time. *Contrast:* deceleration time. (C) 610.10-1994w

acceleration, timed *See:* timed acceleration.

accelerator (1) An Xt Intrinsic facility that allows the binding of a widget event to a keyboard action or a series of actions. (C) 1295-1993w

(2) A circuit or device that accelerates some unit in a computer, as in an accelerator board. *See also:* hardware accelerator. (C) 610.10-1994w

accelerator board A printed circuit board that replaces or augments the computer's main processor with a faster processor. (C) 610.10-1994w

accelerometer A device that senses the inertial reaction of a proof mass for the purpose of measuring linear or angular acceleration. *Note:* In its simplest form, an accelerometer con-

partial sum

797

parts program

cation of partial-read pulses or partial-write pulses. *See also:* coincident-current selection. (Std100) 163-1959

partial sum The result obtained from the addition of two or more numbers without regard to carries. *Note:* In the binary numeration system, the partial sum is the same result as is obtained from the exclusive-OR operation. *See also:* cascaded carry. (C) 1084-1986w

partial system downtime A weighted time out of service for switching system failures that put a number of lines out of service simultaneously. This measure is the expected probability that two randomly chosen lines will be out of service simultaneously due to the same fault cause. It may be expressed in minutes per year. It protects customers under circumstances that are not fully covered by individual line downtime or total system downtime. *Synonym:* simultaneous line downtime. (COM/TA) 973-1990w

partial-write pulse Any one of the currents applied that cause selection of a cell for writing. *See also:* coincident-current selection. (Std100) 163-1959w

participant property A kind of property of a state class that reflects that class' knowledge of a relationship in which instances of the class participate. When a relationship exists between two state classes, each class contains a participant property for that relationship. A participant property is a mapping from a state class to a related (not necessarily distinct) state class. The name of each participant property is the name of the role that the other class plays in the relationship, or it may simply be the name of the class at the other end of the relationship (as long as using the class name does not cause ambiguity). A value of a participant property is the identity of a related instance. (C/SE) 1320.2-1998

participate With regard to the action of an S-module during message transmission, to execute the command contained in the HEADER packet of the current message and return packets as required by that command and by the state of the Acknowledge bit in the HEADER packet. The handling of some errors may cause an S-module to cease to participate in a message (e.g., by ceasing to execute the current command, by returning NULL packets when data is expected, by driving a constant value on the MSD signal without regard to packet transmission timing, etc.). (TT/C) 1149.5-1995

participating slave A slave involved in a transaction as either a selected slave, an intervening slave, a broadcast slave, or a slave involved in multiple packet mode. (C/BA) 10857-1994, 896.3-1993w, 896.4-1993w

particle accelerator Any device for accelerating charged particles to high energies, for example, cyclotron, betatron, Van der Graaff generator, linear accelerator, etc. (ED) [45]

particle size distribution The probability density function describing the size distribution of particles in a medium. (AP/PROP) 211-1997

particle velocity (sound field) The velocity of a given infinitesimal part of the medium, with reference to the medium as a whole, due to the sound wave. *Note:* The terms instantaneous particle velocity, effective particle velocity, maximum particle velocity, and peak particle velocity have meanings that correspond with those of the related terms used for sound pressure. (SP) [32]

particulate A small particle that is created by thermal decomposition of organic materials present inside the generator. (PE/EM) 1129-1992r

parting The selective corrosion of one or more components of a solid solution alloy. (IA) [59]

parting limit The maximum concentration of a more-noble component in an alloy, above which parting does not occur within a specific environment. (IA) [59]

Partition An addressable portion of a Partitioned File. (IM/ST) 1451.1-1999

partition (1) (A) A portion of a computer's main storage that is set aside to hold a single program. (B) A portion of a storage medium that is set aside for some special purpose; for

example, the boot partition of a magnetic disk contains operating system files from which the computer can be booted. (C) A portion of a storage medium that is treated as if it were an individual medium; as in a partition of a hard disk. (C) 610.10-1994

(2) A partition is a program or part of a program that can be invoked from outside the Ada implementation. Each partition may run in a separate address space, possibly on a separate computer. An *active partition* is a partition that contains at least one task. Every active partition has an *environment task*, on which all the other tasks of that partition depend. An active partition corresponds to a POSIX process. (C) 1003.5-1999

(3) A subdivision of a file. (IM/ST) 1451.1-1999

(4) A portion of a side of a cartridge that is accessible as a unit. (C/SS) 1244.1-2000

partitioned access The process of storing and retrieving data from storage in such a way that the data is divided into subunits, called members, and the data may be processed as a whole or member by member. *Note:* The directory used to retrieve each member is stored along with the data. *See also:* partitioned data set; basic partitioned access method. (C) 610.5-1990w

partitioned data set A file that is divided into subunits, called members, each of which may be processed individually. *Contrast:* indexed file; sequential file. (C) 610.5-1990w

Partitioned File An instance of the class IEEE1451-PartitionedFile or of a subclass Thereof. (IM/ST) 1451.1-1999

partitioning (software) (software requirements specifications) Decomposition; the separation of the whole into its parts. (C/SE) 830-1984s, 610.12-1990

partition noise Noise caused by random fluctuations in the distribution of current between the various electrodes. (ED) [45]

partly cloudy sky (illuminating engineering) A sky that has 30% to 70% cloud cover. (EEC/IE) [126]

Partner The remote entity in a Link Aggregation Control Protocol exchange. (C/LM) 802.3ad-2000

part programming, computer (numerically controlled machines) The preparation of a manuscript in computer language and format required to accomplish a given task. The necessary calculations are to be performed by the computer. (IA/EEC) [61], [74]

part programming, manual (numerically controlled machines) The preparation of a manuscript in machine control language and format required to accomplish a given task. The necessary calculations are to be performed manually. (IA/EEC) [61], [74]

parts (replacement parts for Class 1E equipment in nuclear power generating stations) Items from which the equipment is assembled (for example, resistors, capacitors, wires, connectors, transistors, tubes, lubricants, O-rings, and springs). (PE/NP) 934-1987w

parts per million $x/10^6$ is x parts per million. (CAS) [13]

parts per million by volume (PPMV) (1) The volume of water vapor in the total SF₆ system at the pressure the system is operated. (SUB/PE) C37.122-1983s

(2) One million times the ratio of the volume of water vapor present in the gas to the total volume of the gas (including water vapor). (PE/IC) 1125-1993

parts per million by weight (PPMW) (1) The weight of water vapor in the total SF₆ system at the pressure the system is operated. (SUB/PE) C37.122-1983s

(2) One million times the ratio of the weight of water vapor present in the gas to the total weight of the gas (including water vapor). (PE/IC) 1125-1993

parts program A set of computer instructions used to control the action of a numerical control machine in producing a particular manufactured part. (C) 610.2-1987

Exhibit 3

NEWTON's TELECOM DICTIONARY

22nd Edition

STAY INFORMED

To be alerted by email to news, updates and corrections
send a blank email to newton@news.cmpbooks.com
or go to www.cmpbooks.com/newton

CMPBooks
San Francisco

NEWTON's TELECOM DICTIONARY

Copyright © 2006 Harry Newton

email: Harry@HarryNewton.com

personal web site: www.HarryNewton.com

business web site: www.InSearchOfThePerfectInvestment.com

All rights reserved under International and Pan-American Copyright conventions,
including the right to reproduce this book or portions thereof in any form whatsoever.

Published in the United States by

CMP Books

An imprint of CMP Media Inc.

600 Harrison Street, San Francisco, CA 94107

Phone 415-947-6615 and Fax 415-947-6015

email: bookorders@cmp.com

www.cmpbooks.com

**CMP**

United Business Media

For individual or quantity orders

CMP Books

6600 Silacci Way Gilroy, CA 95020

Tel: 1-800-500-6875 or 1-800-LIBRARY or 1-408-848-5296

This book is also sold through www.Amazon.com, www.Fatbrain.com,
www.BarnesAndNoble.com and all fine booksellers worldwide.

Distributed to the book trade in the U.S. and Canada by

Publishers Group West

1700 Fourth St., Berkeley, CA 94710

Fax: 408-848-5784 cmp@rushorder.com

ISBN Number 1-57820-319-8

February 2006

Twenty Second Edition

Steven Schoen, Contributing Editor

Valerie Pippin, Publisher

Gail Saari, Managing Editor

Sachie Jones, Marketing Manager

Saul Roldan and Damien Casteneda, Cover design

Brad Greene, Greene Design, Text layout

Accelerated Graphics Port / Access Charge

the charges early on in the asset's life, you get the time value of money, i.e. depreciation charged today (and tax saved today) is worth more than the same amount of depreciation charged (or tax saved) tomorrow.

Accelerated Graphics Port See AGP.

Accelerated Aging A test in which voltage, temperature, etc. are increased above normal operating values to obtain observable deterioration in a relatively short period of time. cables, generally expressed in dB per unit length. See DB.

accelerator 1. A chemical additive which hastens a chemical reaction under specific conditions. A term used in the telecommunications cable manufacturing industry.

2. In a Windows program, an accelerator is a keystroke that dispatches a message to a program, invoking one of its functions. For example, Alt-F4 tells the current Windows application of Windows itself to quit.

accelerator board A board added onto a personal computer's main board and designed to increase the PC's performance in writing to screen or disk, etc. See also Accelerator Card.

accelerator card An add-on product that upgrades the CPU of a PC to a higher speed or more powerful generation of processor. An accelerator card is usually a "daughter board" that clips onto the original CPU or is inserted into the socket that held the original CPU. See also Accelerator Board and Web Accelerator.

Accenture Accenture is the new name for Anderson Consulting. I have no idea why they changed their name. According to Accenture, the firm is reinventing itself to become the market maker, architect and builder of the new economy, bringing innovations to improve the way the world works and lives. I hope they do.

acceptable angle The biggest possible angle between a ray and the center axis. The maximum angle that a fiber optic cable accepts light for further transmission.

Acceptable Use Policy AUP. Many transit networks have policies which restrict the nature of their use or the basis on which access privileges are granted. A well known example is that of the now defunct NSFNET's (National Science Foundation Network) AUP which traditionally did not allow commercial use. Subsequently, of course, NSFNET did become commercialized as the Internet. AUP enforcement varies with the specific network.

acceptance 1. A formal set of criteria which you and your vendor define and agree upon that will determine whether the system he shipped you and/or installed at your place is now as you both agreed upon, and therefore, you are obligated to pay for it. See Acceptance Test.

2. Acceptance also refers to the amount of time within which a buyer has to decide whether a software or hardware element is acceptable. Different from a warranty on performance, acceptance applies to the appearance and performance of the element as initially configured and installed, and as compared to the specifications to which the seller and purchaser have agreed. Acceptance periods usually range from two to four weeks, but are determined on a case-by-case basis. It's commonly used in the secondary telecom equipment marketplace, as well. See Acceptance Test.

acceptance angle The angle over which the core of an optical fiber accepts incoming light. It's usually measured from the fiber axis.

acceptance cone In fiber optics, the cone within which optical power may be coupled into the bound modes of an optical fiber. The acceptance cone is derived by rotating the acceptance angle about the fiber axis.

acceptance pattern Of an antenna, for a given plane, a distribution plot of the off-axis power relative to the on-axis power as a function of angle or position. The acceptance pattern is the equivalent of a horizontal or vertical antenna pattern. The acceptance pattern of an optical fiber or fiber bundle is the curve of total transmitted power plotted against the launch angle.

acceptance test The final test of a new telephone system. If the system passes the test—i.e. it meets all specifications laid down in the sales contract—and is working well, then, and only then, will the customer finish paying for it. See also Acceptance and Acceptance Testing.

acceptance testing Operating and testing of a communication system, subsystem, or component, to ensure that the specified performance characteristics have been met. See Acceptance and Acceptance Test.

acceptance trial A military term. A trial carried out by nominated representatives of the eventual military users of the weapon or equipment to determine if the specified performance and characteristics have been met.

acceptor The Bluetooth device receiving an action from another Bluetooth device. The

device sending the action is called the initiator. The acceptor is typically part of an established link. See Bluetooth.

access 1. A series of digits or characters which must be dialed, typed or entered in some way to get use of something. That "something" might be a PBX or KTS telephone system, a long distance carrier, an electronic mail service, a private corporate network, a mainframe computer, or a local area network. Once the user dials the main number for the service he must then enter his assigned Access Code to get permission to use the system. An Access Code becomes an Authorization Code when it is used for identifying the caller; it becomes an Account Code when used for purposes of identifying the client for billback of associated charges. Access Code may also mean the digit, or digits, a user must dial to be connected to an outgoing trunk. For example, the user picks up his phone and dials "9" for a local line, dials "8" for long distance, dials "76" for the tie line to Chicago, etc. In programming a phone system there are unique Access Codes for Startup, Configuration programming, Administration programming and other functions; technicians who have been certified at various levels are provided with the appropriate Access Codes which allow them to invoke the appropriate level of system privilege.

2. In respect to privacy, an individual's ability to view, modify, and contest the accuracy and completeness of personally identifiable information collected about him or her. Access is an element of the Fair Information Practices Act.

access attempt The process by which one or more users interact with a telecommunications system to request to transfer information. An access attempt begins with an issuance of an access request by an access originator. An access attempt ends either in successful access or in access failure.

access broadband over power line See BPL.

access bus Access Bus is correctly spelled Access.bus. It is a 100 Kbps bus currently being implemented as part of the Video Electronics Standards Association's Display Data Channel for controlling PC monitors. Access.bus allows bidirectional communication between compatible systems and displays, allowing on-the-go installation. Also supports daisy-chaining to reduce cable snarl. Access.bus has four pins per connector. See also USB and Firewire.

access channel capacity Cable television channel capacity dedicated to cablecasting by entities not affiliated with the cable system operator, and over which the cable system operator does not exercise editorial control. Categories of access channel capacity are:

- Public Access, dedicated for use by the general public.
- Educational Access, dedicated to local educational authorities.
- Government Access, dedicated to local government.
- Leased Access, dedicated to commercial users.

The significant distinction here is "editorial control": subject to certain narrowly-defined exceptions (obscenity, "unlawful conduct"), the cable operator is prohibited from exercising editorial control over programming carried on access channels. This distinction affects the records which must be kept in the system's PIF:

- The operator is exempt from the requirements of Section 76.225c of the FCC rules regarding limits on commercial matter in children's programming carried on access channels.
- The operator is required by Section 76.701 (h) to maintain records to verify compliance with rules governing leased-access channels carrying indecent programming.

access charge As part and parcel of the Modified Final Judgement (MFJ) which mandated the breakup of the Bell System though AT&T's divestiture of the Bell Operating Companies, the FCC declared that all end users have easy access to the long distance carrier of their choice. Further, such access to the InterExchange Carrier (IEC, or IXC) was to be provided with equal ease, i.e. on an equal 1+ basis, as traditionally enjoyed by AT&T alone. Additionally, the breakup of the Bell System invalidated the complex settlements structure which served to subsidize the Local Exchange Carriers (LECs) for providing local access to the long distance network(s). In order to replace the traditional settlements process and to compensate the LECs for the use of the vital, expensive and relatively unprofitable local access network, Access Charges were mandated, falling into two general categories.

1. The end user Access Charge, also known as Customer Access Line Charge (CALC) applies to every local loop, and is sensitive to the nature of the circuit. Subject to review and adjustment by the FCC on an annual basis, such Access Charges differ for residential and business users, single lines versus trunks, leased lines versus local access circuits, and so on. A mandatory charge appearing as a separate line item on the user's bill, it applies whether

Exhibit 4

OVER
10,000
ENTRIES

Microsoft

Computer Dictionary

Fifth Edition

- Fully updated with the latest technologies, terms, and acronyms
- Easy to read, expertly illustrated
- Definitive coverage of hardware, software, the Internet, and more!



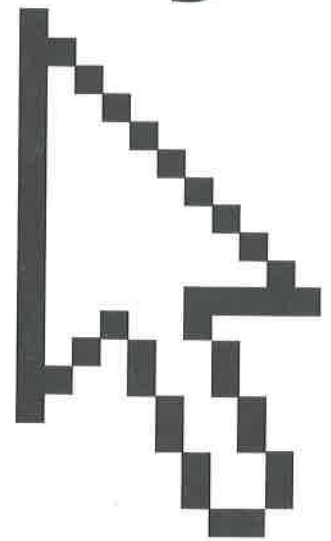
Microsoft

Property of Susman Godfrey

Microsoft

Computer Dictionary

Fifth Edition



RECEIVED
APR 14 2014

BY: _____

PUBLISHED BY

Microsoft Press

A Division of Microsoft Corporation

One Microsoft Way

Redmond, Washington 98052-6399

Copyright © 2002 by Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Cataloging-in-Publication Data

Microsoft Computer Dictionary.--5th ed.

p. ; cm.

