

1 David A. Nelson (*pro hac vice* forthcoming)
2 (Ill. Bar No. 6209623)
3 davenelson@quinnemanuel.com
4 QUINN EMANUEL URQUHART & SULLIVAN, LLP
5 500 West Madison St., Suite 2450
6 Chicago, Illinois 60661
7 Telephone: (312) 705-7400
8 Facsimile: (312) 705-7401

7 Karen P. Hewitt (SBN 145309)
8 kphewitt@jonesday.com
9 Randall E. Kay (SBN 149369)
10 rekay@jonesday.com
11 JONES DAY
12 4655 Executive Drive, Suite 1500
13 San Diego, California 92121
14 Telephone: (858) 314-1200
15 Facsimile: (858) 345-3178

14 Evan R. Chesler (*pro hac vice* forthcoming)
15 (N.Y. Bar No. 1475722)
16 echesler@cavath.com
17 CRAVATH, SWAINE & MOORE LLP
18 Worldwide Plaza, 825 Eighth Avenue
19 New York, NY 10019
20 Telephone: (212) 474-1000
21 Facsimile: (212) 474-3700

20 *Attorneys for Plaintiff*
21 QUALCOMM INCORPORATED

22 UNITED STATES DISTRICT COURT
23 SOUTHERN DISTRICT OF CALIFORNIA

24 QUALCOMM INCORPORATED,
25 Plaintiff,
26 v.
27 APPLE INCORPORATED,
28 Defendant.

Case No. '17CV1375 JAH AGS

**COMPLAINT FOR PATENT
INFRINGEMENT**

[DEMAND FOR A JURY TRIAL]

COMPLAINT

1 Plaintiff Qualcomm Incorporated (“Qualcomm”), by its undersigned
2 attorneys, alleges, with knowledge with respect to its own acts and on information
3 and belief as to other matters, as follows:

4 NATURE OF THE ACTION

5 1. Qualcomm brings this action to compel Apple to cease infringing
6 Qualcomm’s patents and to compensate Qualcomm for Apple’s extensive
7 infringement of several patented Qualcomm technologies.

8 2. Qualcomm is one of the world’s leading technology companies and a
9 pioneer in the mobile phone industry. Its inventions form the very core of modern
10 mobile communication and enable modern consumer experiences on mobile devices
11 and cellular networks.

12 3. Since its founding in 1985, Qualcomm has been designing, developing,
13 and improving mobile communication devices, systems, networks, and products. It
14 has invented technologies that transform how the world communicates. Qualcomm
15 developed fundamental technologies at the heart of 2G, 3G, and 4G cellular
16 communications, is leading the industry to 5G cellular communications, and has
17 developed numerous innovative features used in virtually every modern cell phone.

18 4. Qualcomm also invests in technologies developed by other companies
19 and has acquired companies (and their patented innovative technologies) as part of
20 its emphasis on supporting innovation. Qualcomm’s patent portfolio currently
21 includes more than 130,000 issued patents and patent applications worldwide.
22 Hundreds of mobile device suppliers around the world have taken licenses from
23 Qualcomm.

24 5. Apple is the world’s most profitable seller of mobile devices. Its
25 iPhones and other products enjoy enormous commercial success. But without the
26 innovative technology covered by Qualcomm’s patent portfolio, Apple’s products
27 would lose much of their consumer appeal. Apple was a relatively late entrant in the
28 mobile device industry and its mobile devices rely heavily on the inventions of

1 Qualcomm and other companies that Qualcomm has invested in. Nearly a decade
2 before Apple released the iPhone, Qualcomm unveiled its own full-feature, top-of-
3 the-line smartphone. According to CNN’s 1999 holiday buying guide, Qualcomm’s
4 pdQ 1900 “lets you make calls, keep records, send email, browse the web and run
5 over a thousand different applications, all while on the go. Although a cell phone, it
6 is one of the first truly portable, mobile and multipurpose Internet devices.”¹ While
7 Qualcomm no longer markets phones directly to consumers, it continues to lead the
8 development of cutting-edge technologies that underpin a wide range of important
9 wireless-device features. Other companies, like Apple, now manufacture and
10 market phones that feature Qualcomm’s innovations and the innovations of other
11 technology pioneers that Qualcomm invested in.

12 6. Qualcomm’s innovations in the mobile space have influenced all
13 modern smartphones, and Apple—like other major mobile device makers—utilizes
14 Qualcomm’s technologies. Qualcomm’s patented features enable and enhance
15 popular features that drive consumer demand, for example, enhancements to high
16 performance and power efficient graphics processing architectures, reducing power
17 consumption and increasing battery life through envelope tracking, reducing power
18 consumption and increasing efficiency of integrated circuits through voltage level
19 shifter improvements, “flashless boot” technology that reduces memory expenses
20 with little to no performance impact, reducing circuit power consumption by
21 minimizing power-intensive bus activation, and power-efficient carrier aggregation
22 allowing faster network performance and longer battery life, among many others.

23 7. In short, Qualcomm invented many core technologies that make the
24 iPhone (and other smartphones and mobile devices) desirable to consumers in their
25 daily lives.

26

27

28 ¹ <http://edition.cnn.com/1999/TECH/ptech/12/03/qualcomm.pdq/>

1 8. While Apple built the most successful consumer products in history by
2 relying significantly on technologies pioneered by Qualcomm, Apple refuses to pay
3 for those technologies. Apple’s founder boasted that Apple “steals” the great ideas
4 of others—specifically, that “we have always been shameless about stealing great
5 ideas.”² Apple employees likewise admit that Apple—a relatively late entrant in the
6 mobile space—did not invent many of the iPhone’s features. Instead, Apple
7 incorporated, marketed, and commercialized the work of others: “I don’t know how
8 many things we can come up with that you could legitimately claim we did first. . . .
9 We had the first commercially successful version of many features but that’s
10 different than launching something to market first.”³

11 9. Rather than pay Qualcomm for the technology Apple uses, Apple has
12 taken extraordinary measures to avoid paying Qualcomm for the fair value of
13 Qualcomm’s patents. On January 20, 2017, Apple sued Qualcomm in this district,
14 asserting an array of excuses to avoid paying fair-market, industry-standard rates for
15 the use of certain of Qualcomm’s pioneering patents that are critical to a modern
16 smartphone like the iPhone. *See* Case No. 3:17-cv-00108-GPC-MDD. Apple also
17 encouraged the companies that manufacture the iPhone to breach their contracts
18 with Qualcomm by refusing to pay for the Qualcomm technology in iPhones,
19 something that those manufacturers had done for many years, without complaint,
20 before Apple’s direction to stop. Further, Apple misled governmental agencies
21 around the world into investigating Qualcomm in an effort to indirectly exert

22
23 ² Interview with Steve Jobs, available at
24 <https://www.youtube.com/watch?v=CW0DUg63lqU> (“Picasso had a saying, ‘good
25 artists copy, great artists steal.’ And we have always been shameless about stealing
26 great ideas.”).

27 ³ April 2010 email from Apple’s iPhone Product Marketing Manager, Steve
28 Sinclair, reported in: Rick Merritt, *Schiller ‘shocked at ‘copycat’ Samsung phone*,
Embedded (Aug. 3, 2012), <http://www.embedded.com/print/4391702> (April 21,
2017 snapshot of page, accessed via Google’s cache).

1 leverage over Qualcomm.

2 10. Many of Qualcomm’s patents are essential to certain cellular or other
3 standards (“Standard Essential Patents,” or “SEPs”), such that the use of an
4 underlying technological standard would require use of the patent. Qualcomm owns
5 a wide range of non-standard-essential patents for inventions in various technologies
6 related to mobile devices.

7 11. In this suit, Qualcomm asserts a set of six non-standard-essential
8 patents infringed by Apple’s mobile electronic devices. The patents asserted in this
9 suit represent only a small fraction of the Qualcomm non-standard-essential patents
10 that Apple uses without a license.

11 12. Qualcomm repeatedly offered to license its patents to Apple. But
12 Apple has repeatedly refused offers to license Qualcomm’s patents on reasonable
13 terms. Qualcomm therefore seeks to enforce its rights in the patents identified
14 below and to address and remedy Apple’s flagrant infringement of those patents.

15 PARTIES

16 13. Qualcomm is a Delaware corporation with its principal place of
17 business at 5775 Morehouse Drive, San Diego, California. Since 1989, when
18 Qualcomm publicly introduced Code Division Multiple Access (“CDMA”) as a
19 commercially successful digital cellular communications standard, Qualcomm has
20 been recognized as an industry leader and innovator in the field of mobile devices
21 and cellular communications. Qualcomm owns more than 130,000 patents and
22 patent applications around the world relating to cellular technologies and many
23 other valuable technologies used by mobile devices. Qualcomm is a leader in the
24 development and commercialization of wireless technologies and the owner of the
25 world’s most significant portfolio of cellular technology patents. Qualcomm derives
26 a substantial portion of its revenues and profits from licensing its intellectual
27 property. Qualcomm is also a world leader in the sale of chips, chipsets, and
28 associated software for mobile phones and other wireless devices.

1 audio quality was poor, users sometimes heard portions of others calls, handoffs
2 were noisy, and calls frequently dropped. Qualcomm played a central role in the
3 revolutionary transformation of cellular communications technologies. Today,
4 cellular devices are remarkably powerful and can deliver reliable voice service and
5 lightning-fast data to billions of consumers around the world at affordable prices.

6 20. Qualcomm is now one of the largest technology, semiconductor, and
7 telecommunications companies in the United States. It employs over 18,000 people
8 in the United States, 68 percent of whom are engineers, and it occupies more than
9 92 buildings (totaling over 6.5 million sq. ft.) in seventeen states and the District of
10 Columbia.

11 21. Qualcomm's industry-leading research and development efforts,
12 focused on enabling cellular systems and products, are at the core of Qualcomm's
13 business. Since its founding, Qualcomm has invested tens of billions of dollars in
14 research and development related to cellular, wireless communications, and mobile
15 processor technology. Qualcomm's massive research and development investments
16 have produced numerous innovations. Because of this ongoing investment,
17 Qualcomm continues to drive the development and commercialization of successive
18 generations of mobile technology and is one of a handful of companies leading the
19 development of the next-generation 5G standard.

20 22. In addition to Qualcomm's investments in research and development
21 internally, Qualcomm has a rich history of investing in and acquiring technologies
22 developed by other industry leaders. By purchasing companies and patents from
23 companies who desire to sell their innovations, Qualcomm fosters innovation by
24 enabling those companies to realize a return on their research and development
25 investments and, therefore, incentivizes additional research and development.

26 23. As a result of the strength and value of Qualcomm's patent portfolio,
27 virtually every major handset manufacturer in the world has taken a royalty-bearing
28 license to Qualcomm's patent portfolio. The licenses to Qualcomm's patents allow

1 manufacturers to use numerous forms of critical and innovative Qualcomm
2 technology without having to bear the multi-billion dollar, multi-year costs of
3 developing those innovations themselves.

4 **Apple Background**

5 24. Apple has built the most profitable company in the world, thanks in
6 large part to products that rely on Qualcomm's patented technologies. With a
7 market capitalization of more than \$700 billion, \$246 billion in cash reserves, and a
8 global sphere of influence, Apple has more money and more influence than many
9 countries. Relying heavily on Qualcomm technology and technology Qualcomm
10 has acquired, Apple has become the dominant player in mobile device sales.
11 Apple's dominance has grown every year since the iPhone's launch in 2007. In
12 recent years, Apple has captured upwards of *90 percent of all profits* in the
13 smartphone industry.

14 **Qualcomm's Technology Leadership**

15 25. The asserted patents reflect the breadth of Qualcomm's dedication and
16 investment in research and development relating to wireless technology. Qualcomm
17 invented numerous proprietary solutions that are used to optimize products around
18 the globe. Many of those inventions are reflected in Qualcomm's non-standard-
19 essential patents (such as the patents asserted in this case).

20 26. As mobile electronic devices have become more powerful with greater
21 functionality, device manufacturers have faced numerous problems with power
22 consumption, noise reduction, battery charging, graphics processing and heat
23 dissipation, among others. The asserted patents disclose and claim Qualcomm
24 technology that solves many of these problems by enhancing chip performance
25 through advanced carrier aggregation, power-efficient envelope tracking, power-
26 efficient boot up and inter-chip communication techniques, area- and power-
27 efficient circuit designs, and more powerful and efficient graphics processing
28 circuitry and techniques.

1 27. For example, Qualcomm pioneered various “envelope tracking”
2 techniques for mobile devices to save power and reduce heat inside a mobile device
3 when transmitting at different signal strength. Using one of these techniques as set
4 forth in Qualcomm’s ‘558 patent, the radio frequency (RF) amplifier power supply
5 is continuously adjusted and dynamically boosted, as necessary, to ensure that the
6 amplifier is operating at peak efficiency for the power required during transmission.
7 Envelope tracking allows for a thinner, lighter mobile electronic device that
8 generates less heat. Without envelope tracking, power is wasted and battery life is
9 shorter.

10 28. As another example, Apple has touted the capability of its newest
11 mobile electronic devices to support “carrier aggregation” technology. This means
12 that a mobile device can simultaneously transmit radio signals for multiple carriers,
13 which again allows for a more efficient use of power and longer battery life.
14 Indeed, Apple’s Senior Vice President of Worldwide Marketing proclaimed that one
15 of the differentiating features of the iPhone 6 is that its enhanced speed is done
16 “with a technology called carrier aggregation.”⁴ Qualcomm has pioneered and
17 patented technologies that allow this carrier aggregation to be utilized more
18 efficiently and with less wasted power.

19 29. Qualcomm has also pioneered breakthrough mobile graphics
20 technologies. Each of Qualcomm's industry-leading Snapdragon processors employ
21 advanced Adreno series graphics processing units (“GPUs”). Mobile device GPUs
22 must be powerful enough to meet the increasing computational demands of mobile
23 operating systems without unnecessarily draining the battery life of the mobile
24 device. The Adreno GPU architecture is the “highest-performance GPU ever
25
26

27 ⁴ [https://singjupost.com/apple-iphone-6-keynote-september-2014-launch-event-
28 \[full-transcript/?singlepage=1\]\(https://singjupost.com/apple-iphone-6-keynote-september-2014-launch-event-full-transcript/?singlepage=1\)](https://singjupost.com/apple-iphone-6-keynote-september-2014-launch-event-full-transcript/?singlepage=1) (emphasis added).

1 designed by Qualcomm,” providing 40% lower power consumption with 40% faster
2 performance than the previous Adreno series.⁵

3 30. Qualcomm’s patented graphics processing architecture delivers
4 efficiency gains while simultaneously providing excellent performance—receiving
5 praise from Forbes for “dazzl[ing] in GPU performance.”⁶ Apple heavily markets
6 its GPU performance, claiming that the GPU in the iPhone 7 would “deliver 50%
7 more graphics performance than the [the previous version].”⁷ Apple has chosen to
8 use Qualcomm’s patented improvements to graphics processing architecture,
9 including U.S. Patent No. 8,633,936, without paying for them, to deliver high speed
10 and power-efficient graphics that Apple promises and its customers now demand.

11 31. Qualcomm has invested substantially in both advancing standards and
12 the quality of service provided to anyone using that standard, as well as proprietary
13 implementations of modem technology (*e.g.*, innovations in lowering costs and
14 driving efficiency). For example, Qualcomm is a pioneer in “flashless boot”
15 technology, to which the ’949 patent relates, which enables phone manufacturers to
16 use less storage in relation to wireless modems. Storage in a handheld device can be
17 both expensive and contribute to weight and size concerns. Indeed, incremental
18 increases in flash storage for currently available iPhones can cost \$100,⁸ and Apple
19 has also touted its iPhone devices as thin and light.

20 32. Similarly, Qualcomm has invested and developed novel solutions to
21 save power in mobile devices, including by synchronizing messages within a mobile
22

23 ⁵ [https://www.theinquirer.net/inquirer/news/2421804/qualcomm-outs-next-gen-
24 adreno-gpu-ahead-of-snapdragon-820-launch](https://www.theinquirer.net/inquirer/news/2421804/qualcomm-outs-next-gen-adreno-gpu-ahead-of-snapdragon-820-launch).

25 ⁶ [https://www.forbes.com/sites/moorinsights/2017/03/23/qualcomms-new-
26 snapdragon-835-dazzles-in-gpu-performance/#6b7099262075](https://www.forbes.com/sites/moorinsights/2017/03/23/qualcomms-new-snapdragon-835-dazzles-in-gpu-performance/#6b7099262075).

27 ⁷ [https://www.extremetech.com/computing/235140-apples-new-a10-fusion-
28 quad-core-high-efficiency-and-a-more-powerful-gpu](https://www.extremetech.com/computing/235140-apples-new-a10-fusion-quad-core-high-efficiency-and-a-more-powerful-gpu).

⁸ <https://www.apple.com/shop/buy-iphone/iphone-7>.

1 device to reduce both time and power needed for inter-processor communication.
2 The '490 patent allows mobile devices to operate just as efficiently with lower
3 power consumption, which in turn prolongs the battery life of those devices. Apple
4 benefits from this technology, simultaneously touting the long battery life of its
5 devices while relying upon a relatively small battery as compared to the iPhone's
6 peers.

7 33. As a final example, Qualcomm's development of power efficient
8 innovations is unmatched. Qualcomm has invented circuit-level solutions that allow
9 devices to reduce the overall operating voltages of the integrated circuits within the
10 phone, thereby reducing power consumption and increasing efficiency of the
11 integrated circuit. A particularly important circuit in Apple devices is therefore one
12 of the inventions in Qualcomm's '658 patent, relating to a voltage level shifter.
13 Qualcomm's solution significantly reduces the size of level shifting circuits. The
14 compact design claimed in the '658 patent allows Apple to fit more circuits into a
15 layout, thereby saving costs through smaller die sizes and reduced power
16 consumption, which has all the benefits that Apple regularly touts.

17 **The Accused Devices**

18 34. As set forth below, a variety of Apple's devices—including certain of
19 Apple's iPhones and iPads —practice one or more of the Patents-in-Suit.

