ANDREA WEISS JEFFRIES (SBN 183408)
(andrea.jeffries@wilmerhale.com)
WILMER CUTLER PICKERING
HALE AND DORR LLP
350 S. Grand Ave., Suite 2100
Los Angeles, CA 90071
Telephone: +213 443 5300
Facsimile: +213 443 5400

WILLIAM F. LEE (*pro hac vice* to be filed)
(william.lee@wilmerhale.com)
DOMINIC E. MASSA (*pro hac vice* to be filed)
(dominic.massa@wilmerhale.com)
WILMER CUTLER PICKERING
HALE AND DORR LLP
60 State Street
Boston, MA 02109
Telephone: +1 617 526 6000
Facsimile: +1 617 526 5000

Attorneys for Plaintiff
BROADCOM CORPORATION

FILED
2013 MAY 29 AM 11:02
CLERK U.S. DISTRICT COURT
CENTRAL DIST. OF CALIF.
SANTA ANA
BY

UNITED STATES DISTRICT COURT

CENTRAL DISTRICT OF CALIFORNIA

BROADCOM CORPORATION

Plaintiff,

vs.

NXP SEMICONDUCTORS N.V.,
NXP B.V., &
NXP SEMICONDUCTORS USA, INC.

Defendants.

SACV13-00829 CJC (MANx)
CASE NUMBER:

COMPLAINT FOR PATENT INFRINGEMENT

Demand for Jury Trial

Plaintiff Broadcom Corporation ("Broadcom"), for its Complaint against Defendants NXP Semiconductors N.V., NXP B.V., and NXP Semiconductors USA, Inc. (collectively, "NXP" or "Defendants"), alleges as follows:

JURISDICTION AND VENUE

1. This is an action for patent infringement arising under the patent laws of the United States, Title 35 U.S.C. §§ 1 et seq. This Court has subject matter jurisdiction over those claims pursuant to 28 U.S.C. §§ 1331 and 1338(a).

2. NXP is a global semiconductor company with operations in more than 25 countries, including the United States and California.

3. On information and belief, NXP directly or through its distributors and other intermediaries makes, uses, sells, imports into the United States, and/or distributes throughout the United States integrated circuit products in the RF, analog, power management, interface, security, and digital processing markets, including the products accused of infringement in this case.

4. NXP Semiconductors N.V. has publicly reported that it conducts substantial business in the United States.

5. Defendant NXP Semiconductors USA, Inc. is a subsidiary of Defendant NXP B.V., which is in turn a subsidiary of Defendant NXP Semiconductors N.V.

6. NXP Semiconductors USA, Inc. is registered to conduct business in California, and maintains business operations in California and within this District.

7. On information and belief, NXP Semiconductors USA, Inc. operates as the alter-ego of its parent companies, Defendants NXP Semiconductors N.V. and NXP B.V.

8. NXP has distributors and sales offices located in the United States and California, including for one or more of the products accused of infringement in this case.

9. Each of the Defendants has ongoing and systematic contacts within, and/or has otherwise directed activities toward, the State of California and this District, including but not limited to: (i) by conducting at least a portion of the infringement alleged herein in the state of California and within this District, and (ii) by regularly conducting or soliciting business, engaging in other persistent courses of conduct, and/or deriving substantial revenue from goods and services provided to the residents of California and this District.

10. On information and belief, Defendants, directly or through distributors and other intermediaries, use, market, import, distribute, offer for sale, and/or sell products in the United States, the State of California, and this District, including one or more of the products accused of infringement in this case.

11. On information and belief, Defendants have purposefully and voluntarily placed one or more infringing products into the stream of commerce with the expectation that they will be incorporated into downstream products and purchased by consumers in this District, including by directly or indirectly working

with subsidiaries, distributors, and other entities located in California to ensure that its products reach California and this District.

12. Defendants have availed themselves of the benefits and privileges of conducting business in the State of California, and the exercise of jurisdiction over Defendants would not offend traditional notions of fair play and substantial justice.

13. Venue is proper in this District pursuant to 28 U.S.C. §§ 1391 and 1400(b). On information and belief, NXP has committed, and is continuing to commit, acts of infringement in this District, NXP maintains regular business operations in this District, and NXP is subject to personal jurisdiction in this District.

THE PARTIES

14. Plaintiff Broadcom is a corporation organized and existing under the laws of the state of California, with its principal place of business located at 5300 California Avenue, Irvine, California 92617.

15. On information and belief, Defendants NXP Semiconductors N.V. and NXP B.V. are corporations organized and existing under the laws of The Netherlands, with their principal place of business located at High Tech Campus 60, 5656 AG, Eindhoven, The Netherlands.

16. On information and belief, Defendant NXP Semiconductors USA, Inc. is a corporation organized and existing under the laws of Delaware with its

principal place of business located at 411 East Plumeria Drive, San Jose, CA 95134.

FACTUAL ALLEGATIONS

17. This action arises out of NXP's infringement of five patents owned by Broadcom.

18. On April 8, 2008, United States Patent No. 7,355,485 (the "'485 Patent"), entitled "Apparatus and Method for Phase Lock Loop Gain Control Using Unit Current Sources," was duly and legally issued by the United States Patent and Trademark Office. The '485 Patent has been duly and legally assigned to Broadcom. A true and correct copy of the '485 Patent is attached hereto as Exhibit A.

19. On July 15, 2008, United States Patent No. 7,400,722 (the "'722 Patent"), entitled "Methods and Apparatus for Performing Hash Operations in a Cryptography Accelerator," was duly and legally issued by the United States Patent and Trademark Office. The '722 Patent has been duly and legally assigned to Broadcom. A true and correct copy of the '722 Patent is attached hereto as Exhibit B.

20. On February 2, 2010, United States Patent No. 7,656,643 (the "'643 Patent"), entitled "Scalable Integrated Circuit High Density Capacitors," was duly and legally issued by the United States Patent and Trademark Office. The '643

Patent has been duly and legally assigned to Broadcom. A true and correct copy of the '643 Patent is attached hereto as Exhibit C.

21. On November 20, 2012, United States Patent No. 8,315,381 (the "'381 Patent"), entitled "Methods and Apparatus Performing Hash Operations in a Cryptography Accelerator," was duly and legally issued by the United States Patent and Trademark Office. The '381 Patent has been duly and legally assigned to Broadcom. A true and correct copy of the '381 Patent is attached hereto as Exhibit D.

22. On December 18, 2012, United States Patent No. 8,333,317 (the "'317 Patent"), entitled "System and Method for Authenticating the Proximity of a Wireless Token to a Computing Device," was duly and legally issued by the United States Patent and Trademark Office. The '317 Patent has been duly and legally assigned to Broadcom. A true and correct copy of the '317 Patent is attached hereto as Exhibit E.

23. Broadcom is the owner of the '485,'722, '643, '381, and '317 Patents ("Broadcom Patents in Suit") with the full and exclusive right to bring suit to enforce the Broadcom Patents in Suit.

24. The Broadcom Patents in Suit are valid and enforceable.

25. NXP has knowledge of the Broadcom Patents in Suit at least by virtue of the filing of this Complaint.

26. NXP designs, makes, uses, sells, offers for sale, and/or supports semiconductor integrated circuit products, including but not limited to the PN544 NFC Controller IC, the PN65 NFC SmartConnect Module IC, and the P5DF081 MIFARE SAM AV2 integrated circuit products, and other integrated circuit products that incorporate similar circuitry or have similar features or functionality.

27. NXP's semiconductor integrated circuit products are integrated into devices made, used, and sold by original equipment manufacturers ("OEMs"), distributors, and/or other third parties.

28. As alleged herein, NXP has engaged in actions that have directly and indirectly infringed the Broadcom Patents in Suit.

29. For one or more of the Broadcom Patents in Suit, NXP has directly infringed by making, using, selling, offering to sell, and/or importing infringing products into the United States.

30. For one or more of the Broadcom Patents in Suit, NXP has contributorily infringed by offering to sell or selling within the United States or importing into the United States a component of a patented machine, manufacture, combination, or composition, or a material or apparatus for use in practicing a patented process, constituting a material part of the claimed invention(s), knowing the same to be especially made or especially adapted for use in an infringement of such patent, and not a staple article or commodity of commerce suitable for substantial noninfringing use.

31. For one or more of the Broadcom Patents in Suit, NXP is actively inducing infringement by encouraging acts of direct infringement, when NXP knows (or believes there is a high probability, but is taking deliberate steps to avoid knowing) that it is inducing direct infringement. NXP's acts of encouragement include but are not limited to (a) selling to, supplying to, encouraging, helping design, and/or supporting OEMs to incorporate certain NXP products into devices sold by OEMs and/or other third parties, and (b) encouraging and/or instructing third parties how to use certain NXP products, with the intent that these products be used by such third party entities to infringe the Broadcom Patents in Suit.

COUNT ONE
Infringement of the '485 Patent

32. Plaintiff repeats and re-alleges the allegations of paragraphs 1–31 above as if fully set forth herein.

33. NXP has infringed and continues to infringe, literally and/or under the doctrine of equivalents, one or more claims of the '485 Patent, including but not limited to claim 1, directly and/or indirectly, in violation of 35 U.S.C. § 271(a), (b), (c) and/or (f), by making, using, selling, offering to sell and/or importing integrated circuit products in the United States (and/or by contributing to and/or inducing these acts) including but not limited to the PN544 NFC Controller IC product and

other integrated circuit products that incorporate similar circuitry and/or have similar features or functionality.

34. NXP has knowledge of the '485 Patent at least by virtue of the filing of this complaint.

35. On information and belief, NXP's actions as alleged herein have actively induced and/or are continuing to actively induce infringement of the '485 Patent by encouraging acts of direct infringement, and NXP knows (or believes there is a high probability, but is taking deliberate steps to avoid knowing) that it is inducing direct infringement by (a) selling to, supplying to, encouraging, helping design, supporting, and/or instructing OEMs, distributors and/or other third party entities, to incorporate NXP products into devices sold by such entities, and by (b) encouraging and/or instructing third party entities how to use such NXP products, with the intent that these products be used by such third party entities to infringe the '485 Patent.

36. For example, on information and belief, NXP offers, sells and markets (and after the filing of this complaint is continuing to offer, sell and market) its products, including the PN544 NFC Controller IC product, through its web site and through various channels including U.S. distributors and/or other third parties.

37. On information and belief, the PN544 NFC Controller IC product when used (including when used by NXP) performs all the steps of the "method of

compensating the gain of a phase lock loop (PLL)" described in the claims of the '485 Patent.

38. NXP's product literature for the PN544 NFC Controller IC product, including data sheets, instructs and encourages NXP's customers and other third parties to use the PN544, including in the United States, and thereby perform the method described in the claims of the '485 Patent. NXP is therefore aware that NXP's customers and other third parties are infringing the '485 Patent by using the PN544, as instructed and encouraged by NXP.

39. On information and belief, NXP's actions as alleged herein have contributed and are continuing to contribute to infringement of the '485 Patent by offering to sell or selling within the United States or importing into the United States a component of a patented machine, manufacture, combination, or composition, or a material or apparatus for use in practicing a patented process, constituting a material part of the claimed invention(s) of the '485 Patent, knowing the same to be especially made or especially adapted for use in an infringement of such patent, and not a staple article or commodity of commerce suitable for substantial non-infringing use.

40. For example, as set forth above, on information and belief, NXP's PN544 NFC Controller IC product is designed, manufactured, and sold, so as to perform the method described in the claims of the '485 Patent when used. On

information and belief, the PN544 NFC Controller IC product therefore has no substantial non-infringing uses.

COUNT TWO
Infringement of the '722 Patent

41. Plaintiff repeats and re-alleges the allegations of paragraphs 1–40 above as if fully set forth herein.

42. NXP has infringed and continues to infringe, literally and/or under the doctrine of equivalents, one or more claims of the '722 Patent, including but not limited to claim 1, directly and/or indirectly, in violation of 35 U.S.C. § 271(a), (b), (c) and/or (f), by making, using, selling, offering to sell and/or importing integrated circuit products in the United States (and/or by contributing to and/or inducing these acts) including but not limited to the P5DF081 MIFARE SAM AV2 integrated circuit product and other integrated circuit products that incorporate similar circuitry and/or have similar features or functionality.

43. NXP has knowledge of the '722 Patent at least by virtue of the filing of this complaint.

44. On information and belief, NXP's actions as alleged herein have actively induced and/or are continuing to actively induce infringement of the '722 Patent by encouraging acts of direct infringement, and NXP knows (or believes there is a high probability, but is taking deliberate steps to avoid knowing) that it is inducing direct infringement by (a) selling to, supplying to, encouraging, helping

design, supporting, and/or instructing OEMs, distributors, and/or other third party entities, to incorporate NXP products into devices sold by such entities, and by (b) encouraging and/or instructing third party entities how to use such NXP products, with the intent that these products be used by such third party entities to infringe the '722 Patent.

45. For example, on information and belief, NXP offers, sells and markets (and after the filing of this complaint is continuing to offer, sell and market) products such as the P5DF081, through its website and through various channels including U.S. distributors and/or other third parties. The P5DF081 IC product, as designed, manufactured, and sold, comprises a first hash core, a persistent memory, and a temporary memory, all of which are configured as described in the claims of the '722 Patent.

46. NXP's product literature for the P5DF081 IC product, including data sheets, instructs and encourages NXP's customers and other third parties to use the P5DF081, and thereby infringe the '722 Patent. NXP is therefore aware that NXP's customers and other third parties are infringing the '722 Patent by using the P5DF081 IC product, as instructed and encouraged by NXP.

47. On information and belief, NXP's actions as alleged herein have contributed and are continuing to contribute to infringement of the '722 Patent by offering to sell or selling within the United States or importing into the United States a component of a patented machine, manufacture, combination, or

composition, or a material or apparatus for use in practicing a patented process, constituting a material part of the claimed invention(s) of the '722 Patent, knowing the same to be especially made or especially adapted for use in an infringement of such patent, and not a staple article or commodity of commerce suitable for substantial non-infringing use.

48. For example, as set forth above, NXP's P5DF081 IC product, as designed, manufactured, and sold, comprises a first hash core, a persistent memory, and a temporary memory, all of which are configured as described in the claims of the '722 Patent. On information and belief, such systems have no substantial non-infringing uses.

COUNT THREE
Infringement of the '643 Patent

49. Plaintiff repeats and re-alleges the allegations of paragraphs 1–48 above as if fully set forth herein.

50. NXP has infringed and continues to infringe, literally and/or under the doctrine of equivalents, one or more claims of the '643 Patent, including but not limited to claim 1, directly and/or indirectly, in violation of 35 U.S.C. § 271(a), (b), (c) and/or (f), by making, using, selling, offering to sell and/or importing integrated circuit products in the United States (and/or by contributing to and/or inducing these acts) including but not limited to the PN544 NFC Controller IC product and

other integrated circuit products that incorporate similar circuitry and/or have similar features or functionality.

51. NXP has knowledge of the '643 Patent at least by virtue of the filing of this complaint.

52. On information and belief, NXP's actions as alleged herein have actively induced and/or are continuing to actively induce infringement of the '643 Patent by encouraging acts of direct infringement, and NXP knows (or believes there is a high probability, but is taking deliberate steps to avoid knowing) that it is inducing direct infringement by (a) selling to, supplying to, encouraging, helping design, supporting, and/or instructing OEMs, distributors, and/or other third party entities, to incorporate NXP products into devices sold by such entities, and by (b) encouraging and/or instructing third party entities how to use such NXP products, with the intent that these products be used by such third party entities to infringe the '643 Patent.

53. For example, on information and belief, NXP offers, sells and markets (and after the filing of this complaint is continuing to offer, sell and market) its products, including the PN544 NFC Controller IC product, through its web site and through various channels including U.S. distributors and/or other third parties.

54. On information and belief, the PN544 NFC Controller IC product contains an integrated circuit capacitor comprising a first metal layer and second

metal layer, and a first via layer, as these elements are described in the claims of the '643 Patent.

55. NXP's product literature for the PN544 NFC Controller IC product, including data sheets, instructs and encourages NXP's customers and other third parties to use the PN544, including in the United States, and thereby infringe claims of the '643 Patent. NXP is therefore aware that NXP's customers and other third parties are infringing the '643 Patent by using the PN544, as instructed and encouraged by NXP.

56. On information and belief, NXP's actions as alleged herein have contributed and are continuing to contribute to infringement of the '643 Patent by offering to sell or selling within the United States or importing into the United States a component of a patented machine, manufacture, combination, or composition, or a material or apparatus for use in practicing a patented process, constituting a material part of the claimed invention(s) of the '643 Patent, knowing the same to be especially made or especially adapted for use in an infringement of such patent, and not a staple article or commodity of commerce suitable for substantial non-infringing use.

57. For example, as set forth above, on information and belief, NXP's PN544 NFC Controller IC product is designed, manufactured, and sold so as to contain an integrated circuit capacitor, as claimed by the '643 Patent. On

information and belief, the PN544 NFC Controller IC product therefore has no substantial non-infringing uses.

COUNT FOUR
Infringement of the '381 Patent

58. Plaintiff repeats and re-alleges the allegations of paragraphs 1–57 above as if fully set forth herein.

59. NXP has infringed and continues to infringe, literally and/or under the doctrine of equivalents, one or more claims of the '381 Patent, including but not limited to claim 1, directly and/or indirectly, in violation of 35 U.S.C. § 271(a), (b), (c) and/or (f), by making, using, selling, offering to sell and/or importing integrated circuit products in the United States (and/or by contributing to and/or inducing these acts) including but not limited to the P5DF081 MIFARE SAM AV2 integrated circuit product and other integrated circuit products that incorporate similar circuitry and/or have similar features or functionality.

60. NXP has knowledge of the '381 Patent at least by virtue of the filing of this complaint.

61. On information and belief, NXP's actions as alleged herein have actively induced and/or are continuing to actively induce infringement of the '381 Patent by encouraging acts of direct infringement, and NXP knows (or believes there is a high probability, but is taking deliberate steps to avoid knowing) that it is inducing direct infringement by (a) selling to, supplying to, encouraging, helping

design, supporting, and/or instructing OEMs, distributors, and/or other third party entities, to incorporate NXP products into devices sold by such entities, and by (b) encouraging and/or instructing third party entities how to use such NXP products, with the intent that these products be used by such third party entities to infringe the '381 Patent.

62. For example, on information and belief, NXP offers, sells and markets (and after the filing of this complaint is continuing to offer, sell and market) products such as the P5DF081, through its website and through various channels including U.S. distributors and/or other third parties. The P5DF081 IC product, as designed, manufactured, and sold, comprises a cryptographic accelerator, a processing unit, and a memory, all of which are configured as described in the claims of the '381 Patent.

63. NXP's product literature for the P5DF081 IC product, including data sheets, instructs and encourages NXP's customers and other third parties to use the P5DF081, and thereby infringe the '381 Patent. NXP is therefore aware that NXP's customers and other third parties are infringing the '381 Patent by using the P5DF081 IC product, as instructed and encouraged by NXP.

64. On information and belief, NXP's actions as alleged herein have contributed and are continuing to contribute to infringement of the '381 Patent by offering to sell or selling within the United States or importing into the United States a component of a patented machine, manufacture, combination, or

composition, or a material or apparatus for use in practicing a patented process, constituting a material part of the claimed invention(s) of the '381 Patent, knowing the same to be especially made or especially adapted for use in an infringement of such patent, and not a staple article or commodity of commerce suitable for substantial non-infringing use.

65. For example, as set forth above, NXP's P5DF081 IC product, as designed, manufactured, and sold, comprises a cryptographic accelerator, a processing unit, and a memory, all of which are configured as described in the claims of the '381 Patent. On information and belief, such systems have no substantial non-infringing uses.

COUNT FIVE
Infringement of the '317 Patent

66. Plaintiff repeats and re-alleges the allegations of paragraphs 1–65 above as if fully set forth herein.

67. NXP has infringed and continues to infringe, literally and/or under the doctrine of equivalents, one or more claims of the '317 Patent, including but not limited to claim 1, directly and/or indirectly, in violation of 35 U.S.C. § 271(a), (b), (c) and/or (f), by making, using, selling, offering to sell and/or importing integrated circuit products in the United States (and/or by contributing to and/or inducing these acts) including but not limited to the PN65 NFC SmartConnect Module IC

product and other integrated circuit products that incorporate similar circuitry and/or have similar features or functionality.

68. NXP has knowledge of the '317 Patent at least by virtue of the filing of this complaint.

69. On information and belief, NXP's actions as alleged herein have actively induced and/or are continuing to actively induce infringement of the '317 Patent by encouraging acts of direct infringement, and NXP knows (or believes there is a high probability, but is taking deliberate steps to avoid knowing) that it is inducing direct infringement by (a) selling to, supplying to, encouraging, helping design, supporting, and/or instructing OEMs, distributors, and/or other third party entities, to incorporate NXP products into devices sold by such entities, and by (b) encouraging and/or instructing third party entities how to use such NXP products, with the intent that these products be used by such third party entities to infringe the '317 Patent.

70. For example, on information and belief, NXP offers, sells and markets (and after the filing of this complaint is continuing to offer, sell and market) NFC SmartConnect Module IC products such as the PN65, through its website and through various channels including U.S. distributors and/or other third parties. The PN65 IC product, as designed, manufactured, and sold, comprises a proximity reader, a secure processor, an external interface, and a security boundary, all of which are configured as described in the claims of the '317 Patent.

71. NXP's product literature for the PN65 IC product, including data sheets, instructs and encourages NXP's customers and other third parties to use the PN65, and thereby infringe the '317 Patent. NXP is therefore aware that NXP's customers and other third parties are infringing the '317 Patent by using the PN65 IC product, as instructed and encouraged by NXP.

72. On information and belief, NXP's actions as alleged herein have contributed and are continuing to contribute to infringement of the '317 Patent by offering to sell or selling within the United States or importing into the United States a component of a patented machine, manufacture, combination, or composition, or a material or apparatus for use in practicing a patented process, constituting a material part of the claimed invention(s) of the '317 Patent, knowing the same to be especially made or especially adapted for use in an infringement of such patent, and not a staple article or commodity of commerce suitable for substantial non-infringing use.

73. For example, as set forth above, NXP's PN65 IC product, as designed, manufactured, and sold, comprises a proximity reader, a secure processor, an external interface, and a security boundary, all of which are configured as described in the claims of the '317 Patent. On information and belief, such systems have no substantial non-infringing uses.

DAMAGES

74. By reason of the acts alleged herein, Broadcom has suffered, is suffering, and unless restrained by the Court, will continue to suffer injury to its business and property rights, for which it is entitled to damages pursuant to 35 U.S.C. §§ 284–285 in an amount to be proven at trial.

INJUNCTION

75. By reason of the acts alleged herein, Broadcom has suffered, is suffering, and unless restrained by the Court, will continue to suffer irreparable harm for which there is no adequate remedy at law, and for which Broadcom is entitled to permanent injunctive relief pursuant to 35 U.S.C. § 283.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff Broadcom requests that the Court enter judgment in its favor and against NFC as follows:

a. Declare that NXP has infringed one or more claims of the Broadcom Patents in Suit;

b. Enter a permanent injunction prohibiting NXP, its subsidiaries, divisions, agents, servants, employees, and all those acting in concert with and/or who are in privity with NXP and/or any of the foregoing from infringing, contributing to the infringement of, and/or inducing infringement of one or more of the Broadcom Patents in Suit, and for all further proper injunctive relief;

c. Award to Plaintiff damages for NXP's infringement, with pre-judgment and post-judgment interest pursuant to 35 U.S.C. §§ 284 and/or 285; and

d. Grant any and all such further relief as the Court deems just and proper.

DEMAND FOR JURY TRIAL

In accordance with Fed. R. Civ. P. 38(b) and Local Rule 38–1, Plaintiff Broadcom demands a trial by jury on all issues so triable.

Dated: May 29, 2013

WILMER CUTLER PICKERING HALE AND DORR LLP

By: _____
Andrea Weiss Jeffries (SBN 183408)

Attorneys for Plaintiff
BROADCOM CORPORATION

Exhibit A



(12) **United States Patent**
Gomez

(10) **Patent No.: US 7,355,485 B2**
(45) **Date of Patent: *Apr. 8, 2008**

(54) **APPARATUS AND METHOD FOR PHASE LOCK LOOP GAIN CONTROL USING UNIT CURRENT SOURCES**

(75) Inventor: **Ramon A. Gomez**, San Juan Capistrano, CA (US)

(73) Assignee: **Broadcom Corporation**, Irvine, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/095,117**

(22) Filed: **Mar. 31, 2005**

(65) **Prior Publication Data**

US 2005/0168297 A1 Aug. 4, 2005

Related U.S. Application Data

(60) Division of application No. 11/019,451, filed on Dec. 23, 2004, which is a continuation of application No. 10/443,741, filed on May 23, 2003, now Pat. No. 6,838,947, which is a continuation of application No. 09/811,611, filed on Mar. 20, 2001, now Pat. No. 6,583,675.

(51) **Int. Cl.**
H03L 7/00 (2006.01)

(52) **U.S. Cl.** **331/15**; 331/16; 331/17

(58) **Field of Classification Search** ..... 331/17, 331/16, 183, 36 C, 177 V, 2, 7, 8, 15, 1 A; 327/157; 375/374

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,526,859 A 9/1970 Pützer
3,538,450 A 11/1970 Andrea et al.
4,568,888 A 2/1986 Kimura et al.
4,970,472 A 11/1990 Kennedy et al.

(Continued)

FOREIGN PATENT DOCUMENTS

EP 0627820 12/1994

(Continued)

OTHER PUBLICATIONS

Abidi et al., "Power-Conscious Design Of Wireless Circuits And Systems", IEEE Proceedings, Oct. 2000, vol. 88, No. 10, pp. 1528-1545.

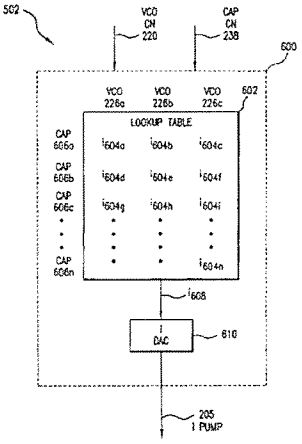
(Continued)

Primary Examiner—Joseph Chang
(74) *Attorney, Agent, or Firm*—McAndrews, Held & Malloy, Ltd.

(57) **ABSTRACT**

A gain compensator compensates for the gain variation of a varactor-tuned voltage tuned oscillator (VCO) in a phase lock loop (PLL). The VCO includes a parallel LC circuit having multiple fixed capacitors that can be switched-in or switched-out of the LC circuit according to a capacitor control signal to perform band-select tuning of the VCO. The gain compensator compensates for the variable VCO gain by generating a charge pump reference current that is based on the same capacitor control signal that controls the fixed capacitors in the LC circuit. The gain compensator generates the charge pump reference current by replicating a reference scale current using unit current sources. The number of times the reference scale current is replicated is based on the fixed capacitance that is switched-in to the LC circuit and therefore the frequency band of the PLL. The reference scale current is generated based on a PLL control that specifics certain PLL characteristics such as reference frequency, loop bandwidth, and loop damping. Therefore, the reference pump current can be efficiently optimized for changing PLL operating conditions, in addition to compensating for variable VCO gain.

20 Claims, 13 Drawing Sheets



U.S. PATENT DOCUMENTS

5,030,926	A		7/1991	Walden
5,121,085	A	*	6/1992	Brown 331/8
5,126,692	A		6/1992	Shearer et al.
5,254,958	A		10/1993	Flach et al.
5,315,270	A	*	5/1994	Leonowich ..... 331/1 A
5,362,990	A		11/1994	Alvarez et al.
5,369,376	A		11/1994	Leblebicioglu
5,485,125	A		1/1996	Dufour
5,563,553	A		10/1996	Jackson
5,625,325	A		4/1997	Rotzoll et al.
5,648,744	A		7/1997	Prakash et al.
5,739,730	A		4/1998	Rotzoll
5,783,972	A		7/1998	Nishikawa
5,821,818	A		10/1998	Idei et al.
6,028,488	A		2/2000	Landman et al.
6,091,304	A		7/2000	Harrer
6,133,797	A		10/2000	Lovelace et al.
6,163,184	A		12/2000	Larsson
6,680,653	B2	*	1/2004	Griffith et al. 331/17
6,891,414	B1	*	5/2005	Bunch et al. 327/156
2004/0012425	A1	*	1/2004	Hsu 327/157
2005/0024152	A1	*	2/2005	Lyden et al. 331/16

FOREIGN PATENT DOCUMENTS

EP	0642227	3/1995
EP	1075086	2/2001

OTHER PUBLICATIONS

Alvarez et al., "A Wide-Bandwidth Low-Voltage PLL For Power PCä Microprocessors", 1994 Symposium On VLSI Circuits Digest of Technical Papers, Apr. 1995, vol. 30, No. 4, pp. 384-391.

Alvarez et al., "A Wide-Bandwidth Low-Voltage PLL For Power PCä Microprocessors", 1994 Symposium On VLSI Circuits Digest of Technical Papers, Jun. 9-11, 1994, Sec. 4.2, pp. 37-38.

Bayer et al., "A Low Noise CMOS Frequency Synthesizer With Dynamic Bandwidth Control", IEEE 1994 Custom Integrated Circuits Conference, May 1-4, 1994, Sec. 8.5.1, pp. 171-174.

Boerstler, "A Low-Jitter PLL Clock Generator For Microprocessors With Lock Range Of 340-612 MHz", IEEE J. Of Solid State Circuits, Apr. 1999, vol. 34, No. 4, pp. 513-519.

Craninckx et al., "A Fully Integrated CMOS DCS-1800 Frequency Synthesizer", IEEE International Solid State Circuits Conference '98 Proceedings, Feb. 17. 1998, pp. 372-373, 466.

Craninckx et al., "A Fully Integrated CMOS DCS-1800 Frequency Synthesizer", IEEE J. of Solid State Circuits, Dec. 1998, vol. 33, No. 12, pp. 2054-2065.

Diorio et al., "A Low-Noise GaAs/AlGaAs, Microwave Frequency-Synthesizer IC", IEEE J. of Solid State Circuits, Sep. 1998, vol. 33, No. 9, pp. 1306-1312.

Gardner, "Charge-Pump Phase-Lock Loops," IEEE Trans. On Comms., Nov. 1980, vol. COM-28, No. 11, pp. 1849-1858.

Kelkar et al., "A Wide-Range Low-Hitter Fully Integrated programmable Frequency Synthesizer Building block in A 1.5V/0.25 um DMOS Process", IEEE Int'l. ASIC/SOC Conference Proceedings, Sep. 15, 1999-Sep. 18, 1999, pp. 357-361.

Lam et al., "A 2.6-GHz/5.2GHz Frequency Synthesizer in 0.4 um CMOS Technology", IEEE J. Of Solid State Circuits, May 2000, pp. 551-557.

Larsson, "A 2-1600-MHz CMOS Clock Recovery PLL With Low-Vdd Capability", IEEE J. Of Solid-State Circuits, Dec. 1999, vol. 34, No. 12, pp. 1951-1960.

Lin et al., "A 900MHz, 2.5mA CMOS Frequency Synthesizer With An Automatic SC Tuning Loop", Proceedings Of Custom Integrated Circuits Conference; May 2000, pp. 375-378.

Lo et al., "A 1.5-V 900-MHz Monolithic CMOS Fast-Switching Frequency Synthesizer For Wireless Applications", Symposium On VLSI Circuits Digest of Technical Papers, Jun. 2000, pp. 238-241.

Razavi, "Monolithic Phase-Locked Loops And Clock Recovery Circuits, Theory and Design", IEEE, 1996, pp. 1-498.

Roh et al, "Optimum Phase-Acquisition Technique For Charge-Pump PLL", IEEE Transactions On Circuits And Systems-II: Analog and Digital Signal Processing, Sep. 1997, vol. 44, No. 9, pp. 729-740.

Rhode, "Digital PLL Frequency Synthesizers Theory and Design", Prentice Hall, 1993.

Sun et al., "A Fully Integrated Dual-Frequency Push-Push VCO For Wideband Wireless Applications," Proc. Of Gallium Arsenide Applications Symposium, Oct. 2-6, 2000.

Sun, "An Analog PLL-Based Clock And Data Recovery Circuit With High Input Jitter Tolerance", IEEE J. Of Solid-State Circuits, Apr. 1989, vol. 24, No. 2, pp. 325-330.

Tournier, "A New Tuning And Display Monochip For Radio Bands Emitters And Receivers", IEEE, 1988.

Wolaver, "Phase-Locked Loop Circuit Design", Prentice Hall, 1991.

Yang, "A Low Jitter 0.3-165 MHz CMOS PLL Frequency Synthesizer For 3 V/5 V Operation", IEEE J. Of Solid State Circuits, Apr. 1997, vol. 32, No. 4, pp. 582-586.

"DS8908B AM/FM Digital Phase-Locked Loop Frequency Synthesizer", National Semiconductor, Jun. 1990, pp. 1-10.

Byrd et al., "A Fast Locking Scheme for PLL Frequency Synthesizers", National Semiconductor App. Note 1000, Jul. 1995.

Notice Regarding Issuance Of Initial Determination And Recommended Determination On Remedy And Bond, Oct. 10, 2006.

Kishine et al., "A 2.5 Gb/s Clock And Data Recovery IC With Tunable Jitter Characteristics For Use In LAN's And WAN's", IEEE J. of Solid-State Circuits, Jun. 1999, vol. 34, No. 6, pp. 805-812.

Kral et al., "RF-CMOS Oscillators With Switched Tuning", Custom IC Conference, Santa Clara, CA, May 1998, pp. 555-558.

Kral, "A 2.4 GHz CMOS Frequency Synthesizer", Final Report, UCLA Integrated Circuits And Systems Laboratory, Mar. 1998.

Best, "Phase-Locked Loops, Design, Simulations And Applications" (McGraw-Hill), 1997.

Gray et al., "Analysis And Design Of Analog Integrated Circuits", Wiley 2nd Ed., 1984.

Posthearing Brief Of The Commission Investigative Staff, United States International Trade Commission, Apr. 3, 2006.

Craninckx et al., Wireless CMOS Frequency Synthesizer Design, 1998.

Invalidity Expert Report Of German Gutierrez, Dec. 5, 2006.

Rebuttal Expert Report Of Linda Milor, Ph.D., Regarding Validity Of U.S. Patent No. 6,583,675, Dec. 14, 2005.

Crawford, J.A., "Frequency Synthesizer Design Handbook," Artech House, pp. 248-254 and 292-304 (1994).

Rohde, U.L., "Digital PLL Frequency Synthesizers," Prentice-Hall, pp. 55-58 (1983).

European Search Report issued Jun. 6, 2002, for EP 02252014, 3 pp.

An original executed Declaration of Ramon A. Gomez, Dec. 6, 1999.

Press Release, "Broadcom Delivers World's First CMOS TV Tuner," 5 pp. (Dec. 6, 1999).

Respondent Qualcomm Incorporated Notice of Prior Art, with Exhibit C attached thereto, Nov. 15, 2005.

Qualcomm's Supplemental Responses To Interrogatories 3-6, pp. 2-28, Nov. 11, 2005.

* cited by examiner

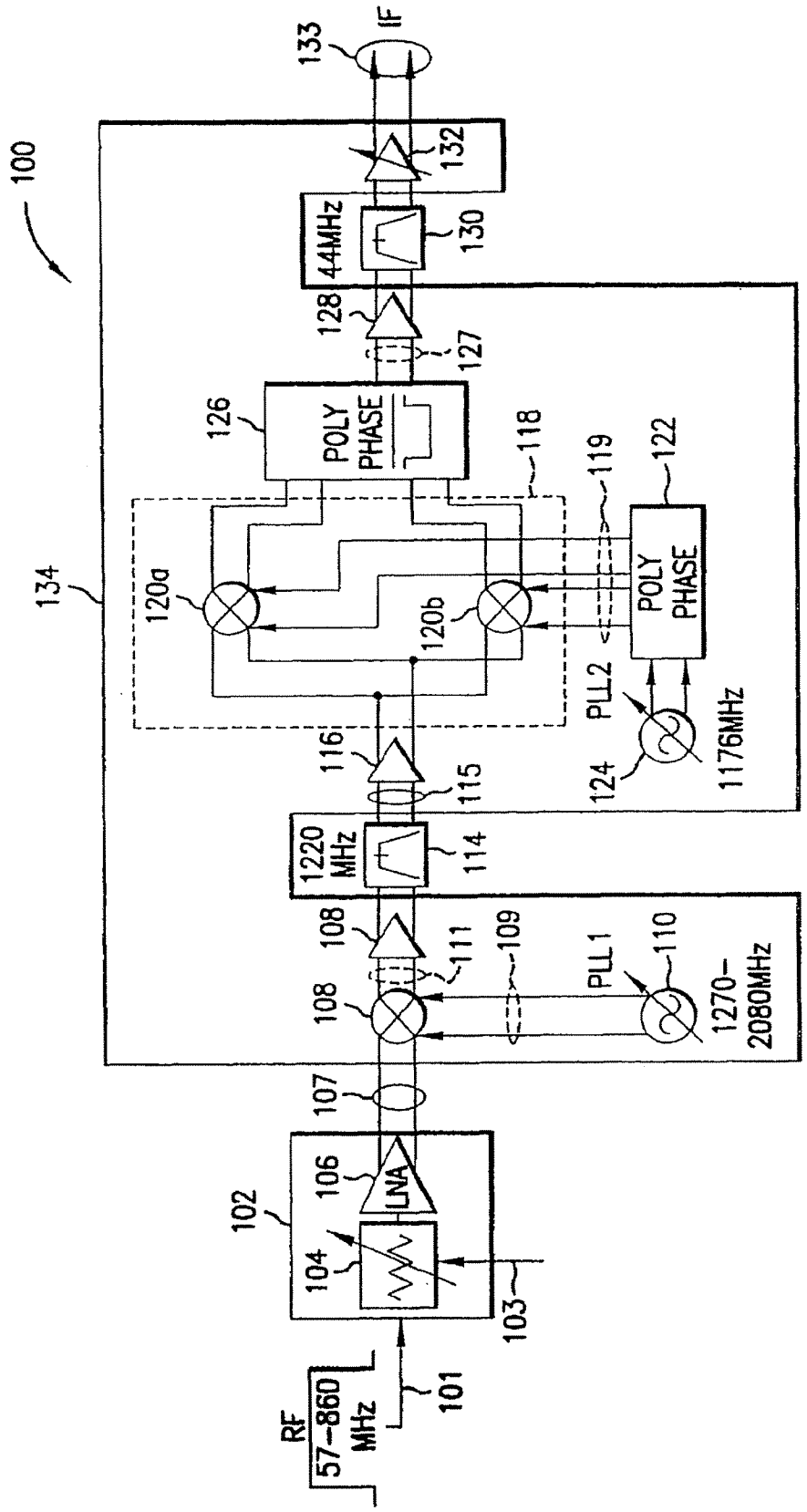

100
133
IF
132
44MHz
130
128
127
126
POLY PHASE
118
119
122
134
120a
120b
POLY PHASE
PLL2
116
115
124
1176MHz
1220 MHz
114
108
111
109
PLL1
110
108
1270-2080MHz
107
106
LNA
102
104
103
RF 57-860 MHz
101

FIG. 1A

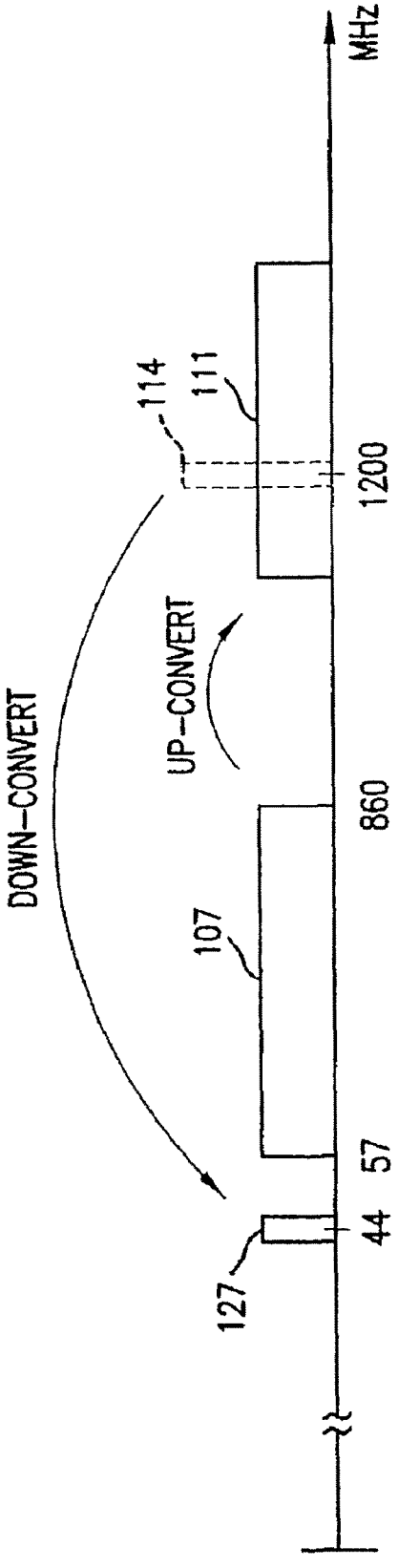


FIG. 1B

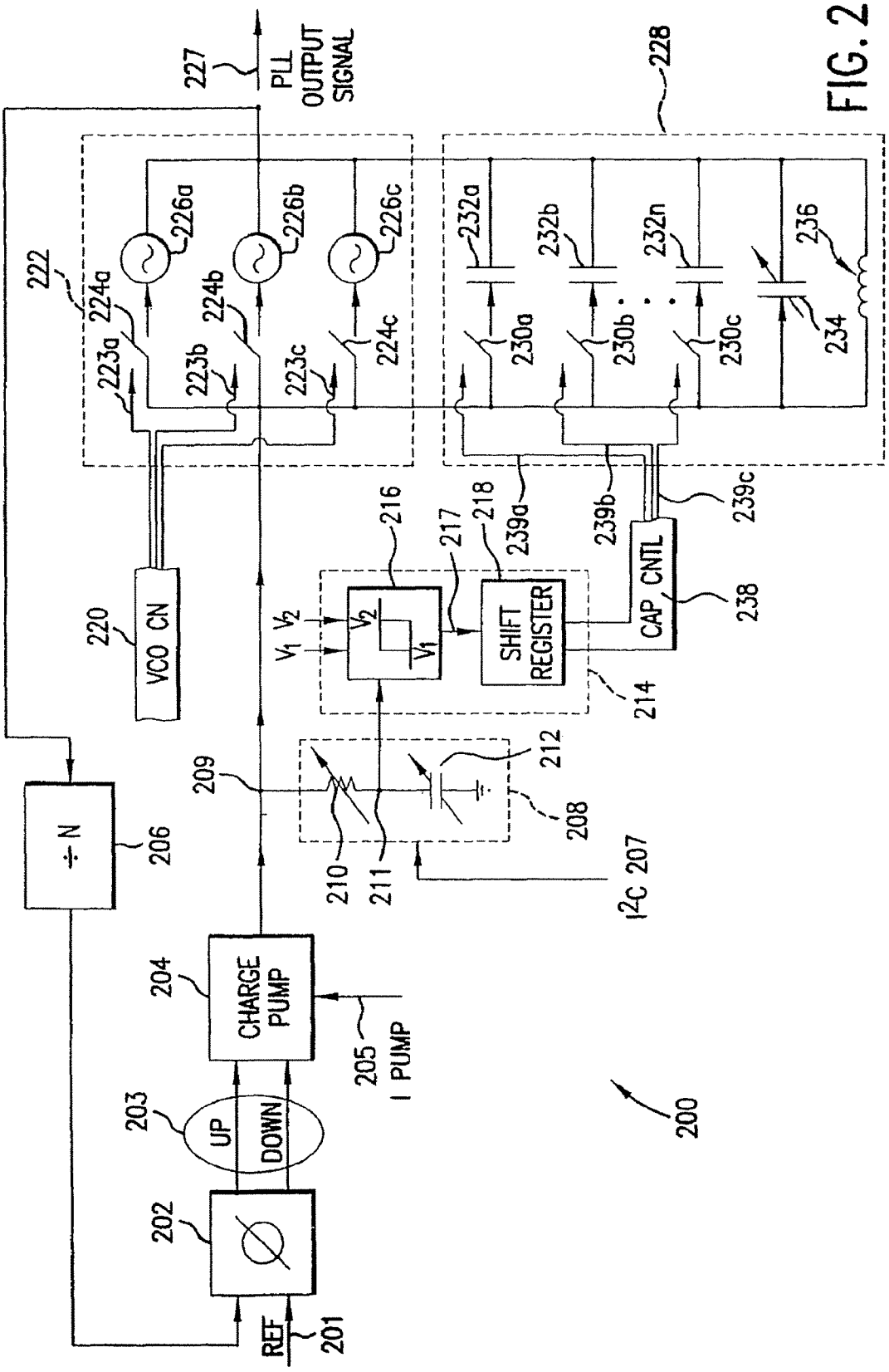

227
PLL OUTPUT SIGNAL
228
222
226a
226b
226c
224a
224b
224c
223a
223b
223c
232a
232b
232n
236
230a
230b
230c
234
216
217
218
239a
239b
239c
220
VCO CN
V1 V2
SHIFT REGISTER
CAP CNTL
238
214
209
212
208
210
211
I2C 207
÷ N
206
204
CHARGE PUMP
205
I PUMP
203
UP
DOWN
202
REF
201
200

FIG. 2

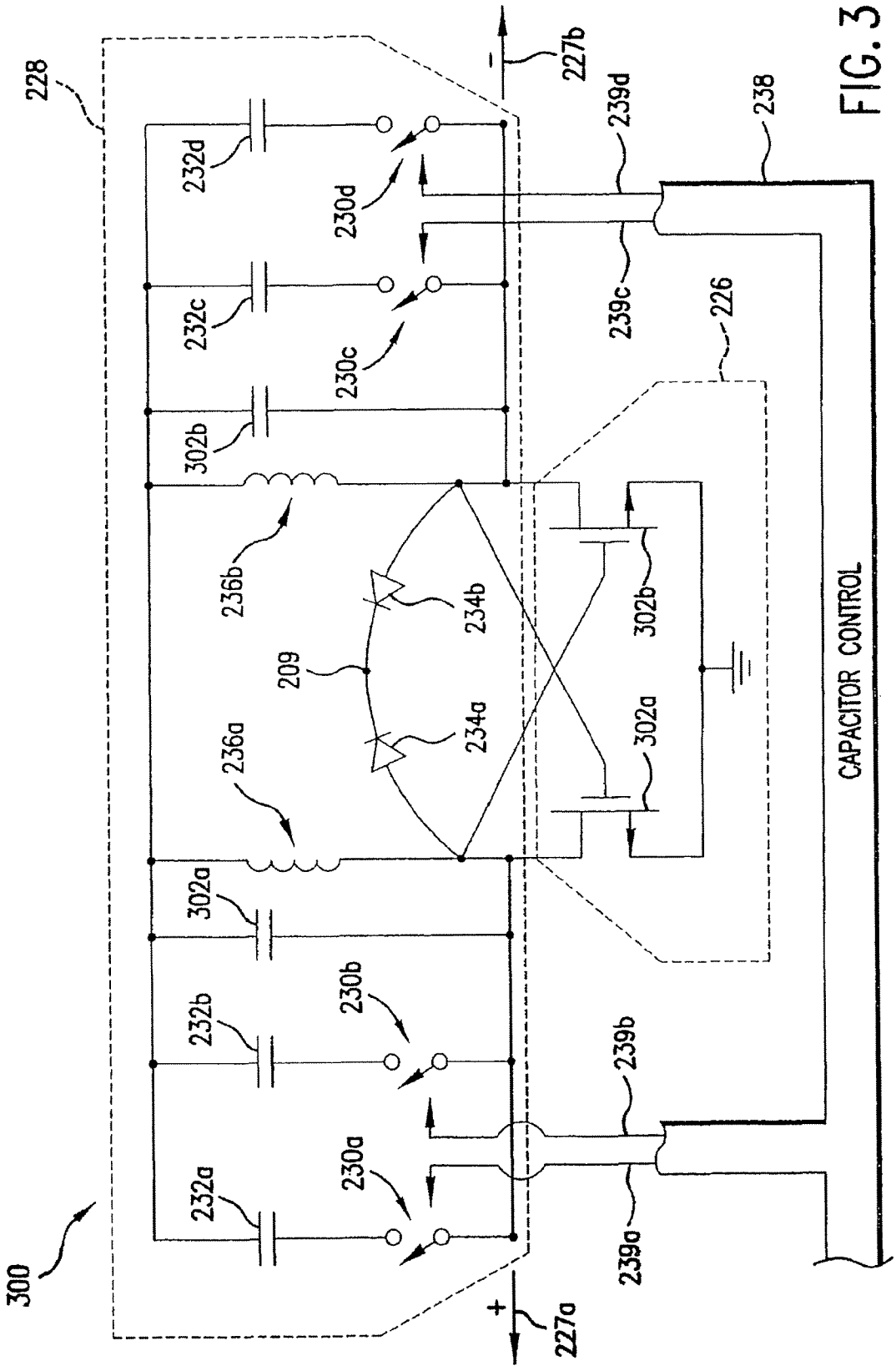

228
227b
239d
238
232d
230d
232c
230c
239c
226
302b
236b
234b
209
234a
236a
302b
302a
CAPACITOR CONTROL
302a
230b
232b
239b
230a
232a
239a
300
227a

FIG. 3

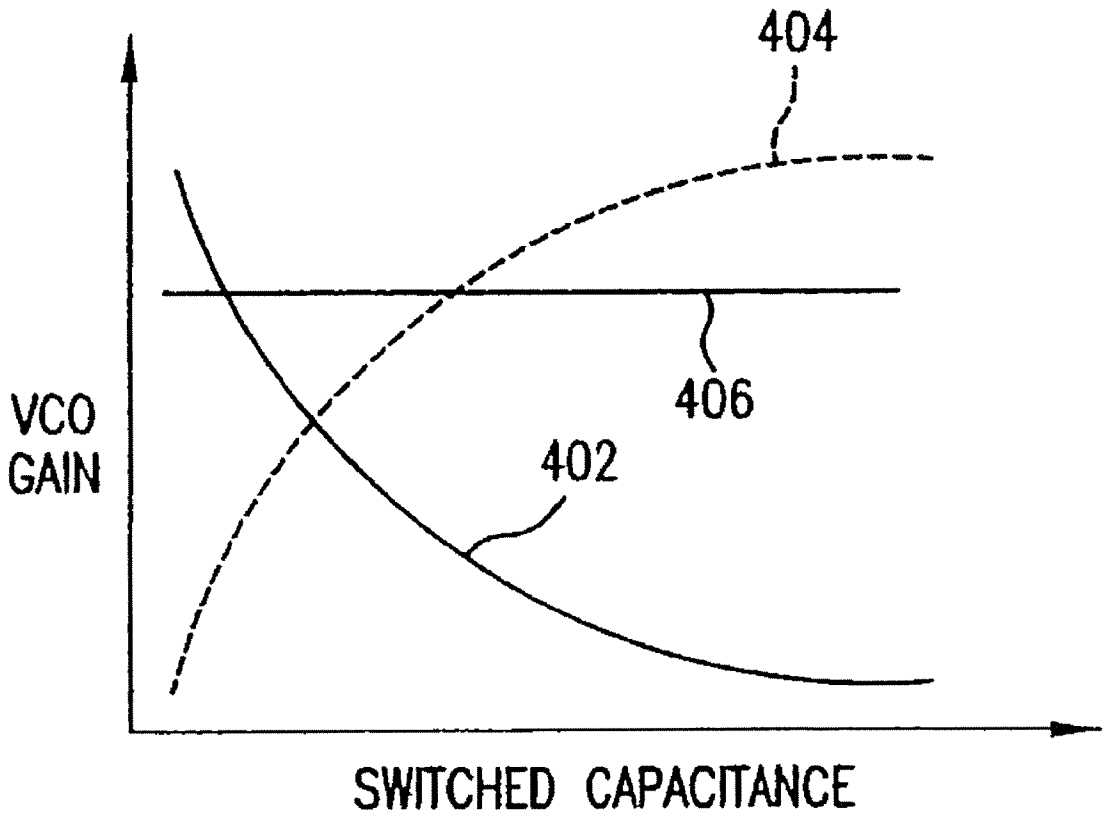


FIG. 4

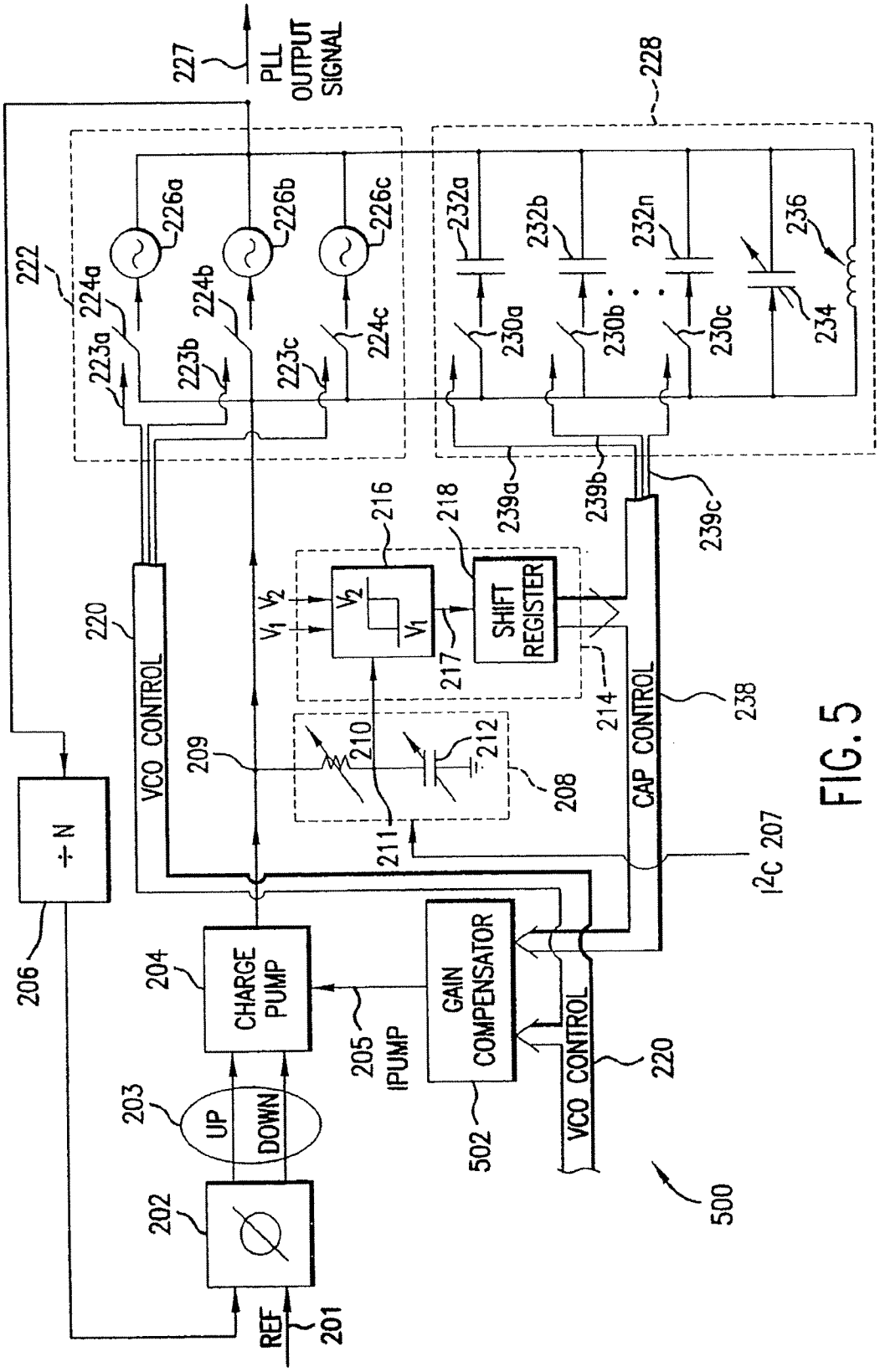

227
PLL OUTPUT SIGNAL
228
222
226a
226b
226c
224a
224b
224c
223a
223b
223c
232a
232b
232n
236
230a
230b
230c
234
239a
239b
239c
216
218
220
VCO CONTROL
V1 V2
SHIFT REGISTER
217
214
238
CAP CONTROL
209
210
212
208
211
I2C 207
÷ N
206
204
CHARGE PUMP
205
IPUMP
GAIN COMPENSATOR
203
UP
DOWN
202
502
VCO CONTROL
220
500
REF
201

FIG. 5

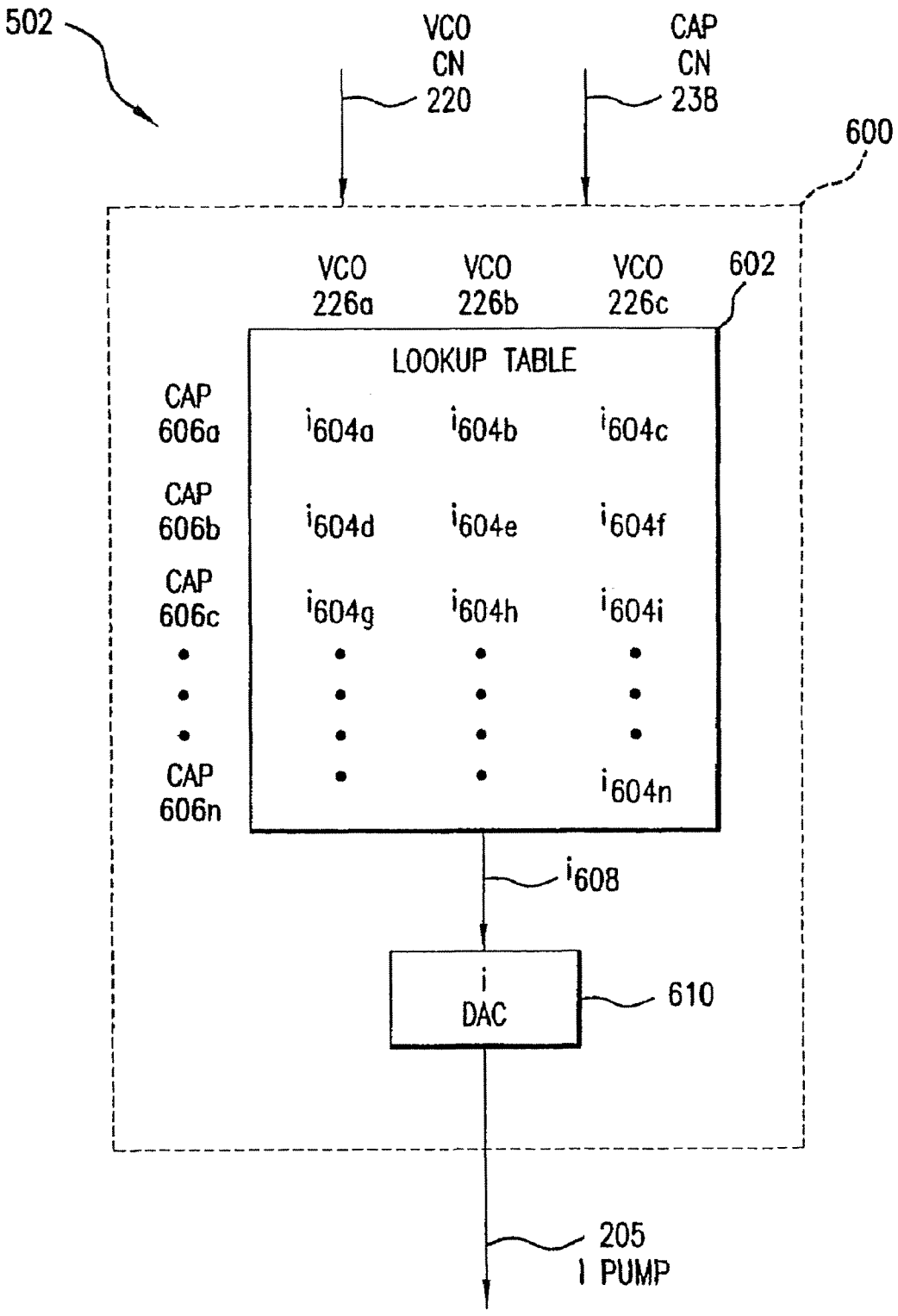

502
VCO CN 220
CAP CN 238
600
VCO 226a
VCO 226b
VCO 226c
602
LOOKUP TABLE
CAP 606a
CAP 606b
CAP 606c
CAP 606n
i604a
i604b
i604c
i604d
i604e
i604f
i604g
i604h
i604i
i604n
i608
i DAC
610
205 I PUMP

FIG. 6

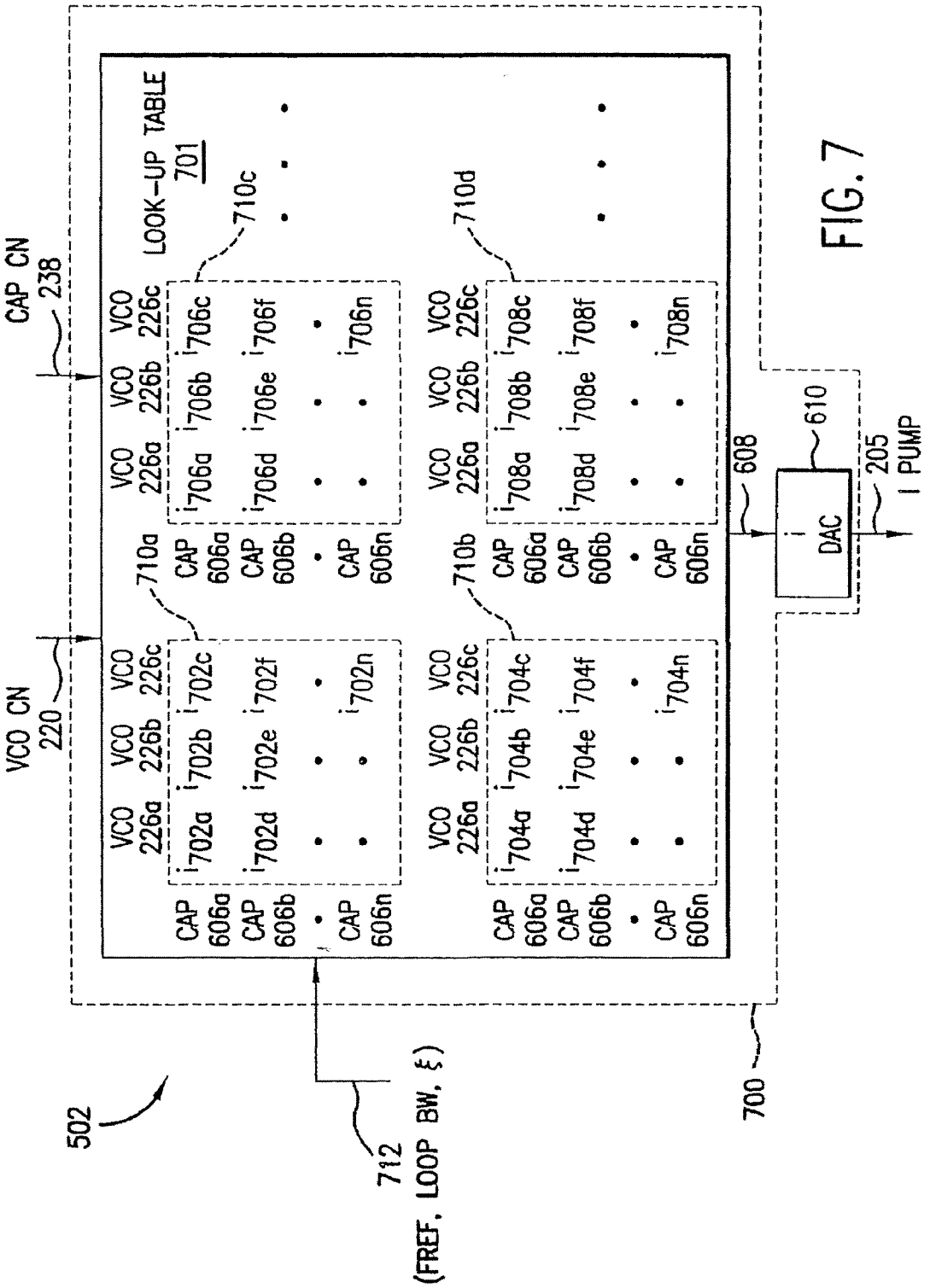

LOOK-UP TABLE
701
CAP CN
238
VCO CN
220
VCO
226a
VCO
226b
VCO
226c
CAP
606a
CAP
606b
CAP
606n
710a
710b
710c
710d
DAC
608
610
205
I PUMP
502
712
(FREF, LOOP BW, ξ)
700

FIG. 7

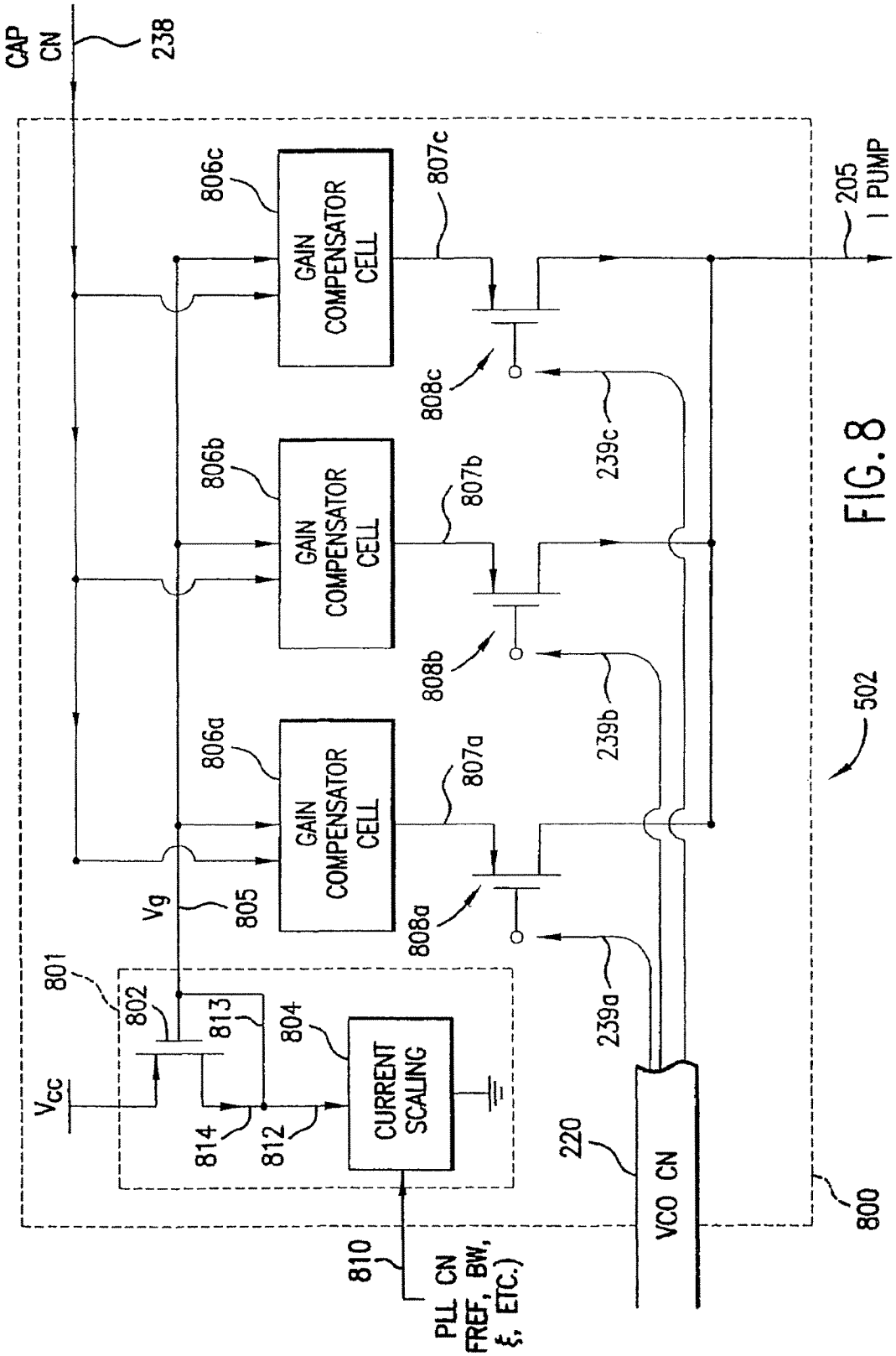
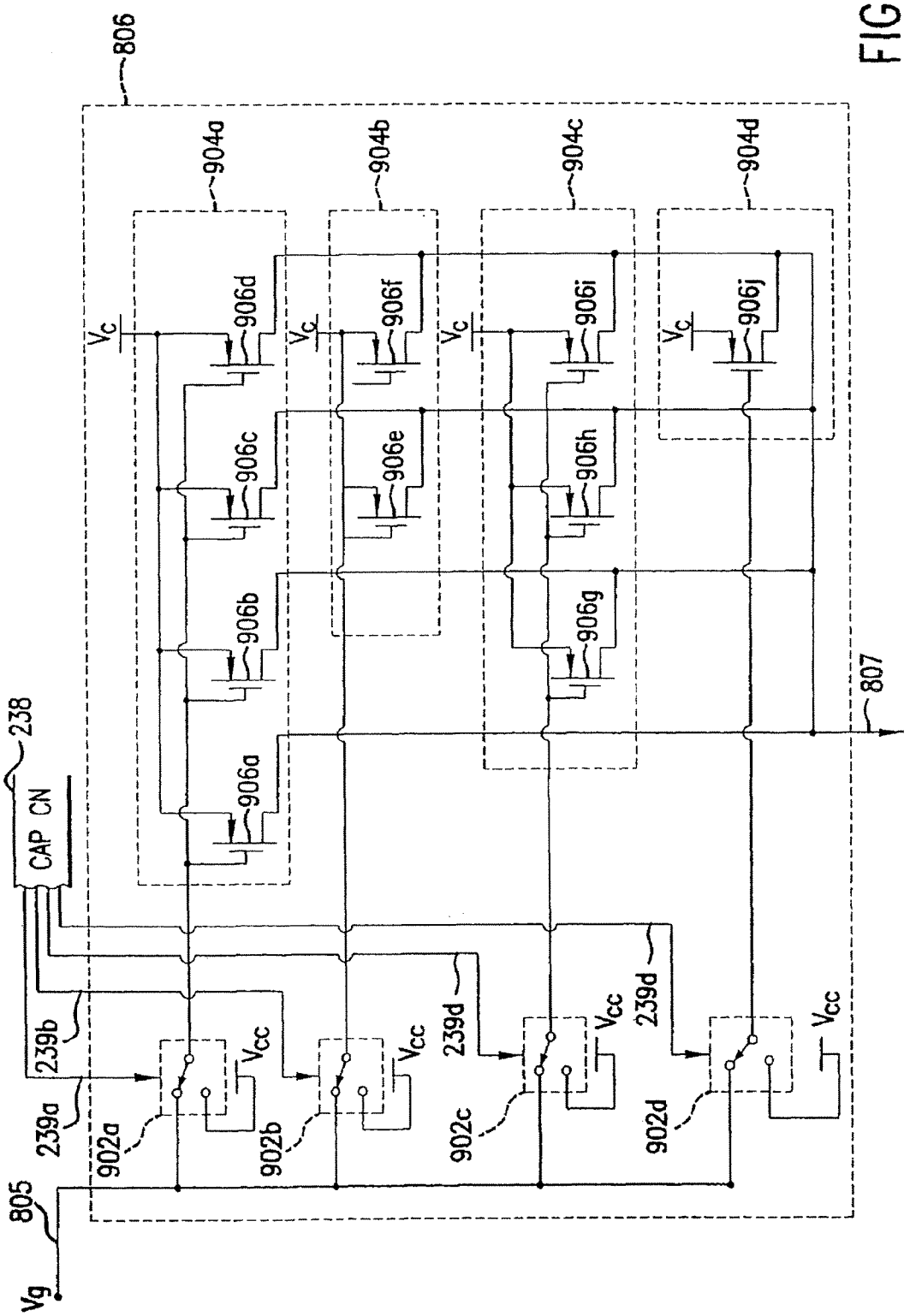

CAP CN
238
806c
GAIN COMPENSATOR CELL
807c
205
I PUMP
808c
239c
806b
GAIN COMPENSATOR CELL
807b
808b
239b
806a
GAIN COMPENSATOR CELL
807a
808a
239a
502
Vg
805
801
802
813
804
CURRENT SCALING
Vcc
814
812
220
VCO CN
800
810
PLL CN
FREF, BW, ξ, ETC.)