ISBN 0-7356-1495-4

1. Computers--Dictionaries. 2. Microcomputers--Dictionaries.

AQ76.5. M52267 2002

004'.03--dc21

200219714

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWT 7 6 5 4 3 2

Distributed in Canada by Penguin Books Canada Limited.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to mspinput@microsoft.com.

Active Desktop, Active Directory, ActiveMovie, ActiveStore, ActiveSync, ActiveX, Authenticode, BackOffice, BizTalk, ClearType, Direct3D, DirectAnimation, DirectDraw, DirectInput, DirectMusic, DirectPlay, DirectShow, DirectSound, DirectX, Entourage, FoxPro, FrontPage, Hotmail, IntelliEye, IntelliMouse, IntelliSense, JScript, MapPoint, Microsoft, Microsoft Press, Mobile Explorer, MS-DOS, MSN, Music Central, NetMeeting, Outlook, PhotoDraw, PowerPoint, SharePoint, UltimateTV, Visio, Visual Basic, Visual C++, Visual FoxPro, Visual InterDev, Visual J++, Visual SourceSafe, Visual Studio, Win32, Win32s, Windows, Windows Media, Windows NT, Xbox are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Acquisitions Editor: Alex Blanton

Project Editor: Sandra Haynes

Body Part No. X08-41929

s and therefore can
sses are defined so that
make them concrete by
s. *See also* class, instan-
ion 2). *Compare* con-

nning, a data set
s of the information it
it can be performed
ore generalized than a
erties of the objects it
pe "pet" is more gener-
," "pet bird," and "pet
n illustrating an abstract
ion of memory used to
emporary basis. As an
ply a structure onto
d) and from which they
e of value, such as inte-
The way in which the
bstract data types is
rest of the program.

umer to change the defi-
ons without introducing
s the abstract data type.
intermediate step between
t-oriented program-
oriented programming.

e of specialized soft-
mming interface (API),
rom device dependen-
ing software. For
bles programs to focus
s, instead of on individ-
ications devices. **2.** In
process of reducing an
ne necessary elements
es an object in terms of
s (functionality), and
g with other objects).

or a processor that is not
represents a model for
age. Its instruction set
sely resemble the com-
is used by an actual
ake the implementation
ther platforms.

abstract machine language *n.* **1.** An intermediate pro-
gramming language used by an interpreter or compiler.
2. *See* pseudocode (definition 1).

abstract syntax *n.* A data structure description that is
independent of hardware structures and encodings.

Abstract Syntax Notation One *n.* The ISO standard
notation for independent specification of data types and
structures for syntax conversion. *Acronym:* ASN.1. *See*
also data type, ISO, syntax.

abstract syntax tree *n.* A treelike representation of pro-
grams used in many integrated programming environ-
ments and structure-oriented editors.

Abstract Window Toolkit *n.* A library of Java GUIs
(graphical user interfaces) that provides the connections
between a Java application and the native GUI of the com-
puter on which the application runs. *Also called:* AWT.

A/B switch box *n.* A switch box with two outputs. By
flipping the switch, the user can select which to use. *See*
also switch (definition 1), switch box.

AC *n.* *See* alternating current.

AC adapter *n.* An external power supply that converts
from a 110 VAC or 220 VAC domestic electric supply
("house current" or "main power") to low-voltage DC,
which is required to operate solid-state electronic equip-
ment (such as a laptop computer) that does not include an
internal power supply.

Accelerated Graphics Port *n.* *See* AGP.

accelerator *n.* **1.** In applications, a key or key combination
used to perform a defined function. *Also called:* shortcut
key. **2.** In hardware, a device that speeds or enhances the
operation of one or more subsystems, leading to improved
program performance. *See also* accelerator card, Win-
dows-based accelerator.

accelerator board *n.* *See* accelerator card.

accelerator card *n.* A printed circuit board that replaces
or augments the computer's main microprocessor, result-
ing in faster performance. *Also called:* accelerator board.
See also expansion board, graphics accelerator.

acceptable use policy *n.* A statement issued by an ISP
(Internet service provider) or an online information ser-
vice that indicates what activities users may or may not
engage in while logged into the service. For example,
some providers prohibit users from engaging in commer-
cial activity on the network. *Acronym:* AUP. *See also* ISP,
online information service.

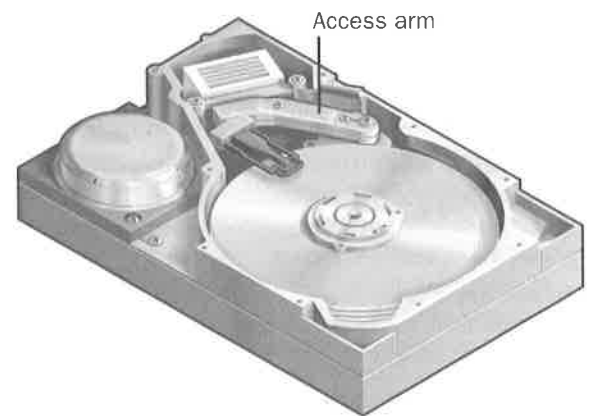
acceptance test *n.* A formal evaluation of a hardware
product performed by the customer, usually at the fac-
tory, to verify that the product is performing according to
specifications.

access¹ *n.* **1.** The act of reading data from or writing data
to memory. **2.** Connection to the Internet or other network
or system.

access² *vb.* To gain entry to memory in order to read or
write data.

Access *n.* Microsoft's relational database-management
software for the Windows desktop platform. Part of the
family of Microsoft Office products, Access in its most
recent version (Access 2002) supports Web technology for
building, managing, and sharing data. Access 2002 also
includes new and improved tools for accessing and view-
ing information and offers integration with Microsoft's
BackOffice database product, SQL Server. *See also*
Office.

access arm *n.* A mechanical arm that moves the read/
write head(s) over the surface of a disk in a disk drive. *See*
the illustration. *Also called:* head arm.



Access arm.

ACCESS.bus *n.* A bidirectional bus for connecting
peripherals to a PC. The ACCESS.bus can connect up to
125 low-speed peripherals, such as printers, modems, mice,
and keyboards, to the system through a single, general-pur-
pose port. Peripherals that support the ACCESS.bus pro-
vide a connector or port connection that is similar to a
phone-jack connector and are daisy-chained together.
However, the PC communicates directly with each periph-
eral and vice versa. Connecting an ACCESS.bus device
(for example, a printer) to a system results in the system

parrallaxing

parrallaxing *n.* A 3-D animation technique, often used by computer game developers, where backgrounds are displayed using different levels of speed to achieve realism. For example, distant levels move at a slower speed than closer levels, thereby giving the illusion of depth. *See also* animation.

parse *vb.* To break input into smaller chunks so that a program can act upon the information.

parser *n.* An application or device that breaks data into smaller chunks so that an application can act on the information. *See also* parse.

partition *n.* **1.** A logically distinct portion of memory or a storage device that functions as though it were a physically separate unit. **2.** In database programming, a subset of a database table or file.

Partition Boot Sector *n.* The first sector in the system (startup) partition of a computer's bootable hard disk, or the first sector of a bootable floppy disk. On an x86-based computer, the Partition Boot Sector is read into memory at startup by the Master Boot Record. It is the Partition Boot Sector that contains the instructions required to begin the process of loading and starting the computer's operating system. *See also* Master Boot Record, partition table.

partition table *n.* A table of information in the first sector of a computer's hard disk that tells where each partition (discrete portion of storage) on the disk begins and ends. The physical locations are given as the beginning and ending head, sector, and cylinder numbers. In addition to these "addresses," the partition table identifies the type of file system used for each partition and identifies whether the partition is bootable—whether it can be used to start the computer. Although it is a small data structure, the partition table is a critical element on the hard disk.

partnership *n.* The settings on a desktop computer and Windows CE device that allow information to be synchronized, as well as copied or moved between the computer and device. The mobile device can have partnerships with up to two desktop computers. *See also* synchronization (definition 6).

Pascal *n.* A concise procedural language designed between 1967 and 1971 by Niklaus Wirth. Pascal, a compiled, structured language built upon ALGOL, simplifies syntax while adding data types and structures such as

subranges, enumerated data types, files, records, and sets. *See also* ALGOL, compiled language. *Compare* C.

pASP *n.* *See* pocket Active Server Pages.

pass¹ *n.* In programming, the carrying out of one complete sequence of events.

pass² *vb.* To forward a piece of data from one part of a program to another. *See also* pass by address, pass by value.

pass by address *n.* A means of passing an argument or parameter to a subroutine. The calling routine passes the address (memory location) of the parameter to the called routine, which can then use the address to retrieve or modify the value of the parameter. *Also called:* pass by reference. *See also* argument, call¹. *Compare* pass by value.

pass by reference *n.* *See* pass by address.

pass by value *n.* A means of passing an argument or a parameter to a subroutine. A copy of the value of the argument is created and passed to the called routine. When this method is used, the called routine can modify the copy of the argument, but it cannot modify the original argument. *See also* argument, call¹. *Compare* pass by address.

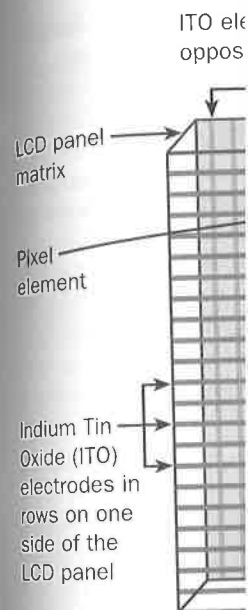
passivation *n.* In Sun Microsystems's J2EE network platform, the process of "turning off" an enterprise java bean (EJB) by caching it from memory to secondary storage. *See also* Enterprise JavaBeans, J2EE. *Compare* activation.

passive hub *n.* A type of hub used on ARCnet networks that passes signals along but has no additional capability. *See also* ARCnet. *Compare* active hub, Intelligent hub.

passive-matrix display *n.* An inexpensive, low-resolution liquid crystal display (LCD) made from a large array of liquid crystal cells that are controlled by transistors outside of the display screen. One transistor controls an entire row or column of pixels. Passive-matrix displays are commonly used in portable computers, such as laptops and notebooks, because of their thin width. While these displays have good contrast for monochrome screens, the resolution is weaker for color screens. These displays are also difficult to view from any angle other than straight on, unlike active-matrix displays. However, computers with passive-matrix displays are considerably cheaper than those with active-matrix screens. *See the illustration. Also called:* dual-scan display. *See also* liquid crystal display, supertwist display, transistor, twisted nematic display. *Compare* active-matrix display.

passive-matrix display

passive node



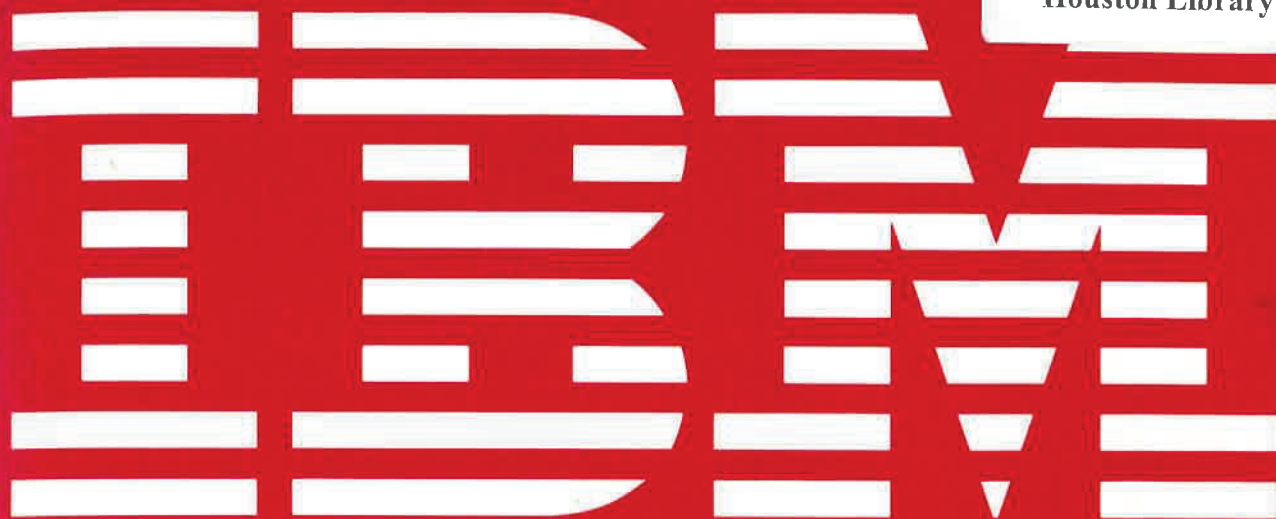
Passive-matrix display.

passive node *n.* A network node that is not active in the network; typical of a bus network, node (definition 1).

Passport *n.* A suite of products from Microsoft that consists of a web browser and other information. With the service, a user enters one's name and address at a port site on the Internet; at the port site, a user can sign in to receive information. Passport also includes a service that stores credit card information. Kids Passport service, a service for children, is one of the founding members of the .NET initiative. *See also* .NET, sign-on, wallet.

pass-through *adj.* **1.** In networking, a device that acts as an intermediary between two networks. For example, a pass-through device allows access to an internal (private) network from the requesting client without direct access. **2.** Pertaining to a device that allows

Exhibit 5



Dictionary of Computing

- ▼ The most comprehensive computing dictionary ever published
- ▼ More than 18,000 entries

IBM DICTIONARY OF COMPUTING

Compiled and edited by
GEORGE McDANIEL

McGRAW-HILL, INC.
New York San Francisco Washington, D.C. Auckland Bogotá
Caracas Lisbon London Madrid Mexico City Milan
Montreal New Delhi San Juan Singapore
Sydney Tokyo Toronto

Limitation of Liability

While the Editor and Publisher of this book have made reasonable efforts to ensure the accuracy and timeliness of the information contained herein, neither the Editor nor the Publisher shall have any liability with respect to loss or damage caused or alleged to be caused by reliance on any information contained herein.

Copyright © 1994 by International Business Machines Corporation. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

1 2 3 4 5 6 7 8 9 0 DOC/DOC 9 9 8 7 6 5 4 3

ISBN 0-07-031488-8 (HC)
ISBN 0-07-031489-6 (PBK)

The sponsoring editor for this book was Daniel A. Gonneau and the production supervisor was Thomas G. Kowalczyk.

Printed and bound by R. R. Donnelley & Sons Company.

Tenth Edition (August 1993)

This is a major revision of the *IBM Dictionary of Computing*, SC20-1699-8, which is made obsolete by this edition. Changes are made periodically to the information provided herein.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country. Comments may be addressed to IBM Corporation, Department E37/656, P. O. Box 12195, Research Triangle Park, NC 27709.

International Edition

Copyright © 1994 by International Business Machines Corporation. Exclusive rights by McGraw-Hill, Inc. for manufacture and export. This book cannot be re-exported from the country to which it is consigned by McGraw-Hill. The International Edition is not available in North America.