20 **The Patents-in-Suit**

21 35. The following patents are infringed by Apple ("Patents-in-Suit"): U.S.
22 Patent No. 8,633,936 ("the '936 patent"), U.S. Patent No. 8,698,558 ("the '558
23 patent"), U.S. Patent No. 8,487,658 ("the '658 patent"), U.S. Patent No. 8,838,949
24 ("the '949 patent"), U.S. Patent No. 9,535,490 ("the '490 patent"), and U.S. Patent
25 No. 9,608,675 ("the '675 patent").

26 36. As described below, Apple has been and is still infringing, contributing
27 to infringement, and/or inducing others to infringe the Patents-in-Suit by making,
28 using, offering for sale, selling, or importing devices that practice the Patents-in-

1 Suit. Apple's acts of infringement have occurred within this District and elsewhere
2 throughout the United States.

3 **U.S. Patent No. 8,633,936**

4 37. The '936 patent was duly and legally issued on January 21, 2014 to
5 Qualcomm, which is the owner of the '936 patent and has the full and exclusive
6 right to bring actions and recover damages for Apple's infringement of the '936
7 patent. The '936 patent is valid and enforceable. A copy of the '936 Patent is
8 attached hereto as Exhibit A.

9 38. The '936 patent relates generally to a graphics processing architecture.
10 The '936 patent discloses novel methods and structures for forming graphics
11 processing circuitry incorporating multiple execution units for processing graphics
12 instructions at different graphics precision levels, and for converting graphics data to
13 the correct precision level prior to processing the associated graphics instruction.
14 As a result of the invention of the '936 patent, graphics processors are able to use
15 lower precision execution units, processing graphics data in a higher performance
16 and more power efficient manner, thereby extending battery life.

17 **U.S. Patent No. 8,698,558**

18 39. The '558 patent was duly and legally issued on April 15, 2014, and
19 Qualcomm is the current owner of the '558 patent and has the full and exclusive
20 right to bring action and recover damages for Apple's infringement of the '558
21 patent. The '558 patent is valid and enforceable. A copy of the '558 Patent is
22 attached hereto as Exhibit B.

23 40. The '558 patent relates generally to envelope tracking technology,
24 which addresses the efficient use of power by a power amplifier in transmitting an
25 output radio frequency (RF) signal. In particular, the power amplifier may require
26 varying degrees of power supply voltage depending on the type of RF signal being
27 transmitted. In the past, the use of a constant power supply voltage did not match
28 the varying power requirements of the power amplifier, and led to unnecessary

1 dissipation of power (and devices that, due to this unnecessary power dissipation,
2 quickly drained the battery). Envelope tracking adjusts the power supply voltage
3 based on information from the modem to match the needs of the power amplifier.
4 The '558 patent discloses novel circuitry for efficiently and effectively boosting
5 power supply voltage to continuously match the peak efficiency necessary over the
6 RF envelope. As a result of the invention of the '558 patent, electronic devices are
7 able to reduce power consumption and extend battery life.

8 **U.S. Patent No. 8,487,658**

9 41. The '658 patent was duly and legally issued on July 16, 2013 to
10 Qualcomm, which is the owner of the '658 patent and has the full and exclusive
11 right to bring action and recover damages for Apple's infringement of the '658
12 patent. The '658 patent is valid and enforceable. A copy of the '658 Patent is
13 attached hereto as Exhibit C.

14 42. The '658 patent relates generally to voltage level shifter circuitry.
15 Integrated circuit devices incorporating different types of functional circuitry are
16 often required to handle multiple voltage levels. These devices typically contain a
17 high-voltage circuit driven by a relatively high voltage power supply and a low-
18 voltage circuit driven by a relatively low-voltage power supply. Reducing the
19 overall operating voltages of the integrated circuit reduces power consumption and
20 increases efficiency of the integrated circuit. However, some circuits are more
21 amenable to lower operating voltages while others must operate at a higher voltage.
22 For circuits operating at two different voltages to communicate with each other, a
23 level shifter circuit is required as an interface to shift the signal from one voltage
24 level to another to avoid circuit dysfunction. However, because the level shifter
25 itself operates with two different voltages, it is required to have at least two N-wells,
26 one for each voltage. In addition, constraints placed on the N-wells may require
27 them to be separated by a minimum distance. Therefore, incorporating multiple
28 level shifters into a single chip can consume a significant portion of the available

1 chip area. The '658 patent is directed to a compact and robust multi-bit voltage
2 level shifter design and layout, which may reduce the area of the level shifters.

3 **U.S. Patent No. 8,838,949**

4 43. The '949 patent was duly and legally issued on September 16, 2014 to
5 Qualcomm, which is the owner of the '949 patent and has the full and exclusive
6 right to bring actions and recover damages for Apple's infringement of the '949
7 patent. The '949 patent is valid and enforceable. A copy of the '949 Patent is
8 attached hereto as Exhibit D.

9 44. The '949 patent relates generally to "flashless boot," *i.e.*, booting up a
10 secondary processor that does not have its own non-volatile memory to store the
11 system image. The '949 patent discloses novel techniques for implementing
12 flashless boot for secondary processors in multi-processor systems by using a scatter
13 loader to directly transfer the image into memory of the secondary processor. As a
14 result of the invention of the '949 patent, multi-processor systems—which
15 encompass a device including at least an application processor and a modem
16 processor—can avoid requiring a non-volatile memory for each processor with
17 minimal negative performance impact.

18 **U.S. Patent No. 9,535,490**

19 45. The '490 patent was duly and legally issued on January 3, 2017 and
20 Qualcomm is the current owner of the '490 patent and has the full and exclusive
21 right to bring action and recover damages for Apple's infringement of the '490
22 patent. The '490 patent is valid and enforceable. A copy of the '490 Patent is
23 attached hereto as Exhibit E.

24 46. The '490 patent relates generally to reducing power consumption in
25 electronic devices. The '490 patent discloses novel techniques for controlling power
26 consumption by disclosing methods to minimize the time during which system
27 buses are in a high-power consumption state. As a result of the invention of the
28 '490 patent, computing devices can operate just as efficiently with lower power

1 consumption, which in turn prolongs the battery life of those devices.

2 **U.S. Patent No. 9,608,675**

3 47. The '675 patent was duly and legally issued on February 11, 2013 to
4 Qualcomm, which is the owner of the '675 patent and has the full and exclusive
5 right to bring action and recover damages for Apple's infringement of the '675
6 patent. The '675 patent is valid and enforceable. A copy of the '675 Patent is
7 attached hereto as Exhibit F.

8 48. The '675 patent relates generally to techniques for generating a power
9 supply voltage for a power amplifier that processes multiple transmit signals sent
10 simultaneously, such as multiple transmissions sent simultaneously on multiple
11 carriers at different frequencies. As one example, the '675 patent discloses a power
12 tracker that generates a single power tracking signal based on inputs from a plurality
13 of carrier aggregated transmit signals; a power supply generator for generating a
14 single power supply voltage based on the power tracking signal; and a power
15 amplifier that receives the single power supply voltage and the plurality of carrier
16 aggregated transmit signals to produce a single output RF signal. As one result of
17 the invention of the '675 patent, electronic devices can more efficiently support and
18 perform carrier aggregation.

19 COUNT 1 (PATENT INFRINGEMENT – U.S. PATENT NO. 8,633,936)

20 49. Qualcomm repeats and re-alleges the allegations of paragraphs 1
21 through 48 above as if fully set forth herein.

22 50. Qualcomm is the lawful owner of the '936 patent, and has the full and
23 exclusive right to bring actions and recover damages for Apple's infringement of
24 said patent.

25 51. In violation of 35 U.S.C. § 271, Apple has been and is still
26 infringing, contributing to infringement, and/or inducing others to infringe the '936
27 patent by making, using, offering for sale, selling, or importing iPhone 7 and iPhone
28 7 Plus devices.

1 52. The accused devices contain a GPU, which is a single-chip,
2 programmable streaming processor. The GPU receives graphic instructions,
3 including an indication of the data precision, and conversion instruction to convert
4 the graphics data to the indicated data precision. Both the graphics and conversion
5 instructions are generated by a compiler. One of many execution units in the GPU
6 is selected due to the indicated data precision and executes the graphic and
7 conversion instructions. As a result, the accused devices are able to extend its
8 battery by using lower precision execution units and processing graphics data in a
9 more power efficient manner.

10 53. The accused devices infringe at least claims 1, 10, 11-18, 19, 20-27, 29,
11 38, 49, 55, 56-60, 67, and 68 of the '936 patent.

12 54. For example, with respect to claims 1, 10, 19, 29, and 38, the accused
13 devices incorporate an Apple A10 GPU, which is a version of the PowerVR
14 GT7600 that is part of the PowerVR Series 7XT GPU line. The Apple A10 GPU
15 receives graphics instructions and executes them within a programmable streaming
16 processor. The indication of the data precision is contained in the graphic
17 instructions, which is generated by the graphics driver's runtime compiler that
18 compiles graphics application instructions. On information and belief, Apple
19 designs its own custom shader compiler and driver. The Apple A10 GPU receives
20 conversion instructions, such as "pck" and "unpck" instructions, that are different
21 than the graphics instruction but also generated by the runtime compiler. The
22 conversion instructions are executed by the Apple A10 GPU in order to convert the
23 graphics data associated with the graphics instruction into the indicated data
24 precision. Execution units, such as ALUs, within the processor are selected based
25 on the precision required and used to convert the graphics data to the indicated data
26 precision before processing the graphics instruction.

27 55. With respect to claim 11, the accused devices contain further
28 instructions for the Apple A10 GPU to receive the graphics data associated with

1 graphics instructions, and generate and output a computation result with the
2 indicated data precision.

3 56. With respect to claim 12, the accused devices contain further
4 instructions for selecting an execution unit from a first set of execution units when
5 the indicated data precision is the first data precision and for selecting an execution
6 unit from a second set of execution units when the indicated data precision is the
7 second data precision, which is different from the first data precision.

8 57. With respect to claim 13, the two data precisions are different such that
9 the first data precision is a full data precision while the second data precision is a
10 half data precision.

11 58. With respect to claims 14 and 20 the accused devices contain
12 instructions for different sets of execution units to execute instructions with
13 corresponding data precision using the graphics data.

14 59. With respect to claim 15, the accused devices contain further
15 instructions to select and use an appropriate execution unit based on the indicated
16 data precision to execute the graphics instructions.

17 60. With respect to claim 16, the accused devices contain further
18 instructions to receive a second, different graphics and conversion instructions to be
19 executed by an execution unit in a second set of execution units with the indicated
20 second data precision.

21 61. With respect to claim 17, there are instructions programmed to cause
22 the GPU to decode the graphics instruction in order to determine the indicated data
23 precision.

24 62. With respect to claim 18, the graphics data associated with the graphics
25 instructions includes at least either vertex graphics data or pixel graphics data.

26 63. With respect to claim 21, the accused devices contain several execution
27 units, like ALUs, including at least one full precision execution unit and at least four
28 half-precision execution units.

1 64. With respect to claim 22, the controller in the accused devices is
2 configured to select full-precision execution units when full precision for the
3 graphics data is required.

4 65. With respect to claim 23, the controller in the accused devices is
5 configured to select half-precision execution units when half precision for the
6 graphics data is required.

7 66. With respect to claim 24, the accused devices contain at least one full-
8 precision register bank and four half-precision register banks to store the respective
9 computation results when the instructions are executed.

10 67. With respect to claim 25, the accused devices contain at least one full-
11 precision execution unit and one half-precision execution unit, where, on
12 information and belief, the full-precision execution unit is shut down when the
13 indicated data precision is half-precision and the half-precision execution unit will
14 execute the graphics instruction using the graphics data.

15 68. With respect to claim 26, the processor in the accused devices contains
16 a shader processor. On information and belief, Apple designs its own custom shader
17 compiler and driver.

18 69. With respect to claim 27, the accused devices are wireless
19 communication device handsets.

20 70. With respect to claims 49 and 55, the accused devices will use a
21 compiler executed by a GPU to analyze several application instructions for a
22 graphics application. On information and belief, Apple designs its own custom
23 shader compiler and driver. Each application instruction specifying a first data
24 precision level comprising a full data precision level will cause the compiler to
25 generate corresponding compiled instructions each indicating the full data precision
26 level for execution. Conversion instructions are also generated by the compiler to
27 convert the graphics data from a second, different data precision level to the first
28 data precision level when the compiled instructions are executed.

1 71. With respect to claim 56, the second data precision level is a half data
2 precision level.

3 72. With respect to claim 57, the accused devices contain instructions
4 where compiled instructions indicating a full data precision level is generated when
5 a corresponding application instruction specifies the full data precision level for its
6 execution.

7 73. With respect to claim 58, the accused devices contain instructions
8 where compiled instructions indicating a half data precision level is generated when
9 a corresponding application instruction specifies the half data precision level for its
10 execution.

11 74. With respect to claim 59, the accused devices contain instructions
12 where the compiler will generate compiled instructions with predefined field to
13 include information regarding the first data precision level when the corresponding
14 application instruction specifies the first data precision level for its execution.

15 75. With respect to claim 60, the accused devices contain instructions to
16 cause the GPU to store the generated compiled instructions in memory for
17 subsequent execution.

18 76. With respect to claim 67, the accused devices contain executable
19 instructions that are generated by a compiler and can support at least one function of
20 a graphics application. Each executable instruction indicates the first data precision
21 level for its execution. The second data precision level is included in each second
22 executable instruction and is different from the first precision level, which
23 comprises a full data precision level. Each third executable instructions support at
24 least a function of the graphics application and converts the graphics data from the
25 second data precision level to the first data precision level when the first executable
26 instructions are executed.

27 77. With respect to claim 68, the second data precision level is claim 67 is
28 half data precision level.

1 78. On information and belief, Apple also knowingly induces and/or
2 contributes to the infringement of at least claims 1 and 49 of the '936 patent by
3 others. On information and belief, Apple has had knowledge of the '936 patent, and
4 its infringement of the '936 patent, at least since the time this lawsuit was filed. On
5 information and belief, Apple tests, demonstrates, or otherwise operates the accused
6 devices in the United States, thereby performing the claimed methods and directly
7 infringing any asserted claims of the '936 patent requiring such operation.
8 Similarly, Apple's customers and the end users of the accused devices test and/or
9 operate the accused devices in the United States in accordance with Apple's
10 instructions contained in, for example, its user manuals, thereby also performing the
11 claimed methods and directly infringing the asserted claims of the Asserted Patents
12 requiring such operation.

13 79. Apple also contributes to infringement of the '936 patent by selling for
14 importation into the United States, importing into the United States, and/or selling
15 within the United States after importation the accused devices and the non-staple
16 constituent parts of those devices, which are not suitable for substantial non-
17 infringing use and which embody a material part of the invention described in the
18 '936 patent. These mobile electronic devices are known by Apple to be especially
19 made or especially adapted for use in the infringement of the '936 patent. Apple
20 also contributes to the infringement of the '936 patent by selling for importation into
21 the United States, importing into the United States, and/or selling within the United
22 States after importation components, such as the chipsets or software containing the
23 infringing functionality, of the accused devices, which are not suitable for
24 substantial non-infringing use and which embody a material part of the invention
25 described in the '936 patent. These mobile devices are known by Apple to be
26 especially made or especially adapted for use in the infringement of the '936 patent.
27 Specifically, on information and belief, Apple sells the accused devices to resellers,
28 retailers, and end users with knowledge that the devices are used for infringement.

1 End users of those mobile electronic devices directly infringe the '936 patent.

2 80. Apple's acts of infringement have occurred within this district and
3 elsewhere throughout the United States.

4 81. Qualcomm has been damaged and will suffer additional damages and
5 irreparable harm unless Apple is enjoined from further infringement. Qualcomm
6 will prove its irreparable harm and damages at trial.

7 COUNT 2 (PATENT INFRINGEMENT – U.S. PATENT NO. 8,698,558)

8 82. Qualcomm repeats and re-alleges the allegations of paragraphs 1
9 through 81 above as if fully set forth herein.

10 83. Qualcomm is the lawful owner of the '558 patent and has the full and
11 exclusive right to bring actions and recover damages for Apple's infringement of
12 said patent.

13 84. In violation of 35 U.S.C. § 271, Apple has been and is still infringing,
14 contributing to infringement, and/or inducing others to infringe the '558 patent by
15 making, using, offering for sale, selling, or importing devices that practice the
16 patent, such as mobile devices including but not limited to iPhone 7 and iPhone 7
17 Plus devices.

18 85. The accused devices infringe at least claims 1 and 6-20 of the '558
19 patent. For example, with respect to claim 1, the accused devices incorporate a
20 Qorvo 81003M Envelope Tracker Modulator, which includes a boost converter.
21 The boost converter receives a supply voltage and generates a signal with an
22 increased voltage. The devices further include an envelope amplifier that receives
23 an envelope signal and a boosted supply voltage. The envelope amplifier generates
24 a second supply voltage based on the envelope signal and the boosted supply
25 voltage. The envelope amplifier receives a supply voltage and generates a second
26 supply voltage based on the first supply voltage or the boosted supply voltage. The
27 envelope amplifier includes an operational amplifier that receives an envelope signal
28 and amplifies a signal, a driver that receives an amplified signal and provides

1 multiple control signals, a PMOS transistor with a gate receiving a control signal, a
2 source receiving a boosted supply voltage or a supply voltage and a drain providing
3 a supply voltage, and a NMOS transistor having a gate receiving a control signal, a
4 drain providing a supply voltage, and a source coupled to a ground.

5 86. With respect to claim 6, a power amplifier receives and amplifies an
6 input radio frequency signal to provide an amplified output signal. The device also
7 includes a supply generator for receiving an envelope signal and supply voltage and
8 generating a boosted supply voltage. The supply generator incorporates an
9 operational amplifier to receive the envelope signal and provide an amplified signal,
10 a driver that receives the amplified signal and provides a first control signal and a
11 second control signal, a P-channel metal oxide semiconductor (PMOS) transistor,
12 which has a gate receiving a first control signal, a source receiving the boosted
13 supply voltage or the first supply voltage and a drain providing the second supply
14 voltage, and an N-channel metal oxide semiconductor (NMOS) transistor having a
15 gate receiving the second control signal, a drain providing the second supply
16 voltage, and a source coupled to circuit ground.