FIG. 8

FIG. 9


806
904a
904b
904c
904d
906d
906f
906i
906j
906c
906e
906h
906b
906g
906a
Vc
807
238
CAP CN
239b
239a
239d
239d
Vcc
Vcc
Vcc
Vcc
902a
902b
902c
902d
805
Vg

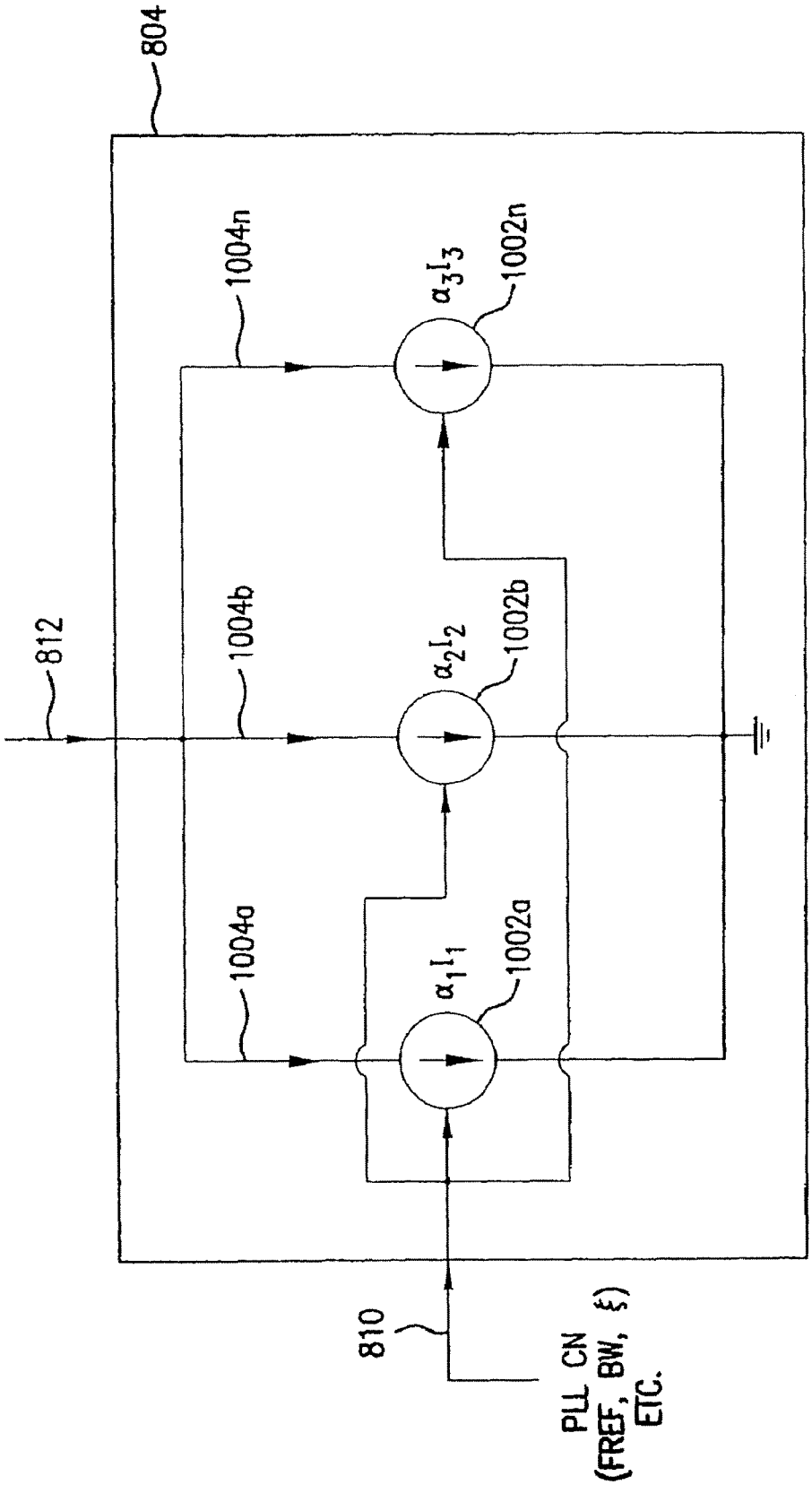

804
1004n
$\alpha_3 I_3$
1002n
812
1004b
$\alpha_2 I_2$
1002b
1004a
$\alpha_1 I_1$
1002a
810
PLL CN
(FREF, BW, ξ)
ETC.

FIG. 10

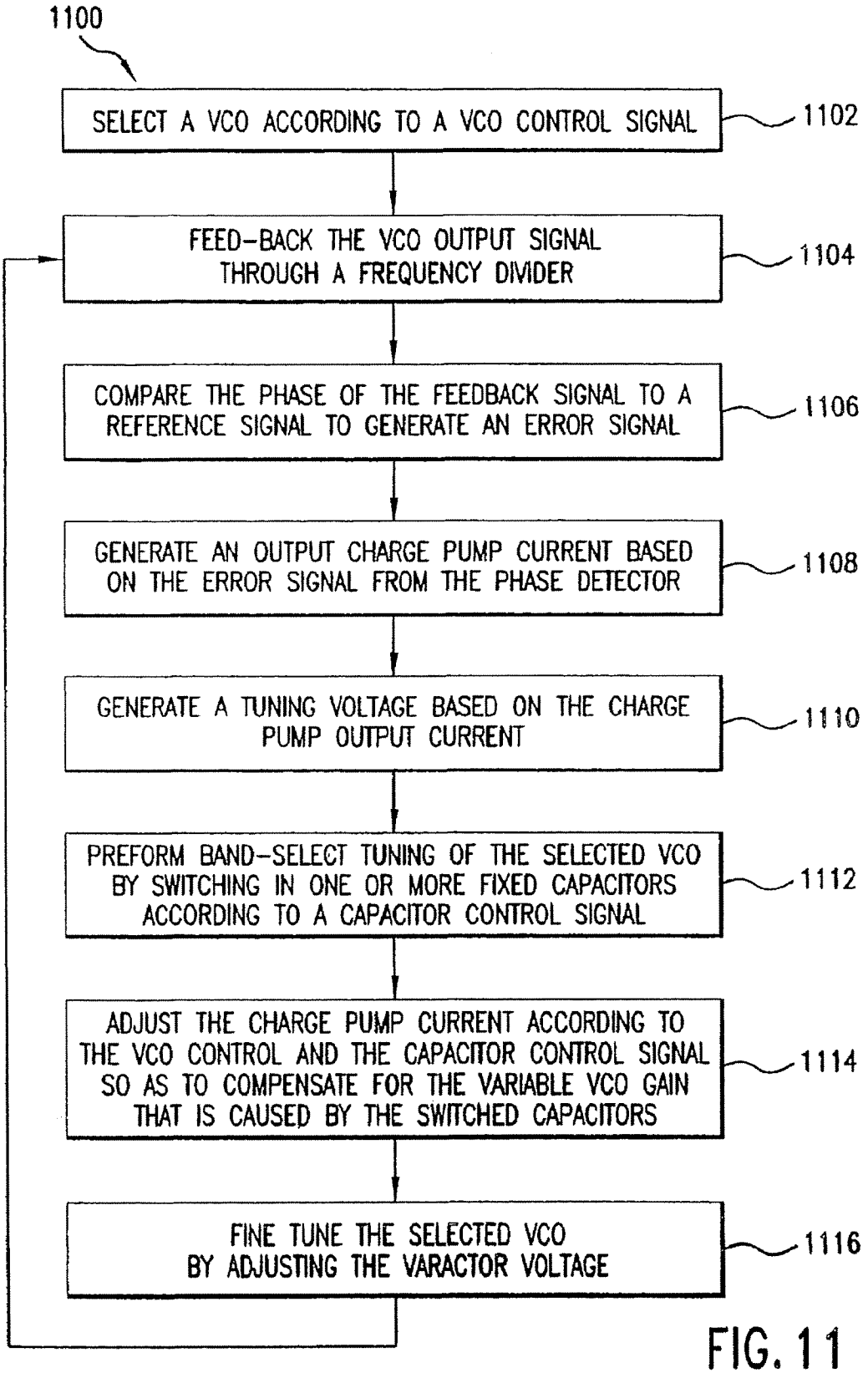

1100
SELECT A VCO ACCORDING TO A VCO CONTROL SIGNAL
1102
FEED-BACK THE VCO OUTPUT SIGNAL THROUGH A FREQUENCY DIVIDER
1104
COMPARE THE PHASE OF THE FEEDBACK SIGNAL TO A REFERENCE SIGNAL TO GENERATE AN ERROR SIGNAL
1106
GENERATE AN OUTPUT CHARGE PUMP CURRENT BASED ON THE ERROR SIGNAL FROM THE PHASE DETECTOR
1108
GENERATE A TUNING VOLTAGE BASED ON THE CHARGE PUMP OUTPUT CURRENT
1110
PREFORM BAND-SELECT TUNING OF THE SELECTED VCO BY SWITCHING IN ONE OR MORE FIXED CAPACITORS ACCORDING TO A CAPACITOR CONTROL SIGNAL
1112
ADJUST THE CHARGE PUMP CURRENT ACCORDING TO THE VCO CONTROL AND THE CAPACITOR CONTROL SIGNAL SO AS TO COMPENSATE FOR THE VARIABLE VCO GAIN THAT IS CAUSED BY THE SWITCHED CAPACITORS
1114
FINE TUNE THE SELECTED VCO BY ADJUSTING THE VARACTOR VOLTAGE
1116

FIG. 11

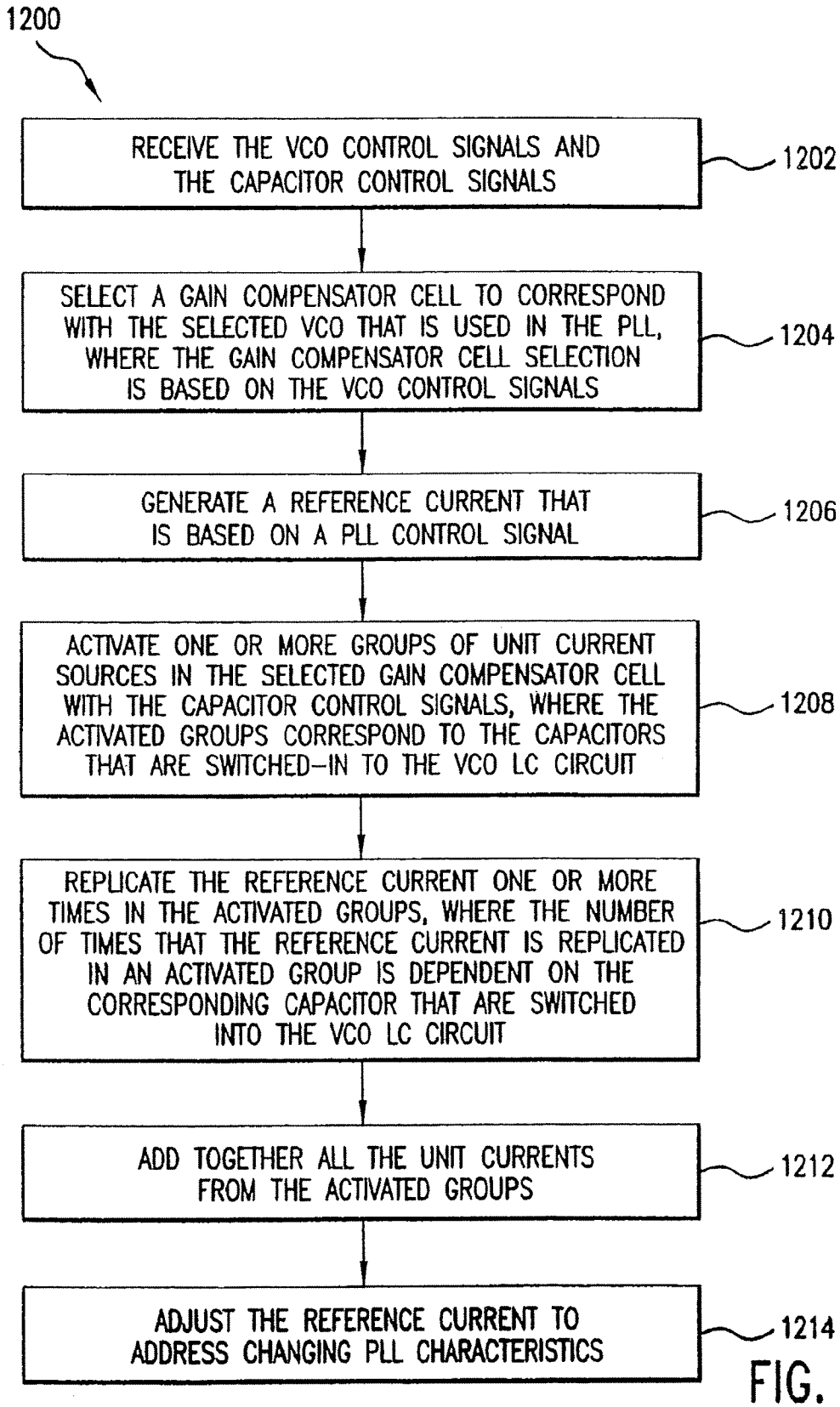

1200
RECEIVE THE VCO CONTROL SIGNALS AND THE CAPACITOR CONTROL SIGNALS
1202
SELECT A GAIN COMPENSATOR CELL TO CORRESPOND WITH THE SELECTED VCO THAT IS USED IN THE PLL, WHERE THE GAIN COMPENSATOR CELL SELECTION IS BASED ON THE VCO CONTROL SIGNALS
1204
GENERATE A REFERENCE CURRENT THAT IS BASED ON A PLL CONTROL SIGNAL
1206
ACTIVATE ONE OR MORE GROUPS OF UNIT CURRENT SOURCES IN THE SELECTED GAIN COMPENSATOR CELL WITH THE CAPACITOR CONTROL SIGNALS, WHERE THE ACTIVATED GROUPS CORRESPOND TO THE CAPACITORS THAT ARE SWITCHED-IN TO THE VCO LC CIRCUIT
1208
REPLICATE THE REFERENCE CURRENT ONE OR MORE TIMES IN THE ACTIVATED GROUPS, WHERE THE NUMBER OF TIMES THAT THE REFERENCE CURRENT IS REPLICATED IN AN ACTIVATED GROUP IS DEPENDENT ON THE CORRESPONDING CAPACITOR THAT ARE SWITCHED INTO THE VCO LC CIRCUIT
1210
ADD TOGETHER ALL THE UNIT CURRENTS FROM THE ACTIVATED GROUPS
1212
ADJUST THE REFERENCE CURRENT TO ADDRESS CHANGING PLL CHARACTERISTICS
1214

FIG. 12

APPARATUS AND METHOD FOR PHASE LOCK LOOP GAIN CONTROL USING UNIT CURRENT SOURCES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a divisional of U.S. application Ser. No. 11/019,451, filed Dec. 23, 2004, which is a continuation of U.S. application Ser. No. 10/443,741, filed May 23, 2003 (now issued U.S. Pat. No. 6,838,947 B2), which is a continuation of U.S. application Ser. No. 09/811,611, filed Mar. 20, 2001 (now issued U.S. Pat. No. 6,583,675 B2). The above-identified applications are hereby incorporated herein by reference in their entirety.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention generally relates to gain control in a phase lock loop, and more specifically to phase lock loop gain control using scaled unit current sources.

2. Background Art

Radio frequency (RF) transmitters and receivers perform frequency translation by mixing an input signal with a local oscillator (LO) signal. Preferably, the LO signal should have a frequency spectrum that is as close to a pure tone as possible in order to maximize system performance during the signal mixing operation. The deviation of the LO signal from a pure tone is quantified as phase noise or phase jitter, and is generally referred to as spectral purity. In other words, a LO signal with good spectral purity has low phase noise.

Phase-locked loops (PLLs) are often used in frequency synthesizers to generate the LO signal. A PLL frequency synthesizer produces an output signal, typically a sinewave or square wave, that is a frequency multiple of an input reference signal. The PLL output signal is also in phase synchronization with the input reference signal. PLLs are feedback loops, and therefore are susceptible to instability. Therefore, loop stability is a key performance parameter for PLLs, in addition to spectral purity of the output signal.

A resonant-tuned voltage controlled oscillator (VCO) is typically utilized in a PLL to generate the PLL output signal. A resonant tuned VCO includes an active device and a resonant LC circuit, where the impedance of the resonant LC circuit becomes a short or an open at a resonant frequency. When the resonant circuit is connected in parallel with the active device, a positive feedback path is created in the active device at the resonant frequency of the LC circuit. The positive feedback path causes the active device to oscillate at the resonant frequency of the LC circuit.

The resonant tuned LC circuit typically includes multiple fixed capacitors that can be switched in or out of the LC circuit, a varactor diode, and at least one inductor. The resonant frequency of the LC circuit (and therefore the oscillation frequency of the VCO) is tuned via a coarse tuning mechanism and a fine tuning mechanism. Coarse frequency tuning (or band-selection) is performed by switching one or more of the fixed capacitors in the LC circuit. Whereas, fine frequency tuning is performed by changing the voltage across the varactor diode, which produces a capacitance that varies depending on the applied tuning voltage. Both tuning mechanisms operate by changing the capacitance, and therefore the resonant frequency of the LC circuit. The varactor tuning range is slightly larger than one fixed capacitor, and therefore provides some overlap between the fixed capacitors.

VCO gain is defined as the VCO frequency shift per unit change in the varactor tuning voltage. A problem with varactor-tuned VCOs is that the VCO gain verses fixed capacitance is variable. In other words, the VCO frequency shift verses tuning voltage is dependent on the fixed capacitance that is switched-in to the LC circuit. The variable VCO gain creates difficulties when designing a PLL because the entire PLL loop gain, bandwidth, and damping response varies with respect to the oscillator frequency. This in turn makes it difficult to optimize the output phase noise and reduces overall spectral purity. Therefore, it is desirable to compensate for the variable VCO gain, in order to maintain the overall PLL gain at a desired optimum value.

In addition to the VCO gain, it is desirable to adjust or tune other PLL characteristics, such as loop bandwidth, reference frequency, and damping factor, without having to tune or replace PLL components.

BRIEF SUMMARY OF THE INVENTION

The gain compensator invention compensates for gain variation in a varactor-tuned VCO in order to maintain the overall PLL gain at a desired level over frequency. The VCO includes a LC circuit that has multiple fixed capacitors that are arranged in parallel with the varactor diode and the active portion of the VCO. The fixed capacitors are switched-in to the LC circuit by corresponding capacitor control signals. Coarse frequency tuning (also called band-select tuning) is performed by adding or subtracting one or more of the fixed capacitors to the LC circuit according to the capacitor control signal. Fine frequency tuning is performed by adjusting the tuning voltage on the varactor diode, where the VCO gain is defined as the frequency shift per unit change in varactor tuning voltage. VCO gain varies with the fixed capacitance that is switched-in to the LC circuit, and therefore changes with band-select tuning of the VCO. The gain compensator compensates for the variable VCO gain by generating a reference charge pump current for the PLL based on information that is carried in the capacitor control signal. Therefore, the gain compensator is able to simultaneously adjust the charge pump current to maintain an overall flat PLL gain as fixed capacitors are incrementally added to (or subtracted from) the LC circuit.

The gain compensator includes one or more cells that each correspond to a particular VCO that can be switched into the PLL at a given time. A VCO control signal selects a particular VCO for the PLL based on frequency, and also activates the appropriate cell. Each cell includes a plurality of unit current sources, where each unit current source substantially replicates (or copies) a pre-defined reference scale current. The unit current sources are arranged into one or more groups, where each group corresponds to a fixed capacitor in the LC circuit. Each group of unit current generates a portion of the total pump current when the corresponding capacitor is switched-in to the LC circuit. The number of unit current sources in each group is determined to compensate for the variable VCO gain that occurs when the corresponding fixed capacitor is switched-in to the LC circuit. Each group of unit current sources is activated by the same capacitor control signal that controls the corresponding fixed capacitor. Therefore, when a fixed capacitor is switched-in to the LC circuit, the corresponding group of unit current sources is simultaneously activated and switched-in to the cell to compensate for the variable VCO gain that is caused by the fixed capacitor.

An advantage of the gain compensator invention is that the number of unit current sources that are activated for a

corresponding fixed capacitor is arbitrary, but the current produced is linearly proportional to the reference scale current. In other words, there is no predefined relationship between the number of unit current sources in each group that would restrict the relative amount of current produced by each group. Therefore, the total pump current can be freely optimized to incrementally adjust for the variable VCO gain that is associated with various combinations of fixed capacitors.

A further advantage of the gain compensator invention is that the reference scale current for the gain compensator cells is generated based on a PLL control signal. The PLL control signal specifies various PLL characteristics, such as the frequency of the reference signal, the PLL bandwidth, and the PLL damping factor, etc. Since the unit current sources are configured to replicate the reference scale current, all of the unit current sources can be simultaneously adjusted by changing the reference scale current. Therefore, the charge pump current can be efficiently adjusted to tune the mentioned characteristics of PLL for different operating conditions, without requiring the replacement of PLL components.

Further features and advantages of the present invention, as well as the structure and operation of various embodiments of the present invention, are described in detail below with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS/FIGURES

The present invention is described with reference to the accompanying drawings. In the drawings, like reference numbers indicate identical or functionally similar elements. Additionally, the left-most digit(s) of a reference number identifies the drawing in which the reference number first appears.

FIG. **1**A illustrates a tuner **100** that is an example tuner environment for the present invention;

FIG. **1**B illustrates dual frequency conversion that is performed by the tuner **100**;

FIG. **2** illustrates a PLL **200** that can be used with the tuner **100**;

FIG. **3** illustrates a VCO **300** that can be used with the PLL **200**;

FIG. **4** illustrates variable VCO gain;

FIG. **5** illustrates a PLL **500** that includes a gain compensator **502**, according to embodiments of the present invention;

FIG. **6** illustrates a ROMDAC **600** that is one embodiment of a gain compensator, according to embodiments of the present invention;

FIG. **7** illustrates a ROMDAC **700** having an expanded look-up table **701**, according to embodiments of the present invention;

FIG. **8** illustrates a gain compensator **800** having a current scaler **804** that forms a current mirror configuration with one or more gain compensator cells **806**, according to embodiments of the present invention;

FIG. **9** illustrates a gain compensator cell **806** having multiple unit current sources, according to embodiments of the present invention;

FIG. **10** illustrates the current scaler **804**, according to embodiments of the present invention;

FIG. **11** illustrates a flowchart **1100** that describes the operation of a PLL having compensation for nonlinear VCO gain, according to embodiments of the present invention; and

FIG. **12** illustrates a flowchart **1200** that describes the operation of a gain compensator cell, according to embodiments of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

1. Example Tuner Application

Before describing the invention in detail, it is useful to describe an example tuner application for the invention. The invention is not limited to the tuner application that is described here, and is applicable to other tuner and non-tuner applications as will be understood to those skilled in the relevant arts based on the discussions given herein.

FIG. **1**A illustrates a schematic of a tuner assembly **100** that has an RF automatic gain control circuit (AGC) **102**, and a tuner **134**. The tuner assembly **100** receives an RF input signal **101** having multiple channels and down-converts a selected channel to an IF frequency, to produce an IF signal **133**. For instance, the RF input signal **101** can include multiple TV channels that typically have 6 MHZ frequency spacings and cover a range of 57–860 MHZ, and where the selected channel is down-converted to an IF frequency at 44 MHZ, 36 MHZ or some other desired IF frequency for further processing. The structure and operation of the AGC circuit **102** and the tuner **134** are described in further detail below.

The AGC circuit **102** provides automatic gain control using a variable resistor **104** and a low noise amplifier (LNA) **106**. The variable resistor **104** attenuates the RF input signal **101** according to a control signal **103**. In embodiments, the control signal **103** is based on the signal amplitude of the IF signal **133** so that the RF front-end gain can be adjusted to achieve a desired amplitude for the IF signal **133**. The LNA **106** provides low noise amplification and converts a single-ended input signal to a differential RF signal **107**.

The tuner **134** has a dual conversion architecture (one up-conversion, one down-conversion) that includes an input mixer **108** and an image reject mixer **118**. The input mixer **108** is driven by a first phase locked loop (PLL) **110** that has coarse tuning capability from 1270 2080 MHz. The image reject mixer **118** has two component mixers **120***a* and **120***b* that are driven in quadrature by a second PLL **124** through a quadrature polyphase filter **122**. The PLL **124** has a relatively fixed frequency of 1176 MHZ (for a 44 MHZ IF) and has fine frequency tuning capability. A polyphase filter **126** is coupled to the output of the image reject mixer **118** to combine the quadrature outputs of the mixers **120**. Two separate off-chip surface acoustic wave (SAW) filters **114** and **130** are used to perform IF filtering in the tuner **134**. The first SAW filter **114** is connected between the first mixer **108** and the image reject mixer **118**. The passband of the SAW filter **114** is centered at 1220 MHZ, and is only a few channels wide (e.g. 1–3 channels wide or 18 MHZ for 6 MHZ TV channel spacings). The second SAW filter **130** has a passband at 44 MHZ and is coupled to the output of the polyphase filter **126**. Additionally, various on-chip amplifiers **108**, **116**, **128**, and **132** are included throughout the tuner **134** to provide signal amplification, as necessary.

The operation of the tuner **134** is described as follows and in reference to the frequency spectrum that is illustrated in FIG. **1**B. The first mixer **108** mixes the RF signal **107** with a LO signal **109** that is generated by the PLL **110**. Since to a first IF **111** having a frequency that is above the 57–860 MHZ input frequency band. The first IF **111** is sent off-chip

to the SAW filter **114**, which has a narrow passband window centered at 1220 MHz. The first SAW filter **114** selects a desired channel **115** that is within its narrow passband window, and substantially rejects all of the remaining channels. Therefore, a particular channel is selected by varying the frequency of the LO signal **109** so that the desired channel is up-converted into the narrow passband of the IF filter **114**. The desired channel **115** (at 1220 MHZ) is sent back on-chip to the image reject mixer **118** that is driven by a quadrature LO signal **119** from the polyphase filter **122**. The image reject mixer **118** down-converts the desired channel **115** to a 44 MHZ IF signal **127** that appears at the output of the polyphase filter **126**, where I and Q components of the IF signal **127** are combined in the polyphase filter **126**. Finally, the IF signal **127** is filtered a second time by the bandpass SAW filter **130** to reject any unwanted frequency harmonics, producing the output IF signal **133** at 44 MHZ and carrying the information in the desired channel.

The specific frequencies mentioned in the description of the tuner assembly **100**, and throughout this application, are given for example purposes only and are not meant to be limiting. Those skilled in the arts will recognize other frequency applications for the tuner assembly **100** based on the discussion given herein. These other frequency applications are within the scope and spirit of the present invention.

2. Phase Lock Loop:

The first PLL **110** and the second PLL **124** are represented by the PLL **200** that is illustrated in FIG. **2**. The PLL **200** generates a PLL output signal **227** that is a frequency multiple of a reference signal **201**, and where the output signal **227** is phase-locked to the reference signal **201**. The PLL **200** self-corrects for any phase (and therefore frequency) variations between the reference signal **201** and the output signal **227** via a feedback mechanism that is described as follows. The structure and operation of the PLL **200** are described as follows.

The PLL **200** structure includes: a phase detector **202**, a charge pump **204**, a frequency divider **206**, a loop filter **208**, a coarse tuning circuit **214**, a VCO assembly **222**, and a LC resonant circuit **228**. The loop filter **208** includes a variable resistor **210** and a variable capacitor **212** that are controlled by an I^2C signal **207**. The coarse tuning circuit **214** includes a comparator **216** and a shift register **218**. The VCO assembly **222** includes multiple component VCOs **226***a–c*, where each VCO **226** preferably covers a particular frequency band. A VCO **226** is switched-in to the PLL **200** by closing a corresponding switch **224**. The switches **224***a–c* are controlled by corresponding control signals **223***a–c* that make-up a VCO control bus **220**. The LC resonant circuit **228** is connected in parallel with the VCO assembly **222** and includes: multiple fixed capacitors **232***a–n* having corresponding switches **230***a–n*, a varactor **234**, and an inductor **236**. One or more of the fixed capacitors **232** are switched in-parallel with the selected VCO **226** by closing the corresponding switch(s) **230**. The switches **230** are controlled by corresponding control signals **239***a n* that make-up a capacitor control bus **238**.

Each VCO **226** is a resonant tuned oscillator whose oscillation frequency is controlled by the resonant frequency of the parallel LC circuit **228**. The resonant frequency of the LC circuit **228** is determined by the relative total capacitance and inductance according to the equation:

$$f_0=(1/2\pi)\cdot 1/\mathrm{sqrt}(LC) \qquad \text{Eq. 1}$$

As discussed further below, coarse frequency tuning (e.g. band-selection) of the selected VCO **226** is performed by switching in one or more of the fixed capacitors **232** into the LC circuit **228**. This changes the resonant frequency of the LC circuit **228**, and therefore the oscillation frequency of the selected VCO **226**. Fine frequency tuning is performed by changing the control voltage on the varactor **234**, which has a variable capacitance that changes with applied voltage. The VCO gain is defined as the change in the VCO output frequency per unit change in the voltage across the varactor **234**.

The PLL **200** operates based on known PLL feedback principles. A VCO **226** is selected based on the desired frequency of operation for the PLL **200**, and is switched-in to the PLL **200** by closing the appropriate switch **224** using the appropriate control signal **223**. The PLL output signal **227** from the selected VCO **226** is fed back to a phase detector **202** through the frequency divider **206**. The frequency divider **206** normalizes the frequency of the output signal **227** to that of the reference signal **201** for comparison in the phase detector **202**. The phase detector **202** compares the phase of the output signal **227** to the reference signal **201**, and generates a DC error signal **203** that represents the phase difference between the two signals. The charge pump **204** receives the error signal **203** and a reference pump current **205**. The charge pump **204** sources (or sinks) a percentage of the pump current **205** based on the error signal **203**, as will be understood by those skilled in the arts. The output current of the charge pump **204** drives the loop filter **208** to produce a tuning voltage **209**. Part of the tuning voltage **209** is dropped across the variable capacitor **212** to generate a tuning voltage **211**. As discussed further below, the tuning voltages **209** and **211** control the oscillation frequency of the selected VCO **226**.

The tuning voltages **209** and **211** adjust the resonant frequency of the LC circuit **228** (and therefore the oscillation frequency of the selected VCO **226**) via a coarse tuning mechanism and a fine tuning mechanism, respectively. More specifically, the coarse tuning circuit **214** adds (or subtracts) one or more of the fixed capacitors **232***a–n* to the LC circuit **228** based on the tuning voltage **211**. Similarly, the tuning voltage **209** directly adjusts the voltage (and therefore the capacitance) of the varactor **234** to implement fine frequency tuning. Both tuning mechanisms adjust the oscillation frequency of the VCO **226** by changing the capacitance of the LC circuit **228**, which shifts the resonant frequency of the LC circuit **228**. The tuning range of the varactor **234** is slightly larger than one fixed capacitor **232**, and therefore provides some tuning overlap between the fixed capacitors **232**. The coarse tuning circuit **214** is described further below.

The coarse tuning circuit **214** includes a window comparator **216** and a bi-directional shift register **218**. The window comparator **216** receives the tuning voltage **211** and also receives input reference voltages v_1 and v_2. The window comparator **216** determines if the voltage **211** is within a voltage "window" that is defined between the input references voltages v_1 and v_2, and generates a control signal **217** that controls the bi-directional shift register **218** based on this determination. The shift register **218** stores a series of bits that control the capacitor switches **230** via the control bus **238** to add (or subtract) the corresponding capacitors **232** to (or from) the LC circuit **228**. A "1" bit on the control line **239** causes the corresponding switch **230** to close and thereby adds the corresponding capacitor **232** to the LC circuit **228**. A "0" bit on the control line **239** causes the switch **230** to open and thereby subtracts the corresponding capacitor **232** from the LC circuit **228**.

The coarse tuning circuit **214** operates to self-correct coarse variations in the oscillation frequency of the selected VCO **226** by adding or subtracting capacitors **232**, based on the tuning voltage **211**. If the comparator **216** determines that the voltage **211** is below v_1, then the comparator **216** causes a series of "1"s to be shifted through the shift register **218**, which incrementally adds capacitors **232** to the LC circuit **228** until the tuning voltage **211** is within the v_1-to-v_2 voltage window. If the comparator **216** determines that the voltage **211** is above the voltage v_2, then the comparator **216** causes a series of "0"s to be shifted through the shift register **218**, which incrementally subtracts capacitors **232** from the LC circuit **228** until the tuning voltage **211** is within the v_1-to-v_2 voltage window. As described above, the frequency of the selected oscillator **226** changes whenever capacitance is added to, or subtracted from, the LC circuit **228**. If the comparator **216** determines that the voltage **211** is within the voltage window defined by v_1 and v_2, then no action is taken and the fixed capacitance in the LC circuit **228** remains unchanged. In other words, the tuning voltage **211** is within an acceptable voltage range (or "window"), and correspondingly, the frequency of the output signal **227** is within an acceptable frequency range. Therefore the number of the fixed capacitors **232** that are switched-in to the LC circuit **228** is not changed.

3. Example VCO Configuration

FIG. **3** illustrates a differential VCO **300** as one embodiment of VCO **226** and the LC resonant circuit **228**. The VCO **300** is meant for example purposes only and is not meant to limit the invention in any way. Other oscillator configurations could be utilized to practice the invention, as will be understood by those skilled in the relevant arts based on the discussions given herein.

The VCO **300** includes the active VCO portion **226** and the resonant LC circuit **228**. The active portion includes a pair of cross coupled transistors **302***a* and **302***b* that oscillate at the resonant frequency of LC circuit **228**. In this cross-coupled configuration, the drain of transistor **302***a* is connected to the gate of transistor **302***b*. Likewise, the drain of transistor **302***b* is connected to the gate of the transistor **302***a*. The LC circuit **228** is also coupled to the drains of the transistors **302**. At resonance, the LC circuit **228** causes a positive feedback path between the cross-coupled transistors **302**, which causes the transistors to oscillate at the resonant frequency of the LC circuit **228**, producing the differential output signal **227**.

The oscillation frequency of the VCO **300** can be tuned by two mechanisms. Coarse frequency tuning (or band selection) is performed by adding or subtracting the fixed capacitors **232** using the corresponding switches **230**. Fine frequency tuning is performed by the tuning voltage **209**, which varies the capacitance produced by the series-connected varactor diodes **234***a* and **234***b* that are attached to the drains of the transistors **302**. The frequency change of VCO **300** per unit change in varactor **234** voltage is defined as the VCO gain. As stated above, the tuning range of the varactor **234** is slightly larger than the capacitance of one fixed capacitor **232**, and therefore provides some tuning overlap between the fixed capacitors **232**.

In one embodiment, the varactors **234** are PN junction varactors, and in an alternate embodiment these varactors **234** are MOSFET varactors, depending on the designer's preference.

4. PLL Gain Compensation

PLL gain is defined as the frequency change of the output signal verses the phase difference between the feedback signal and the reference signal. The forward PLL gain is determined as follows:

$$G(s)=K_{PHI}\cdot(R_{LF}+1/sC_{LF})\cdot K_{VCO}/s; \quad \text{Eq. 2}$$

where:
K_{PHI}=Phase detector gain (mA/radian)
R_{LF}=Loop filter resistance
C_{LF}=Loop filter capacitance
K_{VCO}=VCO gain (MHZ/volt)
s represents frequency

The feedback PLL gain H(s)=1/N, where N is the feedback frequency division ratio. The overall open loop gain is G(s)H(s), and the overall closed-loop gain is G(s)/[1+G(s)H(s)].

As described above, the PLL **200** performs coarse frequency tuning by incrementally adding (or subtracting) one or more of the fixed capacitors **232** that are in-parallel with the selected VCO **226**. Fine frequency tuning is performed by adjusting the voltage on the varactor **234**, where the VCO gain is defined as the frequency shift per unit change in the tuning voltage **209**. A problem with varactor-tuned VCOs is that the VCO gain verses the fixed capacitance **232** is variable. FIG. **4** illustrates this characteristic with a graph of VCO gain **402** verses fixed capacitance. As shown, the VCO gain curve **402** is reduced for a large fixed capacitance and is increased for a small fixed capacitance. Variable VCO gain is undesirable because it causes the PLL forward gain to change according to Eq. 2. In VCO applications with a large minimum-to-maximum capacitance tuning range, this VCO gain variability can cause loop instability, and reduced spectral purity in the PLL output signal. In a preferred embodiment, the VCO gain variability is compensated for by a compensator gain **404** so that the overall PLL gain **406** remains relatively flat for variations in fixed capacitance (and therefore VCO frequency). More specifically, the charge pump current **205** is compensated to counter the variable VCO gain so that the overall PLL gain is flat.

FIG. **5** illustrates a PLL **500** that has a gain compensator **502** to adjust the charge pump current **205** in order to linearize (and flatten) the overall PLL gain of the PLL **500**. The gain compensator **502** generates the pump current **205** based on the control information carried by the VCO control bus **220** and the capacitor control bus **238**. As discussed above, the VCO control bus **220** selects the appropriate VCO **226** based on the desired frequency range for the PLL output signal **227**. The capacitor control bus **238** selects the fixed capacitors **232** that are switched-in parallel with the selected VCO **226** for coarse frequency tuning of the VCO **226**. Therefore, the gain compensator **502** can tailor the reference pump current **205** for a specified VCO **226** at a specified fixed capacitance **232** value, and thereby compensate for the variable VCO gain vs. fixed capacitance.

FIG. **6** illustrates a read only memory digital-to-analog converter (ROMDAC) **600** that is one example embodiment of the gain compensator **502**, according to embodiments of the invention. Referring to FIG. **6**, the ROMDAC **600** includes a look-up table **602** and a current digital-to-analog converter **610**. The lookup table **602** stores pump current values **604***a–n* that are indexed by the selected VCO **226** and a fixed capacitance total **606**, where the fixed capacitance total **606** is the parallel sum of the capacitors **232** that are switched-in to the LC circuit **228**. The pump current values **604** are selected to compensate for the variable VCO gain vs.

capacitance, given an identified VCO **226** and the fixed capacitance total **606**. Preferably, the PLL **200** is characterized beforehand for each VCO **226** to determine the pump current values **604** that produces a flat overall PLL gain for various capacitance totals **606**. The look-up table **602** outputs a pump current value **608** that corresponds to identified VCO **226** and the fixed capacitance total **606**. The DAC **610** converts the pump current value **608** to the actual analog pump current **205** that drives the charge pump **204**. As capacitors **232** are added to or subtracted from the LC circuit **228**, the lookup table **602** selects the appropriate pump current value **604** so as to maintain a flat overall PLL gain. Therefore, the pump current **205** is adjusted for various total capacitance **606** to counteract the variable gain of the selected VCO **226**, and thereby flatten the overall gain of the PLL **500**.

An advantage of the ROMDAC **600** is that the pump current values **604** can be totally arbitrary and mathematically unrelated to each other. In other words, the pump currents **604** can be individually selected to produce an optimum overall PLL gain for a given VCO **226** and capacitance total **606**, without being restricted by any mathematical relationship. In an alternate embodiment, the various pump currents **604** are mathematically related to each other, or to the VCO control signal **220** or the capacitor control signal **238**.

In addition to PLL gain, it is desirable to tune various other PLL characteristics, such input reference frequency, loop bandwidth, damping factor, etc. This allows the same PLL to be used in different operating environments. For instance, it is often desirable to have a PLL configuration that is operable with a number of different reference frequencies. If the frequency of the reference signal **201** increases by factor of two, the PLL loop gain should preferably be adjusted to compensate for this increase so that the PLL loop remains stable and accurate. The PLL loop gain can be appropriately adjusted by reducing the frequency division of the frequency divider **206** by a factor of two. However, this would require replacement of the frequency divider **206** for each possible reference frequency, or the use of a programable frequency divider. Alternatively, the charge pump current could be reduced by a factor of two to get the same effect.

FIG. **7** illustrates a ROMDAC **700** as another embodiment of the gain compensator **502**, according to embodiments of the present invention. The ROMDAC **700** has an expanded lookup table **701** that has multiple sets **710***a–d* of pump current values, where the sets **710** tune various PLL characteristics in addition to compensating for variable VCO gain. Some PLL characteristics include, but are not limited to, PLL reference signal frequency, loop bandwidth, loop damping, etc. For example, sets **710***a* and **710***b* have pump current values **702***a–n* and **704***a–n*, respectively, which are customized for different reference frequencies. The pump current values **702***a n* can correspond to a first reference signal **201** frequency, and the pump current values **704***a–n* can correspond to a second reference signal **201** frequency. Therefore, if the frequency of the reference signal **201** changes, then the pump current value **608** can be selected from the appropriate pump current set **710**. In another example, the pump current sets **710***c* and **710***d* are customized to maintain loop bandwidth for different loop damping factors. The loop damping factor is increased or decreased by adjusting the variable resistor **210** in the loop filter **208**, which also determines the loop bandwidth. If the damping factor is changed, then the loop bandwidth can be held constant by selecting the appropriate set **710***c* or **710***d* that adjusts the charge pump current **205** to sufficiently counter the effect on the loop bandwidth.

To summarize, by storing multiple sets **710** of charge pump values in the lookup table **701**, multiple PLL characteristics can be adjusted or tuned in addition to PLL gain. This allows the same PLL **500** to be used under different PLL operating conditions, without replacing PLL components. The number of pump current sets **710** can be expanded to adjust any number of PLL characteristics, assuming there is sufficient memory space in the look-up table **701**.

FIG. **8** illustrates a gain compensator **800** that is another embodiment of the gain compensator **502** in FIG. **5**. The gain compensator **800** includes: a voltage generator **801**, gain compensator cells **806***a–c* that correspond to VCOs **226***a–c*, and PFETs **808***a–c* that correspond to the gain compensator cells **806***a–c*. Each gain compensator cell **806** generates a prospective pump current **807** that compensates for the variable VCO gain of its corresponding VCO **226** caused by the fixed capacitors **232**. Since only one VCO **226** is operational at a given time, only one prospective pump current **807** becomes the actual pump current **205** that feeds the charge pump **204**. The PFETs **808** operate as switches that are controlled by the VCO control signals **239** and select the appropriate prospective pump current **807** to correspond with the selected VCO **226**. For example, if the VCO **226***a* is the selected VCO **226**, then the control signal **239***a* causes the PFET **808***a* to conduct so that the current **807***a* becomes the feed for the pump current **205**. Accordingly, control signals **239***b* and **239***c* cutoff their respective PFETs **808***b* and **808***c*, and therefore only the current **807***a* feeds the pump current **205**.

The structure of the gain compensator cell **806** is shown in FIG. **9** and includes: switches **902***a–d* that are controlled by the respective capacitor control signals **239***a–d*, and unit current sources **906***a–j* that are arranged in groups **904***a–d*. Preferably, each unit current source **906** generates substantially the same amount of unit current (within transistor tolerances), where the amount of unit current is based on a gate voltage **805** that is generated by the voltage generate **801**. Each group **904** corresponds to a capacitor **232**, and generates a portion of the total pump current **205** when the respective capacitor **232** is switched-in to the LC circuit **228**. The number of unit current sources **906** in each group **904** is selected to compensate for the variable VCO gain that occurs when the corresponding capacitor **232** is switched-in to the LC circuit **228**. For example, group **904***a* corresponds to capacitor **232***a*, and has 4 unit current sources **906** to compensate for variable VCO gain that is caused by the capacitor **232***a*. Whereas, group **904***b* only has 2 unit current sources **906** to address the variable VCO gain caused by the capacitor **232***b*, and so on. Note that the number of current sources **4**,**2**,**3**,**1** that are shown in FIG. **9** for the groups **902***a–d* are for illustration purposes only, and is not meant to be limiting. Furthermore, the number of groups **904**, namely 4 as shown, is not meant to be limiting. In embodiments of the invention, the number of groups **904** should be less than or equal to the number of fixed capacitors **232**.

A group **904** is switched into the gain compensator cell **806** when the corresponding switch **902** connects Vg **805** to the unit current sources **906** in the group **904**. Once connected to a group **904**, the Vg **805** activates the current sources **906** and determines the current produced by each current source **906**. The switches **902** are controlled by the same capacitor control signals **239** that switches-in the respective capacitors **232** into the LC circuit **228**. Therefore, when a capacitor **232** is switched-in to the LC circuit **228**,

the corresponding group **904** will be switched-in to the gain compensator cell **806**, and therefore contribute to the prospective pump current **807**. For instance, if the capacitor **232***a* is switched-in to the LC circuit **228** by the capacitor control signal **239***a*, then the group **904***a* of unit current sources **906** will be switched-in to the gain compensator cell **806** by the same control signal **239***a*. Therefore, the current from the group **904***a* will contribute to the prospective pump current **807**, and thereby compensate for the variable VCO gain that is caused by the capacitor **232***a*. If the capacitor **232***b* is then switched-in to the LC circuit **228**, then the group **904***b* is switched-in to the gain compensator cell **806** to compensate for the variable VCO gain that is caused by the capacitor **232***b*. As such, the charge pump current **205** is simultaneously adjusted to maintain a flat overall PLL as the capacitors **232** are incrementally added to (or subtracted from) the LC circuit **228**.

Each unit current source **906** is preferably a PFET transistor, as shown. However, other transistor devices and configurations could be used for the unit current sources **906**, including N-FET transistors, as will be understood by those skilled in the relevant arts based on the discussions given herein. These other transistor devices and configurations are within the scope and spirit of the present invention. For example, simultaneous use of NFET and PPET current sources would permit the gain compensator to compensate for a non-monotonic VCO gain verses fixed capacitance characteristic.

The voltage generator **801** and the current sources **906** operate as a "current mirror", where the drain currents of the selected unit current sources **906** copy or "mirror" a reference scale current **812**. More specifically, the current scaler **804** sets the reference scale current **812**, which operates as a current sink for the PFET **802**. The PFET **802** operates as a diode because the gate and drain of the PFET **802** are shorted together by a conductor **813**. The drain current **814** of the PFET **802** is substantially the same as the reference scale current **812** because there is substantially zero current on the conductor **813**. The diode-connected PFET **802** generates the gate voltage **805** at its gate terminal to correspond with the drain current **814**, and therefore to the reference scale current **812**. If the drain current **814** deviates from the reference scale current **812** for some reason, then charge flows to/from the gate of the PFET **802** to bring the current **814** and the scale current **812** back in-line with each other. The gate voltage **805** is applied to the gate of the current sources **906** when their respective group **904** is selected by the capacitor control signals **239**. The current sources **906** will reproduce (or "mirror") the drain current **814** due to the common gate voltage **805**, if the device characteristics of the current sources **906** are sufficiently similar to those of the PFET **802**. This current mirror effect occurs because two or more FETs that have a common gate-to-source voltage and similar device characteristics will generate substantially the same drain current. If a group **904** is not switched-in by the corresponding capacitor control signal **239** (because the corresponding capacitor **232** is not switched in the LC circuit **228**), then the gates of the corresponding current sources **906** are connected to Vcc by the corresponding switch **902**. When connected to Vcc, these non-selected current sources **906** are cutoff and do not generate a unit current.

Preferably, the PFET **802** and the current sources **906** are fabricated on the same semiconductor wafer using the same process, which improves the commonality of device characteristics. However, if the size of the unit current sources **906** is scaled relative to the size of the PFET **802**, then the unit current sources **906** will generate a current that is proportional to the scale factor, as will be understood by those skilled in the relevant arts. This increases the flexibility of the gain compensator cell **806**, as the current sources **906** can be scaled relative to the PFET **802** as well as relative to each other.

The current scaler **804** sets the reference scale current **812** based on a PLL control signal **810**, where the PLL control signal **810** dictates various PLL characteristics such as the frequency of the reference signal **201**, the PLL loop bandwidth, and PLL loop damping, etc. FIG. **10** illustrates one embodiment of the current scaler **804** and includes weighted current sources **1002***a–n*. The weighted current sources **1002***a–n* sink currents **1004***a–n* based the PLL variables in the PLL control signal **810**. For example, the current source **1002***a* can be adapted to generate a current **1004***a* that is proportional to the frequency of the reference signal **201**, and the current source **1002***b* can be adapted to generate a current **1004***b* that is proportional to the desired loop bandwidth, etc. The currents **1004***a–n* are summed together to form the reference scale current **812** that feeds the diode-connected PFET **802**. Therefore, changes in the PLL variables are reflected in the reference scale current **812**, and ultimately in the drain currents of the unit current sources **906** because of the current mirror effect described herein. More specifically, the PFET drain current **814** is substantially the same as the reference scale current **812**, and gets copied to the drain currents of the unit current sources **906**.

An advantage of using the current scaler **800** is that all of the current sources **906** (that are in a selected group **904**) are simultaneously adjusted for changing PLL characteristics, in addition to compensating for variable VCO gain. Therefore, the prospective pump current **807** (and ultimately the final pump current **205**) can be efficiently tuned to compensate for changing PLL characteristics. This allows the same PLL to be utilized under different operating conditions. Furthermore, the current scaler **804** reduces the size of the overall gain compensator because multiple sets of current sources **906** are not needed to address changing PLL characteristics. In contrast, the ROMDAC **700** requires multiple sets **710** of current values to address changing PLL characteristics, which increases the size of the ROMDAC **700**.

The following examples illustrate the flexibility of the PLL **500** when using the current scaler **804** to adjust for changing PLL characteristics (besides VCO gain). In a first example, the frequency of the reference signal **201** increases by a factor of two, but the frequency divider **206** ratio is to remain constant. The same the frequency divider **206** can be used in the PLL **500** if the charge pump current **205** is reduced by approximately a factor of two. This is accomplished by reducing the reference scale current **812** that is generated by the current scaler **804**, causing a corresponding reduction in the gate voltage **805**. Through the current mirror effect, the current produced by the selected current sources **906** will be proportionally reduced by a factor of two. Therefore, the prospective current **807** (and the pump **205**) will also be reduced by a factor of two as desired, and the same PLL **500** can be reused for the new reference frequency.

In a second example, the PLL damping factor ζ is to be increased, but the PLL bandwidth is to be held constant. The PLL damping factor ζ is increased by increasing the resistance of the variable resistor **210** in the loop filter **208**. However, this also changes the loop bandwidth as will be understood by those skilled in the arts. To compensate, the current scaler **804** adjusts the reference scale current **812**,

and therefore the unit current sources **906** to produce a reference pump current **205** that compensates for the loop bandwidth.

In summary, and based on the examples herein, the gain compensator **800** is able to compensate for variable VCO gain and simultaneously tune other PLL characteristics by using the current mirror configuration described herein. These other PLL characteristics include but are not limited to changes in reference frequency, damping factor, and bandwidth.

The flowchart **1100** further describes the operation of the gain compensator **800** and VCO gain compensation according to embodiments of the present invention. The order of the steps in the flowchart **1100** is not limiting as all or some of the steps can be performed simultaneously or in a different order, as will be understood by those skilled in the arts.

In step **1102**, a VCO **226** is selected from the VCO **226***a–c* based on the desired frequency of the output signal **227**. The selection is made by closing the appropriate switch **230** using the control signals **239** to switch-in the desired VCO **226**.

In step **1104**, the VCO output signal **227** is fed back to the phase detector **202** through a frequency divider **206**. The frequency divider **206** normalizes the frequency of the output signal **227** to that of the reference signal **201** for comparison in the phase detector **202**.

In step **1106**, the phase detector **202** compares the phase of the output signal **227** to the reference signal **201**, and generates a DC error signal **203** that represents the phase difference between the two signals.

In step **1108**, the charge pump **204** sources or sinks a percentage of a reference pump current **205** based the error signal **203**.

In step **1110**, the output current from the charge pump **204** drives the loop filter **208** to produce a tuning voltage **209**.

In step **1112**, one or more fixed capacitors **232** are switched-in to (or switched-out of) the LC resonant circuit **228** based on the tuning voltage **209**, to perform coarse frequency tuning of the selected VCO **226**. The fixed capacitors **232** perform coarse frequency tuning by shifting the resonant frequency of the LC circuit **228**, and therefore the selected VCO **226**. The fixed capacitors **232** are switched-in to (or switched-out of) the LC circuit **228** by switching the corresponding switches **230** using the control signals **239**.

In step **1114**, the gain compensator **800** adjusts the charge pump reference current **205** to compensate for variable VCO gain that is caused by adding or subtracting the fixed capacitors **232**. The reference current **205** is adjusted based on the VCO control signals **239** and also the capacitor control signals **239**. In embodiments, the reference current **205** is adjusted simultaneously with the switching of the fixed capacitors **232** by the capacitor control signals **239**.

In step **1116**, the tuning voltage **209** fine tunes the frequency of the selected VCO **226** by changing voltage across the varactor **234**. The VCO gain vs. fixed capacitance is substantially linearized by the gain compensator **800** in step **1114**, thereby flattening the PLL gain and improving the PLL spectral purity.

Flowchart **1200** further describes step **1114**, where the gain compensator **800** adjusts the charge pump current to compensate for variable VCO gain. The order of the steps in the flowchart **1200** is not limiting as all or some of the steps can be performed simultaneously or in a different order, as will be understood by those skilled in the arts.

In step **1202**, the gain compensator **800** receives the VCO control signals **239** and the capacitor control signals **239**. The VCO control signals **239** determine which VCO **226** is switched-in to the PLL **500**. The capacitor control signals **239** determine which fixed capacitors **232** are switched-in to the LC circuit **228**.

In step **1204**, a gain compensator cell **806** is selected to correspond to the VCO **226** that is switched-in to the PLL **500**, as indicated by the VCO control signals **239**. More specifically, the control signals **239** turn-on the appropriate P-FET **808** for the gain compensator cell **806** that corresponds to the selected VCO **226**.

In step **1206**, the current scaler **804** generates a reference scale current **812** that is based on a PLL control signal **810**, where the PLL control signal **810** defines certain PLL characteristics including reference frequency, loop bandwidth, and damping factor.

In step **1208**, the switches **902** activate one or more groups **904** of unit current sources **906** according to the capacitor control signals **239**. The groups **904** that are activated correspond to the capacitors **232** that are switched-in to the LC circuit **228**, as indicated by the capacitor control signals **239**. The remaining (non-selected) current sources **906** are cutoff.

In step **1210**, the activated groups **904** replicate (or copy) the reference scale current **812** one or more times, where the number of times that the reference scale current **812** is replicated is dependent on the capacitors **232** that are switched-in to the LC circuit **228**. More specifically, the activated groups **904** replicate the reference scale current enough times to sufficiently compensate the variable VCO gain that is caused by the corresponding capacitors **232**.

In step **1212**, the currents from the activated current sources **906** are added together to generate the charge pump reference current **205**.

In step **1214**, the current scaler **804** adjusts the reference scale current **812** to address changing PLL characteristics, such as reference frequency, loop bandwidth, and damping factor. By adjusting the reference scale current **812**, all of the replicated currents in step **1210** are simultaneously adjusted to address the changing PLL characteristics.

5. Other Applications

The gain compensation invention described herein has been discussed in reference to a tuner application. However, the gain compensation invention is not limited to tuners, and is applicable to other non-tuner applications that can benefit from flat PLL gain. Additionally, the gain compensation invention is applicable to other non-PLL circuits that can benefit from compensating for variable VCO gain. The application of the gain compensation invention to these non-PLL circuits will be understood by those skilled in the relevant arts based on the discussions given herein, and are within the scope and spirit of the present invention.

6. CONCLUSION

Example embodiments of the methods, systems, and components of the present invention have been described herein. As noted elsewhere, these example embodiments have been described for illustrative purposes only, and are not limiting. Other embodiments are possible and are covered by the invention. Such other embodiments will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

What is claimed is:

1. A method of compensating the gain of a phase lock loop (PLL), comprising:

receiving a first control signal that is representative of an output frequency of said PLL;

receiving a second control signal that is representative of a characteristic of said PLL; and

generating a charge pump current for said PLL based on said first control signal and said second control signal,

wherein said generating comprises:

selecting a charge pump current value from a plurality of charge pump current values based on said first control signal and said second control signal, and

converting said selected charge pump current value from a digital value to analog.

2. The method of claim 1, wherein said characteristic of said PLL comprises one of a reference frequency, a loop bandwidth, and a damping factor of said PLL.

3. The method of claim 1, wherein said plurality of charge pump current values correspond to different combinations of characteristics of said PLL.

4. The method of claim 1, wherein said plurality of charge pump current values correspond to different output frequencies of said PLL.

5. The method of claim 1, wherein said plurality of charge pump current values comprise multiple sets of charge pump current values, each set of charge pump current values corresponding to a different output frequency of said PLL.

6. The method of claim 5, wherein individual charge pump current values in said set of charge pump current values correspond to different combination of characteristics of said PLL, but have a common output frequency of said PLL.

7. The method of claim 1, wherein said PLL comprises a voltage controlled oscillator (VCO) comprising a number of capacitors forming a resonant circuit for said VCO, and said first control signal represents a capacitor switched into said resonant circuit of said VCO.

8. The method of claim 7, wherein each of said plurality of charge pump current values is determined so as to compensate for variable VCO gain that occurs when a corresponding capacitor of said number of capacitors is switched into said VCO.

9. A circuit that compensates for the gain of a phase lock loop (PLL), comprising:

means for receiving a first control signal that is representative of an output frequency of said PLL;

means for receiving a second control signal that is representative of a characteristic of said PLL; and

means for generating a charge pump current for said PLL based on said first control signal and said second control signal,

wherein said generating means comprises:

means for selecting a charge pump current value from a plurality of charge pump current values based on said first control signal and said second control signal, and

means for converting said selected charge pump current value from a digital value to analog.

10. The circuit of claim 9, wherein said plurality of charge pump current values correspond to different output frequencies of said PLL.

11. The circuit of claim 9, wherein said plurality of charge pump current values comprise multiple sets of charge pump current values, each set of charge pump current values corresponding to a different output frequency of said PLL.

12. The circuit of claim 9, wherein individual charge pump current values in said set of charge pump current values correspond to different combination of characteristics of said PLL, but have a common output frequency of said PLL.

13. The circuit of claim 9, wherein said PLL comprises a voltage controlled oscillator (VCO) comprising a number of capacitors forming a resonant circuit for said VCO, and said first control signal represents a capacitor switched into said resonant circuit of said VCO.

14. The circuit of claim 9, wherein each of said plurality of charge pump current values is determined so as to compensate for variable VCO gain that occurs when a corresponding capacitor of said number of capacitors is switched into said VCO.

15. A gain compensation circuit, comprising:

a memory device; and

a digital-to-analog converter operatively coupled to the memory device,

wherein the memory device receives a first control signal that is representative of an output frequency of a phase lock loop (PLL),

wherein the memory device receives a second control signal that is representative of a characteristic of the PLL,

wherein the memory device selects a charge pump current value from a plurality of charge pump current values based on the first control signal and the second control signal, and

wherein the digital-to-analog converter converts the selected charge pump current value from a digital value into an analog charge pump current.

16. The gain compensation circuit according to claim 15, wherein the plurality of charge pump current values correspond to different output frequencies of the PLL.

17. The circuit gain compensation circuit according to claim 15, wherein the plurality of charge pump current values comprise multiple sets of charge pump current values, each set of charge pump current values corresponding to a different output frequency of the PLL.

18. The circuit gain compensation circuit according to claim 15, wherein individual charge pump current values in the set of charge pump current values correspond to different combinations of characteristics of the PLL, but have a common output frequency of the PLL.

19. The circuit gain compensation circuit according to claim 15,

wherein the PLL comprises a voltage controlled oscillator (VCO) that comprises a number of capacitors forming a resonant circuit for the VCO, and

wherein the first control signal represents a capacitor switched into the resonant circuit of the VCO.

20. The circuit gain compensation circuit according to claim 15, wherein each of the plurality of charge pump current values is determined so as to compensate for variable VCO gain that occurs when a corresponding capacitor of the number of capacitors is switched into the VCO.

* * * * *

Exhibit B



(12) **United States Patent**
Qi et al.

(10) **Patent No.: US 7,400,722 B2**
(45) **Date of Patent: Jul. 15, 2008**

(54) **METHODS AND APPARATUS FOR PERFORMING HASH OPERATIONS IN A CRYPTOGRAPHY ACCELERATOR**

(75) Inventors: **Zheng Qi**, Milpitas, CA (US); **Ronald Squires**, Castro Valley, CA (US); **Mark Buer**, Gilbert, AZ (US); **David K. Chin**, Mountain View, CA (US)

(73) Assignee: **Broadcom Corporation**, Irvine, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 706 days.

(21) Appl. No.: **10/330,694**

(22) Filed: **Dec. 24, 2002**

(65) **Prior Publication Data**

US 2003/0185391 A1 Oct. 2, 2003

Related U.S. Application Data

(60) Provisional application No. 60/368,583, filed on Mar. 28, 2002.

(51) **Int. Cl.**
H04L 9/00 (2006.01)

(52) **U.S. Cl.** **380/28**; 380/44; 713/189; 713/193

(58) **Field of Classification Search** 713/151, 713/153, 160–161, 168, 170, 189, 193; 726/2, 726/14; 380/28, 30, 44; 705/67
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,265,164	A *	11/1993	Matyas et al.	380/30
5,949,881	A *	9/1999	Davis	713/189
6,694,436	B1 *	2/2004	Audebert	726/9
6,983,366	B1 *	1/2006	Huynh et al.	713/168
7,024,695	B1 *	4/2006	Kumar et al.	726/26
2002/0097724	A1 *	7/2002	Halme et al.	370/392
2002/0191791	A1 *	12/2002	Anand	380/255
2003/0058274	A1 *	3/2003	Hill et al.	345/751
2003/0084309	A1 *	5/2003	Kohn	713/189

FOREIGN PATENT DOCUMENTS

WO	WO 01/61912 A1	8/2001

OTHER PUBLICATIONS

Deepakumara et al., FPGA Implementation of MD5 Hash Algorithm. Electrical and Computer Engineering, 2001. Canadian Conference on vol. 2, May 13-16, 2001, pp. 919-924.*
Jiang et al., Securing Web Servers Against Insider Attack, Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th annual, Dec. 10-14, 2001, pp. 265-276.*
Thomas A Stephen, SSL and TLS Essentials Securing the Web, 2000. pp. 4-6, 41-45, 96-104.*

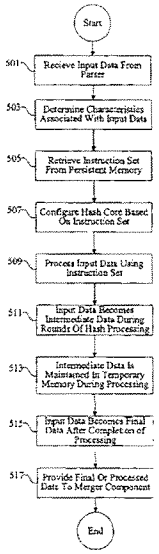
* cited by examiner

Primary Examiner—Minh Dieu Nguyen
(74) *Attorney, Agent, or Firm*—Sterne, Kessler, Goldstein & Fox P.L.L.C

(57) **ABSTRACT**

Methods and apparatus are provided for implementing a cryptography accelerator for performing operations such as hash operations. The cryptography accelerator recognizes characteristics associated with input data and retrieves an instruction set for processing the input data. The instruction set is used to configure or control components such as MD5 and SHA-1 hash cores, XOR components, memory, etc. By providing a cryptography accelerator with access to multiple instruction sets, a variety of hash operations can be performed in a configurable cryptographic accelerator.