When ordering this title, use ISBN 0-07-113383-6.

control language program

es and buffers are made
users and application pro-
l operation.

set of body groups that
of a control data item or

ay begin with a control
oting, and contain detail

a report group that is
of the control group of

a designated sequence
l by the positional order
es within a CONTROL

signation hole. See also

r Deprecated term for

th area of direct access
records and creates dis-
n key-sequenced data set
ted to by an entry in the
e control interval is the
M transmits to or from
ol interval always com-
ysical records.

systems with VSAM, the
contents of a control

ld (CIDE) In VSAM, a
es at the end of each
he free space, if any, in

tems with VSAM, the
ed records in a control
al to facilitate insertion
does not fit in the ori-

t of all commands with
tions. Synonym for
control language.

n executable object that
ting entirely of control

control language variable

control language variable A program variable declared in a control language program and available only to the program.

controlled cancel In System/36, the system action that ends a job step being run and saves any new data that has been created. The job that is running can continue with the next job entry.

controlled maintenance A method to sustain a desired quality of service by the systematic application of analysis techniques using centralized supervisory facilities and/or sampling to minimize preventive maintenance and to reduce corrective maintenance. (T)

controlled slip A slip in which a fixed number of digits are lost or gained.

controller A device that coordinates and controls the operation of one or more input/output devices, such as workstations, and synchronizes the operation of such devices with the operation of the system as a whole. See communication controller, store controller, subsystem controller. See also input/output control unit.

controller card In the AS/400 system, any of the I/O controller logic cards, such as a magnetic storage device controller, workstation controller, or communications controller.

controller configuration facility (CCF) Set of macrostatements to be coded by a user and modules in Programmable Store System Host Support, used to define and create a 3651 or 7480 Store Controller operational environment.

controller creation parameter table (CCPT) In PSS, a table containing pointers to all tables and data areas used by the controller configuration facility in defining the 3651 or 7480 Store Controller operational environment.

controller data IBM-provided modules, tables, lists, and control blocks that are used at a host system to create an operational environment for a controller.

controller description (CTLD) In the AS/400 system, an object that contains a description of the characteristics of a controller that is either directly attached to the system or attached to a communication line.

controller disk In the 3650 Retail Store System, an integral part of the retail store controller that is used for auxiliary storage of controller data, user files, and application programs.

controller function An action or series of actions built into a controller and taken by the controller in

[145]

control logical unit (CLU)

response to a request from a terminal, another controller function, or a host system.

controller information fields In the 3790 Communication System, a set of fields, assigned to each 3790 program and to which the program has access, that contain information the program may need while being executed, such as terminal operator ID, date, and record numbers of print data set and transaction data set records being used by the program.

controller storage In the 3650 Retail Store System, the generic name for auxiliary (disk) storage and active (monolithic) storage in a 3651 store controller.

controller storage save In the 3650 Retail Store System, automatic writing of the critical areas of store controller active storage onto the integrated disk when power is turned off or when a power failure is detected.

control-level indicator In RPG, an indicator used to specify certain fields as control fields and to control the operations that are to be performed at total and detail time in the RPG program cycle.

control lever Synonym for joy stick.

controlling application program In VTAM programs, an application program with which a logical unit other than a secondary application program is automatically put in session whenever the logical unit is available. See also automatic logon.

controlling logical unit In VTAM programs, a logical unit with which a secondary logical unit, other than an application program, is automatically put in session whenever the secondary logical unit is available. A controlling logical unit can be either an application program or a device-type logical unit. See also automatic logon, controlling application program.

controlling subsystem An interactive subsystem that is started automatically when the system is started and through which the system operator controls the system.

control logic In the IBM 8100 Information System, I/O hardware that can decode and perform commands addressed to it by I/O instructions.

control logical unit (CLU) A logical unit that resides in a Transaction Processing Facility (TPF) type 2.1 node and that is used to pass private protocol request units between this TPF type 2.1 node and the logon manager (a VTAM application program). The communication flow between the control logical unit and the logon manager enables a logical unit controlled by VTAM to establish a session with TPF.

Exhibit 6

**EXHIBIT 1 – U.S. PATENT NO. 8,648,867 VS NVIDIA ACCELERATED COMPUTING PRODUCTS & SERVICES –
HOPPER ARCHITECTURE**

I. INTRODUCTION

The chart below demonstrates how Defendant Nvidia Corporation (“Nvidia” or “Defendant”), as well as Defendant’s partners, customers, and end users, directly infringe, either literally or under the doctrine of equivalents, claims 16-19 of U.S. Patent No. 8,648,867 (“the ’867 Patent”).

Nvidia has committed acts of infringement within this District. Nvidia uses the Accused Products in this District in manners that practice the ’867 Patent, including by testing the Accused Products and by using the Accused Products at its offices and premises in this District. Defendant makes, uses, advertises, offers for sale, and/or sells hardware for accelerated computing, including GPUs, CPUs, and SoCs; computers for accelerated computing (e.g., supercomputers, servers, and data centers for high performance computing); and computer platform software-as-a-service (“SaaS”) that implements accelerated computing (including the Accused Products) in the State of Texas and in this District directly and/or through its partnerships

Defendant also provides data center and HPC services that practice the ’867 Patent in the State of Texas and in this District directly and/or through its partnerships with businesses in the State of Texas and in this District.

Nvidia sells, offers for sale, advertises, makes, installs, and/or otherwise provides hardware, software, firmware, and/or computer platforms for accelerated computing and data center and HPC services, including the Accused Products, the use of which infringes the ’867 Patent in this District and the State of Texas. (See <https://www.nvidia.com/en-us/data-center/solutions/accelerated-computing/>.) Nvidia performs these acts directly and/or through its partnerships with other entities. (See *id.* (“NVIDIA has defined a range of accelerated platforms that each consist of hardware systems designed according to the needs of the use case as well as the software stack that enables the operation and management of the business applications. These hardware systems and software are available from NVIDIA and our partners.”).)

Nvidia also uses a network of partners, which comprise re-sellers, managed service providers, and product and solution experts, to provide the Accused Products and implementation services for the Accused Products to customers in this District. Each of these partners sells, offers for sale, installs, and/or implements Nvidia’s accelerated computing hardware, software, and/or computer platform services. (See <https://www.nvidia.com/en-us/about-nvidia/partners/>.)

Nvidia’s partners include “Data Center Provider[s].” (See <https://www.nvidia.com/en-us/about-nvidia/partners/>.) Nvidia’s Data Center Provider partners “offer colocation services such as high-density data center facilities, interconnected infrastructure, and state-of-art

Key Features	Accused Products
	<pre> * * * * * __host__ cudaError_t cudaMallocArray (cudaArray_t* array , const cudaChannelFormatDesc* desc , size_t width , size_t height = 0, unsigned int flags = 0) Allocate an array on the device. * * * * * __host__ cudaError_t cudaMemcpy2D (void* dst , size_t dpitch , const void* src , size_t spitch , size_t width , size_t height , cudaMemcpyKind kind) Copies data between host and device. </pre> (See https://docs.nvidia.com/cuda/cuda-runtime-api/group__CUDART__MEMORY.html (emphasis added).)
[16.2] transferring, by an accelerator controller, the first input data into a first partition, referenced by first pointer, of an accelerator memory before a first computational cycle of the numerical simulation;	The Accused Products practice a method that includes <i>transferring, by an accelerator controller, the first input data into a first partition, referenced by first pointer, of an accelerator memory before a first computational cycle of the numerical simulation.</i> For instance, the GPU architecture of the Accused Products implements a controller. As an example, the Hopper GPU architecture implements “HBM3 memory controllers” including “12 512-bit memory controllers” (e.g., <i>accelerator controller</i>) coupled GPU memory including “6 HBM3 or HBM2e stacks,” “80 GB HBM3, 5 HBM3 stacks,” and “80 GB HBM2e, 5 HBM2e stacks.” The <u>NVIDIA GH100 GPU</u> is composed of multiple GPU processing clusters (GPCs), texture processing clusters (TPCs), streaming multiprocessors (SMs), L2 cache, and <u>HBM3 memory controllers</u>. The <u>full implementation of the GH100 GPU</u> includes the following units: • 8 GPCs, 72 TPCs (9 TPCs/GPC), 2 SMs/TPC, 144 SMs per full GPU • 128 FP32 CUDA Cores per SM, 18432 FP32 CUDA Cores per full GPU • 4 fourth-generation Tensor Cores per SM, 576 per full GPU • 6 HBM3 or HBM2e stacks, 12 512-bit memory controllers • <u>60 MB L2 cache</u> • <u>Fourth-generation NVLink and PCIe Gen 5</u>

Key Features	Accused Products
	The <u>NVIDIA H100 GPU with SXM5 board form-factor</u> includes the following units: • 8 GPCs, 66 TPCs, 2 SMs/TPC, 132 SMs per GPU • 128 FP32 CUDA Cores per SM, 16896 FP32 CUDA Cores per GPU • 4 fourth-generation Tensor Cores per SM, 528 per GPU • 80 GB HBM3, 5 HBM3 stacks, 10 512-bit memory controllers • 50 MB L2 cache • Fourth-generation NVLink and PCIe Gen 5 The <u>NVIDIA H100 GPU with a PCIe Gen 5 board form-factor</u> includes the following units: • 7 or 8 GPCs, 57 TPCs, 2 SMs/TPC, 114 SMs per GPU • 128 FP32 CUDA Cores/SM, 14592 FP32 CUDA Cores per GPU • 4 fourth-generation Tensor Cores per SM, 456 per GPU • 80 GB HBM2e, 5 HBM2e stacks, 10 512-bit memory controllers • 50 MB L2 cache • Fourth-generation NVLink and PCIe Gen 5 (See https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/ (emphasis added).) As previously stated, the Accused Products implement CUDA, Nvidia's parallel computing platform. CUDA enables NVIDIA GPUs to be used for general purpose computing tasks and includes specialized GPU-acceleration libraries such as cuDNN. CUDA implements numerous "memory management functions" that "[c]op[y] data between host [CPU] and device [GPU]" (e.g., <i>transferring, by an accelerator controller, the first input data into a first partition, referenced by first pointer, of an accelerator memory</i>), including "cudaMemcpy," "cudaMemcpy2D," "cudaMemcpy2DArrayToArray," "cudaMemcpy2DAsync," "cudaMemcpy2DFromArray," "cudaMemcpy2DFromArrayAsync," "cudaMemcpy2DToArray," "cudaMemcpy2DToArrayAsync," and "cudaMemcpyAsync."

Key Features	Accused Products
	<u>6.11. Memory Management</u> This section describes the memory management functions of the CUDA runtime application programming interface. Some functions have overloaded C++ API template versions documented separately in the C++ API Routines module. Functions * * * * * <pre> __host__ cudaError_t cudaMemcpy (void* dst , const void* src Copies data between host and device. __host__ cudaError_t cudaMemcpy2D (void* dst , size_t dpitch Copies data between host and device. __host__ cudaError_t cudaMemcpy2DArrayToArray (cudaArray_t d Copies data between host and device. __host__ __device__ cudaError_t cudaMemcpy2DAsync (void* dst , size_t d Copies data between host and device. __host__ cudaError_t cudaMemcpy2DFromArray (void* dst , size Copies data between host and device. __host__ cudaError_t cudaMemcpy2DFromArrayAsync (void* dst , Copies data between host and device. __host__ cudaError_t cudaMemcpy2DToArray (cudaArray_t dst , Copies data between host and device. __host__ cudaError_t cudaMemcpy2DToArrayAsync (cudaArray_t d Copies data between host and device. __host__ cudaError_t cudaMemcpy3D (const cudaMemcpy3DParms* Copies data between 3D objects. __host__ __device__ cudaError_t cudaMemcpy3DAsync (const cudaMemcpy3DPa Copies data between 3D objects. __host__ cudaError_t cudaMemcpy3DPeer (const cudaMemcpy3DPee Copies memory between devices. __host__ cudaError_t cudaMemcpy3DPeerAsync (const cudaMemcpy Copies memory between devices asynchronously __host__ __device__ cudaError_t cudaMemcpyAsync (void* dst , const void Copies data between host and device. </pre>

Key Features	Accused Products
	(See https://docs.nvidia.com/cuda/cuda-runtime-api/group__CUDART__MEMORY.html (emphasis added).) Parameters used in CUDA include pointers “dst” (“Destination memory address”) and “src” (“Source memory address”). For example, CUDA function “cudaMemcpy” copies “bytes [data] from the memory area pointed to by src [source memory address pointer] to the memory area pointed to by dst [destination memory address pointer], where kind [type of transfer] specifies the direction of the copy” (e.g., <i>referenced by first pointer</i>). One of the destinations is “cudaMemcpyHostToDevice,” or host (CPU) to device (GPU) (e.g., <i>transferring, by an accelerator controller, the first input data into a first partition, referenced by first pointer, of an accelerator memory</i>). <pre data-bbox="583 651 1583 678">__host__ <u>cudaError_t</u> cudaMemcpy (void* dst , const void* src , size_t count , <u>cudaMemcpyKind</u> kind)</pre> Copies data between host and device. Parameters <u>dst</u> - Destination memory address <u>src</u> - Source memory address <u>count</u> - Size in bytes to copy <u>kind</u> - Type of transfer Returns <u>cudaSuccess</u>, <u>cudaErrorInvalidValue</u>, <u>cudaErrorInvalidMemcpyDirection</u> Description Copies <u>count</u> bytes from the memory area pointed to by <u>src</u> to the memory area pointed to by <u>dst</u>, where <u>kind</u> specifies the <u>direction of the copy</u>, and must be one of <u>cudaMemcpyHostToHost</u>, <u>cudaMemcpyHostToDevice</u>, <u>cudaMemcpyDeviceToHost</u>, <u>cudaMemcpyDeviceToDevice</u>, or <u>cudaMemcpyDefault</u>. Passing <u>cudaMemcpyDefault</u> is recommended, in which case the type of transfer is inferred from the <u>pointer values</u>. However, <u>cudaMemcpyDefault</u> is only allowed on systems that support unified virtual addressing. Calling <u>cudaMemcpy()</u> with dst and src pointers that do not match the direction of the copy results in an undefined behavior.

Key Features	Accused Products
	(See https://docs.nvidia.com/cuda/cuda-runtime-api/group__CUDART__MEMORY.html (emphasis added).) In a related example, CUDA function “cudaMemcpy2DArrayToArray” also “[c]opies data between host [CPU] and device [GPU]” and includes pointers “dst” and “src.” cudaMemcpy2DArrayToArray “[c]opies a matrix (height rows of width bytes each) from the CUDA array src [source memory address pointer] starting at hOffsetSrc [source starting Y offset (rows)] rows and wOffsetSrc [source starting X or width offset (columns in bytes)] bytes from the upper left corner to the CUDA array dst [destination memory address pointer] starting at hOffsetDst [destination starting Y or height offset (rows)] rows and wOffsetDst [destination starting X offset (columns in bytes)] bytes from the upper left corner, where kind specifies the direction of the copy.” One of the destinations is, again, “cudaMemcpyHostToDevice,” or host (CPU) to device (GPU).

Key Features	Accused Products
	<pre data-bbox="577 240 1795 324">__host__ cudaError_t cudaMemcpy2DFromArray (cudaArray_t dst, size_t wOffsetDst, size_t hOffsetDst, cudaArray_const_t src, size_t wOffsetSrc, size_t hOffsetSrc, size_t width, size_t height, cudaMemcpyKind kind = cudaMemcpyDeviceToDevice)</pre> <u>Copies data between host and device.</u> Parameters dst - Destination memory address wOffsetDst - Destination starting X offset (columns in bytes) hOffsetDst - Destination starting Y offset (rows) src - Source memory address wOffsetSrc - Source starting X offset (columns in bytes) hOffsetSrc - Source starting Y offset (rows) width - Width of matrix transfer (columns in bytes) height - Height of matrix transfer (rows) kind - Type of transfer Returns cudaSuccess, cudaErrorInvalidValue, cudaErrorInvalidMemcpyDirection

Key Features	Accused Products
	Description Copies a matrix (<u>height</u> rows of <u>width</u> bytes each) from the CUDA array <u>src</u> starting at <u>hOffsetSrc</u> rows and <u>wOffsetSrc</u> bytes from the upper left corner to the CUDA array <u>dst</u> starting at <u>hOffsetDst</u> rows and <u>wOffsetDst</u> bytes from the upper left corner, where <u>kind</u> specifies the direction of the copy, and must be one of <u>cudaMemcpyHostToHost</u>, <u>cudaMemcpyHostToDevice</u>, <u>cudaMemcpyDeviceToHost</u>, <u>cudaMemcpyDeviceToDevice</u>, or <u>cudaMemcpyDefault</u>. Passing <u>cudaMemcpyDefault</u> is recommended, in which case the type of transfer is inferred from the <u>pointer values</u>. However, <u>cudaMemcpyDefault</u> is only allowed on systems that support unified virtual addressing. <u>wOffsetDst + width</u> must not exceed the width of the CUDA array <u>dst</u>. <u>wOffsetSrc + width</u> must not exceed the width of the CUDA array <u>src</u>. (See https://docs.nvidia.com/cuda/cuda-runtime-api/group__CUDART__MEMORY.html (emphasis added).) As a further example, CUDA function “cudaMalloc” is used for “[a]llocat[ing] memory on the device [GPU].” Parameters include pointer “devPtr” (“Pointer to allocated device memory”). The cudaMalloc function “[a]llocates size bytes [data] of linear memory on the device [GPU] and returns in *devPtr a pointer to the allocated memory [of the GPU]” (e.g., <i>the first input data into a first partition, referenced by first pointer, of an accelerator memory before a first computational cycle of the numerical simulation</i>).