17 87. With respect to claim 7, the accused devices include a supply generator
18 that generates the second supply voltage based on the envelope signal and either the
19 boosted supply voltage or the first supply voltage.

20 88. With respect to claim 8, the accused devices generate a boosted supply
21 voltage based on a first supply voltage, where the boosted supply voltage has a
22 higher voltage than the first supply voltage. Further, the devices generate a second
23 supply voltage based on an envelope signal and the boosted supply voltage. The
24 devices have a second supply voltage that is generated by an envelope amplifier that
25 produces the second supply voltage using an operational amplifier (op-amp) that
26 receives the envelope signal and provides an amplified signal, a driver that receives
27 the amplified signal and provides a first control signal and a second control signal, a
28 P-channel metal oxide semiconductor (PMOS) transistor that receives the first

1 control signal, a source that receives the boosted supply voltage or the first supply
2 voltage, and a drain providing the second supply voltage and an N-channel metal
3 oxide semiconductor (NMOS) transistor that receives the second control signal at a
4 gate and provides a second supply voltage through a drain, and a source for circuit
5 grounding.

6 89. With respect to claim 9, the accused devices generate the second supply
7 voltage based on the envelope signal and either the boosted supply voltage or the
8 first supply voltage.

9 90. With respect to claim 10, the accused devices are used for generating a
10 boosted supply voltage based on a first supply voltage, the boosted supply voltage
11 having a higher voltage than the first supply voltage. The accused devices are also
12 used for generating a second supply voltage based on the envelope signal and the
13 boosted supply voltage. The accused devices are also used for generating the
14 second supply voltage incorporating an envelope amplifier that produces the second
15 supply voltage using an operational amplifier that receives the envelope signal and
16 provides an amplified signal, a driver that receives the amplified signal and provides
17 a first control signal and a second control signal, a P-channel metal oxide
18 semiconductor (PMOS) transistor that receives the first control signal, a source that
19 receives the boosted supply voltage or the first supply voltage, and a drain providing
20 the second supply voltage and an N-channel metal oxide semiconductor (NMOS)
21 transistor that receives the second control signal at a gate and provides a second
22 supply voltage through a drain, and a source for circuit grounding.

23 91. With respect to claim 11, the accused devices are used for generating
24 the second supply voltage based on an envelope signal and either the boosted supply
25 voltage or the first supply voltage.

26 92. With respect to claim 12, the accused devices include a switcher
27 operative to receive a first supply voltage and provide a first supply current and an
28 envelope amplifier operative to receive an envelope signal and provide a second

1 supply current based on the envelope signal. The power amplifier is operative to
2 receive a total supply current comprising the first supply current and the second
3 supply current, wherein the switcher comprises a current sense amplifier operative
4 to sense the first supply current, or the second supply current, or the total supply
5 current and provide a sensed signal. The accused devices also include a driver
6 operative to receive the sensed signal and provide a first control signal and a second
7 control signal; a P-channel metal oxide semiconductor (PMOS) transistor having a
8 gate receiving the first control signal, a source receiving the first supply voltage, and
9 a drain providing a switching signal for an inductor providing the first supply
10 current; and an N-channel metal oxide semiconductor (NMOS) transistor having a
11 gate receiving the second control signal, a drain providing the switching signal, and
12 a source coupled to circuit ground.

13 93. With respect to claim 13, the accused devices include a boost converter
14 operative to receive the first supply voltage and provide a boosted supply voltage
15 having a higher voltage than the first supply voltage, wherein the envelope amplifier
16 operates based on the first supply voltage or the boosted supply voltage.

17 94. With respect to claim 14, the accused devices include a first supply
18 current that comprises direct current (DC) and low frequency components, and the
19 second supply current comprises higher frequency components.

20 95. With respect to claim 15, the accused devices include an inductor
21 operative to receive a switching signal and provide a supply current. The accused
22 devices also include a switcher operative to sense an input current and generate the
23 switching signal to charge and discharge the inductor to provide the supply current.
24 The switcher adds an offset to the input current to generate a larger supply current
25 via the inductor than without the offset. The switcher in the accused devices
26 includes a summer that sums the input current and an offset current and provide a
27 summed current. The accused devices include a current sense amplifier operative to
28 receive the summed current and provide a sensed signal. The accused devices also

1 include a driver operative to receive the sensed signal and provide at least one
2 control signal used to generate the switching signal for the inductor.

3 96. With respect to claim 16, in the accused devices the switcher operates
4 based on a first supply voltage, where an offset is determined based on the first
5 supply voltage.

6 97. With respect to claim 17, the accused devices contain a first control
7 signal and a second control signal, where the switcher includes a P-channel metal
8 oxide semiconductor (PMOS) transistor having a gate receiving the first control
9 signal, a source receiving a first supply voltage, and a drain providing the switching
10 signal, and an N-channel metal oxide semiconductor (NMOS) transistor having a
11 gate receiving the second control signal, a drain providing the switching signal, and
12 a source coupled to circuit ground.

13 98. With respect to claim 18, the accused devices include an envelope
14 amplifier that receives an envelope signal and provides a second supply current
15 based on the envelope signal, where a total supply current comprises the supply
16 current from the switcher and the second supply current from the envelope
17 amplifier.

18 99. With respect to claim 19, the accused devices include a boost converter
19 operative to receive the first supply voltage and provide a boosted supply voltage
20 having a higher voltage than the first supply voltage, where the envelope amplifier
21 operates based on the first supply voltage or the boosted supply voltage.

22 100. With respect to claim 20, the accused devices include a power amplifier
23 so that the accused devices can receive the supply current from the inductor and
24 receive and amplify an input radio frequency (RF) signal and provide an output RF
25 signal.

26 101. On information and belief, Apple also knowingly induces and/or
27 contributes to the infringement of at least claims 8-9 of the '558 patent by others.
28 On information and belief, Apple has had knowledge of the '558 patent, and its

1 infringement of the '558 patent, at least since the time this lawsuit was filed.
2 Additionally, Qualcomm has provided technical assistance and solutions to Apple,
3 including envelope tracking technology, under non-disclosure agreements. Apple
4 was aware of, and implemented, Qualcomm's technology in certain of its devices
5 without authorization. On information and belief, Apple tests, demonstrates, or
6 otherwise operates the accused devices in the United States, thereby performing the
7 claimed methods and directly infringing any asserted claims of the '558 patent
8 requiring such operation. Similarly, Apple's customers and the end users of the
9 accused devices test and/or operate the accused devices in the United States in
10 accordance with Apple's instructions contained in, for example, its user manuals,
11 thereby also performing the claimed methods and directly infringing the asserted
12 claims of the Asserted Patents requiring such operation.

13 102. Apple also contributes to infringement of the '558 patent by selling for
14 importation into the United States, importing into the United States, and/or selling
15 within the United States after importation the accused devices and the non-staple
16 constituent parts of those devices, which are not suitable for substantial non-
17 infringing use and which embody a material part of the invention described in the
18 '558 patent. These mobile electronic devices are known by Apple to be especially
19 made or especially adapted for use in the infringement of the '558 patent. Apple
20 also contributes to the infringement of the '558 patent by selling for importation into
21 the United States, importing into the United States, and/or selling within the United
22 States after importation components, such as the chipsets or software containing the
23 infringing functionality, of the accused devices, which are not suitable for
24 substantial non-infringing use and which embody a material part of the invention
25 described in the '558 patent. These mobile devices are known by Apple to be
26 especially made or especially adapted for use in the infringement of the '558 patent.
27 Specifically, on information and belief, Apple sells the accused devices to resellers,
28 retailers, and end users with knowledge that the devices are used for infringement.

1 End users of those mobile electronic devices directly infringe the '558 patent.

2 103. Apple's acts of infringement have occurred within this district and
3 elsewhere throughout the United States.

4 104. Qualcomm has been damaged and will suffer additional damages and
5 irreparable harm unless Apple is enjoined from further infringement. Qualcomm
6 will prove its irreparable harm and damages at trial.

7 COUNT 3 (PATENT INFRINGEMENT – U.S. PATENT NO. 8,487,658)

8 105. Qualcomm repeats and re-alleges the allegations of paragraphs 1
9 through 104 above as if fully set forth herein.

10 106. Qualcomm is the lawful owner of the '658 patent and has the full and
11 exclusive right to bring actions and recover damages for Apple's infringement of
12 said patent.

13 107. In violation of 35 U.S.C. § 271, Apple has been and is still infringing,
14 contributing to infringement, and/or inducing others to infringe the '658 patent by
15 making, using, offering for sale, selling, or importing mobile devices that practice
16 the patent, including but not limited to iPhone 7 and iPhone 7 Plus devices.

17 108. The accused devices use four-bit voltage level shifters arranged in a
18 certain layout to reduce the area of the four-bit voltage level shifters. Each bit is
19 shifted from a first voltage level logic to a second voltage level logic by forming
20 four one-bit voltage level shifter circuit, where each one-bit voltage level shifter
21 circuit is formed over two N-wells.

22 109. The accused devices infringe at least claims 9, 12, 14, 20, 21 and 22 of
23 the '658 patent.

24 110. For example, with respect to claim 9, the accused devices each include
25 a Qorvo ET modulator, which includes four voltage level shift (VLS) circuits. Each
26 VLS circuit shifts a bit from a first voltage level logic to a second voltage level
27 logic. The Qorvo ET modulator includes 3 N-wells formed in the substrate, with the
28 second and third N-well adjacent to the first N-well but also opposite from each

1 other. The four-bit multi-voltage circuit of the Qorvo ET modulator includes four
2 one-bit VLS circuit, each of which is formed on two N-wells.

3 111. With respect to claim 12, the accused devices have the first, second and
4 third N-wells arranged in a row with the first N-well at a center position.

5 112. With respect to claim 14, the accused devices have the first N-well
6 biased at the first voltage level, and the second and third N-wells biased at the
7 second voltage level.

8 113. With respect to claim 20 of the '658 patent, the accused devices
9 perform the method of reducing die area and switching power in the four-bit multi-
10 voltage circuit in the Qorvo ET modulator by shifting each of the four bits from a
11 first voltage level logic to a second voltage level logic. To achieve this, three N-
12 wells are formed where the second and third N-well are each adjacent to the first N-
13 well and opposite from each other. Then, four one-bit VLS circuits are formed,
14 each covering a portion of two N-wells.

15 114. With respect to claim 21 of the '658 patent, the accused devices contain
16 a Qorvo ET modulator, which reduces the die area and switching power in the four-
17 bit multi-voltage circuit to shift each of the four bits from the first voltage level
18 logic to the second voltage level logic by forming three N-wells in the substrate,
19 where the second and third N-wells are each adjacent to the first N-well and
20 opposite of each other. Logic is configured to form four one-bit VLS circuit, each
21 of which covers a portion of two N-wells.

22 115. With respect to claim 22 of the '658 patent, the accused devices contain
23 a Qorvo ET modulator, which reduces the die area and switching power in the four-
24 bit multi-voltage circuit to shift each of the four bits from the first voltage level
25 logic to the second voltage level logic by forming three N-wells in the substrate,
26 where the second and third N-wells are each adjacent to the first N-well and
27 opposite of each other. The Qorvo ET modulator also has means for forming four
28 one-bit VLS circuit, each of which covers a portion of two N-wells.

1 116. Apple’s acts of infringement have occurred within this district and
2 elsewhere throughout the United States.

3 117. Qualcomm has been damaged and will suffer additional damages and
4 irreparable harm unless Apple is enjoined from further infringement. Qualcomm
5 will prove its irreparable harm and damages at trial.

6 COUNT 4 (PATENT INFRINGEMENT – U.S. PATENT NO. 8,838,949)

7 118. Qualcomm repeats and re-alleges the allegations of paragraphs 1
8 through 117 above as if fully set forth herein.

9 119. Qualcomm is the lawful owner of the ’949 patent, and has the full and
10 exclusive right to bring actions and recover damages for Apple’s infringement of
11 said patent.

12 120. In violation of 35 U.S.C. § 271, Apple has been and is still infringing,
13 contributing to infringement, and/or inducing others to infringe the ’949 patent by
14 making, using, offering for sale, selling, or importing mobile devices that practice
15 the patent, including but not limited to iPhone 7 and iPhone 7 Plus devices.

16 121. The accused devices allow a secondary processor that does not have its
17 own flash memory to boot up. For example, an image for the secondary processor is
18 stored in a memory coupled to a primary processor. A scatter loader directly
19 transfers the image from the memory coupled to the primary processor into a
20 memory of the secondary processor, allowing the secondary processor to boot up.

21 122. The accused devices infringe at least claims 1-8, 10-14, 16, 20, and 22
22 of the ’949 patent.

23 123. The accused devices infringe claims 1, 10, 16, 20, and 22 of the ’949
24 patent as follows. Each accused device is a multi-processor system. Each accused
25 device includes a primary processor—an Apple A10 application processor. Each
26 accused device also includes a secondary processor—a baseband processor. An
27 interface such as a PCIe interface communicatively couples the primary processor
28 and the secondary processor. The primary processor is coupled with a memory

1 storing an executable software image for the secondary processor. The secondary
2 processor includes a system memory and a hardware buffer for receiving an image
3 header and at least one data segments of an executable software image. The
4 secondary processor receives the image header and each data segment separately
5 over the interface. The secondary processor includes a scatter loader controller,
6 which is configured to load the image header and scatter load each received data
7 segment based at least in part on the loaded image header directly from the hardware
8 buffer to the system memory of the secondary processor. Thus, the accused devices
9 infringe claims 1, 10, 16, 20, and 22 of the '949 patent.

10 124. With respect to claims 2 and 12 of the '949 patent, the scatter loader
11 controller loads the executable software image directly from the hardware buffer to
12 the system memory of the secondary processor without copying data between
13 system memory locations on the secondary processor. Indeed, this is inherent in the
14 scatter loader controller scatter loading each data segment of the executable
15 software image directly on the to the system memory of the secondary processor as
16 recited in claims 1 and 10. Thus, the accused devices infringe claims 2 and 12 of
17 the '949 patent.

18 125. With respect to claim 3 of the '949 patent, the secondary processor
19 receives raw image data of the executable software image via the interface. Thus,
20 the accused devices infringe claim 3 of the '949 patent.

21 126. With respect to claim 4 of the '949 patent, the secondary processor is
22 configured to process the image header, which includes destination addresses of
23 each data segment, to determine at least one location within the system memory of
24 the secondary processor to store the at least one data segment. Thus, the accused
25 devices infringe claim 4 of the '949 patent.

26 127. With respect to claim 5 of the '949 patent, the secondary processor is
27 configured to determine the at least one location within the system memory to store
28 that data segment based on the received image header, which includes destination

1 addresses of each data segment, before receiving that data segment. Thus, the
2 accused devices infringe claim 5 of the '949 patent.

3 128. With respect to claim 6 of the '949 patent, the secondary processor
4 includes a non-volatile memory such as a boot read only memory (ROM) storing a
5 boot loader that initiates transfer of the executable software images for the
6 secondary processor. Thus, the accused devices infringe claim 6 of the '949 patent.

7 129. With respect to claims 7 and 14 of the '949 patent, the primary
8 processor and the secondary processor are located on different chips. Thus, the
9 accused devices infringe claims 7 and 14 of the '949 patent.

10 130. With respect to claim 8 of the '949 patent, a hardware buffer of a
11 transport mechanism, such as a hardware buffer at endpoint (EP) of a PCIe interface
12 that buffers data received from root complex (RC), receives data segments and the
13 scatter loader controller scatter loads the received data segments directly to the
14 system memory of the secondary processor. As the hardware buffer at the EP of a
15 PCIe interface does not have the capacity to store an entire executable software
16 image, the secondary processor loads a portion of the executable software image
17 into its system memory without an entire executable software image being stored in
18 the hardware buffer. Thus, the accused devices infringe claim 8 of the '949 patent.

19 131. With respect to claim 11 of the '949 patent, the accused devices each
20 boot the secondary processor using the executable software image. Thus, the
21 accused devices infringe claim 11 of the '949 patent.

22 132. With respect to claim 13 of the '949 patent, the accused devices each
23 processes the image header prior to the loading of each data segment. Thus, the
24 accused devices infringe claim 13 of the '949 patent.

25 133. On information and belief, Apple also knowingly induces and/or
26 contributes to the infringement of at least claims 10-14 and 22 of the '949 patent by
27 others. On information and belief, Apple has had knowledge of the '949 patent, and
28 its infringement of the '949 patent, at least since the time this lawsuit was filed.

1 Additionally, Qualcomm has provided technical assistance and solutions to Apple
2 under non-disclosure agreements. Apple was aware of, and implemented,
3 Qualcomm's technology in certain of its devices without authorization. On
4 information and belief, Apple tests, demonstrates, or otherwise operates the accused
5 devices in the United States, thereby performing the claimed methods and directly
6 infringing any asserted claims of the '949 patent requiring such operation.
7 Similarly, Apple's customers and the end users of the accused devices test and/or
8 operate the accused devices in the United States in accordance with Apple's
9 instructions contained in, for example, its user manuals, thereby also performing the
10 claimed methods and directly infringing the asserted claims of the Asserted Patents
11 requiring such operation.

12 134. Apple also contributes to infringement of the '949 patent by selling for
13 importation into the United States, importing into the United States, and/or selling
14 within the United States after importation the accused devices and the non-staple
15 constituent parts of those devices, which are not suitable for substantial non-
16 infringing use and which embody a material part of the invention described in the
17 '949 patent. These mobile electronic devices are known by Apple to be especially
18 made or especially adapted for use in the infringement of the '949 patent. Apple
19 also contributes to the infringement of the '949 patent by selling for importation into
20 the United States, importing into the United States, and/or selling within the United
21 States after importation components, such as the chipsets or software containing the
22 infringing functionality, of the accused devices, which are not suitable for
23 substantial non-infringing use and which embody a material part of the invention
24 described in the '949 patent. These mobile devices are known by Apple to be
25 especially made or especially adapted for use in the infringement of the '949 patent.
26 Specifically, on information and belief, Apple sells the accused devices to resellers,
27 retailers, and end users with knowledge that the devices are used for infringement.
28 End users of those mobile electronic devices directly infringe the '949 patent.