32 Claims, 10 Drawing Sheets



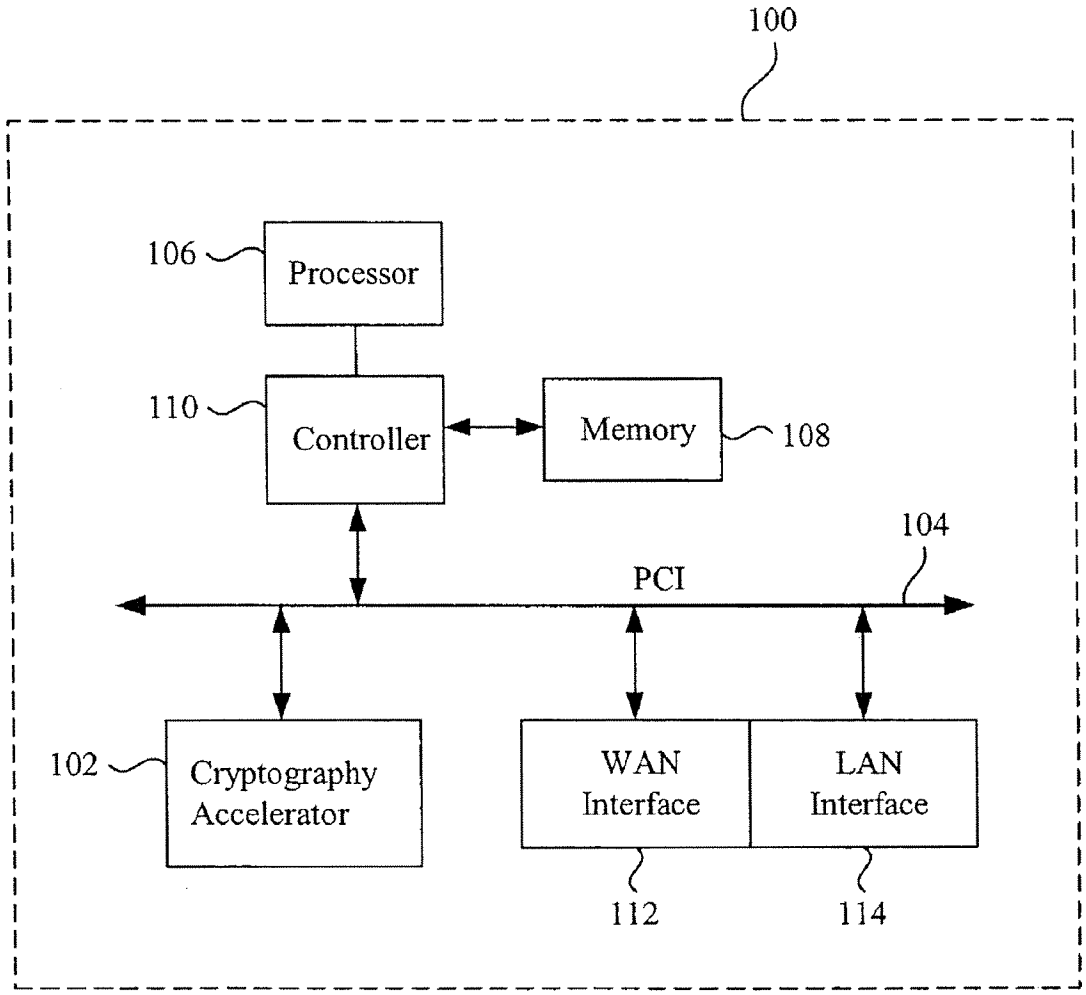

100
106
Processor
110
Controller
Memory
108
104
PCI
102
Cryptography Accelerator
WAN Interface
LAN Interface
112
114

Figure 1

Figure 2A

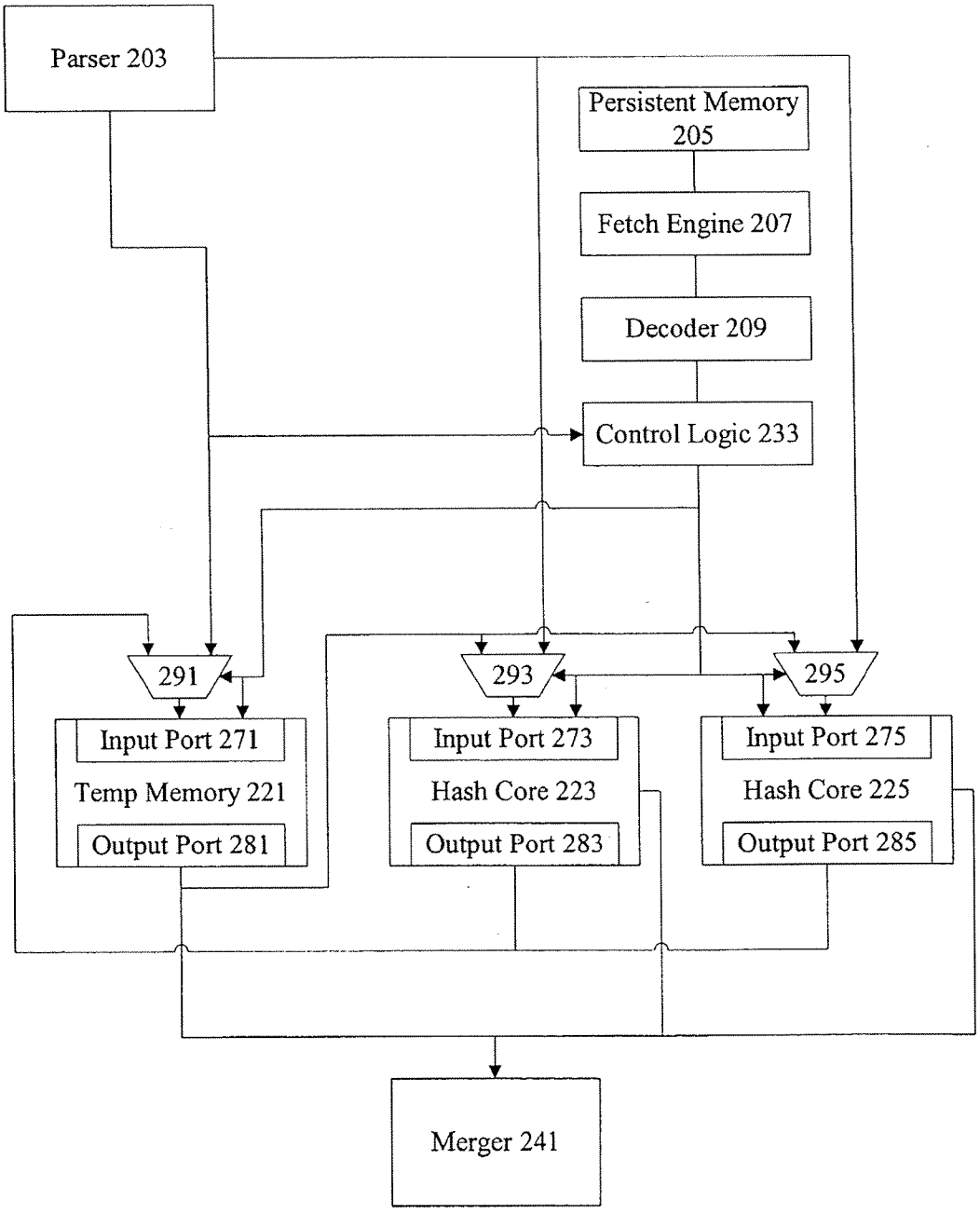

Parser 203
Persistent Memory 205
Fetch Engine 207
Decoder 209
Control Logic 233
291
293
295
Input Port 271
Temp Memory 221
Output Port 281
Input Port 273
Hash Core 223
Output Port 283
Input Port 275
Hash Core 225
Output Port 285
Merger 241

Figure 2B

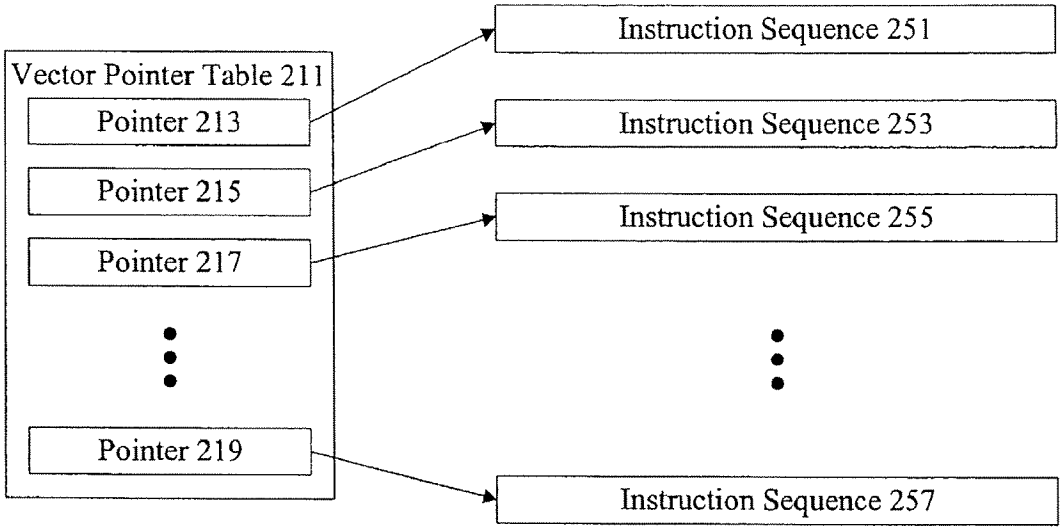


Figure 2C

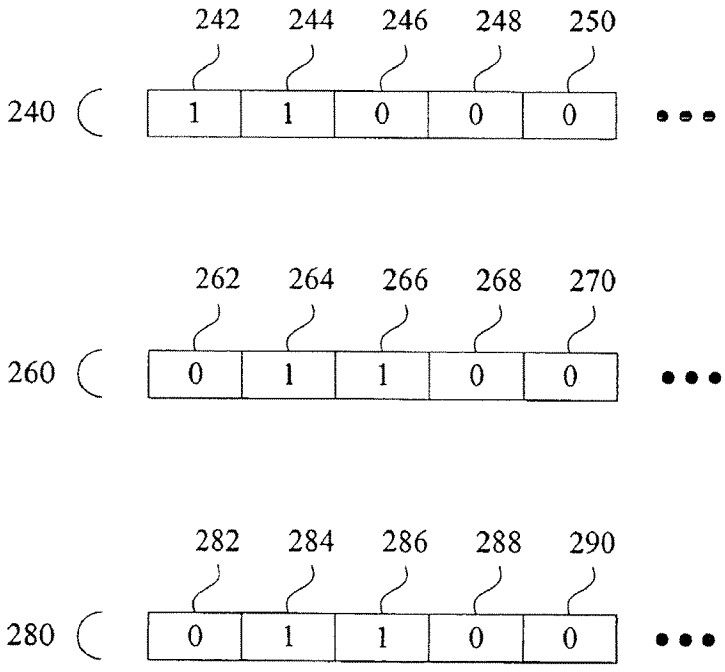
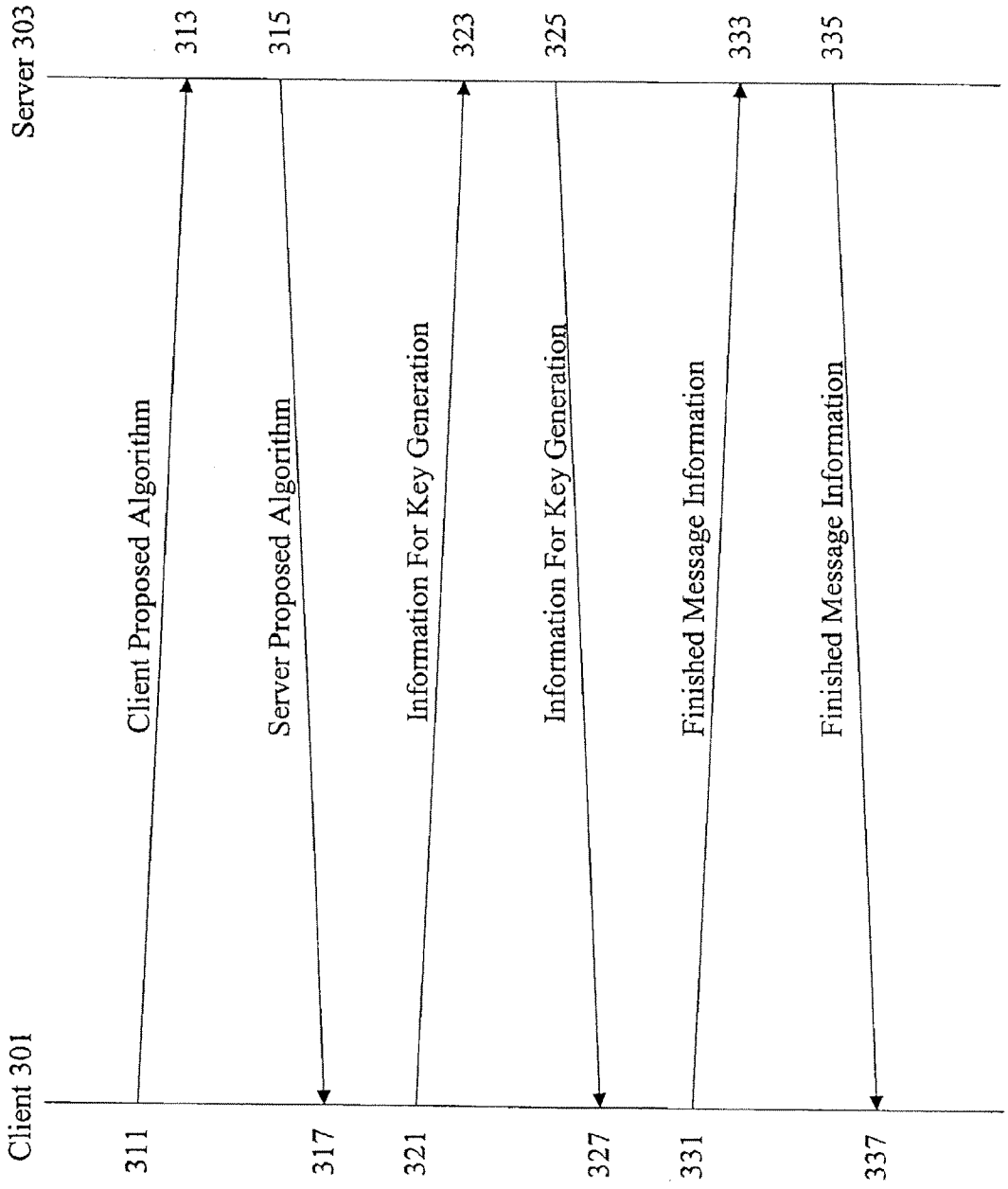


Figure 3


Server 303
Client 301
313
315
323
325
333
335
311
317
321
327
331
337
Client Proposed Algorithm
Server Proposed Algorithm
Information For Key Generation
Information For Key Generation
Finished Message Information
Finished Message Information

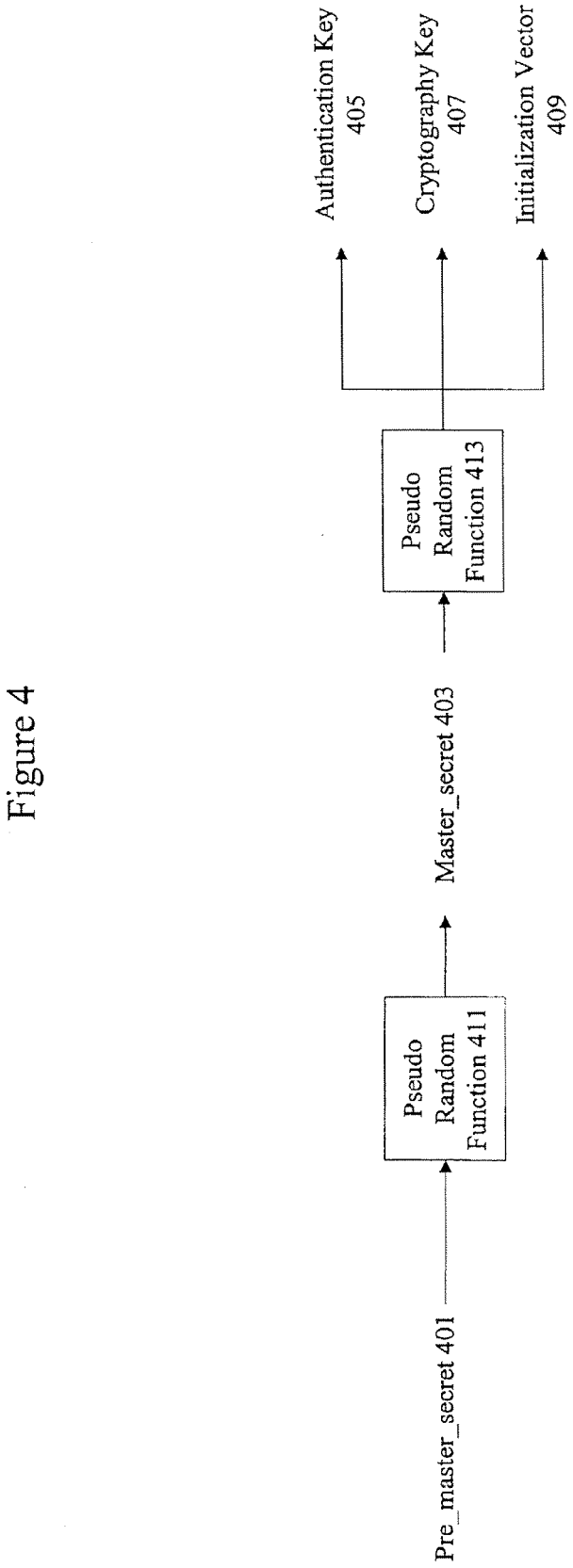

Pre_master_secret 401
Pseudo Random Function 411
Master_secret 403
Pseudo Random Function 413
Authentication Key 405
Cryptography Key 407
Initialization Vector 409

Figure 4

Figure 5

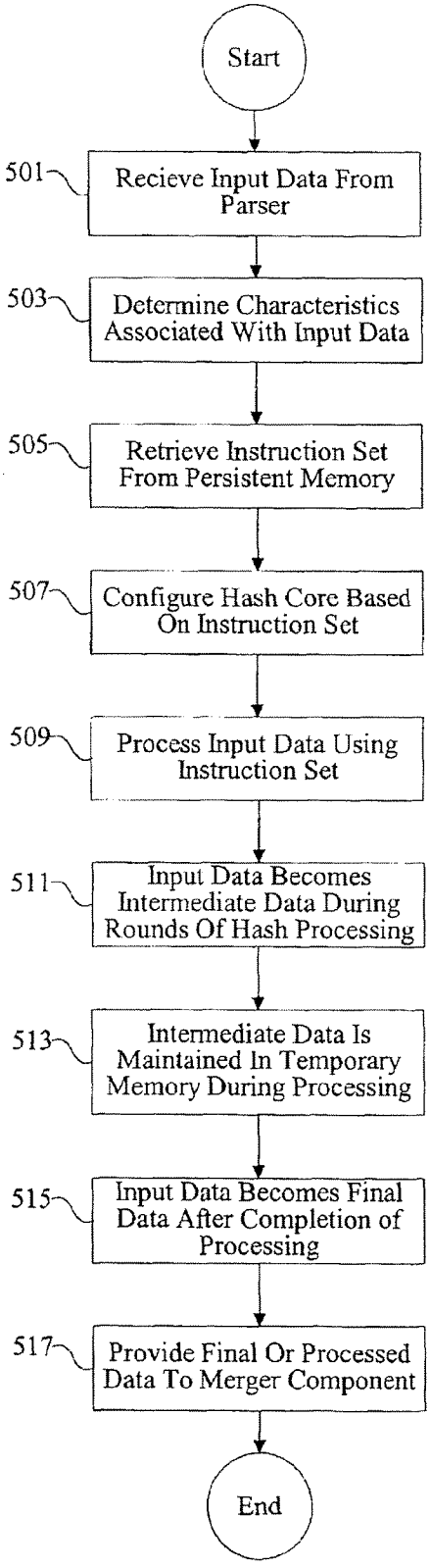

Start
501
Recieve Input Data From Parser
503
Determine Characteristics Associated With Input Data
505
Retrieve Instruction Set From Persistent Memory
507
Configure Hash Core Based On Instruction Set
509
Process Input Data Using Instruction Set
511
Input Data Becomes Intermediate Data During Rounds Of Hash Processing
513
Intermediate Data Is Maintained In Temporary Memory During Processing
515
Input Data Becomes Final Data After Completion of Processing
517
Provide Final Or Processed Data To Merger Component
End

Figure 6

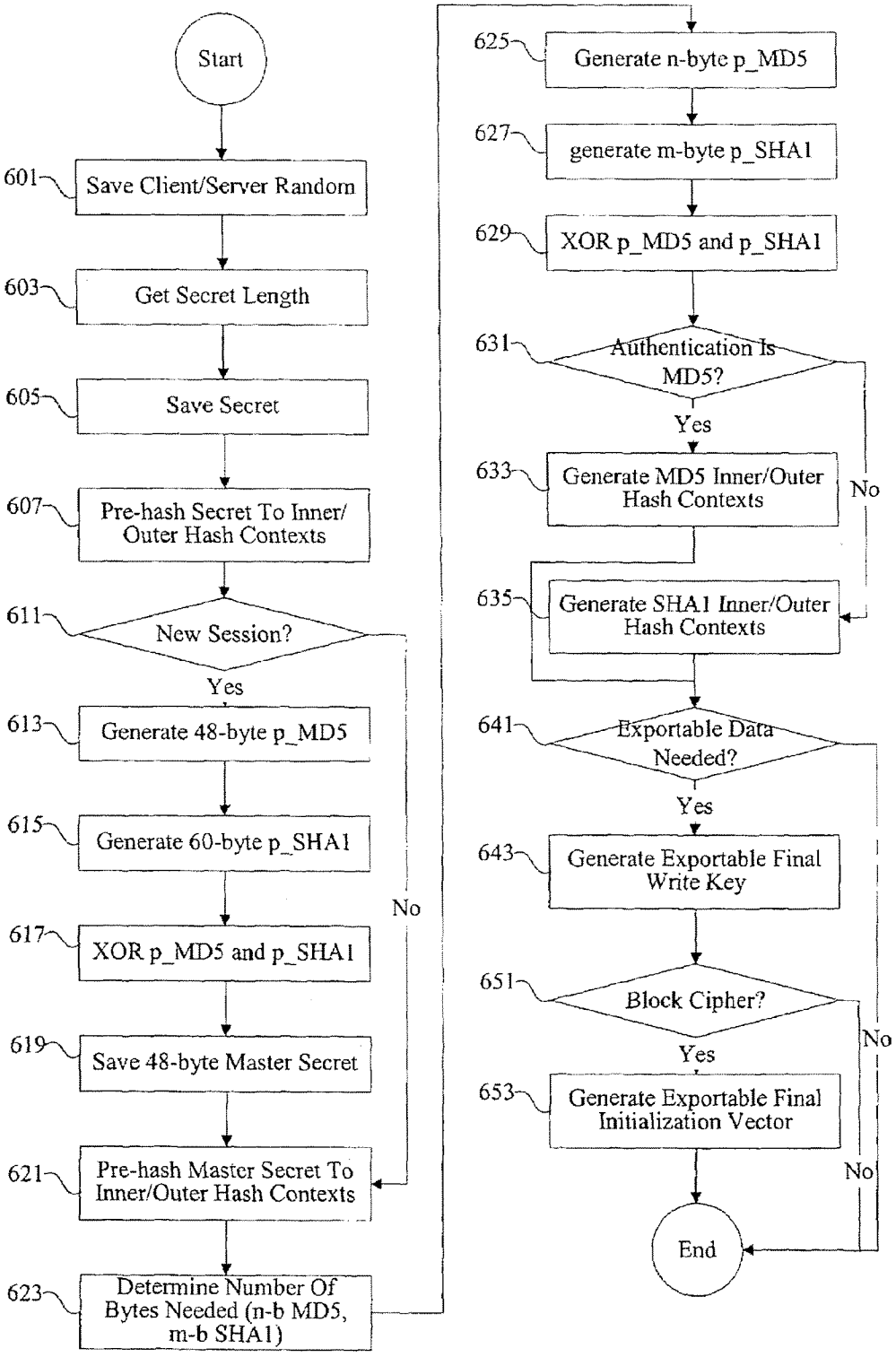

Start
601
Save Client/Server Random
603
Get Secret Length
605
Save Secret
607
Pre-hash Secret To Inner/ Outer Hash Contexts
611
New Session?
Yes
No
613
Generate 48-byte p_MD5
615
Generate 60-byte p_SHA1
617
XOR p_MD5 and p_SHA1
619
Save 48-byte Master Secret
621
Pre-hash Master Secret To Inner/Outer Hash Contexts
623
Determine Number Of Bytes Needed (n-b MD5, m-b SHA1)
625
Generate n-byte p_MD5
627
generate m-byte p_SHA1
629
XOR p_MD5 and p_SHA1
631
Authentication Is MD5?
Yes
No
633
Generate MD5 Inner/Outer Hash Contexts
635
Generate SHA1 Inner/Outer Hash Contexts
641
Exportable Data Needed?
Yes
No
643
Generate Exportable Final Write Key
651
Block Cipher?
Yes
No
653
Generate Exportable Final Initialization Vector
No
End

Figure 7

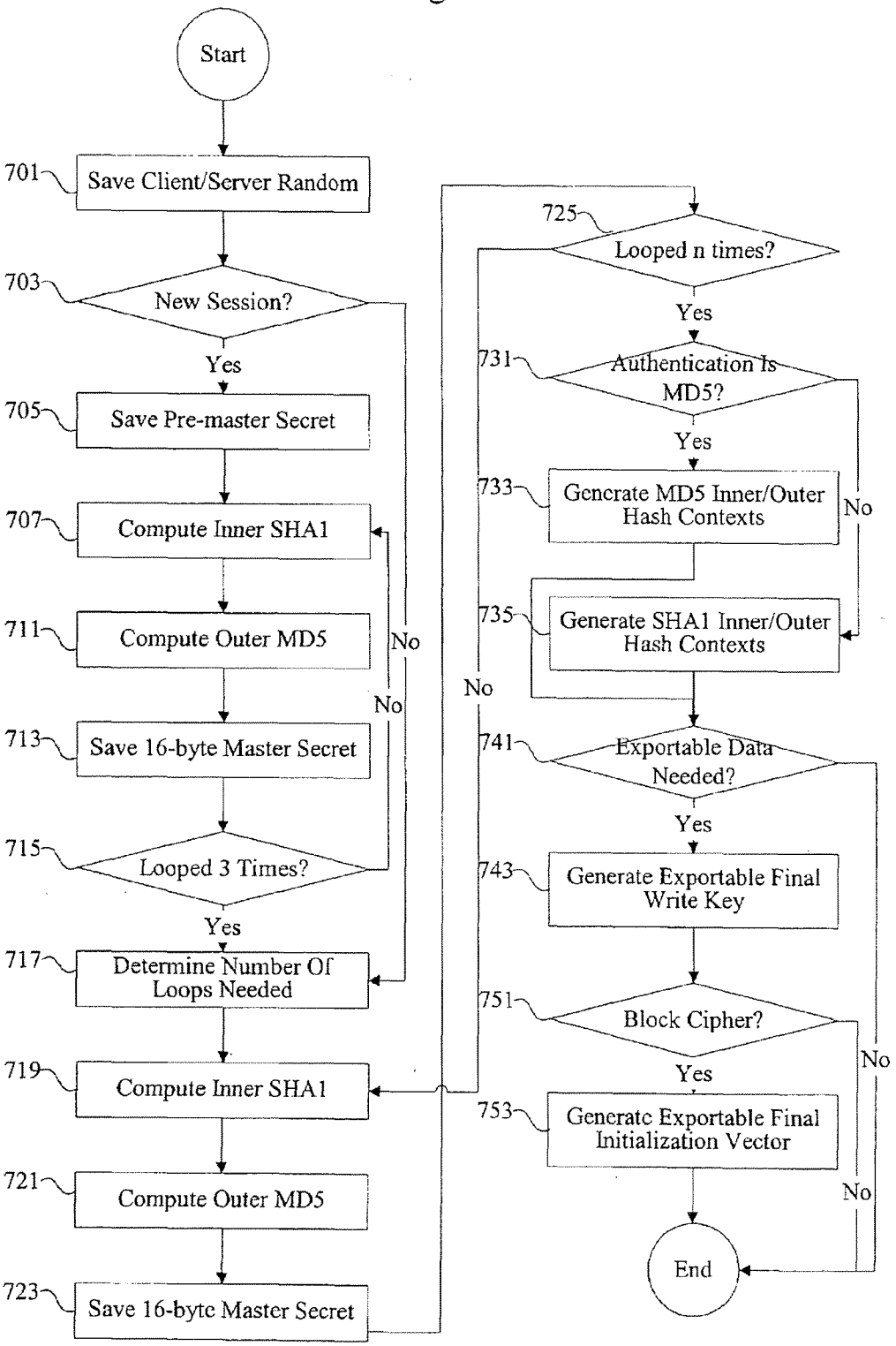

Start
701
Save Client/Server Random
703
New Session?
Yes
No
705
Save Pre-master Secret
707
Compute Inner SHA1
711
Compute Outer MD5
713
Save 16-byte Master Secret
715
Looped 3 Times?
Yes
No
717
Determine Number Of Loops Needed
719
Compute Inner SHA1
721
Compute Outer MD5
723
Save 16-byte Master Secret
725
Looped n times?
Yes
No
731
Authentication Is MD5?
Yes
No
733
Generate MD5 Inner/Outer Hash Contexts
735
Generate SHA1 Inner/Outer Hash Contexts
741
Exportable Data Needed?
Yes
No
743
Generate Exportable Final Write Key
751
Block Cipher?
Yes
No
753
Generate Exportable Final Initialization Vector
End

Figure 8

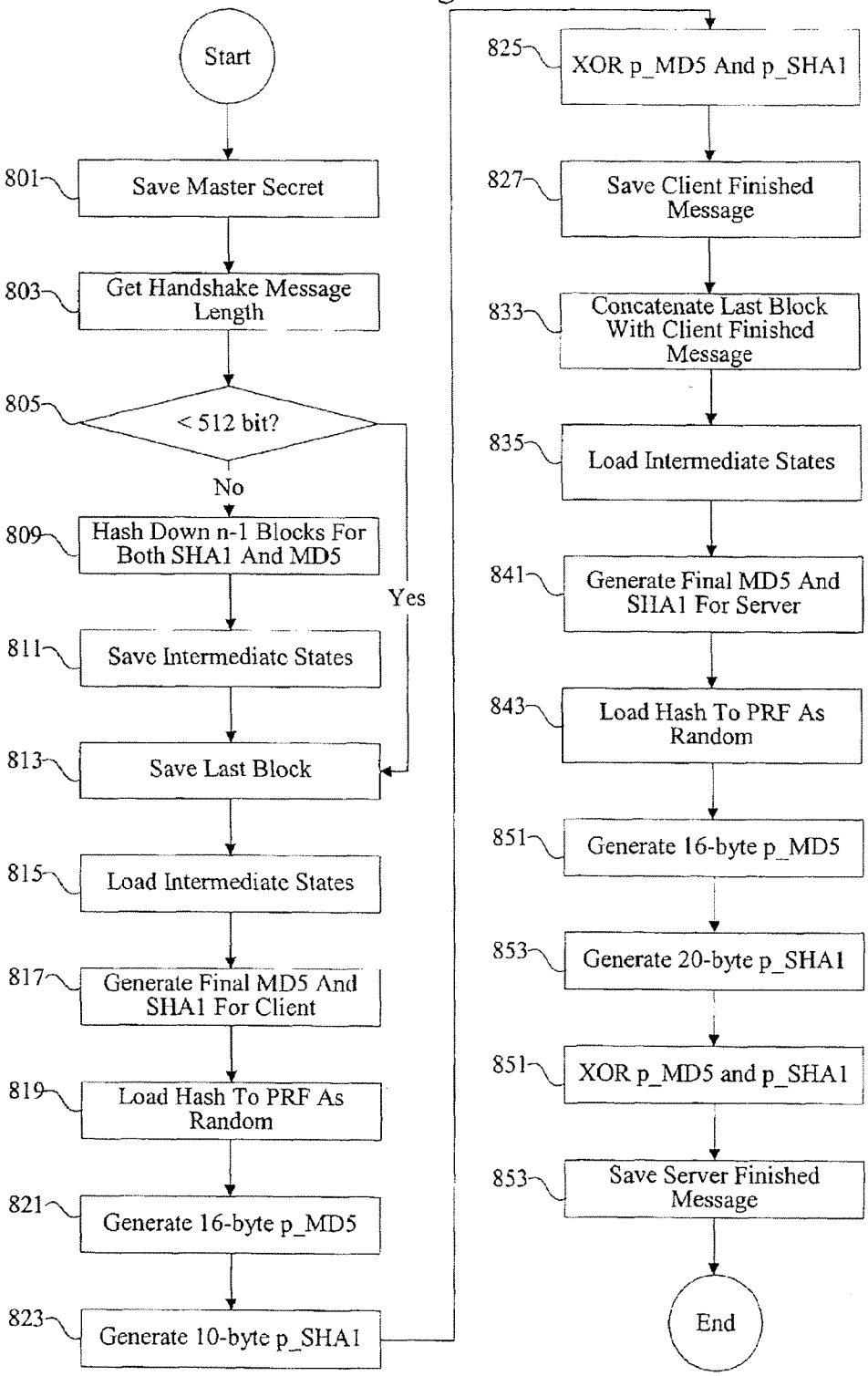
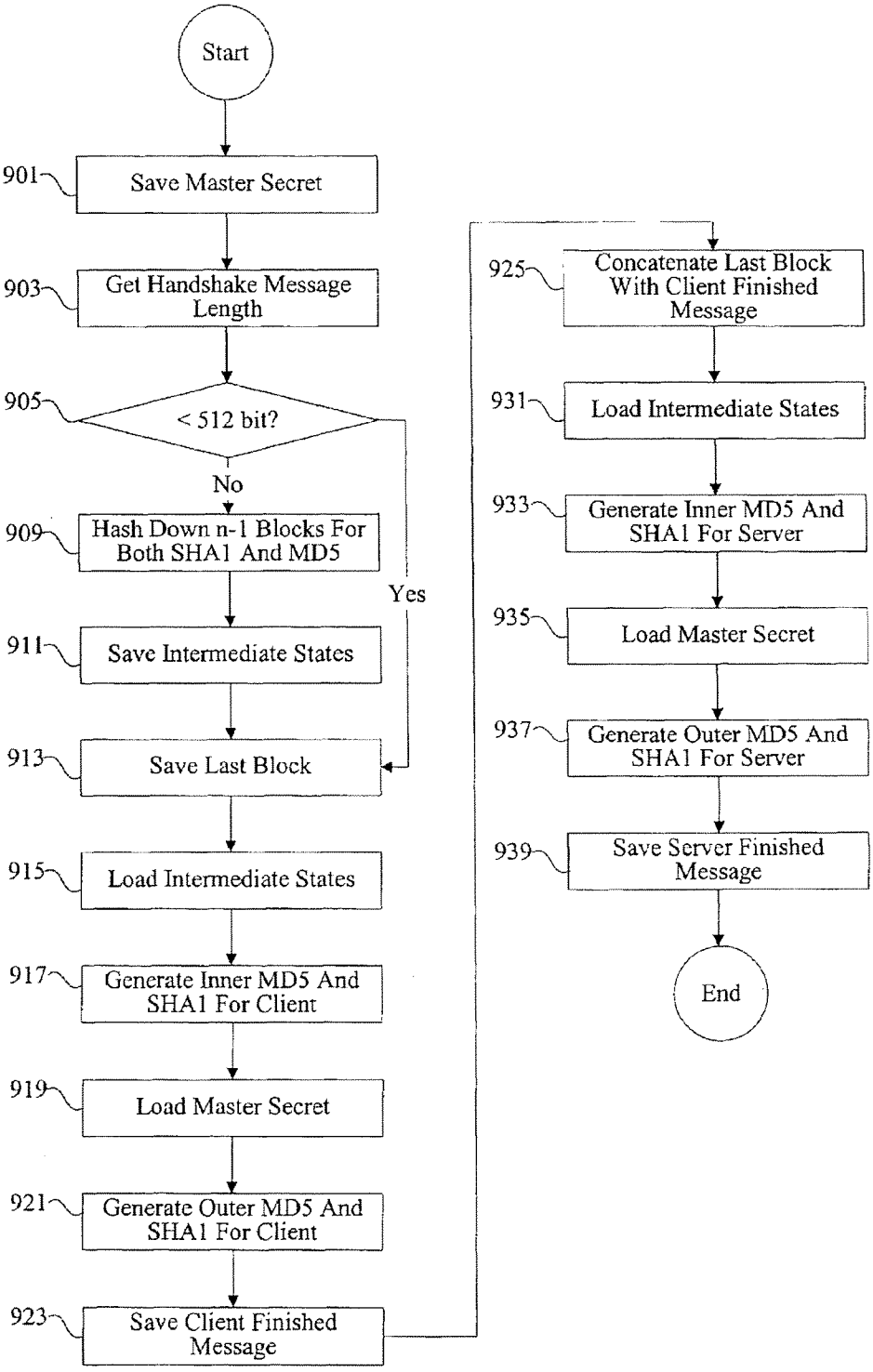

Start
801
Save Master Secret
803
Get Handshake Message Length
805
< 512 bit?
No
Yes
809
Hash Down n-1 Blocks For Both SHA1 And MD5
811
Save Intermediate States
813
Save Last Block
815
Load Intermediate States
817
Generate Final MD5 And SHA1 For Client
819
Load Hash To PRF As Random
821
Generate 16-byte p_MD5
823
Generate 10-byte p_SHA1
825
XOR p_MD5 And p_SHA1
827
Save Client Finished Message
833
Concatenate Last Block With Client Finished Message
835
Load Intermediate States
841
Generate Final MD5 And SHA1 For Server
843
Load Hash To PRF As Random
851
Generate 16-byte p_MD5
853
Generate 20-byte p_SHA1
851
XOR p_MD5 and p_SHA1
853
Save Server Finished Message
End

Figure 9


Start
901
Save Master Secret
903
Get Handshake Message Length
905
< 512 bit?
No
Yes
909
Hash Down n-1 Blocks For Both SHA1 And MD5
911
Save Intermediate States
913
Save Last Block
915
Load Intermediate States
917
Generate Inner MD5 And SHA1 For Client
919
Load Master Secret
921
Generate Outer MD5 And SHA1 For Client
923
Save Client Finished Message
925
Concatenate Last Block With Client Finished Message
931
Load Intermediate States
933
Generate Inner MD5 And SHA1 For Server
935
Load Master Secret
937
Generate Outer MD5 And SHA1 For Server
939
Save Server Finished Message
End

METHODS AND APPARATUS FOR PERFORMING HASH OPERATIONS IN A CRYPTOGRAPHY ACCELERATOR

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims priority under U.S.C. 119(e) from U.S. Provisional Application No. 60/368,583, entitled "Methods And Apparatus For Implementing A Configurable Authentication Accelerator," as of filing on Mar. 28, 2002, the disclosure of which is herein incorporated by reference for all purposes.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present application relates to implementing a cryptography accelerator. More specifically, the present application relates to methods and apparatus for providing a configurable cryptography accelerator with instruction sets for performing hash operations on input data.

2. Description of Related Art

Conventional software and hardware designs for performing hash operations are inefficient. One technique for securing a communication channel between two network entities such as a client and a server specifies that the two entities perform a cryptography handshake sequence. During the cryptographic handshake sequence, the two network entities will typically perform various cryptographic operations such as encryption and authentication operations to verify the identity of the other and to exchange information to establish a secure channel.

In one example, session keys are exchanged after the identity of the other network entity is verified. However, both software, firmware and hardware techniques for performing hash operations, such as hash operations used in cryptography handshake sequences, have been inefficient and resource intensive. Cryptography handshake sequences and hash algorithms are described in Applied Cryptography, Bruce Schneier, John Wiley & Sons, Inc. (ISBN 0471128457), incorporated by reference in its entirety for all purposes.

It is therefore desirable to provide methods and apparatus for improving hash operations with respect to some or all of the performance limitations noted above.

SUMMARY OF THE INVENTION

Methods and apparatus are provided for implementing a cryptography accelerator for performing operations such as hash operations. The cryptography accelerator recognizes characteristics associated with input data and retrieves an instruction set for processing the input data. The instruction set is used to configure or control components such as MD5 and SHA-1 hash cores, XOR components, memory, etc. By providing a cryptography accelerator with access to multiple instruction sets, a variety of hash operations can be performed in a configurable cryptographic accelerator.

According to various embodiments, a cryptography accelerator for performing hash operations is provided. The accelerator includes a first hash core, a persistent memory, and a temporary memory. The first hash core is operable to perform a plurality of rounds of hash computations on input data to derive processed data. A persistent memory contains a plurality of instruction sets. The plurality of instruction sets provide information for the first hash core on operations to perform on input data and intermediate data during the plurality of rounds of hash computations. The temporary memory is coupled to the first hash core. The temporary memory is operable to hold input data and intermediate data.

According to other embodiments, a method for performing hash operations is provided. Input data is received. Characteristics associated with the input data are determined. An instruction set for performing hash operations on input data is selected. The instruction set is selected from a plurality of instruction sets maintained in persistent memory associated with a first hash core. The first hash core is configured using the instruction set. The first hash core is operable to perform hash operations on input data based on the instruction set.

These and other features and advantages of the present invention will be presented in more detail in the following specification of the invention and the accompanying figures, which illustrate by way of example the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may best be understood by reference to the following description taken in conjunction with the accompanying drawings, which are illustrative of specific embodiments of the present invention.

FIG. **1** is a diagrammatic representation of a system that can use the techniques of the present invention.

FIG. **2**A is a diagrammatic representation of an integrated circuit containing a processing core for performing hash operations.

FIG. **2**B is a diagrammatic representation showing a structure referencing instruction sequences.

FIG. **2**C is a diagrammatic representation of mechanisms for determining resource dependencies.

FIG. **3** is an interaction diagram showing a sequence in which the techniques of the present invention can be applied.

FIG. **4** is a diagrammatic representation showing input data, intermediate data, and processed data.

FIG. **5** is a flow process diagram showing techniques for selecting an instruction set.

FIG. **6** is a flow process diagram showing TLS 1.0 key derivation.

FIG. **7** is a flow process diagram showing SSLv3 key derivation.

FIG. **8** is a flow process diagram showing TLS 1.0 finished message generation.

FIG. **9** is a flow process diagram showing SSLv3 finished message generation.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

The present application relates to implementing a cryptography accelerator. More specifically, the present application relates to methods and apparatus for providing a cryptography accelerator capable of performing a variety of different hash operations on input data.

Reference will now be made in detail to some specific embodiments of the invention including the best modes contemplated by the inventors for carrying out the invention. Examples of these specific embodiments are illustrated in the accompanying drawings. While the invention is described in conjunction with these specific embodiments, it will be understood that it is not intended to limit the invention to the described embodiments. On the contrary, it is intended to cover alternatives, modifications, and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims.

For example, the techniques of the present invention will be described in the context of the SHA-1 and MD5 hash algorithms. However, it should be noted that the techniques of the present invention can be applied to a variety of different hash operations for cryptography processing in general. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. The present invention may be practiced without some or all of these specific details. In other instances, well known process operations have not been described in detail in order not to unnecessarily obscure the present invention.

A wide variety of algorithsm are used for encryption and authentication operations. In many conventional implementations, software is used to identify the type of data and the cryptographic processing needed for the particular data sequence. However, cryptographic operations implemented entirely in software on a generic processor such as a reduced instruction set (RISC) or complex instruction set (CISC) processors are highly inefficient. In many environments, it is beneficial to use specialized accelerators for performing cryptographic operations, such as DES and SHA-1 operations. In typical cryptography accelerator implementations, a cryptography accelerator is configured to perform resource intensive cryptographic operations while software through an external host is configured to perform sequencing. That is, software formats and sequences data and makes function calls to elementary cryptographic operators. In one example, a cryptography accelerator would be responsible for executing a function such as cryptooperation (data, key1, key2) while the software would be responsible for formatting the data properly, acquiring the keys, and making multiple calls to the function when necessary.

More recent efforts have focused on implementing both core processing as well as formatting and sequencing on a cryptography accelerator. In one example, software running on a host such as a CPU external to a cryptography accelerator could simply forward a packet to the cryptography accelerator. Using the packet, the cryptography accelerator would extract information to determine what type of processing and how many rounds of processing need to be performed.

One technique for implementing such a cryptography accelerator that performs both cryptography processing and sequencing uses state tables. Each load or store instruction on the cryptography accelerator is represented by one or more states. However, because many variations in cryptographic algorithms exist, a large number of states exist. Having a significant number of states makes implementation and verification extremely difficult. Furthermore, if a new cryptographic algorithm is developed, substantial work would have to be performed in order to update the states associated with the instructions.

Consequently, the techniques of the present invention provide sequences of instructions for performing cryptographic as well as sequencing operations on data. Instruction sequences can relatively easily be implemented for particular cryptographic operations. When a new algorithm is developed, an additional instruction sequence can be provided on the cryptography accelerator. The variations between cryptographic algorithms can be handled with relative ease. The techniques and mechanisms of the present invention allow for a cryptographic accelerator that has the speed and processing advantages of a customized piece of hardware while retaining the flexibility of a piece of software.

FIG. **1** is a diagrammatic representation of one example of a processing system **100** with a cryptography accelerator according to various embodiments of the present invention. As shown in FIG. **1**, the present invention may be implemented in a stand-alone cryptography accelerator **102** or as part of the system **100**. In the described embodiment, the cryptography accelerator **102** is connected to a bus **104** such as a PCI bus via a standard on-chip PCI interface. The processing system **100** includes a processing unit **106** and a system memory unit **108**. The processing unit **106** and the system memory unit **108** are coupled to the system bus **104** via a bridge and memory controller **110**.

According to various embodiments, the processing unit **106** may be the central processing unit (CPU) of a system **100**. In one example, a LAN interface **114** is provided to couple the processing system **100** to a local area network (LAN) to allow packet receipt and transmission. Similarly, a Wide Area Network (WAN) interface **112** can also be provided to connect the processing system to a WAN (not shown) such as the Internet. The WAN interface manages in-bound and out-bound packets, providing automatic cryptographic processing for IP packets.

In many implementations, the cryptography accelerator **102** is an application specific integrated circuit (ASIC) coupled to the processor **106**. However, the cryptography accelerator **102** can also be a programmable logic device (PLD), field programmable gate array (FPGA), or other device coupled to the processor **106**. According to specific embodiments, the cryptography accelerator **102** is implemented either on a card connected to the bus **104** or as a standalone chip integrated in the system **100**.

In other embodiments, the cryptography accelerator **102** itself is integrated into the processing core of a CPU of system **100**, such as that available from Tensilica Corporation of Santa Clara, Calif. or ARC Cores of San Jose, Calif. In another embodiment, techniques and mechanisms of the present invention are integrated into a CPU such as a CPU available from Intel Corporation of San Jose, Calif. or AMD Corporation of Sunnyvale, Calif. By implementing cryptography accelerator functionality entirely on the processor **106**, a separate card or chip in the system **100** is not needed. In still other embodiments, the processing system **100** including the cryptography accelerator **102** is implemented as a system on a chip (SOC). The network interfaces, memory, processing core, and cryptography accelerator functionality are provided on a single integrated circuit device.

The cryptography accelerator **102** is capable of implementing various network security standards, such as Internet Protocol Security (IPSec), Secure Sockets Layer/Transport Layer Security (SSL/TLS), Internet Key Exchange (IKE) which provide application-transparent encryption and authentication services for network traffic.

Network security standards such as IPsec and SSL/TLS provide authentication through the use of hash algorithms. Two commonly used hash algorithms are MD5 and the Secure Hash algorithm (SHA-1). Other hash algorithms such as MD4 and MD2 are also available. Hash algorithms are described in Applied Cryptography, Bruce Schneier, John Wiley & Sons, Inc. (ISBN 0471128457), incorporated by reference in its entirety for all purposes. Even though many network security standards apply the same hash algorithms, different approaches are taken toward applying the hash algorithms to the actual authentication computation.

Different versions of the same network security standards even vary approaches toward applying the hash algorithms. In IPsec, several approaches such as HMAC-MD5-96 and HMAC-SHA1-96 based on the hash message authentication code (HMAC) algorithm are provided. The approaches HMAC-MD5-96 and HMAC-SHA1-96 are described in RFC 2403 and RFC 2404 respectively, while the HMAC algorithm is described in RFC 2104, the entireties of which are incor-

porated by reference for all purposes. SSL/TLS use similar, but slightly different approaches. In SSLv3, an earlier version of HMAC is used. In TLS 1.0, the same version of HMAC is used as in IPsec, but a different number of bits are taken for the full result.

The TLS 1.0 protocol is described in RFC 2246, the entirety of which is incorporated by reference for all purposes. SSL is described in E. Rescorla, SSL and TLS: Designing and Building Secure Systems (Addison-Wesley, 2001) and S. A. Thomas, SSL & TLS Essentials: Securing the Web (John Wiley & Sons, Inc. 2000), the entireties of which are incorporated by reference for all purposes. In addition, SSL/TLS define a set of functions using a combination of HMAC, MD5, and SHA1 to generate processed data. For example, combinations are used to generate a master secret sequence from a premaster secret sequence, to generate key blocks from a master secret sequence, or to perform hash operations for finished message processing and client certificate verification.

Typical cryptography accelerators use hash cores for performing hash operations. When a client or server participates in an authentication sequence such as a key exchange, clients and servers need cryptography accelerators specifically configured for particular versions of specified network security standards. In one example, if the server needs to perform TLS 1.0 operations, a cryptography accelerator such as an ASIC specifically microcoded with a TLS 1.0 instruction set would be required. A state machine can be used to perform operations associated with each network security standard version. As noted above, however, a state machine that can handle the number of standards in existence would be extremely complicated and difficult to implement.

Consequently, many cryptography accelerators typically contain only functionality for performing basic hash operations such as MD5 or SHA1 operations. Authentication specific functionality on a cryptography accelerator is often limited to MD5 or SHA1 hash cores. The external processor such as an external CPU would pass data to a cryptography accelerator when MD5 or SHA1 processing was needed. In one example, if a network security standard specified repeated calls to a MD5 or SHA1 function, the external processor would pass data to the cryptography accelerator during each function call, receive data output by the cryptography accelerator, and alter data as needed before passing the data back to the cryptography accelerator for another function call.

Typically, only a single hash function call would be performed on data before sending the data back to an external processor. In another example, if XOR operations were specified for data output from the MD5 and SHA1 cores, the external processor would perform the XOR operations even if the XOR operations were a specific part of the cryptographic processing. Because of inefficiencies such as the passing of data between the external processor and the cryptography accelerator between function calls, cryptographic processing for a server or client expecting many different versions of network security protocols has been limited.

The techniques of the present invention, however, provide not only for a cryptography accelerator specifically configured for a particular type of hash operations without the need to send and receive data to an external processor between various calls to a particular function implemented on a chip, the techniques of the present invention provide an automatically configurable cryptographic accelerator that recognizes characteristics of the input data and automatically performs cryptographic processing such as SSLv3 or TLS 1.0 key derivation.

It should be noted that recognizing characteristics of the input data can include operations such as analyzing the input data, retrieving information associated with the input data, or recognizing characteristics of instruction sequences associated with the input data. A single cryptographic accelerator, for example, with an MD5 and a SHA1 core can perform cryptographic processing associated with a variety of operations using the MD5 and SHA1 hash operations. In one example, the cryptography accelerator can perform cryptographic operations associated with IPsec and SSL/TLS processing.

FIG. **2A** is a diagrammatic representation of one example of a cryptography accelerator according to various embodiments. The cryptography accelerator includes an interface having a parser **203** coupled to an entity such as external processor for receiving and delineating input data sequences. In one example, the parser **203** receives a data sequence associated with SSLv3 key derivation. The control logic **233** determines that key derivation operations associated with SSLv3 should be performed on the data sequence. The control logic **233** retrieves an instruction set associated with SSLv3 key derivation from persistent memory **205**. Memory that retains data after hash operations are completed is referred to herein as persistent memory. Persistent memory also typically remains intact when power is disconnected. In one embodiment, persistent memory is a read-only memory (ROM) on a cryptography accelerator chip, although persistent memory can also be components such as flash memory. In another embodiment, persistent memory **205** and temporary memory **221** are contained in the same component. A component such as a random access memory (RAM) can be loaded with instruction sets and can provide the capability to function as both a persistent memory and as a temporary memory, although such access may be slow.

According to various embodiments, persistent memory **205** includes a table with various types of operations and security protocols identified in the entries. The entries correspond to instruction sets for configuring the cryptography accelerator. Logic and mechanisms for configuring a cryptography accelerator for performing a particular type of cryptographic operation such as key derivation or finished message processing is referred to herein as an instruction set. The fetch engine **207** retrieves the instruction set from persistent memory **205**. According to various embodiments, the decoder **209** receives and interprets the instruction set for control logic **233**. In one embodiment, control logic **233** retrieves microcode for performing cryptographic operations on an input data sequence. Logic and mechanisms for configuring or managing components such as hash cores for authentication processing is herein referred to as control logic. In one example, control logic manages cryptographic processing in components such as hash core **223**, hash core **225**, and temporary memory **221**.

In one embodiment, hash cores **223** and **225** as well as temporary memory **221** also receive input data from parser **203**. After a round of processing in hash core **223** or hash core **225**, data can be provided to temporary memory **221**. Input data that has undergone one or more rounds of hash operations is referred to herein as intermediate data. Temporary memory **221** can store the intermediate data and subsequently provide the intermediate data for additional rounds of hash processing through output port **281** to the input ports **273** and **275** associated with hash cores **223** and **225**. According to various embodiments, hash cores **223** and **225** both are capable of performing either MD5 or SHA-1 processing. After the specified number of rounds of hash processing have occurred as determined by the eontrol logic **233**, hash cores

223 and 225 can provide the final or processed data through output port 283 and 285 to merger component 241. Merger component 241 can then send the processed data to the external entity.

According to various embodiments, components for performing other operation such as XOR operations are also included in the cryptography accelerator. In one example, the XOR component is coupled to the output ports 283 and 285 so that SHA-1 and MD5 processed data can be combined together. It should be noted that the cryptography accelerator can include a number of other components including cryptography blocks such as DES, triple DES, and RC4 cores. The cryptography accelerator can include encryption functionality, central processing cores, bypass circuitry, etc.

FIG. 2B is a diagrammatic representation providing one example on an instruction sequence is provided to control logic 233. In one example, a parser loads a program counter with a pointer from a vector pointer table 211. Each pointer 213, 215, 217, and 219 may be configured to refer to an instruction sequence 251, 253, 255, and 257. According to various embodiments, each instruction sequence is a sequence of loads, stores, moves, sets, etc., for performing cryptographic operations. In one example, the fetch engine gets the sequence of instructions from persistent memory as long as there is room in an instruction queue. The instructions are decoded in order to determine resource dependencies to allow instructions to be executed out of order. According to various embodiments, several hash engines are provided in a cryptographic accelerator and instructions are performed as resources become available. Consequently, mechanisms are provided to track the resource dependencies. In some examples, resources include memory ports, hash engine ports, and counters.

FIG. 2C is a diagrammatic representation of one example of a mechanism for tracking resource dependencies. According to various embodiments, global resource vector 240 indicates which resources are being used. In one example, resource 242 represents a memory input port being used and resource 244 represents a hash engine input port that is in use. Dependency vector 260 shows which resources are needed for a particular instruction in an instruction sequence. In one example, resource 264 and 266 represent the hash engine input port and the hash engine output port are needed for the instruction to execute. Consequently, the instruction may not execute until the global resource vector returns to a state shown in vector 280, when resource 284 and 286 representing the hash engine input and output ports become available.

FIG. 3 shows one example of a cryptographic handshake sequence between a client 301 and a server 303. A wide variety of cryptographic handshake sequences associated with key exchanges are available. FIG. 3 is merely one example of a handshake. At 311, the client 301 transmits a message with a security enable parameter to a server 303. The authentication message contains an identifier such as a user name or an authentication identifier that allows the receiver to select an authentication mechanism out of a possible set of mechanisms. According to various embodiments, server 303 already has information associated with the client. The server 303 identifies the security enable parameter along with any client proposed algorithms and transmits an acknowledgement at 315 to client 301 indicating the selection of an algorithm.

As noted above, a client 301 transmits a user name to a server 303 and a server 303 at 315 transmits a value such as a salt associated with the user name back to the client 301. According to other embodiments, protocol version, session ID, cipher suite, and compression method are exchanged along with a client random value and a server random value.

At 317, client 301 computes the combined hash using the salt and the actual password associated with the user name. According to various embodiments, the client 301 then provides public information at 321 to server 303. Similarly, server 303 at 325 provides public information to client 301. Information that would not compromise security between a client and a server if accessed by a third party is referred to herein as public information. At 327, both client 301 and server 303 can derive a common value such as a common symmetric key using values available to each of them. Many techniques for key derivation are available. According to various embodiments, a cryptographic accelerator with hash cores according to various embodiments are capable or deriving keys based on selected algorithms in a highly efficient manner.

For example, client 301 generates a common key using public information from server 303, its own private information used to generate public information provided to server 303, and the combined hash calculated by operating on the password appended to a salt. Similarly, server 303 generates a symmetric key by using public information from client 301, a verifier derived from the hash of the combined salt and password, and private information used to generate public information provided to client 301. If the password used to derive the verifier at server 303 is the same as the password used to generate the combined hash value at client 301, the symmetric keys derived at client 301 and server 303 will be the same.

According to various embodiments, the session key can be used for communications between client 301 and server 303. It should be noted that a variety of different cryptographic handshake sequences and communication sequences in general can use the techniques of the present invention. For example, a session key can further be hashed to derive a possibly stronger session key.

At 331, client 301 sends a hash of the session key combined with other public information to server 303. The server 303 then performs a hash of the derived session key combined with the other information known to server 303 to verify the identity of the client 301. Similarly, at 335, server 303 sends a hash of the session key along with other information known to client 301 to allow client 301 to verify the identify of server 303. According to various embodiments, a cryptography accelerator with hash cores according to the techniques of the present invention makes generation of finished messages highly efficient.

It should be noted that in the above implementation, a password is never transmitted over the network. Instead, both network entities use derivatives of the password to generate the session key and other cryptographic information used for secure transmission. Both the password and the session key need not ever be transmitted over the network.

According to various embodiments, a cryptography accelerator speeds operations such as key derivation and finished message generation on both the server and the client side. It is contemplated that a cryptography accelerator can be used in any network entity. It should be noted that the cryptographic handshake sequence shown in FIG. 3 is only one example of a sequence that can use the mechanisms and techniques of the present invention.

FIG. 4 is a diagrammatic representation showing data processing according to various embodiments. In one embodiment, a premaster secret 401 is associated with input data provided to a cryptography accelerator. The cryptography accelerator is used to apply a pseudo-random function (PRF)

411 to the premaster secret 401 to derive a master secret 403. A function that takes one or more inputs and derives an indeterminate output is referred to herein as a pseudo-random function. A master secret 403 corresponds to intermediate data. Another pseudo-random function 413 can be applied to the master secret 403 to derive final or processed data such as authentication keys 405, cryptography keys 407, or initialization sectors 409. The final data generated varies depending on the protocol, protocol version, and type of processing requested.

FIG. 5 is a flow process diagram showing one example of a technique for configuring components such as hash cores in a cryptography accelerator. At 501, input data is received from a component such as parser. According to various embodiments, a parser organizes the data into a form readable by a hash core. At 503, characteristics associated with the input data are determined. Information associated with how to process input data is referred to herein as characteristics of input data. Input data can include information such as protocol version, session ID, cipher suite, and compression method. In one example it is determined what algorithm is being applied to the input data. Algorithms can include versions of TLS, SSL, and IKE as well as other protocols and variants to the protocols.

Determining characteristics can also include determining what kind of operation is to be applied to the data. For example, a key may need to be derived from the data or finished message processing may need to be performed. At 505, an instruction set is retrieved from persistent memory based on the characteristics associated with the input data. Persistent memory may include multiple instruction sets for configuring processing of input data in a variety of manners. At 507, a hash core is configured based on the instruction set. It should be noted that other components such as XOR processing components and temporary memory may also be configured at this point.

Configuring the components may include loading microcode associated with the instruction set into control logic associated with the various hash cores and configurable components. Alternatively, instructions such as microcode can be loaded into a single control logic component associated with the various components. At 509, input data is processed using the instruction set. After a round of processing, input data becomes intermediate data. At 513, intermediate data is maintained in temporary memory during processing. Using temporary memory, data can be manipulated, padded, truncated, etc. At 515, input data and intermediate data finally become final or processed data after completion of processing at 515. The final or processed data is provided back to a merger component for forwarding to an external entity such as an external processor.

FIG. 6 is a process flow diagram showing operations for performing TLS 1.0 key derivation according to various embodiments. TLS 1.0 key derivation can be used during a cryptography handshake sequence such as that shown in FIG. 3. A cryptography accelerator such as that shown in FIG. 2 having an MD5 core and a SHA-1 core can be used for key derivation. According to various embodiments, the inputs to the key derivation operations are a premaster secret, client random information, and server random information. At 601, client and server random information is saved. Client/server random information can be saved in a component such as temporary memory.

At 603, the length of the premaster secret is acquired. At 605, the premaster secret is saved. At 607, a prehash operation is performed on the premaster secret using the MD5 and a SHA-1 hash cores. At 611, it is determined if the current session is a new session. If the current session is a new session, a 48-byte p_MD5 is generated at 613, a 60-byte p_SHA-1 is generated at 615, and the resulting p_MD5 and p_SHA-1 are combined with an XOR operation to acquire the master secret key. The 48-byte master secret key is saved at 619. If it is determined at 611 that the current session is not a new session, the premaster secret to master secret generation is skipped.

At 621, a prehash is performed on the master secret. At 623, the number of bytes needed for the MD5 and SHA-1 operations is determined. The number of bytes needed can be determined by control logic, as the number of bytes needed may be one of the characteristics of the input data stream. Based on the number of bytes needed, MD5 and SHA-1 operations are performed at 625 and 627 using the MD5 and SHA-1 cores as configured by the control logic. The result is combined with an XOR at 629. According to various embodiments, the operations such as sending data to an XOR component are determined based on an instruction set selected by the control logic. A persistent memory allows storage of instruction sets for a variety of operations.

It is determined at 631 if authentication is MD5 or SHA-1. If authentication is MD5 at 631, MD5 inner and outer hash contexts are generated at 633. Otherwise, SHA-1 inner and outer hash contexts are generated at 635. It is determined at 641 whether exportable data is needed. In one example, export restrictions may limit the length of the key. If exportable data is needed, an exportable final write key is generated at 643. If exportable data is not needed, the process is completed. It is also determined at 651 whether the key is needed for a block cipher or a stream cipher. If the key is needed for a block cipher, an initialization vector that is exportable is generated at 653. Otherwise, the operations are complete for TLS 1.0 key derivation. It should be noted that when the operations are complete, data in temporary memory may be removed when the processed data is passed back to an external source. However, instructions sets for configuring the cryptography accelerator can remain in persistent memory.

FIG. 7 is a process flow diagram showing SSLv3 key derivation according to various embodiments. At 701, client and server random information is saved. At 703, it is determined whether the current session is a new session. If the current session is a new session, the master secret is generated by saving the premaster secret at 705, computing an inner hash using a SHA-1 component at 707, and computing an outer hash using an MD5 component at 71 1. A 16-byte master secret is saved at 713. The inner hash and outer hash computations are repeated three times at 715.

If the current session is not a new session, the number of loops needed is determined at 717. Control logic can determine the number of loops needed. The master secret is generated by computing an inner hash using a SHA-1 component at 719 and computing an outer hash using an MD5 component at 721. The master secret is saved at 723. The inner hash and outer hash computations are repeated based on the number of loops needed at 725.

It is determined at 731 if authentication is MD5 or SHA-1. If authentication is MD5 at 731, MD5 inner and outer hash contexts are generated at 733. Otherwise, SHA-1 inner and outer hash contexts are generated at 735. It is determined at 741 whether exportable data is needed. If exportable data is needed, an exportable final write key is generated at 743. If exportable data is not needed, the process is completed. It is also determined at 751 whether the key is needed for a block cipher or a stream cipher. If the key is needed for a block

cipher, an initialization vector that is exportable is generated at **753**. Otherwise, the operations are complete for SSLv3 key derivation.

FIG. **8** is a flow processed diagram showing TLS 1.0 finished message generation, according to various embodiments. Finished message or verification messages are used to confirm that two network entities were successful in key exchange and authentication processes. The finished message is typically the first message associated with the recently negotiated algorithms, keys, and secret information. Network entities that receive finished messages verify that the contents are correct.

In one example, once a client has generated and sent its own finished message to a server and has received and validated a finished message from the server, the client can begin to send and receive application related data to the server. To generate a finished message according to TLS 1.0, the master secret is saved at **801**. At **803**, the length of the handshake message used for finished message generation is acquired. It is determined at **805** whether the handshake message length is less than 512 bits. If the handshake message length is not less than 512 bits, both SHA-1 and MD5 hash algorithms are performed on 512 bit blocks of the handshake message. At **811**, intermediate states are saved. At **813**, the last block of the handshake message is saved. If the handshake message length itself is less than 512 bits, the handshake message is simply saved at **813**. At **815**, the intermediate states are loaded.

At **817**, a final MD5 and SHA-1 hash are performed. The resulting data is loaded at **819** into a pseudo-random function. At **821**, a 16-byte p_MD5 hash is generated and at **823** at 10-byte SHA-1 hash is generated. The results are combined with an XOR at **825**. The client finished message is saved at **827** and concatenated with the last block at **833**. At **835**, intermediate states are loaded and a final MD5 and SHA-1 hash are generated for the server. The resulting data is loaded at **843** into a pseudo-random function. A 16-byte p_MD5 hash is generated at **851** and a 20-byte SHA-1 hash is generated at **853**. The result is combined with an XOR at **851**. The server finished message is saved at **853**.

FIG. **9** is a flow process diagram showing finished message generation for SSLv3. At **901**, the master secret is saved. At **903**, the length of the handshake message is determined. It is determined at **905** whether the handshake message length is less than 512 bits. If the handshake message length is not less than 512 bits, both SHA-1 and MD5 hash algorithms are performed on 512 bit blocks of the handshake message. At **911**, intermediate states are saved. At **913**, the last block of the handshake message is saved. If the handshake message length itself is less than 512 bits, the handshake message is simply saved at **913**. At **915**, the intermediate states are loaded. At **917**, a final MD5 and SHA-1 hash are performed.

At **919**, the master secret is loaded. An outer MD5 hash and SHA-1 hash are generated for the client at **921**. The client finished message is saved at **923** and concatenated with the last block at **925**. Intermediate states are loaded at **931**. An inner MD5 hash and SHA-1 hash are generated for the server at **933**. The master secret is loaded at **935**. At **937**, an outer MD5 hash and SHA-1 hash are generated for the server. The server finished messages are saved at **939**.

FIGS. **6-9** are process flow diagrams showing hash operations that can be performed according to various embodiments of the present invention. The operations can be performed using components such as hash cores, XOR components, and temporary memory configured using instruction sets maintained in persistent memory. It should be noted that the operations shown are specified for particular key derivation and finished message generation operations associated with TLS 1.0 and SSLv3. However, the techniques and mechanisms of the present invention should not be restricted to these two protocols and the specified versions of these two protocols.

While the invention has been particularly shown and described with reference to specific embodiments thereof, it will be understood by those skilled in the art that changes in the form and details of the disclosed embodiments may be made without departing from the spirit or scope of the invention. It is therefore intended that the invention be interpreted to include all variations and equivalents that fall within the true spirit and scope of the present invention.

What is claimed is:

1. A cryptography accelerator, the accelerator comprising:

a first hash core that is configurable and operable to perform a plurality of rounds of hash computations on input data to derive processed data;

a persistent memory containing a plurality of instruction sets, wherein the plurality of instruction sets configure the first hash core to enable operations on the input data and intermediate data during the plurality of rounds of hash computations, wherein one of said plurality of instruction sets is chosen for configuration of said first hash core on the basis of characteristics of said input data, said characteristics comprising at least one of a protocol version, a session ID, a compression method, and a cipher suite; and

a temporary memory coupled to the first hash core, the temporary memory operable to hold input data and intermediate data.

2. The accelerator of claim **1**, further comprising a second hash core, the second hash core configurable and operable to perform a plurality of rounds of hash computations on input data to derive processed data.

3. The accelerator of claim **2**, wherein the first hash core is configurable to operate as either a SHA-1 or an MD5 hash core.

4. The accelerator of claim **2**, wherein the second hash core is configurable to operate as either a SHA-1 or an MD5 hash core.

5. The accelerator of claim **4**, wherein the first hash core is configured as the inner hash and the second hash core is configured as the outer hash for HMAC operations.

6. The accelerator of claim **2**, further comprising control logic operable to determine characteristics associated with the input data and select an instruction set based on the input data characteristics.

7. The accelerator of claim **6**, wherein the control logic configures the first hash core using the instruction set.

8. The accelerator of claim **7**, wherein the control logic configures the second hash core using the instruction set.

9. The accelerator of claim **6**, wherein the control logic is configured to manage the first hash core using the instruction set.

10. The accelerator of claim **9**, wherein the control logic is configured to manage the second hash core using the instruction set.

11. The accelerator of claim **1**, wherein the characteristics of the input data further comprise information associated with a premaster sequence, an initialization vector, export information, and key length.

12. The accelerator of claim **11**, wherein the characteristics of the input data further comprise information associated with how encryption and authentication will be performed.

13. The accelerator of claim 1, wherein the plurality of instruction sets in persistent memory comprise instructions for performing TLS 1.0 and SSLv3 key derivation and finished message generation.

14. The accelerator of claim 13, wherein the persistent memory and the temporary memory are provided in the same component.

15. A method for performing hash operations, the method comprising:

receiving input data;

determining characteristics of the input data, the characteristics comprising at least one of a protocol version, a session ID, a compression method, and a cipher suite;

on the basis of the characteristics, selecting an instruction set for performing hash operations on the input data, wherein the instruction set is selected from a plurality of instruction sets maintained in persistent memory associated with a first hash core;

configuring the first hash core using the instruction set, wherein the first hash core is operable to perform one or more hash operations on the input data based on the instruction set, thereby creating intermediate data;

storing the intermediate data in a temporary memory; and

performing, in the first hash core, one or more additional hash operations on the intermediate data.