Key Features	Accused Products
	<pre data-bbox="590 245 1339 272">__host__ __device__ <u>cudaError_t</u> cudaMalloc (void** devPtr , size_t size)</pre> <u>Allocate memory on the device.</u> Parameters <code>devPtr</code> - Pointer to allocated device memory <code>size</code> - Requested allocation size in bytes Returns <code>cudaSuccess</code>, <code>cudaErrorInvalidValue</code>, <code>cudaErrorMemoryAllocation</code> Description Allocates <code>size</code> bytes of linear memory on the device and returns in <code>*devPtr</code> a pointer to the allocated memory. The allocated memory is suitably aligned for any kind of variable. The memory is not cleared. <code>cudaMalloc()</code> returns <code>cudaErrorMemoryAllocation</code> in case of failure. The device version of <code>cudaFree</code> cannot be used with a <code>*devPtr</code> allocated using the host API, and vice versa. (See https://docs.nvidia.com/cuda/cuda-runtime-api/group__CUDART__MEMORY.html (emphasis added).) In a related example, the CUDA function “<code>cudaMallocArray</code>” is used for “[a]llocat[ing] an array [pointer] on the device [GPU].” Parameters include pointer “<code>array</code>” (“Pointer to allocated array in device memory”), “<code>desc</code>” (“Requested channel format”), “<code>width</code>” (“Requested array allocation width”), “<code>height</code>” (“Requested array allocation height”), and “<code>flags</code>” (“Requested properties of allocated array”). The <code>cudaMallocArray</code> function “[a]llocates a CUDA array according to the <code>cudaChannelFormatDesc</code> structure <code>desc</code> [requested channel format] and returns a handle to the new CUDA array in <code>*array</code> [pointer to allocated array in device or GPU memory].”

Key Features	Accused Products
	<pre data-bbox="577 240 1812 297">__host__ <u>cudaError_t</u> cudaMallocArray (<u>cudaArray_t</u>* array , const <u>cudaChannelFormatDesc</u>* desc , size_t width , size_t height = 0, unsigned int flags = 0)</pre> <u>Allocate an array on the device.</u> Parameters <u>array</u> - Pointer to allocated array in device memory <u>desc</u> - Requested channel format <u>width</u> - Requested array allocation width <u>height</u> - Requested array allocation height <u>flags</u> - Requested properties of allocated array Returns <u>cudaSuccess</u>, <u>cudaErrorInvalidValue</u>, <u>cudaErrorMemoryAllocation</u> Description <u>Allocates a CUDA array according to the <u>cudaChannelFormatDesc</u> structure <u>desc</u> and returns a handle to the new CUDA array in <u>*array</u>.</u> (See https://docs.nvidia.com/cuda/cuda-runtime-api/group__CUDART__MEMORY.html (emphasis added).) As a further example, device “shared memory” is partitioned into “memory modules (banks).” CUDA C++ Best Practices Guide <u>The programming guide to using the CUDA Toolkit to obtain the best performance from <u>NVIDIA GPUs</u>.</u> * * * * *

Key Features	Accused Products
	9.2.3. Shared Memory Because it is on-chip, shared memory has much higher bandwidth and lower latency than local and global memory - provided there are no bank conflicts between the threads, as detailed in the following section. 9.2.3.1. Shared Memory and Memory Banks To achieve high memory bandwidth for concurrent accesses, shared memory is divided into equally sized memory modules (<u>banks</u>) that can be accessed simultaneously. Therefore, any memory load or store of n addresses that spans n distinct memory banks can be serviced simultaneously, yielding an effective bandwidth that is n times as high as the bandwidth of a single bank. However, if multiple addresses of a memory request map to the same memory bank, the accesses are serialized. The hardware splits a memory request that has bank conflicts into as many separate conflict-free requests as necessary, decreasing the effective bandwidth by a factor equal to the number of separate memory requests. The one exception here is when multiple threads in a warp address the same shared memory location, resulting in a broadcast. In this case, multiple broadcasts from different banks are coalesced into a single multicast from the requested shared memory locations to the threads. To minimize bank conflicts, it is important to understand how memory addresses map to memory banks and how to optimally schedule memory requests. On devices of compute capability 5.x or newer, each bank has a bandwidth of 32 bits every clock cycle, and successive 32-bit words are assigned to successive banks. The warp size is 32 threads and the number of banks is also 32, so bank conflicts can occur between any threads in the warp. See Compute Capability 5.x in the CUDA C++ Programming Guide for further details. (See https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html#shared-memory/ (emphasis added).)
[16.3] performing, by at least one graphics processing unit during the first computational cycle, at least one calculation on the first portion of the input data as to generate first output data;	The Accused Products practice a method that includes <i>performing, by at least one graphics processing unit during the first computational cycle, at least one calculation on the first portion of the input data as to generate first output data.</i> For instance, CUDA uses “streams” to execute a sequence of commands in order. As shown below, an exemplary CUDA function “cudaMemcpyAsync” is used to copy data between a host (CPU) and a device (GPU). “Each stream copies its portion of input array hostPtr [pointer for CPU] to array inputDevPtr in device [GPU] memory” (e.g., <i>performing, by at least one graphics processing unit</i>). The stream then “processes inputDevPtr on the device [GPU] by calling MyKernel(), and copies the result outputDevPtr back to the same portion of hostPtr” (e.g., <i>performing, by at least one graphics processing unit during</i>

Exhibit 7

Dictionary of Computer Terms

Fourth Edition

Douglas A. Downing, Ph.D.
*School of Business and Economics
Seattle Pacific University*

Michael A. Covington, Ph.D.
*Artificial Intelligence Center
The University of Georgia*

Melody Mauldin Covington
*Graphic Designer
Athens, Georgia*



CONTENTS

About the Authors	vi
Preface to the Fourth Edition	vii
List of Tables	viii
List of Programs	ix
Computer Terms	1
A Visual Dictionary of Characters and Symbols	353

© Copyright 1995 by Barron's Educational Series, Inc.
 Prior editions © copyright 1992, 1989, and 1986
 by Barron's Educational Series, Inc.

All rights reserved.
 No part of this book may be reproduced in any form, by photostat, microfilm,
 xerography, or any other means, or incorporated into any information
 retrieval system, electronic or mechanical, without the written permission
 of the copyright owner.

All inquiries should be addressed to:
 Barron's Educational Series, Inc.
 250 Wireless Boulevard
 Hightstown, New York 11788

Library of Congress Catalog Card No. 95-7217

International Standard Book No. 0-8120-9023-3

Library of Congress Cataloging-in-Publication Data

Downing, Douglas.
 Dictionary of computer terms / Douglas A. Downing, Michael
 Covington, Melody Mauldin Covington.—4th ed.
 p. cm.
 Michael Covington's name appears first on earlier editions.
 ISBN 0-8120-9023-3
 I. Computers—Dictionaries. I. Michael Covington A., 1957—
 II. Covington, Melody Mauldin. III. Title.
 QA76.15.C68 1995
 004'.3—dc20 95-7217
 CIP

PRINTED IN THE UNITED STATES OF AMERICA

678 9770 98765432

PART 15 DEVICE

The process is complete when the input string is empty and all of the elements of a sentence have been found.

In bottom-up parsing, the computer accepts elements from the input string and tries to put them together, thus:

Accept *the*, which is a Determiner
 Accept *dog*, which is a Noun
 Determiner + Noun make a Noun Phrase
 Accept *chased*, which is a Verb
 Accept *the*, which is a Determiner
 Accept *black*, which is an Adjective
 Accept *cat*, which is a Noun
 Determiner + Adjective + Noun make a Noun Phrase
 Verb + Noun Phrase make a Verb Phrase
 Noun Phrase + Verb Phrase make a Sentence

Parsing algorithms must be able to *backtrack* (back up and try alternatives) because the grammar provides alternatives. For example, a noun phrase may or may not contain an adjective, and a word like *leaves* may be a verb or a noun. Further, parsing algorithms usually use *recursion* to handle the recursive structure of human languages. For example, a noun phrase can contain a noun phrase, which can contain another noun phrase, as in *the discoverer of the solution to the problem*. See RECURSION; BACKTRACKING; NATURAL LANGUAGE PROCESSING.

PART 15 DEVICE a radio transmitter that is allowed to operate without a license under the terms of Part 15 of the Federal Communications Commission's regulations (known to lawyers as 47 CFR 15). Examples include cordless telephones, wireless intercoms, and some kinds of wireless computer communication devices (wireless LANs). Because individual Part 15 transmitters are not licensed, there is no way to guarantee that they will not interfere with each other, but spread-spectrum technology makes interference unlikely. By contrast, licensed transmitters can be given exclusive use of a particular frequency in a particular area. See also SPREAD-SPECTRUM.

PARTITION a part of a **HARD DISK** that is treated by the computer as if it were a separate disk drive. Naturally, most hard disks consist of only one partition. Multiple partitions are sometimes used with very large drives, especially if the drive is larger than those the operating system was designed for.

Under MS-DOS and OS/2, the **FDISK** command sets the partition structure of a hard disk. Any change in the partition structure makes the previous contents of the disk completely unreadable, so don't run **FDISK** unless you have backed up all your data first.

PASCAL a programming language developed by Niklaus Wirth in the early 1970s. Pascal is essentially a modernized version of ALGOL, and it has greatly influenced the design of other languages, as well as becoming popular in its own right. See **TURBO PASCAL** for information on a popular version.

A Pascal program consists of:

- A **PROGRAM** statement to give the program a name (and, in older versions, to declare input and output files);
- Declarations of global variables;
- Declarations of procedures and functions;
- The keyword **BEGIN**, the action part of the main program, the keyword **END**, and finally a period to mark the end.

Procedures and functions, in turn, can contain their own declarations, including more procedures and functions (a kind of nesting that is not permitted in C).

Pascal is not case-sensitive (e.g., *X* and *x* are equivalent). In this book, uppercase letters are used for *reserved words* (keywords that cannot be redefined, such as **IF**, **THEN**, and **ELSE**), and basic keywords of the language, which cannot be redefined) and lowercase letters for everything else.

Semicolons are used as separators *between* statements. Thus a statement will end with a semicolon only if what follows is the beginning of another statement. This contrasts with C and PL/I, which end every statement with a semicolon regardless of the context.

Comments in Pascal are enclosed in braces, { }, or the symbols (**). A statement does not have to fit on one line; lines can be broken anywhere that blank space is permitted.

Pascal provides four standard data types: real, integer, Boolean, and character. Integer variables can take on only values that are whole numbers or the negatives of whole numbers. Real variables can take on numerical values that include fractional parts, such as 23.432. Boolean variables are logic variables that can only have two possible values: true or false. Char variables can take on single character values. Here is an example of how to declare variables:

```
VAR count, total: INTEGER;
    average: REAL;
    initial: CHAR;
```

A character string is simply an array of char items. Most modern versions of Pascal, however, have added variable-length strings.

An assignment statement in Pascal looks like this:

```
x := 3;
```

This statement gives the value 3 to the variable *x*. Note that the symbol for assignment is **:=**, not **=**. The arithmetic operators are **+** for addition, **-** for subtraction, ***** for multiplication, **/** for floating-point division, **div** for integer division, and **mod** for modulo (remainder from division). Some implementations have added ****** for exponentiation.

Exhibit 8

WILEY ELECTRICAL AND ELECTRONICS ENGINEERING DICTIONARY

Steven M. Kaplan
Lexicographer



A JOHN WILEY & SONS, INC., PUBLICATION

Copyright © 2004 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400, fax 978-646-8600, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representation or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic format.

Library of Congress Cataloging-in-Publication Data is available.

Kaplan, Steven M.

Wiley Electrical and Electronics Engineering Dictionary

ISBN 0-471-40224-9

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

part failure

part failure The condition in which a part of a circuit, device, piece of equipment, or system no longer performs the function it was intended to, or is not able to do so at a level that equals or exceeds established minimums.

part identification Same as **part recognition**.

part recognition In robotics, the detection and identification of components or parts. Artificial vision may be used, for instance, to distinguish between parts that are nearly identical. Also called **part identification**, or **part detection**.

partial 1. Not a whole. Also, pertaining to that which is not a whole. 2. One of the multiple components of a complex tone, such as a harmonic, though it need not be a harmonic.

partial carry In computers, a carry technique in which the carries that result from an operation are temporarily stored. This contrasts with a **complete carry**, where carries are propagated to other digit positions.

partial node In a standing-wave system or vibrating system, a point between the minimum and maximum amplitude or displacement. The amplitude of a partial node lies between that of an antinode and that of a node.

partial pressure The pressure exerted by one of multiple gases which are mixed together.

Partial Response Maximum Likelihood A technique utilizing digital signal processing to differentiate a desired signal from a noise signal. Used, for instance, to increase the storage density of a magnetic disk. Its abbreviation is **PRML**.

particle 1. A very small or infinitesimal portion of matter, such as a molecule, atom, proton, or quark. 2. Any of the fundamental constituents of all matter in the universe. Elementary particles include protons, neutrons, electrons, photons, and quarks. Categories of such particles include leptons and hadrons. An electron, for instance, is a type of lepton. When an elementary particle is defined as having no internal structure, particles such as protons and neutrons are excluded, as these are composed of quarks. The four fundamental forces of nature are carried by elementary particles. For example, photons carry the electromagnetic force. Also called **elementary particle**. 3. A comparatively small subdivision of matter, such as magnetic particles or dust.

particle accelerator A device, such as a linac or cyclotron, in which charged particles are greatly accelerated to achieve high energies. Also called **accelerator** (4), or **high-energy particle accelerator**.

particle beam A narrow stream of particles traveling in the same direction, with approximately the same speed, and emitted from the same source. Electron beams and ion beams are examples.

particle bombardment The subjecting of a target to impacts by high-energy particles. Also called **bombardment**.

particle counter Also called **particle detector**. 1. An instrument which registers and counts ionizing radiation, such as alpha rays, given off by radioactive entities. For example, a Geiger counter or a scintillation counter. Also called **radiation counter**, **counter** (3), or **ionization counter**. 2. A device which registers and counts particles.

particle detector Same as **particle counter**.

particle fluence The passing of particles, such as electrons or photons, through a given area, such as a square centimeter. May be expressed, for instance, in particles per second. Also called **fluence**.

particle physics A branch of physics which deals with the properties, interactions, and applications of subatomic particles, especially when they have high energies. Also called **high-energy physics**.

particle velocity 1. The distance a particle travels per unit time. For instance, electrons in a linear accelerator may travel at nearly the speed of light. 2. In a sound field, the velocity of an infinitesimal part of the medium, relative to

the medium as a whole. May be expressed, for instance, in meters per second. A velocity microphone responds to particle velocity rather than to sound pressure.

particle-wave duality The dual nature of electromagnetic radiation, in which it has particle-like properties, as seen in the photoelectric effect, and wavelike properties, as evidenced by phenomena such as diffraction and interference. Certain particles such as electrons also exhibit this duality, as they have wavelike properties, as seen, for instance, in electron diffraction. Also called **wave-particle duality**.

partition One of multiple portions created by **partitioning**.

partition boot sector The first sector of a hard disk or a bootable floppy disk. During startup, its contained data and instructions start the process of loading the operating system into memory.

partition noise In an active device, such as a transistor or electron tube, noise resulting from the random distribution of electrons between electrodes.

partition table A table which lists the partitions of a hard disk. Such a table includes the information necessary to access each partition, and indicates which is used for booting.

partitioning 1. The dividing of a hard disk into multiple virtual disks, each of which appear as a separate disk to the operating system. For instance, a user may have drives labeled C:, D:, E:, and F:, but only have a single physical disk. Used, for instance, when setting up disks with different operating systems. 2. The dividing of computer memory into fixed portions. 3. The dividing of a block of data into fixed portions.

partner robots Same as **personal robots**.

parts per billion A unit of proportion, equal to a given amount of anything per billion of the same thing or something else. Used, for instance, as a measure of concentration or to quantify error. For example, five erroneous bits per billion transmitted bits. Its abbreviation is **ppb**.

parts per million A unit of proportion, equal to a given amount of anything per million of the same thing or something else. Used, for instance, as a measure of concentration or to quantify error. For example, nine atoms of xenon per million of air. Its abbreviation is **ppm**.

parts per thousand A unit of proportion, equal to a given amount of anything per thousand of the same thing or something else. Used, for instance, as a measure of concentration. For example, 35 molecules of salt per thousand of water. Its abbreviation is **ppt**.

party line An arrangement in which the telephones of two or more end users are connected to the same telephone line or local loop. When utilized, a mechanism such as selective ringing may be employed to distinguish for which user an incoming call is meant. Party lines are saddled by privacy and congestion problems. Also called **multiparty line**.

PAS Abbreviation of **photoacoustic spectroscopy**.

pascal The SI unit of pressure. It is equal to the pressure resulting from a force of one Newton applied uniformly to an area of one square meter. That is, $\text{Pa} = \text{N/m}^2$. It is also equal to approximately $9.869233 \times 10^{-6} \text{ atm}$, $1.0 \times 10^{-5} \text{ bar}$, about $7.500615 \times 10^{-3} \text{ mmHg}$, approximately 7.500617 torr , or about $0.0208854 \text{ lbf/ft}^2$. Its symbol is **Pa**.