1 135. Apple's acts of infringement have occurred within this district and
2 elsewhere throughout the United States.

3 136. Qualcomm has been damaged and will suffer additional damages and
4 irreparable harm unless Apple is enjoined from further infringement. Qualcomm
5 will prove its irreparable harm and damages at trial.

6 COUNT 5 (PATENT INFRINGEMENT – U.S. PATENT NO. 9,535,490)

7 137. Qualcomm repeats and re-alleges the allegations of paragraphs 1
8 through 136 above as if fully set forth herein.

9 138. Qualcomm is the lawful owner of the '490 patent, and has the full and
10 exclusive right to bring actions and recover damages for Apple's infringement of
11 said patent.

12 139. In violation of 35 U.S.C. § 271, Apple has been and is still infringing,
13 contributing to infringement, and/or inducing others to infringe the '490 patent by
14 making, using, offering for sale, selling, or importing devices that practice the
15 patent, including but not limited to iPhone 7 and iPhone 7 Plus devices.

16 140. The accused devices allow reduction in the time in which a bus
17 coupling a modem processor to an application processor is in a high power state,
18 which advantageously reduces power consumption. For example, downlink data to
19 be transferred from the modem processor to the application processor is held until a
20 timer such as a modem timer or a downlink timer expires. Further, uplink data to be
21 transferred from the application processor to the modem processor is held until the
22 bus is in the active state to transfer the downlink data. Syncing the timing of data
23 transfers reduces the amount of time the bus is in the active state, resulting in power
24 savings.

25 141. The accused devices infringe at least claims 1-6, 8, 10, 16, 17, and 31
26 of the '490 patent.

27 142. The accused devices infringe claims 1, 16, and 31 of the '490 patent as
28 follows. Each accused device is a mobile terminal including an Apple A10

1 application processor and a modem processor. Each accused device also includes an
2 interconnectivity bus such as a PCIe bus communicatively coupling the application
3 processor and the modem processor. Each accused device further includes a modem
4 timer or a downlink timer. The modem processor is configured to hold “modem
5 processor to application processor data” such as downlink data until expiration of
6 the modem timer/downlink timer. The application processor is configured to hold
7 “application processor to modem processor data” such as uplink data until the bus is
8 in an active state to transfer downlink data from the modem processor to the
9 application processor, and transmits the uplink data to the modem processor before
10 the interconnectivity bus transitions from an active power state to a low power state.
11 For example, the modem processor pulls downlink data from the application
12 processor after transmitting one or more downlink data packets to the application
13 processor. The application processor holds the uplink data until triggered by receipt
14 of one or more downlink data packets from the modem processor, and sends one or
15 more uplink data packets to the modem processor responsive to the receipt of one or
16 more downlink data packets from the modem processor. The application processor
17 holds the uplink data, for example application data generated by an application
18 associated with the application processor, until receipt of one or more downlink data
19 packets from the modem processor or until expiration of an uplink timer, whichever
20 occurs first. Thus, the accused devices infringe claims 1, 16, and 31 of the ’490
21 patent.

22 143. With respect to claims 2 and 17 of the ’490 patent, the interconnectivity
23 bus includes a peripheral component interconnect (PCI) compliant bus. Thus, the
24 accused devices infringe claims 2 and 17 of the ’490 patent.

25 144. With respect to claim 3 of the ’490 patent, the PCI compliant bus
26 includes a PCI express (PCIe) bus. Thus, the accused devices infringe claim 3 of
27 the ’490 patent.

28 145. With respect to claim 4 of the ’490 patent, the application processor

1 starts an application timer that has a period longer than a period of the modem timer.
2 Thus, the accused devices infringe claim 4 of the '490 patent.

3 146. With respect to claim 5 of the '490 patent, the application processor is
4 configured to hold the uplink data until receipt of the downlink data from the
5 modem processor or expiration of an uplink timer having a period longer than a
6 period of the modem timer, whichever occurs first. Thus, the accused devices
7 infringe claim 5 of the '490 patent.

8 147. With respect to claim 6 of the '490 patent, each accused device
9 includes a modem timer implemented in software. Thus, the accused devices
10 infringe claim 6 of the '490 patent.

11 148. With respect to claim 8 of the '490 patent, each accused device
12 includes a modem timer. The modem processor includes the modem timer as
13 recited in claim 8; thus, the accused devices infringe claim 8.

14 149. With respect to claim 10 of the '490 patent, each accused device
15 includes an application timer, and the modem processor is configured to instruct the
16 application processor to send an interrupt if no data is received within one time slot
17 of the application timer. Thus, the accused devices infringe claim 10 of the '490
18 patent.

19 150. On information and belief, Apple also knowingly induces and/or
20 contributes to the infringement of at least claims 16-17 of the '490 patent by others.
21 On information and belief, Apple has had knowledge of the '490 patent, and its
22 infringement of the '490 patent at least since the time this lawsuit was filed.
23 Additionally, Qualcomm has provided technical assistance and solutions to Apple
24 under non-disclosure agreements. Apple was aware of, and implemented,
25 Qualcomm's technology in certain of its devices without authorization. On
26 information and belief, Apple tests, demonstrates, or otherwise operates the accused
27 devices in the United States, thereby performing the claimed methods and directly
28 infringing any asserted claims of the '490 patent requiring such operation.

1 Similarly, Apple's customers and the end users of the accused devices test and/or
2 operate the accused devices in the United States in accordance with Apple's
3 instructions contained in, for example, its user manuals, thereby also performing the
4 claimed methods and directly infringing the asserted claims of the Asserted Patents
5 requiring such operation.

6 151. Apple also contributes to infringement of the '490 patent by selling for
7 importation into the United States, importing into the United States, and/or selling
8 within the United States after importation the accused devices and the non-staple
9 constituent parts of those devices, which are not suitable for substantial non-
10 infringing use and which embody a material part of the invention described in the
11 '490 patent. These mobile electronic devices are known by Apple to be especially
12 made or especially adapted for use in the infringement of the '490 patent. Apple
13 also contributes to the infringement of the '490 patent by selling for importation into
14 the United States, importing into the United States, and/or selling within the United
15 States after importation components, such as the chipsets or software containing the
16 infringing functionality, of the accused devices, which are not suitable for
17 substantial non-infringing use and which embody a material part of the invention
18 described in the '490 patent. These mobile devices are known by Apple to be
19 especially made or especially adapted for use in the infringement of the '490 patent.
20 Specifically, on information and belief, Apple sells the accused devices to resellers,
21 retailers, and end users with knowledge that the devices are used for infringement.
22 End users of those mobile electronic devices directly infringe the '490 patent.

23 152. Apple's acts of infringement have occurred within this district and
24 elsewhere throughout the United States.

25 153. Qualcomm has been damaged and will suffer additional damages and
26 irreparable harm unless Apple is enjoined from further infringement. Qualcomm
27 will prove its irreparable harm and damages at trial.

28 COUNT 6 (PATENT INFRINGEMENT – U.S. PATENT NO. 9,608,675)

1 154. Qualcomm repeats and re-alleges the allegations of paragraphs 1
2 through 153 above as if fully set forth herein.

3 155. Qualcomm is the lawful owner of the '675 patent, and has the full and
4 exclusive right to bring actions and recover damages for Apple's infringement of
5 said patent.

6 156. In violation of 35 U.S.C. § 271, Apple has been and is still infringing,
7 contributing to infringement, and/or inducing others to infringe the '675 patent by
8 making, using, offering for sale, selling, or importing mobile devices that practice
9 the patent, such as iPhone 7 and iPhone 7 Plus devices.

10 157. The accused devices infringe at least claims 1-3 and 7-14 of the
11 '675 patent. For example, with respect to claim 1, the accused devices include a
12 power tracker that determines a power tracking signal based on in-phase (I) and
13 quadrature (Q) components of one or more carrier aggregated transmit signals being
14 sent simultaneously. In the accused devices, a power tracker receives I and Q
15 components corresponding to carrier aggregated transmit signals and further
16 generates a power tracking signal based on a combination of the plurality of I and Q
17 components. Further, in the accused devices the carrier aggregated transmit signals
18 are Orthogonal Frequency Division Multiplexing (OFDM) or Single Carrier
19 Frequency Division Multiple Access (SC-FDMA) signals. In the accused devices, a
20 power supply generator generates a power supply voltage based on a power tracking
21 signal. The accused devices also include a power amplifier configured to receive
22 the power supply voltage and the carrier aggregated transmit signals being sent
23 simultaneously in a single output radio frequency (RF) signal.

24 158. With respect to claim 2, the accused devices include a power
25 tracker that determines an overall power of the plurality of carrier aggregated
26 transmit signals based on the I and Q components of the plurality of carrier
27 aggregated transmit signals, and determines the single power tracking signal based
28 on the overall power of the plurality of carrier aggregated transmit signals. On

1 information and belief, the envelope bandwidth of the power tracking signal is
2 reduced compared to the combined enveloped bandwidth of the transmit signals.
3 On information and belief, the accused devices determine an overall power of the
4 dual CA transmit signals based on their I and Q components, and determine the
5 single power tracking signal based on the overall power of the CA transmit signals.

6 159. With respect to claim 3, the accused devices include a power
7 tracker to determine a power of each transmit signal in the plurality of carrier
8 aggregated transmit signals based on the I and Q components of each transmit
9 signal, and determine the single power tracking signal based on a sum of said power
10 of each transmit signal of the plurality of carrier aggregated transmit signals. On
11 information and belief, the envelope bandwidth of the power tracking signal
12 generated by the baseband processor is smaller than the combined envelope
13 bandwidth of the transmit signals. On information and belief, the baseband
14 processor performs in one of two ways, namely, determining a power of each
15 transmit signal in the dual CA transmit signals based on the I and Q components of
16 each transmit signal, and determining the single power tracking signal based on a
17 sum of said power of each transmit signal of the dual CA transmit signals.

18 160. With respect to claim 7, the accused devices include a power
19 supply generator with a power tracking amplifier configured to receive the power
20 tracking signal and generate the power supply voltage.

21 161. With respect to claim 8, the accused devices include a power
22 supply generator with a switcher configured to sense a first current from the power
23 tracking amplifier and provide a second current for the power supply voltage based
24 on the sensed first current.

25 162. With respect to claim 9, the accused devices include a power
26 supply generator with a boost converter configured to receive a battery voltage and
27 provide a boosted voltage for the power tracking amplifier.

28

1 (b) Awarding damages in an amount to be proven at trial, but in no event
2 less than a reasonable royalty for its infringement including pre-judgment and post-
3 judgment interest at the maximum rate permitted by law;

4 (c) Ordering a permanent injunction enjoining Apple, its officers, agents,
5 servants, employees, attorneys, and all other persons in active concert or
6 participation with Apple from infringing the Patents-in-Suit;

7 (d) Ordering an award of reasonable attorneys' fees to Qualcomm as
8 provided by 35 U.S.C. § 285;

9 (e) Awarding expenses, costs, and disbursements in this action, including
10 prejudgment interest; and

11 (f) Awarding such other and further relief as the Court deems just and
12 proper.

13 DEMAND FOR JURY TRIAL

14 Pursuant to Rule 38(b) of the Federal Rules of Civil Procedure, Qualcomm
15 demands a jury trial on all issues triable by jury.

16 Dated: July 6, 2017

s/ Randall E. Kay

Randall E. Kay
rekay @jonesday.com

JONES DAY

Karen P. Hewitt (SBN 145309)

kphewitt@jonesday.com

Randall E. Kay (SBN 149369)

rekay@jonesday.com

4655 Executive Drive, Suite 1500

San Diego, California 92121

Telephone: (858) 314-1200

Facsimile: (858) 345-3178

QUINN EMANUEL URQUHART &

SULLIVAN, LLP

David A. Nelson (*pro hac vice* forthcoming)

(Ill. Bar No. 6209623)

davenelson@quinnemanuel.com

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

500 West Madison St., Suite 2450
Chicago, Illinois 60661
Telephone: (312) 705-7400
Facsimile: (312) 705-7401

Richard W. Erwine (*pro hac vice* forthcoming)
(N.Y. Bar No. 2753929)
richarderwine@quinnemanuel.com
Alexander Rudis (*pro hac vice* forthcoming)
(N.Y. Bar No. 4232591)
alexanderrudis@quinnemanuel.com
51 Madison Avenue, 22nd Floor
New York, NY 10010
Telephone: (212) 849-7000
Facsimile: (212) 849-7100

Sean S. Pak (SBN 219032)
seanpak@quinnemanuel.com
50 California Street, 22nd Floor
San Francisco, CA 94111
Telephone: (415) 875-6600
Facsimile: (415) 875-6700

S. Alex Lasher (SBN 224034)
alexlasher@quinnemanuel.com
777 6th Street NW, 11th Floor
Washington, DC 20001
Telephone: (202) 538-8000
Facsimile: (202) 538-8100

CRAVATH, SWAINE & MOORE LLP
Evan R. Chesler (*pro hac vice* forthcoming)
(N.Y. Bar No. 1475722)
echesler@cravath.com
Keith R. Hummel (*pro hac vice* forthcoming)
(N.Y. Bar No. 2430668)
khummel@cravath.com
Richard J. Stark (*pro hac vice* forthcoming)
(N.Y. Bar No. 2472603)
rstark@cravath.com
Gary A. Bornstein (*pro hac vice* forthcoming)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

(N.Y. Bar No. 2916815)
gbornstein@cravath.com
J. Wesley Earnhardt (*pro hac vice* forthcoming)
(N.Y. Bar No. 4331609)
wearnhardt@cravath.com
Yonatan Even (*pro hac vice* forthcoming)
(N.Y. Bar No. 4339651)
yeven@cravath.com
Vanessa A. Lavelly (*pro hac vice* forthcoming)
(N.Y. Bar No. 4867412)
vlavelly@cravath.com
Worldwide Plaza, 825 Eighth Avenue
New York, NY 10019
Telephone: (212) 474-1000
Facsimile: (212) 474-3700

Attorneys for Plaintiff
QUALCOMM INCORPORATED

EXHIBIT INDEX

EXHIBIT	DESCRIPTION	PAGES
A	U.S. Patent No. 8,633,936	1-24
B	U.S. Patent No. 8,698,558	25-43
C	U.S. Patent No. 8,487,658	44-60
D	U.S. Patent No. 8,838,949	61-79
E	U.S. Patent No. 9,535,490	80-109
F	U.S. Patent No. 9,608,675	110-134

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

EXHIBIT A

U 7629935

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

**UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office**

April 06, 2017

**THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM
THE RECORDS OF THIS OFFICE OF:**

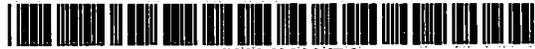
**U.S. PATENT: 8,633,936
ISSUE DATE: *January 21, 2014***

**By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office**



A handwritten signature in black ink that reads "T. Lawrence".

**T. LAWRENCE
Certifying Officer**



US008633936B2

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

(10) **Patent No.:** **US 8,633,936 B2**
(45) **Date of Patent:** ***Jan. 21, 2014**

(54) **PROGRAMMABLE STREAMING
PROCESSOR WITH MIXED PRECISION
INSTRUCTION EXECUTION**

7,079,156 B1 * 7/2006 Hutchins et al. 345/606
7,418,606 B2 * 8/2008 Holmer 713/320
7,685,579 B2 * 3/2010 Knowles 717/140
7,716,655 B2 * 5/2010 Uchida 717/140

(Continued)

(75) **Inventors:** **Yun Du**, San Diego, CA (US); **Chun Yu**,
San Diego, CA (US); **Guofang Jiao**, San
Diego, CA (US); **Stephen Molloy**,
Carlsbad, CA (US)

FOREIGN PATENT DOCUMENTS

CN 101131768 A 2/2008
JP 04135277 5/1992

(73) **Assignee:** **QUALCOMM Incorporated**, San
Diego, CA (US)

(Continued)

(*) **Notice:** Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 1014 days.

OTHER PUBLICATIONS

"Modifiers for ps_2_0 and Above" <http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/directx9_c_summer_03/directx/graphics/reference/assemblylanguageshaders/pixelshaders/instructions/modifiers_ps_2_0.asp> Microsoft DirectX 9.0 SDK Update (Summer 2003).

This patent is subject to a terminal disclaimer.

(Continued)

(21) **Appl. No.:** **12/106,654**

(22) **Filed:** **Apr. 21, 2008**

Primary Examiner — Joni Richer

(65) **Prior Publication Data**

(74) *Attorney, Agent, or Firm* — James R. Gambale, Jr.

US 2009/0265528 A1 Oct. 22, 2009

(57) **ABSTRACT**

(51) **Int. Cl.**
G06T 1/00 (2006.01)
G06F 15/00 (2006.01)
G06F 15/16 (2006.01)

The disclosure relates to a programmable streaming processor that is capable of executing mixed-precision (e.g., full-precision, half-precision) instructions using different execution units. The various execution units are each capable of using graphics data to execute instructions at a particular precision level. An exemplary programmable shader processor includes a controller and multiple execution units. The controller is configured to receive an instruction for execution and to receive an indication of a data precision for execution of the instruction. The controller is also configured to receive a separate conversion instruction that, when executed, converts graphics data associated with the instruction to the indicated data precision. When operable, the controller selects one of the execution units based on the indicated data precision. The controller then causes the selected execution unit to execute the instruction with the indicated data precision using the graphics data associated with the instruction.