16. The method of claim 15, further comprising:

configuring a second hash core using the instruction set, wherein the second hash core is operable to perform the hash operations on the input data based on the instruction set.

17. The method of claim 16, wherein performing the hash operations on the input data comprises performing a plurality of rounds of hash computations on the input data to derive the intermediate data and processed data.

18. The method of claim 16, wherein the first hash core is a SHA-1 hash core.

19. The method of claim 18, wherein the second hash core is a MD5 hash core.

20. The method of claim 19, wherein the selected instruction set comprises instructions for performing key derivation or finished message generation.

21. The method of claim 16, wherein the first hash core is configured as the inner hash and the second hash core is configured as the outer hash for HMAC operations.

22. The method of claim 15, wherein the characteristics of the input data further comprise information associated with a premaster sequence, an initialization vector, export information, and key length.

23. The method of claim 22, wherein the characteristics of the input data further comprise information associated with how encryption and authentication will be performed.

24. An apparatus for performing hash operations, the apparatus comprising:

means for receiving input data;

means for determining characteristics of the input data, the characteristics comprising at least one of a protocol version, a session ID, a compression method, and a cipher suite;

means for selecting, on the basis of the characteristics, an instruction set for performing hash operations on the input data, wherein the instruction set is selected from a plurality of instruction sets maintained in persistent memory associated with a first hash core;

means for configuring the first hash core using the instruction set, wherein the first hash core is operable to perform one or more hash operations on the input data based on the instruction set to create intermediate data;

means for storing the intermediate data in a temporary memory; and

means for transferring the intermediate data from the temporary memory to the first hash core for performing additional hash operations on the intermediate data.

25. The apparatus of claim 24, further comprising:

means for configuring a second hash core using the instruction set, wherein the second hash core is operable to perform the hash operations on the input data based on the instruction set.

26. The apparatus of claim 25, wherein performing the hash operations on the input data comprises performing a plurality of rounds of hash computations on the input data to derive the intermediate data and processed data.

27. The apparatus of claim 25, wherein the first hash core is a SHA-1 hash core.

28. The apparatus of claim 27, wherein the second hash core is a MD5 hash core.

29. The apparatus of claim 28, wherein the selected instruction set comprises instructions for performing key derivation or finished message generation.

30. The apparatus of claim 25, wherein the first hash core is configured as the inner hash and the second hash core is configured as the outer hash for HMAC operations.

31. The apparatus of claim 24, wherein the characteristics of the input data further comprise information associated with a premaster sequence, an initialization vector, export information, and key length.

32. The apparatus of claim 31, wherein the characteristics of the input data further comprise information associated with how encryption and authentication will be performed.

* * * * *

Exhibit C



(12) **United States Patent**
Fong et al.

(10) **Patent No.: US 7,656,643 B2**
(45) **Date of Patent: Feb. 2, 2010**

(54) **SCALABLE INTEGRATED CIRCUIT HIGH DENSITY CAPACITORS**

(75) Inventors: **Victor Chiu-Kit Fong**, San Jose, CA (US); **Eric Bruce Blecker**, San Jose, CA (US); **Tom W. Kwan**, Cupertino, CA (US); **Ning Li**, Fremont, CA (US); **Sumant Ranganathan**, Sunnyvale, CA (US); **Chao Tang**, Mountain View, CA (US); **Pieter Vorenkamp**, Laguna Niguel, CA (US)

(73) Assignee: **Broadcom Corporation**, Irvine, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 173 days.

(21) Appl. No.: **11/878,696**

(22) Filed: **Jul. 26, 2007**

(65) **Prior Publication Data**

US 2008/0266749 A1 Oct. 30, 2008

Related U.S. Application Data

(63) Continuation of application No. 11/013,789, filed on Dec. 17, 2004, now Pat. No. 7,259,956.

(60) Provisional application No. 60/530,648, filed on Dec. 19, 2003.

(51) **Int. Cl.**
H01G 4/228 (2006.01)

(52) **U.S. Cl.** **361/306.2**; 361/303; 361/306.1; 361/311; 361/313; 361/301.4

(58) **Field of Classification Search** 361/306.1, 361/306.3, 321.1, 311–313, 306.2, 303–305, 361/301.1, 321.2; 257/306, 307
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,583,359	A *	12/1996	Ng et al.	257/306
5,872,695	A	2/1999	Fasano et al.	
5,939,766	A	8/1999	Stolmeijer et al.	
6,061,228	A	5/2000	Palmer et al.	
6,066,537	A	5/2000	Poh	
6,266,226	B1 *	7/2001	Hayashi	361/303

(Continued)

OTHER PUBLICATIONS

Roberto Aparicio, *Student Member, IEEE*, and Ali Hajimiri, *Member, IEEE*: Capacity Limits and Matching Properties of Integrated Capacitors; IEEE Journal of Solid-State Circuits, pp. 384-393, vol. 37, No. 3, Mar. 2002.

Hirad Samavati, *Student Member, IEEE*, Ali Hajimiri, Arvin R. Shahani, Gitty N. Nasserbakht, and Thomas H. Lee, *Member, IEEE*: Fractal Capacitors; IEEE Journal of Solid-State Circuits, pp. 2035-2041, vol. 33, No. 12, Dec. 1998.

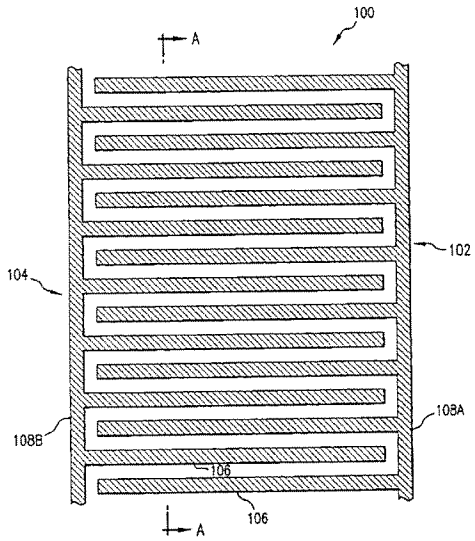
(Continued)

Primary Examiner—Nguyen T Ha
(74) *Attorney, Agent, or Firm*—Sterne, Kessler, Goldstein & Fox P.L.L.C.

(57) **ABSTRACT**

The present invention provides several scalable integrated circuit high density capacitors and their layout techniques. The capacitors are scaled, for example, by varying the number of metal layers and/or the area of the metal layers used to form the capacitors. The capacitors use different metallization patterns to form the metal layers, and different via patterns to couple adjacent metal layers. In embodiments, optional shields are included as the top-most and/or bottom-most layers of the capacitors, and/or as side shields, to reduce unwanted parasitic capacitance.

19 Claims, 37 Drawing Sheets



U.S. PATENT DOCUMENTS

6,414,835	B1	7/2002	Wolf et al.	
6,542,351	B1 *	4/2003	Kwang	361/303
6,653,681	B2	11/2003	Appel	
6,690,570	B2	2/2004	Hajimiri et al.	
6,737,698	B1 *	5/2004	Paul et al.	257/306
6,801,422	B2	10/2004	Mosley	
6,933,551	B1	8/2005	Stribley et al.	
7,009,832	B1	3/2006	Chen et al.	
7,151,660	B2	12/2006	Chen et al.	
7,259,956	B2	8/2007	Fong et al.	
2003/0206389	A1	11/2003	Hajimiri et al.	
2004/0031982	A1	2/2004	Devries et al.	
2005/0135042	A1	6/2005	Fong et al.	

OTHER PUBLICATIONS

Jay Rajagopalan and Haris Basit; Optimization of Metal-Metal Comb-Capacitors for RF Applications, 4 pages, http://www.oea.com/document/Optimiz_Metal.pdf., Jan. 12, 2006.

Tirdad Sowlati, Vickram Vathulya and Domine Leenaerts; High Density Capacitance Structures in Submicron CMOS for Low Power RF Applications, *ISLPED* '01, Aug. 6-7, 2001, pp. 243-246.

* cited by examiner

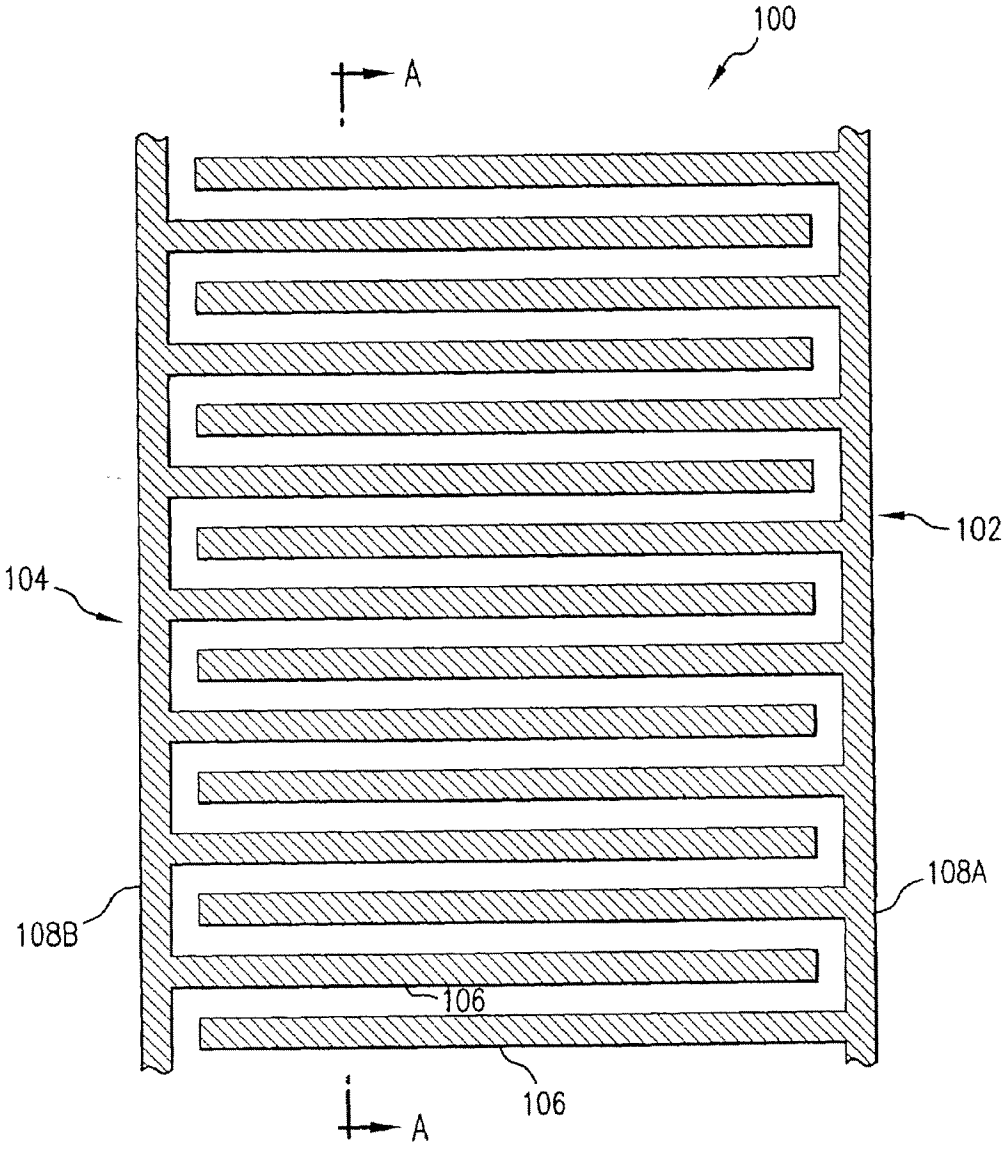


FIG. 1A

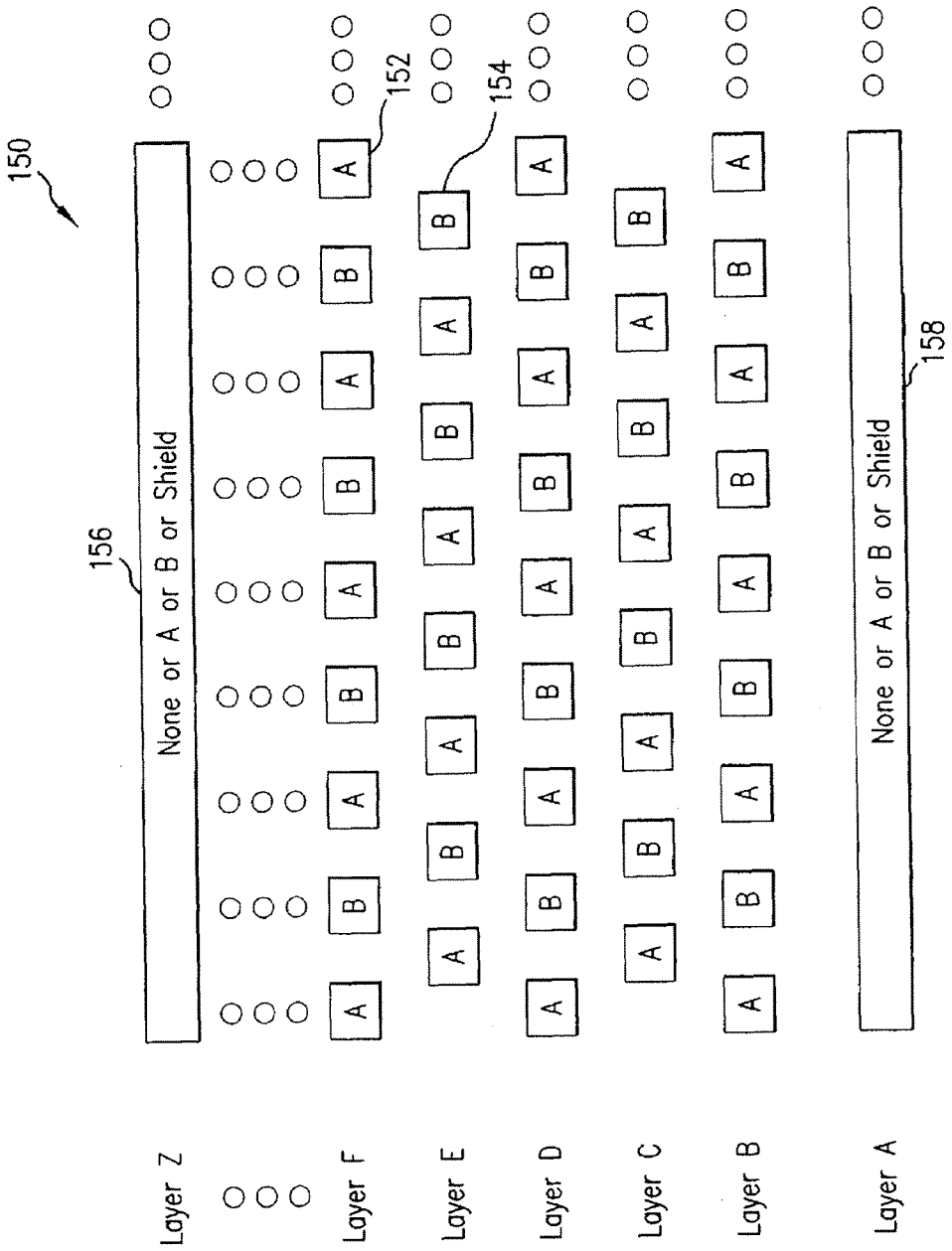

150
152
154
156
158
None or A or B or Shield
None or A or B or Shield
Layer Z
Layer F
Layer E
Layer D
Layer C
Layer B
Layer A
A
B

FIG. 1B

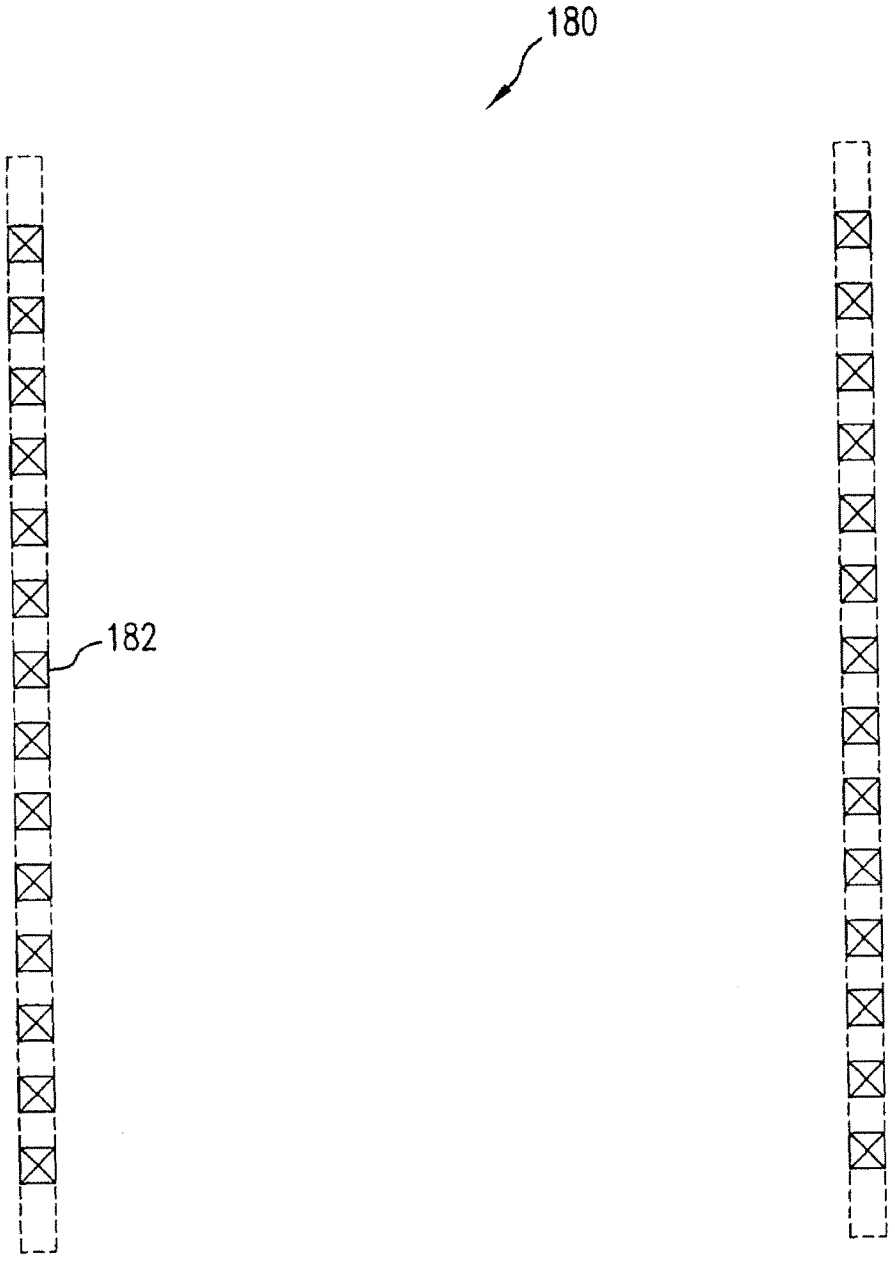


FIG. 1C

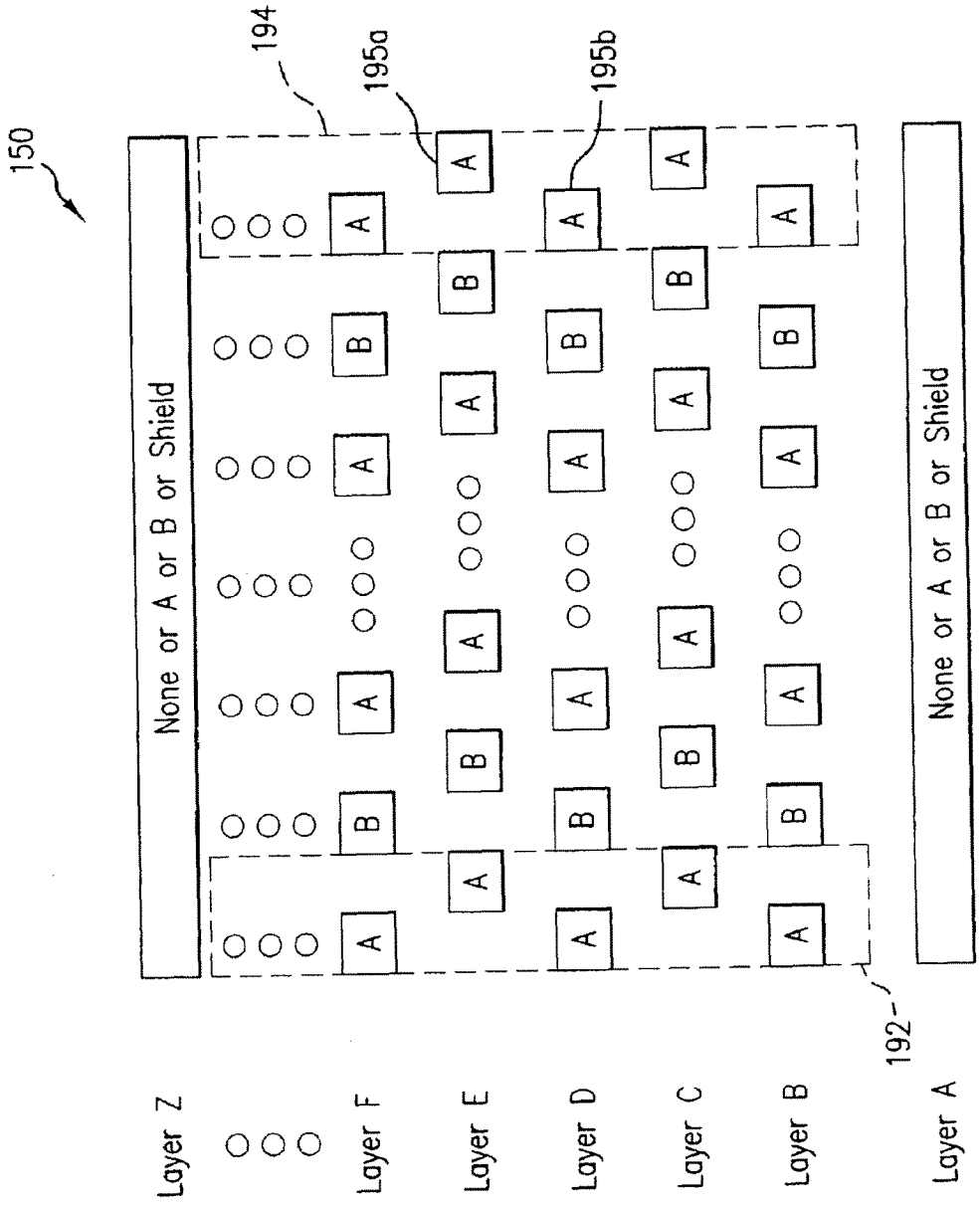

150
194
195a
195b
None or A or B or Shield
None or A or B or Shield
192
Layer Z
Layer F
Layer E
Layer D
Layer C
Layer B
Layer A

FIG. 1D

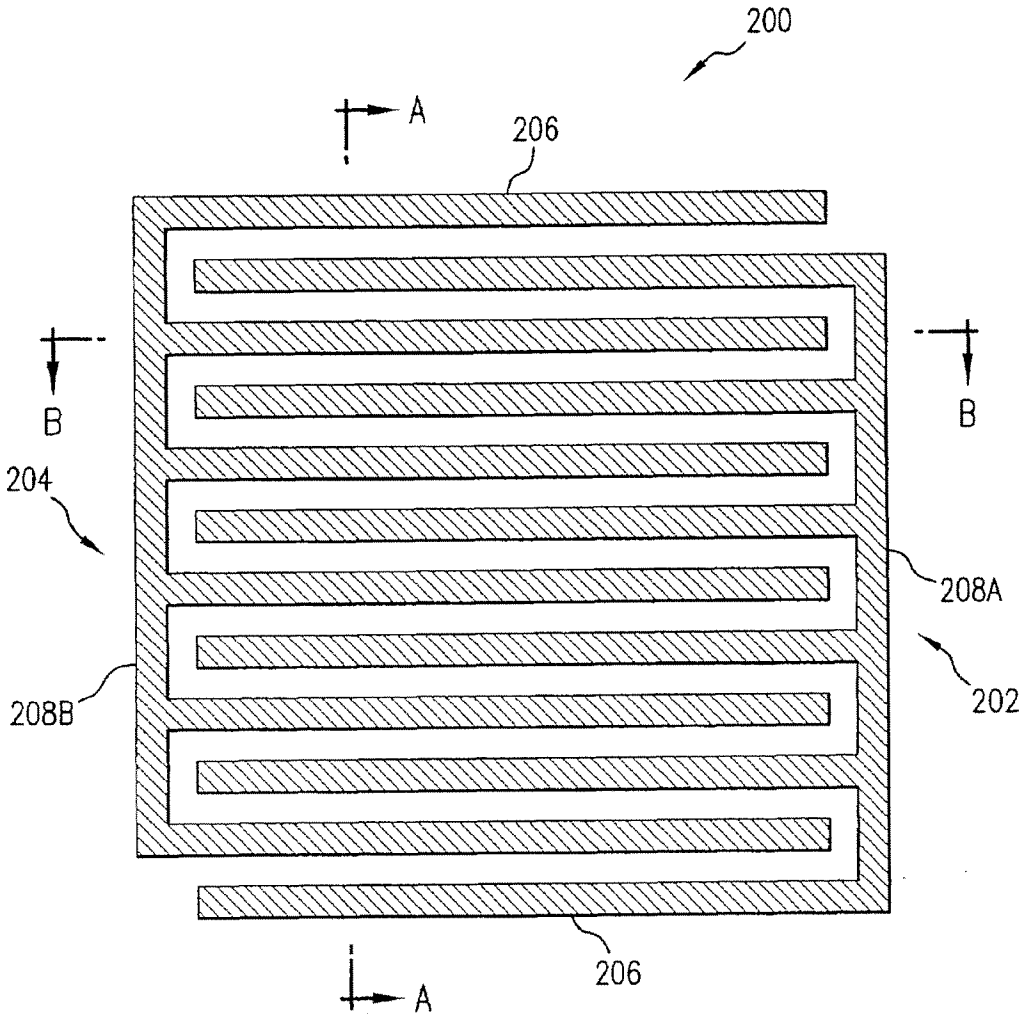


FIG.2A

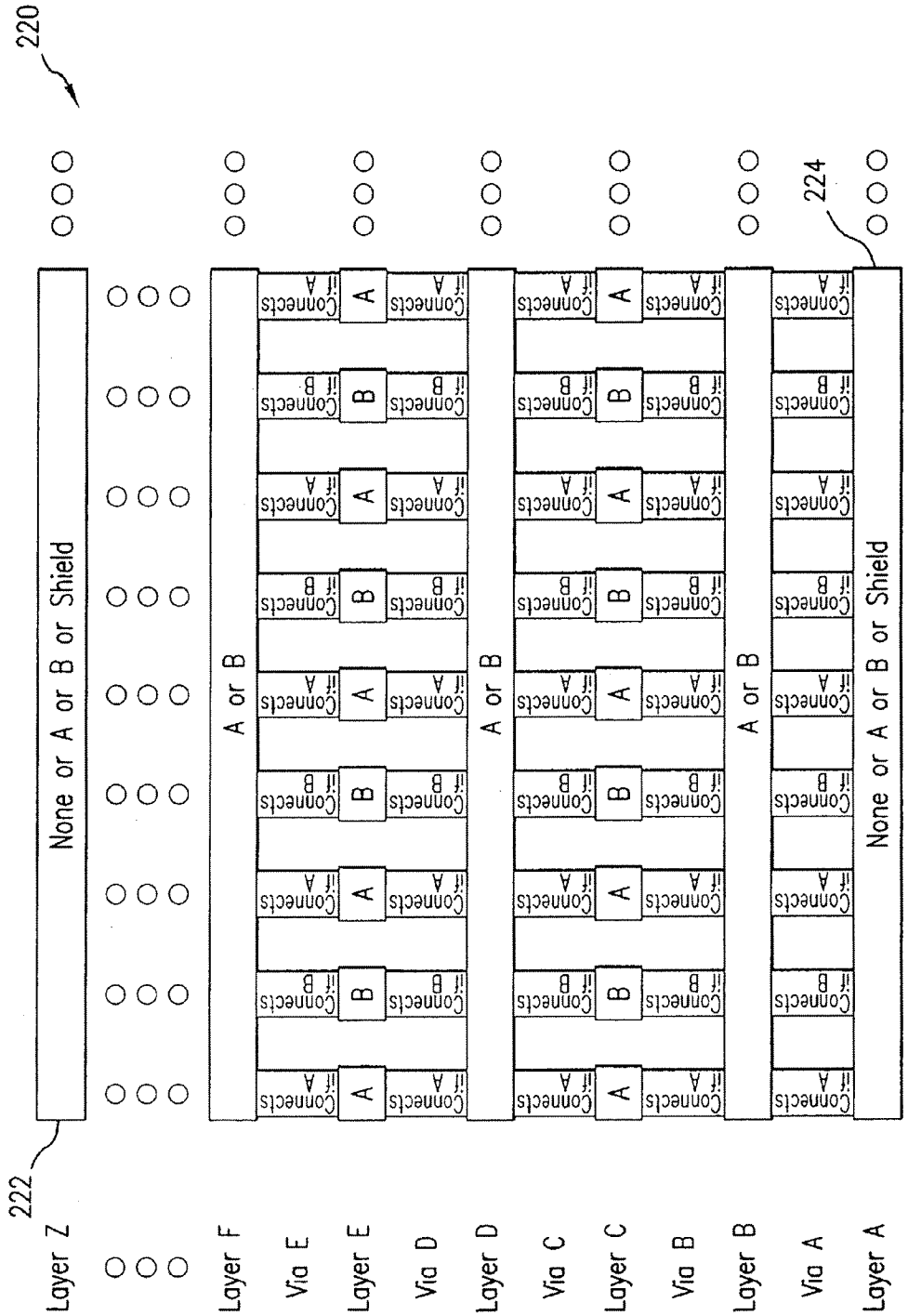

220
224
222
Layer Z
Layer F
Via E
Layer E
Via D
Layer D
Via C
Layer C
Via B
Layer B
Via A
Layer A
None or A or B or Shield
A or B
A or B
A or B
None or A or B or Shield
A
B
Connects if A
Connects if B

FIG.2B

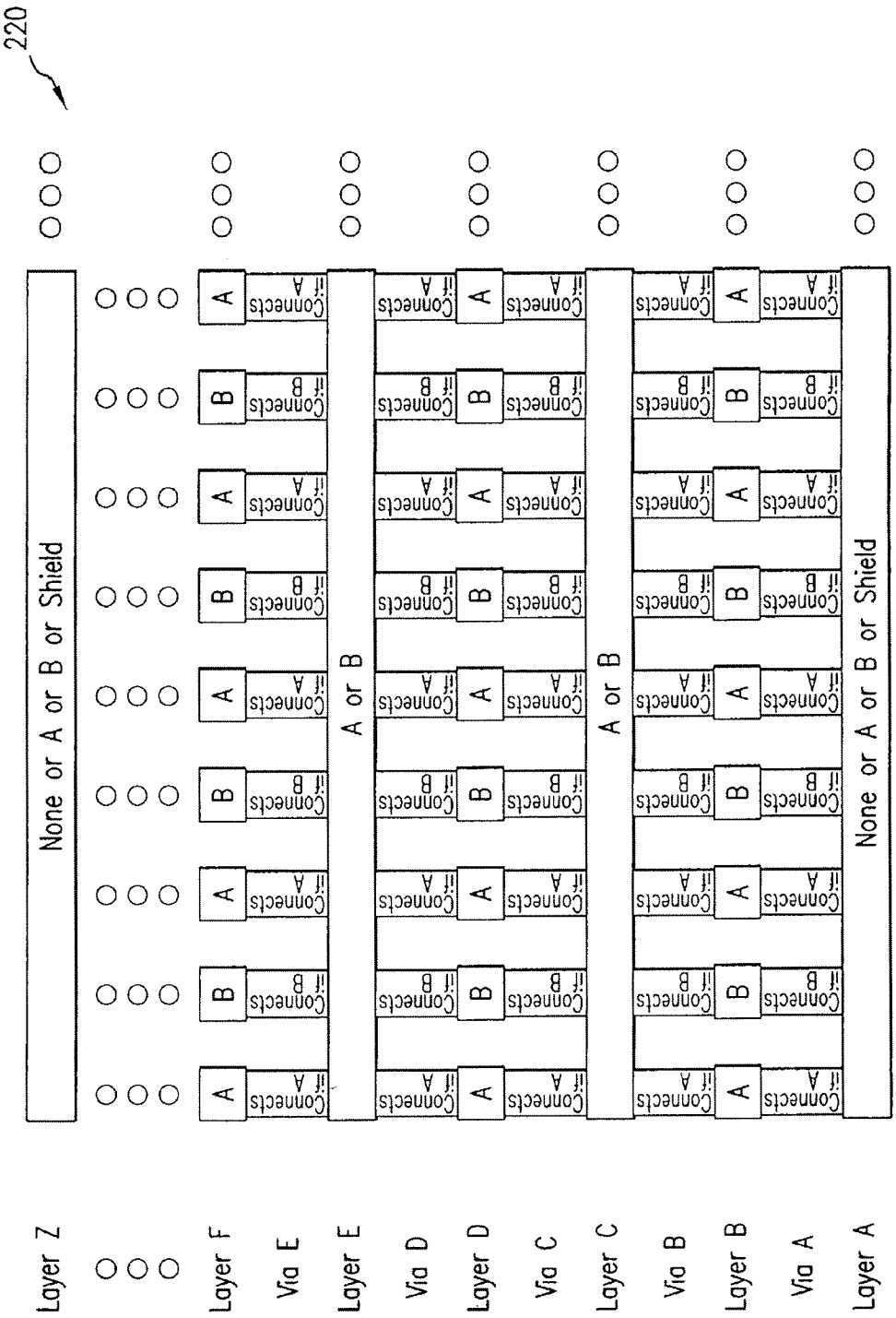

220
Layer Z
Layer F
Via E
Layer E
Via D
Layer D
Via C
Layer C
Via B
Layer B
Via A
Layer A
None or A or B or Shield
A or B
A or B
None or A or B or Shield
A
B
Connects if A
Connects if B

FIG.2C

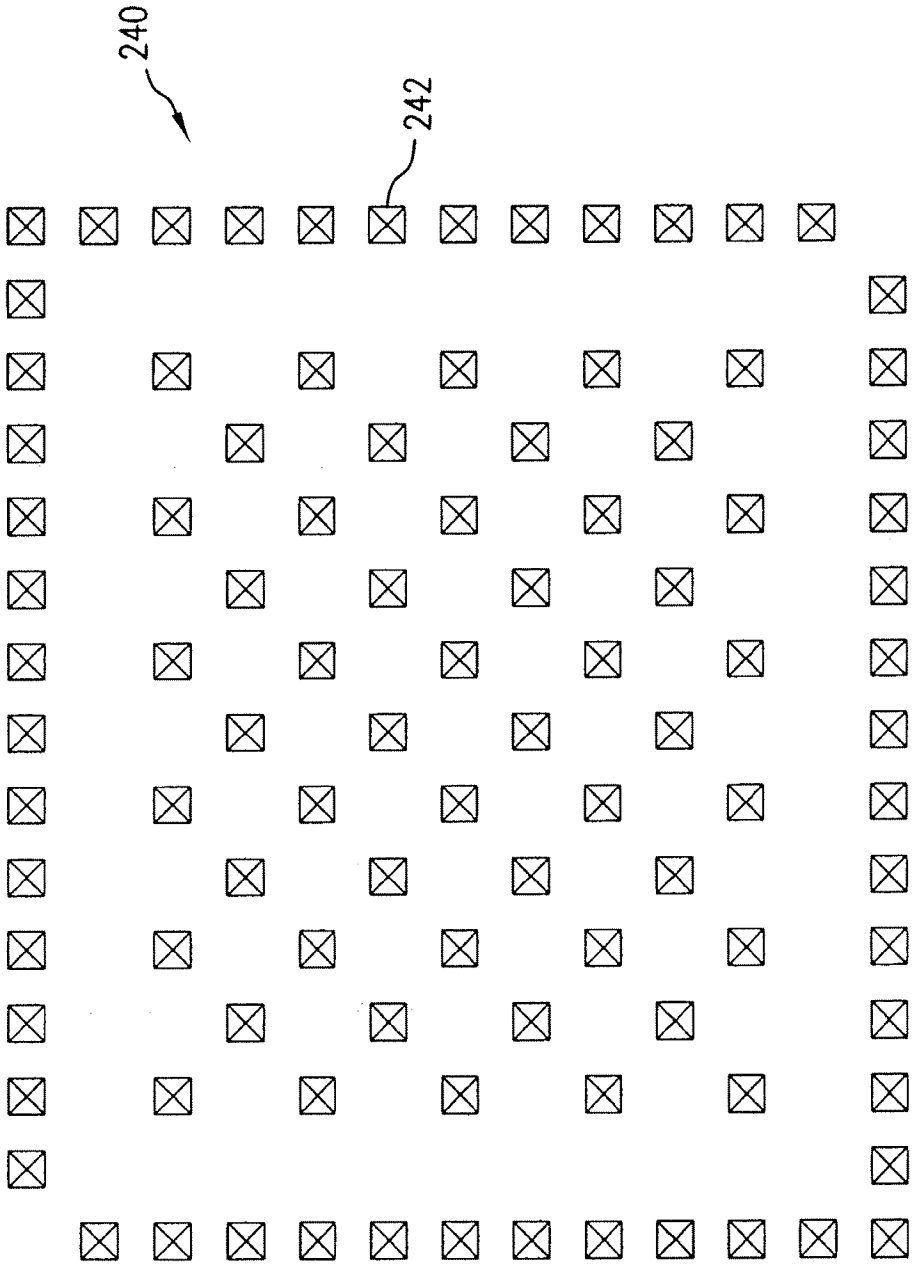


FIG. 2D

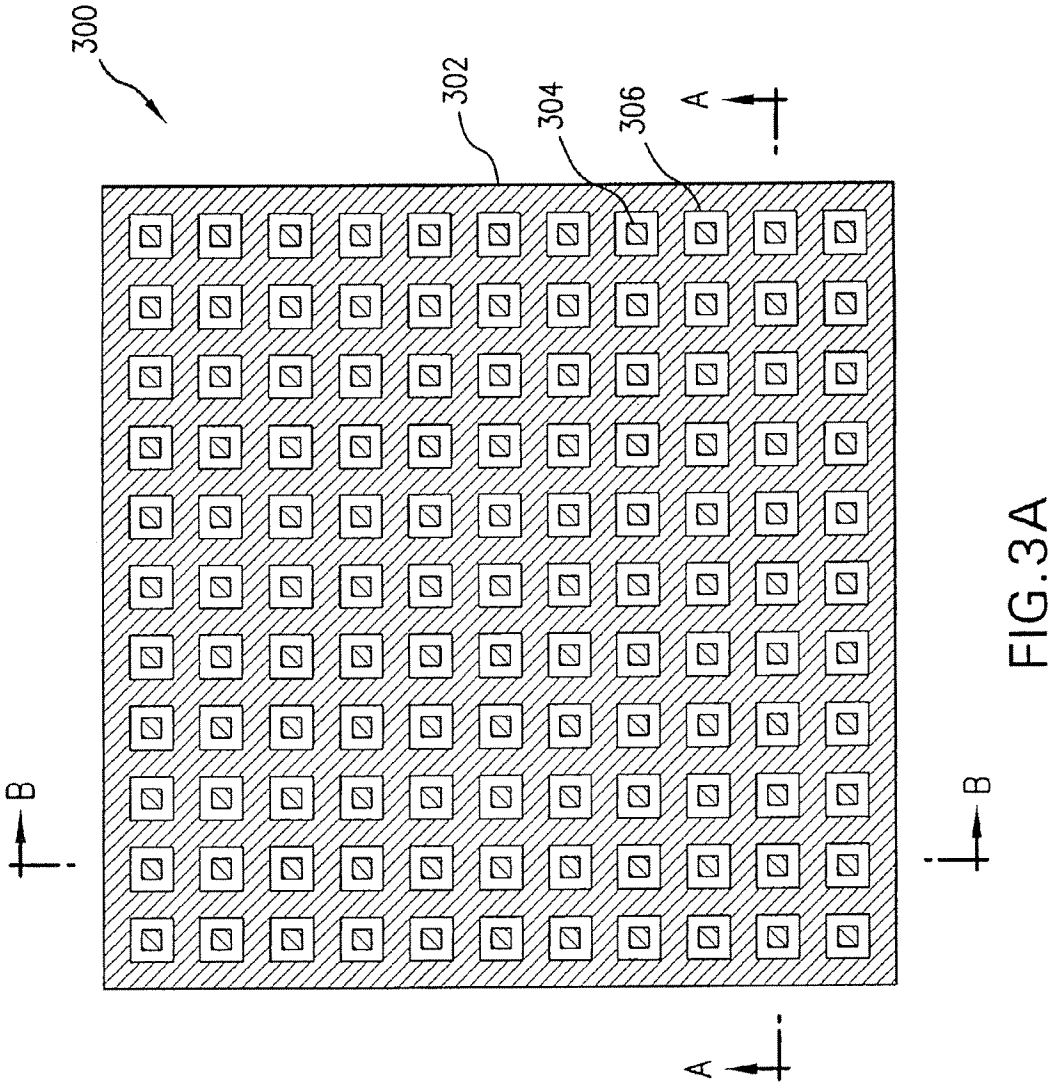


FIG.3A

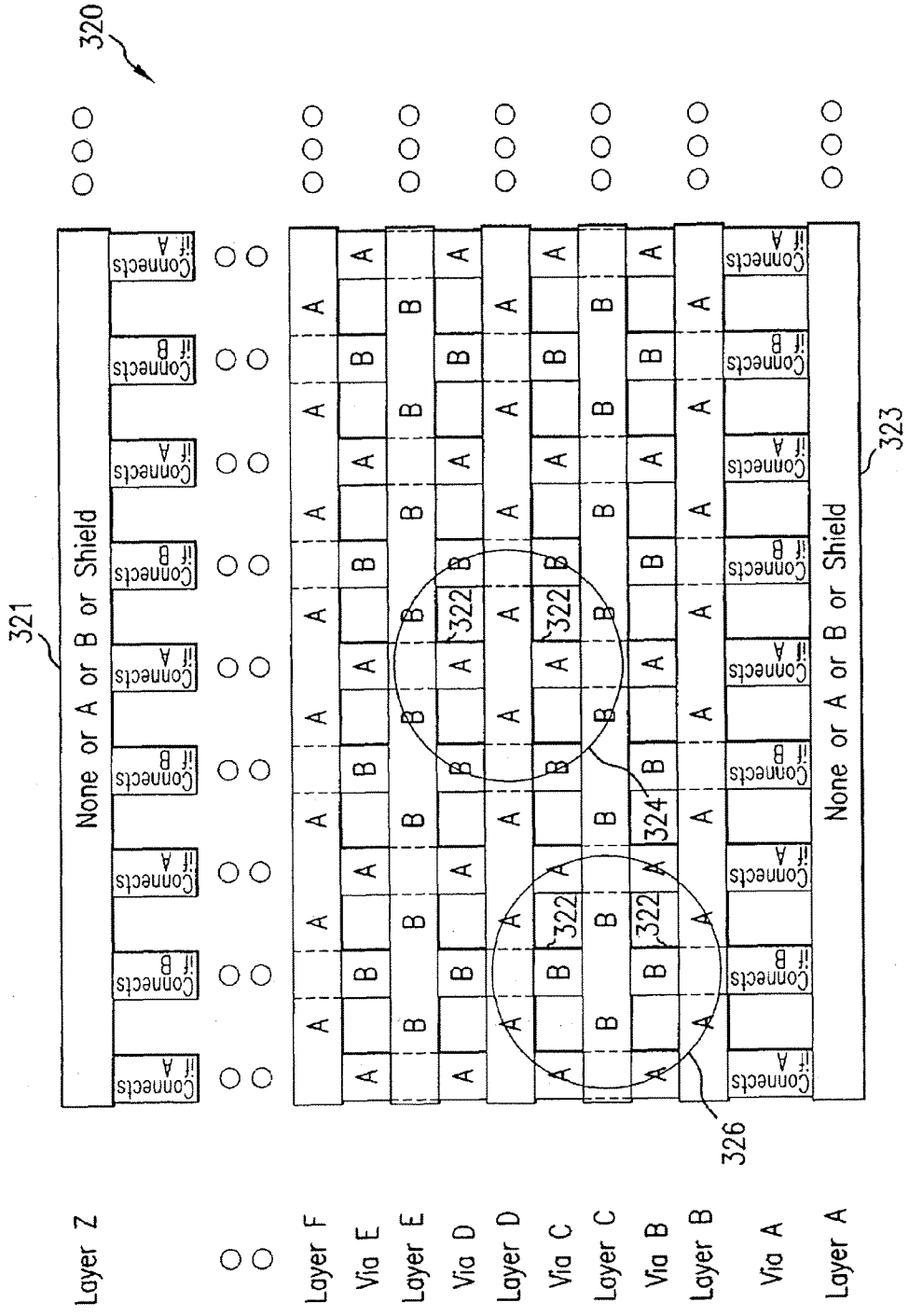

320
321
323
326
322
324
None or A or B or Shield
None or A or B or Shield
Connects A
Connects B
Layer Z
Layer F
Via E
Layer E
Via D
Layer D
Via C
Layer C
Via B
Layer B
Via A
Layer A

FIG.3B

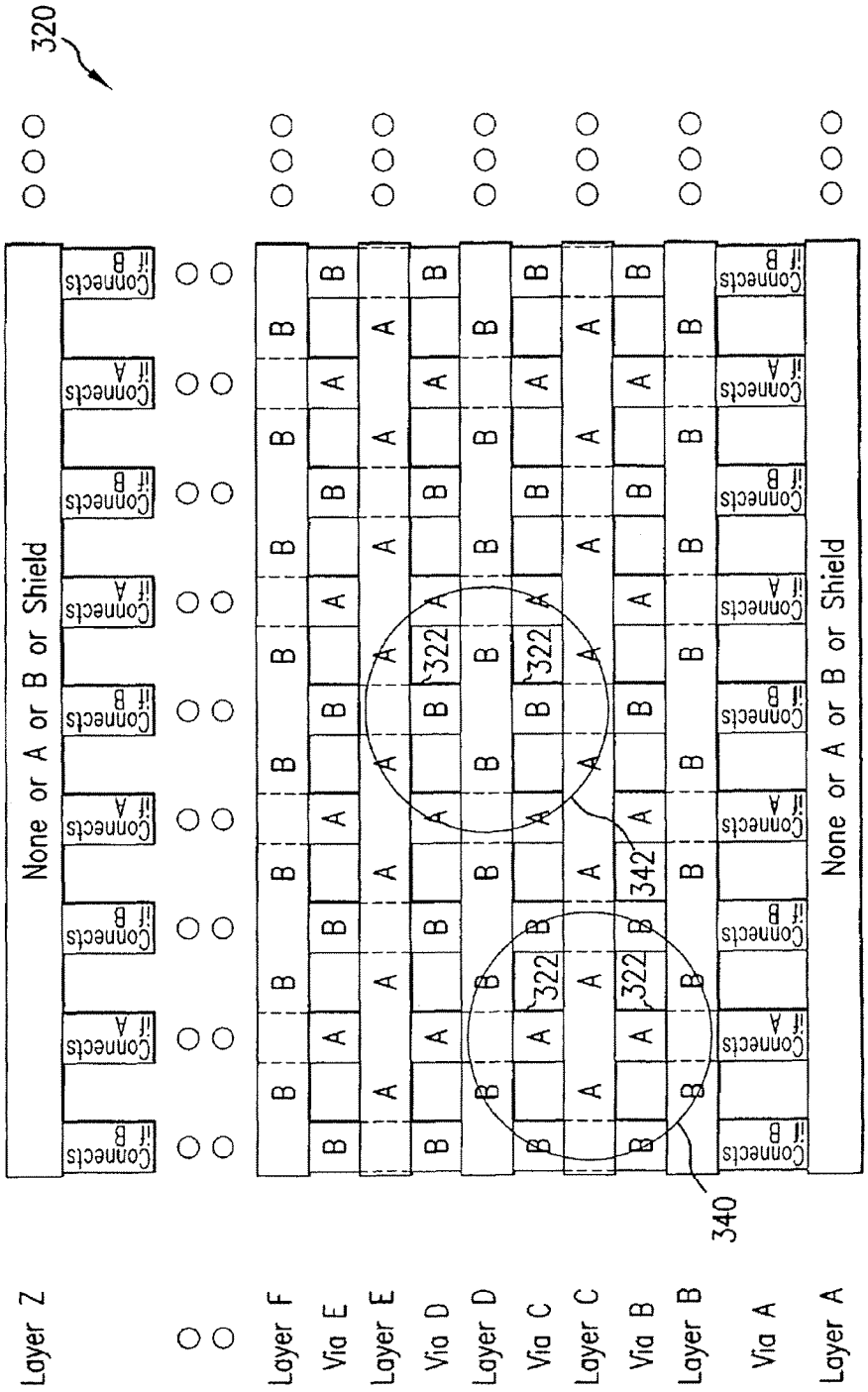

320
None or A or B or Shield
Connects B
Connects A
Layer Z
Layer F
Via E
Layer E
Via D
Layer D
Via C
Layer C
Via B
Layer B
Via A
Layer A
None or A or B or Shield
322
342
340

FIG. 3C

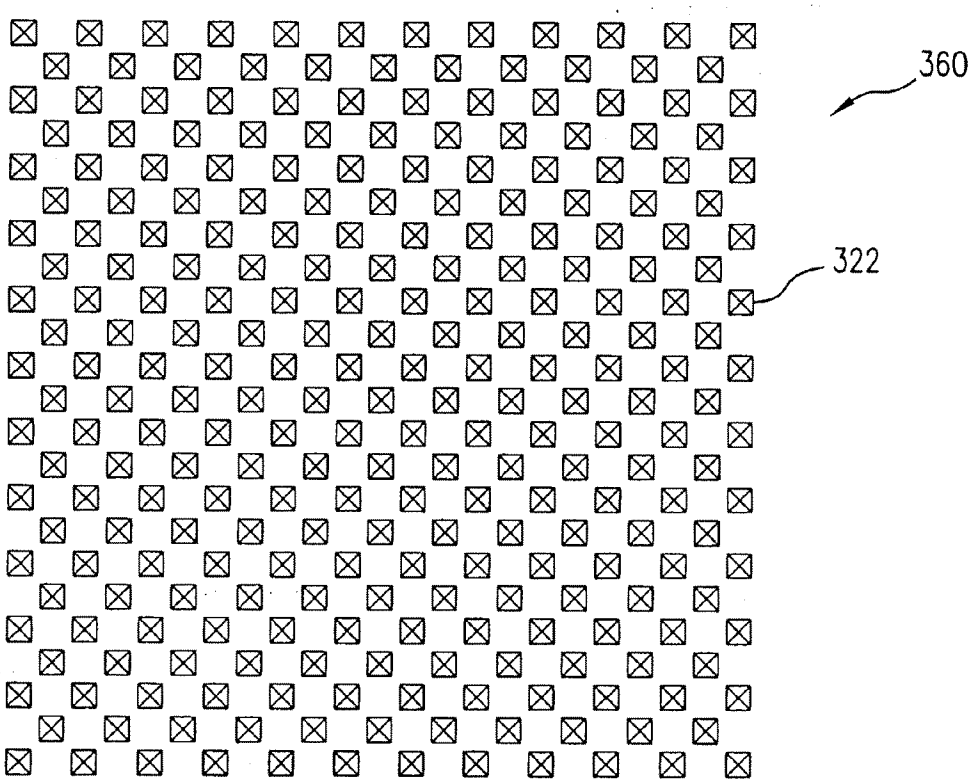


FIG.3D

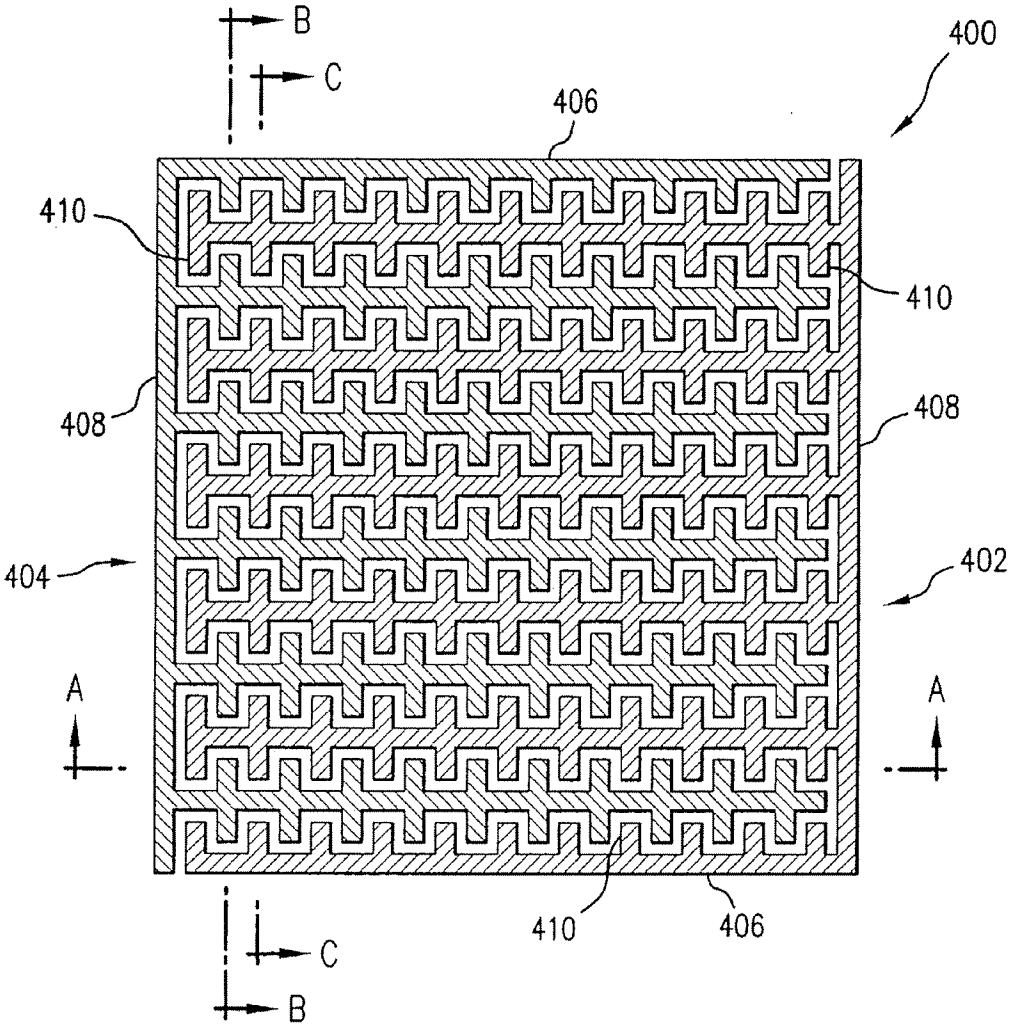


FIG.4A

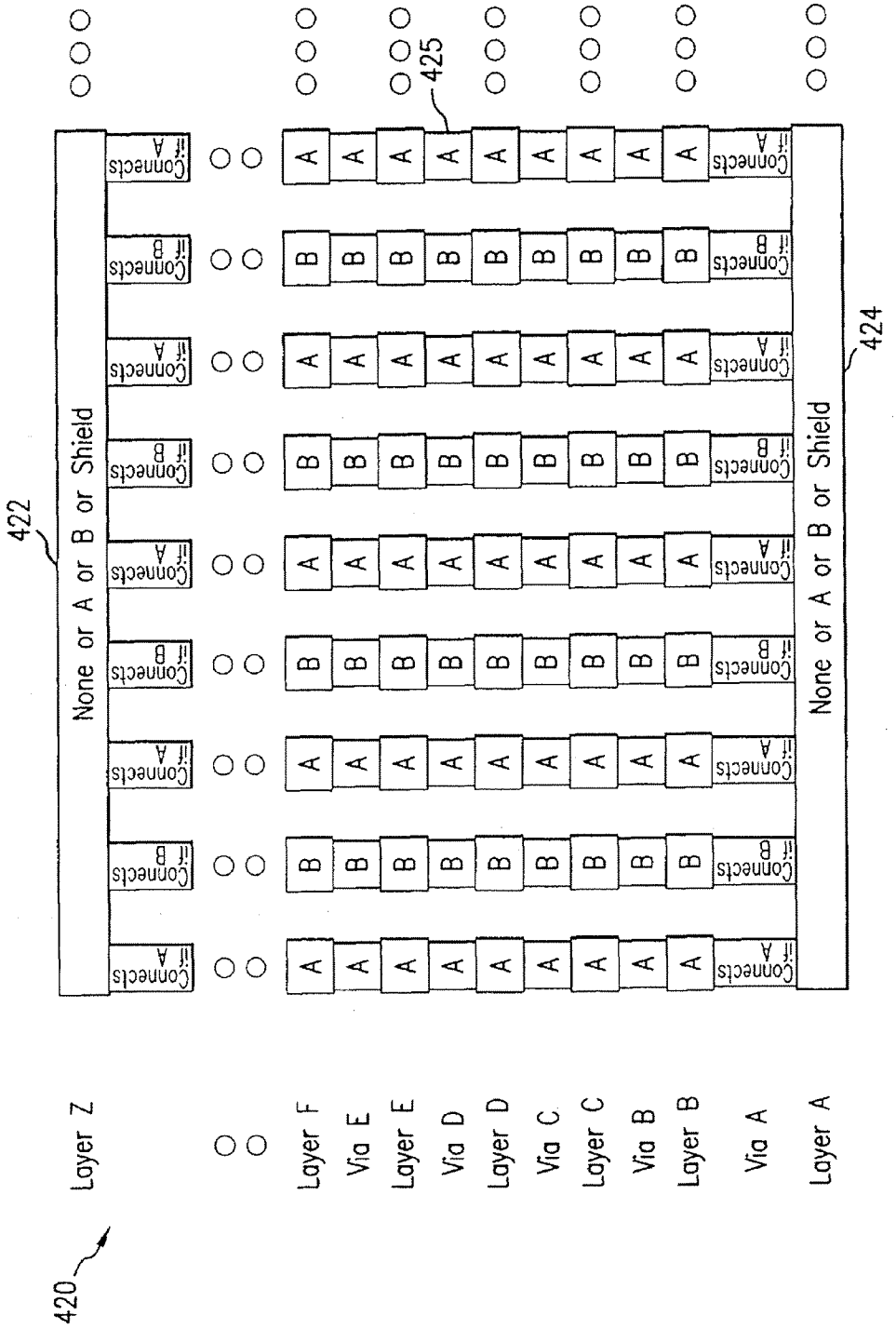

420
422
424
425
Layer Z
Layer F
Via E
Layer E
Via D
Layer D
Via C
Layer C
Via B
Layer B
Via A
Layer A
None or A or B or Shield
None or A or B or Shield
If A Connects
If B Connects

FIG.4B

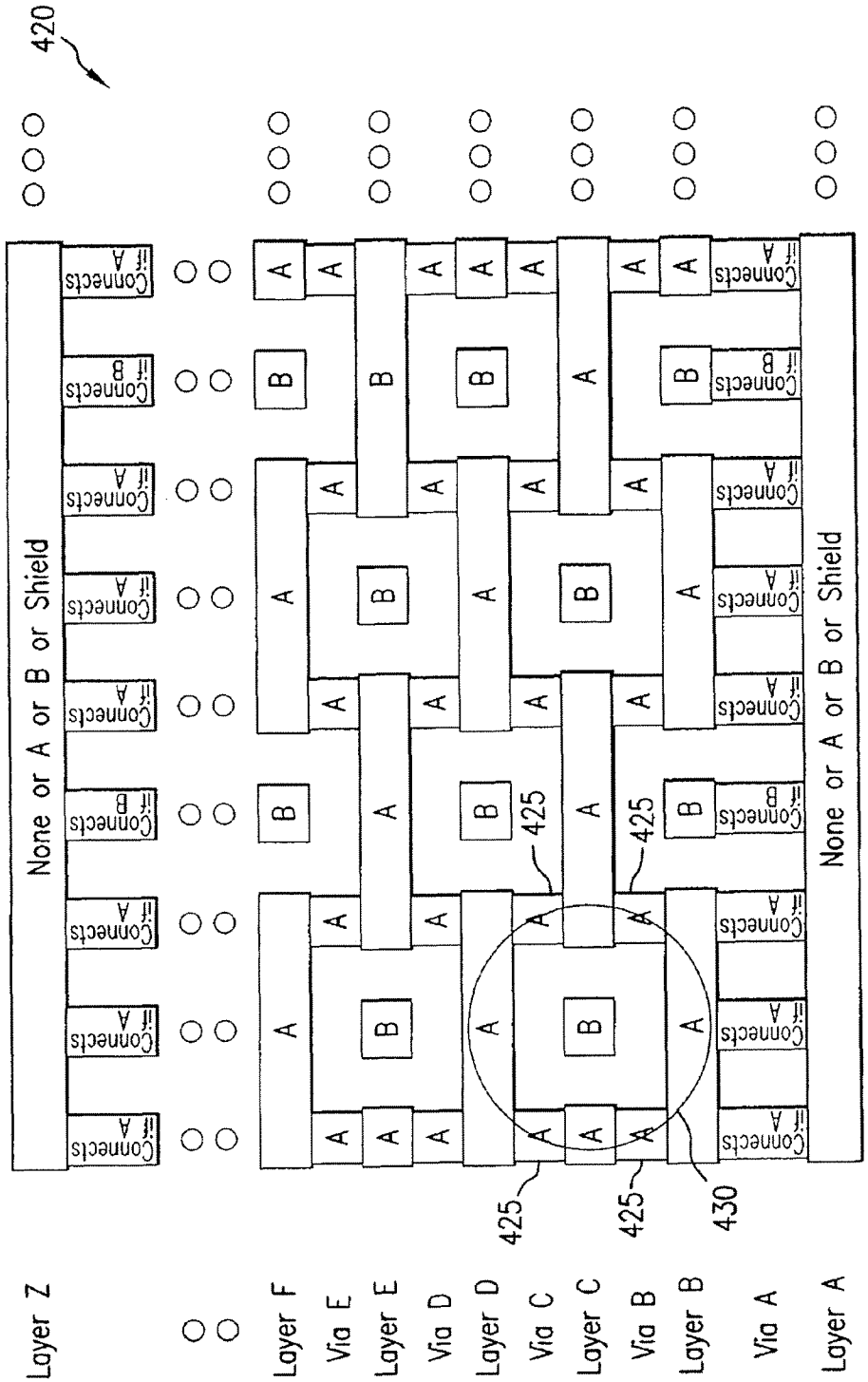

420
None or A or B or Shield
Connects A
Connects B
A
B
425
425
425
430
Layer Z
Layer F
Via E
Layer E
Via D
Layer D
Via C
Layer C
Via B
Layer B
Via A
Layer A
None or A or B or Shield

FIG.4C

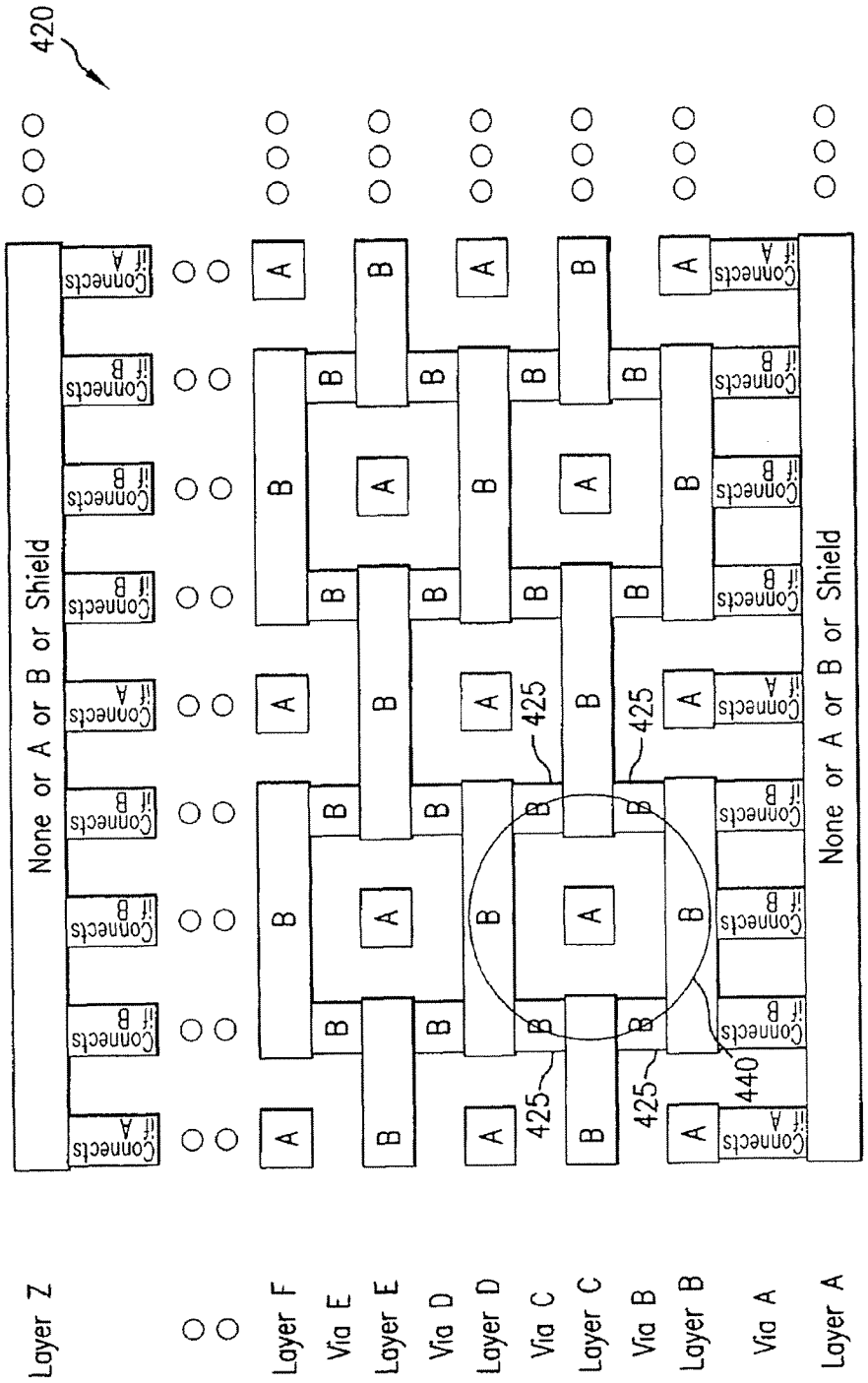

420
None or A or B or Shield
Connects A
Connects B
425
440
Layer Z
Layer F
Via E
Layer E
Via D
Layer D
Via C
Layer C
Via B
Layer B
Via A
Layer A