Pascal A procedure-oriented high-level programming language which lends itself to structured programming.

Paschen-Back effect In spectroscopy, an effect in which spectral lines split by the Zeeman effect seem to coalesce when the light source is subjected to a strong magnetic field.

Paschen's law A law stating that the breakdown potential that initiates a discharge between two parallel plates in a gas is a function of the product of the gas pressure and the distance between said plates. Other factors such as gas surface irregularities and electrode geometry also affect the breakdown potential.

Exhibit 9

SECOND EDITION
COMPREHENSIVE
DICTIONARY
OF
ELECTRICAL
ENGINEERING

EDITOR-IN-CHIEF
Phillip A. Laplante



Taylor & Francis
Taylor & Francis Group

Boca Raton London New York Singapore

CRC PRESS, a Taylor & Francis title, part of the Taylor and Francis Group.

Neural AI, LLC, Exhibit 2010

Page 2010 - 81

IPR2025-00609, NVIDIA Corporation v. Neural AI, LLC

Published in 2005 by
CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2005 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-10: 0-8493-3086-6 (Hardcover)
International Standard Book Number-13: 978-0-8493-3086-5 (Hardcover)
Library of Congress Card Number 2004058572

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

No part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC) 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Comprehensive dictionary of electrical engineering / editor-in-chief Phillip A. Laplante.-- 2nd ed.
p. cm.

ISBN 0-8493-3086-6 (alk. paper)

1. Electric engineering--Dictionaries. I. Title: Electrical engineering. II. Laplante, Phillip A.

TK9.C575 2005
621.3'03--dc22

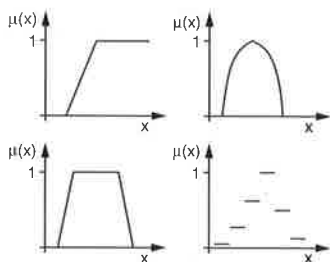
2004058572

T&F informa

Taylor & Francis Group
is the Academic Division of T&F Informa plc.

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>



Membership function: some examples of normalized fuzzy sets.

membership grade the degree to which an element or object belongs to a fuzzy set. It is also referred to as degree of membership.

membrane the functional boundary of a cell. Nerve cells possess membranes that are excitable by virtue of their nonlinear electrical conductance properties.

membrane probe used for performing high power on wafer test for large periphery MMIC power amplifiers and discrete devices with non-50-W interfaces.

memory (1) area for storing computer instructions and data for either short-term or long-term purposes.

(2) the property of a display pixel that allows it to remain stable in an initially established state of luminance. Memory gives a display high luminance and absence of flicker.

memory access time the time from when a read (i.e., load) request is submitted to memory to the time when the corresponding data becomes available. Usually smaller than the memory cycle time.

memory address computation the computation required to produce an effective memory address; may include indexing and translation from a virtual to a physical address.

memory address register (MAR) a register inside the CPU that holds the address of the memory location being accessed while the access is taking place.

memory alignment matching data to the physical characteristics of the computer memory. Computer memory is generally addressed in bytes, while memories handle data in units of 4, 8, or 16 bytes. If the “memory width” is 64 bits, then reading or writing an 8 byte (64 bit) quantity is more efficient if data words are aligned to the 64 bit words of the physical memory. Data that is not aligned may require more memory accesses and more-or-less complex masking and shifting, all of which slow the operations.

Some computers insist that operands be properly aligned, often raising an exception or interrupt on unaligned addresses. Others allow unaligned data, but at the cost of lower performance.

memory allocation the act of reserving memory for a particular process.

memory bandwidth the maximum amount of data per unit time that can be transferred between a processor and memory.

memory bank a subdivision of memory that can be accessed independently of (and often in parallel with) other memory banks.

memory bank conflict conflict when multiple memory accesses are issued to the same memory bank, leading to additional buffer delay for such accesses that reach the memory bank while it is busy serving a previous access. *See also* interleaved memory.

memory block contiguous unit of data that is transferred between two adjacent levels of a memory hierarchy. The size of a block will vary according to the distance from the CPU, increasing as levels get farther from the CPU, in order to make transfers efficient.

Exhibit 10

McGRAW-HILL DICTIONARY OF SCIENTIFIC AND TECHNICAL TERMS

**Sixth
Edition**

McGraw-Hill

New York Chicago San Francisco
Lisbon London Madrid Mexico City
Milan New Delhi San Juan Seoul Singapore Sydney Toronto
Neural AI, LLC, Exhibit 2010
Page 2010-85
IPR2025-00609, NVIDIA Corporation v. Neural AI, LLC

On the cover: Representation of a fullerene molecule with a noble gas atom trapped inside. At the Permian-Triassic sedimentary boundary the noble gases helium and argon have been found trapped inside fullerenes. They exhibit isotope ratios quite similar to those found in meteorites, suggesting that a fireball meteorite or asteroid exploded when it hit the Earth, causing major changes in the environment. (Image copyright © Dr. Luann Becker. Reproduced with permission.)

Over the six editions of the Dictionary, material has been drawn from the following references: G. M. Garrity et al., *Taxonomic Outline of the Prokaryotes*, Release 2, Springer-Verlag, January 2002; D. W. Linzey, *Vertebrate Biology*, McGraw-Hill, 2001; J. A. Pechenik, *Biology of the Invertebrates*, 4th ed., McGraw-Hill, 2000; U.S. Air Force Glossary of Standardized Terms, AF Manual 11-1, vol. 1, 1972; F. Casey, ed., *Compilation of Terms in Information Sciences Technology*, Federal Council for Science and Technology, 1970; *Communications-Electronics Terminology*, AF Manual 11-1, vol. 3, 1970; P. W. Thrush, comp. and ed., *A Dictionary of Mining, Mineral, and Related Terms*, Bureau of Mines, 1968; *A DOD Glossary of Mapping, Charting and Geodetic Terms*, Department of Defense, 1967; J. M. Gilliland, *Solar-Terrestrial Physics: A Glossary of Terms and Abbreviations*, Royal Aircraft Establishment Technical Report 67158, 1967; W. H. Allen, ed., *Dictionary of Technical Terms for Aerospace Use*, National Aeronautics and Space Administration, 1965; *Glossary of Stinfo Terminology*, Office of Aerospace Research, U.S. Air Force, 1963; *Naval Dictionary of Electronic, Technical, and Imperative Terms*, Bureau of Naval Personnel, 1962; R. E. Huschke, *Glossary of Meteorology*, American Meteorological Society, 1959; *ADP Glossary*, Department of the Navy, NAVSO P-3097; *Glossary of Air Traffic Control Terms*, Federal Aviation Agency; *A Glossary of Range Terminology*, White Sands Missile Range, New Mexico, National Bureau of Standards, AD 467-424; *Nuclear Terms: A Glossary*, 2d ed., Atomic Energy Commission.

McGRAW-HILL DICTIONARY OF SCIENTIFIC AND TECHNICAL TERMS, Sixth Edition

Copyright © 2003, 1994, 1989, 1984, 1978, 1976, 1974 by The McGraw-Hill Companies, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

1 2 3 4 5 6 7 8 9 0 DOW/DOW 0 8 7 6 5 4 3 2

ISBN 0-07-042313-X

Library of Congress Cataloging-in-Publication Data

McGraw-Hill dictionary of scientific and technical terms--6th ed.

p. cm.

ISBN 0-07-042313-X (alk. paper)

1. Science--Dictionaries. 2. Technology--Dictionaries. I. Title: Dictionary of scientific and technical terms.

Q123.M15 2002
503--dc21

Neural AI, LLC, Exhibit 2010
2002026436 Page 2010 - 86
IPR2025-00609, NVIDIA Corporation v. Neural AI, LLC

melting profile [BIOCHEM] A plot of the degree of denaturation of the strands in a nucleic acid duplex in a specified time as a function of temperature. { 'melt·iŋ ,prō·fl }
melting rate [MET] In electric arc welding, the weight or length of electrode melted in a specified unit of time. Also known as burn-off rate; melt-off rate. { 'melt·iŋ ,rāt }

melting ratio [MET] The ratio of metal weight to fuel weight in a melting process. { 'melt·iŋ ,rā·shō }

melting temperature [BIOCHEM] The temperature at which denaturing occurs for half of the double helices of deoxyribonucleic acid. { 'melt·iŋ ,tem·prə·chər }

melt instability [MECH] Instability of the plastic melt flow through a die. { 'melt ,in·stə·bīl·əd·ē }

melt loading [ORD] Process of melting solid explosive by heat and pouring into bombs, projectiles, and the like to solidify. Also known as cast loading. { 'melt ,lōd·iŋ }

melt-off rate See melting rate. { 'melt,ōf ,rāt }

melton [TEXT] A fabric with all-wool or cotton warp and woolen weft; the face is napped carefully to raise the nap straight up, showing the weave clearly. Also known as beaver cloth; kersey. { 'mel·tən }

melt spinning [TEXT] A process by which nylon, polyester, or glass is melted to allow it to be extruded into fibers through a spinneret. { 'melt,spin·iŋ }

melt strength [MECH] Strength of a molten plastic. { 'melt ,streŋkθ }

melt-through [NUCLEO] An accident in a nuclear reactor in which melting of the fuel core (meltdown) leads to runaway melting of nuclear fuel out of the bottom of the reactor, down through the concrete mat below, and into the earth. Also known as China syndrome. { 'melt ,θru }

meltwater [HYD] Water derived from melting ice or snow, especially glacier ice. { 'melt,wōd·ər }

Melusinidae [INV ZOO] A family of orthorrhaphous dipteran insects in the series Nematocera. { ,mel·ə·sīn·ə·dē }

Melyridae [INV ZOO] The soft-winged flower beetles, a large family of cosmopolitan coleopteran insects in the superfamily Cleroidea. { mə·līr·ə·dē }

member [CIV ENG] A structural unit such as a wall, column, beam, or tie, or a combination of any of these. [GEOL] A rock stratigraphic unit of subordinate rank comprising a specially developed part of a varied formation. [MATH] 1. An individual object that belongs to a set. Also known as element. 2. For an equation, the expression on either side of the equality sign. { 'mem·bər }

membership function [MATH] The characteristic function of a fuzzy set, which assigns to each element in a universal set a value between 0 and 1. { 'mem·bər,ʃɪp ,fəŋk·shən }

Membracidae [INV ZOO] The treehoppers, a family of homopteran insects included in the series Auchenorrhyncha having a pronotum that extends backward over the abdomen, and a vertical upper portion of the head. { mem·bras·əd·ē }

membrane [BUILD] In built-up roofing, a weather-resistant (flexible or semiflexible) covering consisting of alternate layers of felt and bitumen, fabricated in a continuous covering and surfaced with aggregate or asphaltic material. [CHEM ENG] 1. The medium through which the fluid stream is passed for purposes of filtration. 2. The ion-exchange medium used in dialysis, diffusion, osmosis and reverse osmosis, and electrophoresis. [HISTOL] A thin layer of tissue surrounding a part of the body, separating adjacent cavities, lining cavities, or connecting adjacent structures. { 'mem·brān }

membrane analogy [MECH] A formal identity between the differential equation and boundary conditions for a stress function for torsion of an elastic prismatic bar, and those for the deflection of a uniformly stretched membrane with the same boundary as the cross section of the bar, subjected to a uniform pressure. { 'mem·brān ə,nəl·ə·jē }

membrane bone See dermal bone. { 'mem·brān ,bɒn }

membrane carrier [CELL MOL] Any protein that facilitates the movement of small molecules across cell membranes. { 'mem·brān ,kar·ē·ər }

membrane curing See membrane waterproofing. { 'mem·brān ,kyūr·iŋ }

membrane distillation [CHEM ENG] A separation method that uses a nonwetting, microporous membrane, with a liquid feed phase on one side and a condensing permeate phase on

the other. Also known as membrane evaporation; thermopervaporation; transmembrane distillation. { 'mem·brān ,dīs·tə·lā·shən }

membrane evaporation See membrane distillation. { 'mem·brān ,vəp·ə·rā·shən }

membrane keyboard [COMPUT SCI] A flat keyboard, used with microcomputers and hand-held calculators, that consists of two closely spaced membranes separated by a flat sheet called a spacer with holes corresponding to the keys. { 'mem·brān ,kē,bɔrd }

membrane mimetic chemistry [ORG CHEM] The study of processes and reactions that have been developed by using information obtained from biological membrane systems. { 'mem·brān mī·med·ik 'kem·ə·strē }

membrane potential [PHYSIO] A potential difference across a living cell membrane. { 'mem·brān pə,tən·chəl }

membrane separation [CHEM ENG] The use of thin barriers (membranes) between miscible fluids for separating a mixture; a suitable driving force across the membrane, for example concentration or pressure differential, leads to preferential transport of one or more feed components. { 'mem·brān ,sep·ə·rā·shən }

membrane stress [MECH] Stress which is equivalent to the average stress across the cross section involved and normal to the reference plane. { 'mem·brān ,stres }

membrane waterproofing [CIV ENG] Curing concrete, especially in pavements, by spraying a liquid material over the surface to form a solid, impervious layer which holds the mixing water in the concrete. Also known as membrane curing. { 'mem·brān 'wōd·ər,pruf·iŋ }

membranous glomerulonephritis [MED] A type of glomerulonephritis characterized by thickening of the basement membrane due to deposition of electron-dense material. { 'mem·brə·nəs gla·mer·yə·lō·ne·frīd·əs }

membranous labyrinth [ANAT] The membranous portion of the inner ear of vertebrates. { 'mem·brə·nəs 'lab·ə·rɪnθ }

membranous pregnancy [MED] Gestation in which there has been a rupture of the amniotic sac and the fetus is in direct contact with the wall of the uterus. { 'mem·brə·nəs 'preg·nən·sē }

membranous urethra [ANAT] The part of the urethra between the two facial layers of the urogenital diaphragm. { 'mem·brə·nəs yu·rē·θrə }

MEMC See methoxyethylmercury chloride.

memex [COMPUT SCI] A hypothetical machine described by Vannevar Bush, which would store written records so that they would be available almost instantly by merely pushing the right button for the information desired. { 'me,meks }

memistor [ELEC] Nonmagnetic memory device consisting of a resistive substrate in an electrolyte; when used in an adaptive system, a direct-current signal removes copper from an anode and deposits it on the substrate, thus lowering the resistance of the substrate; reversal of the current reverses the process, raising the resistance of the substrate. { me·mīs·tər }

memomotion study [IND ENG] A technique of work measurement and methods analysis using a motion picture camera operated at less than normal camera speed. Also known as camera study; micromotion study. { 'mem·əm·mō·shən ,stəd·ē }

memory [COMPUT SCI] Any apparatus in which data may be stored and from which the same data may be retrieved; especially, the internal, high-speed, large-capacity working storage of a computer, as opposed to external devices. Also known as computer memory. [PSYCH] The recollection of past events or sensations, or the performance of previously learned skills without practice. { 'mem·rē }

memory address register [COMPUT SCI] A special register containing the address of a word currently required. { 'mem·rē 'ad·res ,rej·ə·stər }

memory bank [COMPUT SCI] A physical section of a computer memory, which may be designed to handle information transfers independently of other such transfers in other such sections. { 'mem·rē ,bæŋk }

memory buffer register [COMPUT SCI] A special register in which a word is stored as it is read from memory or just prior to being written into memory. { 'mem·rē 'baf·ər ,rej·ə·stər }

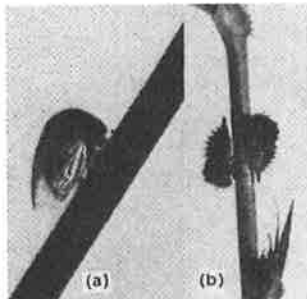
memory capacity See storage capacity. { 'mem·rē kə'pas·əd·ē }

memory card [COMPUT SCI] A small card, typically with Neural AI, LLC, Exhibit 2010

Page 2010 - 87

IPR2025-00609, NVIDIA Corporation v. Neural AI, LLC

MEMBRACIDAE



Membracids on stems. (a) Adult. (b) Nymphs. (Courtesy of C. H. Hanson)

Exhibit 11

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: GORCHETCHNIKOV et al.

Title: GRAPHIC PROCESSOR
BASED ACCELERATOR
SYSTEM AND METHOD

Appl. No.: 11/860254

Filing Date: 9/24/2007

Examiner: Maurice L. McDowell, Jr.

Art Unit: 2628

Confirmation 6266
Number:

AMENDMENT AND REPLY UNDER 37 CFR 1.111

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

This communication is responsive to the Non-Final Office Action dated September 16, 2010, concerning the above-referenced patent application.