(52) **U.S. Cl.**
USPC 345/522; 345/501; 345/502

(58) **Field of Classification Search**
USPC 345/426, 501, 502, 522, 530, 536, 559,
345/561

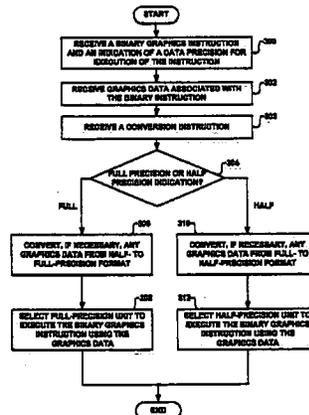
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,734,874 A * 3/1998 Van Hook et al. 345/559
5,784,588 A * 7/1998 Leung 712/216
5,953,237 A * 9/1999 Indermaur et al. 700/295
6,044,216 A * 3/2000 Bhargava et al. 717/114

68 Claims, 6 Drawing Sheets



(56)

References Cited

U.S. PATENT DOCUMENTS

2005/0066205 A1 3/2005 Holmer
2007/0186082 A1 8/2007 Prokopenko et al.
2007/0273698 A1 11/2007 Du et al.
2007/0283356 A1 12/2007 Du et al.
2007/0292047 A1 12/2007 Jiao et al.
2007/0296729 A1 12/2007 Du et al.
2008/0235316 A1 9/2008 Du et al.

FOREIGN PATENT DOCUMENTS

JP 5303498 A 11/1993
JP 6297031 A 10/1994

JP 2005293386 A 10/2005
JP 2007079844 A 3/2007
JP 2007514209 A 5/2007
JP 2009140491 A 6/2009
TW 200519730 6/2005

OTHER PUBLICATIONS

International Search Report & Written Opinion—PCT/US2009/041268, International Search Authority—European Patent Office—Sep. 29, 2009.

U. J. Kapasi et al.: "Programmable Stream Processors" Computer, Aug. 2003, pp. 54-62, XP002543695 Published by IEEE Computer Society the whole document.