FIG. 4D



FIG.4E

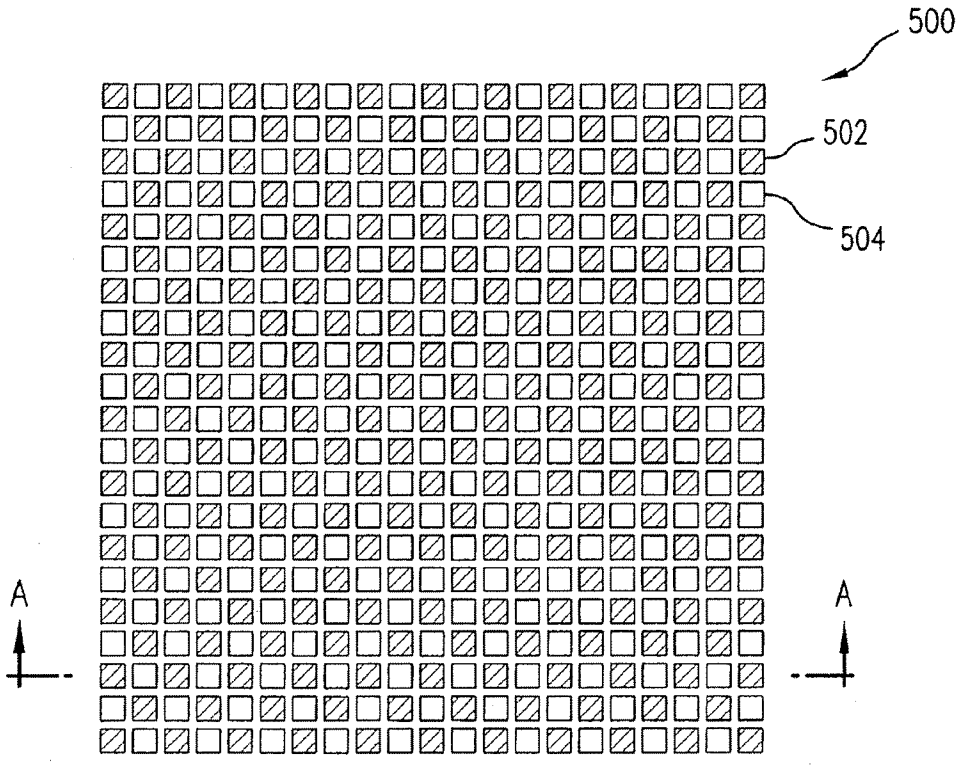


FIG.5A

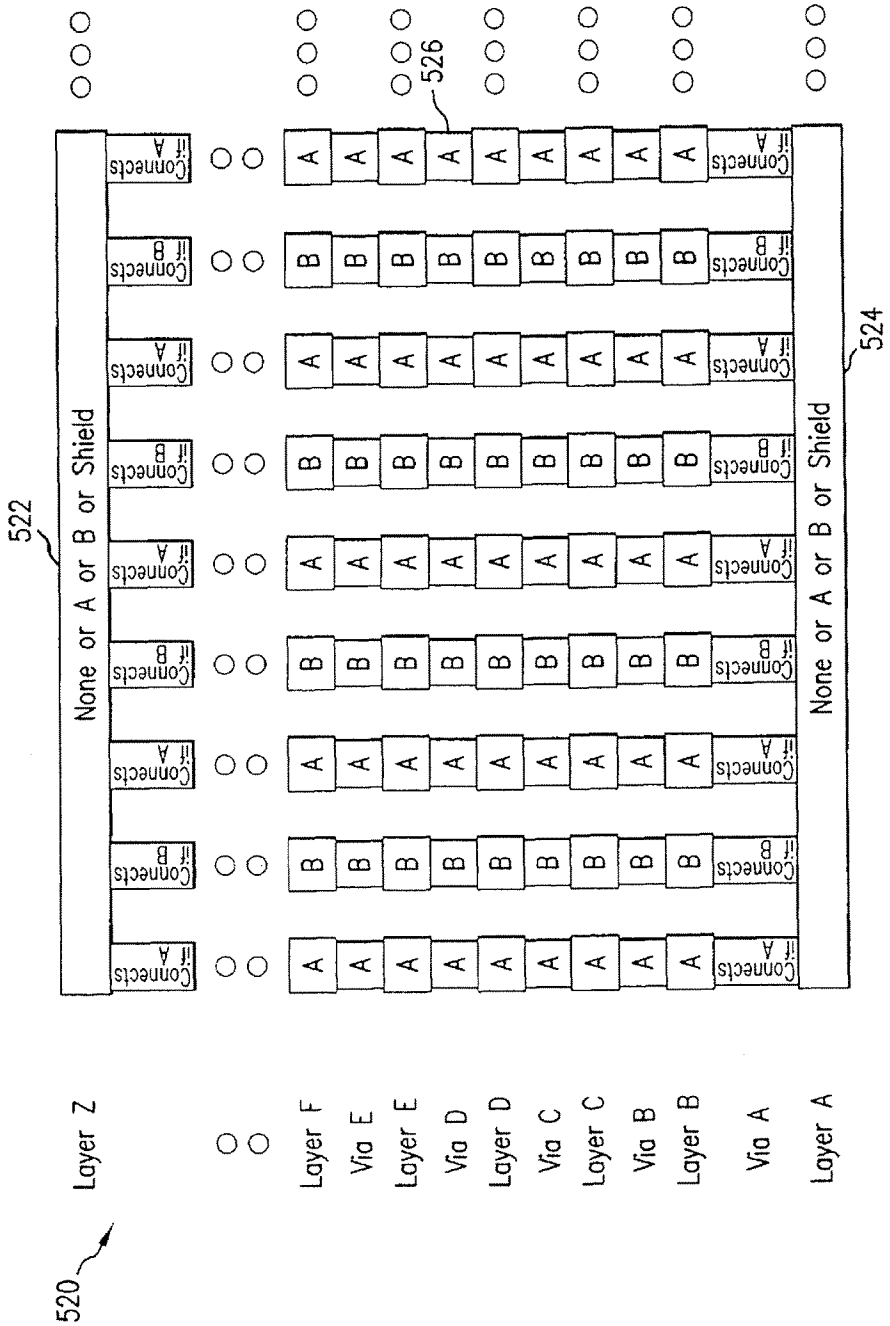

520
522
524
526
Layer Z
Layer F
Via E
Layer E
Via D
Layer D
Via C
Layer C
Via B
Layer B
Via A
Layer A
None or A or B or Shield
None or A or B or Shield
Connects A
Connects B
A
B

FIG.5B

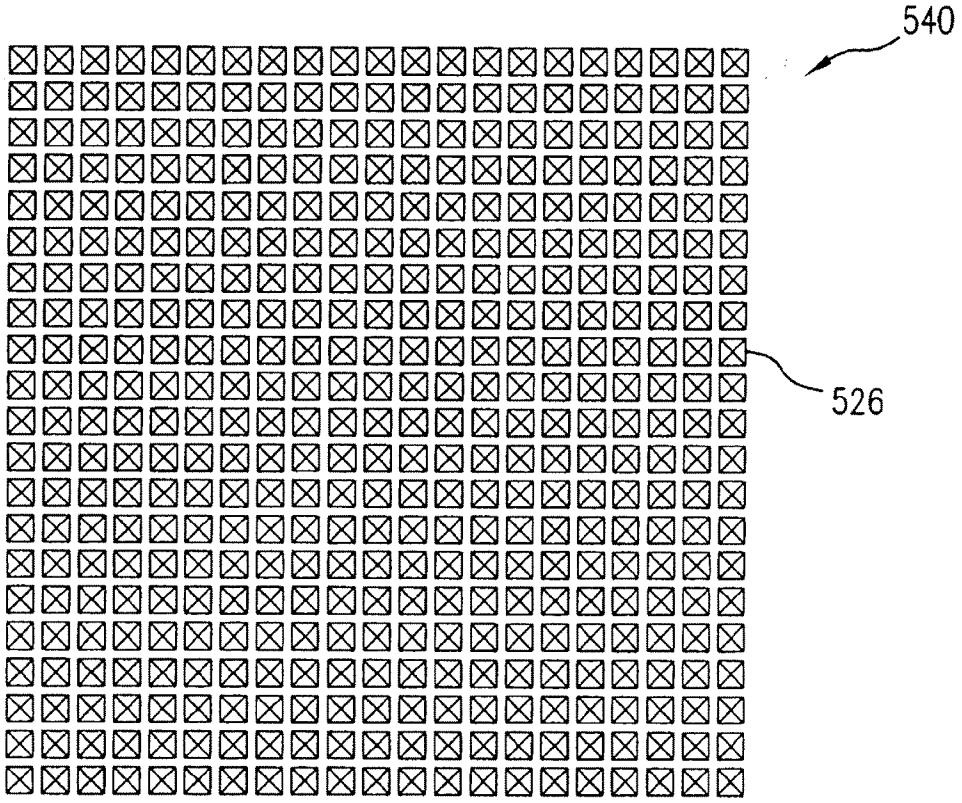


FIG.5C

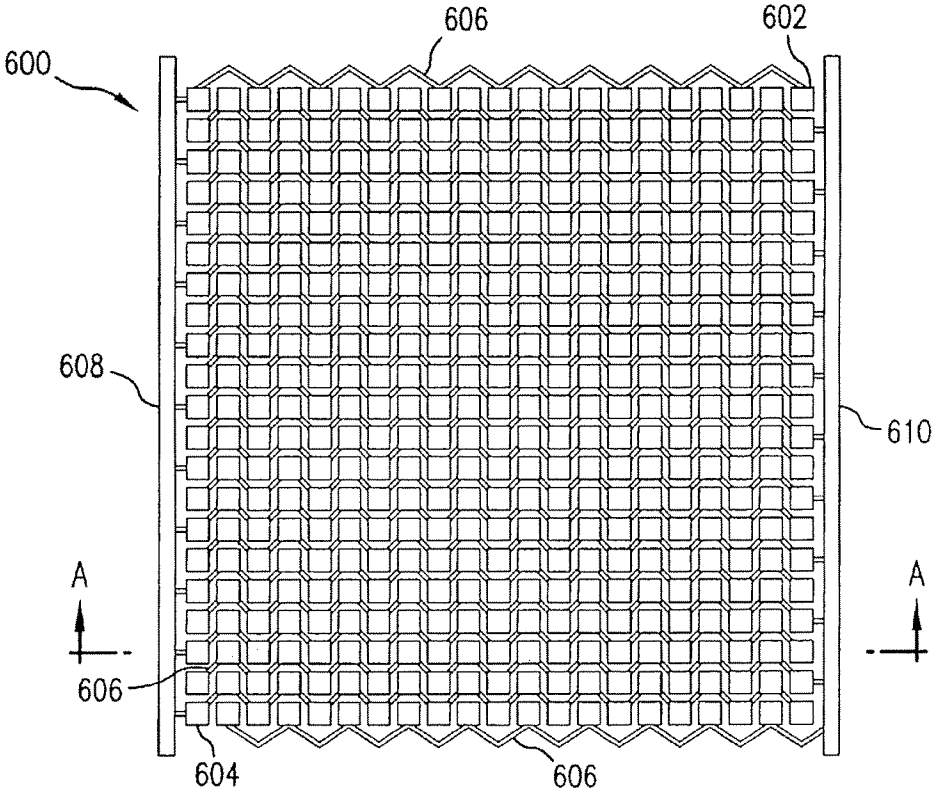


FIG.6A

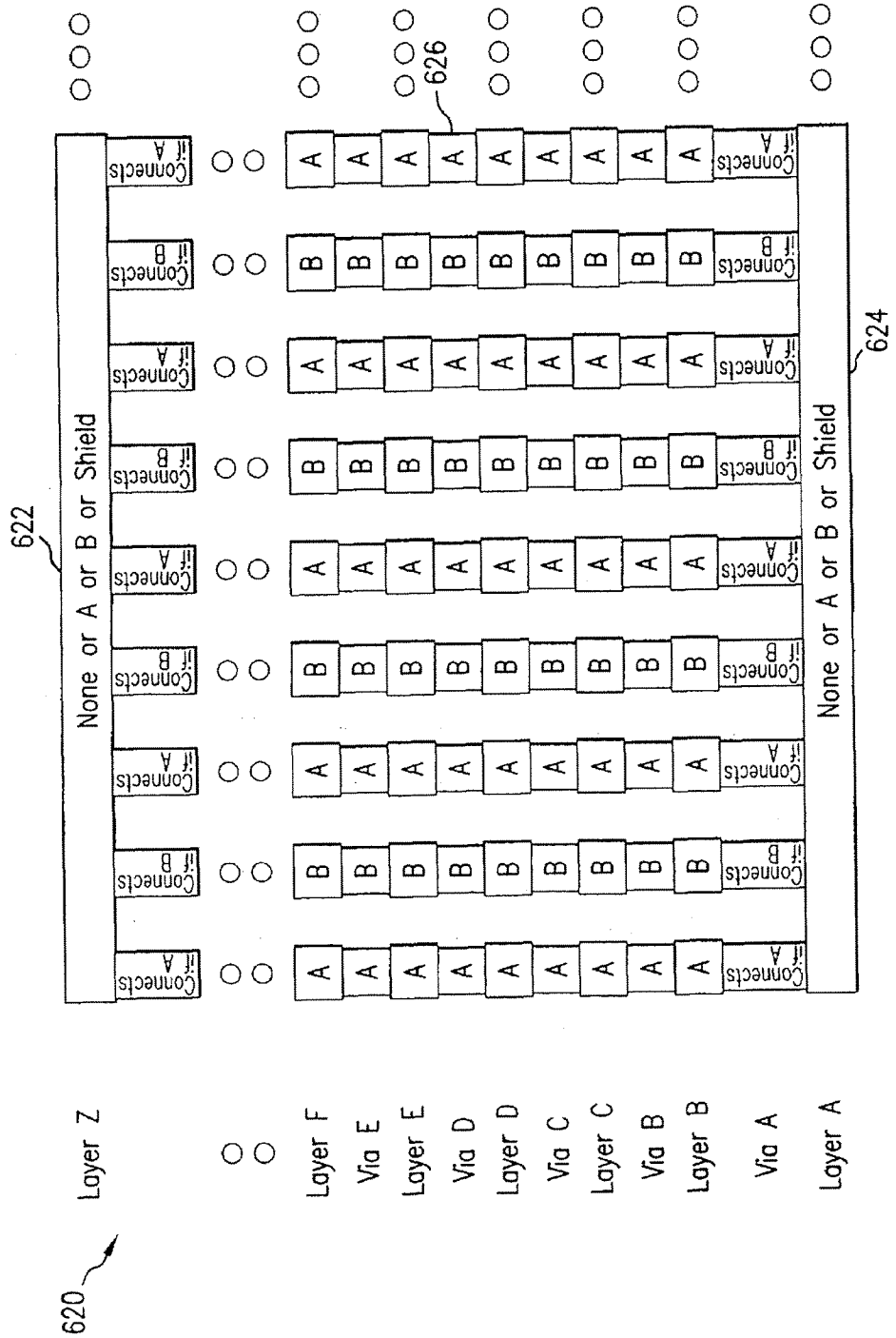

620
622
624
626
Layer Z
Layer F
Via E
Layer E
Via D
Layer D
Via C
Layer C
Via B
Layer B
Via A
Layer A
None or A or B or Shield
None or A or B or Shield
If A Connects
If B Connects

FIG.6B

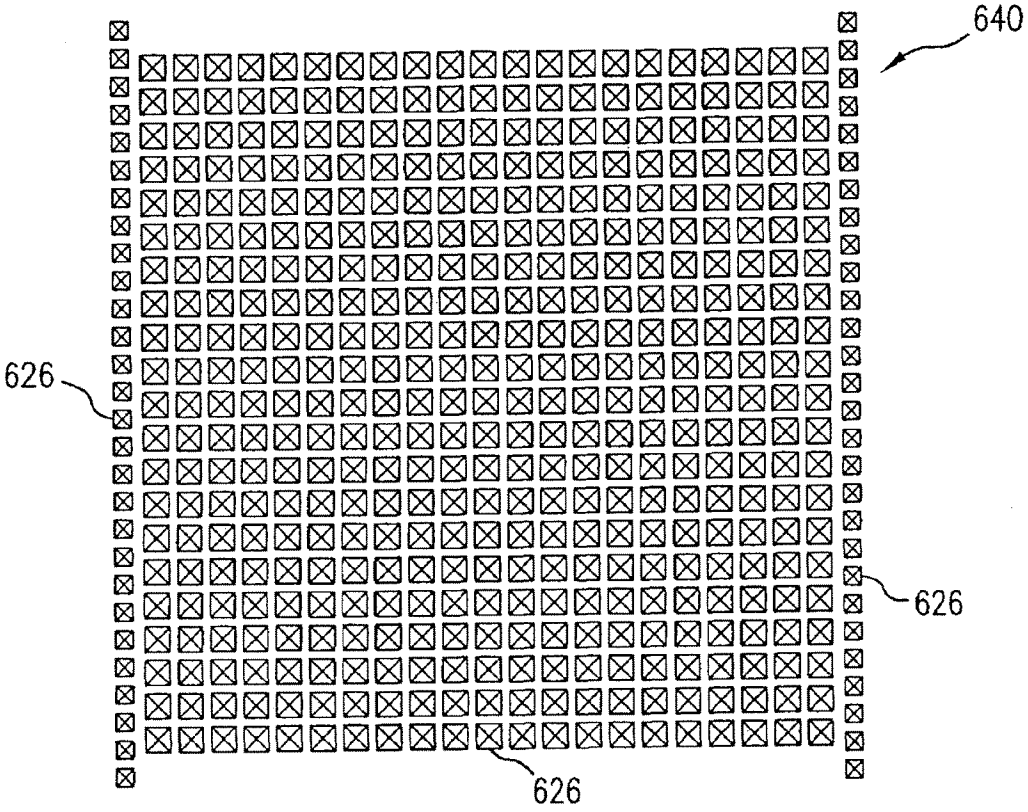


FIG.6C

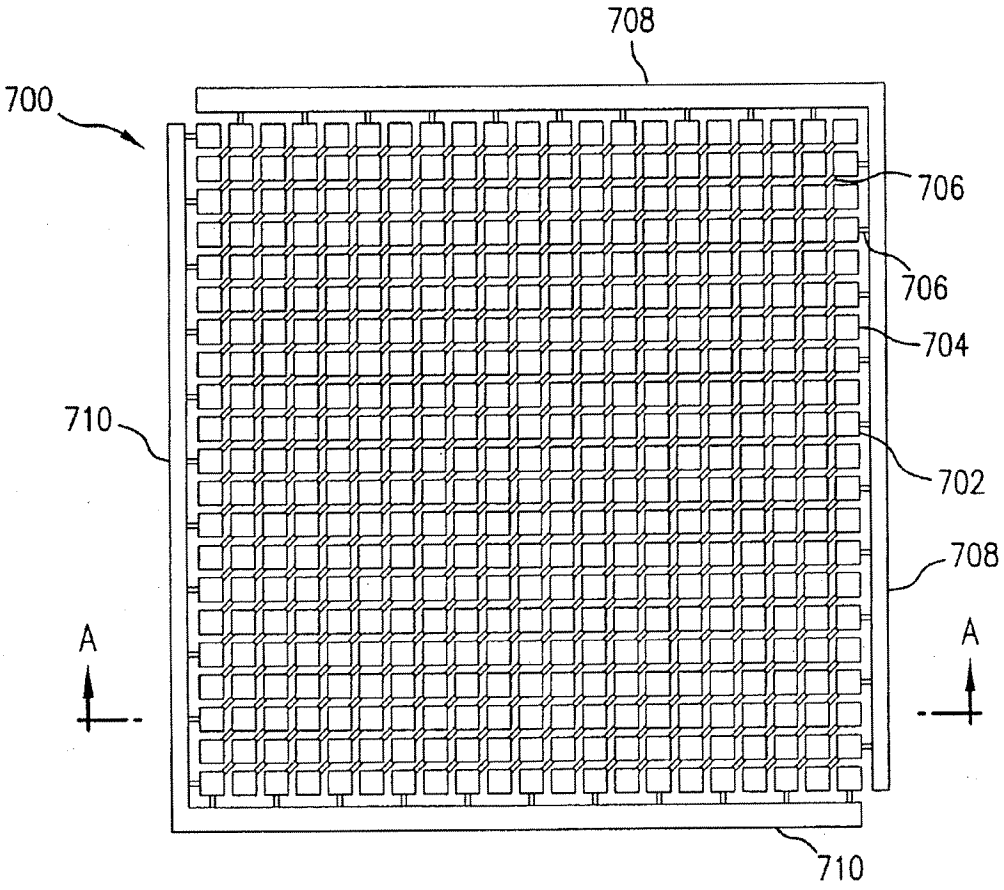


FIG. 7A

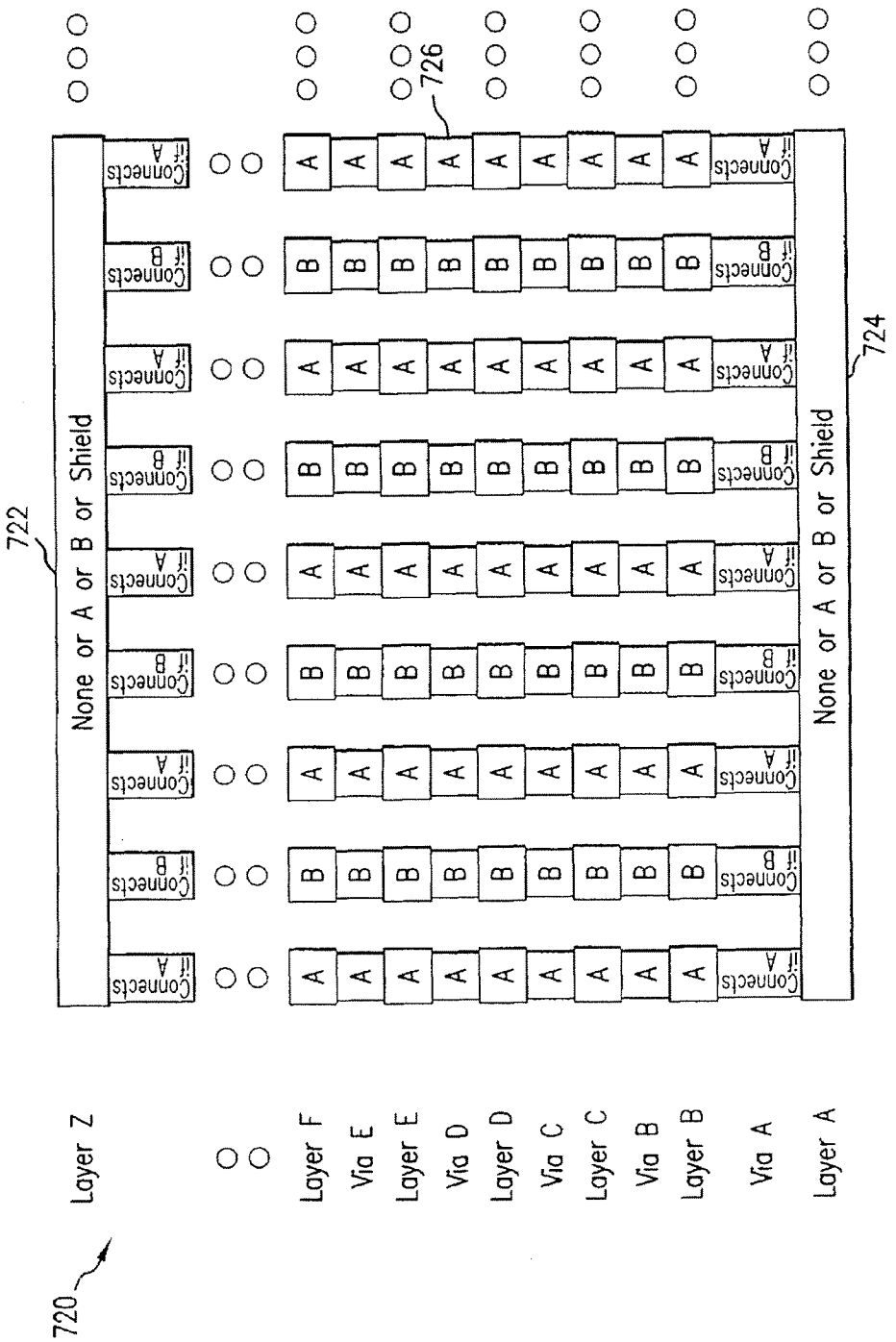

720
722
724
726
Layer Z
Layer F
Via E
Layer E
Via D
Layer D
Via C
Layer C
Via B
Layer B
Via A
Layer A
None or A or B or Shield
None or A or B or Shield
Connects A
Connects B
A
B

FIG. 7B

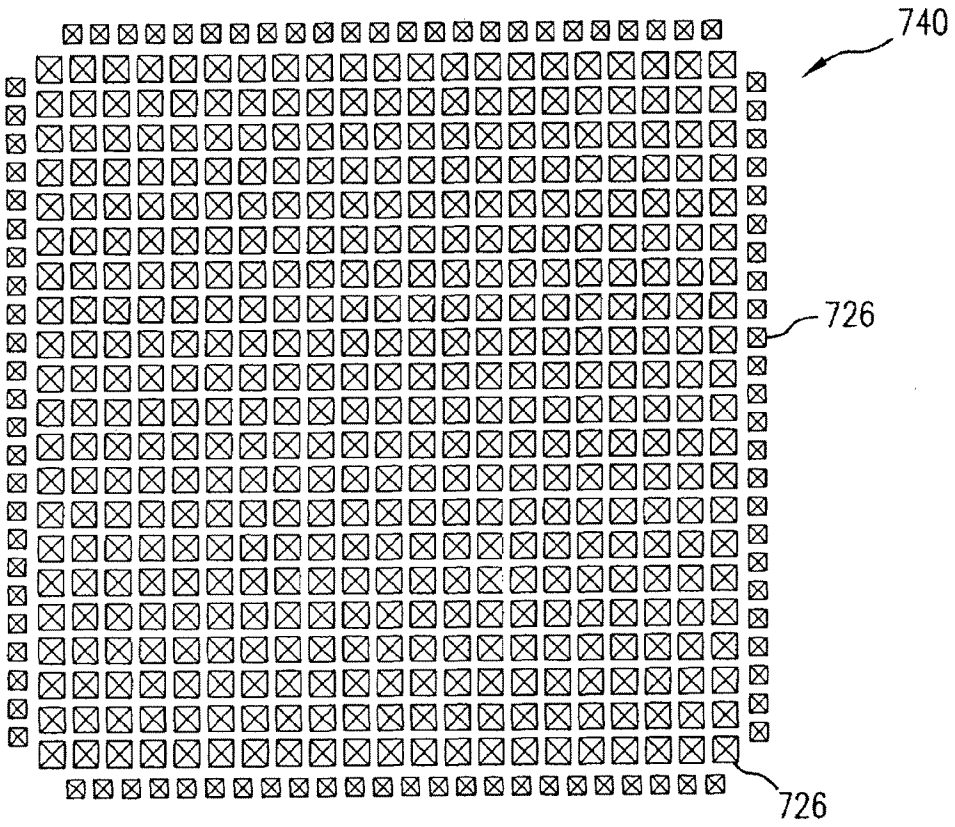


FIG. 7C

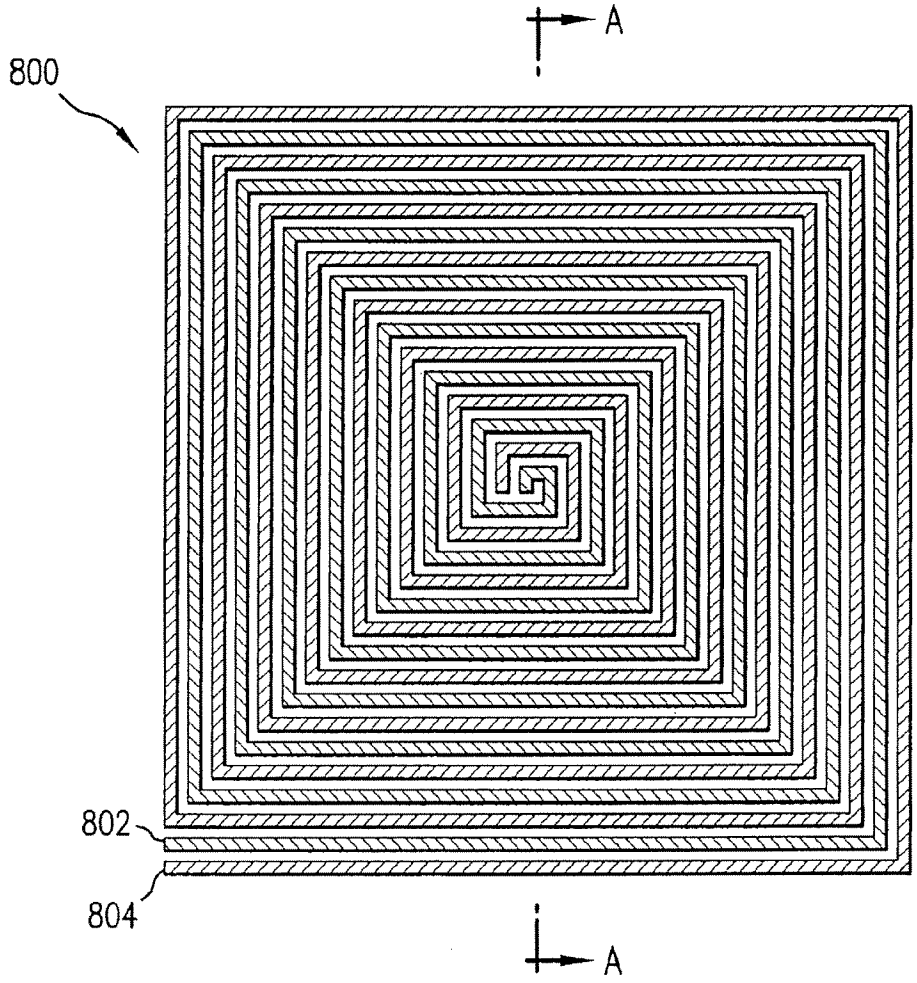


FIG.8A

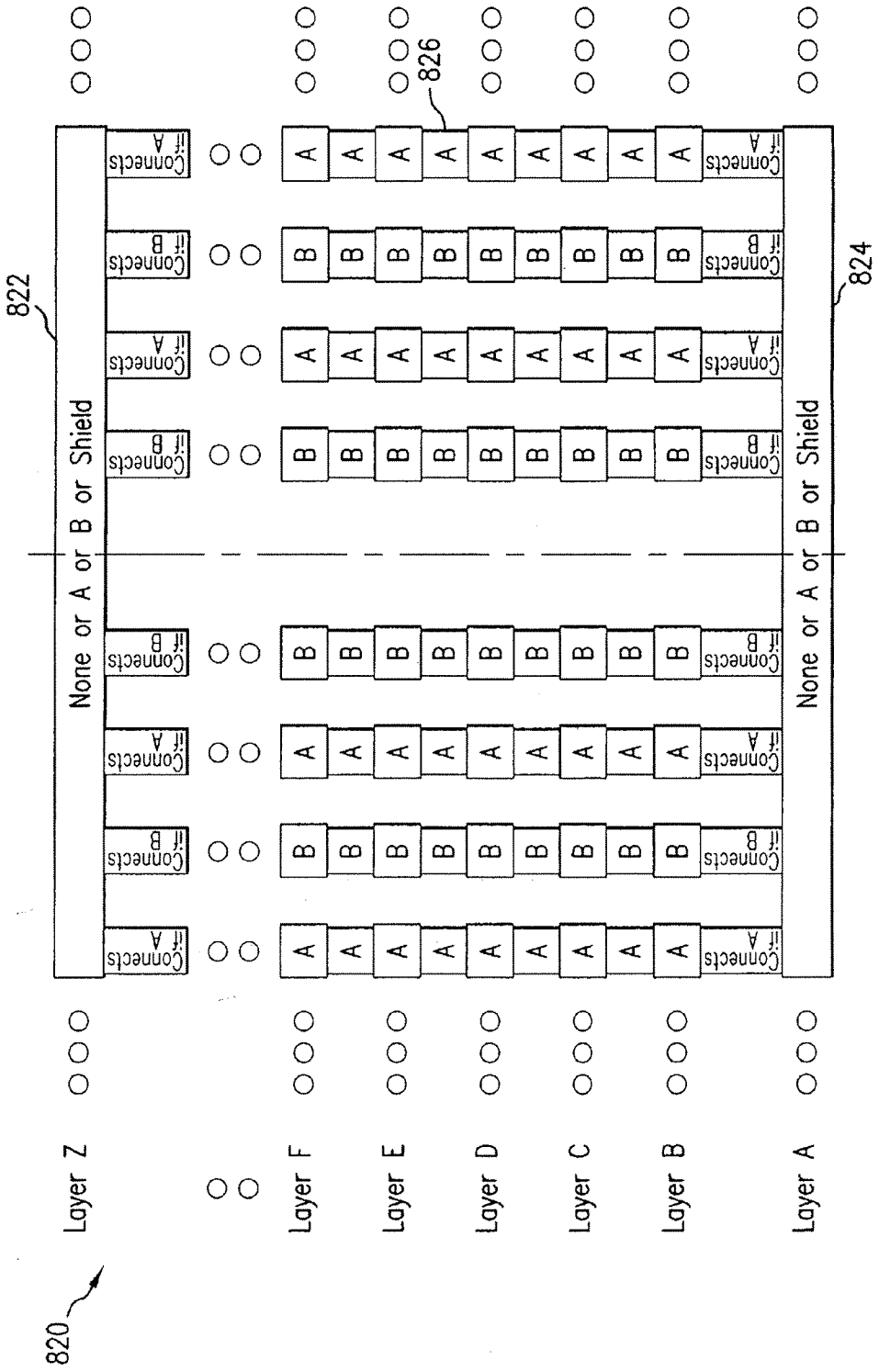

820
822
824
826
Layer Z
Layer F
Layer E
Layer D
Layer C
Layer B
Layer A
None or A or B or Shield
None or A or B or Shield
Connects to A
Connects to B

FIG.8B

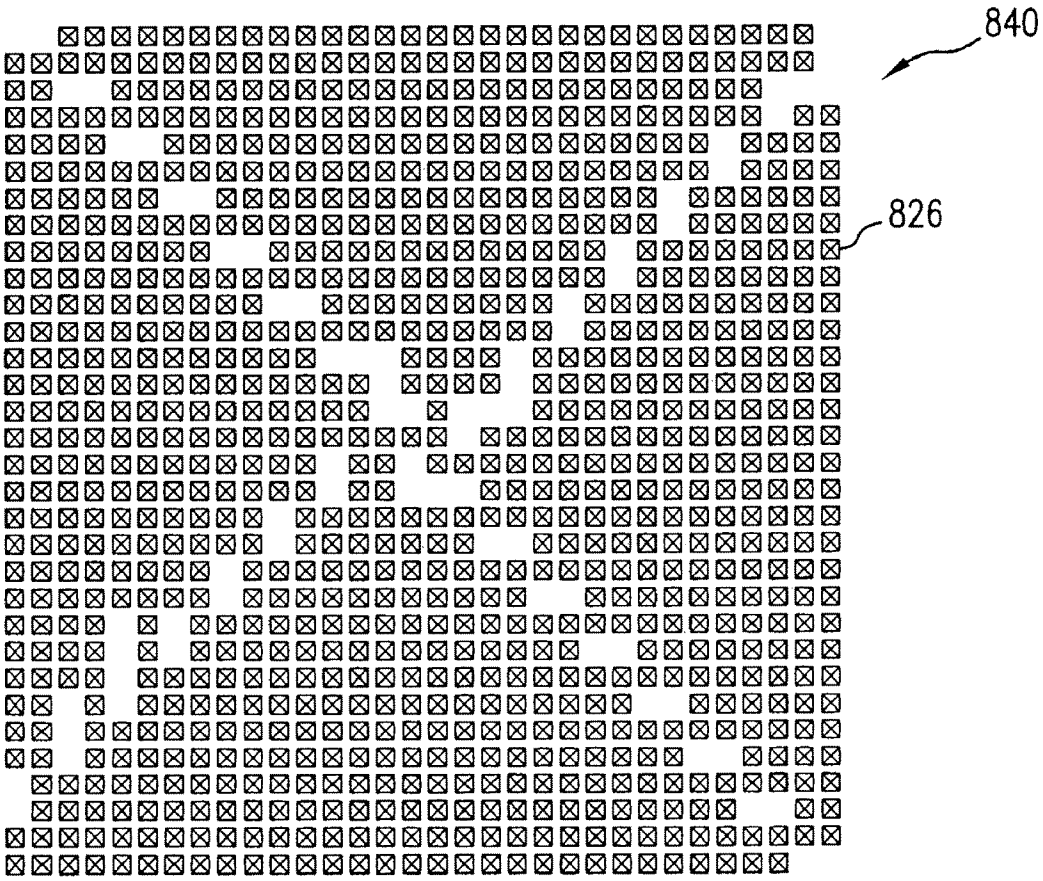


FIG.8C

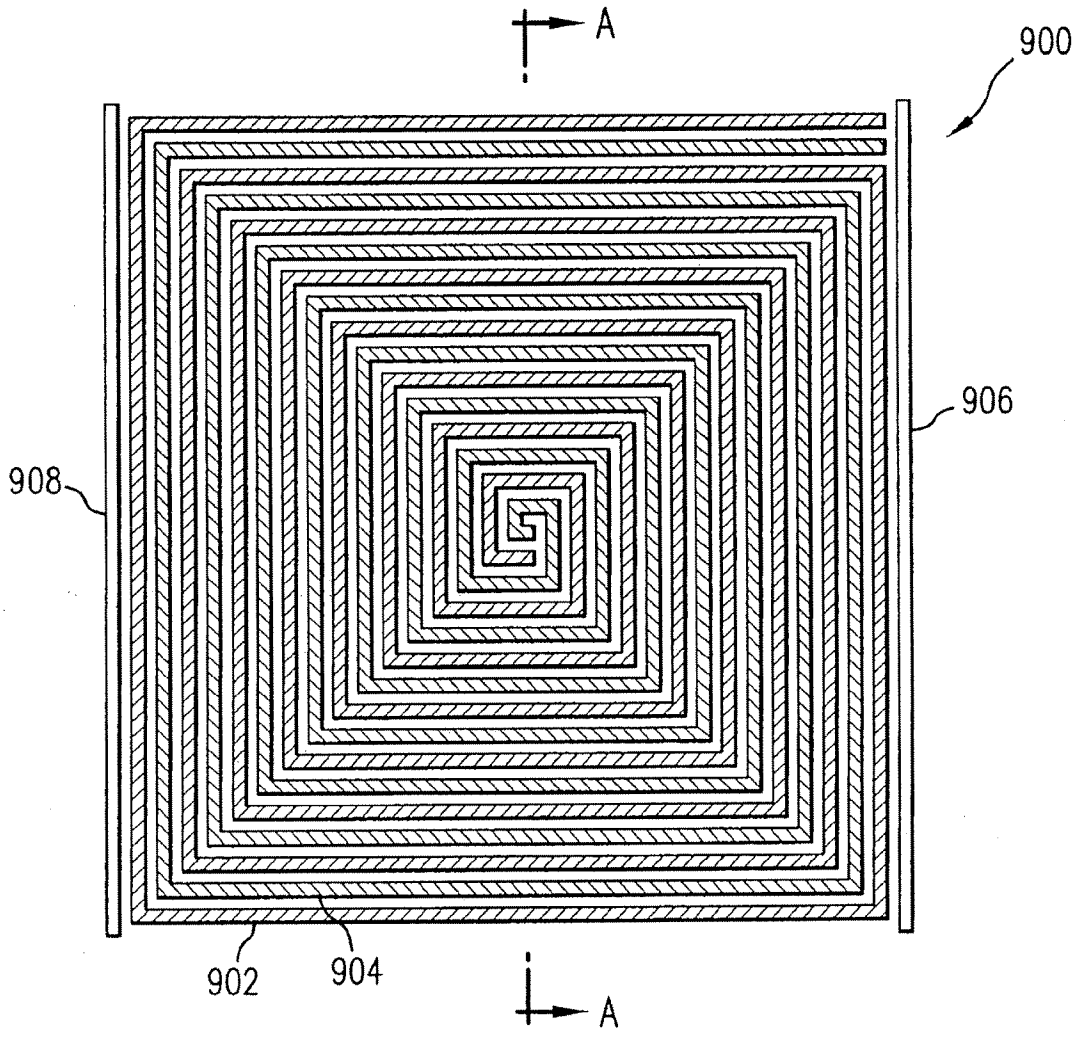


FIG.9A

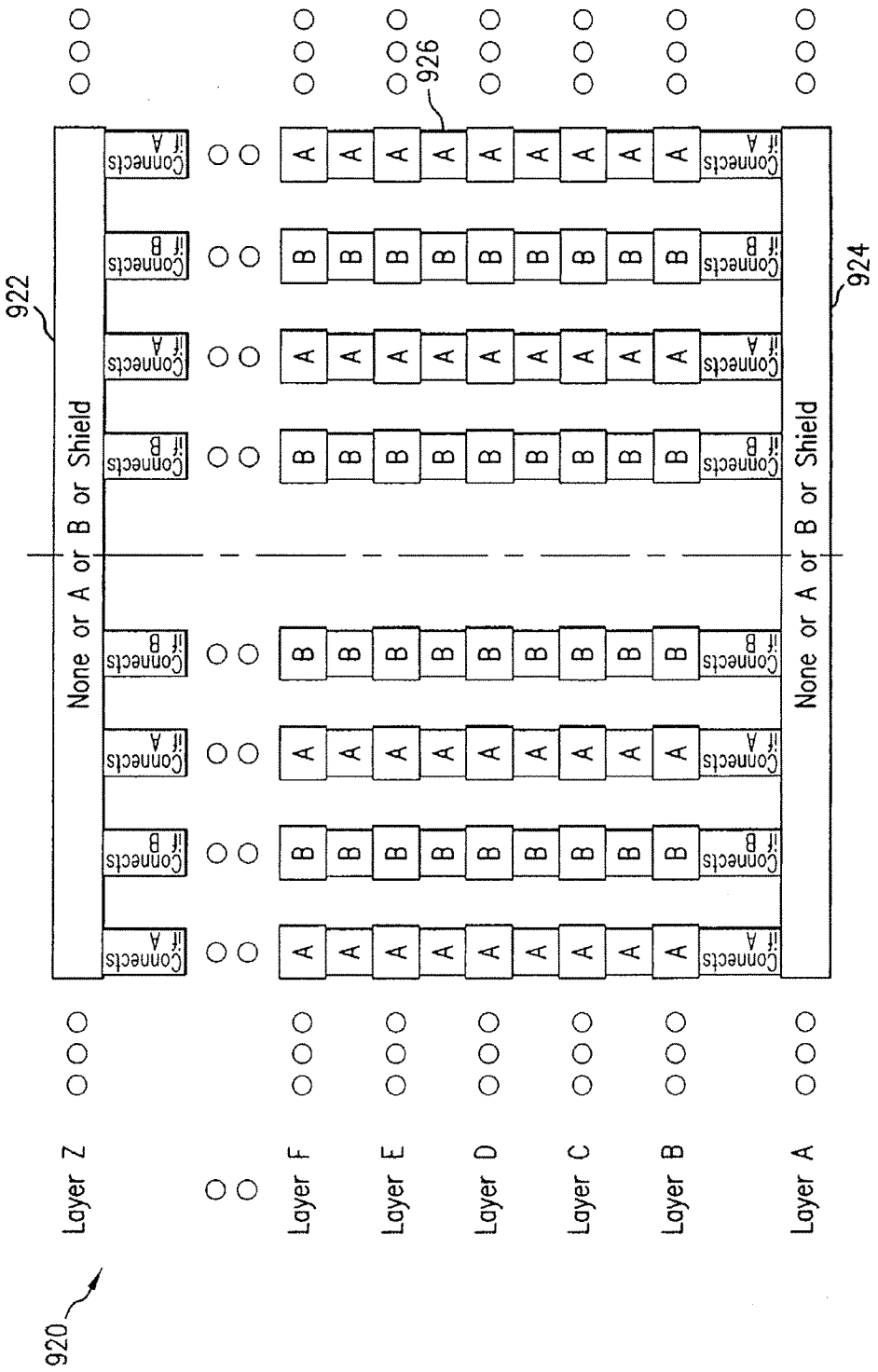

920
922
924
926
Layer Z
Layer F
Layer E
Layer D
Layer C
Layer B
Layer A
None or A or B or Shield
None or A or B or Shield
Connects A
Connects B

FIG.9B

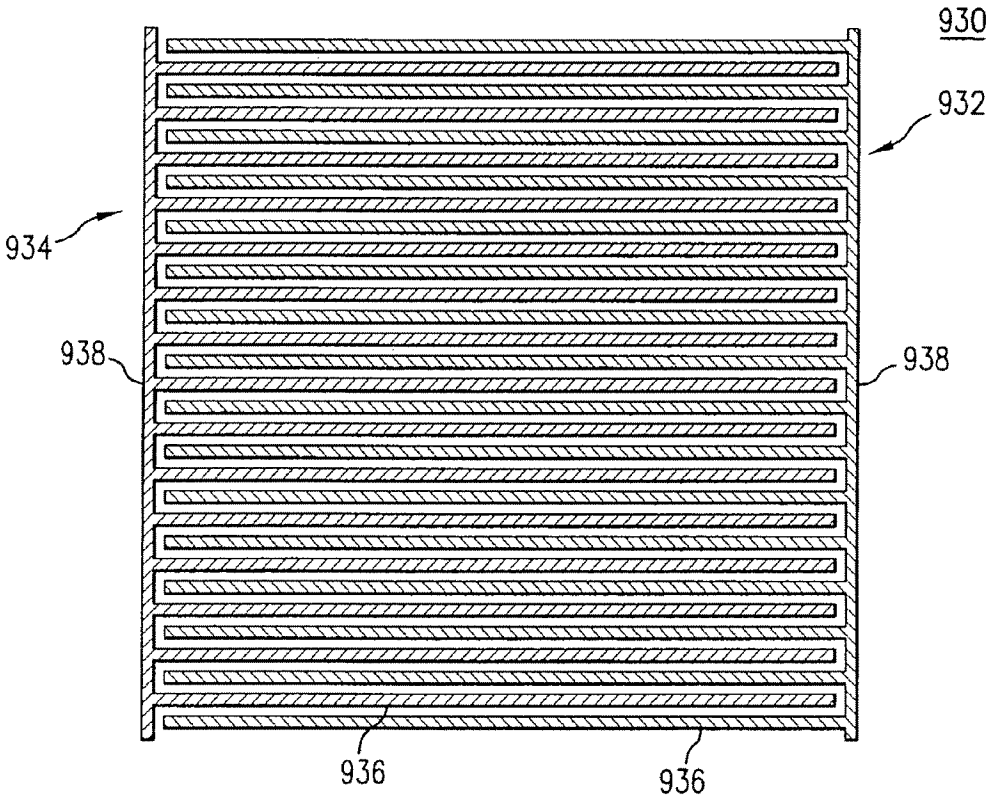


FIG.9C

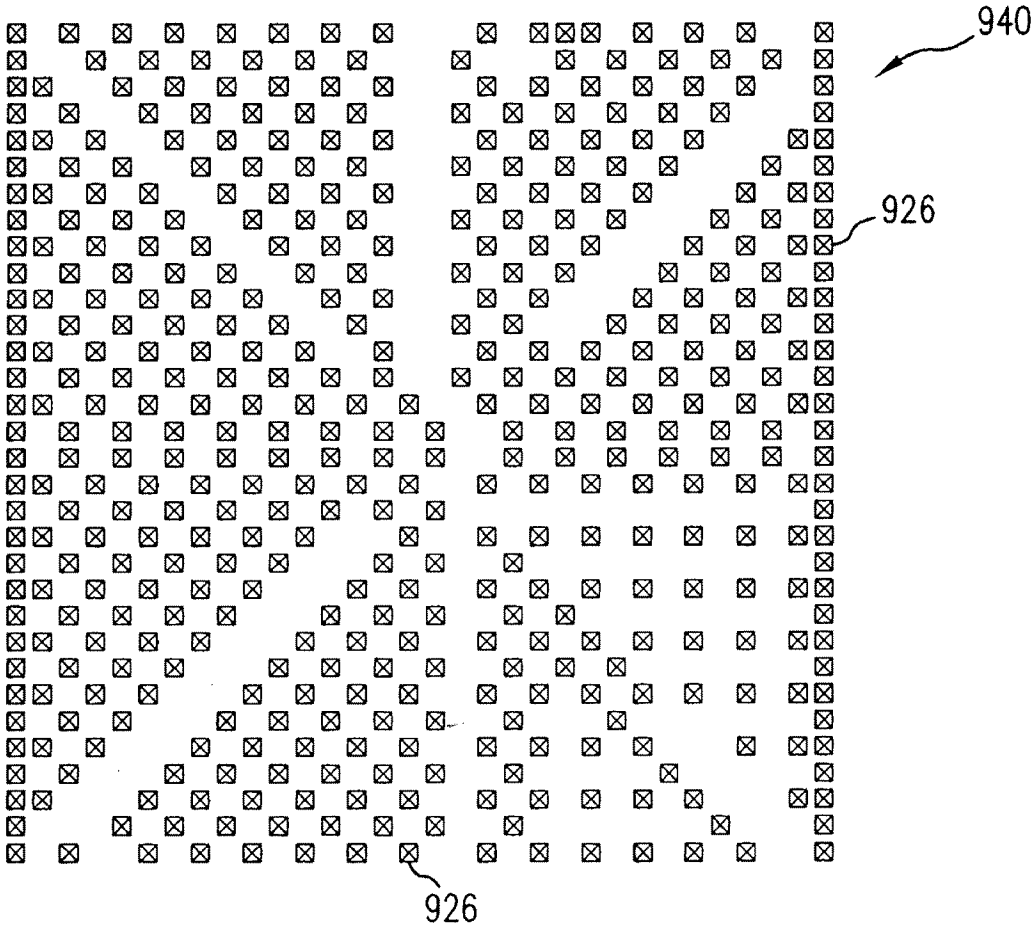


FIG.9D

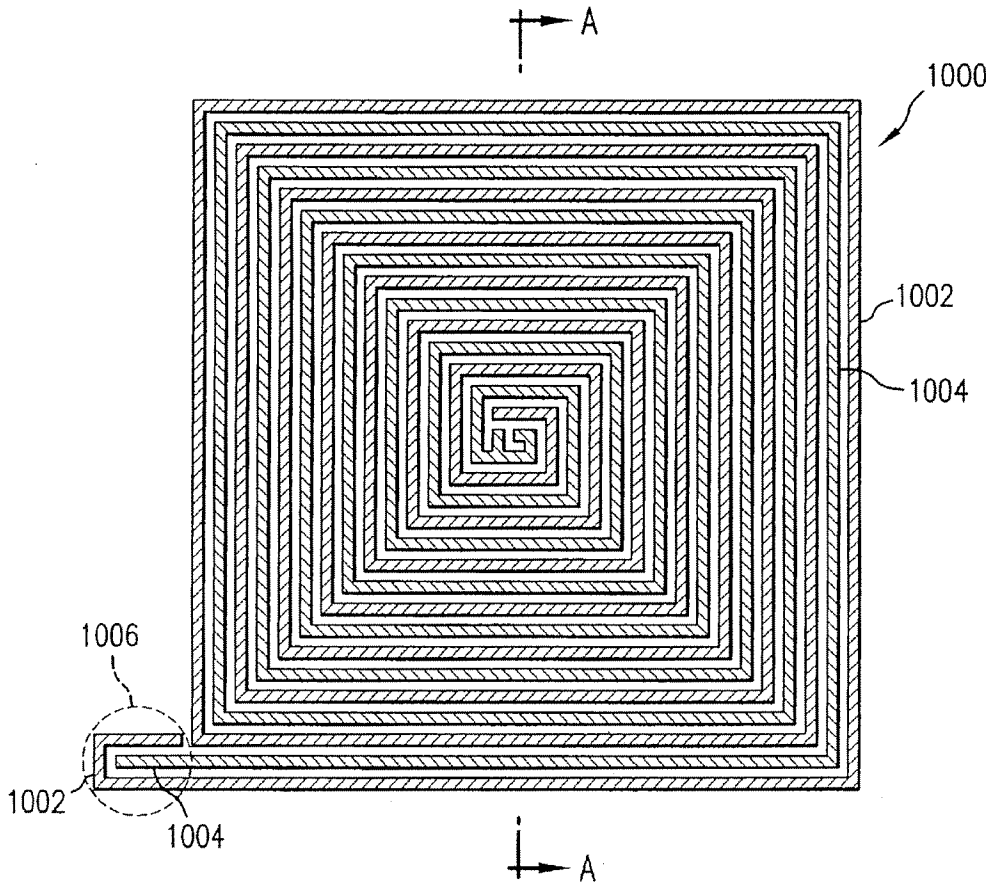


FIG.10A

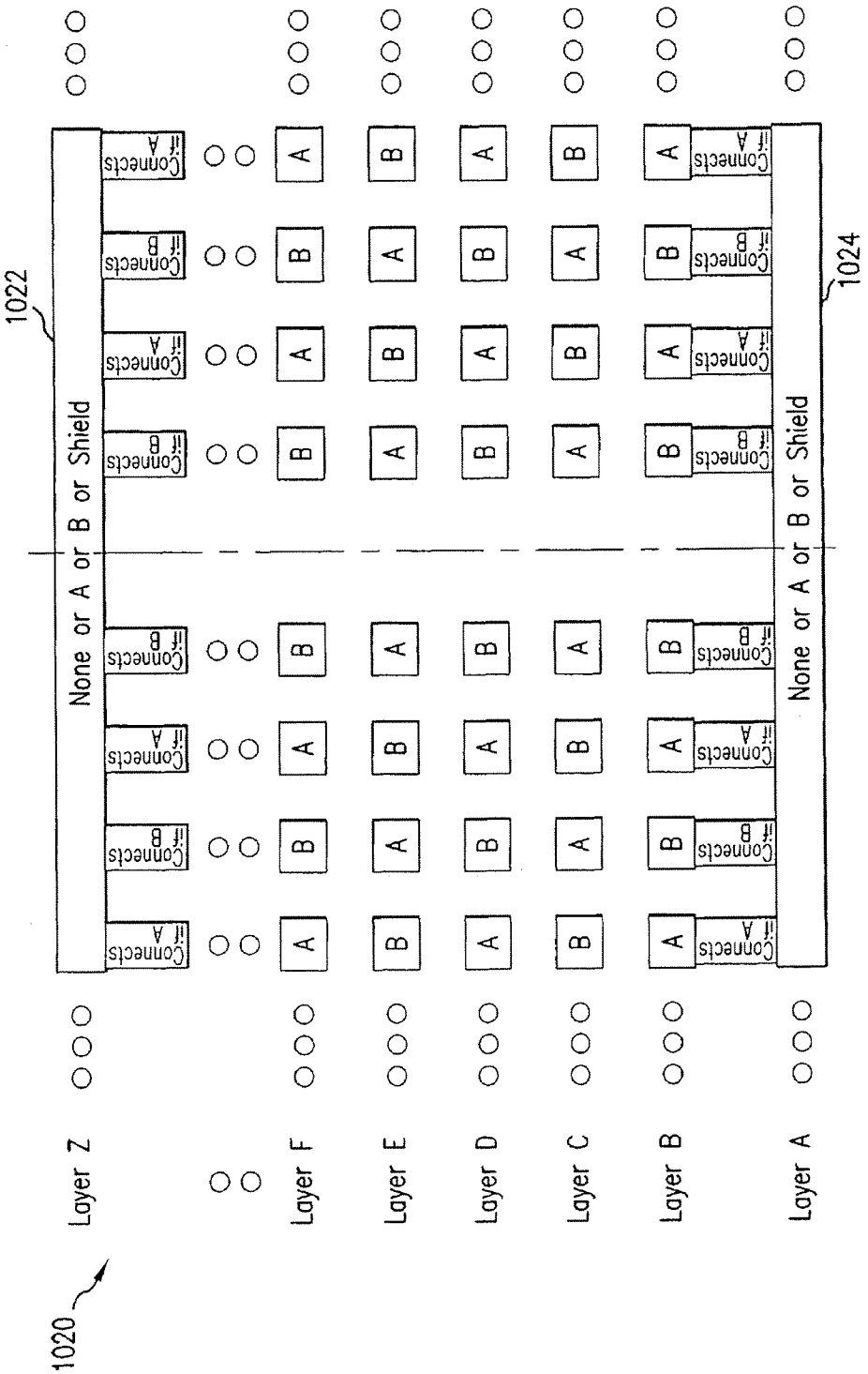

1020
1022
1024
Layer Z
Layer F
Layer E
Layer D
Layer C
Layer B
Layer A
None or A or B or Shield
None or A or B or Shield
A
B
Connects

FIG. 10B

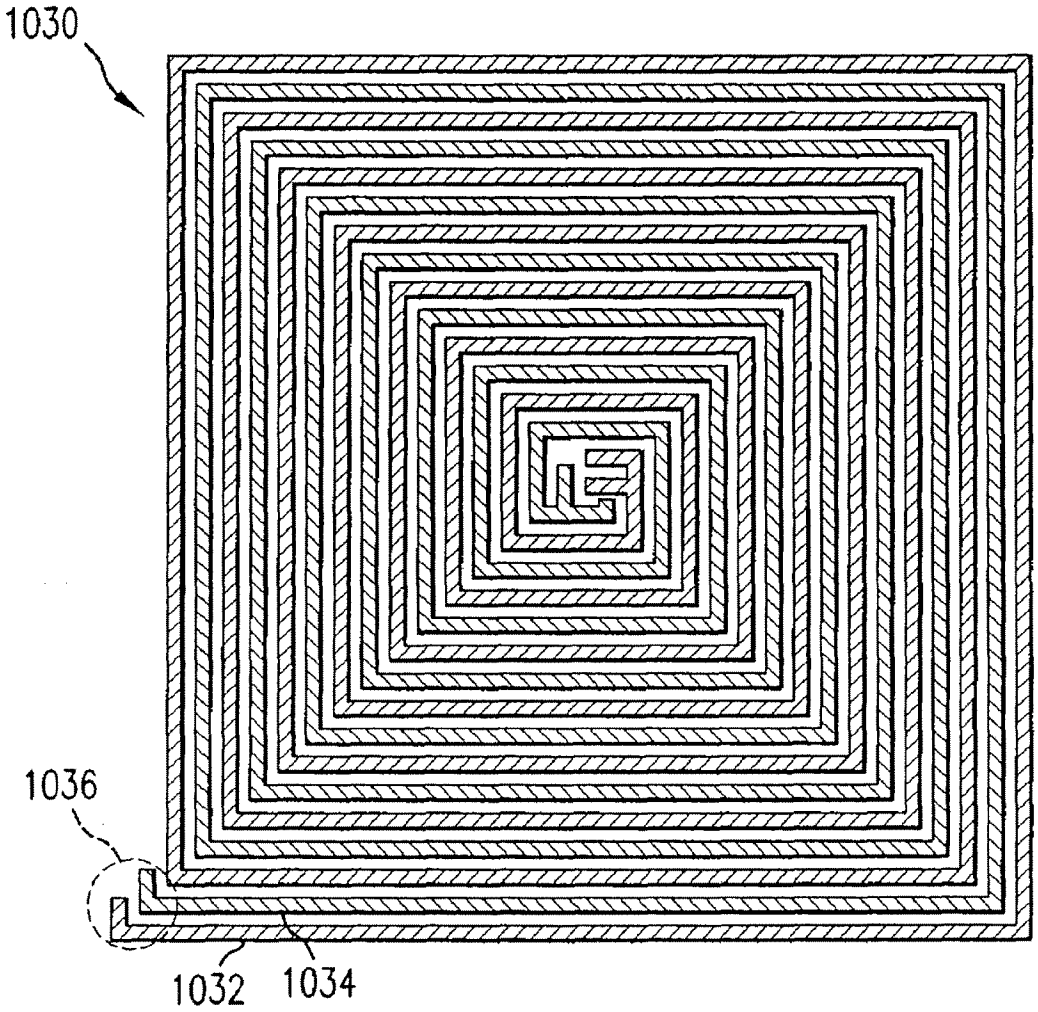


FIG. 10C

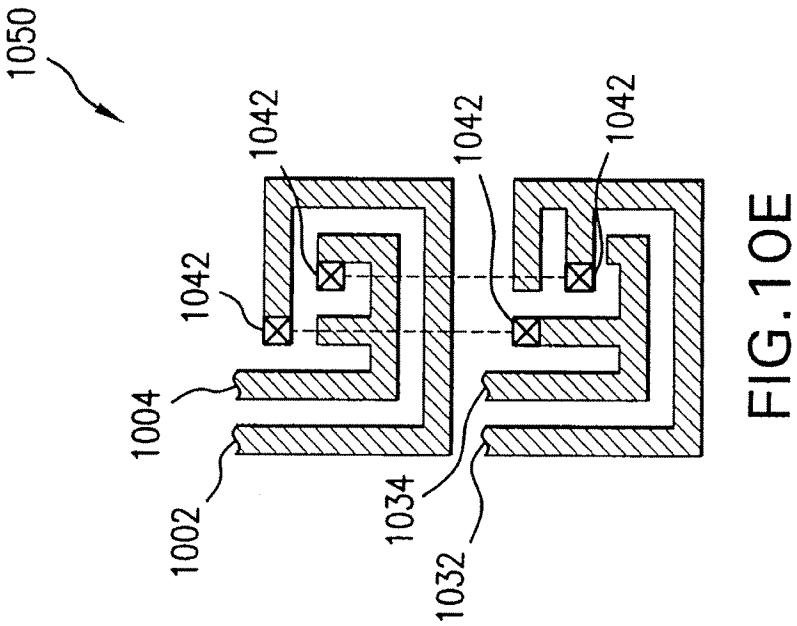


FIG. 10E

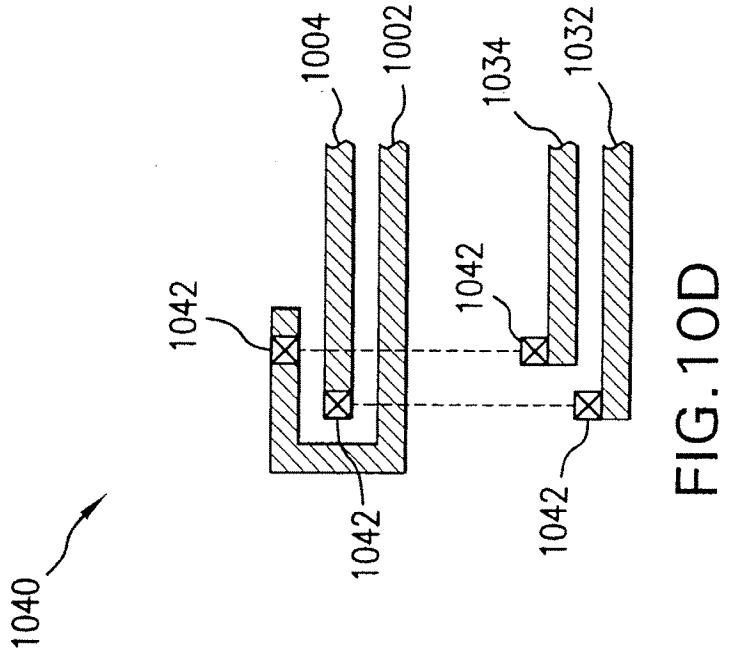


FIG. 10D

SCALABLE INTEGRATED CIRCUIT IIIGII DENSITY CAPACITORS

CROSS REFERENCE TO RELATED APPLICATION

This application is a continuation of U.S. patent application Ser. No. 11/013,789, filed on Dec. 17, 2004, which claims the benefit of U.S. Provisional Patent Application No. 60/530,648, filed Dec. 19, 2003, both of which are incorporated herein by reference in their entirety.

FIELD OF THE INVENTION

The present invention relates to capacitors. More particularly, it relates to integrated circuit capacitors.

BACKGROUND OF THE INVENTION

Advances in semiconductor manufacturing technology have led to the integration of millions of circuit elements on a single integrated circuit. In order to integrate these increasing numbers of circuit elements onto an integrated circuit, it has been necessary to reduce the dimensions of the various component parts, including capacitors, which are basic building blocks for electrical circuits. Therefore, a need exists for new capacitors that improve area utilization and/or integrated circuit performance.

BRIEF SUMMARY OF THE INVENTION

The present invention provides several scalable integrated circuit high density capacitors and their layout techniques. The capacitors are scalable in that they can be formed using a selected number of metal layers and the area of these metal layers is variable.

It is a feature of the present invention that optional shields can be included as the top-most and/or bottom-most layers of the capacitors, as well as one or more sides of the capacitors, to reduce unwanted parasitic capacitance.

Further embodiments, features, and advantages of the present invention, as well as the structure and operation of the various embodiments of the present invention, are described in detail below with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1**A is a schematic diagram depicting a top view of an example metal layer of a scalable integrated circuit stagger-structure high density capacitor according to an embodiment of the present invention.

FIG. **1**B is a schematic diagram depicting a cross sectional view (A-A) of the scalable integrated circuit stagger-structure high density capacitor of FIG. **1**A.

FIG. **1**C is a schematic diagram depicting a top view of an example via layer of the scalable integrated circuit stagger-structure high density capacitor of FIG. **1**A.

FIG. **1**D is a schematic diagram depicting optional side shields for the scalable integrated circuit stagger-structure high density capacitor of FIG. **1**A.

FIG. **2**A is a schematic diagram depicting a top view of an example metal layer of a scalable integrated circuit cross-structure high density capacitor according to an embodiment of the present invention.

FIG. **2**B is a schematic diagram depicting a cross sectional view (A-A) of the scalable integrated circuit cross-structure high density capacitor of FIG. **2**A.

FIG. **2**C is a schematic diagram depicting a cross sectional view (B-B) of the scalable integrated circuit cross-structure high density capacitor of FIG. **2**A.

FIG. **2**D is a schematic diagram depicting a top view of an example via layer of the scalable integrated circuit cross-structure high density capacitor of FIG. **2**A.

FIG. **3**A is a schematic diagram depicting a top view of an example metal layer of a scalable integrated circuit loop-structure high density capacitor according to an embodiment of the present invention.

FIG. **3**B is a schematic diagram depicting a cross sectional view (A-A) of the scalable integrated circuit loop-structure high density capacitor of FIG. **3**A.

FIG. **3**C is a schematic diagram depicting a cross sectional view (B-B) of the scalable integrated circuit loop-structure high density capacitor of FIG. **3**A.

FIG. **3**D is a schematic diagram depicting a top view of an example via layer of the scalable integrated circuit loop-structure high density capacitor of FIG. **3**A.

FIG. **4**A is a schematic diagram depicting a top view of an example metal layer of a scalable integrated circuit weave-structure high density capacitor according to an embodiment of the present invention.

FIG. **4**B is a schematic diagram depicting a cross sectional view (A-A) of the scalable integrated circuit weave-structure high density capacitor of FIG. **4**A.

FIG. **4**C is a schematic diagram depicting a cross sectional view (B-B) of the scalable integrated circuit weave-structure high density capacitor of FIG. **4**A.

FIG. **4**D is a schematic diagram depicting a cross sectional view (C-C) of the scalable integrated circuit weave-structure high density capacitor of FIG. **4**A.

FIG. **4**E is a schematic diagram depicting a top view of an example via layer of the scalable integrated circuit weave-structure high density capacitor of FIG. **4**A.

FIG. **5**A is a schematic diagram depicting a top view of an example metal layer of a first scalable integrated circuit vertical-structure high density capacitor according to an embodiment of the present invention.

FIG. **5**B is a schematic diagram depicting a cross sectional view (A-A) of the first scalable integrated circuit vertical-structure high density capacitor of FIG. **5**A.

FIG. **5**C is a schematic diagram depicting a top view of an example via layer of the first scalable integrated circuit vertical-structure high density capacitor of FIG. **5**A.

FIG. **6**A is a schematic diagram depicting a top view of an example metal layer of a second scalable integrated circuit vertical-structure high density capacitor according to an embodiment of the present invention.

FIG. **6**B is a schematic diagram depicting a cross sectional view (A-A) of the second scalable integrated circuit vertical-structure high density capacitor of FIG. **6**A.

FIG. **6**C is a schematic diagram depicting a top view of an example via layer of the second scalable integrated circuit vertical-structure high density capacitor of FIG. **6**A.

FIG. **7**A is a schematic diagram depicting a top view of an example metal layer of a third scalable integrated circuit vertical-structure high density capacitor according to an embodiment of the present invention.

FIG. **7**B is a schematic diagram depicting a cross sectional view (A-A) of the third scalable integrated circuit vertical-structure high density capacitor of FIG. **7**A.

FIG. **7**C is a schematic diagram depicting a top view of an example via layer of the third scalable integrated circuit vertical-structure high density capacitor of FIG. **7**A.

FIG. **8**A is a schematic diagram depicting a top view of an example metal layer of a first scalable integrated circuit spiral-structure high density capacitor according to an embodiment of the present invention.

FIG. **8**B is a schematic diagram depicting a cross sectional view (A-A) of the first scalable integrated circuit spiral-structure high density capacitor of FIG. **8**A.

FIG. **8**C is a schematic diagram depicting a top view of an example via layer of the first scalable integrated circuit spiral-structure high density capacitor of FIG. **8**A.

FIG. **9**A is a schematic diagram depicting a top view of a first example metal layer of a second scalable integrated circuit spiral-structure high density capacitor according to an embodiment of the present invention.

FIG. **9**B is a schematic diagram depicting a cross sectional view (A-A) of the second scalable integrated circuit spiral-structure high density capacitor of FIG. **9**A.

FIG. **9**C is a schematic diagram depicting a top view of a second example metal layer for the second scalable integrated circuit spiral-structure high density capacitor of FIG. **9**A.

FIG. **9**D is a schematic diagram depicting a top view of an example via layer of the second scalable integrated circuit spiral-structure high density capacitor of FIG. **9**A.

FIG. **10**A is a schematic diagram depicting a top view of a first example metal layer of a third scalable integrated circuit spiral-structure high density capacitor according to an embodiment of the present invention.

FIG. **10**B is a schematic diagram depicting a cross sectional view (A-A) of the third scalable integrated circuit spiral-structure high density capacitor of FIG. **10**A.

FIG. **10**C is a schematic diagram depicting a top view of a second example metal layer for the third scalable integrated circuit spiral-structure high density capacitor of FIG. **10**A.