Applicant has enclosed with this amendment a Petition to Revive under 37 CFR 1.137 to make this response timely.

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this document.

Remarks begin on page 9 of this document.

Please amend the application as follows:

U.S. Serial No. 11/860,254
 Reply to Office Action dated 9/16/2010
 Page 2 of 16

Atty. Dkt. No. 104821-0103

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer system for performing a numerical simulation over a plurality of computational cycles including at least a first computational cycle and a second computational cycle, the computer system comprising:
 - a central processing unit;
 - a main memory, operably coupled to the central processing unit, to store input data to be accessed by the central processing unit in performing the numerical simulation;
 - a video system, operably coupled to the central processing unit, to drive for driving a video monitor to display an indication of the numerical simulation in response to the central processing unit computer system performing the numerical simulation;
 - [[and]]
 - an accelerator, operably coupled to the central processing unit, to receive at least a portion of the receiving input data from the central processing unit and providing to provide first output data generated during the first computational cycle to the central processing unit after a conclusion of the first computational cycle, the accelerator comprising:
 - at least one graphics processing unit to generate second output data, during the second computational cycle, by performing at least one calculation on the first output data; and
 - an accelerator memory, operably coupled to for the at least one graphic processing unit, the accelerator memory comprising:
 - a first partition, referenced by a first pointer, to store the first output data during the second computational cycle; and
 - a second partition, referenced by a second pointer, to store the

U.S. Serial No. 11/860,254
 Reply to Office Action dated 9/16/2010
 Page 3 of 16

Atty. Dkt. No. 104821-0103

second output data generated during the second computational cycle; and
an accelerator controller, operably coupled to the accelerator memory and the
central processing unit, to transfer the at least the portion of ~~for moving~~ the input data
into the accelerator memory before the first computational cycle, moving to transfer the
first output data from the accelerator memory to the main memory during the second
computational cycle, to direct the second output data into the second partition during the
second computational cycle, and to swap the first pointer and the second pointer at the
conclusion of the second computational cycle such that the second output data becomes
an input for a third computational cycle of the plurality of computational cycles
~~supervising data exchange between the accelerator memory and at least one graphic~~
~~processing unit.~~

2. (Canceled).
3. (Currently Amended) [[A]] The computer system as claimed in claim 1, wherein the
accelerator controller ~~dictates~~ is configured to dictate an order of execution of instructions
to the at least one graphics processing unit.
4. (Canceled).
5. (Currently Amended) [[A]] The computer system as claimed in claim 1, wherein the
accelerator controller comprises an interface controller ~~that enables the accelerator to~~
communicate with the central processing unit over a bus of the computer system ~~with the~~
~~central processing unit.~~
6. (Currently Amended) [[A]] The computer system as claimed in claim 1, wherein the
accelerator memory comprises a texture memory bank ~~for storing simulation~~ to store the
at least the portion of the input data and the first output data and a shader memory bank
~~for storing to store instructions for performing a set of operations to be performed on the~~
~~simulation~~ at least the portion of the input data by the at least one graphic processing unit.

U.S. Serial No. 11/860,254
 Reply to Office Action dated 9/16/2010
 Page 4 of 16

Atty. Dkt. No. 104821-0103

7. (Currently Amended) [[A]] The computer system as claimed in claim 6, wherein the texture memory is partitioned into sections in which a first section holds input data patterns the first partition, the second partition, a third partition to store second section holds data textures representing internal variables, a fourth partition to store data a third section holds data textures used as input at a particular computation cycle of the plurality of computational cycles step, and a fourth section holds data textures used to store an output of a computational step on the at least one graphic processing unit.

8. (Currently Amended) [[A]] The computer system as claimed in claim 1, wherein the accelerator controller inputs the at least the portion of the input data into at least one graphic processing unit and a series of instructions [[to]] into the at least one graphic processing unit, which wherein the at least one graphics processing unit then successively executes the instructions on the at least the portion of the input data.

9. (Canceled).

10. (Currently Amended) An accelerator system for a computer system performing a numerical simulation, the accelerator system comprising:
 - at least one graphics processing unit to generate output data by performing at least one computation during a first computational cycle of the numerical simulation;
 - an accelerator memory, operably coupled to the at least one for the graphic graphics processing unit, to store data used to perform the at least one computation; and
 - an accelerator controller for moving, operably coupled to the accelerator memory and the at least one graphics processing unit, to execute:
 - (i) a computational stream controlling performance of the at least one computation by the at least one graphics processing unit;
 - (ii) an output stream controlling transfer of the output data between from the at least one graphics processing unit [[and]] to the accelerator memory during the first computational cycle; and

U.S. Serial No. 11/860,254
 Reply to Office Action dated 9/16/2010
 Page 5 of 16

Atty. Dkt. No. 104821-0103

(iii) an input stream controlling transfer of input data to the accelerator memory for use by the at least one graphics processing unit during a second computational cycle of the numerical simulation.

11. (Currently Amended) ~~A computer~~ The accelerator system as claimed in claim 10, wherein the accelerator controller ~~dictates~~ is configured to dictate an order of execution of instructions to the at least one graphics processing unit.
12. (Currently Amended) ~~A computer~~ The accelerator system as claimed in claim 10, wherein the accelerator controller ~~sends~~ is configured to send instructions to the at least one graphics processing unit, ~~which executes~~ wherein the at least one graphics processing unit is configured to execute the instructions on the input data, and wherein the accelerator controller is configured to transfer the output data from the accelerator memory to a main memory during the execution of the instructions by the at least one graphics processing unit, the accelerator controller transfers output data from the accelerator system.
13. (Currently Amended) ~~A computer~~ The accelerator system as claimed in claim 10, wherein the accelerator memory comprises a texture memory bank ~~for storing simulation~~ to store the input data and the output data and a shader memory bank ~~for storing~~ to store instructions for processing operations to be performed on the simulation input data by the at least one graphic processing unit.
14. (Currently Amended) ~~A computer~~ The accelerator system as claimed in claim 13, wherein the texture memory is partitioned into ~~sections in which~~ a first section holds partition to store the input data patterns, a second section holds data textures representing partition to store internal variables, a third section holds data textures used as partition to store data textures used as input at a particular computation cycle of the numerical simulation [[step]], and a fourth ~~section holds data textures used to store an~~ partition to store the output data of a computational step on the at least one graphic processing unit.

U.S. Serial No. 11/860,254
 Reply to Office Action dated 9/16/2010
 Page 6 of 16

Atty. Dkt. No. 104821-0103

15. (Canceled).
16. (Currently Amended) ~~A computer~~ The accelerator system as claimed in claim [[15]] 10, wherein the accelerator controller ~~performs~~ is configured to perform successive iterations computational cycles of the numerical simulation by feeding the output data generated by the at least one graphic processing unit from a previous computational cycle and the input data for a next ~~iteration~~ computational cycle into the at least one graphic processing unit.
17. (Currently Amended) A method for performing ~~a numerical simulations~~ simulation on input data in a computer system including a central processing unit and an accelerator, the method comprising:
 - ~~a central processing unit loading input data into an accelerator system from main memory of the central processing unit;~~
 - receiving, by an accelerator, first input data from the central processing unit;
 - transferring, by an accelerator controller, the first ~~transferring the~~ input data into a first partition, referenced by first pointer, of an accelerator memory before a first computational cycle of the numerical simulation ~~to a graphics processing unit with instructions to be performed on the input data;~~
 - performing, by at least one graphics processing unit during the first computational cycle, at least one calculation on the first portion of the input data as to generate first output data;
 - ~~the accelerator controller, transferring output data generated by the graphic processing unit to the central processing unit as output data;~~
 - storing, by the accelerator controller, the first output data into a second partition, referenced by a second pointer, of the accelerator memory; and
 - swapping the first pointer with the second pointer at the end of the first computational cycle, such that the first output data becomes an input for a second computational cycle of the numerical simulation.

U.S. Serial No. 11/860,254
 Reply to Office Action dated 9/16/2010
 Page 7 of 16

Atty. Dkt. No. 104821-0103

18. (Currently Amended) [[A]] The method as claimed in claim 17, further comprising:
sending, by the accelerator controller, sending instructions for performing the at
least one calculation to the at least one graphics processing unit, which executes the
instructions on data; and
during the execution of the instructions by the at least one graphics processing
unit, the accelerator controller transferring output data from the accelerator memory to
main memory of the central processing unit.
19. (Currently Amended) [[A]] The method as claimed in claim 17, further comprising:
partitioning the accelerator memory of accelerator system a into the first section
holding partition to store input data patterns, the second partition to store output data of a
computational step of the at least one graphic processing unit, a second section holding
data textures representing third partition to store internal variables, and a third section
holding fourth partition to store data textures used as input at a particular computation
cycle of the numerical simulation step, and a fourth section holding data textures used to
store an output of a computational step on the at least one graphic processing unit.
20. (Canceled)
21. (New) The method of claim 17, further comprising:
 transferring, by the accelerator controller, the first output data to the main memory
 during the second computational cycle.
22. (New) The computer system as claimed in claim 1, wherein the accelerator controller
 comprises a set of instructions stored on a memory.
23. (New) The computer system as claimed in claim 1, wherein the at least the portion of the
 input data represents an initial condition of the numerical simulation.

U.S. Serial No. 11/860,254
Reply to Office Action dated 9/16/2010
Page 8 of 16

Atty. Dkt. No. 104821-0103

24. (New) The accelerator system as claimed in claim 10, wherein the accelerator controller comprises a set of instructions stored in a memory.

U.S. Serial No. 11/860,254
Reply to Office Action dated 9/16/2010
Page 9 of 16

Atty. Dkt. No. 104821-0103

REMARKS

In response to the Office Action dated September 16, 2010, in the above-identified application, Applicants respectfully request reconsideration. All of the issues raised in the Office Action have been carefully considered and are addressed in this response. The application as now presented is believed to be in allowable condition.

I. Status of the Claims and Summary of Amendments

Upon entry of this response, claims 1, 3, 5–8, 10–19, and 21–24 will be pending in this application, of which claims 1, 10, and 17 are independent claims.

In this paper, claims 1, 3, 5–8, and 10–19 are amended, claims 2, 4, 9, 15, and 20 are canceled, and claims 21–24 are new.

No new matter is added by the amendments or new claims herein; for example, support for the amendments may be found throughout the specification and drawings, including Figure 2 and paragraphs [0030], [0038] and [0138]–[0144] as originally filed.

II. Brief Summary of the Claimed Subject Matter

The following summary of the claimed subject matter is provided to facilitate an understanding of some of the salient innovative and patentable concepts underlying the claims without limiting the claims.

In general, the independent claims are drawn to systems and methods for performing numerical simulations. In one example, the system includes a main memory operably coupled to a central processing unit (CPU). The system also includes a video system to drive a video monitor.

Additionally, the system includes an accelerator that is operably coupled to the CPU. The accelerator receives input data from the CPU at the beginning of a computational cycle, and provides the CPU with the output data from the numerical simulation at the end of the computational cycle. The accelerator includes at least one graphics processing unit (GPU), which performs the numerical simulation. In particular, the GPU uses output data from a particular

U.S. Serial No. 11/860,254
Reply to Office Action dated 9/16/2010
Page 10 of 16

Atty. Dkt. No. 104821-0103

computational cycle as input for the next computational cycle, which results in the generation of more output data.

The accelerator further includes an accelerator memory that is operably coupled to the GPU. The accelerator memory includes a first partition for storing the output data used as the input to particular computational cycle and a second partition for storing the output data generated during that computational cycle. Each partition is referenced by a respective pointer.

The system also includes an accelerator controller that is operably coupled to the accelerator memory and the CPU. The accelerator controller transfers the input data received from the CPU into the accelerator memory before the first computational cycle. The accelerator controller also transfers output data from the n^{th} computational cycle from the accelerator memory to the main memory during the $n+1^{\text{th}}$ computational cycle, and stores the output data into the second partition at the end of the $n+1^{\text{th}}$ computational cycle. Also, the accelerator controller swaps the first pointer and the second pointer at the conclusion of the $n+1^{\text{th}}$ computational cycle such that the output data from the $n+1^{\text{th}}$ computational cycle becomes an input for the $n+2^{\text{th}}$ computational cycle.

It should be appreciated that the foregoing summary is provided solely for the convenience of the Examiner, and is not intended to characterize the scope of any claim in any way. Accordingly, the Examiner is requested to rely upon the language specifically used in the claims to determine whether each of the claims distinguishes over the prior art of record.

III. Rejections Under 35 U.S.C. § 103

In the Office Action, claims 1, 4, 5, 10, 12, 17, and 18 (including independent claims 1, 10, and 17) were rejected under 35 U.S.C. § 103(a) as unpatentable over U.S. Patent Publication No. 2007/0052713 (hereafter “Chung”) in view of U.S. Patent Publication No. 2005/0166042 (hereafter “Evans”). More specifically, pages 2 and 3 of the Office Action state that it would have been obvious to combine the computer system of Chung with the accelerator of Evans to arrive at the computer system of claim 1. Applicants respectfully traverse these rejections to the extent they are maintained over the claims as amended herein.

U.S. Serial No. 11/860,254
Reply to Office Action dated 9/16/2010
Page 11 of 16

Atty. Dkt. No. 104821-0103

A. Independent Claims 1 and 17

Independent claim 1 as amended recites:

1. A computer system for performing a numerical simulation over a plurality of computational cycles including at least a first computational cycle and a second computational cycle, the computer system comprising:
 - a central processing unit;
 - a main memory, operably coupled to the central processing unit, to store input data to be accessed by the central processing unit in performing the numerical simulation;
 - a video system, operably coupled to the central processing unit, to drive a video monitor to display an indication of the numerical simulation in response to the computer system performing the numerical simulation;
 - an accelerator, operably coupled to the central processing unit, to receive at least a portion of the input data from the central processing unit and to provide first output data to the central processing unit at a conclusion of the first computational cycle, the accelerator comprising:
 - at least one graphics processing unit to generate second output data, during the second computational cycle, by performing at least one calculation on the first output data; and
 - an accelerator memory, operably coupled to for the at least one graphic processing unit, the accelerator memory comprising:
 - a first partition, referenced by a first pointer, to store the at least the portion of the input data and the first output data; and
 - a second partition, referenced by a second pointer, to store the second output data; and
 - an accelerator controller, operably coupled to the accelerator memory and the central processing unit, to transfer the at least the portion of the input data into the accelerator memory before the first computational cycle, to transfer the first output data from the accelerator memory to the main memory during the second computational cycle, to direct the second output data into the second partition at the end of the second computational cycle, and to swap the first pointer and the second pointer at the conclusion of the second computational cycle such that the second output data becomes an input for a third computational cycle of the plurality of computational cycles.

Claim 17 recites a method with similar limitations to those recited in claim 1. Applicants respectfully submit that Chung and Evans, alone or in combination, fail to teach or suggest all of the elements recited in claims 1 and 17. More particularly, the cited references fail to teach or suggest at least the “accelerator controller” recited in both of these independent claims.

U.S. Serial No. 11/860,254
Reply to Office Action dated 9/16/2010
Page 12 of 16

Atty. Dkt. No. 104821-0103

On page 2, the Office Action concedes that Chung does not teach an accelerator controller. The Office Action then relies on Evans to purportedly provide the accelerator controller that is missing from Chung.

More specifically, on page 3, the Office Action refers to Figures 3, 4, and the paragraphs describing Figures 3 and 4 (paragraphs [0061] and [0064], respectively) of Evans as providing purportedly relevant disclosure in connection with the accelerator controller. The cited paragraphs of Evans read:

[0061] Step 300 provides unencrypted data on the video card. This data is the data that is typically processed by the video card's GPU and is provided in the video card's memory. Step 302 encrypts the unencrypted data on the video card. This step can be implemented in response to a request to transfer the data to the system's memory. For example, the CPU may send an instruction to move the data from the video card to the system's memory. Alternately, the GPU may determine that it does not need the data any longer and therefore initiates a transfer to the system's memory. The data can be encrypted on the video card using any suitable encryption techniques. Step 304 then sends the encrypted data over a bus between the video card and the system. Exemplary buses can be the PCI bus and the AGP bus. Step 306 provides the encrypted data into system memory.

[0064] Step 400 provides data in the system memory. Such data is typically data that is utilized by the video card for processing. The data can be provided in the system memory as either encrypted or unencrypted data. Step 402 can encrypt the data (if it is unencrypted in the system memory) and can send the data to a bus between the system memory and the video card. This step can be implemented in response to a desire to process the data on the video card. Step 404 receives the encrypted data on the video card. Step 406 then decrypts the encrypted data on the video card and step 408 provides the decrypted data into video memory for processing by the GPU.