* cited by examiner

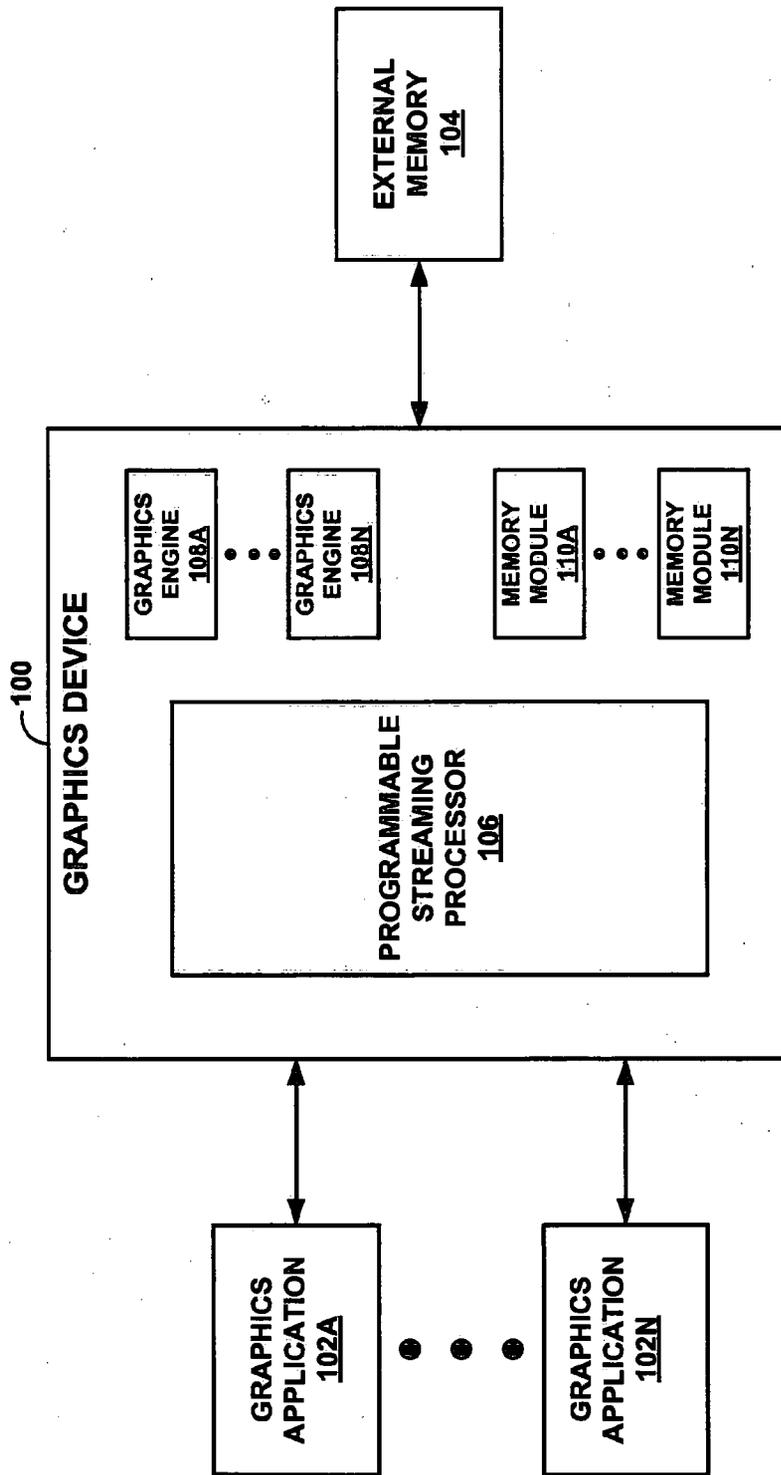


FIG. 1

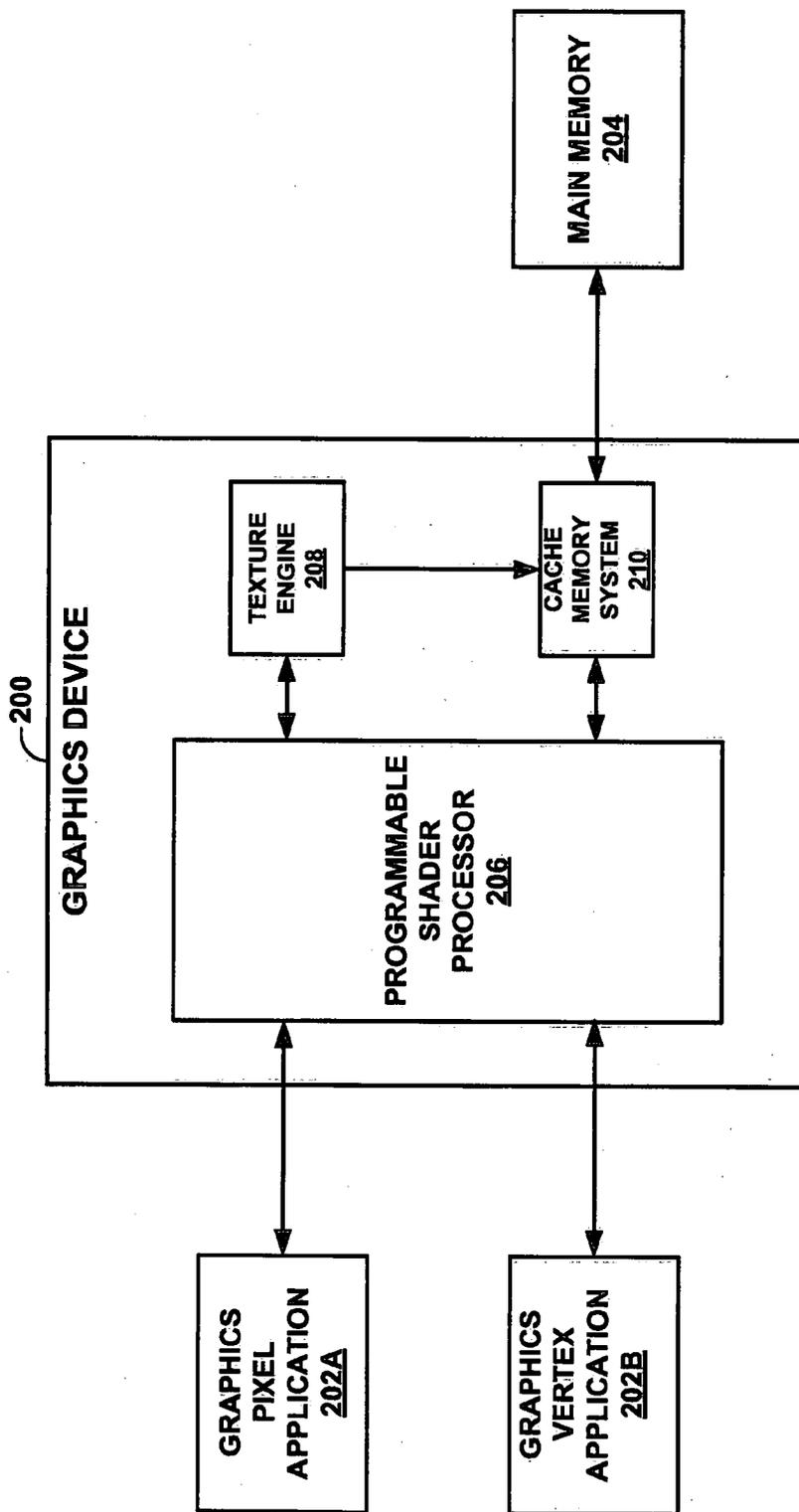


FIG. 2A

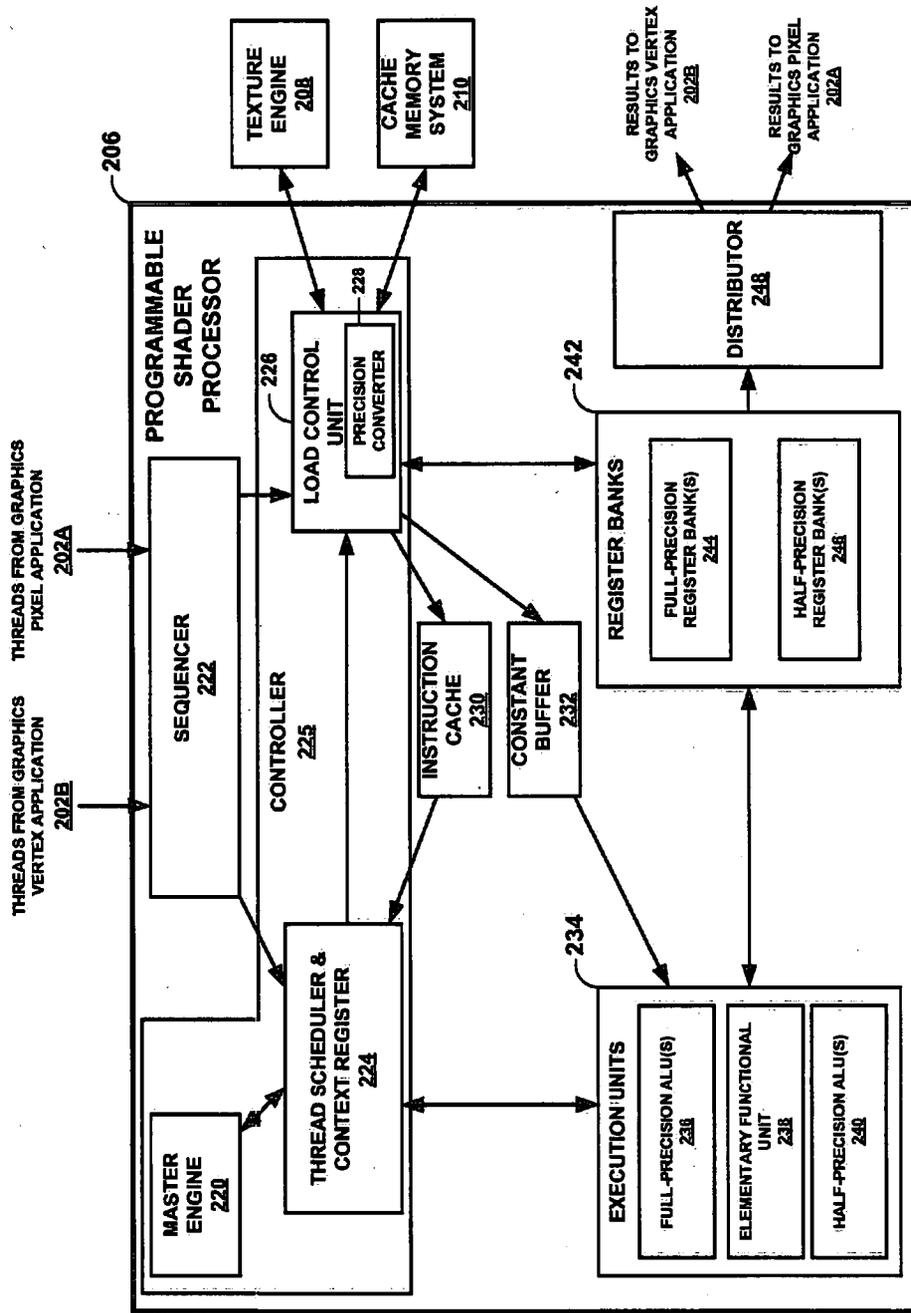


FIG. 2B

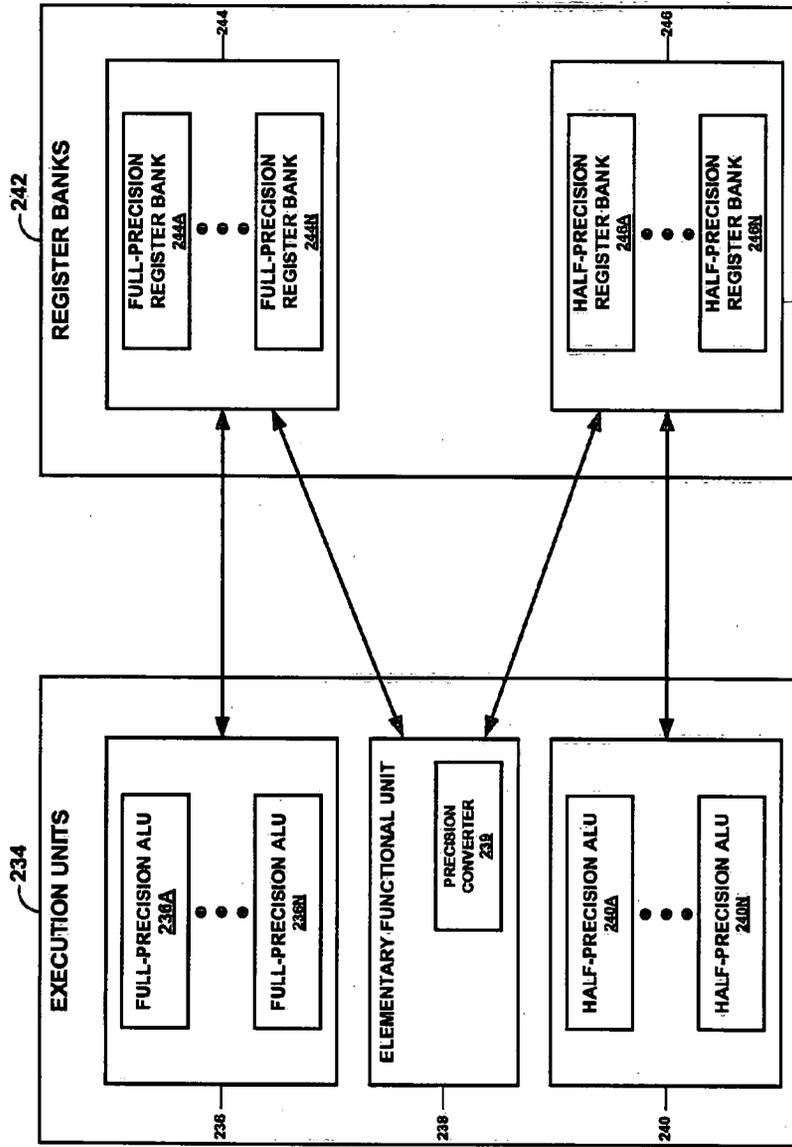


FIG. 2C

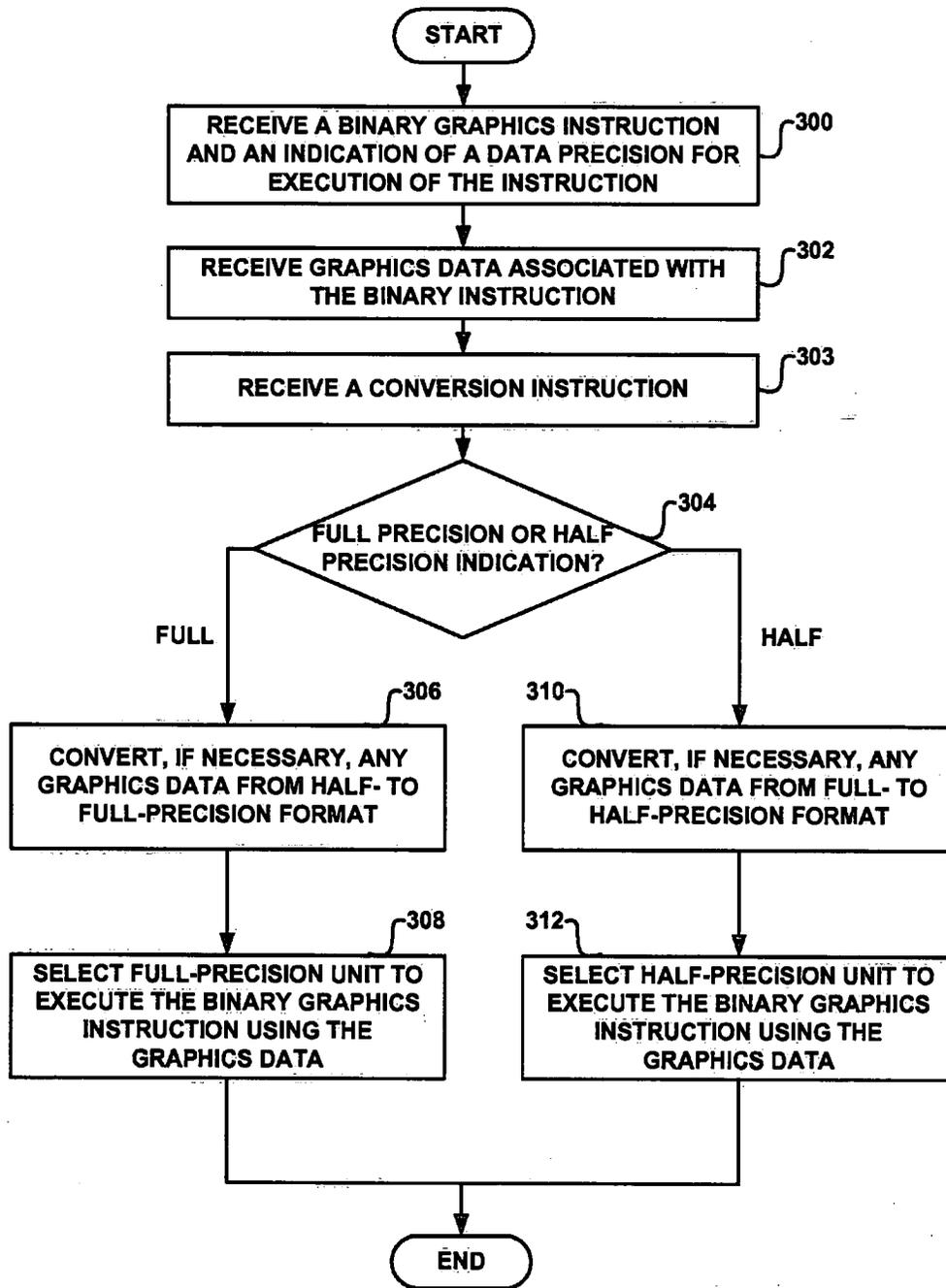


FIG. 3

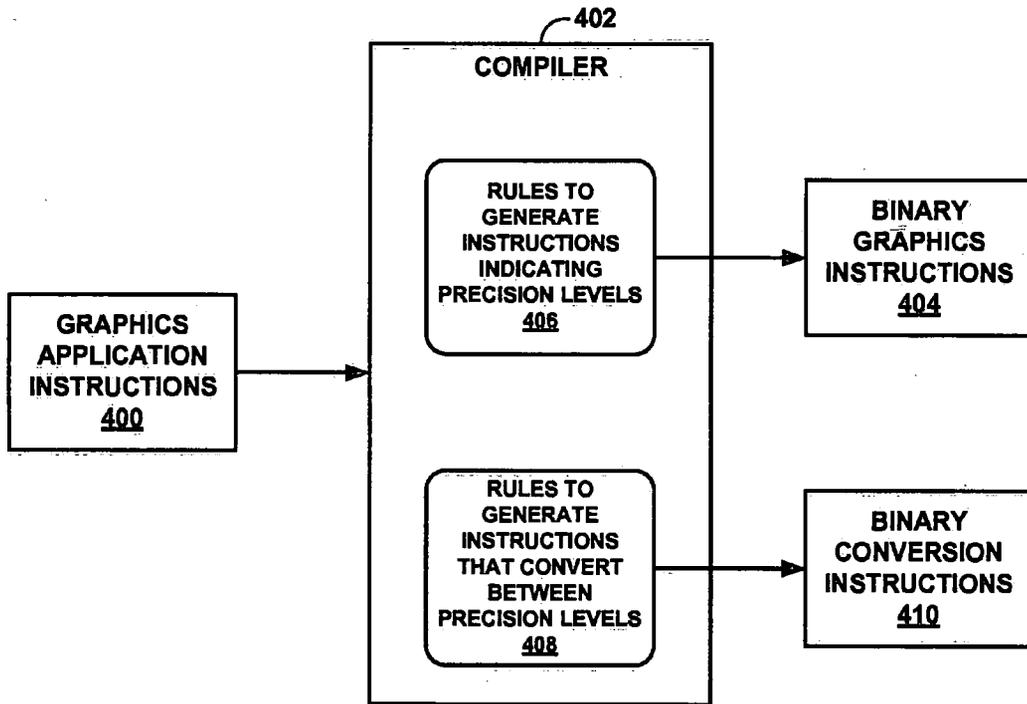


FIG. 4

**PROGRAMMABLE STREAMING
PROCESSOR WITH MIXED PRECISION
INSTRUCTION EXECUTION**

TECHNICAL FIELD

The disclosure relates to graphics processing and, more particularly, to graphics processor architectures.

BACKGROUND

Graphics devices are widely used to render 2-dimensional (2-D) and 3-dimensional (3-D) images for various applications, such as video games, graphics programs, computer-aided design (CAD) applications, simulation and visualization tools, imaging, and the like. A graphics device may perform various graphics operations to render an image. The graphics operations may include rasterization, stencil and depth tests, texture mapping, shading, and the like. A 3-D image may be modeled with surfaces, and each surface may be approximated with polygons, such as triangles. The number of triangles used to represent a 3-D image for rendering purposes is dependent on the complexity of the surfaces as well as the desired resolution of the image.

Each triangle may be defined by three vertices, and each vertex is associated with various attributes such as space coordinates, color values, and texture coordinates. When a graphics device uses a vertex processor during the rendering process, the vertex processor may process vertices of the various triangles. Each triangle is also composed of picture elements (pixels). When the graphics device also, or separately, uses a pixel processor during the rendering process, the pixel processor renders each triangle by determining the values of the components of each pixel within the triangle.

In many cases, a graphics device may utilize a shader processor to perform certain graphics operations such as shading. Shading is a highly complex graphics operation involving lighting and shadowing. The shader processor may need to execute a variety of different instructions when performing rendering, and typically includes one or more execution units to aid in the execution of these instructions. For example, the shader processor may include arithmetic logic units (ALU's) and/or an elementary functional unit (EFU) as execution units. Often, these execution units are capable of executing instructions using full data-precision circuitry. However, such circuitry can often require more power, and the execution units may take up more physical space within the shader processor integrated circuit used by the graphics device.

SUMMARY

In general, the disclosure relates to a programmable streaming processor of a graphics device that is capable of executing mixed-precision (e.g., full-precision, half-precision) instructions using different execution units. For example, the programmable processor may include one or more full-precision execution units along with one or more half-precision execution units. Upon receipt of a binary instruction and an indication of a data precision for execution of the instruction, the processor is capable of selecting an appropriate execution unit for executing the received instruction with the indicated data precision. The processor may comprise an instruction-based, adaptive streaming processor for mobile graphics applications.

By doing so, the processor may avoid using one execution unit to execute instructions with various different data preci-

sions. As a result, unnecessary precision promotion may be reduced or eliminated. In addition, application programmers may have increased flexibility when writing application code. An application programmer may specify different data precision levels for different application instructions, which are then compiled into one or more binary instructions that are processed by the processor.

In one aspect, the disclosure is directed to a method that includes receiving a graphics instruction for execution within a programmable streaming processor, receiving an indication of a data precision for execution of the graphics instruction, and receiving a conversion instruction that, when executed by the processor, converts graphics data associated with the graphics instruction to the indicated data precision, wherein the conversion instruction is different than the graphics instruction. The method further includes selecting one of a plurality of execution units within the processor based on the indicated data precision, and using the selected execution unit to execute the graphics instruction with the indicated data precision using the graphics data associated with the graphics instruction.

In one aspect, the disclosure is directed to a computer-readable medium including instructions for causing a programmable streaming processor to receive a graphics instruction for execution within the processor, to receive an indication of a data precision for execution of the graphics instruction, and to receive a conversion instruction that, when executed by the processor, converts graphics data associated with the graphics instruction to the indicated data precision, wherein the conversion instruction is different than the graphics instruction. The computer-readable medium further includes instructions for causing the processor to select one of a plurality of execution units within the processor based on the indicated data precision, and to use the selected execution unit to execute the graphics instruction with the indicated data precision using the graphics data associated with the graphics instruction.

In one aspect, the disclosure is directed to a programmable streaming processor that includes a controller and multiple execution units. The controller is configured to receive a graphics instruction for execution and to receive an indication of a data precision for execution of the graphics instruction. The controller is also configured to receive a conversion instruction that, when executed by the processor, converts graphics data associated with the graphics instruction to the indicated data precision, wherein the conversion instruction is different than the graphics instruction. When operable, the controller selects one of the execution units based on the indicated data precision. The controller then causes the selected execution unit to execute the graphics instruction with the indicated data precision using the graphics data associated with the graphics instruction.

In another aspect, the disclosure is directed to a computer-readable medium that includes instructions for causing a processor to analyze a plurality of application instructions for a graphics application, and, for each application instruction that specifies a first data precision level for its execution, to generate one or more corresponding compiled instructions that each indicate the first data precision level for its execution. The computer-readable medium includes further instructions for causing the processor to generate one or more conversion instructions to convert graphics data from a second, different data precision level to the first data precision level when the one or more compiled instructions are executed.

In one aspect, the disclosure is directed to a computer-readable data storage medium having one or more first

executable instructions that, when executed by a programmable streaming processor, support one or more functions of a graphics application, wherein each of the first executable instructions indicates a first data precision level for its execution. The computer-readable data storage medium further includes one or more second executable instructions that, when executed by the processor, support one or more functions of the graphics application, wherein each of the second executable instructions indicates a second data precision level different from the first data precision level for its execution. The computer-readable data storage medium further includes one or more third executable instructions that, when executed by the processor, support one or more functions of the graphics application, wherein each of the third executable instructions converts graphics data from the second data precision level to the first data precision level when the one or more first executable instructions are executed.

The details of one or more aspects of the disclosure are set forth in the accompanying drawings and the description below. Other features, objects, and advantages will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF DRAWINGS

FIG. 1 is a block diagram illustrating various components that may be included within a graphics processing system, according to an aspect of the disclosure.

FIG. 2A is a block diagram illustrating an exemplary graphics processing system that includes a programmable shader processor, according to an aspect of the disclosure.

FIG. 2B is a block diagram illustrating further details of the shader processor shown in FIG. 2A, according to an aspect of the disclosure.

FIG. 2C is a block diagram illustrating further details of the execution units and register banks shown in FIG. 2B, according to an aspect of the disclosure.

FIG. 3 is a flow diagram illustrating an exemplary method that may be performed by the shader processor shown in FIGS. 2A-2B, according to an aspect of the disclosure.

FIG. 4 is a block diagram illustrating a compiler that may be used to generate graphics instructions to be executed by the streaming processor shown in FIG. 1 or by the shader processor shown in FIGS. 2A-2B, according to an aspect of the disclosure.

DETAILED DESCRIPTION

FIG. 1 is a block diagram illustrating various components that may be included within a graphics processing system, according to one aspect of the disclosure. This graphics processing system may be a stand-alone system or may be part of a larger system, such as a computing system or a wireless communication device (such as a wireless communication device handset), or part of a digital camera or other video device. The exemplary system shown in FIG. 1 may include one or more graphics applications 102A-102N, a graphics device 100, and external memory 104. Graphics device 100 may be communicatively coupled to external memory 104 and each of graphics applications 102A-102N. In one aspect, graphics device 100 may be included on one or more integrated circuits, or chips.

The graphics applications 102A-102N may include various different applications, such as video game, video, camera, or other graphics or streaming applications. These graphics applications 102A-102N may run concurrently and are each able to generate threads of execution to achieve desired results. A thread indicates a specific task that may be performed with a sequence of one or more graphics instructions.

Threads allow graphics applications 102A-102N to have multiple tasks performed simultaneously and to share resources.

Graphics device 100 receives the threads from graphics applications 102A-102N and performs the tasks indicated by these threads. In the aspect shown in FIG. 1, graphics device 100 includes a programmable streaming processor 106, one or more graphics engines 108A-108N, and one or more memory modules 110A-110N. Processor 106 may perform various graphics operations, such as shading, and may compute transcendental elementary functions for certain applications. In one aspect, processor 106 may comprise an instruction-based, adaptive streaming processor for mobile graphics applications. Graphics engines 108A-108N may perform other graphics operations, such as texture mapping. Memory modules 110A-110N may include one or more caches to store data and graphics instructions for processor 106 and graphics engines 108A-108N.

Graphics engines 108A-108N may include one or more engines that perform various graphics operations, such as triangle setup, rasterization, stencil and depth tests, attribute setup, and/or pixel interpolation. External memory 104 may be a large, slower memory with respect to memory modules 110A-110N. In one aspect, external memory 104 is located further away (e.g., off-chip) from graphics device 100. External memory 104 stores data and graphics instructions that may be loaded into one or more of the memory modules 110A-110N.

In one aspect, processor 106 is capable of executing mixed-precision (e.g., full-precision, half-precision) graphics instructions using different execution units, given that different graphics applications 102A-102N may have different requirements regarding ALU precision, performance, and input/output formats. As an example, processor 106 may include one or more full-precision execution units along with one or more partial-precision execution units. The partial-precision execution units may be, for example, half-precision execution units. Processor 106 may use its execution units to execute graphics instructions for one or more of graphics applications 102A-102N. Upon receipt of a binary instruction (such as from external memory 104 or one of memory modules 110A-110N), and also an indication of a data precision for execution of the graphics instruction, processor 106 may select an appropriate execution unit for executing the received instruction with the indicated data precision using graphics data. Processor 106 may also receive a separate conversion instruction that, when executed, converts graphics data associated with the graphics instruction to the indicated data precision. In one aspect, the conversion instruction is a separate instruction that is different from the graphics instruction.

The graphics data may be provided by graphics applications 102A-102N, or may be retrieved from external memory 104 or one of memory modules 110A-110N, or may be provided by one or more of graphics engines 108A-108N. By selectively executing instructions in different execution units based upon indicated data precisions, processor 106 may avoid using a single execution unit to execute both full-precision and half-precision instructions. In addition, programmers of graphics applications 102A-102N may have increased flexibility when writing application code. For example, an application programmer may specify data precision levels for application instructions, which are then compiled into one or more binary instructions that are processed by processor 106. Processor 106 selects appropriate execution units to execute the binary instructions based on the data precision associated with the execution units and the binary instructions. In addition, processor 106 may execute the

received conversion instruction to convert the graphics data associated with the instruction to the indicated data precision, if necessary. For example, if the provided graphics data has a data precision that is different from the indicated data precision, processor 106 may execute the conversion instruction to convert the graphics data to the indicated data precision, such that the graphics instruction may be executed by the selected execution unit.

FIG. 2A is a block diagram illustrating an exemplary graphics processing system that includes a programmable shader processor 206, according to one aspect. In this aspect, the graphics processing system shown in FIG. 2A is an exemplary instantiation of the more generic system shown in FIG. 1. In one aspect, shader processor 206 is a streaming processor. In FIG. 2A, the exemplary system includes two graphics applications 202A and 202B that are each communicatively coupled to a graphics device 200. In the example of FIG. 2A, graphics application 202A is a pixel application that is capable of processing and managing graphics imaging pixel data. In the example of FIG. 2A, graphics application 202B is a vertex application that is capable of processing and managing graphics imaging vertex data. In one aspect, graphics pixel application 202A comprises a pixel processing application, and graphics vertex application 202B comprises a vertex processing application.

In many cases, graphics pixel application 202A implements many functions that use a lower-precision (such as a half-precision) data format, but it may implement certain functions using a higher-precision (such as a full-precision) data format. Graphics pixel application 202A may also specify quad-based execution of instructions for pixel data. Typically, graphics vertex application 202B implements functions using a higher-precision data format, but may not specify quad-based execution of instructions for vertex data. Thus, different applications, such as applications 202A and 202B, and corresponding API's to graphics device 200, may specify different data precision requirements. And, within a given application 202A or 202B (and corresponding API), execution of mixed-precision instructions may be specified. For example, a shading language for graphics pixel application 202A may provide a precision modifier for shader instructions to be executed by shader processor 206. Thus, certain instructions may specify one precision level for execution while other instructions may specify another precision level. Shader processor 206 within graphics device 200 is capable of executing mixed-precision instructions in a uniform way.

In one aspect, shader processor 206 interacts with graphics applications 202A and 202B via one or more application program interfaces, or API's (not shown). For example, graphics pixel application 202A may interact with shader processor 206 via a first API, and graphics vertex application 202B may interact with shader processor 206 via a second API. The first API and second API may, in one aspect, comprise a common API. The API's may define one or more standard programming specifications used by graphics applications 202A and 202B to cause graphics device 200 to perform various graphical operations, including shading operations that may be performed by shader processor 206.

Graphics device 200 includes a shader processor 206. Shader processor 206 is capable of performing shading operations. Shader processor 206 is capable of exchanging pixel data with graphics pixel application 202A, and is further capable of exchanging vertex data with graphics vertex application 202B.

In the example of FIG. 2A, shader processor 206 also communicates with a texture engine 208 and a cache memory

system 210. Texture engine 208 is capable of performing texture-related operations, and is also communicatively coupled to cache memory system 210. Cache memory system 210 is coupled to main memory 204. Cache memory system 210 includes both an instruction cache and a data cache in an aspect. Instructions and/or data may be loaded from main memory 204 into cache memory system 210, which are then made available to texture engine 208 and shader processor 206. Shader processor 206 may communicate with external devices or components via either a synchronous or an asynchronous interface.

In one aspect, shader processor 206 is capable of executing mixed-precision graphics instructions using different execution units. In this aspect, shader processor 206 includes one or more full-precision execution units along with one or more half-precision execution units. Shader processor 206 may invoke its execution units to execute graphics instructions for one or both of graphics applications 202A and 202B. Upon receipt of a binary instruction (such as from cache memory system 210), and also an indication of a data precision for execution of the instruction, shader processor 206 is capable of selecting an appropriate execution unit for executing the received instruction with the indicated data precision using graphics data. Graphics pixel application 202A may provide, for example, pixel data to shader processor 206, and graphics vertex application 202B may provide vertex data to shader processor 206.