FIG. **10**D is a schematic diagram depicting a top view of a first portion of an example via layer of the third scalable integrated circuit spiral-structure high density capacitor of FIG. **10**A used to connect the metal layer of FIG. **10**A to the metal layer of FIG. **10**C.

FIG. **10**E is a schematic diagram depicting a top view of a second portion of the example via layer used to connect the metal layer of FIG. **10**A to the metal layer of FIG. **10**C.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention provides scalable integrated circuit high density capacitors and their layout techniques. The figures detail the capacitor structures, graphically. In an embodiment of the present invention, optional shields designated as top-most and/or bottom-most layers of the capacitor structure, and/or optional side shields, are connected to nodes other than the capacitor's terminals (i.e., A or B) to help reduce unwanted parasitic capacitance from terminals A and/or B to other circuit nodes. In an embodiment, optional shields designated as top-most and/or bottom-most layers of the capacitor structure, and/or optional side shields, are connected to one of the capacitor's terminals (e.g., A) to minimize parasitic capacitance from the other capacitor terminal (e.g., B) to other circuit nodes.

FIG. **1**A is a schematic diagram depicting a top view of an example metal layer **100** of a scalable integrated circuit stagger-structure high density capacitor **150** (see FIG. **1**B) according to an embodiment of the present invention. As shown in FIG. **1**A, metal layer **100** includes two metallization patterns **102** and **104**. Metallization pattern **102** has a plurality of finger tracks **106** coupled approximately perpendicular to a backbone track **108**A. Similarly, metallization pattern **104** has a plurality of finger tracks **106** coupled approximately perpendicular to a backbone track **108**B. The fingers **106** of metallization patterns **102** and **104** are interlaced as shown in FIG. **1**A. By approximately perpendicular, it is meant that the finger tracks **106** are sufficiently perpendicular so that they can be interlaced. The metallization patterns **102** and **104** can be formed using any conducting metal. In an embodiment, copper or aluminum is used to form metallization patterns **102** and **104**, but the invention is not limited to these metals.

FIG. **1**B is a schematic diagram depicting a cross sectional view (A-A) (see FIG. **1**A) of the scalable integrated circuit stagger-structure high density capacitor **150**. As shown in FIG. **1**B, capacitor **150** is scalable in that it can be formed using a selected number of metal layers such as, for example, metal layer **100**. Additionally, the area of these metal layers can be varied to meet design requirements. Accordingly, the number and area of the metal layers illustrated for capacitor **150**, or any other capacitor described herein, is not to be used to limit the invention. As shown in FIG. **1**B, adjacent metal layers of capacitor **150** are staggered. This staggering is illustrated by the offset of portion **152** of metal layer F from portion **154** of metal layer E.

Although the width of finger tracks **106** are depicted as being equal in size to the spacing between the finger tracks, in some embodiments, the widths of finger tracks **106** are selected to be larger than the spacing between the finger tracks so that finger tracks of adjacent metal layers overlap. In other embodiments, the widths of finger tracks **106** are selected to be smaller than the spacing between the finger tracks so that finger tracks of adjacent metal layers will not overlap.

Optional shields **156** and/or **158** can be included as the top-most and/or bottom-most layers of capacitor **150** to help reduce unwanted parasitic capacitance. In an embodiment, the shields **156** and/or **158** are connected to nodes other than the terminals (i.e., A and B shown in FIG. **1**B) of capacitor **150**. In another embodiment, shields **156** and/or **158**, designated as top-most and/or bottom-most layers of capacitor **150**, are connected to one of the capacitor's terminals (e.g., A) to minimize parasitic capacitance from the other capacitor terminal (e.g., B) to other circuit nodes. Any type of shield, including side shields (see FIG. **1**D), can be used in accordance with the present invention. In embodiments, the shields **156** and **158** are formed, for example, using outer stripes of all conductive layers, metal plates, or any other known structure that can serve as a shield.

FIG. **1**C is a schematic diagram depicting a top view of an example via layer **180** of the scalable integrated circuit stagger-structure high density capacitor **150**. Via layer **180** includes a plurality of vias **182** used to couple adjacent metal layers of capacitor **150** such as, for example, two metal layers having the metallization patterns shown in FIG. **1**A. The vias **182** shown in FIG. **1**C couple together backbone tracks **108** of adjacent metal layers. The number and shape of the vias shown in FIG. **1**C are only illustrative and not limiting. Via layers having more or less vias than shown in FIG. **1**C can be used as well as vias of different shapes and/or sizes.

FIG. **1**D illustrates optional side shields **192** and **194** for high density capacitor **150**. In the embodiment shown in FIG. **1**D, side shields **192** and **194** are formed using metal layers which have end finger tracks, such as end finger tracks **195***a* and **195***b*, that are associated with a particular terminal of capacitor **150** (e.g., A). It is noted here, however, that this embodiment is only illustrative and not limiting. Any type of side shields can be used. For example, in one embodiment, the width of the end finger tracks **195** are varied so that the finger

tracks overlap, either partially or completely. In another embodiment, side tracks are formed using stacked vias rather than end finger tracks.

In embodiments of the present invention, the optional side shields are connected to nodes other than the terminals (i.e., A and B) of capacitor **150**. In one embodiment, the shields are connected to one of the capacitor's terminals (e.g., A as shown in FIG. 1D) to minimize parasitic capacitance from the other capacitor terminal (e.g., B) to other circuit nodes.

FIG. **2**A is a schematic diagram depicting a top view of an example metal layer **200** of a scalable integrated circuit cross-structure high density capacitor **220** (see FIG. 2B) according to an embodiment of the present invention. As shown in FIG. **2**A, metal layer **200** includes two metallization patterns **202** and **204**. Metallization pattern **202** has a plurality of finger tracks **206** coupled approximately perpendicular to backbone track **208**A. Metallization pattern **204** has a plurality of finger tracks **206** coupled approximately perpendicular to backbone track **208**B. The finger tracks **206** of metallization patterns **202** and **204** are interlaced as shown in FIG. **2**A. As with capacitor **150**, the metallization patterns **202** and **204** can be formed using any conducting metal. For brevity, it is noted here that the metallization patterns for each of the capacitors described herein can be formed using any conducting metal so that this feature need not be repeated below in the description of other capacitor according to the present invention.

FIG. **2**B is a schematic diagram depicting a cross sectional view (A-A) (see FIG. **2**A) of the scalable integrated circuit cross-structure high density capacitor **220**. As shown in FIG. **2**B, capacitor **220** is scalable in that it can be formed using a selected number of metal layers such as, for example, metal layer **200**, and the area of these metal layers can be varied, as is the case for each capacitor described herein. Optional shields **222** and/or **224** can be included as the top-most and/or bottom-most layers of capacitor **220**, as well as optional side shields (not shown), to help reduce unwanted parasitic capacitance. Any type of shield can be used with capacitor **220** in accordance with the present invention, as with all the capacitors according to the present invention.

FIG. **2**C is a schematic diagram depicting a cross sectional view (B-B) (see FIG. **2**A) of the scalable integrated circuit cross-structure high density capacitor **220**. FIG. **2**B and FIG. **2**C illustrate the relationship of the metal layers that make up capacitor **220**. These figures also illustrate how the metal layers of capacitor **220** are coupled together by vias. The A's and B's shown in FIG. **2**B and FIG. **2**C depict whether a part is associated with a terminal A or a terminal B of capacitor **220**.

FIG. **2**D is a schematic diagram depicting a top view of an example via layer **240** of the scalable integrated circuit cross-structure high density capacitor **220**. As shown in FIG. **2**D, in an embodiment, both the finger tracks **206** and the backbone tracks **208** of selected metal layers are coupled together by vias **242**.

FIG. **3**A is a schematic diagram depicting a top view of an example metal layer **300** of a scalable integrated circuit loop-structure high density capacitor **320** (see FIG. 3B) according to an embodiment of the present invention. As shown in FIG. **3**A, metal layer **300** includes a metallization pattern **302** and a plurality of metallization patterns **304**. The metallization pattern **302** has loop openings **306** in which the metallization patterns **304** are located. One metallization pattern **304** is located in each loop opening **306** of metallization pattern **302**. The shapes of the metallization patterns and loop openings shown in FIG. **3**A are illustrative only. Other shapes and/or sizes can be used.

FIG. **3**B is a schematic diagram depicting a cross sectional view (A-A) (see FIG. **3**A) of the scalable integrated circuit loop-structure high density capacitor **320**. As shown in FIG. **3**B, capacitor **320** is scalable in that it can be formed using a selected number of metal layers such as, for example, metal layer **300**, and the area of these metal layers can be varied. Optional shields **321** and/or **323** can be included as the top-most and/or bottom-most layers of capacitor **320**, as well as optional side shields, to help reduce unwanted parasitic capacitance. Vias **322** couple one metal layer to another metal layer as illustrated, for example, in regions **324** and **326** of capacitor **320**. In region **324**, two vias **322** are shown coupling a metallization pattern **302** of metal layer D to a metallization pattern **304** in metal layer E and a metallization pattern **304** in metal layer C. In region **326**, two vias **322** are shown coupling a metallization pattern **302** of metal layer C to a metallization pattern **304** in metal layer D and a metallization pattern **304** in metal layer B.

FIG. **3**C is a schematic diagram depicting a second cross sectional view (B-B) (see FIG. **3**A) of the scalable integrated circuit loop-structure high density capacitor **320**. FIG. **3**C further illustrates how vias **322** couple one metal layer of capacitor **320** to another metal layer. In region **340** of capacitor **320**, two vias **322** are shown coupling a metallization pattern **302** of metal layer C to a metallization pattern **304** in metal layer D and a metallization pattern **304** in metal layer B. In region **342**, two vias **322** are shown coupling a metallization pattern **302** of metal layer D to a metallization pattern **304** in metal layer E and a metallization pattern **304** in metal layer C.

FIG. **3**D is a schematic diagram depicting a top view of an example via layer **360** of the scalable integrated circuit loop-structure high density capacitor **320**. Via layer **360** includes a plurality of vias **322** used to couple adjacent metal layers of capacitor **320**.

FIG. **4**A is a schematic diagram depicting a top view of an example metal layer **400** of a scalable integrated circuit weave-structure high density capacitor **420** (see FIG. **4**B) according to an embodiment of the present invention. As shown in FIG. **4**A, metal layer **400** includes a metallization pattern **402** and a metallization pattern **404**. The metallization patterns **402** and **404** each have a plurality of finger tracks **406** coupled approximately perpendicular to backbone tracks **408**. The finger tracks **406** of metallization patterns **402** and **404** are interlaced as shown in FIG. **4**A. Additionally, each finger track **406** has a plurality of spur tracks **410**. The spur tracks **410** are coupled to two sides of a finger track **406**, except for finger tracks at an end of a backbone track **408**, as shown in FIG. **4**A. The spur tracks **410** are coupled approximately perpendicular to a finger track **406** so that spur tracks **410** of adjacent finger tracks **406** interlace. By approximately perpendicular, it is meant that the finger tracks **406** and the spur tracks **410** are sufficiently perpendicular so that they can be interlaced as illustrated in FIG. **4**A.

FIG. **4**B is a schematic diagram depicting a cross sectional view (A-A) (see FIG. **4**A) of the scalable integrated circuit weave-structure high density capacitor **420**. As shown in FIG. **4**B, capacitor **420** is scalable in that it can be formed using a selected number of metal layers such as, for example, metal layer **400**, and the area of these metal layers can be varied. Optional shields **422** and/or **424** can be included as the top-most and/or bottom-most layers of capacitor **420**, as well as optional side shields, to help reduce unwanted parasitic capacitance. Vias **425** couple one metal layer to another metal layer.

FIG. **4**C is a schematic diagram depicting a cross sectional view (B-B) (see FIG. **4**A) of the scalable integrated circuit weave-structure high density capacitor **420**. FIG. **4**C illustrates how vias **425** in a region **430** of capacitor **420** couple the spur tracks **410** of finger tracks **406** in one metal layer to the spur tracks **410** of the finger tracks **406** of other metal layers. Region **430** also illustrates how portions of capacitor **420** associated with a terminal A of capacitor **420** loop around a portion of capacitor **420** associated with a terminal B.

FIG. 4D is a schematic diagram depicting a cross sectional view (C-C) (see FIG. 4A) of the scalable integrated circuit weave-structure high density capacitor **420**. FIG. 4D illustrates how in a region **440** of capacitor **420** portions of capacitor **420** associated with terminal B loop around a portion of capacitor **420** associated with terminal A.

FIG. 4E is a schematic diagram depicting a top view of an example via layer **450** of the scalable integrated circuit weave-structure high density capacitor **420**. Via layer **450** includes a plurality of vias **425** used to couple adjacent metal layers of capacitor **420**.

FIG. 5A is a schematic diagram depicting a top view of an example metal layer **500** of a first scalable integrated circuit vertical-structure high density capacitor **520** (see FIG. 5B) according to an embodiment of the present invention. Metal layer **500** includes a plurality of first metallization patterns **502** and a plurality of second metallization patterns **504** interspersed or arranged in a checkerboard pattern, as shown in FIG. 5A. The metallization patterns **502** and **504** are depicted as being square, but other shapes can be used.

FIG. 5B is a schematic diagram depicting a cross sectional view (A-A) (see FIG. 5A) of the first scalable integrated circuit vertical-structure high density capacitor **520**. As shown in FIG. 5B, capacitor **520** is scalable in that it can be formed using a selected number of metal layers such as, for example, metal layer **500**, and the area of these metal layers can be varied. Optional shields **522** and/or **524** can be included as the top-most and/or bottom-most layers of capacitor **520**, as well as optional side shields, to help reduce unwanted parasitic capacitance. Vias **526** couple one metal layer to another metal layer as illustrated. The portions of capacitor **520** labeled A are associated with an A terminal of capacitor **520**, and the portions labeled B are associated with a B terminal of capacitor **520**. Similar labeling is also used to identify portions of a capacitor associated with the A and/or B terminals of the capacitor in other figures (see for example FIG. 1B and FIG. 10B)

FIG. 5C is a schematic diagram depicting a top view of an example via layer **540** of the first scalable integrated circuit vertical-structure high density capacitor **520**. Via layer **540** includes a plurality of vias **526** used to couple adjacent metal layers of capacitor **520**.

FIG. 6A is a schematic diagram depicting a top view of an example metal layer **600** of a second scalable integrated circuit vertical-structure high density capacitor **620** (see FIG. 6B) according to an embodiment of the present invention. Metal layer **600** includes a plurality of first metallization patterns **602** and a plurality of second metallization patterns **604** interspersed or arranged in a checkerboard pattern, similar to that shown in FIG. 5A. The metallization patterns **602** are coupled together and to a backbone track **608** by connecting tracks **606**, as depicted in FIG. 6A. The metallization patterns **604** also are coupled together and to a backbone track **610** by connecting tracks **606**. The metallization patterns **602** and **604** are shown as being square, but other shapes can be used.

FIG. 6B is a schematic diagram depicting a cross sectional view (A-A) (see FIG. 6A) of the second scalable integrated circuit vertical-structure high density capacitor **620**. As shown in FIG. 6B, capacitor **620** is scalable in that it can be formed using a selected number of metal layers such as, for example, metal layer **600**, and the area of these metal layers can be varied. Optional shields **622** and/or **624** can be included as the top-most and/or bottom-most layers of capacitor **620**, as well as optional side shields, to help reduce unwanted parasitic capacitance. Vias **626** couple one metal layer to another metal layer as illustrated in FIG. 6B. In one embodiment, the metallization patterns **602** and **604** are coupled such that vertically aligned metallization patterns of two adjacent metal layers belong to the same terminal of capacitor **620**. In another embodiment, the metallization patterns **602** and **604** are coupled such that vertically aligned metallization patterns of two adjacent metal layers belong to different terminals of capacitor **620**.

FIG. 6C is a schematic diagram depicting a top view of an example via layer **640** of the second scalable integrated circuit vertical-structure high density capacitor **620**. Via layer **640** includes a plurality of vias **626** used to couple adjacent metal layers of capacitor **620**. As shown, the vias **626** couple both the metallization patterns **602** and **604** as well as the backbone tracks **608** and **610**.

FIG. 7A is a schematic diagram depicting a top view of an example metal layer **700** of a third scalable integrated circuit vertical-structure high density capacitor **720** (see FIG. 7B) according to an embodiment of the present invention. Metal layer **700** includes a plurality of first metallization patterns **702** and a plurality of second metallization patterns **704** interspersed or arranged in a checkerboard pattern, similar to that shown in FIG. 5A. The metallization patterns **702** are coupled together and to a backbone track **708** by connecting tracks **706**, as depicted in FIG. 7A. The metallization patterns **704** also are coupled together and to a backbone track **710** by connecting tracks **706**. The metallization patterns **702** and **704** are shown as being square, but other shapes also can be used.

FIG. 7B is a schematic diagram depicting a cross sectional view (A-A) (see FIG. 7A) of the third scalable integrated circuit vertical-structure high density capacitor **720**. As shown in FIG. 7B, capacitor **720** is scalable in that it can be formed using a selected number of metal layers such as, for example, metal layer **700**, and the area of these metal layers can be varied. Optional shields **722** and/or **724** can be included as the top-most and/or bottom-most layers of capacitor **720**, as well as optional side shields, to help reduce unwanted parasitic capacitance. Vias **726** couple one metal layer to another metal layer as illustrated in FIG. 7B.

FIG. 7C is a schematic diagram depicting a top view of an example via layer **740** of the third scalable integrated circuit vertical-structure high density capacitor **720**. Via layer **740** includes a plurality of vias **726** used to couple adjacent metal layers of capacitor **720**. As shown, the vias **726** couple both the metallization patterns **702** and **704** as well as the backbone tracks **708** and **710**.

FIG. 8A is a schematic diagram depicting a top view of an example metal layer **800** of a first scalable integrated circuit spiral-structure high density capacitor **820** (see FIG. 8B) according to an embodiment of the present invention. Metal layer **800** includes a first metallization pattern **802** and a second metallization pattern **804**. The metallization patterns **802** and **804** are interleaved with one another in a spiral arrangement as depicted in FIG. 8A.

FIG. 8B is a schematic diagram depicting a cross sectional view (A-A) (see FIG. 8A) of the first scalable integrated circuit spiral-structure high density capacitor **820**. As shown in FIG. 8B, capacitor **820** is scalable in that it can be formed using a selected number of metal layers such as, for example, metal layer **800**, and the area of these metal layers can be varied. Optional shields **822** and/or **824** can be included as the top-most and/or bottom-most layers of capacitor **820**, as well as optional side shields, to help reduce unwanted parasitic capacitance. Vias **826** couple one metal layer to another metal layer as illustrated in FIG. 8B. The portions of capacitor **820** labeled A are associated with an A terminal of capacitor **820**, and the portions labeled B are associated with a B terminal of capacitor **820**.

FIG. 8C is a schematic diagram depicting a top view of an example via layer **840** of the first scalable integrated circuit spiral-structure high density capacitor **820**. Via layer **840** includes a plurality of vias **826** used to couple adjacent metal

layers of capacitor **820**. The number of vias shown in FIG. **8**C is exemplary, as is the case for the other via layer figures described herein.

FIG. **9**A is a schematic diagram depicting a top view of an example metal layer **900** of a second scalable integrated circuit spiral-structure high density capacitor **920** (see FIG. **9**B) according to an embodiment of the present invention. Metal layer **900** includes a first metallization pattern **902** and a second metallization pattern **904**. The metallization patterns **902** and **904** are interleaved with one another in a spiral arrangement as depicted in FIG. **9**A. In addition, metal layer **900** includes two border tracks **906** and **908** located proximate to two sides of the interleaved metallization patterns **902** and **904**.

FIG. **9**B is a schematic diagram depicting a cross sectional view (A-A) (see FIG. **9**A) of the second scalable integrated circuit spiral-structure high density capacitor **920**. As shown in FIG. **9**B, capacitor **920** is scalable in that it can be formed using a selected number of metal layers such as, for example, metal layer **900**, and the area of these metal layers can be varied. Optional shields **922** and/or **924** can be included as the top-most and/or bottom-most layers of capacitor **920**, as well as optional side shields, to help reduce unwanted parasitic capacitance. Vias **926** couple one metal layer to another metal layer as illustrated in FIG. **9**B.

FIG. **9**C is a schematic diagram depicting a top view of an example metal layer **930** for the second scalable integrated circuit spiral-structure high density capacitor **920**. Metal layer **930** has a cross bar shape that can be used to connect the inner parts of capacitor **920**. Metal layer **930** also can be used, for example, for the shield **922** or **924**. Alternatively, optional shields **922** and **924** can be formed using a solid shape, like a plate.

As shown in FIG. **9**C, metal layer **930** includes two metallization patterns **932** and **934**. Metallization pattern **932** has a plurality of finger tracks **936** coupled approximately perpendicular to a backbone track **938**. Similarly, metallization pattern **934** has a plurality of finger tracks **936** coupled approximately perpendicular to a second backbone track **938**. The fingers **936** of metallization patterns **932** and **934** are interlaced as shown in FIG. **9**C. The finger tracks **936** increase the capacitance. The metallization patterns **932** and **934** can be formed using any conducting metal. In an embodiment, copper or aluminum is used to form the metallization patterns **932** and **934**, but the invention is not limited to these metals.

Presently, in some processes, the minimum wire width of higher metal layers may not be the same as for lower metal layers, which may complicate forming a spiral of the same dimensions in each metal layer. In such cases, metal layer **930** can be used for metal layers having a relatively large minimum width. In general, scalable high density capacitors according to the present invention can be formed using combinations of the structures described herein. For example, the structure of capacitor **920** resembles a combination of the structures of capacitors **150** and **820**.

FIG. **9**D is a schematic diagram depicting a top view of an example via layer **940** of the second scalable integrated circuit spiral-structure high density capacitor **920**. Via layer **940** includes a plurality of vias **926** used to couple adjacent metal layers of capacitor **920**. The vias **926** couple both the metallization patterns **932** and **934** and the border tracks **908**.

FIG. **10**A is a schematic diagram depicting a top view of a first example metal layer **1000** of a third scalable integrated circuit spiral-structure high density capacitor **1020** (see FIG. **10**B) according to an embodiment of the present invention. Metal layer **1000** includes a first metallization pattern **1002** and a second metallization pattern **1004**. The metallization patterns **1002** and **1004** are interleaved with one another in a spiral arrangement as depicted in FIG. **10**A. In region **1006** of capacitor **1020**, a end of metallization pattern **1002** optionally wraps around an end of metallization pattern **1004**, as shown in FIG. **10**A.

FIG. **10**B is a schematic diagram depicting a cross sectional view (A-A) (see FIG. **10**A) of the third scalable integrated circuit spiral-structure high density capacitor **1020**. As shown in FIG. **10**B, capacitor **1020** is scalable in that it can be formed using a selected number of metal layers such as, for example, metal layer **1000**, and the area of these metal layers can be varied. Optional shields **1022** and/or **1024** can be included as the top-most and/or bottom-most layers of capacitor **1020**, as well as optional side shields, to help reduce unwanted parasitic capacitance.

FIG. **10**C is a schematic diagram depicting a top view of a second example metal layer **1030** for the third scalable integrated circuit spiral-structure high density capacitor **1020**. Metal layer **1030** includes a first metallization pattern **1032** and a second metallization pattern **1034**. The metallization patterns **1032** and **1034** are interleaved with one another in a spiral arrangement as depicted in FIG. **10**C. In region **1036** of capacitor **1020**, note that the end of metallization pattern **1032** does not wrap around the end of metallization pattern **1034**, as is shown in region **1006** (see FIG. **10**A). Note also the similarities and differences present in the center of the spiral of metal layers **1000** and **1030**. These differences between the metal layers enable, for example, every other metal layer (e.g., metal layers B, D, and F) to be connected to a first terminal (e.g., A) of capacitor **1020**, and the remaining metal layers (e.g., C and E) to be connected to a second terminal (e.g., B) of capacitor **1020**. Structures other than that shown, however, can be used to achieve this interconnection.

FIG. **10**D is a schematic diagram depicting a top view of a first portion **1040** of an example via layer of the third scalable integrated circuit spiral-structure high density capacitor **1020**. The example via layer is used to couple metal layer **1000** to metal layer **1030**. As shown in FIG. **10**D, a first via **1042** couples an end of metallization pattern **1002** to an end of metallization pattern **1032**. A second via **1042** couples an end of metallization pattern **1004** to an end of metallization pattern **1034**.

FIG. **10**E is a schematic diagram depicting a top view of a second portion **1050** of the example via layer used to couple metal layer **1000** to metal layer **1030**. As shown in FIG. **10**E, a via **1042** couples a second end of metallization pattern **1002** to a second end of metallization pattern **1032**. A via **1042** also couples a second end of metallization pattern **1004** to a second end of metallization pattern **1034**. In embodiments, vias in addition to these four vias are used to couple a metal layer **1000** to a metal layer **1030**.

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. It will be apparent to persons skilled in the relevant art that various changes in form and detail can be made therein without departing from the spirit and scope of the invention. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

What is claimed is:

1. A scalable integrated circuit capacitor, comprising:

a first metal layer and a second metal layer each including a first metallization pattern and a plurality of second metallization patterns, said first metallization pattern having a plurality of loop openings wherein each loop opening encompasses one of said plurality of second metallization patterns; and

a first via layer including a plurality of vias that couple said first metallization pattern of said first metal layer to said

plurality of second metallization patterns of said second metal layer and said plurality of second metallization patterns of said first metal layer to said first metallization pattern of said second metal layer.

2. The capacitor of claim 1, further comprising:

a shield proximate to said first metal layer that reduces unwanted parasitic capacitance.

3. The capacitor of claim 2, wherein said shield is coupled to said first metallization pattern of said first metal layer.

4. A scalable integrated circuit capacitor, comprising:

a first metal layer and a second metal layer each including a first metallization pattern and a second metallization pattern, said first metallization pattern and said second metallization pattern each having a backbone track and a plurality of finger tracks approximately perpendicular to a backbone track, each of said plurality of finger tracks having a plurality of spur tracks, wherein said plurality of finger tracks of said first metallization pattern are interlaced with said plurality of finger tracks of said second metallization pattern and said plurality of spur tracks of said first metallization pattern are interlaced with said plurality of spur tracks of said second metallization pattern; and

a first via layer including a plurality of vias that couple said first metallization pattern of said first metal layer to said second metallization pattern of said second metal layer and said second metallization pattern of said first metal layer to said first metallization pattern of said second metal layer.

5. The capacitor of claim 4, wherein said plurality of vias couple said plurality of spur tracks of said first metallization pattern of said first metal layer to said plurality of spur tracks of said first metallization pattern of said second metal layer and said plurality of spur tracks of said second metallization pattern of said first metal layer to said plurality of spur tracks of said second metallization pattern of said second metal layer.

6. The capacitor of claim 4, further comprising:

a shield proximate to said first metal layer that reduces unwanted parasitic capacitance.

7. The capacitor of claim 6, wherein said shield is coupled to said first metallization pattern of said first metal layer.

8. A scalable integrated circuit capacitor, comprising:

a first metal layer and a second metal layer each including a plurality of first metallization patterns and a plurality of second metallization patterns, wherein said plurality of first metallization patterns and said plurality of second metallization patterns are interspersed with one another ma checkerboard arrangement in said first metal layer; and

a first via layer including a plurality of vias that couple said plurality of first metallization patterns of said first metal layer to said plurality of first metallization patterns of said second metal layer and said plurality of second metallization patterns of said first metal layer to said plurality of second metallization patterns of said second metal layer.

9. The capacitor of claim 8, wherein a majority of each of said plurality of first metallization patterns of each metal layer are coupled to two other first metallization patterns in the same metal layer by connecting tracks, and a majority of each of said plurality of second metallization patterns of each metal layer are coupled to two other second metallization patterns in the same metal layer by connecting tracks.

10. The capacitor of claim 9, wherein each metal layer further comprises:

a first backbone track that couples together each of said plurality of first metallization patterns; and

a second backbone track that couples together each of said plurality of second metallization patterns.

11. The capacitor of claim 8, further comprising:

a shield proximate to said first metal layer that reduces unwanted parasitic capacitance.

12. The capacitor of claim 11, wherein said shield is coupled to each of said plurality of first metallization patterns of said first metal layer.

13. A scalable integrated circuit capacitor, comprising:

a first metal layer and a second metal layer each including a first metallization pattern and a second metallization pattern, wherein said first metallization pattern and said second metallization pattern are interleaved with one another in a spiral arrangement of more than 180 degrees; and

a first via layer including a plurality of vias that couple said first metallization pattern of said first metal layer to said first metallization pattern of said second metal layer and said second metallization pattern of said first metal layer to said second metallization pattern of said second metal layer.

14. The capacitor of claim 13, further comprising:

a shield proximate to said first metal layer that reduces unwanted parasitic capacitance.

15. The capacitor of claim 14, wherein said shield is coupled to said first metallization pattern of said first metal layer.

16. A scalable integrated circuit capacitor, comprising:

a first metal layer and a second metal layer each including a first metallization pattern and a second metallization pattern, wherein said first metallization pattern and said second metallization pattern are interleaved with one another in a spiral arrangement of more than 180 degrees; and

a first via layer including a plurality of vias that couple a first end of said first metallization pattern of said first metal layer to a first end of said second metallization pattern of said second metal layer and a first end of said second metallization pattern of said first metal layer to a first end of said first metallization pattern of said second metal layer.

17. The capacitor of claim 16, wherein said first via layer further comprises:

vias that couple a second end of said first metallization pattern of said first metal layer to a second end of said second metallization pattern of said second metal layer and a second end of said second metallization pattern of said first metal layer to a second end of said first metallization pattern of said second metal layer.

18. The capacitor of claim 17, further comprising:

a shield proximate to said first metal layer that reduces unwanted parasitic capacitance.

19. The capacitor of claim 18, wherein said shield is coupled to said first metallization pattern of said first metal layer.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,656,643 B2
APPLICATION NO. : 11/878696
DATED : February 2, 2010
INVENTOR(S) : Victor Chiu-Kit Fong et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

At column 11, line 49, replace "another ma checkerboard" with -- another in a checkerboard --.

Signed and Sealed this

Thirty-first Day of August, 2010

David J. Kappos

David J. Kappos
Director of the United States Patent and Trademark Office

Exhibit D



(12) **United States Patent**
Qi et al.

(10) **Patent No.: US 8,315,381 B2**
(45) **Date of Patent: *Nov. 20, 2012**

(54) **METHODS AND APPARATUS PERFORMING HASH OPERATIONS IN A CRYPTOGRAPHY ACCELERATOR**

(75) Inventors: **Zheng Qi**, Milpitas, CA (US); **Ronald Squires**, Castro Valley, CA (US); **Mark Buer**, Gilbert, AZ (US); **David K. Chin**, Mountain View, CA (US)

(73) Assignee: **Broadcom Corporation**, Irvine, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 425 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/216,243**

(22) Filed: **Jul. 1, 2008**

(65) **Prior Publication Data**

US 2009/0028326 A1 Jan. 29, 2009

Related U.S. Application Data

(63) Continuation of application No. 10/330,694, filed on Dec. 24, 2002, now Pat. No. 7,400,722.

(60) Provisional application No. 60/368,583, filed on Mar. 28, 2002.

(51) **Int. Cl.**
H04L 9/00 (2006.01)

(52) **U.S. Cl.** **380/28**; 380/29; 380/44; 713/156; 713/166; 713/187

(58) **Field of Classification Search** 713/168, 713/189, 156, 166, 187; 380/28–29, 44
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,265,164	A	11/1993	Matyas et al.	
5,841,870	A *	11/1998	Fieres et al.	713/156
5,949,881	A	9/1999	Davis	
6,249,229	B1 *	6/2001	Eckstein et al.	340/572.4
6,694,436	B1	2/2004	Audebert	
6,829,220	B1 *	12/2004	Thorup et al.	370/235
6,983,366	B1	1/2006	Huynh et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

WO	WO 01/61912 A1	8/2001

OTHER PUBLICATIONS

Deepakumara et al., FPGA Implementation of MD5 Hash Algorithm, Electrical and Computer Engineering, 2001. Canadian Conference on vol. 2, May 13-16, 2001, pp. 919-924.

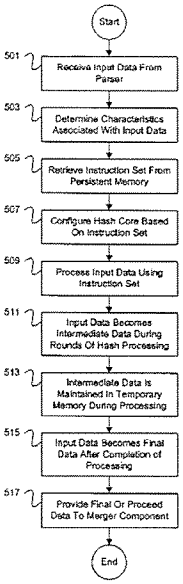
(Continued)

Primary Examiner — Thanhnga B Truong
(74) *Attorney, Agent, or Firm* — Sterne, Kessler, Goldstein & Fox P.L.L.C.

(57) **ABSTRACT**

Methods and apparatus are provided for implementing a cryptography accelerator for performing operations such as hash operations. The cryptography accelerator recognizes characteristics associated with input data and retrieves an instruction set for processing the input data. The instruction set is used to configure or control components such as MD5 and SHA-1 hash cores, XOR components, memory, etc. By providing a cryptography accelerator with access to multiple instruction sets, a variety of hash operations can be performed in a configurable cryptographic accelerator.

28 Claims, 10 Drawing Sheets



U.S. PATENT DOCUMENTS

7,024,695	B1	4/2006	Kumar et al.	
7,400,722	B2	7/2008	Qi et al.	
2002/0097724	A1	7/2002	Halme et al.	
2002/0191791	A1	12/2002	Anand	
2003/0058274	A1*	3/2003	Hill et al.	345/751
2003/0084309	A1	5/2003	Kohn	
2003/0172257	A1*	9/2003	Greenblat et al.	712/234

OTHER PUBLICATIONS

Jiang et al., Securing Web Servers Against Insider Attack, Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th annual, Dec. 10-14, 2001, pp. 265-276.

Thomas A. Stephen, SSL and TLS Essentials Securing the Web, 2000, pp. 4-6, 41-45, and 96-104.

* cited by examiner

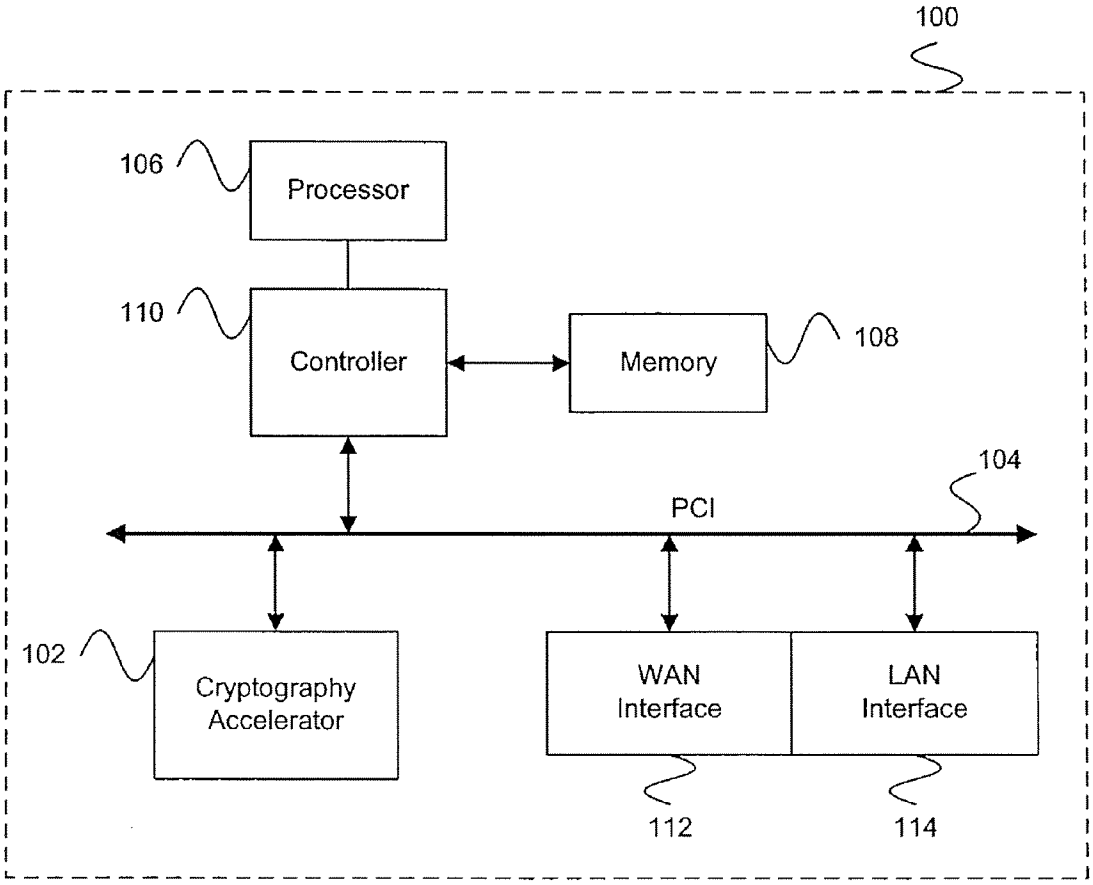

100
106
Processor
110
Controller
Memory
108
104
PCI
102
Cryptography Accelerator
WAN Interface
LAN Interface
112
114

FIG. 1

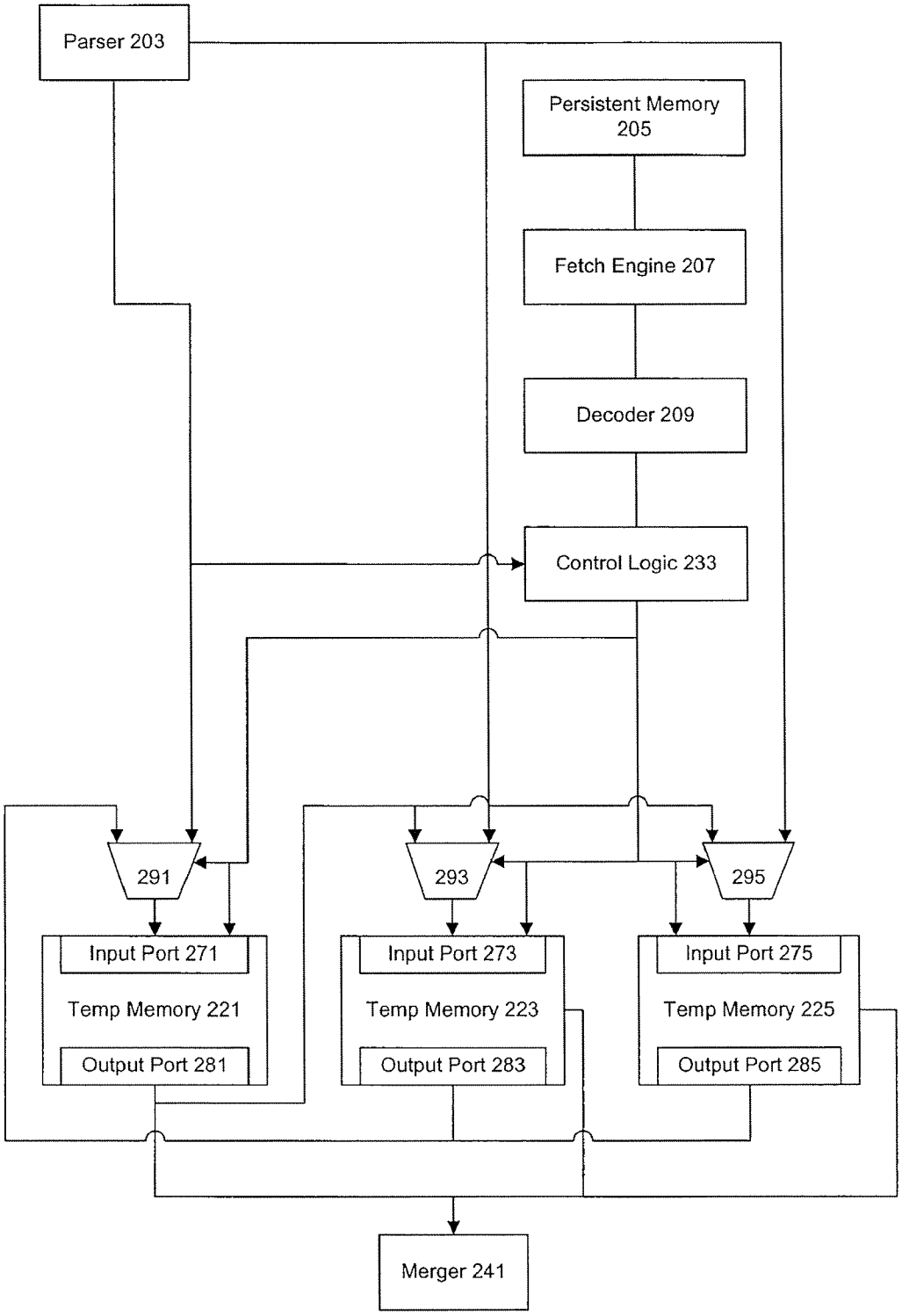

Parser 203
Persistent Memory 205
Fetch Engine 207
Decoder 209
Control Logic 233
291
293
295
Input Port 271
Temp Memory 221
Output Port 281
Input Port 273
Temp Memory 223
Output Port 283
Input Port 275
Temp Memory 225
Output Port 285
Merger 241

FIG. 2A

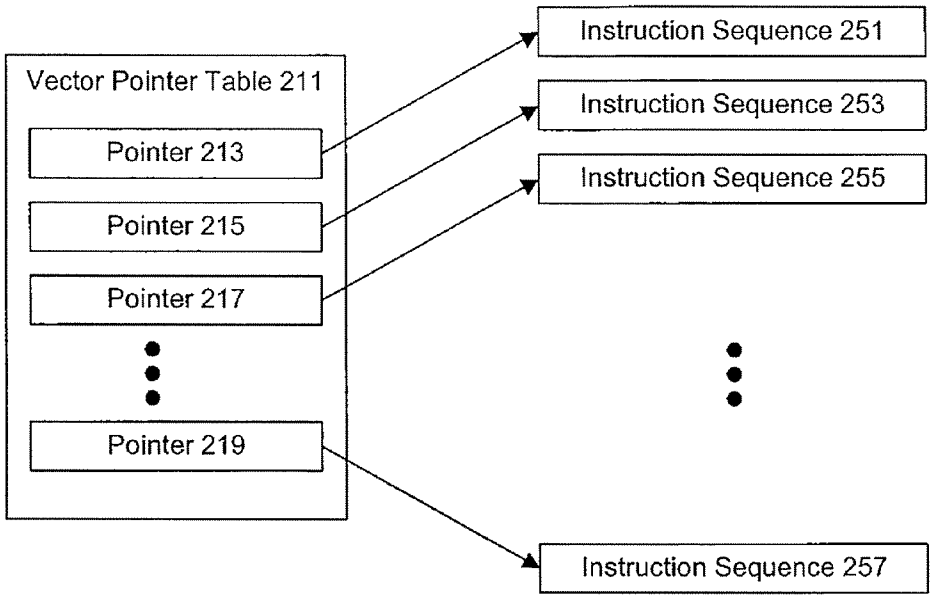

Vector Pointer Table 211
Pointer 213
Pointer 215
Pointer 217
Pointer 219
Instruction Sequence 251
Instruction Sequence 253
Instruction Sequence 255
Instruction Sequence 257

FIG. 2B

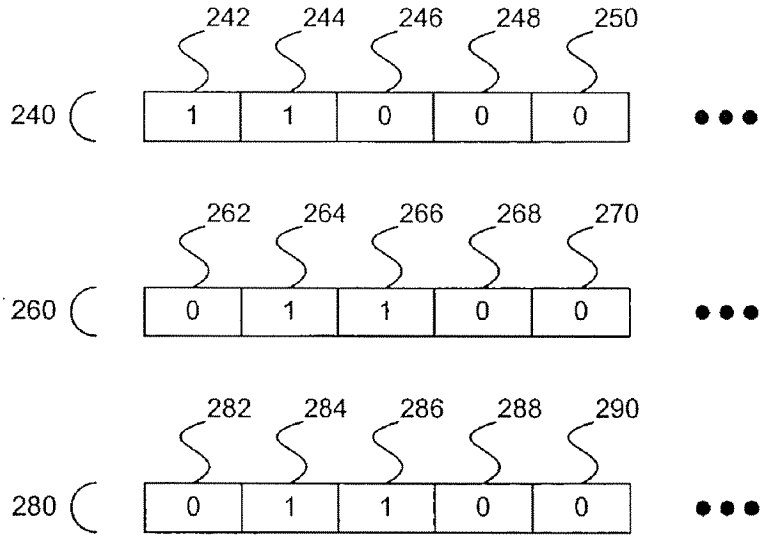

242 244 246 248 250
240
1 1 0 0 0
262 264 266 268 270
260
0 1 1 0 0
282 284 286 288 290
280
0 1 1 0 0

FIG. 2C

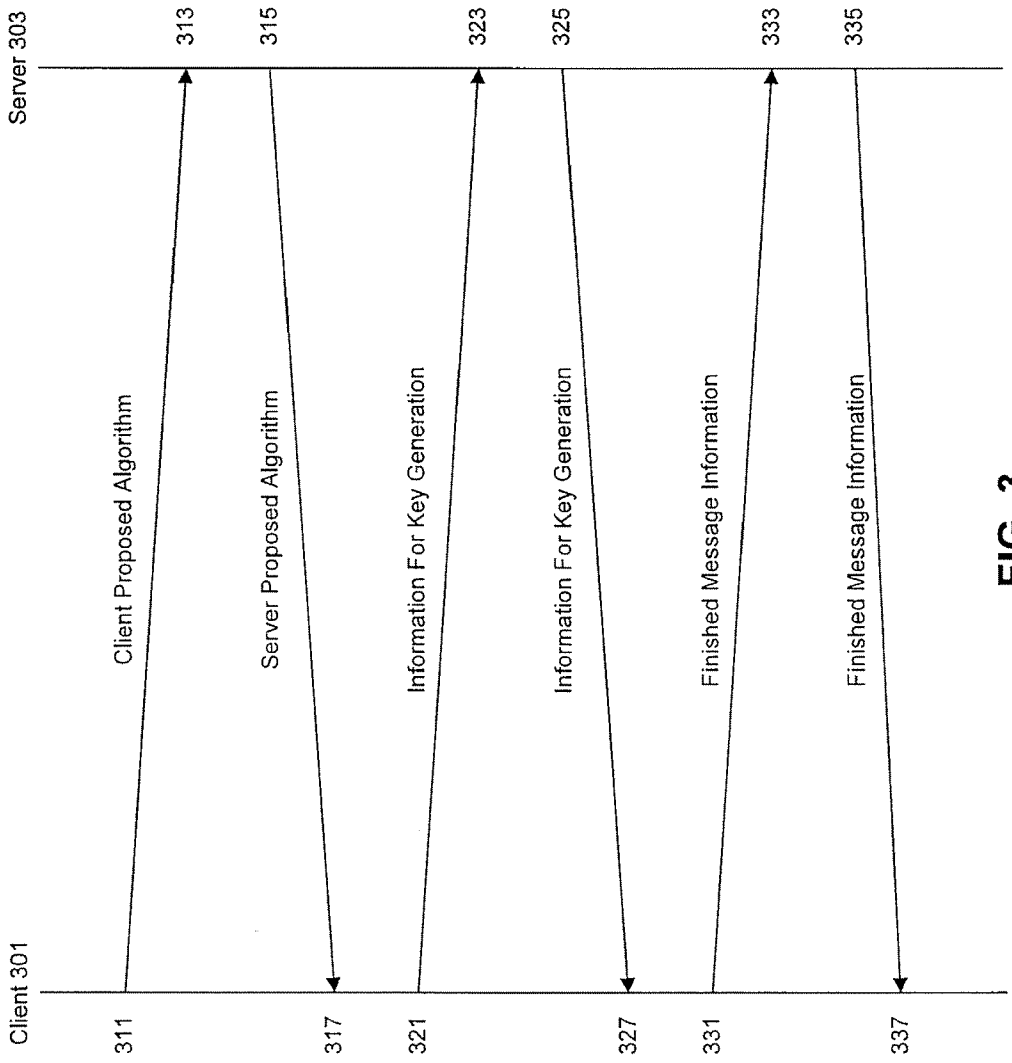

Server 303
Client 301
313
315
323
325
333
335
311
317
321
327
331
337
Client Proposed Algorithm
Server Proposed Algorithm
Information For Key Generation
Information For Key Generation
Finished Message Information
Finished Message Information

FIG. 3

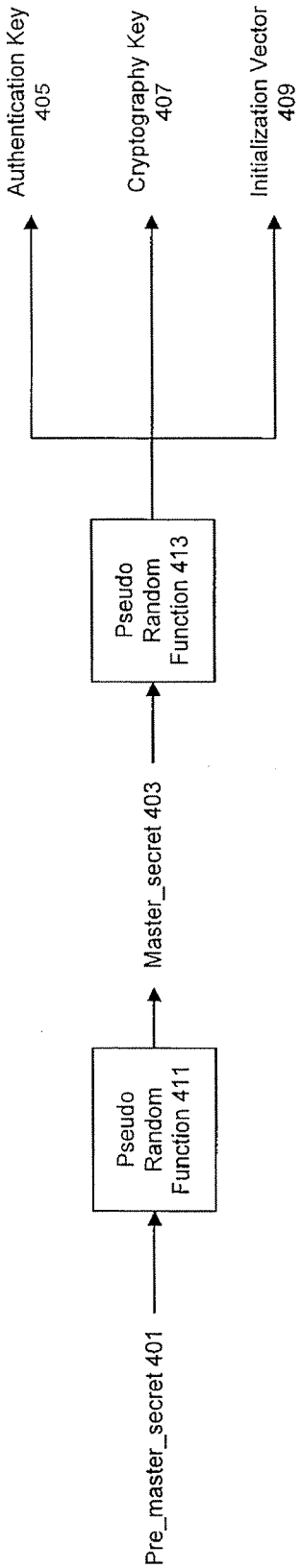

Pre_master_secret 401
Pseudo Random Function 411
Master_secret 403
Pseudo Random Function 413
Authentication Key 405
Cryptography Key 407
Initialization Vector 409

FIG. 4

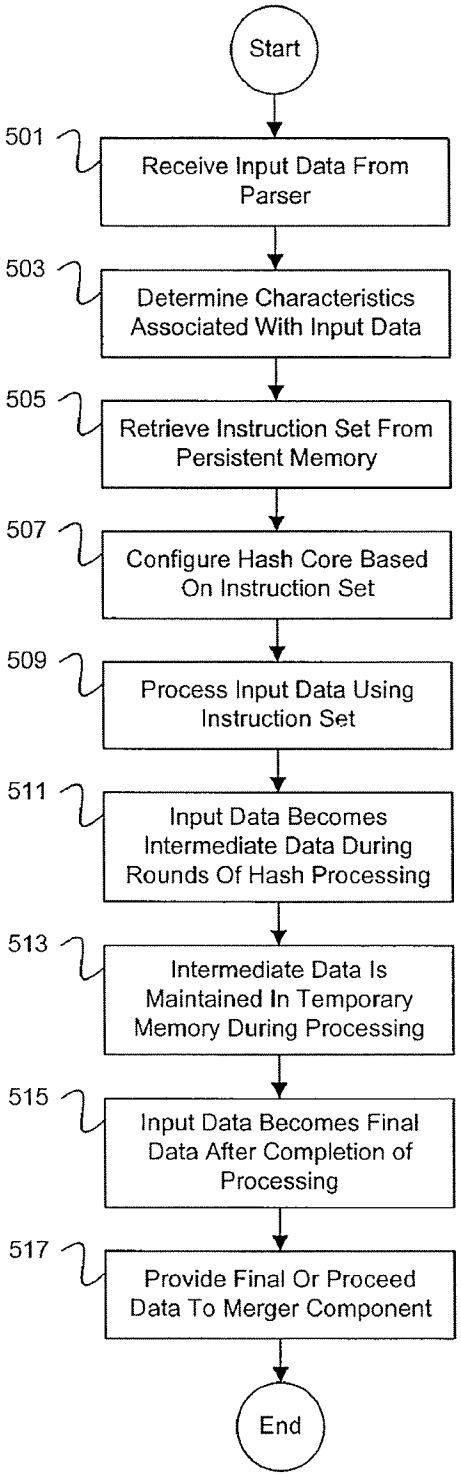

Start
501
Receive Input Data From Parser
503
Determine Characteristics Associated With Input Data
505
Retrieve Instruction Set From Persistent Memory
507
Configure Hash Core Based On Instruction Set
509
Process Input Data Using Instruction Set
511
Input Data Becomes Intermediate Data During Rounds Of Hash Processing
513
Intermediate Data Is Maintained In Temporary Memory During Processing
515
Input Data Becomes Final Data After Completion of Processing
517
Provide Final Or Proceed Data To Merger Component
End

FIG. 5

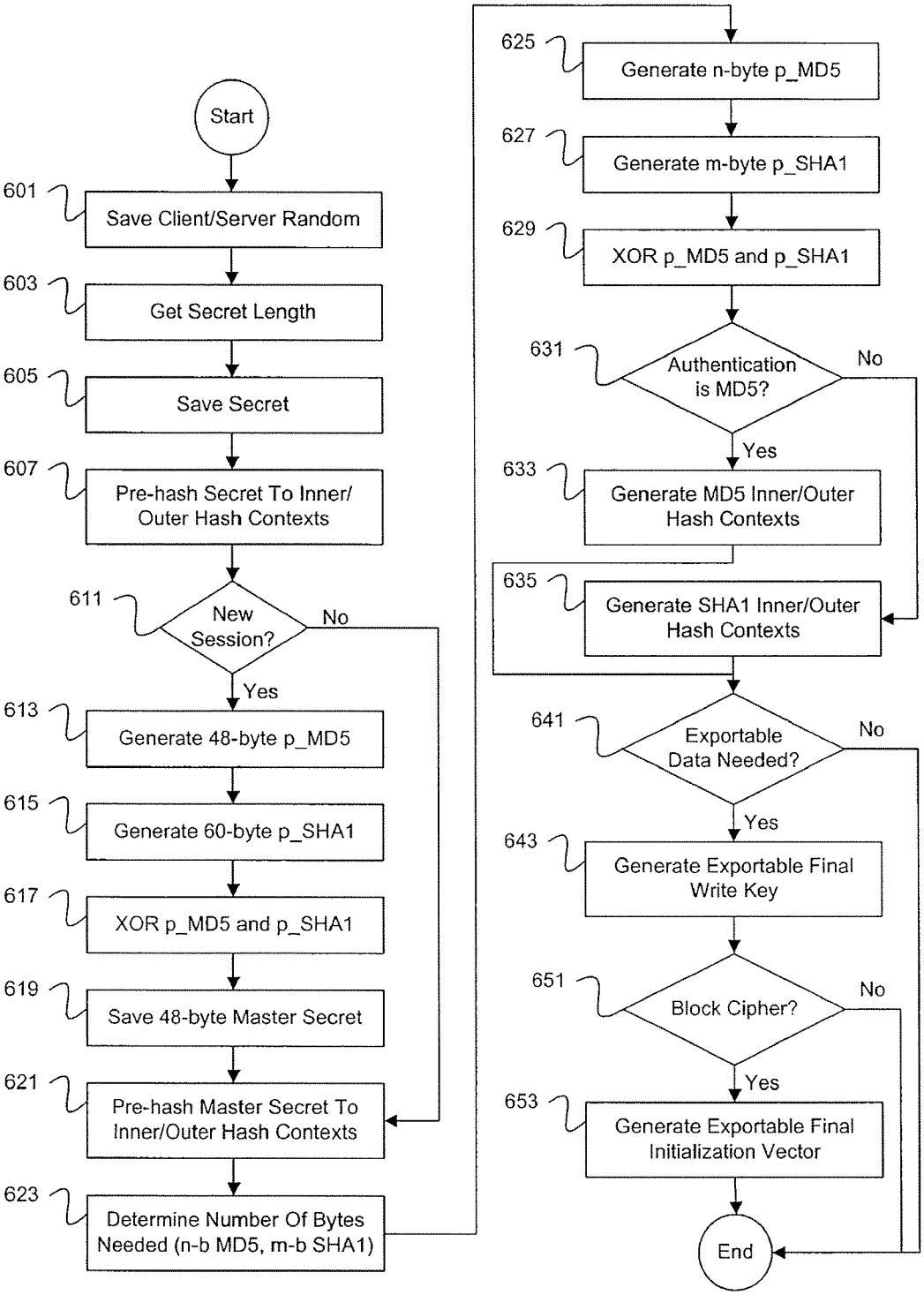

Start
601
Save Client/Server Random
603
Get Secret Length
605
Save Secret
607
Pre-hash Secret To Inner/ Outer Hash Contexts
611
New Session?
No
Yes
613
Generate 48-byte p_MD5
615
Generate 60-byte p_SHA1
617
XOR p_MD5 and p_SHA1
619
Save 48-byte Master Secret
621
Pre-hash Master Secret To Inner/Outer Hash Contexts
623
Determine Number Of Bytes Needed (n-b MD5, m-b SHA1)
625
Generate n-byte p_MD5
627
Generate m-byte p_SHA1
629
XOR p_MD5 and p_SHA1
631
Authentication is MD5?
No
Yes
633
Generate MD5 Inner/Outer Hash Contexts
635
Generate SHA1 Inner/Outer Hash Contexts
641
Exportable Data Needed?
No
Yes
643
Generate Exportable Final Write Key
651
Block Cipher?
No
Yes
653
Generate Exportable Final Initialization Vector
End

FIG. 6

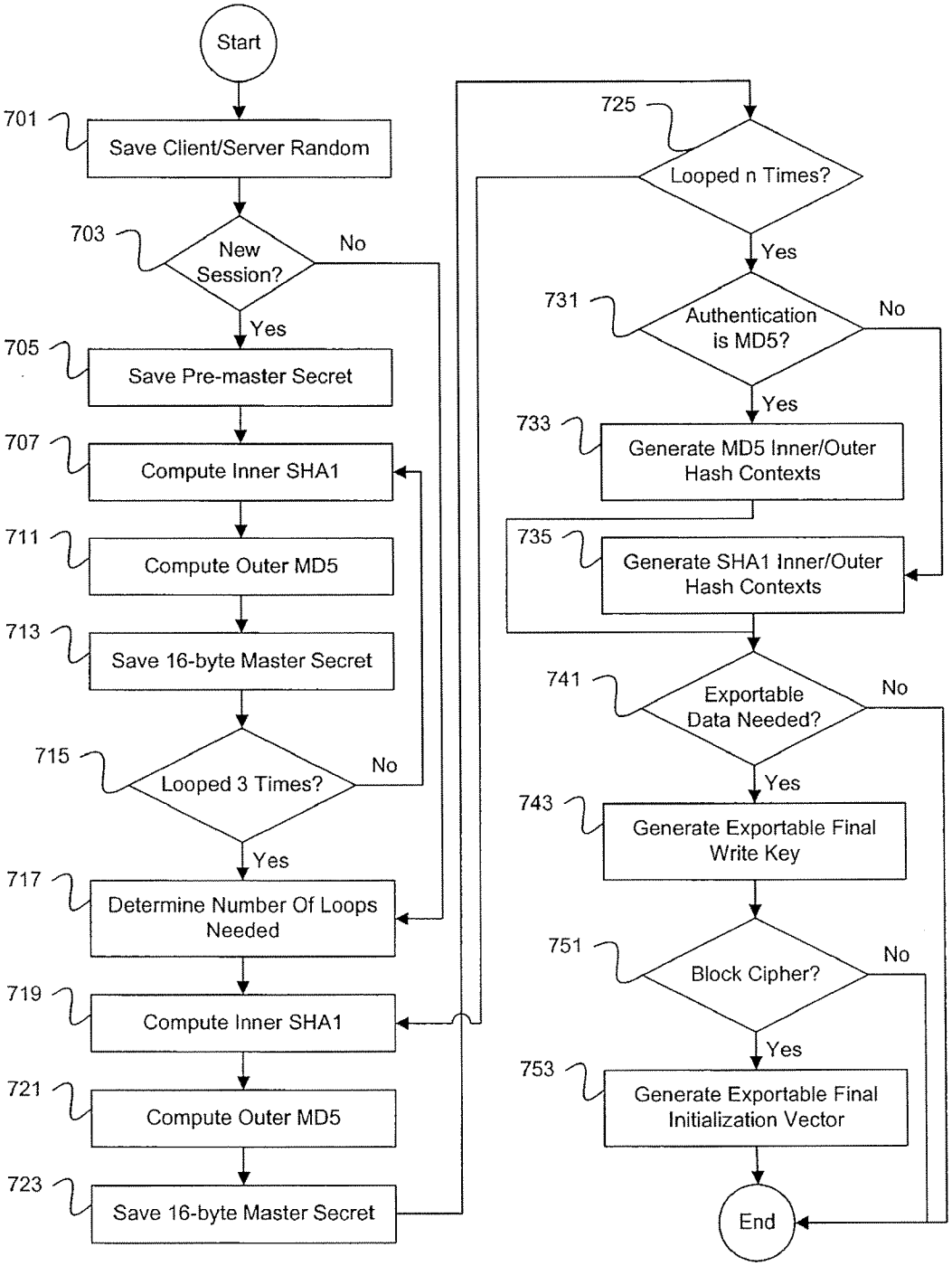

Start
701
Save Client/Server Random
703
New Session?
No
Yes
705
Save Pre-master Secret
707
Compute Inner SHA1
711
Compute Outer MD5
713
Save 16-byte Master Secret
715
Looped 3 Times?
No
Yes
717
Determine Number Of Loops Needed
719
Compute Inner SHA1
721
Compute Outer MD5
723
Save 16-byte Master Secret
725
Looped n Times?
Yes
731
Authentication is MD5?
No
Yes
733
Generate MD5 Inner/Outer Hash Contexts
735
Generate SHA1 Inner/Outer Hash Contexts
741
Exportable Data Needed?
No
Yes
743
Generate Exportable Final Write Key
751
Block Cipher?
No
Yes
753
Generate Exportable Final Initialization Vector
End

FIG. 7

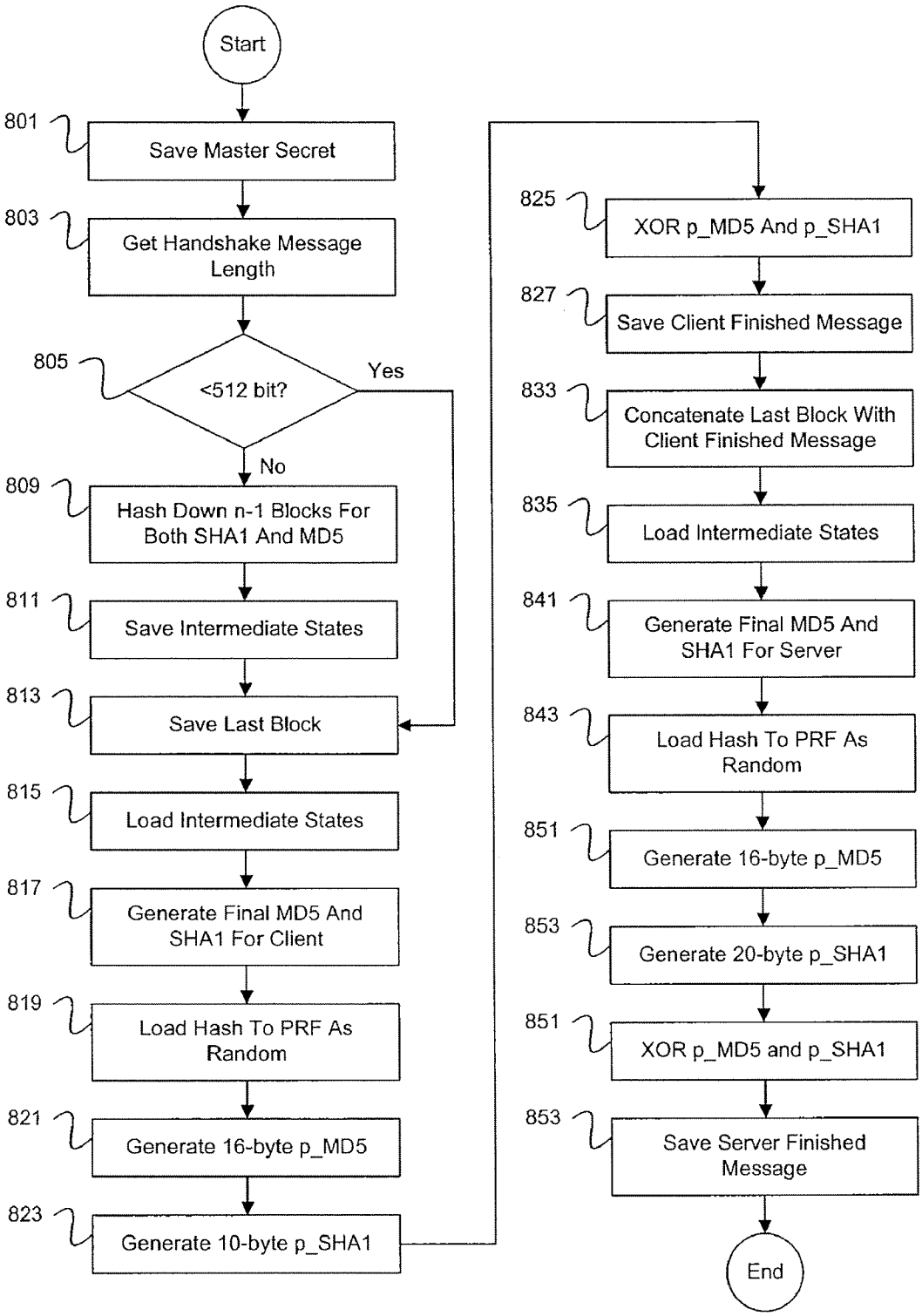

Start
801
Save Master Secret
803
Get Handshake Message Length
805
<512 bit?
Yes
No
809
Hash Down n-1 Blocks For Both SHA1 And MD5
811
Save Intermediate States
813
Save Last Block
815
Load Intermediate States
817
Generate Final MD5 And SHA1 For Client
819
Load Hash To PRF As Random
821
Generate 16-byte p_MD5
823
Generate 10-byte p_SHA1
825
XOR p_MD5 And p_SHA1
827
Save Client Finished Message
833
Concatenate Last Block With Client Finished Message
835
Load Intermediate States
841
Generate Final MD5 And SHA1 For Server
843
Load Hash To PRF As Random
851
Generate 16-byte p_MD5
853
Generate 20-byte p_SHA1
851
XOR p_MD5 and p_SHA1
853
Save Server Finished Message
End

FIG. 8

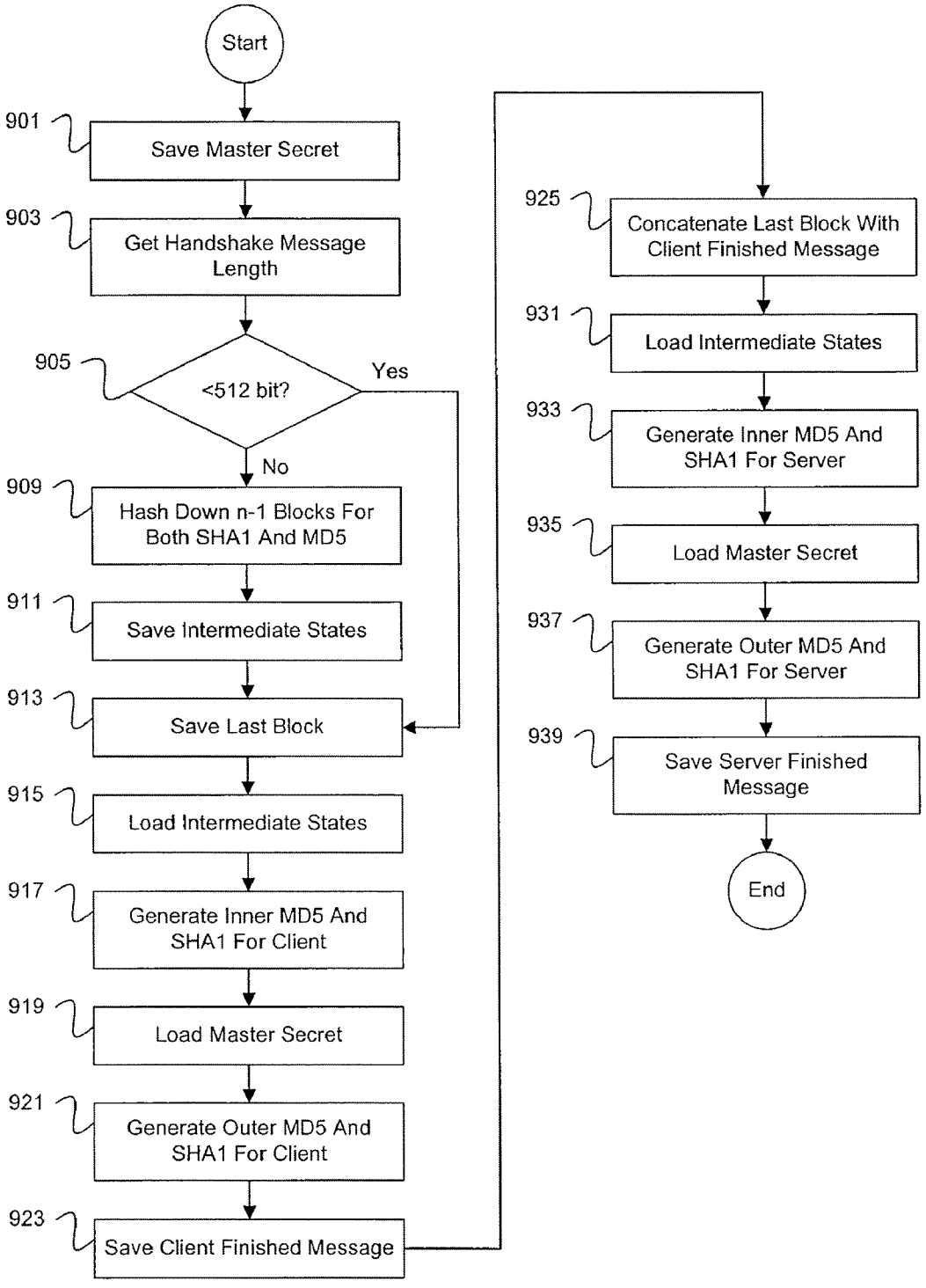

Start
901
Save Master Secret
903
Get Handshake Message Length
905
<512 bit?
Yes
No
909
Hash Down n-1 Blocks For Both SHA1 And MD5
911
Save Intermediate States
913
Save Last Block
915
Load Intermediate States
917
Generate Inner MD5 And SHA1 For Client
919
Load Master Secret
921
Generate Outer MD5 And SHA1 For Client
923
Save Client Finished Message
925
Concatenate Last Block With Client Finished Message
931
Load Intermediate States
933
Generate Inner MD5 And SHA1 For Server
935
Load Master Secret
937
Generate Outer MD5 And SHA1 For Server
939
Save Server Finished Message
End

FIG. 9

METHODS AND APPARATUS PERFORMING HASH OPERATIONS IN A CRYPTOGRAPHY ACCELERATOR

CROSS REFERENCE TO RELATED APPLICATIONS

This application is a continuation of prior application Ser. No. 10/330,694, filed on Dec. 24, 2002, which claims priority under U.S.C. 119(e) from U.S. Provisional Application No. 60/368,583 entitled "Methods and Apparatus for Implementing a Configurable Authentication Accelerator," filed on Mar. 28, 2002, both of which are herein incorporated by reference for all purposes.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present application relates to implementing a cryptography accelerator. More specifically, the present application relates to methods and apparatus for providing a configurable cryptography accelerator with instruction sets for performing hash operations on input data.