The above passage of Evans, like the rest of Evans' disclosure, is completely silent with respect to the salient limitations of Applicants' independent claim 1. More specifically, nowhere in the reference does Evans disclose or suggest an accelerator controller, operably coupled to the accelerator memory and the central processing unit, to transfer the at least the portion of the input data into the accelerator memory before the first computational cycle, to transfer the first output

U.S. Serial No. 11/860,254
 Reply to Office Action dated 9/16/2010
 Page 13 of 16

Atty. Dkt. No. 104821-0103

data from the accelerator memory to the main memory during the second computational cycle, to direct the second output data into the second partition at the end of the second computational cycle, and to swap the first pointer and the second pointer at the conclusion of the second computational cycle such that the second output data becomes an input for a third computational cycle of the plurality of computational cycles as recited in independent claim 1. Similarly,

Instead, at best the disclosure of Evans suggests an element that simply moves data into and out of a video card. There is no disclosure whatsoever in Evans of an accelerator controller that swaps respective pointers referencing a portion of the input data and the output data at the end of a first computational cycle as recited in claim 1. Therefore, no combination of Chung and Evans yields an accelerator controller as recited in claim 1 as amended.

Like claim 1 as amended, independent claim 17 as amended recites an accelerator controller that transfers “the first input data into a first partition, referenced by first pointer, of an accelerator memory before a first computational cycle of the numerical simulation” and that loads first output data generated during the first computational cycle “into a second partition, referenced by a second pointer, of the accelerator memory.” And Claim 17 as amended also recites “swapping the first pointer with the second pointer at the end of the first computational cycle, such that the first output data becomes an input for a second computational cycle of the numerical simulation.” Thus, like the subject matter of claim 1 as amended, the subject matter of claim 17 as amended is patentably distinct from Chung and Evans.

Given these distinctions, claims 1 and 17 as now presented are patentable over Chung and Evans. Claims 4 and 5 depend from claim 1 and claim 18 depends from claim 17, and thus are patentable for at least the same reasons. Accordingly, Applicants respectfully request withdrawal of the rejections under 35 U.S.C. § 103(a) of these claims.

B. Claim 10

Independent claim 10 as amended recites:

10. (Currently Amended) An accelerator system for a computer system performing a numerical simulation, the accelerator system comprising:

U.S. Serial No. 11/860,254
 Reply to Office Action dated 9/16/2010
 Page 14 of 16

Atty. Dkt. No. 104821-0103

at least one graphics processing unit to generate output data by performing at least one computation during a first computation cycle of the numerical simulation;

an accelerator memory, operably coupled to the at least one graphics processing unit, to store data used to perform the at least one computation; and
 an accelerator controller, operably coupled to the accelerator memory and the at least one graphics processing unit, to execute:

- (i) a computational stream controlling performance of the first computation by the at least one graphics processing unit;
- (ii) an output stream controlling transfer of the output data from the at least one graphics processing unit to the accelerator memory at the end of the first computation cycle; and
- (iii) an input stream controlling transfer of input data to the accelerator memory for use by the at least one graphics processing unit during a second computation cycle of the numerical simulation.

Applicants respectfully submit that Chung and Evans, alone or in combination, fail to disclose or suggest all of the elements recited in claim 10. More specifically, nowhere does Chung or Evans disclose or suggest an accelerator controller that executes a computational stream to control performance of a first computation, an output stream to transfer data at the end of the first computational cycle, and an input stream to control the transfer of input data into the accelerator memory. Therefore, no combination of Chung and Evans yields an accelerator system that includes the accelerator controller of claim 10 as amended.

Accordingly, claim 10 is patentable over Chung and Evans for at least the reasons set forth above. Claim 12 depends on Claim 10 and thus is patentable for at least the same reasons. Accordingly, Applicants respectfully request withdrawal of the rejections under 35 U.S.C. § 103(a) of these claims.

C. Claims 3, 6–9, 11, 13–16, 19, and 20

In the Office Action, dependent claims 3, 6–9, 11, 13–16, 19, and 20 were rejected under 35 U.S.C. § 103(a) as allegedly obvious over Chung in view of Evans in further view of U.S. Patent No. 5,388,206 (hereafter “Poulton”). Applicants respectfully traverse these rejections. In any event, as each of the above-identified claims depends from independent claims 1, 10 or 17,

U.S. Serial No. 11/860,254
Reply to Office Action dated 9/16/2010
Page 15 of 16

Atty. Dkt. No. 104821-0103

each of which are in patentable condition, the rejections of these dependent claims respectfully are believed to be moot.

IV. General Comments on Dependent Claims

Since each of the dependent claims depends from a base claim that is believed to be in condition for allowance, Applicants believe that it is unnecessary at this time to argue the allowability of each of the dependent claims individually. However, Applicants do not necessarily concur with the interpretation of the dependent claims as set forth in the Office Action, nor do the Applicants concur that the basis for the rejection of any of the dependent claims is proper. Therefore, Applicants reserve the right to specifically address the patentability of the dependent claims in the future.

CONCLUSION

It is respectfully believed that all of the rejections set forth in the Office Action have been addressed. However, the absence of a reply to a specific rejection, objection, or comment set forth in the Office Action does not signify agreement with or concession of that rejection, objection, or comment. In addition, there may be reasons for patentability of any or all pending claims (or other claims) that have not been expressed. Furthermore, nothing in this paper should be construed as an intent to concede any issue with regard to any claim.

Applicants believe that the present application is now in condition for allowance. Favorable reconsideration of the application as amended is respectfully requested.

The Examiner is invited to contact the undersigned by telephone if it is felt that a telephone interview would advance the prosecution of the present application.

The Commissioner is hereby authorized to charge any additional fees which may be required regarding this application under 37 C.F.R. §§ 1.16–1.17, or credit any overpayment, to Deposit Account No. 19-0741. Should no proper payment be enclosed herewith, as by the credit card payment instructions in EFS-Web being incorrect or absent, resulting in a rejected or incorrect credit card transaction, the Commissioner is authorized to charge the unpaid amount to

U.S. Serial No. 11/860,254
Reply to Office Action dated 9/16/2010
Page 16 of 16

Atty. Dkt. No. 104821-0103

Deposit Account No. 19-0741. If any extensions of time are needed for timely acceptance of papers submitted herewith, Applicants hereby petition for such extension under 37 C.F.R. § 1.136 and authorizes payment of any such extensions fees to Deposit Account No. 19-0741.

Respectfully submitted,

Date July 31, 2013

By /Christopher Max Colice/

FOLEY & LARDNER LLP
Customer Number: 48329
Telephone: (617) 502-3287
Facsimile: (617) 342-4001

Christopher Max Colice
Attorney for Applicant
Registration No. 65,634

Exhibit 12

**EXHIBIT 15 – U.S. PATENT NO. RE48438 VS NVIDIA ACCELERATED COMPUTING PRODUCTS & SERVICES –
HOPPER ARCHITECTURE**

I. INTRODUCTION

The chart below demonstrates how Defendant Nvidia Corporation (“Nvidia” or “Defendant”), as well as Defendant’s partners, customers, and end users, directly infringe, either literally or under the doctrine of equivalents, claims 1-10, 12-14, 16-18, 20-32, and 40-54 of U.S. Patent No. RE48438 (“the ’438 Patent”).

Nvidia has committed acts of infringement within this District. Nvidia uses the Accused Products in this District in manners that practice the ’438 Patent, including by testing the Accused Products and by using the Accused Products at its offices and premises in this District. Defendant makes, uses, advertises, offers for sale, and/or sells hardware for accelerated computing, including GPUs, CPUs, and SoCs; computers for accelerated computing (e.g., supercomputers, servers, and data centers for high performance computing); and computer platform software-as-a-service (“SaaS”) that implements accelerated computing (including the Accused Products) in the State of Texas and in this District directly and/or through its partnerships

Defendant also provides data center and HPC services that practice the ’438 Patent in the State of Texas and in this District directly and/or through its partnerships with businesses in the State of Texas and in this District.

Nvidia sells, offers for sale, advertises, makes, installs, and/or otherwise provides hardware, software, firmware, and/or computer platforms for accelerated computing and data center and HPC services, including the Accused Products, the use of which infringes the ’438 Patent in this District and the State of Texas. (See <https://www.nvidia.com/en-us/data-center/solutions/accelerated-computing/>.) Nvidia performs these acts directly and/or through its partnerships with other entities. (See *id.* (“NVIDIA has defined a range of accelerated platforms that each consist of hardware systems designed according to the needs of the use case as well as the software stack that enables the operation and management of the business applications. These hardware systems and software are available from NVIDIA and our partners.”).)

Nvidia also uses a network of partners, which comprise re-sellers, managed service providers, and product and solution experts, to provide the Accused Products and implementation services for the Accused Products to customers in this District. Each of these partners sells, offers for sale, installs, and/or implements Nvidia’s accelerated computing hardware, software, and/or computer platform services. (See <https://www.nvidia.com/en-us/about-nvidia/partners/>.)

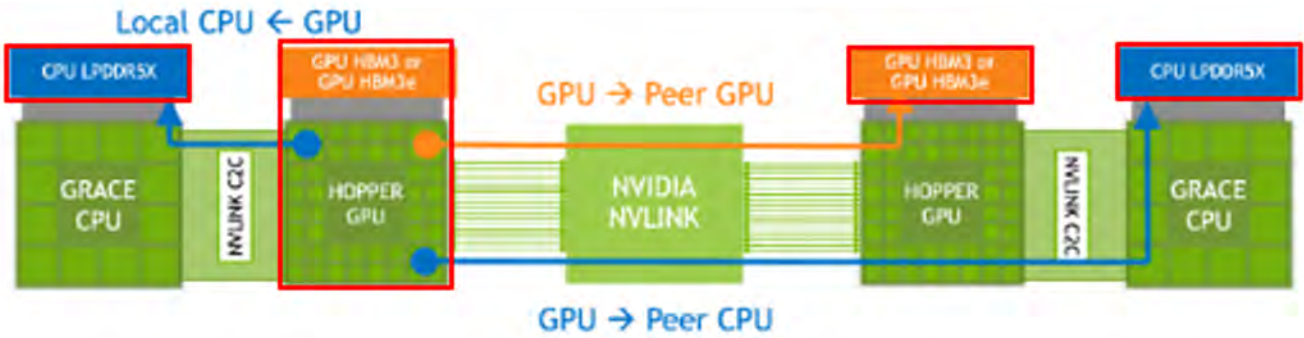
Nvidia’s partners include “Data Center Provider[s].” (See <https://www.nvidia.com/en-us/about-nvidia/partners/>.) Nvidia’s Data Center Provider partners “offer colocation services such as high-density data center facilities, interconnected infrastructure, and state-of-art

Key Features	Accused Products
	Relatedly, cuDNN functions being used for accessing data or outputs from RNN layers are depicted below. When the user selected <code>algo0 (CUDNN_RNN_ALGO_STANDARD)</code> in <code>cudaRnnBackwardData_v8()</code> or invoked the legacy functions, such as <code>cudaRnnBackwardDataEx()</code>, <code>cudaRnnBackwardData()</code>, and the <u>number of RNN layers</u> was more than eight in a unidirectional model or more than four in a bidirectional model, then some internal streams used to parallelize computations may be default streams (aka stream 0). The computational performance would most likely be affected in those cases. This issue has been fixed in this release. (See https://docs.nvidia.com/deeplearning/cudnn/archives/cudnn-893/release-notes/index.html#rel-831 (emphasis added).)
[6] The computer system of claim 1, wherein accelerator memory comprises: a first memory bank to store parameters common to all of the computations in the sequence of computations; and a second memory bank to store data specific to at least one computation in the sequence of computations.	The Accused Products embody <i>the computer system of claim 1, wherein accelerator memory comprises: a first memory bank to store parameters common to all of the computations in the sequence of computations; and a second memory bank to store data specific to at least one computation in the sequence of computations.</i> <i>See [1.6] (a controller, operably coupled to the at least one graphics processing unit and the accelerator memory, to initialize textures and shaders in the accelerator memory for performing the sequence of computations, to control performance of the sequence of computations by the at least one graphics processing unit, to transfer the at least a portion of the input data into the accelerator memory during performance of the intermediate computations in the sequence of computations by the at least one graphics processing unit, and to transfer at least a portion of the output data from the accelerator memory to the main memory during performance of the intermediate computations in the sequence of computations by the at least one graphics processing unit), supra.</i> For instance, the “CUDA programming model” includes programmatic functions, primitives, and executable libraries for CPUs and GPUs that are used by the Accused Products to perform numerical simulations. “The host is the CPU available in the system” and “system memory associated with the CPU is called host memory” (e.g., <i>a first memory bank to store parameters common to all of the computations</i>

Key Features	Accused Products
	<i>in the sequence of computations</i>). As an example, the first main CUDA program execution step is “[c]opy[ing] the input data from host [CPU] memory.” After the “host-to-device transfer,” the second main step is “[l]oad the GPU program and execute, caching data on-chip for performance” (e.g., <i>a second memory bank to store data specific to at least one computation in the sequence of computations</i>). Let me introduce two keywords widely used in <u>CUDA programming model: host and device.</u> <u>The host is the CPU available in the system.</u> The system memory associated with the CPU is called host memory. The GPU is called a device and GPU memory likewise called device memory. To execute any CUDA program, there are three main steps: • <u>Copy the input data from host memory to device memory, also known as host-to-device transfer.</u> • Load the GPU program and execute, <u>caching data on-chip for performance.</u> • Copy the results from device memory to host memory, also called device-to-host transfer. (See https://developer.nvidia.com/blog/cuda-refresher-cuda-programming-model/ (emphasis added).) As a further example, “shared memory” of the Accused Products is partitioned into “memory modules (banks)” (e.g., <i>a first memory bank and a second memory bank</i>). CUDA C++ Best Practices Guide The <u>programming guide to using the CUDA Toolkit to obtain the best performance from</u> <u>NVIDIA GPUs.</u> * * * * *

Key Features	Accused Products
	9.2.3. Shared Memory Because it is on-chip, shared memory has much higher bandwidth and lower latency than local and global memory - provided there are no bank conflicts between the threads, as detailed in the following section. 9.2.3.1. Shared Memory and Memory Banks To achieve high memory bandwidth for concurrent accesses, shared memory is divided into equally sized memory modules (<u>banks</u>) that can be accessed simultaneously. Therefore, any memory load or store of n addresses that spans n distinct memory banks can be serviced simultaneously, yielding an effective bandwidth that is n times as high as the bandwidth of a single bank. However, if multiple addresses of a memory request map to the same memory bank, the accesses are serialized. The hardware splits a memory request that has bank conflicts into as many separate conflict-free requests as necessary, decreasing the effective bandwidth by a factor equal to the number of separate memory requests. The one exception here is when multiple threads in a warp address the same shared memory location, resulting in a broadcast. In this case, multiple broadcasts from different banks are coalesced into a single multicast from the requested shared memory locations to the threads. To minimize bank conflicts, it is important to understand how memory addresses map to memory banks and how to optimally schedule memory requests. On devices of compute capability 5.x or newer, each bank has a bandwidth of 32 bits every clock cycle, and successive 32-bit words are assigned to successive banks. The warp size is 32 threads and the number of banks is also 32, so bank conflicts can occur between any threads in the warp. See Compute Capability 5.x in the CUDA C++ Programming Guide for further details. (See https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html#shared-memory/ (emphasis added).) As an example, the Grace Hopper Superchip “is designed to accelerate applications” using “Extended GPU Memory.” As depicted in the diagram of the Grace Hopper architecture below, a GPU (“HOPPER GPU”) can access “Local CPU,” “Peer CPU,” and “Peer GPU” memory via “NVLink.”