Shader processor may also receive a separate conversion instruction that, when executed, converts graphics data associated with the graphics instruction to the indicated data precision. In one aspect, the conversion instruction is a separate instruction that is different from the graphics instruction.

Graphics data may also be loaded from main memory 204 or cache memory system 210, or may be provided by texture engine 208. Graphics pixel application 202A and/or graphics vertex application 202B invoke threads of execution which cause shader processor 206 to load one or more binary instructions from cache memory system 210 for execution. In one aspect, each loaded instruction indicates a data precision for execution of the instruction. In addition, shader processor 206 may execute the received conversion instruction to convert the graphics data associated with the instruction to the indicated data precision, if necessary. For example, if the provided graphics data has a data precision that is different from the indicated data precision, shader processor 206 may execute the conversion instruction to convert the graphics data to the indicated data precision, such that the graphics instruction may be executed by the selected execution unit. By selectively executing instructions in different execution units based upon indicated data precisions, shader processor 206 may avoid using a single execution unit to execute both full-precision and half-precision instructions.

FIG. 2B is a block diagram illustrating further details of the shader processor 206 shown in FIG. 2A, according to one aspect. Within shader processor 206, a sequencer 222 receives threads from graphics applications 202A and 202B, and provides these threads to a thread scheduler & context register 224. In one aspect, sequencer 222 comprises a multiplexer (MUX). In one aspect, sequencer 222 determines which threads should be accepted, and may also allocate multiple-precision register space and/or other resources for each accepted thread. For example, sequencer 222 may allocate register space for half-precision instructions, and may also allocate register space for full-precision instructions.

In one aspect, pixel data received from graphics pixel application 202A includes attribute information in a pixel quad-based format (i.e., four pixels at a time). In this aspect,

execution units 234 may process four pixels at a time. In one aspect, execution units 234 may process data from graphics vertex application 202B one vertex at a time.

Thread scheduler 224 performs various functions to schedule and manage execution of threads, and may control execution sequence of threads. For each thread, thread scheduler 224 may determine whether resources required for that thread are ready, push the thread into a sleep queue if any resource (e.g., instruction, register file, or texture read) for the thread is not ready, and move the thread from the sleep queue to an active queue when all of the resources are ready, according to one aspect. Thread scheduler 224 interfaces with a load control unit 226 in order to synchronize the resources for the threads. In one aspect, thread scheduler 224 is part of a controller 225. FIG. 2B shows an example of controller 225. Controller 225 may control various functions related to the processing of instructions and data within shader processor 206. In the example of FIG. 2B, controller 225 includes thread scheduler 224, load control unit 226, and master engine 220. In certain aspects, controller 225 includes at least one of master engine 220, thread scheduler 224, and load control unit 226.

Thread scheduler 224 also manages execution of threads. Thread scheduler 224 fetches instructions for each thread from an instruction cache 230, decodes each instruction if necessary, and performs flow control for the thread. Thread scheduler 224 selects active threads for execution, checks for read/write port conflict among the selected threads and, if there is no conflict, sends instructions for one thread to execution units 234, and sends instructions for another thread to load control unit 226. Thread scheduler 224 maintains a program/instruction counter for each thread and updates this counter as instructions are executed or program flow is altered. Thread scheduler 224 also issues requests to fetch for missing instructions from instruction cache 230 and removes threads that are completed.

In one aspect, thread scheduler 224 interacts with a master engine 220. In this aspect, thread scheduler 224 may delegate certain responsibilities to master engine 220. In one aspect, thread scheduler 224 may decode instructions for execution, or may maintain the program/instruction counter for each thread and update this counter as instructions are executed. In one aspect, master engine 220 sets up state for instruction execution, and may also control the state update sequence during instruction execution.

Instruction cache 230 stores instructions for the threads. These instructions indicate specific operations to be performed for each thread. Each operation may be, for example, an arithmetic operation, an elementary function, a memory access operation, or another form of instruction. Instruction cache 230 may be loaded with instructions from cache memory system 210 or main memory 204 (FIG. 2A), as needed, via load control unit 226. These instructions are binary instructions that have been compiled from graphics application code, according to one aspect. Each binary instruction indicates a data precision used for its execution within shader processor 206. For example, an instruction type associated with the instruction may indicate whether the instruction is a full-precision instruction or a half-precision instruction. Or, a particular flag or field within the instruction may indicate whether it is a full-precision or a half-precision instruction, according to one exemplary aspect. Thread scheduler 224 may be capable of decoding instructions and determining a data precision for each instruction (such as full- or half-precision). Thread scheduler 224 can then route each instruction to an execution unit that is capable of executing the instruction with the indicated data precision. This execu-

tion unit loads any graphics data needed for instruction execution from a constant buffer 232 or register banks 242, which are described in more detail below.

In the aspect shown in FIG. 2B, execution units 234 includes one or more full-precision ALU's (Arithmetic Logic Units) 236, one or more half-precision ALU's 240, and an elementary functional unit 238 that executes transcendental elementary operations. ALU's 236 and 240 may include one or more floating point units, which enable floating computations, and/or one or more integer logic units, which enable integer and logic operations. When necessary, execution units 234 load in data, such as graphics data, from constant buffer 232 or from register banks 242 during instruction execution. Both the full-precision ALU's 236 and the half-precision ALU's 240 are capable of performing arithmetic operations (such as addition, subtraction, multiplication, multiply and accumulate, etc.) and also logical operations (such as AND, OR, XOR, etc.). Each ALU unit may comprise a single quad ALU or four scalar ALU's, according to one aspect. When four scalar ALU's are used, attributes for four pixels may be processed in parallel by the ALU's. A quad ALU may be used to process four attributes for a pixel or a vertex in parallel. However, full-precision ALU's 236 execute instructions using full-precision calculations, while half-precision ALU's 240 execute instructions using half-precision calculations.

Elementary functional unit 238 can compute transcendental elementary functions such as sine, cosine, reciprocal, logarithm, exponential, square root, or reciprocal square root, which are widely used in shader instructions. Elementary functional unit 238 may improve shader performance by computing elementary functions in much less time than the time required to perform polynomial approximations of the elementary functions using simple instructions. Elementary functional unit 238 may be capable of executing instructions with full precision, but also may be capable of converting calculation results to a half-precision format as well, according to one aspect of this disclosure.

Load control unit 226, which is part of controller 225 in the exemplary aspect shown in FIG. 2B, controls the flow of data and instructions for various components within shader processor 206. In one aspect, load control unit 226 may evict excess internal data of shader processor 206 to external memory (e.g., cache memory system 210), and may fetch external resources such as instruction, buffer, or texture data from texture engine 208 and/or cache memory system 210. Load control unit 226 interfaces with cache memory system 210 and loads instruction cache 230, constant buffer 232 (which may store uniform data used during instruction execution for graphics applications 202A and/or 202B), and register banks 242 with data and instructions from cache memory system 210. Load control unit 226 also may provide output data from register banks 242 to cache memory system 210. Register banks 242 may receive the output data from one or more execution units 234, and can be shared amongst execution units 234. Load control unit 226 also interfaces with texture engine 208. In certain cases, texture engine 208 may provide data (such as texel data) to shader processor 206 via load control unit 226, and, in certain cases, load control unit 226 may provide data (such as texture coordinate data) and/or instructions (such as a sampler ID instruction) to texture engine 208.

In the example of FIG. 2B, load control unit 226 also includes a precision converter 228. Because the data read into or written out of load control unit 226 may have different data precisions (e.g., full precision, half precision), load control unit 226 may need to convert certain data to a different data precision level before routing it to a different component

(such as to register banks 242, or to cache memory system 210). Precision converter 228 manages such data conversion within load control unit 226.

In one aspect, precision converter 228 operates to convert graphics data from one precision level to another precision level upon execution, by shader processor 206, of a received conversion instruction. When executed, the conversion instruction converts graphics data associated with a received graphics instruction to an indicated data precision. For example, the conversion instruction may convert data in a half-precision format to a full-precision format, or vice versa.

Constant buffer 232 may store constant values that are used by execution units 234 during instruction execution. Register banks 242 store temporary results as well as final results from execution units 234 for executed threads. Register banks 242 include one or more full-precision register banks 244 and one or more half-precision register banks 246. Final execution results can be read from register banks 242 by load control unit 226. In addition, a distributor 248 may also receive the final results for the executed threads from register banks 242 and distribute these results to at least one of graphics vertex application 202B and graphics pixel application 202A.

Graphics applications, such as applications 202A and 202B, may require processing of data using different precision levels. For example, in one aspect, graphics vertex application 202B processes vertex data using full-precision data formats, while graphics pixel application 202A processes pixel data using half-precision formats. In one aspect, graphics pixel application 202A processes certain information using half-precision format, yet processes other information using full-precision format. During execution of threads from graphics vertex application 202B and graphics pixel application 202A, shader processor 206 receives and processes instructions from instruction cache 230 that use different data precision levels for execution.

Thus, in the aspect shown in FIG. 2B, thread scheduler 224 identifies a data precision indicated or associated with a given instruction loaded out of instruction cache 230, and routes the instruction to an appropriate execution unit. For example, if the instruction is decoded as a full-precision instruction (such as through indication by the instruction type or a field/header contained within the instruction), thread scheduler 224 is capable of routing the instruction to one of the full-precision ALU's 236 for execution. Execution results from full precision ALU's 236 may be stored in one or more of the full-precision register banks 244 and provided back to the graphics application (such as graphics vertex application 202B) via distributor 248. If, however, an instruction from the instruction cache 230 is decoded by thread scheduler 224 as a half-precision instruction, thread scheduler 224 is capable of routing the instruction to one of the half-precision ALU's 240 for execution. Execution results from half-precision ALU's 240 may be stored in one or more of the half-precision register banks 246 and provided back to the graphics application (such as graphics pixel application 202A) via distributor 248.

FIG. 2C is a block diagram illustrating further details of the execution units 234 and register banks 242 shown in FIG. 2B, according to one aspect. As described previously, execution units 234 include various different types of execution units. In the example of FIG. 2C, execution units 234 includes one or more full-precision ALU's 236A-236N, one or more half-precision ALU's 240A-240N, and one or more elementary functional units 238. Each full-precision ALU 236A-236N is capable of using data to execute instructions using full-precision computations. Input data used during instruction execution may be retrieved from one or more of full-precision register banks 244A-244N (within register banks 242). In

addition, computation results generated during instruction execution by full-precision ALU's 236A-236N may be stored within one or more of full-precision register banks 244A-244N.

Similarly, each half-precision ALU 240A-240N is capable of using data to execute instructions using half-precision computations. Input data used during instruction execution may be retrieved from one or more of half-precision register banks 246A-246N. In addition, computation results generated during instruction execution by half-precision ALU's 240A-240N may be stored within one or more of half-precision register banks 246A-246N.

As described previously, elementary functional unit 238 is capable of executing full-precision instructions, but storing results in half-precision format. In one aspect, elementary functional unit 238 is capable of storing result data in either full- or half-precision format. As a result, elementary functional unit 238 is communicatively coupled to full-precision register banks 244A-244N, and is also communicatively coupled to half-precision register banks 246A-246N. Elementary functional unit 238 may both retrieve intermediate data from and store final result data to any of the registers within register banks 242, according to one aspect.

In addition, elementary functional unit 238 includes a precision converter 239. In those instances in which elementary functional unit 238 converts between full- and half-precision data formats, it may use precision converter 239 to perform the conversion. For example, unit 238 may load input graphics data from half-precision register banks 246A and use the data to execute a full-precision instruction. Precision converter 239 may convert the input data from a half-precision format to a full-precision format. Unit 238 may then use the converted data to execute the full-precision instruction. If the result data is to be stored back into half-precision register bank 246A, precision converter 239 may convert the result data from a full-precision to a half-precision format, such that it may be stored in half-precision register bank 246A. Alternatively, if the result data is to be stored in one of full-precision register banks 244A-244N, the result data in full-precision format may be directly stored in one of these registers.

Thread scheduler 224 (FIG. 2B) is capable of causing a binary instruction to be loaded from instruction cache 230 and executed in one of execution units 234 based upon the data precision associated with the instruction. For example, thread scheduler 224 may route full-precision instructions to one or more of full-precision ALU's 236A-236N, and may route half-precision instructions to one or more of half-precision ALU's 240A-240N. Thread scheduler 224 may also route elementary instructions to elementary functional unit 238 for execution. Result data can be stored in corresponding registers within register banks 242. In one aspect, data transitions between full-precision ALU's 236A-236N, elementary functional unit 238, and half-precision ALU's 240A-240N go through register banks 242.

In one aspect, each half-precision register bank 246A-246N contains less register storage space, and occupies less physical space on an integrated circuit, than each full-precision register bank 244A-244N. Thus, for example, half-precision register bank 246A contains less register storage space, and occupies a smaller physical space, than full-precision register bank 244A. In one aspect, one full-precision register bank (such as bank 244A) may contain substantially the same amount of register space, and occupy substantially the same amount of physical space, as two half-precision register banks (such as banks 246A and 246B combined).

Similarly, each full-precision ALU 236A-236N may occupy more physical space within an integrated circuit than each half-precision ALU 240A-240N. In addition, each full-precision ALU 236A-236N typically may use more operating power than each half-precision ALU 240A-240N. As a consequence, in certain aspects, it may be desired to limit the number of full-precision ALU's and full-precision register banks, and increase the number of half-precision ALU's and half-precision register banks, that are used, so as to minimize integrated circuit size and reduce power consumption requirements. These aspects may be particularly appropriate or beneficial when shader processor 206 is part of a smaller computing device with certain power constraints, such as a mobile or wireless communication device (e.g., such as a mobile radiotelephone or wireless communication device handset), or a digital camera or video device.

Therefore, in one aspect, execution units 234 may include only one full-precision ALU 236A, and register banks 242 may include only one full-precision register bank 244A. In this aspect, execution units 234 may further include four half-precision ALU's 240A-240D, while register banks 242 may include four half-precision register banks 246A-246D. As a result, execution units 234 may be capable of executing at least one half-precision instruction and one full-precision instruction in parallel. For example, the four half-precision ALU's 240A-240D may execute instructions for attributes of four pixels at a time. Because only one full precision ALU 236A is used, ALU 236A is capable of executing an instruction for one vertex at a time, according to one aspect. As a result, shader processor 206 need not utilize a vertex packing buffer to pack data for multiple vertexes, according to one aspect. In this case, vector-based attribute data for a vertex may be directly processed without having to convert the data to scalar format.

In another aspect, execution units 234 may include four full-precision ALU's 236A-236D, and register banks 242 may include four full-precision register banks 244A-244D. In this aspect, execution units 234 may further include eight half-precision ALU's 240A-240H, while register banks 242 may include eight half-precision register banks 246A-246H. As a result, execution units 234 are capable of executing, for example, two half-precision instructions on two quads and one full-precision instruction on one quad in parallel. Each quad, or thread, is a group of four pixels or four vertexes.

In another aspect, execution units 234 may include four full-precision ALU's 236A-236D, and register banks 242 may include four full-precision register banks 244A-244D. In this aspect, execution units 234 further includes four half-precision ALU's 240A-240H, while register banks 242 includes four half-precision register banks 246A-246H. Various other combinations of full-precision ALU's 236A-236N, full-precision register banks 244A-244N, half-precision ALU's 240A-240N, and half-precision register banks 246A-246N may be used.

In one aspect, shader processor 206 may be capable of using thread scheduler 224 to selectively power down, or disable, one or more of full-precision ALU's 236A-236N and one or more of full-precision register banks 244A-244N. In this aspect, although shader processor 206 includes various full-precision components (such as full-precision ALU's 236A-236N and full-precision register banks 244A-244N) within one or more integrated circuits, it may save, or reduce, power consumption by selectively powering down, or disabling, one or more of these full-precision components when they are not being used. For example, in certain scenarios, shader processor 206 may determine that one or more of these components are not being used, given that various binary

instructions that are loaded are to be executed by one or more of half-precision ALU's 240A-240N. Thus, in these types of scenarios, shader processor 206 may selectively power down, or disable, one or more of the full-precision components for power savings. In this manner, shader processor 206 may selectively power down or disable one or more full-precision components on a dynamic basis as a function of the types and numbers of instructions being processed at a given time.

In one aspect, shader processor 206 may also be capable of using thread scheduler 224 to selectively power down, or disable, one or more of half-precision ALU's 240A-240N and one or more of half-precision register banks 246A-246N. In this aspect, shader processor 206 may save, or reduce, power consumption by selectively powering down, or disabling, one or more of these half-precision components when they are not being used or not needed.

Shader processor 206 may provide various benefits and advantages. For example, shader processor 206 may provide a highly flexible and adaptive interface to satisfy different requirements for execution of mixed-precision instructions, such as full-precision and half-precision instructions. Shader processor 206 may significantly reduce power consumption by avoiding unnecessary precision promotion during execution of mixed-precision instructions. (Precision promotion may occur when shader processor 206 dynamically converts data from a lower precision format, such as a half-precision format, to a higher-precision format, such as a full-precision format. Precision promotion can require additional circuitry within shader processor 206, and also may cause shader core processes to expend additional clock cycles.) Because thread scheduler 224 is capable of recognizing data precisions associated with binary instructions loaded from instruction cache 230, thread scheduler 224 is capable of routing the instruction to an appropriate execution unit within execution units 234 for execution, such as full-precision ALU 236A or half-precision ALU 240A.

Shader processor 206 also may reduce overall register file size in register banks 242 and ALU size in execution units 234 by utilizing fewer full-precision components and by instead utilizing more half-precision components (e.g., ALU's and register banks). In addition, shader processor 206 may increase overall system performance by increasing processing capacity.

In view of the various potential benefits related to lower power consumption and increased performance, shader processor 206 may be used in various different types of systems or devices, such as wireless communications devices, digital camera devices, video recording or display devices, video game devices, or other graphics and multimedia devices. Such devices may include a display to present graphics content generated using shader processor 206. In one aspect, the precision flexibility offered by shader processor 206 allows it to be used with various devices, including multimedia devices, which may provide lower-precision calculations or have lower power requirements than certain other graphics applications.