2. Description of Related Art

Conventional software and hardware designs for performing hash operations are inefficient. One technique for securing a communication channel between two network entities such as a client and a server specifies that the two entities perform a cryptography handshake sequence. During the cryptographic handshake sequence, the two network entities will typically perform various cryptographic operations such as encryption and authentication operations to verify the identity of the other and to exchange information to establish a secure channel.

In one example, session keys are exchanged after the identity of the other network entity is verified. However, both software, firmware and hardware techniques for performing hash operations, such as hash operations used in cryptography handshake sequences, have been inefficient and resource intensive. Cryptography handshake sequences and hash algorithms are described in Applied Cryptography, Bruce Schneier, John Wiley & Sons, Inc. (ISBN 0471128457), incorporated by reference in its entirety for all purposes.

It is therefore desirable to provide methods and apparatus for improving hash operations with respect to some or all of the performance limitations noted above.

SUMMARY OF THE INVENTION

Methods and apparatus are provided for implementing a cryptography accelerator for performing operations such as hash operations. The cryptography accelerator recognizes characteristics associated with input data and retrieves an instruction set for processing the input data. The instruction set is used to configure or control components such as MD5 and SHA-1 hash cores, XOR components, memory, etc. By providing a cryptography accelerator with access to multiple instruction sets, a variety of hash operations can be performed in a configurable cryptographic accelerator.

According to various embodiments, a cryptography accelerator for performing hash operations is provided. The accelerator includes a first hash core, a persistent memory, and a temporary memory. The first hash core is operable to perform a plurality of rounds of hash computations on input data to derive processed data. A persistent memory contains a plurality of instruction sets. The plurality of instruction sets provide information for the first hash core on operations to perform on input data and intermediate data during the plurality of rounds of hash computations. The temporary memory is coupled to the first hash core. The temporary memory is operable to hold input data and intermediate data.

According to other embodiments, a method for performing hash operations is provided. Input data is received. Characteristics associated with the input data are determined. An instruction set for performing hash operations on input data is selected. The instruction set is selected from a plurality of instruction sets maintained in persistent memory associated with a first hash core. The first hash core is configured using the instruction set. The first hash core is operable to perform hash operations on input data based on the instruction set.

These and other features and advantages of the present invention will be presented in more detail in the following specification of the invention and the accompanying figures, which illustrate by way of example the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may best be understood by reference to the following description taken in conjunction with the accompanying drawings, which are illustrative of specific embodiments of the present invention.

FIG. **1** is a diagrammatic representation of a system that can use the techniques of the present invention.

FIG. **2**A is a diagrammatic representation of an integrated circuit containing a processing core for performing hash operations.

FIG. **2**B is a diagrammatic representation showing a structure referencing instruction sequences.

FIG. **2**C is a diagrammatic representation of mechanisms for determining resource dependencies.

FIG. **3** is an interaction diagram showing a sequence in which the techniques of the present invention can be applied.

FIG. **4** is a diagrammatic representation showing input data, intermediate data, and processed data.

FIG. **5** is a flow process diagram showing techniques for selecting an instruction set.

FIG. **6** is a flow process diagram showing TLS 1.0 key derivation.

FIG. **7** is a flow process diagram showing SSLv3 key derivation.

FIG. **8** is a flow process diagram showing TLS 1.0 finished message generation.

FIG. **9** is a flow process diagram showing SSLv3 finished message generation.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

The present application relates to implementing a cryptography accelerator. More specifically, the present application relates to methods and apparatus for providing a cryptography accelerator capable of performing a variety of different hash operations on input data.

Reference will now be made in detail to some specific embodiments of the invention including the best modes contemplated by the inventors for carrying out the invention. Examples of these specific embodiments are illustrated in the accompanying drawings. While the invention is described in conjunction with these specific embodiments, it will be understood that it is not intended to limit the invention to the described embodiments. On the contrary, it is intended to cover alternatives, modifications, and equivalents as may be

included within the spirit and scope of the invention as defined by the appended claims.

For example, the techniques of the present invention will be described in the context of the SHA-1 and MD5 hash algorithms. However, it should be noted that the techniques of the present invention can be applied to a variety of different hash operations for cryptography processing in general. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. The present invention may be practiced without some or all of these specific details. In other instances, well known process operations have not been described in detail in order not to unnecessarily obscure the present invention.

A wide variety of algorithms are used for encryption and authentication operations. In many conventional implementations, software is used to identify the type of data and the cryptographic processing needed for the particular data sequence. However, cryptographic operations implemented entirely in software on a generic processor such as a reduced instruction set (RISC) or complex instruction set (CISC) processors are highly inefficient. In many environments, it is beneficial to use specialized accelerators for performing cryptographic operations, such as DES and SHA-1 operations. In typical cryptography accelerator implementations, a cryptography accelerator is configured to perform resource intensive cryptographic operations while software through an external host is configured to perform sequencing. That is, software formats and sequences data and makes function calls to elementary cryptographic operators. In one example, a cryptography accelerator would be responsible for executing a function such as cryptooperation (data, key1, key2) while the software would be responsible for formatting the data properly, acquiring the keys, and making multiple calls to the function when necessary.

More recent efforts have focused on implementing both core processing as well as formatting and sequencing on a cryptography accelerator. In one example, software running on a host such as a CPU external to a cryptography accelerator could simply forward a packet to the cryptography accelerator. Using the packet, the cryptography accelerator would extract information to determine what type of processing and how many rounds of processing need to be performed.

One technique for implementing such a cryptography accelerator that performs both cryptography processing and sequencing uses state tables. Each load or store instruction on the cryptography accelerator is represented by one or more states. However, because many variations in cryptographic algorithms exist, a large number of states exist. Having a significant number of states makes implementation and verification extremely difficult. Furthermore, if a new cryptographic algorithm is developed, substantial work would have to be performed in order to update the states associated with the instructions.

Consequently, the techniques of the present invention provide sequences of instructions for performing cryptographic as well as sequencing operations on data. Instruction sequences can relatively easily be implemented for particular cryptographic operations. When a new algorithm is developed, an additional instruction sequence can be provided on the cryptography accelerator. The variations between cryptographic algorithms can be handled with relative ease. The techniques and mechanisms of the present invention allow for a cryptographic accelerator that has the speed and processing advantages of a customized piece of hardware while retaining the flexibility of a piece of software.

FIG. **1** is a diagrammatic representation of one example of a processing system **100** with a cryptography accelerator according to various embodiments of the present invention. As shown in FIG. **1**, the present invention may be implemented in a stand-alone cryptography accelerator **102** or as part of the system **100**. In the described embodiment, the cryptography accelerator **102** is connected to a bus **104** such as a PCI bus via a standard on-chip PCI interface. The processing system **100** includes a processing unit **106** and a system memory unit **108**. The processing unit **106** and the system memory unit **108** are coupled to the system bus **104** via a bridge and memory controller **110**.

According to various embodiments, the processing unit **106** may be the central processing unit (CPU) of a system **100**. In one example, a LAN interface **114** is provided to couple the processing system **100** to a local area network (LAN) to allow packet receipt and transmission. Similarly, a Wide Area Network (WAN) interface **112** can also be provided to connect the processing system to a WAN (not shown) such as the Internet. The WAN interface manages in-bound and out-bound packets, providing automatic cryptographic processing for IP packets.

In many implementations, the cryptography accelerator **102** is an application specific integrated circuit (ASIC) coupled to the processor **106**. However, the cryptography accelerator **102** can also be a programmable logic device (PLD), field programmable gate array (FPGA), or other device coupled to the processor **106**. According to specific embodiments, the cryptography accelerator **102** is implemented either on a card connected to the bus **104** or as a standalone chip integrated in the system **100**.

In other embodiments, the cryptography accelerator **102** itself is integrated into the processing core of a CPU of system **100**, such as that available from Tensilica Corporation of Santa Clara, Calif. or ARC Cores of San Jose, Calif. In another embodiment, techniques and mechanisms of the present invention are integrated into a CPU such as a CPU available from Intel Corporation of San Jose, Calif. or AMD Corporation of Sunnyvale, Calif. By implementing cryptography accelerator functionality entirely on the processor **106**, a separate card or chip in the system **100** is not needed. In still other embodiments, the processing system **100** including the cryptography accelerator **102** is implemented as a system on a chip (SOC). The network interfaces, memory, processing core, and cryptography accelerator functionality are provided on a single integrated circuit device.

The cryptography accelerator **102** is capable of implementing various network security standards, such as Internet Protocol Security (IPSec), Secure Sockets Layer/Transport Layer Security (SSL/TLS), Internet Key Exchange (IKE) which provide application-transparent encryption and authentication services for network traffic.

Network security standards such as IPsec and SSL/TLS provide authentication through the use of hash algorithms. Two commonly used hash algorithms are MD5 and the Secure Hash algorithm (SHA-1). Other hash algorithms such as MD4 and MD2 are also available. Hash algorithms are described in Applied Cryptography, Bruce Schneier, John Wiley & Sons, Inc. (ISBN 0471128457), incorporated by reference in its entirety for all purposes. Even though many network security standards apply the same hash algorithms, different approaches are taken toward applying the hash algorithms to the actual authentication computation.

Different versions of the same network security standards even vary approaches toward applying the hash algorithms. In IPsec, several approaches such as HMAC-MD5-96 and HMAC-SHA1-96 based on the hash message authentication code (HMAC) algorithm are provided. The approaches HMAC-MD5-96 and HMAC-SHA1-96 are described in RFC

2403 and RFC 2404 respectively, while the HMAC algorithm is described in RFC 2104, the entireties of which are incorporated by reference for all purposes. SSL/TLS use similar, but slightly different approaches. In SSLv3, an earlier version of HMAC is used. In TLS 1.0, the same version of HMAC is used as in IPsec, but a different number of bits are taken for the full result.

The TLS 1.0 protocol is described in RFC 2246, the entirety of which is incorporated by reference for all purposes. SSL is described in E. Rescorla, SSL and TLS: Designing and Building Secure Systems (Addison-Wesley, 2001) and S. A. Thomas, SSL & TLS Essentials: Securing the Web (John Wiley & Sons, Inc. 2000), the entireties of which are incorporated by reference for all purposes. In addition, SSL/TLS define a set of functions using a combination of HMAC, MD5, and SHA1 to generate processed data. For example, combinations are used to generate a master secret sequence from a premaster secret sequence, to generate key blocks from a master secret sequence, or to perform hash operations for finished message processing and client certificate verification.

Typical cryptography accelerators use hash cores for performing hash operations. When a client or server participates in an authentication sequence such as a key exchange, clients and servers need cryptography accelerators specifically configured for particular versions of specified network security standards. In one example, if the server needs to perform TLS 1.0 operations, a cryptography accelerator such as an ASIC specifically microcoded with a TLS 1.0 instruction set would be required. A state machine can be used to perform operations associated with each network security standard version. As noted above, however, a state machine that can handle the number of standards in existence would be extremely complicated and difficult to implement.

Consequently, many cryptography accelerators typically contain only functionality for performing basic hash operations such as MD5 or SHA1 operations. Authentication specific functionality on a cryptography accelerator is often limited to MD5 or SHA1 hash cores. The external processor such as an external CPU would pass data to a cryptography accelerator when MD5 or SHA1 processing was needed. In one example, if a network security standard specified repeated calls to a MD5 or SHA1 function, the external processor would pass data to the cryptography accelerator during each function call, receive data output by the cryptography accelerator, and alter data as needed before passing the data back to the cryptography accelerator for another function call.

Typically, only a single hash function call would be performed on data before sending the data back to an external processor. In another example, if XOR operations were specified for data output from the MD5 and SHA1 cores, the external processor would perform the XOR operations even if the XOR operations were a specific part of the cryptographic processing. Because of inefficiencies such as the passing of data between the external processor and the cryptography accelerator between function calls, cryptographic processing for a server or client expecting many different versions of network security protocols has been limited.

The techniques of the present invention, however, provide not only for a cryptography accelerator specifically configured for a particular type of hash operations without the need to send and receive data to an external processor between various calls to a particular function implemented on a chip, the techniques of the present invention provide an automatically configurable cryptographic accelerator that recognizes characteristics of the input data and automatically performs cryptographic processing such as SSLv3 or TLS 1.0 key derivation.

It should be noted that recognizing characteristics of the input data can include operations such as analyzing the input data, retrieving information associated with the input data, or recognizing characteristics of instruction sequences associated with the input data. A single cryptographic accelerator, for example, with an MD5 and a SHA1 core can perform cryptographic processing associated with a variety of operations using the MD5 and SHA1 hash operations. In one example, the cryptography accelerator can perform cryptographic operations associated with IPsec and SSL/TLS processing.

FIG. **2A** is a diagrammatic representation of one example of a cryptography accelerator according to various embodiments. The cryptography accelerator includes an interface having a parser **203** coupled to an entity such as external processor for receiving and delineating input data sequences. In one example, the parser **203** receives a data sequence associated with SSLv3 key derivation. The control logic **233** determines that key derivation operations associated with SSLv3 should be performed on the data sequence. The control logic **233** retrieves an instruction set associated with SSLv3 key derivation from persistent memory **205**. Memory that retains data after hash operations are completed is referred to herein as persistent memory. Persistent memory also typically remains intact when power is disconnected. In one embodiment, persistent memory is a read-only memory (ROM) on a cryptography accelerator chip, although persistent memory can also be components such as flash memory. In another embodiment, persistent memory **205** and temporary memory **221** are contained in the same component. A component such as a random access memory (RAM) can be loaded with instruction sets and can provide the capability to function as both a persistent memory and as a temporary memory, although such access may be slow.

According to various embodiments, persistent memory **205** includes a table with various types of operations and security protocols identified in the entries. The entries correspond to instruction sets for configuring the cryptography accelerator. Logic and mechanisms for configuring a cryptography accelerator for performing a particular type of cryptographic operation such as key derivation or finished message processing is referred to herein as an instruction set. The fetch engine **207** retrieves the instruction set from persistent memory **205**. According to various embodiments, the decoder **209** receives and interprets the instruction set for control logic **233**. In one embodiment, control logic **233** retrieves microcode for performing cryptographic operations on an input data sequence. Logic and mechanisms for configuring or managing components such as hash cores for authentication processing is herein referred to as control logic. In one example, control logic manages cryptographic processing in components such as hash core **223**, hash core **225**, and temporary memory **221**.

In one embodiment, hash cores **223** and **225** as well as temporary memory **221** also receive input data from parser **203**. After a round of processing in hash core **223** or hash core **225**, data can be provided to temporary memory **221**. Input data that has undergone one or more rounds of hash operations is referred to herein as intermediate data. Temporary memory **221** can store the intermediate data and subsequently provide the intermediate data for additional rounds of hash processing through output port **281** to the input ports **273** and **275** associated with hash cores **223** and **225**. According to various embodiments, hash cores **223** and **225** both are

capable of performing either MD5 or SHA-1 processing. After the specified number of rounds of hash processing have occurred as determined by the control logic **233**, hash cores **223** and **225** can provide the final or processed data through output port **283** and **285** to merger component **241**. Merger component **241** can then send the processed data to the external entity.

According to various embodiments, components for performing other operation such as XOR operations are also included in the cryptography accelerator. In one example, the XOR component is coupled to the output ports **283** and **285** so that SHA-1 and MD5 processed data can be combined together. It should be noted that the cryptography accelerator can include a number of other components including cryptography blocks such as DES, triple DES, and RC4 cores. The cryptography accelerator can include encryption functionality, central processing cores, bypass circuitry, etc.

FIG. **2**B is a diagrammatic representation providing one example on an instruction sequence is provided to control logic **233**. In one example, a parser loads a program counter with a pointer from a vector pointer table **211**. Each pointer **213**, **215**, **217**, and **219** may be configured to refer to an instruction sequence **251**, **253**, **255**, and **257**. According to various embodiments, each instruction sequence is a sequence of loads, stores, moves, sets, etc.. for performing cryptographic operations. In one example, the fetch engine gets the sequence of instructions from persistent memory as long as there is room in an instruction queue. The instructions are decoded in order to determine resource dependencies to allow instructions to be executed out of order. According to various embodiments, several hash engines are provided in a cryptographic accelerator and instructions are performed as resources become available. Consequently, mechanisms are provided to track the resource dependencies. In some examples, resources include memory ports, hash engine ports, and counters.

FIG. **2**C is a diagrammatic representation of one example of a mechanism for tracking resource dependencies. According to various embodiments, global resource vector **240** indicates which resources are being used. In one example, resource **242** represents a memory input port being used and resource **244** represents a hash engine input port that is in use. Dependency vector **260** shows which resources are needed for a particular instruction in an instruction sequence. In one example, resource **264** and **266** represent the hash engine input port and the hash engine output port are needed for the instruction to execute. Consequently, the instruction may not execute until the global resource vector returns to a state shown in vector **280**, when resource **284** and **286** representing the hash engine input and output ports become available.

FIG. **3** shows one example of a cryptographic handshake sequence between a client **301** and a server **303**. A wide variety of cryptographic handshake sequences associated with key exchanges are available. FIG. **3** is merely one example of a handshake. At **311**, the client **301** transmits a message with a security enable parameter to a server **303**. The authentication message contains an identifier such as a user name or an authentication identifier that allows the receiver to select an authentication mechanism out of a possible set of mechanisms. According to various embodiments, server **303** already has information associated with the client. The server **303** identifies the security enable parameter along with any client proposed algorithms and transmits an acknowledgement at **315** to client **301** indicating the selection of an algorithm.

As noted above, a client **301** transmits a user name to a server **303** and a server **303** at **315** transmits a value such as a salt associated with the user name back to the client **301**. According to other embodiments, protocol version, session ID, cipher suite, and compression method are exchanged along with a client random value and a server random value.

At **317**, client **301** computes the combined hash using the salt and the actual password associated with the user name. According to various embodiments, the client **301** then provides public information at **321** to server **303**. Similarly, server **303** at **325** provides public information to client **301**. Information that would not compromise security between a client and a server if accessed by a third party is referred to herein as public information. At **327**, both client **301** and server **303** can derive a common value such as a common symmetric key using values available to each of them. Many techniques for key derivation are available. According to various embodiments, a cryptographic accelerator with hash cores according to various embodiments are capable or deriving keys based on selected algorithms in a highly efficient manner.

For example, client **301** generates a common key using public information from server **303**, its own private information used to generate public information provided to server **303**, and the combined hash calculated by operating on the password appended to a salt. Similarly, server **303** generates a symmetric key by using public information from client **301**, a verifier derived from the hash of the combined salt and password, and private information used to generate public information provided to client **301**. If the password used to derive the verifier at server **303** is the same as the password used to generate the combined hash value at client **301**, the symmetric keys derived at client **301** and server **303** will be the same.

According to various embodiments, the session key can be used for communications between client **301** and server **303**. It should be noted that a variety of different cryptographic handshake sequences and communication sequences in general can use the techniques of the present invention. For example, a session key can further be hashed to derive a possibly stronger session key.

At **331**, client **301** sends a hash of the session key combined with other public information to server **303**. The server **303** then performs a hash of the derived session key combined with the other information known to server **303** to verify the identity of the client **301**. Similarly, at **335**, server **303** sends a hash of the session key along with other information known to client **301** to allow client **301** to verify the identify of server **303**. According to various embodiments, a cryptography accelerator with hash cores according to the techniques of the present invention makes generation of finished messages highly efficient.

It should be noted that in the above implementation, a password is never transmitted over the network. Instead, both network entities use derivatives of the password to generate the session key and other cryptographic information used for secure transmission. Both the password and the session key need not ever be transmitted over the network.

According to various embodiments, a cryptography accelerator speeds operations such as key derivation and finished message generation on both the server and the client side. It is contemplated that a cryptography accelerator can be used in any network entity. It should be noted that the cryptographic handshake sequence shown in FIG. **3** is only one example of a sequence that can use the mechanisms and techniques of the present invention.

FIG. **4** is a diagrammatic representation showing data processing according to various embodiments. In one embodiment, a premaster secret **401** is associated with input data

provided to a cryptography accelerator. The cryptography accelerator is used to apply a pseudo-random function (PRF) **411** to the premaster secret **401** to derive a master secret **403**. A function that takes one or more inputs and derives an indeterminate output is referred to herein as a pseudo-random function. A master secret **403** corresponds to intermediate data. Another pseudo-random function **413** can be applied to the master secret **403** to derive final or processed data such as authentication keys **405**, cryptography keys **407**, or initialization sectors **409**. The final data generated varies depending on the protocol, protocol version, and type of processing requested.

FIG. **5** is a flow process diagram showing one example of a technique for configuring components such as hash cores in a cryptography accelerator. At **501**, input data is received from a component such as parser. According to various embodiments, a parser organizes the data into a form readable by a hash core. At **503**, characteristics associated with the input data are determined. Information associated with how to process input data is referred to herein as characteristics of input data. Input data can include information such as protocol version, session ID, cipher suite, and compression method. In one example it is determined what algorithm is being applied to the input data. Algorithms can include versions of TLS, SSL, and IKE as well as other protocols and variants to the protocols.

Determining characteristics can also include determining what kind of operation is to be applied to the data. For example, a key may need to be derived from the data or finished message processing may need to be performed. At **505**, an instruction set is retrieved from persistent memory based on the characteristics associated with the input data. Persistent memory may include multiple instruction sets for configuring processing of input data in a variety of manners. At **507**, a hash core is configured based on the instruction set. It should be noted that other components such as XOR processing components and temporary memory may also be configured at this point.

Configuring the components may include loading microcode associated with the instruction set into control logic associated with the various hash cores and configurable components. Alternatively, instructions such as microcode can be loaded into a single control logic component associated with the various components. At **509**, input data is processed using the instruction set. After a round of processing, input data becomes intermediate data. At **513**, intermediate data is maintained in temporary memory during processing. Using temporary memory, data can be manipulated, padded, truncated, etc. At **515**, input data and intermediate data finally become final or processed data after completion of processing at **515**. The final or processed data is provided back to a merger component for forwarding to an external entity such as an external processor.

FIG. **6** is a process flow diagram showing operations for performing TLS 1.0 key derivation according to various embodiments. TLS 1.0 key derivation can be used during a cryptography handshake sequence such as that shown in FIG. **3**. A cryptography accelerator such as that shown in FIG. **2** having an MD5 core and a SHA-1 core can be used for key derivation. According to various embodiments, the inputs to the key derivation operations are a premaster secret, client random information, and server random information. At **601**, client and server random information is saved. Client/server random information can be saved in a component such as temporary memory.

At **603**, the length of the premaster secret is acquired. At **605**, the premaster secret is saved. At **607**, a prehash operation is performed on the premaster secret using the MD5 and a SHA-1 hash cores. At **611**, it is determined if the current session is a new session. If the current session is a new session, a 48-byte p_MD5 is generated at **613**, a 60-byte p_SHA-1 is generated at **615**, and the resulting p_MD5 and p_SHA-1 are combined with an XOR operation to acquire the master secret key. The 48-byte master secret key is saved at **619**. If it is determined at **611** that the current session is not a new session, the premaster secret to master secret generation is skipped.

At **621**, a prehash is performed on the master secret. At **623**, the number of bytes needed for the MD5 and SHA-1 operations is determined. The number of bytes needed can be determined by control logic, as the number of bytes needed may be one of the characteristics of the input data stream. Based on the number of bytes needed, MD5 and SHA-1 operations are performed at **625** and **627** using the MD5 and SHA-1 cores as configured by the control logic. The result is combined with an XOR at **629**. According to various embodiments, the operations such as sending data to an XOR component are determined based on an instruction set selected by the control logic. A persistent memory allows storage of instruction sets for a variety of operations.

It is determined at **631** if authentication is MD5 or SHA-1. If authentication is MD5 at **631**, MD5 inner and outer hash contexts are generated at **633**. Otherwise, SHA-1 inner and outer hash contexts are generated at **635**. It is determined at **641** whether exportable data is needed. In one example, export restrictions may limit the length of the key. If exportable data is needed, an exportable final write key is generated at **643**. If exportable data is not needed, the process is completed. It is also determined at **651** whether the key is needed for a block cipher or a stream cipher. If the key is needed for a block cipher, an initialization vector that is exportable is generated at **653**. Otherwise, the operations are complete for TLS 1.0 key derivation. It should be noted that when the operations are complete, data in temporary memory may be removed when the processed data is passed back to an external source. However, instructions sets for configuring the cryptography accelerator can remain in persistent memory.

FIG. **7** is a process flow diagram showing SSLv3 key derivation according to various embodiments. At **701**, client and server random information is saved. At **703**, it is determined whether the current session is a new session. If the current session is a new session, the master secret is generated by saving the premaster secret at **705**, computing an inner hash using a SHA-1 component at **707**, and computing an outer hash using an MD5 component at **711**. A 16-byte master secret is saved at **713**. The inner hash and outer hash computations are repeated three times at **715**.

If the current session is not a new session, the number of loops needed is determined at **717**. Control logic can determine the number of loops needed. The master secret is generated by computing an inner hash using a SHA-1 component at **719** and computing an outer hash using an MD5 component at **721**. The master secret is saved at **723**. The inner hash and outer hash computations are repeated based on the number of loops needed at **725**.

It is determined at **731** if authentication is MD5 or SHA-1. If authentication is MD5 at **731**, MD5 inner and outer hash contexts are generated at **733**. Otherwise, SHA-1 inner and outer hash contexts are generated at **735**. It is determined at **741** whether exportable data is needed. If exportable data is needed, an exportable final write key is generated at **743**. If exportable data is not needed, the process is completed. It is also determined at **751** whether the key is needed for a block cipher or a stream cipher. If the key is needed for a block

cipher, an initialization vector that is exportable is generated at **753**. Otherwise, the operations are complete for SSLv3 key derivation.

FIG. **8** is a flow processed diagram showing TLS 1.0 finished message generation, according to various embodiments. Finished message or verification messages are used to confirm that two network entities were successful in key exchange and authentication processes. The finished message is typically the first message associated with the recently negotiated algorithms, keys, and secret information. Network entities that receive finished messages verify that the contents are correct.

In one example, once a client has generated and sent its own finished message to a server and has received and validated a finished message from the server, the client can begin to send and receive application related data to the server. To generate a finished message according to TLS 1.0, the master secret is saved at **801**. At **803**, the length of the handshake message used for finished message generation is acquired. It is determined at **805** whether the handshake message length is less than 512 bits. If the handshake message length is not less than 512 bits, both SHA-1 and MD5 hash algorithms are performed on 512 bit blocks of the handshake message. At **811**, intermediate states are saved. At **813**, the last block of the handshake message is saved. If the handshake message length itself is less than 512 bits, the handshake message is simply saved at **813**. At **815**, the intermediate states are loaded.

At **817**, a final MD5 and SHA-1 hash are performed. The resulting data is loaded at **819** into a pseudo-random function. At **821**, a 16-byte p_MD5 hash is generated and at **823** at 10-byte SHA-1 hash is generated. The results are combined with an XOR at **825**. The client finished message is saved at **827** and concatenated with the last block at **833**. At **835**, intermediate states are loaded and a final MD5 and SHA-1 hash are generated for the server. The resulting data is loaded at **843** into a pseudo-random function. A 16-byte p_MD5 hash is generated at **851** and a 20-byte SHA-1 hash is generated at **853**. The result is combined with an XOR at **851**. The server finished message is saved at **853**.

FIG. **9** is a flow process diagram showing finished message generation for SSLv3. At **901**, the master secret is saved. At **903**, the length of the handshake message is determined. It is determined at **905** whether the handshake message length is less than 512 bits. If the handshake message length is not less than 512 bits, both SHA-1 and MD5 hash algorithms are performed on 512 bit blocks of the handshake message. At **911**, intermediate states are saved. At **913**, the last block of the handshake message is saved. If the handshake message length itself is less than 512 bits, the handshake message is simply saved at **913**. At **915**, the intermediate states are loaded. At **917**, a final MD5 and SHA-1 hash are performed.

At **919**, the master secret is loaded. An outer MD5 hash and SHA-1 hash are generated for the client at **921**. The client finished message is saved at **923** and concatenated with the last block at **925**. Intermediate states are loaded at **931**. An inner MD5 hash and SHA-1 hash are generated for the server at **933**. The master secret is loaded at **935**. At **937**, an outer MD5 hash and SHA-1 hash are generated for the server. The server finished messages are saved at **939**.

FIGS. **6-9** are process flow diagrams showing hash operations that can be performed according to various embodiments of the present invention. The operations can be performed using components such as hash cores, XOR components, and temporary memory configured using instruction sets maintained in persistent memory. It should be noted that the operations shown are specified for particular key derivation and finished message generation operations associated with TLS 1.0 and SSLv3. However, the techniques and mechanisms of the present invention should not be restricted to these two protocols and the specified versions of these two protocols.

While the invention has been particularly shown and described with reference to specific embodiments thereof, it will be understood by those skilled in the art that changes in the form and details of the disclosed embodiments may be made without departing from the spirit or scope of the invention. It is therefore intended that the invention be interpreted to include all variations and equivalents that fall within the true spirit and scope of the present invention.

What is claimed is:

1. An information system, the system comprising:

a cryptographic accelerator;

a processing unit coupled to said cryptographic accelerator; and

a memory coupled to said cryptographic accelerator,

wherein said memory is configured to store a plurality of instruction sets,

wherein said processing unit is configured to receive input data, select one or more instruction sets from said memory based on characteristics of said input data, communicate said selected one or more instruction sets to said cryptographic accelerator, and send at least a portion of said input data to said cryptographic accelerator, and

wherein said cryptographic accelerator comprises a hash core that can be configured upon receipt of said input data by said selected one or more instruction sets,

wherein said selected one or more instruction sets includes an instruction sequence that when executed by said hash core performs a cryptographic hash operation and further includes instructions for sequencing data as required by said cryptographic hash operation.

2. The system of claim **1**, further comprising a bus configured to manage communication between said cryptographic accelerator, said memory, and said processing unit.

3. The system of claim **2**, wherein said bus is a PCI bus and said cryptographic accelerator is connected to said bus via an on-chip PCI interface.

4. The system of claim **1**, further comprising a Local Area Network (LAN) coupled to said cryptographic accelerator.

5. The system of claim **1**, further comprising a Wide Area Network (WAN) coupled to said cryptographic accelerator.

6. The system of claim **1**, wherein said cryptographic accelerator comprises at least one of an application specific integrated circuit, a programmable logic device, and a field programmable gate array.

7. The system of claim **1**, wherein said cryptographic accelerator is integrated into said processing unit.

8. The system of claim **1**, wherein the system is implemented as a single integrated circuit.

9. The system of claim **1**, wherein said memory is a random access memory (RAM).

10. The system of claim **1**, wherein said memory comprises a persistent memory for storing said plurality of instruction sets.

11. The system of claim **10**, wherein said persistent memory is a read only memory.

12. The system of claim **10**, wherein said persistent memory is flash memory.

13. The system of claim **10**, wherein said persistent memory is a disk.

14. The system of claim **1**, wherein said cryptographic accelerator further comprises a second hash core that can be configured upon receipt of said input data.

15. The system of claim 14, wherein the first hash core is configured to operate as one of a MD5 and a SHA-1 hash core.

16. The system of claim 15, wherein said second hash core is configured to operate as one of a MD5 and a SHA-1 hash core.

17. The system of claim 14, wherein said cryptographic accelerator further comprises an XOR (output) processing component for processing the output of said first hash core and said second hash core.

18. The system of claim 17, wherein said XOR (output) processing component is configurable and said processing unit is further enabled to configure said XOR (output) processing component.

19. The system of claim 1, wherein said cryptographic accelerator further comprises other cryptographic blocks which receive the at least a portion of said input data.

20. The system of claim 1, wherein said cryptographic accelerator further includes a central processing core.

21. The system of claim 1, wherein said cryptographic accelerator further includes bypass circuitry.

22. The system of claim 1, wherein said processing unit further comprises a parsing device to parse said input data to determine the characteristics of said input data and what operations must be applied to said input data.

23. The system of claim 1, wherein said processing unit comprises a fetch engine to fetch said instruction set or sets from said memory.

24. the system of claim 1, wherein said processing unit comprises a decoder to decode said instruction set or sets.

25. The system of claim 1, wherein said processing unit comprises a control logic unit to control configurable components.

26. The system of claim 25, wherein said control logic can be loaded with microcode.

27. The system of claim 1, further comprising a merger device in communication with said cryptographic accelerator and said controller, wherein said merger device gathers final or processed data from the cryptographic accelerator or the controller and forwards it on an external entity.

28. A method for configuring a cryptography accelerator in a system having a processor coupled to the cryptography accelerator comprising:

receiving, at the processor, input data;

selecting, based on characteristics of the input data, an instruction set associated with a cryptographic hash operation required for processing at least a portion of the input data;

configuring, upon receipt of the input data, a hash core in the cryptography accelerator to perform the required cryptographic hash operation using the selected instruction set, wherein the selected instruction set includes an instruction sequence that when executed by said hash core performs the required cryptographic hash operation;

performing in the hash core the required cryptographic hash operation on the at least a portion of the input data; and

communicating the result of the required cryptographic hash operation to the processor.

* * * * *

Exhibit E

US008333317B2

(12) **United States Patent**
Buer et al.

(10) **Patent No.: US 8,333,317 B2**
(45) **Date of Patent: Dec. 18, 2012**

(54) **SYSTEM AND METHOD FOR AUTHENTICATING THE PROXIMITY OF A WIRELESS TOKEN TO A COMPUTING DEVICE**

(75) Inventors: **Mark Buer**, Gilbert, AZ (US); **Ed Frank**, Atherton, CA (US); **Nambi Seshadri**, Irvine, CA (US)

(73) Assignee: **Broadcom Corporation**, Irvine, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 896 days.

(21) Appl. No.: **10/955,806**

(22) Filed: **Sep. 30, 2004**

(65) **Prior Publication Data**

US 2005/0105734 A1 May 19, 2005

Related U.S. Application Data

(60) Provisional application No. 60/609,537, filed on Sep. 13, 2004, provisional application No. 60/507,586, filed on Sep. 30, 2003.

(51) **Int. Cl.**
G06K 5/00 (2006.01)

(52) **U.S. Cl.** **235/380**; 235/381; 235/375; 235/382; 713/155; 713/161; 713/169; 713/170; 713/168; 713/172; 713/184; 713/201; 726/2; 726/3; 726/4; 726/5; 726/21; 726/27; 726/28; 726/29; 726/30; 726/9; 380/255; 380/270; 455/411

(58) **Field of Classification Search** 726/9
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,324,647 A * 6/1967 Jedynak 368/6
5,237,616 A * 8/1993 Abraham et al. 713/193
5,668,876 A * 9/1997 Falk et al. 380/271
6,088,450 A 7/2000 Davis et al
7,069,444 B2 * 6/2006 Lowensohn et al. 713/185
(Continued)

OTHER PUBLICATIONS

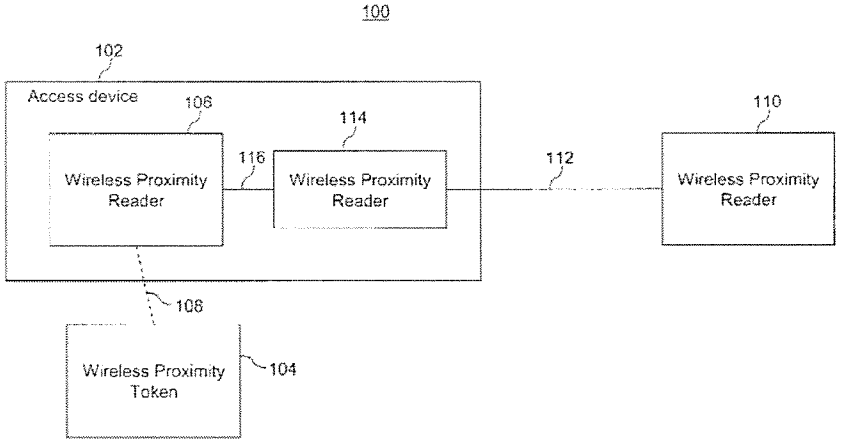
Communication from the EPO dated Jan. 3, 2005 enclosing 1) European Search Report Application No. 04023384.3-1245-, Search Completed Dec. 21, 2004, The Hague (3 pages).

Primary Examiner — Nathan Flynn
Assistant Examiner — Bryan Wright
(74) *Attorney, Agent, or Firm* — Sterne, Kessler, Goldstein & Fox P.L.L.C.

(57) **ABSTRACT**

Access to secured services may be controlled based on the proximity of a wireless token to a computing device through which access to the secured services is obtained. An authorized user may be provided access to a service only when a wireless token assigned to the user is in the proximity of the computing device. A user's credential may be stored on an RFID token and an RFID reader may be implemented within a security boundary on the computing device. Thus, the credential may be passed to the security boundary without passing through the computing device via software messages or applications. The security boundary may be provided, in part, by incorporating the RFID reader onto the same chip as a cryptographic processing component. Once the information is received by the RFID reader it may be encrypted within the chip. As a result, the information may never be presented in the clear outside of the chip. The cryptographic processing component may cryptographically encrypt/sign the credential received from the token so that assurance may be provided to a service provider that the credentials came from a token that was proximate to the computing device. An RFID reader, cryptographic processing components and a wireless network controller may be implemented on a single chip in a mobile device.

17 Claims, 11 Drawing Sheets



U.S. PATENT DOCUMENTS

7,188,360	B2 *	3/2007	Gerdes et al.	726/4
7,231,657	B2 *	6/2007	Honarvar et al.	726/2
7,239,865	B2 *	7/2007	Dyck et al.	455/411
7,357,312	B2 *	4/2008	Gangi	235/380
2001/0021950	A1	9/2001	Hawley et al.	
2002/0087867	A1	7/2002	Oberle et al.	
2003/0046541	A1 *	3/2003	Gerdes et al.	713/168
2003/0087629	A1	5/2003	Juitt et al.	
2004/0153649	A1 *	8/2004	Rhoads et al.	713/176
2004/0203638	A1 *	10/2004	Chan et al.	455/414.1
2005/0058292	A1 *	3/2005	Diorio et al.	380/270
2005/0165695	A1 *	7/2005	Berardi et al.	705/64

* cited by examiner

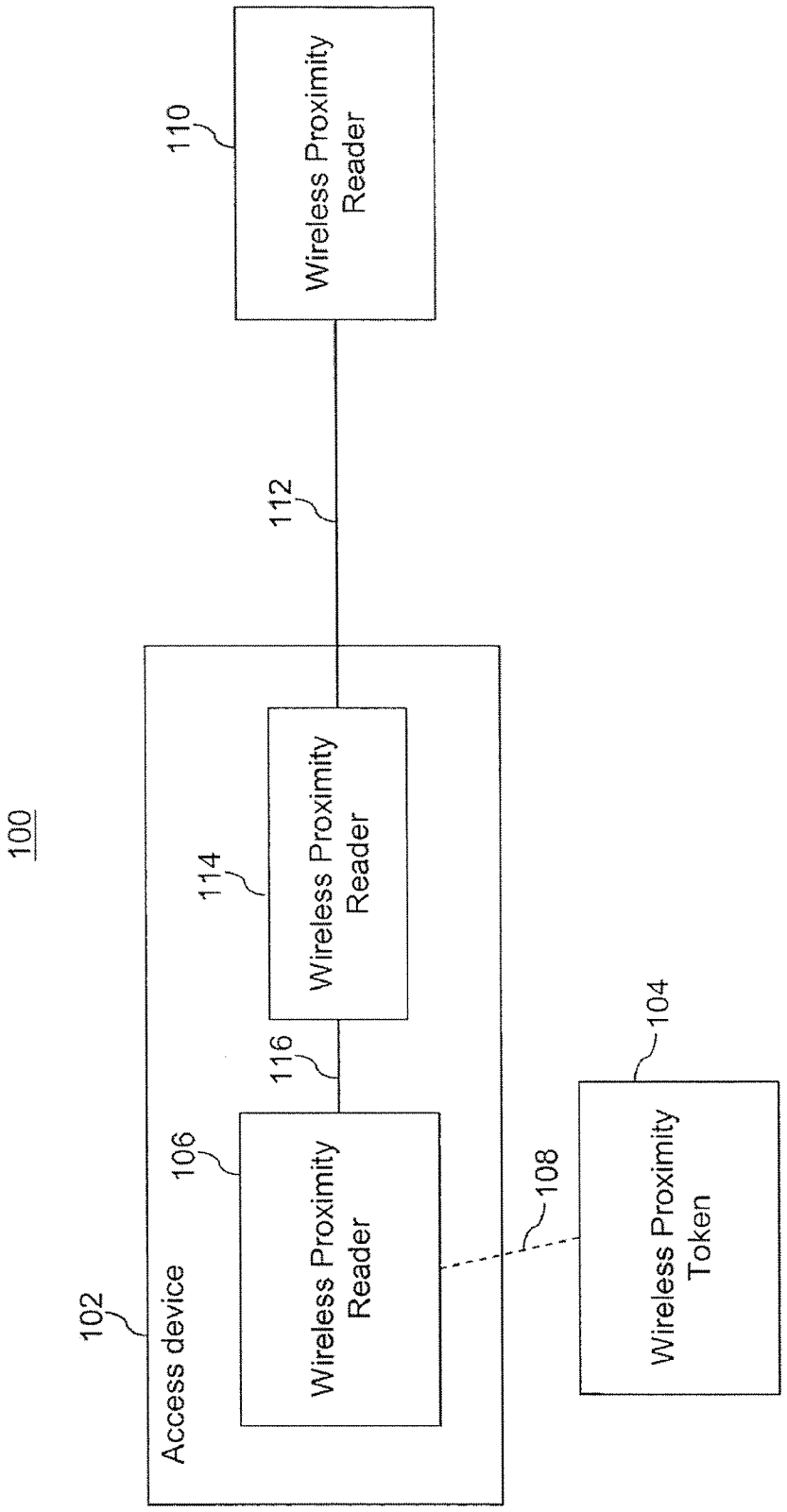

100
Access device
102
Wireless Proximity Reader
106
Wireless Proximity Reader
110
Wireless Proximity Reader
114
Wireless Proximity Token
104
108
112
116

FIG. 1

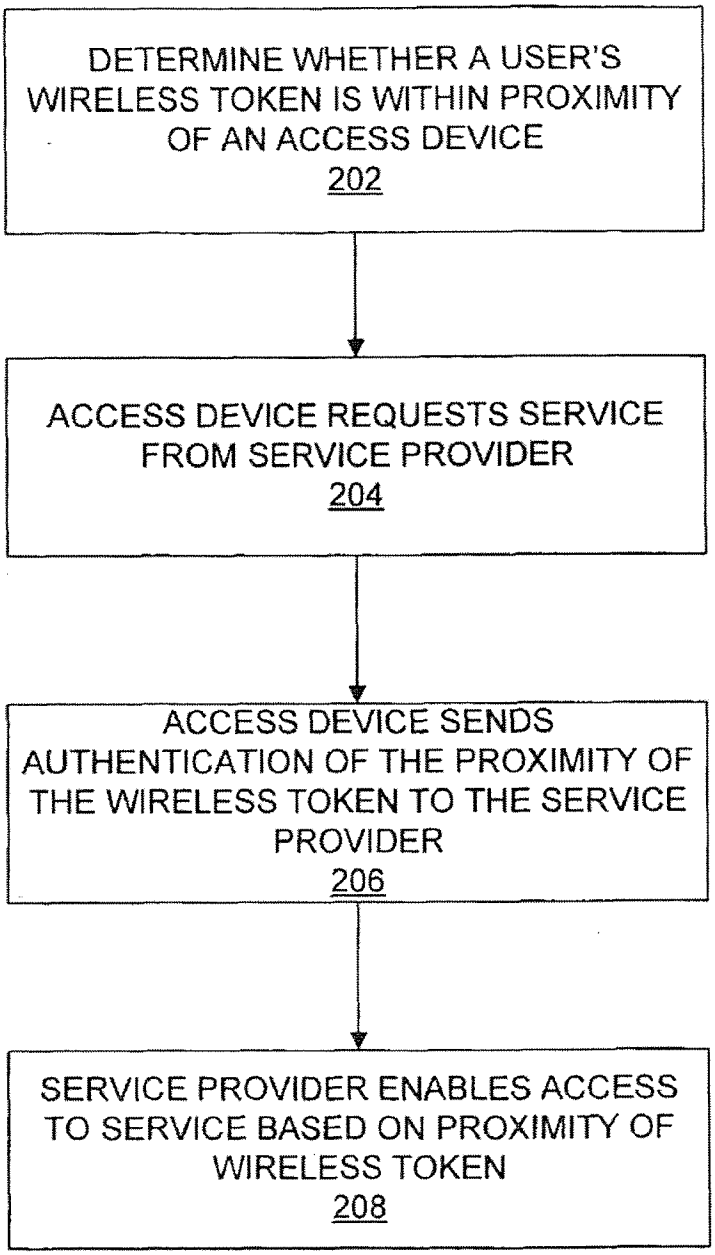

DETERMINE WHETHER A USER'S WIRELESS TOKEN IS WITHIN PROXIMITY OF AN ACCESS DEVICE
202
ACCESS DEVICE REQUESTS SERVICE FROM SERVICE PROVIDER
204
ACCESS DEVICE SENDS AUTHENTICATION OF THE PROXIMITY OF THE WIRELESS TOKEN TO THE SERVICE PROVIDER
206
SERVICE PROVIDER ENABLES ACCESS TO SERVICE BASED ON PROXIMITY OF WIRELESS TOKEN
208

FIG. 2

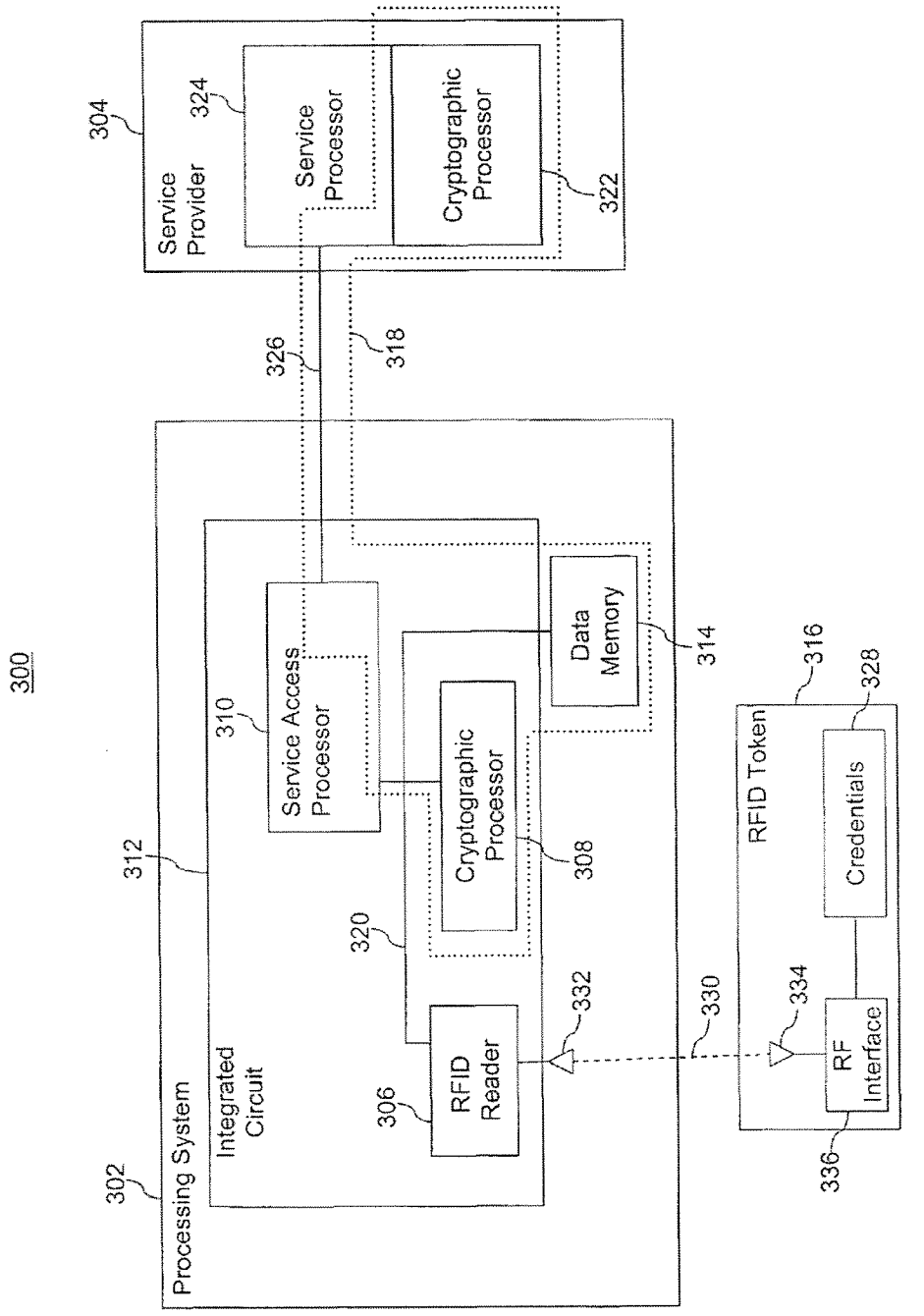

300
304
Service Provider
324
Service Processor
Cryptographic Processor
322
326
318
Processing System
302
Integrated Circuit
312
Service Access Processor
310
Data Memory
314
Cryptographic Processor
308
320
RFID Reader
306
332
330
RFID Token
316
328
Credentials
334
RF Interface
336

FIG. 3

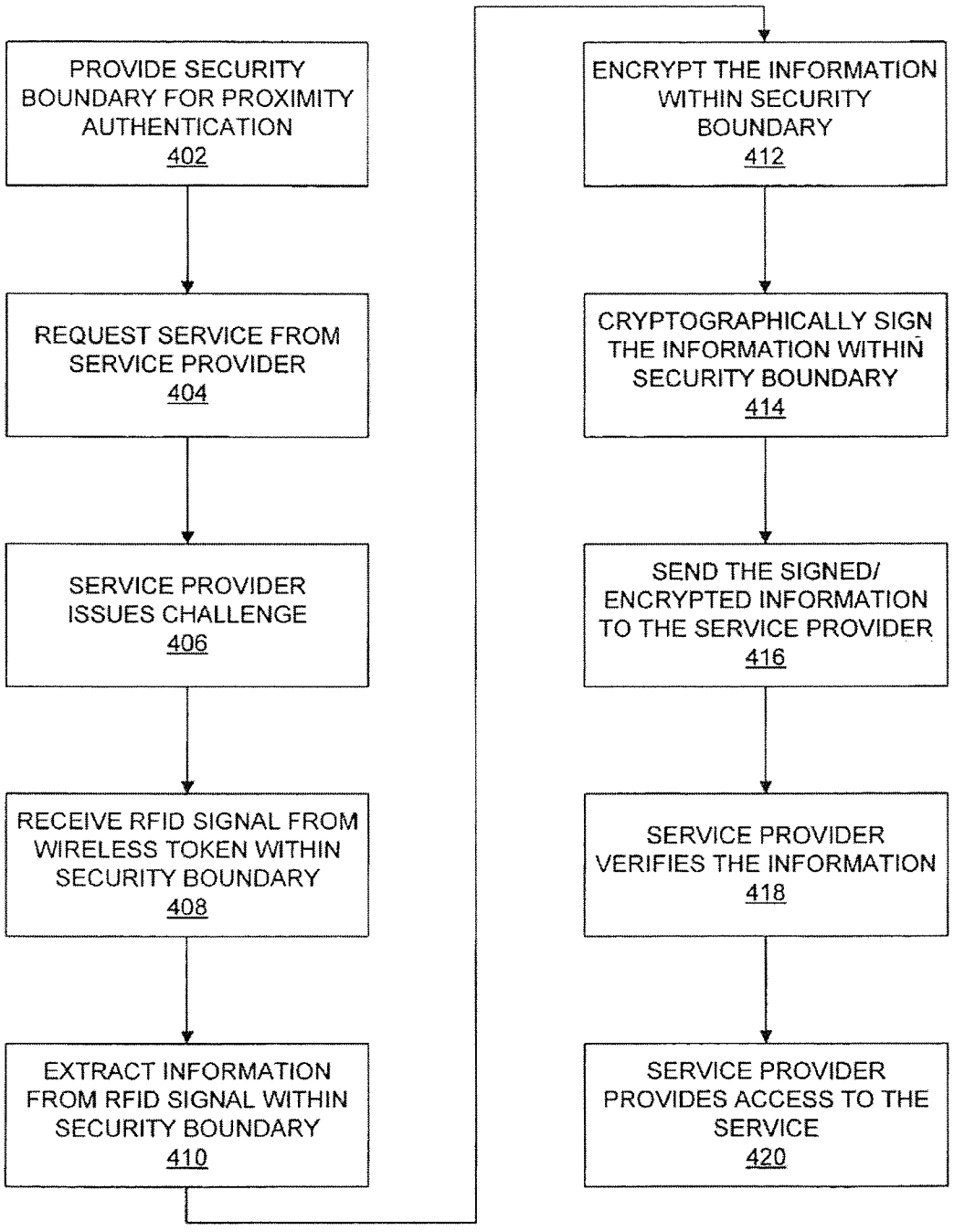

PROVIDE SECURITY BOUNDARY FOR PROXIMITY AUTHENTICATION
402
REQUEST SERVICE FROM SERVICE PROVIDER
404
SERVICE PROVIDER ISSUES CHALLENGE
406
RECEIVE RFID SIGNAL FROM WIRELESS TOKEN WITHIN SECURITY BOUNDARY
408
EXTRACT INFORMATION FROM RFID SIGNAL WITHIN SECURITY BOUNDARY
410
ENCRYPT THE INFORMATION WITHIN SECURITY BOUNDARY
412
CRYPTOGRAPHICALLY SIGN THE INFORMATION WITHIN SECURITY BOUNDARY
414
SEND THE SIGNED/ ENCRYPTED INFORMATION TO THE SERVICE PROVIDER
416
SERVICE PROVIDER VERIFIES THE INFORMATION
418
SERVICE PROVIDER PROVIDES ACCESS TO THE SERVICE
420

FIG. 4

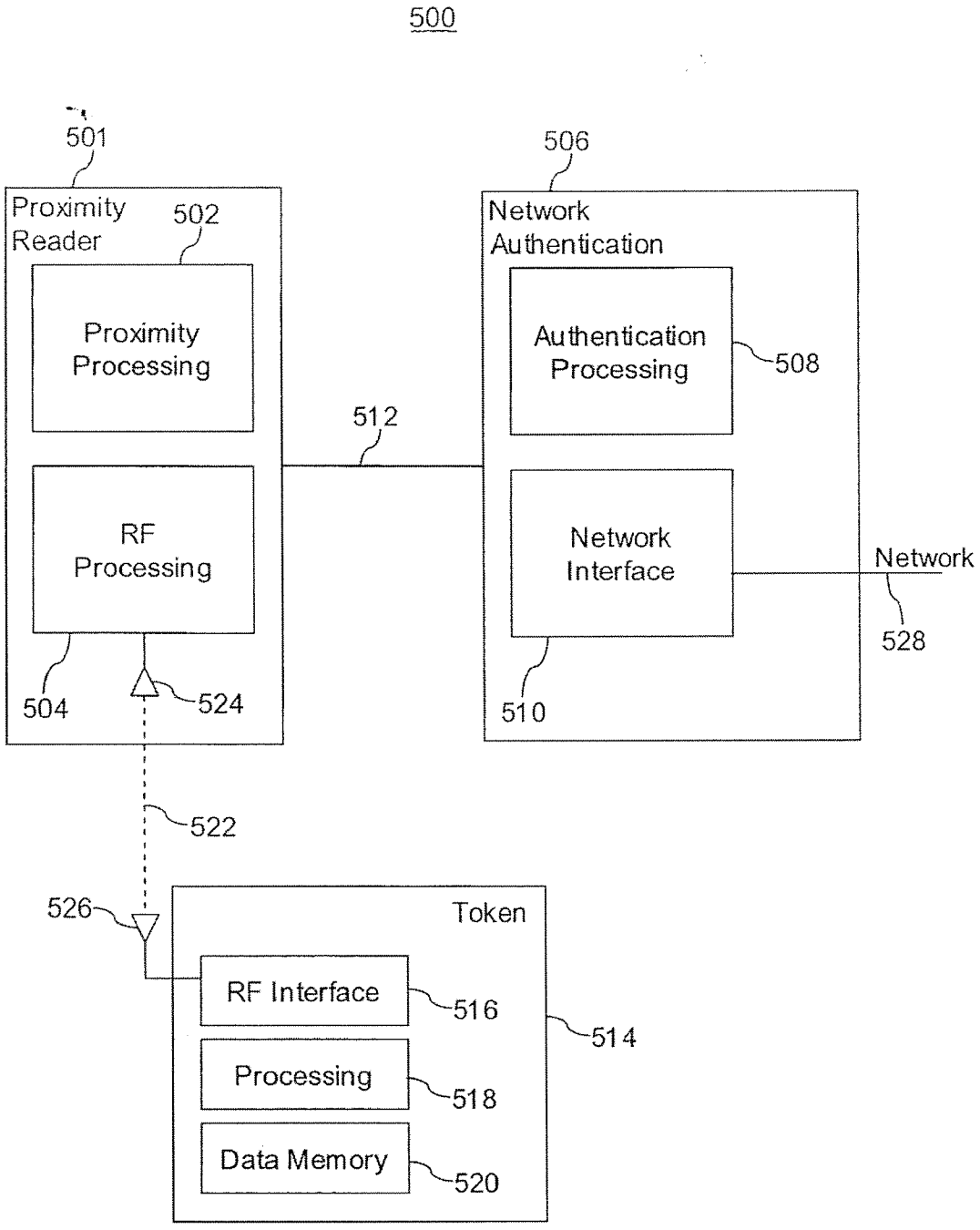

500
501
Proximity Reader
502
Proximity Processing
RF Processing
504
524
512
506
Network Authentication
Authentication Processing
508
Network Interface
510
Network
528
522
526
Token
RF Interface
516
514
Processing
518
Data Memory
520

FIG. 5

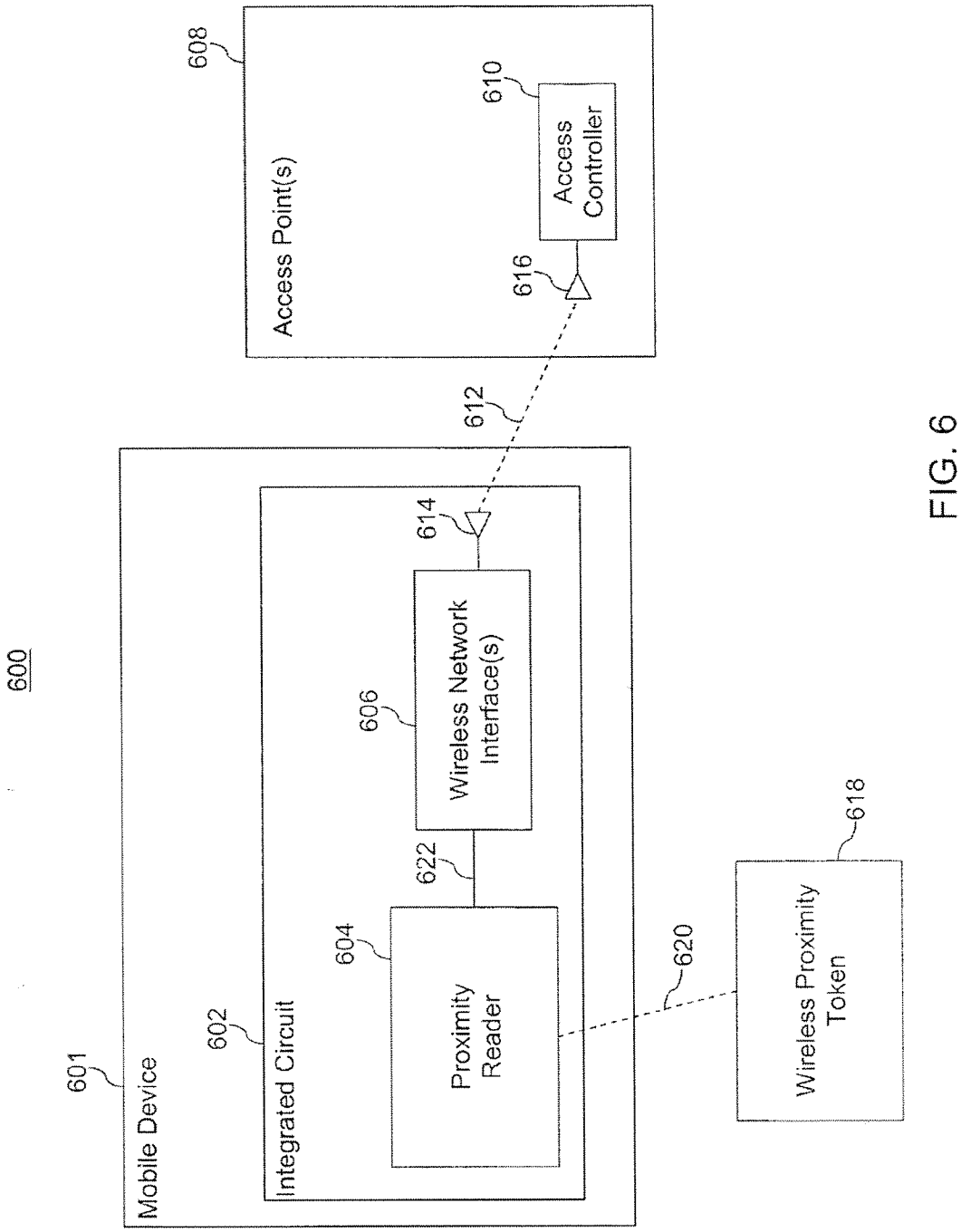

600
601
Mobile Device
602
Integrated Circuit
604
Proximity Reader
606
Wireless Network Interface(s)
608
Access Point(s)
610
Access Controller
612
614
616
618
Wireless Proximity Token
620
622

FIG. 6

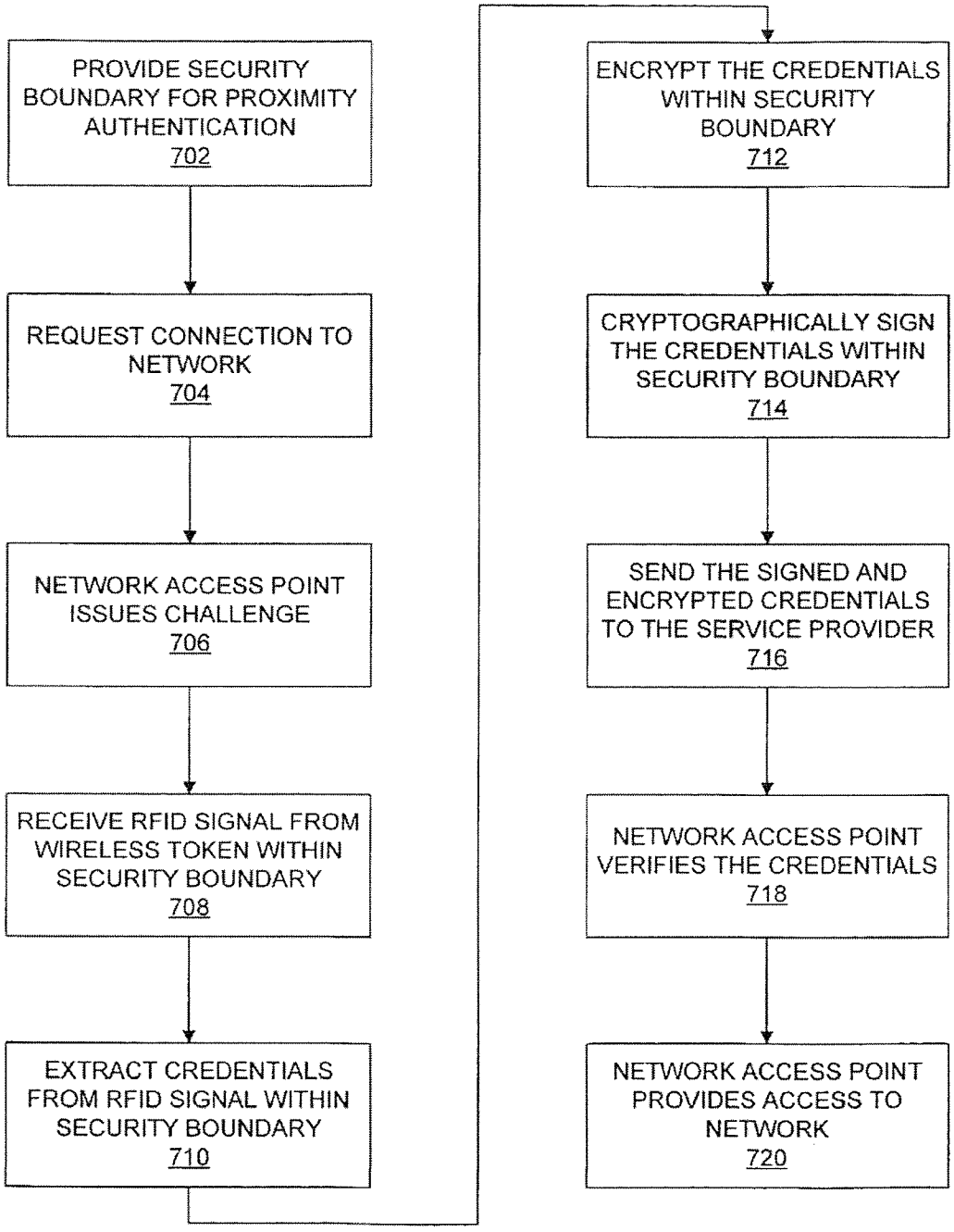

PROVIDE SECURITY BOUNDARY FOR PROXIMITY AUTHENTICATION 702
REQUEST CONNECTION TO NETWORK 704
NETWORK ACCESS POINT ISSUES CHALLENGE 706
RECEIVE RFID SIGNAL FROM WIRELESS TOKEN WITHIN SECURITY BOUNDARY 708
EXTRACT CREDENTIALS FROM RFID SIGNAL WITHIN SECURITY BOUNDARY 710
ENCRYPT THE CREDENTIALS WITHIN SECURITY BOUNDARY 712
CRYPTOGRAPHICALLY SIGN THE CREDENTIALS WITHIN SECURITY BOUNDARY 714
SEND THE SIGNED AND ENCRYPTED CREDENTIALS TO THE SERVICE PROVIDER 716
NETWORK ACCESS POINT VERIFIES THE CREDENTIALS 718
NETWORK ACCESS POINT PROVIDES ACCESS TO NETWORK 720