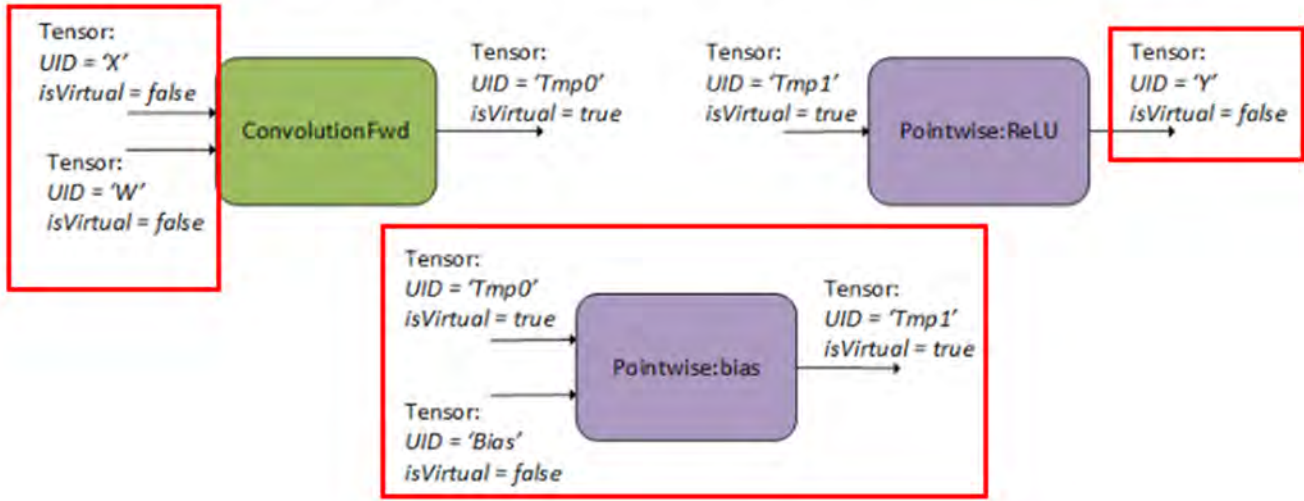
Key Features	Accused Products
	Accelerating Applications with Extended GPU Memory <u>The NVIDIA GH200 is designed to accelerate applications with exceptionally large memory footprints, larger than the capacity of the HBM3 / HBM3e and LPDDR5X memory of a single superchip (see the NVIDIA GH200 Accelerated Applications section below).</u> The Extended GPU Memory (EGM) feature over the high-bandwidth NVLink-C2C enables GPUs to access all the system memory efficiently. EGM provides up to 19.5TBs system memory in a multi-node NVSwitch-connected system. With EGM, physical memory in the system can be allocated to be accessible from any GPU thread. All GPUs can access EGM at the minimum of GPU-GPU NVLink or NVLink-C2C speed. Memory accesses within a Grace Hopper Superchip configuration go through the local high-bandwidth NVLink-C2C at 900GB/s total. Remote memory accesses are performed via GPU NVLink, and depending on the memory being accessed, also NVLink-C2C as shown in Figure 5. With EGM, GPU threads can now access all memory resources available over the NVSwitch fabric, both LPDDR5X and HBM3 or HBM3e, unidirectionally at 450GB/s.

Key Features	Accused Products
	 Figure 5. Memory Accesses across NVLink-connected Grace Hopper Superchips (See https://resources.nvidia.com/en-us-grace-cpu/nvidia-grace-hopper (emphasis added).) For instance, CUDA library cuDNN function “<code>cudaSetRNNDescriptor_v8</code>” “initializes a previously created RNN [recurrent neural network] descriptor object.” This function “store[s] all information needed to compute the total number of adjustable weights/biases in the RNN model” (e.g., <i>a first memory bank to store parameters common to all of the computations in the sequence of computations</i>). In addition, the parameters “<code>dirMode</code>,” “<code>inputMode</code>,” and “<code>datatype</code>” confirm the exchange of calculations and values between the hidden layers of an RNN (e.g., <i>a second memory bank to store data specific to at least one computation in the sequence of computations</i>).

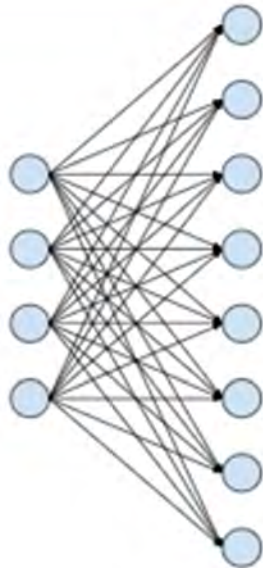
Key Features	Accused Products
	7.2.49. <code>cudaSetRNNDescriptor_v8()</code> This function initializes a previously created RNN descriptor object. The RNN descriptor configured by <u><code>cudaSetRNNDescriptor_v8()</code></u> was enhanced to store all information needed to compute the total number of adjustable weights/biases in the RNN model. <pre> cudaStatus_t cudaSetRNNDescriptor_v8(cudaRNNDescriptor_t rnnDesc, cudaRNNAlgo_t algo, cudaRNNMode_t cellMode, cudaRNNBiasMode_t biasMode, cudaRNNDirectionMode_t dirMode, cudaRNNInputMode_t inputMode, cudaDataType_t dataType, cudaDataType_t mathPrec, cudaMathType_t mathType, int32_t inputSize, int32_t hiddenSize, int32_t projSize, int32_t numLayers, cudaDropoutDescriptor_t dropoutDesc, uint32_t auxFlags); </pre> * * * * * dirMode <i>Input.</i> Specifies the recurrence pattern: CUDNN_UNIDIRECTIONAL or CUDNN_BIDIRECTIONAL. <u>In bidirectional RNNs, the hidden states passed between physical layers are concatenations of forward and backward hidden states.</u> inputMode <i>Input.</i> <u>Specifies how the input to the RNN model is processed by the first layer.</u> When inputMode is CUDNN_LINEAR_INPUT, original input vectors of size inputSize are multiplied by the weight matrix to obtain vectors of hiddenSize. When inputMode is CUDNN_SKIP_INPUT, the original input vectors to the first layer are used as is without multiplying them by the weight matrix. dataType <i>Input.</i> <u>Specifies data type for RNN weights/biases and input and output data.</u>

Key Features	Accused Products
	(See https://docs.nvidia.com/deeplearning/cudnn/archives/cudnn-891/pdf/cuDNN-API.pdf (emphasis added).) In another cuDNN example, an excerpt from Nvidia’s cuDNN release notes exemplifies “numerical results” from RNN API calls as used for deep learning using recurrent neural networks. A “kernel selection,” which prepares memory space and resources for use, “may affect numerical results.” cuDNN Release Notes <u>NVIDIA CUDA Deep Neural Network (cuDNN) is a GPU-accelerated library of primitives for deep neural networks. It provides highly tuned implementations of routines arising frequently in DNN applications. These release notes describe the key features, software enhancements and improvements, and known issues for the NVIDIA cuDNN 8.9.3 and earlier releases.</u> * * * * * • RNN and multihead attention API calls may exhibit nondeterministic behavior when the cuDNN library is built with CUDA Toolkit 10.2 or higher. This is the result of a new buffer management and heuristics in the cuBLAS library. As described in <u>Results Reproducibility</u>, <u>numerical results may not be deterministic when cuBLAS APIs are launched in more than one CUDA stream using the same cuBLAS handle. This happens when two buffer sizes (16 KB and 4 MB) are used in the default configuration.</u> When a larger buffer size is not available at runtime, instead of waiting for a buffer of that size to be released, a smaller buffer may be used with a different GPU kernel. <u>The kernel selection may affect numerical results.</u> The user can eliminate the nondeterministic behavior of cuDNN RNN and multihead attention APIs, by setting a single buffer size in the <u>CUBLAS_WORKSPACE_CONFIG</u> environmental variable, for example, <u>:16:8</u> or <u>:4096:2</u>. The first configuration instructs cuBLAS to allocate eight buffers of 16 KB each in GPU memory while the second setting creates two buffers of 4 MB each. The default buffer configuration in cuBLAS 10.2 and 11.0 is <u>:16:8:4096:2</u>, that is, we have two buffer sizes. In earlier cuBLAS libraries, such as cuBLAS 10.0, it used the <u>:16:8</u> non-adjustable configuration. When buffers of only one size are available, the behavior of cuBLAS calls is deterministic in multi-stream setups.

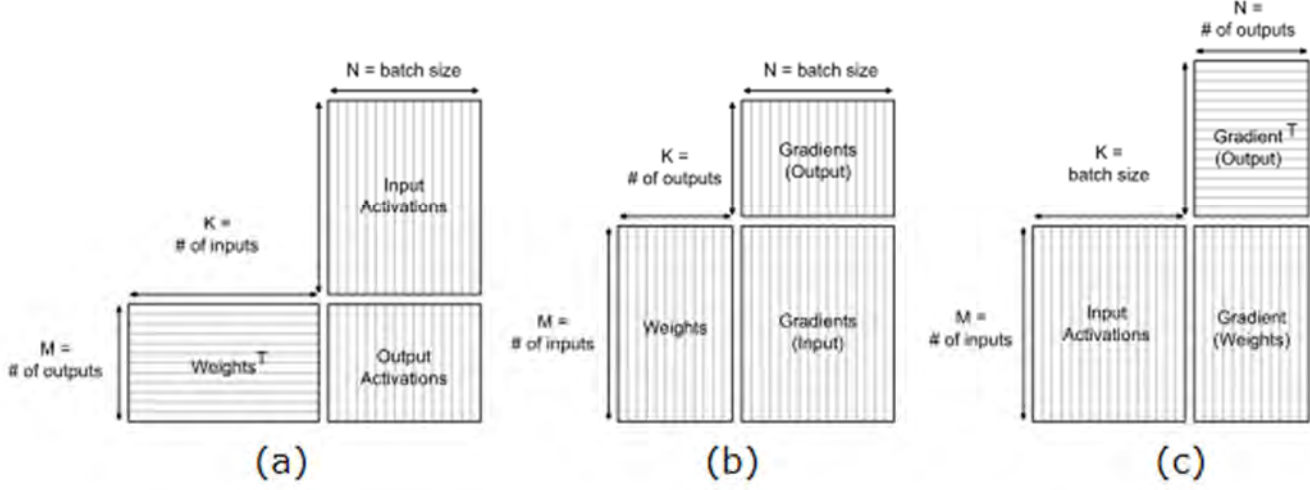
Key Features	Accused Products
	(See https://docs.nvidia.com/deeplearning/cudnn/archives/cudnn-893/release-notes/index.html (emphasis added).) In addition, cuDNN operations below exemplify “tak[ing] tensors as input and produc[ing] tensors as output.” As shown below for a “forward convolution operation” using cuDNN, “backend tensor <i>Tmp0</i> [is] both the output of the convolution operation and the input of the bias operation.” From this, “cuDNN infers that the dataflow runs from the convolution into the bias.” 2.2. Tensors and Layouts Whether using the graph API or the legacy API, cuDNN operations take tensors as input and produce tensors as output * * * * * 3.2. Graph API Example with Operation Fusion The following example implements a fusion of convolution, bias, and activation. 3.2.1. Creating Operation and Tensor Descriptors to Specify the Graph Dataflow First, create three cuDNN backend operation descriptors. As can be seen in Figure 6, the user specified one forward convolution operation (using <code>CUDNN_BACKEND_OPERATION_CONVOLUTION_FORWARD_DESCRIPTOR</code>), a pointwise operation for the bias addition (using <code>CUDNN_BACKEND_OPERATION_POINTWISE_DESCRIPTOR</code> with mode <code>CUDNN_POINTWISE_ADD</code>), and a pointwise operation for the ReLU activation (using <code>CUDNN_BACKEND_OPERATION_POINTWISE_DESCRIPTOR</code> with mode <code>CUDNN_POINTWISE_RELU_FWD</code>). Refer to the NVIDIA cuDNN Backend API for more details on setting the attributes of these descriptors. For an example of how a forward convolution can be set up, refer to the Setting Up An Operation Graph For A Grouped Convolution use case in the cuDNN backend API. You should also create tensor descriptors for the inputs and outputs of all of the operations in the graph. The graph dataflow is implied by the assignment of tensors (refer to Figure 6), for example, by specifying the backend tensor <i>Tmp0</i> as both the output of the convolution operation and the input of the bias operation, cuDNN infers that the dataflow runs from the convolution into the bias. The same applies to tensor <i>Tmp1</i>. If the user doesn't need the intermediate results <i>Tmp0</i> and <i>Tmp1</i> for any other use, then the user can specify them to be virtual tensors, so the memory I/Os can later be optimized out. * * * * *

Key Features	Accused Products
	Figure 6. A set of operation descriptors the user passes to the operation graph  (See https://docs.nvidia.com/deeplearning/cudnn/archives/cudnn-881/developer-guide/index.html#tensors-layouts (emphasis added).) An example below illustrates an exemplary neural network the Accused Products are designed to accelerate using parallel computations. “Input” (four) and “Output” (eight) neurons are depicted below in a full-connected or linear layer structure in which all of the input neurons depicted in a first layer are connected to all of the output neurons depicted in a second layer. Computations for the neural network are performed using, for example, “NVIDIA Matrix Multiplication.” Examples of inputs and outputs for forward propagation, activation gradient computation, and weight gradient computation (as matrix by matrix multiplications) for GEMMs (General Matrix Multiplications) are shown below. For example, for “(a) forward propagation . . . of a fully-connected layer” (the process of feeding input data through a neural network to generate an output), “K = # of inputs” and “M = # of outputs” and in “N = batch size” with results for “Input Activations,” “Output Activations,” and “Weights” (e.g., <i>a first memory bank to store parameters common to all of the computations in the sequence of computations and a second memory bank to store data specific to at least one computation in the sequence of computations</i>).

Key Features	Accused Products
	As a rough guideline, choose batch sizes and neuron counts greater than 128 to avoid being limited by memory bandwidth (NVIDIA® A100-SXM4-80GB; this threshold is similar for other A100 and V100 GPUs); see <u>Batch Size</u>. <u>2. Fully-Connected Layer</u> <u>Fully-connected layers, also known as linear layers, connect every input neuron to every output neuron and are commonly used in neural networks.</u> <i>Figure 1. Example of a small fully-connected layer with <u>four input and eight output neurons.</u></i>

Key Features	Accused Products
	 The diagram illustrates a fully-connected layer within a neural network. It consists of two vertical columns of blue circular nodes. The left column is labeled 'Input Neurons' and contains five nodes. The right column is labeled 'Output Neurons' and contains six nodes. Every node in the input column is connected to every node in the output column by a thin black line, representing a complete bipartite graph. The entire diagram is enclosed in a red rectangular border. Three parameters define a fully-connected layer: batch size, number of inputs, and number of outputs. <u>Forward propagation, activation gradient computation, and weight gradient computation are directly expressed as matrix-matrix multiplications.</u> How the three parameters map to GEMM dimensions (General Matrix Multiplication, background in the <u>NVIDIA Matrix Multiplication Background User's Guide</u>) varies among frameworks, but the underlying principles are the same. For the purposes of the discussion, we adopt the convention used by PyTorch and Caffe where A contains the weights and B the activations. In TensorFlow, matrices take the opposite roles, but the performance principles are the same.

Key Features	Accused Products			
	Table 1. Mapping of inputs, outputs, and batch size to GEMM parameters M, N, K.			
	Computation Phase	M	N	K
	Forward Propagation	Number of outputs	Batch size	Number of inputs
	Activation Gradient	Number of inputs	Batch size	Number of outputs
	Weight Gradient	Number of inputs	Number of outputs	Batch size
* * * * *				

Key Features	Accused Products
	<i>Figure 2. Dimensions of equivalent GEMMs for (a) forward propagation, (b) activation gradient, and (c) weight gradient computations of a fully-connected layer.</i>  Figure 2 consists of three diagrams labeled (a), (b), and (c), each showing the dimensions of matrices involved in GEMM operations for a fully-connected layer. (a) Forward propagation: Shows a matrix multiplication between $Weights^T$ (with dimensions $M = \# \text{ of outputs}$ and $K = \# \text{ of inputs}$) and $Input \text{ Activations}$ (with dimensions $K = \# \text{ of inputs}$ and $N = \text{batch size}$). The result is $Output \text{ Activations}$ (with dimensions $M = \# \text{ of outputs}$ and $N = \text{batch size}$). (b) Activation gradient: Shows a matrix multiplication between $Weights$ (with dimensions $M = \# \text{ of inputs}$ and $K = \# \text{ of outputs}$) and $Gradients (Output)$ (with dimensions $K = \# \text{ of outputs}$ and $N = \text{batch size}$). The result is $Gradients (Input)$ (with dimensions $M = \# \text{ of inputs}$ and $N = \text{batch size}$). (c) Weight gradient: Shows a matrix multiplication between $Input \text{ Activations}$ (with dimensions $M = \# \text{ of inputs}$ and $K = \text{batch size}$) and $Gradient^T (Output)$ (with dimensions $K = \text{batch size}$ and $N = \# \text{ of outputs}$). The result is $Gradient (Weights)$ (with dimensions $M = \# \text{ of inputs}$ and $N = \# \text{ of outputs}$). (See https://docs.nvidia.com/deeplearning/performance/dl-performance-fully-connected/index.html#performance (emphasis added).)
[7] The computer system of claim 1, wherein the controller is configured to transfer the output data from the accelerator	The Accused Products embody the computer system of claim 1, wherein the controller is configured to transfer the output data from the accelerator memory to the main memory without transferring any of the intermediate results from the accelerator memory to the main memory so as to reduce data transfer via the bus.