FIG. 7

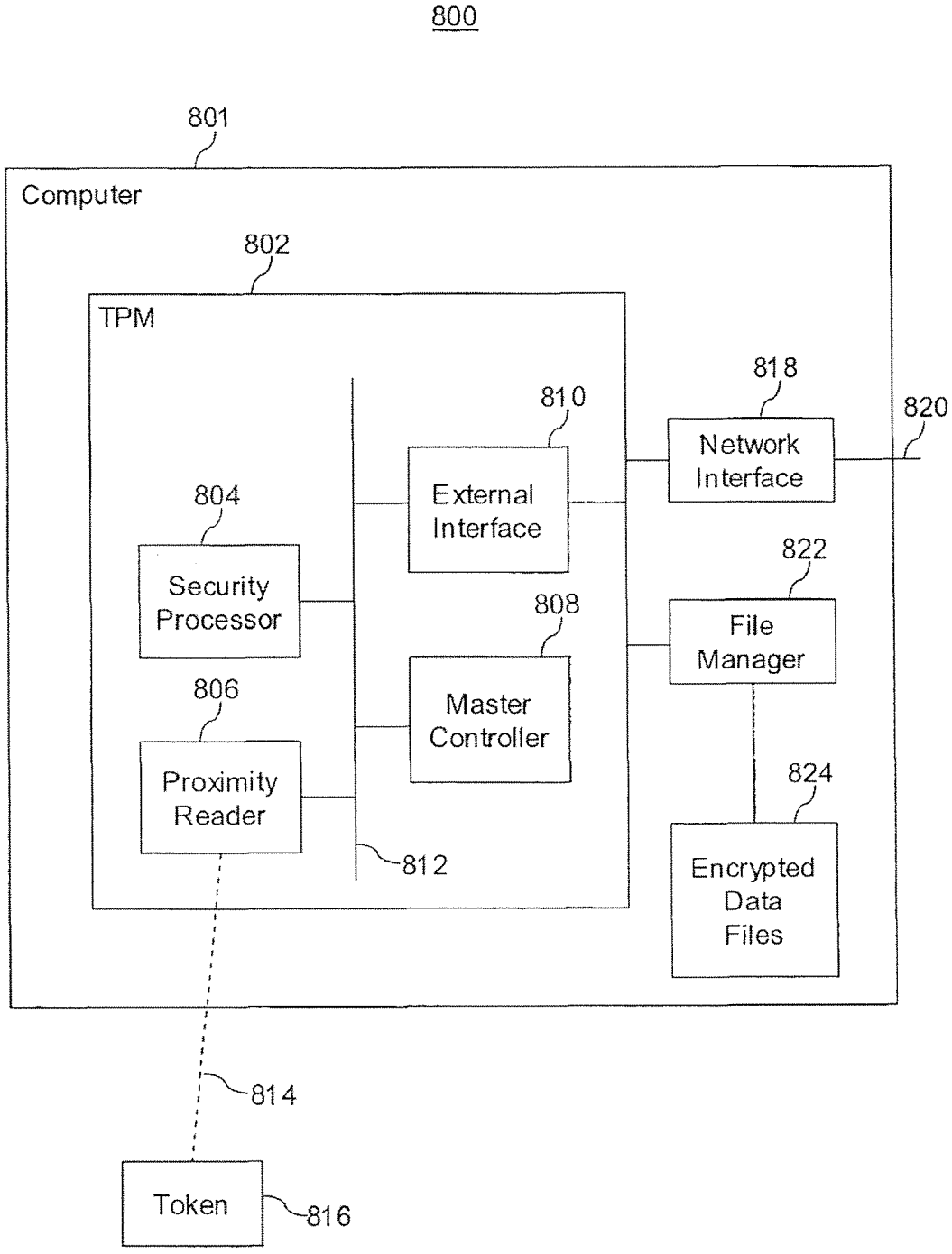

800
801
Computer
802
TPM
804
Security Processor
806
Proximity Reader
808
Master Controller
810
External Interface
812
814
816
Token
818
Network Interface
820
822
File Manager
824
Encrypted Data Files

FIG. 8

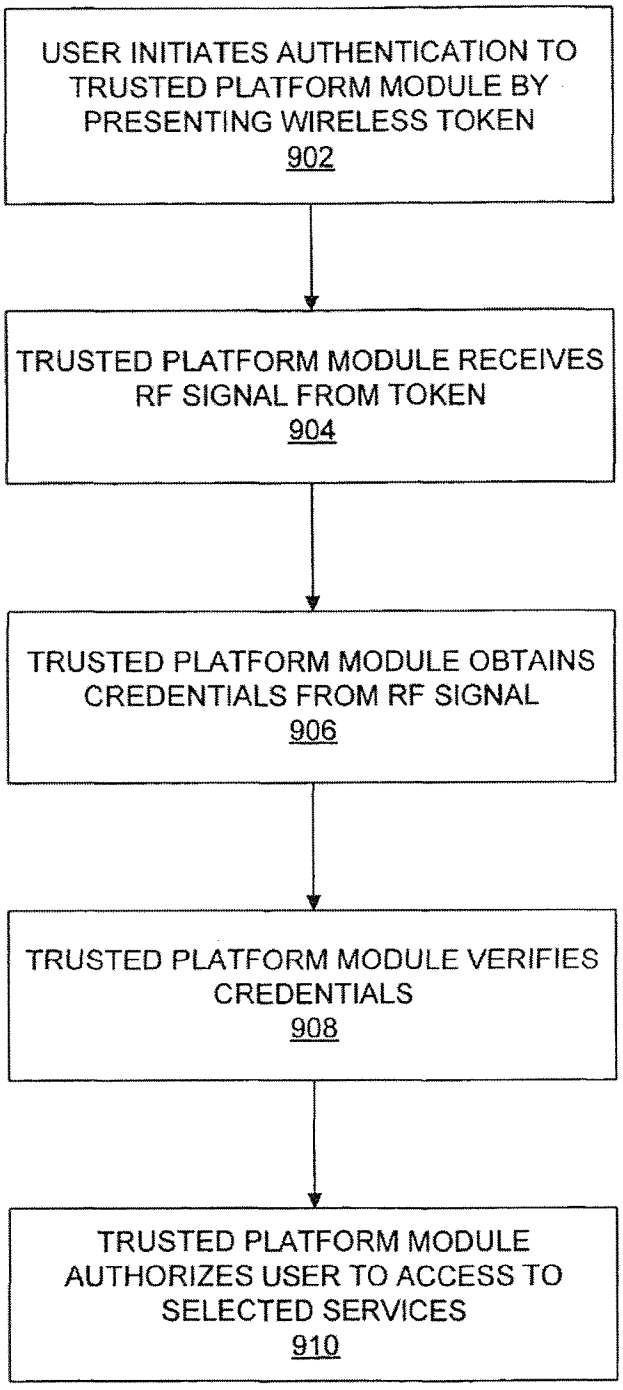

USER INITIATES AUTHENTICATION TO TRUSTED PLATFORM MODULE BY PRESENTING WIRELESS TOKEN
902
TRUSTED PLATFORM MODULE RECEIVES RF SIGNAL FROM TOKEN
904
TRUSTED PLATFORM MODULE OBTAINS CREDENTIALS FROM RF SIGNAL
906
TRUSTED PLATFORM MODULE VERIFIES CREDENTIALS
908
TRUSTED PLATFORM MODULE AUTHORIZES USER TO ACCESS TO SELECTED SERVICES
910

FIG. 9

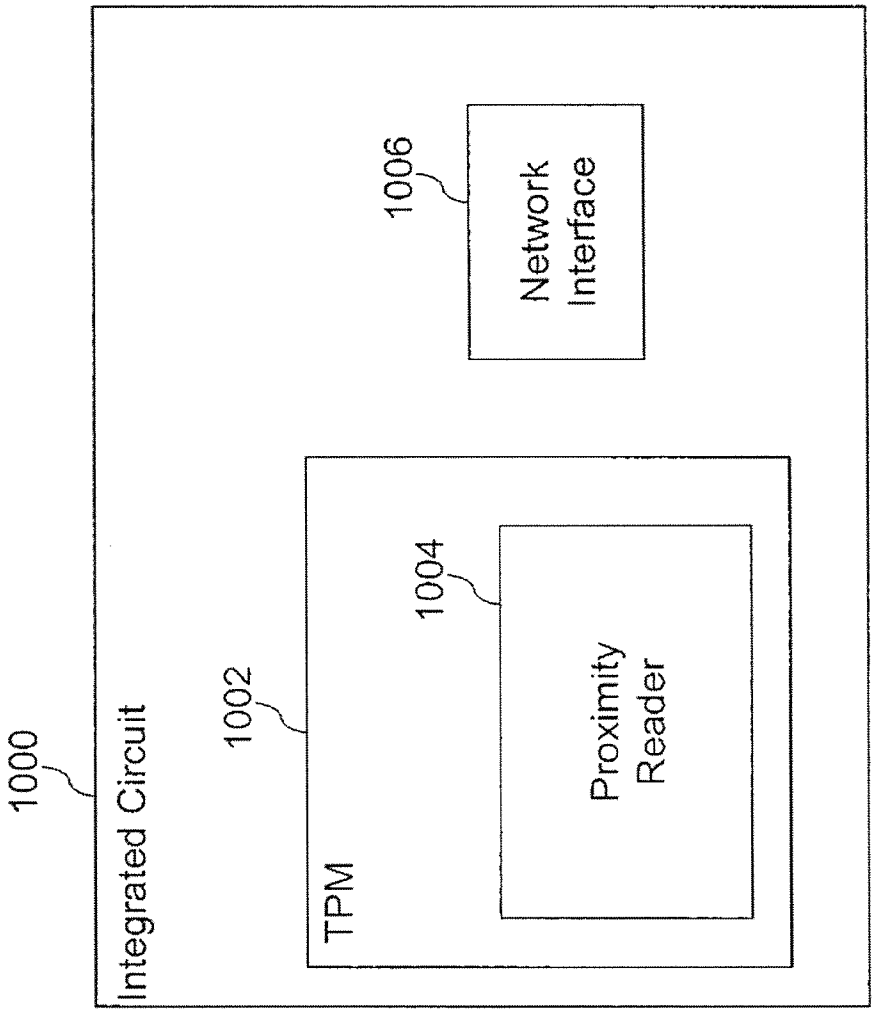

1000
Integrated Circuit
1002
TPM
1004
Proximity Reader
1006
Network Interface

FIG. 10

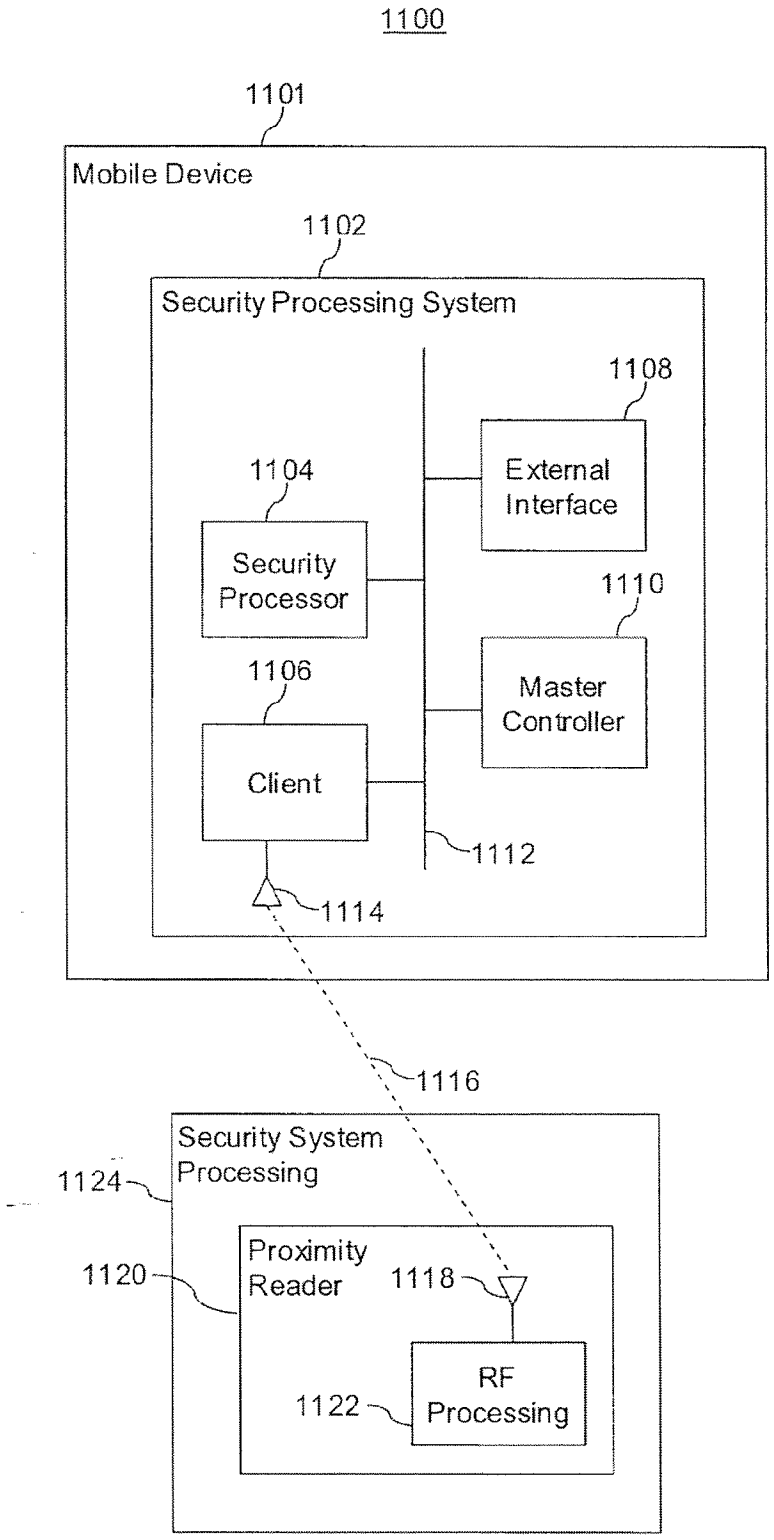

1100
1101
Mobile Device
1102
Security Processing System
1108
External Interface
1104
Security Processor
1110
Master Controller
1106
Client
1112
1114
1116
Security System Processing
1124
Proximity Reader
1120
1118
RF Processing
1122

FIG. 11

SYSTEM AND METHOD FOR AUTHENTICATING THE PROXIMITY OF A WIRELESS TOKEN TO A COMPUTING DEVICE

CROSS-REFERENCE TO RELATED APPLICATION(S)

This application claims the benefit of U.S. Provisional patent application Ser. No. 60/609,537, filed Sep. 13, 2004, and U.S. Provisional Patent Application Ser. No. 60/507,586, filed Sep. 30, 2003, the disclosures of which are hereby incorporated by reference herein.

TECHNICAL FIELD

This application relates to data communication systems and, more specifically, to techniques for authenticating proximity of a wireless token in a communication system.

BACKGROUND

A variety of security techniques are known for protecting information in and controlling the operation of a computing device such as a personal computer ("PC"), a server or a mobile device. For example, physical and/or cryptographic techniques may be employed to control access to the computing device and to data stored in the computing device.

Physical security techniques may include locating the computing device in a secure location, locking the computing device in an enclosure, protecting integrated circuits (i.e., chips) from invasive monitoring by encapsulating the chips in, for example, an epoxy.

Cryptographic techniques may include one or more of encryption, decryption, authentication, signing and verification. In some applications data encryption and decryption techniques may be used to prevent unauthorized applications or persons from accessing data stored in the computing device. For example, security passwords that are used to restrict access a PC may be stored on the PC in an encrypted form. The operating system may then decrypt password when it needs to compare it with a password typed in by a user.

In some applications authentication techniques may be used to verify that a given set of data is authentic. For example, when a server receives a message from a remote client, authentication information associated with the message may used to verify that the message is from a specific source. In this way, the server may ensure that only authorized clients access the applications and data provided by the server.

In practice, there may be circumstances under which the process of sending secret credentials such as a password or cryptographic key may be compromised. For example, when a user uses a computing device to access a secured service, the user may first need to enter the secret credentials into the computing device. The computing device may then forward these credentials to a service provider that then determines whether the user is authorized to use the requested service.

In the event the computing device has been comprised by a hacker or a computer virus, an unauthorized person may gain access to these credentials. As a result, an unauthorized person may be able to access the secured service. Serious consequences may result when the secured service includes sensitive information such as financial data or personal information. Accordingly, a need exists for improved techniques for-providing access to secured services.

SUMMARY

The invention relates to a system and method for authenticating the proximity of a wireless token to a computing device. For convenience, an embodiment of a system constructed or a method practiced according to the invention will be referred to herein simply as an "embodiment."

In one aspect, the invention relates to a system and method for providing access to a secured service based on a user's proximity to a proximity reader. Once the proximity is authenticated the user may then be allowed to access the secured service.

In some embodiments an authorized user is provided access to a service only when a wireless token assigned to the user is in the proximity of a computing device through which access to the secured services is obtained. In this way, a reasonable assumption may be made that the authorized user is in fact using the computing device to request the service. In contrast, if the request was being made by a hacker or a computer virus, access may be denied since the token may not be in the proximity of the computing device.

In some embodiments a user's credential are stored on an RFID token and an RFID reader is implemented within a security boundary on the computing device. In this way, the credential may be passed to the security boundary without passing through the computing device via software messages or applications. As a result, the credentials may not be intercepted by a hacker or computer virus that may have compromised the software executing on the computing system.

In some embodiments the security boundary may be provided, in part, using tamper resistant and/or tamper evident hardware. Thus, in the event the computer was physically tampered with in an attempt to compromise the security of the security boundary, such tampering may be ineffective or it may be evident to the user. In the latter case, the user may then take appropriate steps to re-secure the system.

In some embodiments, the RFID reader is incorporated onto the same chip as a cryptographic processing component. In this way, once the information from the RFID token is received by the RFID reader it may be encrypted within the chip. As a result, the information may never be presented in the clear (e.g., unencrypted) outside of the chip. Accordingly, the information may only be compromised by a clandestine RFID reader or by inspecting the internal contents of the chip. In conventional commercial settings, these scenarios may be unlikely. Accordingly, a system constructed according to the invention may provide improved access control for secured services.

In some embodiments, a cryptographic processing component may cryptographically encrypt and/or sign credentials received from a token. Thus, when a service provider receives the credentials, a high level of assurance may be provided to the effect that the credentials came from a token that was proximate to the particular computing device.

In some embodiments an RFID reader, a cryptographic processing component and one or more wireless network controller(s) may be implemented on a single chip in a mobile device. This may provide a cost effective and secure mechanism to limit access to the wireless network(s). In this case, network access may only be provided to the mobile device when a token is proximate to the mobile device and when that token has been assigned to an authorized user of that mobile device and the network(s).

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features, aspects and advantages of the present invention will be more fully understood when con-

sidered with respect to the following detailed description, appended claims and accompanying drawings, wherein:

FIG. **1** is a simplified block diagram of one embodiment of a proximity-based authentication system constructed in accordance with the invention;

FIG. **2** is a flow chart of one embodiment of proximity-based authentication operations that may be performed in accordance with the invention;

FIG. **3** is a simplified block diagram of one embodiment of a proximity-based authentication system constructed in accordance with the invention;

FIG. **4** is a flow chart of one embodiment of proximity-based authentication operations that may be performed in accordance with the invention;

FIG. **5** is a simplified block diagram of one embodiment of a proximity-based network authentication system constructed in accordance with the invention;

FIG. **6** is a simplified block diagram of one embodiment of a proximity-based wireless network authentication system constructed in accordance with the invention;

FIG. **7** is a flow chart of one embodiment of proximity-based network authentication operations that may be performed in accordance with the invention;

FIG. **8** is a simplified block diagram of one embodiment of proximity-based authentication for a trusted platform module constructed in accordance with the invention;

FIG. **9** is a flow chart of one embodiment of proximity-based authentication operations for a trusted platform module that may be performed in accordance with the invention;

FIG. **10** is a simplified block diagram of one embodiment of an integrated circuit including a trusted platform module constructed in accordance with the invention; and

FIG. **11** is a simplified block diagram of one embodiment of proximity-based authentication for a trusted platform module constructed in accordance with the invention.

In accordance with common practice the various features illustrated in the drawings may not be drawn to scale. Accordingly, the dimensions of the various features may be arbitrarily expanded or reduced for clarity. In addition, some of the drawings may be simplified for clarity. Thus, the drawings may not depict all of the components of a given apparatus or method. Finally, like reference numerals denote like features throughout the specification and figures.

DETAILED DESCRIPTION

The invention is described below, with reference to detailed illustrative embodiments. It will be apparent that the invention may be embodied in a wide variety of forms, some of which may be quite different from those of the disclosed embodiments. Consequently, the specific structural and functional details disclosed herein are merely representative and do not limit the scope of the invention.

FIG. **1** illustrates one embodiment of a system **100** where selected services may be provided to a user via a computing device only when a wireless token assigned to a user is proximate to the computing device. Operations of the system **100** will be explained in more detail in conjunction with the flowchart of FIG. **2**.

As represented by block **202** in FIG. **2** an access device **102** (FIG. **1**) such as a computer includes components that may be used to determine whether a wireless token **104** assigned to a user or users is proximate to the access device **102**. For example, a wireless proximity reader **106** may be configured to receive signals **108** (e.g., RF signals) from the wireless proximity token **104**. The signals **108** may include information that uniquely identifies the wireless proximity token **104**. In addition, this information may include one or more credentials (e.g., a password) that may be used to access a secured service provided by a service provider **110**.

The determination of proximity between the token **104** and the reader **106** may be established using a variety of mechanisms depending on the application. In some embodiments, the token will not generate signals until it is within a given distance of the reader. This may be accomplished, for example, by using a relatively passive token that intercepts signals transmitted by the reader and transmits signals in response to the received signals. Different distances between the token **104** and the reader **106** may be defined as indicative of proximity depending on the requirements of the application and, in some cases, characteristics of the operating environment.

As represented by block **204**, the access device **102** may request access to a service from the service provider **110** by sending a signal over a communication media **112**. Depending upon the particular application, the communication media **112** may comprise, for example, electric wires, optical cables or air.

Typically, access to the service will be initiated by the user's interaction with the access device **102**. For example, the user may use a keyboard or pointing device (e.g., a computer mouse) to request the service. In conjunction with this the user may be asked to input a password and/or provide a biometric (e.g., a fingerprint) to a biometric reader to further verify the authenticity of the user. In this way, access to a service may be restricted until the user satisfies one or more verification queries including, for example, what the user knows (e.g., a password), what the user possesses (e.g., a token) and who the user is (e.g., a physical or biometric characteristic).

In some embodiments, the access device **102** may automatically request a predefined service once the user places the token **104** proximate the access device **102**. For example, the access device **102** may include a database (not shown) that matches a given token (or information from the token) with one or more default services. Thus, when a token associated with default services approaches the access device **102**, the access device **102** may automatically request the services from the service provider **110**.

As represented by block **206**, the access device **102** may send authentication-related information to the service provider **110** to indicate that the token **104** is proximate to the access device **102**. For example, the access device **102** may include an authentication component **114** such that the determination of whether the token **104** is proximate the access device **102** is performed in a secure manner. In addition, the information provided by the token may be maintained within the access device **102** in a secure manner. For example, the information may only pass between the reader **106** and the authentication component **114** via a connection **116** within a common integrated circuit.

In addition, the authentication component **114** may be in secure communication with the service provider **110**. This may be accomplished, for example, by placing the authentication component **114** and the service provider **110** on the same integrated circuit or within secured hardware. In addition, a cryptographically secured communication channel may be established between the authentication component **114** and the service provider **110**.

In some embodiments, the authentication information may include information from the token. In the case where the communications over the media **112** may be cryptographically secured, the authentication component **114** may process (e.g., encrypt or sign) the information before sending it to the

service provider **110**. Since communications from the access device **102** may be trusted in this example, the authentication component **114** thereby provides a cryptographically reliable authentication that the information is from a specific token that is proximate that particular access device. In other words the encryption or cryptographic signing of the information may provide the necessary authentication.

After the service provider **110** has received an authenticated indication that the token is proximate the access device **102**, the service provider **110** may then enable access to the requested service (block **208**). This process may involve verifying that the information sent from the token **104** includes a credential associated with an authorized user and or access device.

As used herein the term service may include, for example, access to data and/or a data processing service. Thus, a service may enable an access device to, for example, read or write data in a data memory, access encrypted data, use cryptographic keys, gain access to cryptographic material such as security associations and keys, access a web page, access a data network or access a processing application.

As used herein the term data may include any information that may be accessed by a computing device including, for example, data files, passwords and cryptographic security associations including keys.

As used herein the term access may include, for example, acquiring, using, invoking, etc. Thus, data may be accessed by providing a copy of the data to the access device. Data also may be accessed by enabling the access device to manipulate or use the data. As an example of the latter, once a user has been authorized to access a service a trusted platform module may use keys to perform operations for the user. For a data network, access may include, for example, sending and/or receiving data over the network. For a processing application access may include, for example, invoking, interacting with or using the application or loading the application onto the access device.

A service provider may comprise hardware and/or software that facilitate providing a service. For example, a service provider may consist of a processing system that processes requests for service, verifies whether the requester is authorized to access the service and provides or facilitates the requested access.

In practice, a service provider (e.g., service provider **110**) may be located local or remote with respect to the entity requesting service (e.g., access device **102**). For example, a local trusted platform module may control access to passwords in a computing system and a remote wireless access point may control a computing system's access to a data network connected to the access point.

An access device may comprise hardware and/or software that facilitate access to a service. For example, a service provider may comprise a computing system such as, without limitation, a personal computer, a server, a cellular phone, a personal data assistant ("PDA"), etc.

For convenience, FIG. **1** only depicts one token, access device and service provider. It should be understood, however, that a system may include any number of these components. For example, a user may use a token to access one or more services via one or more access devices. Thus, an access device may access services from multiple service providers. Also, multiple access devices may access the services provided by a given service provider.

Authorization to access a service may depend on the specific token and access device being used. For example, a user may be assigned one token to access certain services through certain access devices. In addition, the user may be assigned another token to access other services through the same or other devices. Also, multiple sets of information (e.g., credentials) may be included on a single token to enable a user to access different services or to enable multiple users to share a token.

A wireless proximity reader and token may be implemented using one or more of a wide variety of wireless proximity techniques. For example, the proximity reader and the token may support, without limitation, one or more of RFID, ISO 14443 and ISO 15693.

Tokens may be implemented in various physical forms depending upon the needs of the respective applications. For example, a token may be in a form that is easy to carry, similar to a plastic credit card, a "smart card" or a building access card. Also, a token may take the form of a tag or a label that may be attached to another article.

Examples of tokens may include, without limitation, smart cards, credit cards, dongles, badges, biometric devices such as fingerprint readers, mobile devices such as cellular telephones, PDAs, etc. In some embodiments, the token includes circuitry used in a typical smart card. For example, the token may store an encrypted password that may be sent to an authentication system.

Referring now to FIGS. **3** and **4** additional details of proximity-based authentication will be described. FIG. **3** depicts one embodiment of a proximity-based authentication system **300** where a processing system **302** is used to access services provided by a service provider **304**. FIG. **4** describes several proximity-based authentication operations.

As represented by block **402** in FIG. **4** a security boundary is provided within the processing system **302** to, for example, secure the process of gaining access to a service, including securing the authentication process and information used during the authentication process. This security boundary may be established, for example, using hardware and/or cryptographic techniques.

Hardware techniques for providing a security boundary may include, for example, placing components within a single integrated circuit. For example, as shown in FIG. **3** an RFID reader **306**, a cryptographic processor **308** and a service access processor **310** may be incorporated into a single integrated circuit **312**. Thus, any processes performed or information used or stored within the integrated circuit **312** may not be compromised absent physical access to the integrated circuit **310** and the use of an invasive technique for analyzing the internal operations and data of the integrated circuit **312**. For many applications, this form of hardware security boundary may provide an acceptably high level of security.

Other means may be provided to provide a security boundary. For example, one or more integrated circuits (e.g., integrated circuit **312**) may be protected by a physical structure using known techniques (e.g., epoxy encapsulation). Also, the processing system **302** and/or its internal components may be tamper resistant and/or tamper evident.

Cryptographic techniques for providing a security boundary may include encrypting any important information that is sent to or from the integrated circuit via non-secure paths in the system. For example, security associations and keys may only appear in the clear within the integrated circuit **312**. In the event keys need to be sent out of the integrated circuit **312** (e.g., to be stored in a data memory **314**), the keys may first be encrypted.

Similarly, any important information that is sent between the integrated circuit **312** and the service provider **304** may be encrypted. For example, information (e.g., credentials **328**) received from an RFID token **316** may be encrypted before being sent to the service provider **302**.

In FIG. **3** a cryptographic security boundary is represented by the dashed line **318**. The line **318** represents, in part, that encrypted information may be sent between the cryptographic processor **308**, the service access processor **310** and the data memory **314**. Thus, the information may be sent securely even though the mechanism through which this information is sent (e.g., a data bus **320**) may not be secure.

Encrypted information also may be sent between the integrated circuit **312** and a cryptographic processor **322** and a service processor **324** in the service provider **304** via a communication link **326**. In this case, the cryptographic processors **308** and **322** may perform key exchange and encryption, decryption and/or authentication operations necessary to send and receive the encrypted information and provide the information in the clear for internal processing.

In general, the form of protection provided within the system may depend on the requirements of a given application. For example, specifications such as FIPS-140-2 define various levels of security that may be implemented within a system The security boundary provided by the integrated circuit **312** and the cryptographic boundary **318** may be used to provide a secure mechanism for authenticating a user to access a service. For example, credentials **328** received from the RFID token **316** may be provided directly into the integrated circuit **312** via RF signals **330**. Thus, the credentials **328** need not be entered into the processing system **302** via a software mechanism or hardware that is accessible by software. Consequently, such information may not be compromised by hacking or a software virus at this stage of the process.

Once the information is in the integrated circuit **312** it may be protected by the physical boundary of the integrated circuit **312** and by the cryptographic boundary **318**. For example, provisions may be made to ensure that the information does not appear in the clear outside of the integrated circuit **312**. Thus even if rogue software in the processing system **302** were to gain access to the information outside of the chip **312**, the software would not be able to decrypt it without appropriate key information. However, the key information also may be protected within the integrated circuit **312** and the cryptographic boundary **318**. That is, the key information may not appear in the clear outside of the security boundary. As a result, the credentials may be securely routed to the service provider **304**.

Moreover, via this secured mechanism, the processing system **302** may reliably authenticate to the service provider **304** that a specific RFID token **316** is proximate the processing system **302**. First, as discussed above, the credentials may be received in a secure manner. Second, the effective "decision" as to whether the token **316** is adjacent may be made within the security boundary. The cryptographic processor **308** may then cryptographically sign this information using a secure protocol set up between the cryptographic processors **308** and **322** of the processing system **302** and the service provider **304**, respectively. Via this signature the service provider **304** may be assured that a given message came from a specific processing system (e.g., processing system **302**) and that the message has not been compromised. Accordingly, proximity of the token **316** to the processing system **302** may be used as a reliable method of authorizing access to a secured service provided by the service provider **304**.

Referring again to FIG. **4**, an example of operations that may be used to access a service will be described. As represented by block **404** the processing system **302** may request access to a service provided by the service provider **304**. As discussed above this access request may be initiated automatically or as a result of user input.

In some applications, the service provider **304** (via, for example, the service processor **324**) may issue a challenge in response to the request (block **406**). For example, the service processor **324** may request credentials such as a password. Inherent or explicit in this challenge may be a requirement that the user's token be proximate to the device that requested access (e.g., the processing system **302**).

As represented by block **408** when the RFID token **316** is within an appropriate range of the processing system **302**, the RFID reader **306** will receive an RFID signal **330** from the RFID token **316**. As discussed above, the RFID signal **330** may be received by the processing system **302** within a security boundary.

An example of an RFID signaling sequence follows. The RFID reader **306** generates RF signals that are broadcast via an antenna **332**. When a token **316** is relatively close to the RFID reader **306**, an RF signal (e.g., as represented by line **330**) may be received by an antenna **334** and processed by an RF interface **336** in the token **316**. In some embodiments, the reader and the token may designed to communicate with one another when they are within a predefined distance of each other.

In some embodiments, the received RF signal may be used to power and activate the token **316**. Thus, the token **316** may include circuitry that extracts energy from the received RF signal and converts this energy into power for the components on the token **316**.

The token **316** may include circuitry that stores information such as the credentials **328** discussed herein. Thus, authentication information such as network authentication credentials, passwords and/or certificates may be stored in a data memory (e.g., a non-volatile memory) on the token **316**.

As discussed in more detail below, when the token **316** is activated the RF interface **336** may generate an RFID signal that is broadcast by the antenna **334**. Circuitry in the token **316** may be configured to modulate this RFID signal so that it includes some or all of the information stored on the token **316**. The broadcast RFID signal (e.g., as represented by line **330**) may then be received by the antenna **332** and sent to the RFID reader **306**.

As represented by block **410**, the system **300** may be configured so that any information contained within the broadcast RFID signal may be extracted only within a security boundary. For example, as shown in FIG. **3**, the RDIF reader **306** that extracts the information from the RFID signal **330** may be located within an integrated circuit that includes other functionality to protect the information. For example, the integrated circuit **312** may include a cryptographic processor **308** that encrypts the information (block **412**) to prevent the information from being sent out of the integrated circuit **312** in the clear.

As represented by block **414**, this encryption process and/or another cryptographic process may be used to cryptographically sign the information. For example, the cryptographic processor **308** may use a private key to encrypt the information. A public key associated with this private key may be published with a certificate from a trusted entity. This certificate serves to verify that the public key is authentic. The cryptographic processor **322** may then use the public key to decrypt any encrypted information received from the cryptographic processor **308**. A complementary process may be used to securely send information in the other direction across the link **326**.

Accordingly, after the information is signed by the cryptographic processor **308**, the service access processor **310** sends the signed information to the service processor **324** via the link **326** (block **416**). In this way, the information is, in effect,

sent over a secured channel (as represented by the corresponding portion of the line **318**) even though the actual data path may not be secure.

The service processor **324** then sends the received information to the cryptographic processor **322** for decryption and/or authentication processing as necessary. The service processor **324** then verifies that the received information indicates that the user is authorized to access the requested service (block **418**). In conjunction with this process, the service processor **324** has received an indication via the cryptographic signature associated with the information that the token **316** is proximate the processing system **302**.

Accordingly, the service processor **324** may then provide access to the requested service. As discussed above, this may involve a variety of operations depending on the particular service requested. For example, if the service provider **304** is a key manager, the service may involve providing security associations or keys to the processing system. As discussed above, these keys may be sent to the processing system **302** via the secured channel (cryptographic boundary **318**). In addition, the processing system **302** may be configured so that these keys, etc., are maintained within a security boundary.

Referring now to FIGS. **5-7**, embodiments of a wireless proximity and authentication system that controls access to one or more data networks will be discussed. These systems may enable a user to use a device to access, for example, one or more wired or wireless networks after the system verifies that the user (e.g., the user's token) is relatively close to the device. In this way, a reasonable inference may be made that an authorized user (as opposed to an unauthorized user or code) is attempting to access the network(s).

In FIG. **5** a communication system **500** includes a proximity reader **501**, one or more network authentication components **506** and a wireless proximity token **514**. To access a particular data network **528** a user (not shown) initiates a connection and/or authentication process via authentication processing **508** in a corresponding network authentication component **506**. In response to the network's challenge to the request (e.g., the network's request for credentials) the user may bring the token **514** in relatively close physical proximity to the proximity reader **501**. The proximity reader **501** may then receive appropriate authentication information from the token **514** and forward the authentication information to the network authentication component **506**. The authentication processing **508** uses this authentication information to respond to the network's challenge. As a result, the user may be granted access to the data network **528**. This process may then be repeated using different network authentication components to gain access to additional networks, if desired.

In some embodiments the proximity reader **501** and the token **514** communicate via RF signals as represented by dashed line **522**. For example, a proximity processing component **502** controls RF processing **504** in the proximity reader **501** to generate RF signals that are broadcast via an antenna **524**. When the token **514** is moved close enough to the proximity reader **501** so that these RF signals are sufficiently strong, the signals may be received by an antenna **526** and processed by an RF interface **516**. In some embodiments, the received RF signals may be used to power and activate the token **514**. As discussed below, when the token **514** is activated the RF interface **516** may generate RF signals that are sent back to the RF processing **504**. The proximity processing component **502** may then process the received signals to verify that a particular token is within range. In addition, the proximity processing component **502** may process the information received from the token **514** before passing the information to the-network authentication component **506**.

The token **514** may include circuitry that provides the authentication information and proximity signaling. For example, authentication information such as network authentication credentials, passwords and/or certificates may be stored in a data memory **520** on the token. In some embodiments some or all of this authentication information may be encrypted.

The token **514** may include a processing component **518** that controls communication with the proximity reader **501**. For example, the processing component **518** may enable the token **514** to be programmed with the authentication information. In addition, the processing component **518** may process RF signals received from the proximity reader **501** to power and activate the token **514**. Also, the processing component **518** may control the generation of appropriate signals to send the authentication information to the proximity reader **501**.

The network authentication component **506** may connect to one or more of a variety of networks. For example, the network **528** may comprise a wired or wireless network. The network interface **510** provides appropriate hardware and/or software to connect to such a network. In some embodiments the network interface comprises a media access controller ("MAC").

Thus, the system may be used to access one or more wired networks including, without limitation, personal area networks ("PANs"), local area networks ("LANs") and wide area networks ("WANs"). These networks may support a variety of protocols including, without limitation, Ethernet-based protocols.

The wireless network(s) may be implemented using one or more of a wide variety of wireless solutions. For example, a wireless network may comprise, without limitation, cellular telephone networks such as Global System for Mobile communications ("GSM") and General Packet Radio Service ("GPRS"), wireless local area networks such as 802.11a, 802.11b and/or 802.11g (referred to herein for convenience as simply 802.11) and personal area networks such as Bluetooth. Here, a wireless network interface may comprise a media access controller that supports one or more of these networks. In addition multiple wireless network interfaces may be used to provide connectivity to several wireless networks (e.g., 802.11 and Bluetooth).

FIG. **6** depicts an integrated wireless proximity apparatus and wireless communications system **600** that provides wireless communication access using proof of proximity for authentication purposes. In this embodiment, a user may use a mobile device **601** (e.g., a desktop personal computer, a laptop computer, a cellular phone, a PDA, etc.) to access one or more wireless networks (e.g., a cellular network, an 802.11 network, a Bluetooth network, etc.).

The user may use the token **618** in a similar manner as discussed above to quickly and securely access the wireless network. Thus, access to the wireless network may be provided when a user moves the token **618** within relatively close proximity (e.g., as may be defined in the proximity standards listed above) to the reader **604**. The token then sends authentication information to the proximity reader **604** via RF signals **620**.

The proximity reader **604** may, in turn, send authentication information to one or more wireless network interface(s) **606**. Each network interface **606** may perform network authentication and connection processing to enable communication with a given wireless network. This may involve, for example, communicating with a respective access controller **610** in one or more access point(s) **608** via RF signals **612** transmitted and received via one or more antennas **614** and **616**.

FIG. **6** also illustrates that a wireless proximity apparatus and a wireless communications system may be integrated into a single integrated circuit (e.g., "chip") **602**. For example, in the embodiment of FIG. **6**, the proximity reader **604** and one or more wireless network interface(s) **606** (e.g., an 802.11 MAC and a Bluetooth MAC) are implemented within the chip **602**.

In this embodiment, sensitive information such as the authentication information from the token is not exposed in the mobile device outside the chip **602**. Rather, these signals are sent between the proximity reader **604** and the network interface **606** via internal leads **622** in the chip **602**. Accordingly, this information may be maintained within a security boundary as discussed herein.

In contrast, in embodiments where the proximity reader and the network interface are separate components in a computer (e.g., when line **612** in FIG. **6** represents a connection between separate integrated circuits), the authentication information may be passed between these components via conventional software message processing. If such a computer is compromised, the authentication information may be intercepted by unauthorized persons or code.

Referring to FIG. **7**, an example of operations that may be used to access a data network will be described. For convenience, the network authentication component **506** and/or the mobile device **601** may be referred to as an access device in the discussion that follows.

As represented by block **702**, the system may be configured to provide a security boundary as discussed herein. For example, as shown in FIG. **6** a proximity reader **604** and one or more wireless network interface(s) **606** may be incorporated into a single integrated circuit **602**. Thus, information from the token **618** may be securely used within the mobile device **601** by, for example, not allowing the information to leave the integrated circuit **602** in the clear. Moreover, because a wireless connection is used to communicate with one or more access point(s) **608**, the information may only appear within RF signals outside of the integrated circuit **602**. In addition, cryptographic techniques as discussed herein may be used to provide a security boundary for the information. Similar techniques as described herein also may be used to provide a security boundary for the proximity reader **501** and the network authentication component(s) **506** of FIG. **5**. To reduce the complexity of FIGS. **5** and **6**, any cryptographic processing that may be provided in these systems is not shown.

As represented by block **704** the access system may request a connection to a network via a corresponding network access provider (e.g., access point **608**). In a manner similar to the operations discussed above this request may be initiated automatically or as a result of user input.

In some applications, the network access provider may issue a challenge in response to the request (block **706**). Thus, a network access point may request a credential such as a password. Inherent or explicit in this challenge may be a requirement that the user's token be proximate to the access device.

As represented by block **708** when the wireless token (e.g., token **514** or **618**) is brought within an appropriate distance to the access device, the proximity reader (e.g., proximity reader **501** or **604**) will receive an RF signal (e.g., an RFID signal) from the token. As discussed above, the RF signal may be received within the security boundary of the access device.

As represented by block **710**, the system may be configured so that any information contained within the broadcast RF signal may be extracted only within a security boundary. This may be accomplished, for example, using some of the techniques discussed herein. As represented by blocks **712** and **714**, a cryptographic process may be used to cryptographically encrypt/sign the information. After the information is signed by the cryptographic process, the access device sends the signed information to the network access provider (e.g., access point **608**) as represented by block **716**.

After verifying that the credentials are associated with an authorized user at block **718**, the network access provider provides access to the requested network.

Referring now to FIGS. **8-11**, embodiments of security processing systems (i.e., key management systems) that include wireless proximity and authentication will be discussed. These systems may enable a user to access, for example, services controlled by a security processing system on a computer. As discussed herein, service may be provided after the system verifies that the user (e.g., the user's token) is relatively close to the computer.

At a minimum, a security processing system provides key management functions such as generating keys, setting policy (e.g., capabilities and security limits) for keys and using keys in accordance with the corresponding policy (e.g., using the keys securely). In some embodiments a security processing system may implement key backup and recovery, prepare keys for storage, provide key revocation and destruction, provide multiple layers of security and provide other key management functions. Key management functions may involve authorized key usage such as smart card K of N access control or a key may be linked to a particular user, application, etc. Key management functions may involve secure audit logs such as tracking key usage to provide an audit trail.

FIG. **8** depicts one embodiment of computing system **800** that incorporates a proximity reader **806** and a security processing system implemented as a trusted platform module ("TPM") **802**. This embodiment may be used to authenticate access to data networks or other data services in a similar manner as discussed above. For example, a user may access a wired data network **820** or the user may access encrypted data stored in data files **824** (e.g., data memory).

A TPM may provide a set of cryptographic capabilities that enable certain computer functions to be securely executed within the TPM environment (e.g., hardware). Specifications for a TPM are defined by the Trusted Computing Group organization. As shown in FIG. **8**, a TPM **802** (e.g., a TPM integrated circuit) may be incorporated into a computer.

In the embodiment of FIG. **8** the TPM **802** includes a master controller **808** for controlling the overall operation of the TPM, an external interface **810** for interfacing with other components in the computer and a security processor **804**. The security processor **804** may perform cryptographic operations including, for example, encryption, decryption, authentication and key management. In the embodiment of FIG. **8**, these components communicate via a bus **812**.

In one aspect of the embodiment of FIG. **8**, authorization information is stored on the token **816**. For example, the token may store authorization data such a 100 bit encrypted password. The user may use the token **816** (and in some applications, other information or biometrics as discussed above) to authenticate himself or herself to the TPM **802**.

In addition, network authentication credentials may be stored in a data memory (not shown) in the TPM **802**. The TPM **802** may use these credentials to authenticate itself to an associated network and/or service provider.

Referring to FIG. **9**, when a user wishes to access a service provided by the TPM **804**, the user brings the token **816** in relatively close proximity with the computer **801** (block **902**). In this way the user may authenticate himself or herself to the TPM **802**.

As represented by block **904**, the proximity reader **806** in the TPM **802** receives the RF signals **814** from the token **816** and extracts any embedded information such as the user's credentials (block **906**). After verifying the information (block **908**), the TPM **802** may provide the requested access or may facilitate acquiring access to a service from another processing entity (block **910**).

For example, the system of FIG. **8** may be used to access encrypted data (e.g., an encrypted password) stored in a local data memory **824**. In this case, the TPM **802** may store cryptographic information (e.g., keys, security associations, etc.) that enables the TPM to decrypt encrypted data. In some embodiments, encrypted data files **824** may be accessed via other components (e.g., a file manager **822**) in the computer **801**. The security of the encrypted data may be maintained in such components outside the TPM since decryption may be restricted to occur only within the TPM. In other words, sensitive information that is in the clear is not allowed to leave the secure boundary provided by the TPM **802**.

Alternatively, the TPM **802** may provide the information to a service provider connected to a network **820**. For example, the TPM **802** may send information such as network authentication credentials to one or more network access devices (not shown) via one or more network interfaces **818** to enable the user to gain access to one or more networks **820**.

In the embodiment of FIG. **8**, sensitive information such as the authentication information from the token is securely maintained in the computer inside the TPM. In a manner similar to that discussed above, this information may be provided directly to the TPM via the RF signal **814**.

In contrast, in embodiments where the proximity reader and the TPM are separate components in the computer, the authentication information may be passed between these components via the software stack. In this case, if the stack is compromised, the authentication information may be compromised as well.

The network interface(s) **818** may be used to connect to wired and/or wireless network(s) **820**. As discussed herein, cryptographic techniques may be used to ensure the security of data transferred between the TPM **802** and other devices connected to a network **820**. Accordingly, a network connection may be used, for example, to communicate with a key manager to obtain key information (e.g., security associations) and authorization for key usage.

As shown in FIG. **10**, in some embodiments a TPM **1002**, a proximity reader **1004** and one or more network interfaces (e.g., interface(s) **606** in FIG. **6**) **1006** may be combined on a single integrated circuit **1000**. In this way, sensitive information may be maintained within a security boundary defined by the chip **1000**.

In addition, the combination of these components within a single chip may reduce the cost of the system. For example, by providing all of these components within a single CMOS integrated circuit the per chip cost of these components may be relatively small. As a result, it may be cost effective to provide the features discussed herein into a mobile device.

The embodiment of FIG. **10** may be advantageous when the network interface(s) provides connectivity to wireless network(s). As discussed above, by providing the network interface(s) **1006** on the same chip **1000** as the other access device components, sensitive information (e.g., credentials, passwords, keys, etc.) may only leave the secure boundary of the chip **1000** via RF signals. Thus, this may provide a relatively inexpensive and secure path directly to the TPM **1002** for authorization of key usage.

In some embodiments, the computer may comprise a point-of-sale device. That is, the token **816** (e.g., a token with smart card functionality) may provide the user's credit card information to the computer **801** and the computer **801** could securely receive and verify this information to perform a sales transaction.

FIG. **11** depicts one embodiment of a system **1100** where a mobile device **1101** may be used to authenticate a user to another processing system (e.g., one that includes a security processing system **1124** such as a TPM). In this embodiment, the mobile device **1101** includes a security processing system **1102** (e.g., a TPM) that in turn includes an RFID client **1106**. As discussed above, the security processing system (e.g., TPM) may include, for example, a security processor **1104**, a master controller **1110** and an external interface **1108** that communicate via an internal bus **1112**.

An RF signal **1116** transmitted by an antenna **1114** may then be received by a nearby proximity reader **1120** via an antenna **1118**. In this way, for example, the mobile device **1101** (e.g., a laptop computer, a PDA, a cellular phone, etc.) may be used in a manner similar to a smart card. For example, the security processing system **1102** may store credit card information that may be sent to an RF processing component **1122** in a proximity reader **1120** in the security processing system **1124**. Once this information is within the security boundary provided by the security processing system **1124**, the information may be used, for example, by point-of-sale components (not shown) associated with the security processing system **1124** to access data services (e.g., perform a sales transaction).

From the above it should be appreciated that a wireless proximity and authentication system may be used to securely and efficiently access one or more data service(s) that may be associated with a variety of applications. These applications may include, without limitation, authentication applications relating to network access (RSA token), processing of biometrics, credit card transactions, and voice over Internet Protocol ("VoIP").

Different embodiments of the invention may include a variety of hardware and software processing components. In some embodiments of the invention, hardware components such as controllers, state machines and/or logic are used in a system constructed in accordance with the invention. In some embodiment of the invention, code such as software or firmware executing on one or more processing devices may be used to implement one or more of the described operations.

Such components may be implemented on one or more integrated circuits. For example, in some embodiments several of these components may be combined within a single integrated circuit. In some embodiments some of the components may be implemented as a single integrated circuit. In some embodiments some components may be implemented as several integrated circuits.

The components and functions described herein may be connected/coupled in many different ways. The manner in which this is done may depend, in part, on whether the components are separated from the other components. In some embodiments some of the connections represented by the lead lines in the drawings may be in an integrated circuit, on a circuit board and/or over a backplane to other circuit boards. In some embodiments some of the connections represented by the lead lines in the drawings may comprise a data network, for example, a local network and/or a wide area network (e.g., the Internet).

The signals discussed herein may take several forms. For example, in some embodiments a signal may be an electrical signal transmitted over a wire, light pulses transmitted over an optical fiber or electromagnetic (e.g., RF or infrared) radiation transmitter through the air.

A signal may comprise more than one signal. For example, a signal may consist of a series of signals. Also, a differential signal comprises two complementary signals or some other combination of signals. In addition, a group of signals may be collectively referred to herein as a signal.

Signals as discussed herein also may take the form of data. For example, in some embodiments an application program may send a signal to another application program. Such a signal may be stored in a data memory.

A wide variety of devices may be used to implement the data memories discussed herein. For example, a data memory may comprise Flash memory, one-time-programmable (OTP) memory or other types of data storage devices.

The components and functions described herein may be connected/coupled directly or indirectly. Thus, in some embodiments there may or may not be intervening devices (e.g., buffers) between connected/coupled components.

In summary, the invention described herein generally relates to an improved authentication system. While certain exemplary embodiments have been described above in detail and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of and not restrictive of the broad invention. In particular, it should be recognized that the teachings of the invention apply to a wide variety of systems and processes. It will thus be recognized that various modifications may be made to the illustrated and other embodiments of the invention described above, without departing from the broad inventive scope thereof. In view of the above it will be understood that the invention is not limited to the particular embodiments or arrangements disclosed, but is rather intended to cover any changes, adaptations or modifications which are within the scope and spirit of the invention as defined by the appended claims.

What is claimed is:

1. A communication system, comprising:

a proximity reader within a security boundary, configured to receive a radio frequency (RF) signal from a wireless token located within a proximity to the proximity reader and extract a credential from the received RF signal, wherein the wireless token is configured to include a plurality of credentials for accessing different services from one or more service providers;

a secure processor within the security boundary, configured to receive the credential from the proximity reader, to store the credential within the security boundary, and to cryptographically sign or encrypt the credential to allow a service provider system external to the security boundary to authenticate that the credential is from the wireless token; and

an external interface within the security boundary configured to interface with components outside of the security boundary to send the cryptographically signed or encrypted credential to the service provider system in response to a challenge from the service provider system,

wherein the security boundary is configured to prevent software that is executing external to the security boundary from accessing the credential extracted from the received RF signal by the proximity reader and stored by the secure processor.

2. The system of claim 1, wherein the proximity reader is an RFID reader.

3. The system of claim 1, wherein the secure processor is an authentication processor.

4. The system of claim 1, wherein the secure processor is a cryptographic processor.

5. The system of claim 1, further comprising:

a network interface coupled to the external interface and configured to interface to a network.

6. The system of claim 1, wherein the credential comprises a key.

7. The system of claim 1, wherein the wireless token comprises an RFID device, the RFID device comprising:

a data memory for storing the credential;

an RF circuit coupled to the data memory for generating a signal according to the credential; and

an antenna coupled to the RF circuit for transmitting the signal to the proximity reader.

8. The communication system of claim 1, wherein the credential is encrypted and the secure processor is further configured to decrypt the credential.

9. A method, comprising:

receiving, within a security boundary of an access device a radio frequency (RF) signal including a credential from a wireless token within a proximity to the access device, wherein the wireless token includes a plurality of credentials for accessing different services from one or more service providers;

extracting the credential from the RF signal within the security boundary;

cryptographically signing the extracted credential within the security boundary to generate a cryptographic signature to allow a service provider system external to the security boundary to authenticate that the extracted credential is from the wireless token; and

communicating the cryptographic signature outside the security boundary to the service provider system in response to a challenge from the service provider system,

wherein the security boundary is configured to prevent software that is executing external to the security boundary from accessing the credential extracted from the RF signal and cryptographically signed.

10. The method of claim 9, further comprising:

authenticating the credential.

11. The method of claim 9, further comprising:

verifying the identity of the wireless token based on the credential.

12. The method of claim 9, further comprising:

receiving access to a service provided by the service provider system after the the service provider system has authenticated the cryptographic signature.

13. A method, comprising:

receiving within a security boundary of an access device a radio frequency (RF) signal including a credential from a wireless token within a proximity to the access device, wherein the wireless token includes a plurality of credentials for accessing different services from one or more service providers;

extracting the credential from the RF signal within the security boundary;

cryptographically signing or encrypting the credential to allow a service provider system external to the security boundary to authenticate that the credential is from the wireless token;

authenticating the credential within the security boundary; and

communicating the signed or encrypted credential to the service provider system in response to a challenge from the service provider system,

wherein the security boundary is configured to prevent software that is executing external to the security bound-

ary from accessing the credential extracted from the RF signal and cryptographically signed.

14. The method of claim 13, wherein the cryptographically signing or encrypting the credential further comprises:

encrypting the credential within the security boundary and communicating the encrypted credential to the service provider.

15. The method of claim 13, further comprising:

providing access to a service provided by the service provider after the signed or encrypted credential has been authenticated by the service provider system.

16. The method of claim 13, further comprising:

verifying the identity of the wireless token based on the credential.

17. The system of claim 5, wherein the service provider system is coupled to the network and configured to provide access to at least one of data and a data processing service after the credential has been authenticated.

* * * * *

UNITED STATES DISTRICT COURT
CENTRAL DISTRICT OF CALIFORNIA

NOTICE OF ASSIGNMENT TO UNITED STATES MAGISTRATE JUDGE FOR DISCOVERY

This case has been assigned to District Judge Cormac J. Carney and the assigned discovery Magistrate Judge is _____.

The case number on all documents filed with the Court should read as follows:

SACV13- 829 CJC (MANx)

Pursuant to General Order 05-07 of the United States District Court for the Central District of California, the Magistrate Judge has been designated to hear discovery related motions.

All discovery related motions should be noticed on the calendar of the Magistrate Judge

NOTICE TO COUNSEL

A copy of this notice must be served with the summons and complaint on all defendants (if a removal action is filed, a copy of this notice must be served on all plaintiffs).

Subsequent documents must be filed at the following location:

[ ] **Western Division**
312 N. Spring St., Rm. G-8
Los Angeles, CA 90012

[X] **Southern Division**
411 West Fourth St., Rm. 1-053
Santa Ana, CA 92701-4516

[ ] **Eastern Division**
3470 Twelfth St., Rm. 134
Riverside, CA 92501

Failure to file at the proper location will result in your documents being returned to you.

AO 440 (Rev. 06/12) Summons in a Civil Action

UNITED STATES DISTRICT COURT

for the

Central District of California

BROADCOM CORPORATION

Plaintiff(s)

v.

NXP SEMICONDUCTORS N.V.,
NXP B.V., &
NXP SEMICONDUCTORS USA, INC.

Defendant(s)

Civil Action No. **SACV13-00829 CJC (MANx)**

SUMMONS IN A CIVIL ACTION

To: *(Defendant's name and address)*

A lawsuit has been filed against you.

Within 21 days after service of this summons on you (not counting the day you received it) — or 60 days if you are the United States or a United States agency, or an officer or employee of the United States described in Fed. R. Civ. P. 12 (a)(2) or (3) — you must serve on the plaintiff an answer to the attached complaint or a motion under Rule 12 of the Federal Rules of Civil Procedure. The answer or motion must be served on the plaintiff or plaintiff's attorney, whose name and address are:

Andrea Jeffries
WILMER CUTLER PICKERING HALE AND DORR LLP
350 S. Grand Ave., Suite 2100
Los Angeles, CA 90071
Telephone: +213 443 5300
Facsimile: +213 443 5400

If you fail to respond, judgment by default will be entered against you for the relief demanded in the complaint. You also must file your answer or motion with the court.

CLERK OF COURT

Date: MAY 29 2013

A. Gonzalez
Signature of Clerk or Deputy Clerk

SEAL OF THE U.S. DISTRICT COURT CENTRAL DIST. OF CALIFORNIA

1225

Civil Action No.

PROOF OF SERVICE

(This section should not be filed with the court unless required by Fed. R. Civ. P. 4 (l))

This summons for *(name of individual and title, if any)* _____

was received by me on *(date)* _____ .

❒ I personally served the summons on the individual at *(place)* _____

_____ on *(date)* _____ ; or

❒ I left the summons at the individual's residence or usual place of abode with *(name)* _____

_____ , a person of suitable age and discretion who resides there,

on *(date)* _____ , and mailed a copy to the individual's last known address; or

❒ I served the summons on *(name of individual)* _____ , who is

designated by law to accept service of process on behalf of *(name of organization)* _____

_____ on *(date)* _____ ; or

❒ I returned the summons unexecuted because _____ ; or

❒ Other *(specify):*

My fees are $ _____ for travel and $ _____ for services, for a total of $ 0.00 .

I declare under penalty of perjury that this information is true.

Date: _____

_____

Server's signature

_____

Printed name and title

_____

Server's address

Additional information regarding attempted service, etc:

CIVIL COVER SHEET

I. (a) PLAINTIFFS (Check box if you are representing yourself ☐)

Broadcom Corporation

DEFENDANTS (Check box if you are representing yourself ☐)

NXP SEMICONDUCTORS N.V.,
NXP B.V., &
NXP SEMICONDUCTORS USA, INC.

(b) Attorneys (Firm Name, Address and Telephone Number. If you are representing yourself, provide same.)

WILMER CUTLER PICKERING HALE AND DORR LLP
350 S. Grand Avenue
Suite 2100
Los Angeles, California 90071

(b) Attorneys (Firm Name, Address and Telephone Number. If you are representing yourself, provide same.)

II. BASIS OF JURISDICTION (Place an X in one box only.)

☐ 1. U.S. Government Plaintiff
☒ 3. Federal Question (U.S. Government Not a Party)
☐ 2. U.S. Government Defendant
☐ 4. Diversity (Indicate Citizenship of Parties in Item III)

III. CITIZENSHIP OF PRINCIPAL PARTIES-For Diversity Cases Only
(Place an X in one box for plaintiff and one for defendant)

	PTF	DEF		PTF	DEF
Citizen of This State	☐ 1	☐ 1	Incorporated or Principal Place of Business in this State	☐ 4	☐ 4
Citizen of Another State	☐ 2	☐ 2	Incorporated and Principal Place of Business in Another State	☐ 5	☐ 5
Citizen or Subject of a Foreign Country	☐ 3	☐ 3	Foreign Nation	☐ 6	☐ 6

IV. ORIGIN (Place an X in one box only.)

☒ 1. Original Proceeding ☐ 2. Removed from State Court ☐ 3. Remanded from Appellate Court ☐ 4. Reinstated or Reopened ☐ 5. Transferred from Another District (Specify) ☐ 6. Multi-District Litigation

V. REQUESTED IN COMPLAINT: JURY DEMAND: ☒ Yes ☐ No (Check "Yes" only if demanded in complaint.)

CLASS ACTION under F.R.Cv.P. 23: ☐ Yes ☒ No ☒ **MONEY DEMANDED IN COMPLAINT:** $ To be decided at trial

VI. CAUSE OF ACTION (Cite the U.S. Civil Statute under which you are filing and write a brief statement of cause. Do not cite jurisdictional statutes unless diversity.)

Patent Infringement under 35 U.S.C. § 271

VII. NATURE OF SUIT (Place an X in one box only).

OTHER STATUTES
☐ 375 False Claims Act
☐ 400 State Reapportionment
☐ 410 Antitrust
☐ 430 Banks and Banking
☐ 450 Commerce/ICC Rates/Etc.
☐ 460 Deportation
☐ 470 Racketeer Influenced & Corrupt Org.
☐ 480 Consumer Credit
☐ 490 Cable/Sat TV
☐ 850 Securities/Commodities/Exchange
☐ 890 Other Statutory Actions
☐ 891 Agricultural Acts
☐ 893 Environmental Matters
☐ 895 Freedom of Info. Act
☐ 896 Arbitration
☐ 899 Admin. Procedures Act/Review of Appeal of Agency Decision
☐ 950 Constitutionality of State Statutes

CONTRACT
☐ 110 Insurance
☐ 120 Marine
☐ 130 Miller Act
☐ 140 Negotiable Instrument
☐ 150 Recovery of Overpayment & Enforcement of Judgment
☐ 151 Medicare Act
☐ 152 Recovery of Defaulted Student Loan (Excl. Vet.)
☐ 153 Recovery of Overpayment of Vet. Benefits
☐ 160 Stockholders' Suits
☐ 190 Other Contract
☐ 195 Contract Product Liability
☐ 196 Franchise

REAL PROPERTY
☐ 210 Land Condemnation
☐ 220 Foreclosure
☐ 230 Rent Lease & Ejectment

REAL PROPERTY CONT.
☐ 240 Torts to Land
☐ 245 Tort Product Liability
☐ 290 All Other Real Property

TORTS
PERSONAL INJURY
☐ 310 Airplane
☐ 315 Airplane Product Liability
☐ 320 Assault, Libel & Slander
☐ 330 Fed. Employers' Liability
☐ 340 Marine
☐ 345 Marine Product Liability
☐ 350 Motor Vehicle
☐ 355 Motor Vehicle Product Liability
☐ 360 Other Personal Injury
☐ 362 Personal Injury-Med Malpratice
☐ 365 Personal Injury-Product Liability
☐ 367 Health Care/ Pharmaceutical Personal Injury Product Liability
☐ 368 Asbestos Personal Injury Product Liability

IMMIGRATION
☐ 462 Naturalization Application
☐ 465 Other Immigration Actions

TORTS
PERSONAL PROPERTY
☐ 370 Other Fraud
☐ 371 Truth in Lending
☐ 380 Other Personal Property Damage
☐ 385 Property Damage Product Liability

BANKRUPTCY
☐ 422 Appeal 28 USC 158
☐ 423 Withdrawal 28 USC 157

CIVIL RIGHTS
☐ 440 Other Civil Rights
☐ 441 Voting
☐ 442 Employment
☐ 443 Housing/ Accomodations
☐ 445 American with Disabilities-Employment
☐ 446 American with Disabilities-Other
☐ 448 Education

PRISONER PETITIONS
Habeas Corpus:
☐ 463 Alien Detainee
☐ 510 Motions to Vacate Sentence
☐ 530 General
☐ 535 Death Penalty
Other:
☐ 540 Mandamus/Other
☐ 550 Civil Rights
☐ 555 Prison Condition
☐ 560 Civil Detainee Conditions of Confinement

FORFEITURE/PENALTY
☐ 625 Drug Related Seizure of Property 21 USC 881
☐ 690 Other

LABOR
☐ 710 Fair Labor Standards Act
☐ 720 Labor/Mgmt. Relations
☐ 740 Railway Labor Act
☐ 751 Family and Medical Leave Act
☐ 790 Other Labor Litigation
☐ 791 Employee Ret. Inc. Security Act

PROPERTY RIGHTS
☐ 820 Copyrights
☒ 830 Patent
☐ 840 Trademark

SOCIAL SECURITY
☐ 861 HIA (1395ff)
☐ 862 Black Lung (923)
☐ 863 DIWC/DIWW (405 (g))
☐ 864 SSID Title XVI
☐ 865 RSI (405 (g))

FEDERAL TAX SUITS
☐ 870 Taxes (U.S. Plaintiff or Defendant)
☐ 871 IRS-Third Party 26 USC 7609

FOR OFFICE USE ONLY: Case Number: **SACV13-00829 CJC (MANx)**

AFTER COMPLETING PAGE 1 OF FORM CV-71, COMPLETE THE INFORMATION REQUESTED ON PAGE 2.

VIII(a). IDENTICAL CASES: Has this action been previously filed in this court and dismissed, remanded or closed? ☒ NO ☐ YES

If yes, list case number(s): _____

VIII(b). RELATED CASES: Have any cases been previously filed in this court that are related to the present case? ☒ NO ☐ YES

If yes, list case number(s): _____

Civil cases are deemed related if a previously filed case and the present case:

(Check all boxes that apply)
- ☐ A. Arise from the same or closely related transactions, happenings, or events; or
- ☐ B. Call for determination of the same or substantially related or similar questions of law and fact; or
- ☐ C. For other reasons would entail substantial duplication of labor if heard by different judges; or
- ☐ D. Involve the same patent, trademark or copyright, and one of the factors identified above in a, b or c also is present.

IX. VENUE: (When completing the following information, use an additional sheet if necessary.)

(a) List the County in this District; California County outside of this District; State if other than California; or Foreign Country, in which **EACH** named plaintiff resides.

☐ Check here if the government, its agencies or employees is a named plaintiff. If this box is checked, go to item (b).

County in this District:*	California County outside of this District; State, if other than California; or Foreign Country
Orange County	

(b) List the County in this District; California County outside of this District; State if other than California; or Foreign Country, in which **EACH** named defendant resides.

☐ Check here if the government, its agencies or employees is a named defendant. If this box is checked, go to item (c).

County in this District:*	California County outside of this District; State, if other than California; or Foreign Country
	NXP Semiconductors NV - The Netherlands NXP BV - The Netherlands NXP Semiconductors USA, Inc.- Santa Clara County, CA

(c) List the County in this District; California County outside of this District; State if other than California; or Foreign Country, in which **EACH** claim arose.
NOTE: In land condemnation cases, use the location of the tract of land involved.

County in this District:*	California County outside of this District; State, if other than California; or Foreign Country
Orange County	

***Los Angeles, Orange, San Bernardino, Riverside, Ventura, Santa Barbara, or San Luis Obispo Counties**
Note: In land condemnation cases, use the location of the tract of land involved

X. SIGNATURE OF ATTORNEY (OR SELF-REPRESENTED LITIGANT): _____ DATE: 5/29/2013

Notice to Counsel/Parties: The CV-71 (JS-44) Civil Cover Sheet and the information contained herein neither replace nor supplement the filing and service of pleadings or other papers as required by law. This form, approved by the Judicial Conference of the United States in September 1974, is required pursuant to Local Rule 3-1 is not filed but is used by the Clerk of the Court for the purpose of statistics, venue and initiating the civil docket sheet. (For more detailed instructions, see separate instructions sheet).

Key to Statistical codes relating to Social Security Cases:

Nature of Suit Code	Abbreviation	Substantive Statement of Cause of Action
861	HIA	All claims for health insurance benefits (Medicare) under Title 18, Part A, of the Social Security Act, as amended. Also, include claims by hospitals, skilled nursing facilities, etc., for certification as providers of services under the program. (42 U.S.C. 1935FF(b))
862	BL	All claims for "Black Lung" benefits under Title 4, Part B, of the Federal Coal Mine Health and Safety Act of 1969. (30 U.S.C. 923)
863	DIWC	All claims filed by insured workers for disability insurance benefits under Title 2 of the Social Security Act, as amended; plus all claims filed for child's insurance benefits based on disability. (42 U.S.C. 405 (g))
863	DIWW	All claims filed for widows or widowers insurance benefits based on disability under Title 2 of the Social Security Act, as amended. (42 U.S.C. 405 (g))
864	SSID	All claims for supplemental security income payments based upon disability filed under Title 16 of the Social Security Act, as amended.
865	RSI	All claims for retirement (old age) and survivors benefits under Title 2 of the Social Security Act, as amended. (42 U.S.C. 405 (g))