FIG. 3 is a flow diagram illustrating an exemplary method that may be performed by the shader processor 206 shown in FIGS. 2A-2B, according to one aspect. In this aspect, the exemplary method includes acts 300, 302, 303, 306, 308, 310, and 312, and also includes a decision point 304.

In act 300, shader processor 206 receives a binary graphics instruction and an indication of a data precision for execution of the instruction. For example, as previously described, thread scheduler 224 may load the instruction from instruction cache 230 (FIG. 2B). In one aspect, decoding of the instruction, by thread scheduler 224, provides information as

to the data precision for execution of the instruction. For example, the instruction may be a full-precision or a half-precision instruction.

In act 302, shader processor 206 receives graphics data associated with the binary instruction. For example, sequencer 222 may receive vertex data from graphics vertex application 202B, and/or may receive pixel data from graphics pixel application 202A. In certain scenarios, load control unit 226 may also load graphics data associated with the instruction from cache memory system 210. In act 303, shader processor 206 further receives a conversion instruction that, if executed, converts the graphics data associated with the binary instruction to the indicated data precision.

At decision point 304, shader processor 206 determines whether the instruction is a full-precision or a half-precision instruction. As noted above, in one aspect, thread scheduler 224 may decode the instruction and determine whether it is a full-precision or half-precision instruction.

If the instruction is a full-precision instruction, shader processor 206, in act 306, converts, if necessary, any received graphics data from half- to full-precision format. In certain cases, the received graphics data, as stored in cache memory system 210 or as processed from graphics application 202A or 202B, may have a half-precision format. In this case, the graphics data is converted to a full-precision format so that it may be used during execution of the full-precision instruction. In one aspect, precision converter 228 of load control unit 226 may manage data format conversion when the received conversion instruction is executed by shader processor 206. In act 308, shader processor 206 selects a full-precision unit, such as unit 236A (FIG. 2C), to execute the binary instruction using the graphics data.

If, however, the instruction is a half-precision instruction, shader processor, in act 310, converts, if necessary, any data from full- to half-precision format. In one aspect, precision converter 228 may manage data format conversion when the received conversion instruction is executed by shader processor 206. In act 312, shader processor 206 then selects a half-precision unit, such as unit 240A (FIG. 2C) to execute the binary instruction using the graphics data.

FIG. 4 is a block diagram illustrating a compiler 402 that may be used to generate instructions to be executed by streaming processor 106 shown in FIG. 1 or by shader processor 206 shown in FIGS. 2A-2B, according to one aspect. In one example aspect, compiler 402 is used to generate instructions to be executed by shader processor 206. In this aspect, application developers may use compiler 402 to generate binary instructions (code) for execution by shader processor 206. Shader processor 206 is part of graphics device 200 (FIG. 2A). Application developers may have access to an application development platform for use with graphics device 200, and may create application-level software for graphics pixel application 202A and/or graphics vertex application 202B. Such application-level software includes graphics application instructions 400 shown in FIG. 4. Graphics application instructions 400 may include instructions written by high-level shading languages, compliant with or translatable to DirectXTM, OpenGLTM, OpenVGTM, or other languages. In one aspect, these shading languages define one or more standard API's that may be used for developing programming code to perform graphics operations.

Compiler 402 may be supported, at least in part, by compiler software executed by a processor to receive and process source code instructions and compile such instructions to produce compiled instructions (e.g., in the form of binary, executable machine instructions). Accordingly, compiler 402 may be formed by one or more processors executing com-

puter-readable instructions associated with the compiler software. In one aspect, these one or more processors may be part of, or implemented in, the application development platform used by application developers. The compiled instructions may be stored on a computer-readable data storage medium for retrieval and execution by one or more processors, such as streaming processor 106 or shader processor 206. For example, the disclosure contemplates a computer-readable data storage medium including one or more first executable instructions, one or more second executable instructions, and one or more third executable instructions.

The first executable instructions, when executed by a processor, may support one or more functions of a graphics application. In addition, each of the first executable instructions may indicate a first data precision level for its execution. The second executable instructions, when executed by a processor, may support one or more functions of the graphics application. In addition, each of the second executable instructions may indicate a second data precision level different from the first data precision level for its execution. The third executable instructions, when executed by the processor, may also support one or more functions of the graphics application, wherein each of the third executable instructions converts graphics data from the second data precision level to the first data precision level when the one or more first executable instructions are executed.

Compiler 402 may be capable of compiling graphics application instructions 400 into binary graphics instructions 404, which are then capable of being executed by shader processor 206. Shader processor 206 may retrieve such instructions from a data storage media such as a memory or data storage device, and execute these instructions to perform computations and other operations in support of a graphics application. Several of graphics applications instructions 400 may specify a particular data precision level for execution. For example, certain instructions may specify that they use full-precision or half-precision operations or calculations. Compiler 402 may be configured to apply rules 406 to analyze and parse graphics application instructions 400 during the compilation process and generate corresponding binary instructions graphics 404 that indicate data precision levels for execution of instructions 404.

Thus, if one of graphics application instructions 400 specifies a full-precision operation or calculation, rules 406 of compiler 402 may generate one or more of binary instructions 404 that are full-precision instructions. If another one of graphics application instructions 400 specifies a half-precision operation or calculation, rules 406 generate one or more of binary instructions 404 that are half-precision instructions. In one aspect, binary instructions 404 each may include an 'opcode' indicating whether the instruction is a full-precision or a half-precision instruction. In one aspect, binary instructions 404 each may indicate a data precision for execution of the instruction using information contained within another predefined field, flag, or header, of the instruction that may be decoded by shader processor 206. In one aspect, the data precision may be inferred based upon the type of instruction to be executed.

Compiler 402 also includes rules 408 that are capable of generating binary conversion instructions 410 that convert between different data precision levels. During compilation, these rules 408 of compiler 402 may determine that such conversion may be necessary during execution of binary instructions 404. For example, rules 408 may generate one or more instructions within conversion instructions 410 that convert data from a full-precision format to a half-precision format. This conversion may be required when shader pro-

cessor 206 executes half-precision instructions within graphics instructions 404. Rules 408 may also generate one or more instructions within conversion instructions 410 that convert data from a half-precision to a full-precision format, which may be required when shader processor 206 executes full-precision instructions within graphics instructions 404.

When rules 408 of compiler 402 generate conversion instructions 410, shader processor 206 may execute these conversion instructions 410 to manage data precision conversion during execution of corresponding graphics instructions 404, according to one aspect. In this aspect, execution of conversion instructions 410 manages such precision conversion, such that shader processor 206 need not necessarily use certain hardware conversion mechanisms to convert data from one precision level to another. Conversion instructions 410 may also allow more efficient data transfer to ALU's using different precision levels, such as to full-precision ALU's 236 and to half-precision ALU's 240.

The components and techniques described herein may be implemented in hardware, software, firmware, or any combination thereof. Any features described as modules or components may be implemented together in an integrated logic device or separately as discrete but interoperable logic devices. In various aspects, such components may be formed at least in part as one or more integrated circuit devices, which may be referred to collectively as an integrated circuit device, such as an integrated circuit chip or chipset. Such an integrated circuit device may be used in any of a variety of graphics applications and devices. In some aspects, for example, such components may form part of a mobile device, such as a wireless communication device handset.

If implemented in software, the techniques may be realized at least in part by a computer-readable medium comprising instructions that, when executed by one or more processors, performs one or more of the methods described above. The computer-readable medium may form part of a computer program product, which may include packaging materials. The computer-readable medium may comprise random access memory (RAM) such as synchronous dynamic random access memory (SDRAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, magnetic or optical data storage media.

The techniques additionally, or alternatively, may be realized at least in part by a computer-readable communication medium that carries or communicates code in the form of instructions or data structures and that can be accessed, read, and/or executed by one or more processors. Any connection may be properly termed a computer-readable medium. For example, if the software is transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. Combinations of the above should also be included within the scope of computer-readable media.

Any software that is utilized may be executed by one or more processors, such as one or more digital signal processors (DSP's), general purpose microprocessors, application specific integrated circuits (ASIC's), field-programmable gate arrays (FPGA's), or other equivalent integrated or discrete logic circuitry. Accordingly, the terms "processor" or "controller," as used herein, may refer to any of the foregoing structures or any other structure suitable for implementation of the techniques described herein. Hence, the disclosure also

contemplates any of a variety of integrated circuit devices that include circuitry to implement one or more of the techniques described in this disclosure. Such circuitry may be provided in a single integrated circuit chip device or in multiple, interoperable integrated circuit chip devices.

Various aspects of the disclosure have been described. These and other aspects are within the scope of the following claims.

The invention claimed is:

1. A method comprising:

receiving a graphics instruction for execution within a programmable streaming processor;

receiving an indication of a data precision for execution of the graphics instruction, wherein the indication of the data precision is contained within the graphics instruction, wherein the graphics instruction is a first executable instruction generated by a compiler that compiles graphics application instructions;

receiving a conversion instruction that, when executed by the programmable streaming processor, converts graphics data, associated with the graphics instruction, from a first data precision to converted graphics data having the indicated data precision, and wherein the conversion instruction is different than the graphics instruction, wherein the conversion instruction is generated by the compiler;

selecting one of a plurality of execution units within the processor based on the indicated data precision; and using the selected execution unit to execute the graphics instruction with the indicated data precision using the converted graphics data associated with the graphics instruction.

2. The method of claim 1, further comprising:

receiving the graphics data associated with the graphics instruction;

generating a computation result with the indicated data precision during execution of the graphics instruction by the selected execution unit; and

providing the computation result as output.

3. The method of claim 1, wherein selecting one of the plurality of execution units comprises:

selecting one of a first set of one or more execution units within the processor that each execute instructions with the first data precision using the graphics data when the indicated data precision is the first data precision; and

selecting one of a second set of one or more execution units within the processor that each execute instructions with a second data precision using the graphics data when the indicated data precision is the second data precision, the second data precision being different than the first data precision.

4. The method of claim 3, wherein the first data precision comprises a full data precision, and wherein the second data precision comprises a half data precision.

5. The method of claim 1, wherein the execution units include a first set of one or more execution units within the processor that each execute instructions with the first data precision using the graphics data, and further include a second set of one or more execution units within the processor that each execute instructions with a second data precision different than the first data precision using the graphics data.

6. The method of claim 5, wherein:

selecting one of the plurality of execution units within the processor based on the indicated data precision comprises selecting one of the execution units in the first set; and

using the selected execution unit to execute the graphics instruction comprises using the selected execution unit in the first set to execute the graphics instruction with the indicated data precision using the graphics data associated with the graphics instruction.

7. The method of claim 6, further comprising:
receiving a second graphics instruction for execution within the processor;

receiving an indication of the second data precision for execution of the second graphics instruction;

receiving a second conversion instruction that, when executed by the processor, converts graphics data associated with the second graphics instruction to the indicated second data precision, the second conversion instruction being different than the second graphics instruction;

selecting one of the execution units in the second set based on the indicated second data precision; and

using the selected execution unit in the second set to execute the second graphics instruction with the indicated second data precision using the graphics data associated with the second graphics instruction.

8. The method of claim 1, wherein receiving the indication of the data precision for execution of the graphics instruction comprises decoding the graphics instruction to determine the data precision.

9. The method of claim 1, wherein the graphics data associated with the graphics instruction comprises at least one of vertex graphics data and pixel graphics data.

10. A non-transitory computer-readable storage medium comprising instructions for causing a programmable streaming processor to:

receive a graphics instruction for execution within the programmable streaming processor;

receive an indication of a data precision for execution of the graphics instruction, wherein the indication of the data precision is contained within the graphics instruction, wherein the graphics instruction is a first executable instruction generated by a compiler that compiles graphics application instructions;

receive a conversion instruction that, when executed by the processor, converts graphics data, associated with the graphics instruction, from a first data precision to converted graphics data having the indicated data precision, and wherein the conversion instruction is different than the graphics instruction, wherein the conversion instruction is generated by the compiler;

select one of a plurality of execution units within the processor based on the indicated data precision; and

use the selected execution unit to execute the graphics instruction with the indicated data precision using the converted graphics data associated with the graphics instruction.

11. The non-transitory computer-readable storage medium of claim 10, further comprising instructions for causing the processor to:

receive the graphics data associated with the graphics instruction;

generate a computation result with the indicated data precision during execution of the graphics instruction by the selected execution unit; and

provide the computation result as output.

12. The non-transitory computer-readable storage medium of claim 10, wherein the instructions for causing the processor to select one of the plurality of execution units comprise instructions for causing the processor to:

select one of a first set of one or more execution units within the processor that each execute instructions with the first data precision using the graphics data when the indicated data precision is the first data precision; and

select one of a second set of one or more execution units within the processor that each execute instructions with a second data precision using the graphics data when the indicated data precision is the second data precision, the second data precision being different than the first data precision.

13. The non-transitory computer-readable storage medium of claim 12, wherein the first data precision comprises a full data precision, and wherein the second data precision comprises a half data precision.

14. The non-transitory computer-readable storage medium of claim 10, wherein the execution units include a first set of one or more execution units within the processor that each execute instructions with the first data precision using the graphics data, and further include a second set of one or more execution units within the processor that each execute instructions with a second data precision different than the first data precision using the graphics data.

15. The non-transitory computer-readable storage medium of claim 14, wherein:

the instructions for causing the processor to select one of the plurality of execution units within the processor based on the indicated data precision comprise instructions for causing the processor to select one of the execution units in the first set; and

the instructions for causing the processor to use the selected execution unit to execute the instruction comprise instructions for causing the processor to use the selected execution unit in the first set to execute the graphics instruction with the indicated data precision using the graphics data associated with the graphics instruction.

16. The non-transitory computer-readable storage medium of claim 15, further comprising instructions for causing the processor to:

receive a second graphics instruction for execution within the processor;

receive an indication of the second data precision for execution of the second graphics instruction;

receive a second conversion instruction that, when executed by the processor, converts graphics data associated with the second graphics instruction to the indicated second data precision, the second conversion instruction being different than the second graphics instruction;

select one of the execution units in the second set based on the indicated second data precision; and

use the selected execution unit in the second set to execute the second graphics instruction with the indicated second data precision using the graphics data associated with the second graphics instruction.

17. The non-transitory computer-readable storage medium of claim 10, wherein the instructions for causing the processor to receive the indication of the data precision for execution of the graphics instruction comprise instructions for causing the processor to decode the graphics instruction to determine the data precision.

18. The non-transitory computer-readable storage medium of claim 10, wherein the graphics data associated with the graphics instruction comprises at least one of vertex graphics data and pixel graphics data.

19. A device comprising:
 a controller configured to receive a graphics instruction for execution within a programmable streaming processor, wherein the indication of the data precision is contained within the graphics instruction and wherein the graphics instruction is a first executable instruction generated by a compiler that compiles graphics application instructions, to receive an indication of a data precision for execution of the graphics instruction, and to receive a conversion instruction that, when executed by the programmable streaming processor, converts graphics data associated, with the graphics instruction, from a first data precision to converted graphics data having a second data precision, wherein the conversion instruction is different than the graphics instruction and wherein the conversion instruction is generated by the compiler; and a plurality of execution units within the processor, wherein the controller is configured to select one of the execution units based on the indicated data precision and cause the selected execution unit to execute the graphics instruction with the indicated data precision using the converted graphics data associated with the graphics instruction.

20. The device of claim 19, wherein the plurality of execution units includes a first execution unit configured to execute instructions with the indicated data precision and a second execution unit configured to execute instructions with a second data precision that is different from the indicated data precision, and wherein the controller is configured to select the first execution unit to execute the graphics instruction with the indicated data precision using the graphics data.

21. The device of claim 19, wherein the plurality of execution units includes one or more full-precision execution units and at least four half-precision execution units.

22. The device of claim 21, wherein when the indicated data precision for execution of the graphics instruction comprises a full precision, the controller is configured to select one of the full-precision execution units to execute the graphics instruction using the graphics data.

23. The device of claim 21, wherein when the indicated data precision for execution of the graphics instruction comprises a half precision, the controller is configured to select one of the half-precision execution units to execute the graphics instruction using the graphics data.

24. The device of claim 21, further comprising:
 at least one full-precision register bank to store computation results when the at least one full-precision execution unit executes instructions; and

at least four half-precision register banks to store computation results when the at least four half-precision execution units execute instructions.

25. The device of claim 19, wherein the plurality of execution units includes at least one full-precision execution unit and at least one half-precision execution unit, and wherein when the indicated data precision for execution of the graphics instruction comprises a half precision, the controller is configured to shut down power to the at least one full-precision execution unit and cause the at least one half-precision execution unit to execute the graphics instruction using the graphics data.

26. The device of claim 19, wherein the processor comprises a shader processor.

27. The device of claim 19, wherein the device comprises a wireless communication device handset.

28. The device of claim 19, wherein the device comprises one or more integrated circuit devices.

29. A device comprising:
 means for receiving a graphics instruction for execution within a programmable streaming processor;
 means for receiving an indication of a data precision for execution of the graphics instruction, wherein the indication of the data precision is contained within the graphics instruction, wherein the graphics instruction is a first executable instruction generated by a compiler that compiles graphics application instructions;

means for receiving a conversion instruction that, when executed by the programmable streaming processor, converts graphics data associated, with the graphics instruction, from a first data precision to converted graphics data having the indicated data precision, and wherein the conversion instruction is different than the graphics instruction, wherein the conversion instruction is generated by the compiler;

means for selecting one of a plurality of execution units within the processor based on the indicated data precision; and

means for using the selected execution unit to execute the graphics instruction with the indicated data precision using the converted graphics data associated with the graphics instruction.

30. The device of claim 29, further comprising:
 means for receiving the graphics data associated with the graphics instruction;

means for generating a computation result with the indicated data precision during execution of the graphics instruction by the selected execution unit; and

means for providing the computation result as output.

31. The device of claim 29, wherein the means for selecting one of the plurality of execution units comprises:

means for selecting one of a first set of one or more execution units within the processor that each execute instructions with the first data precision using the graphics data when the indicated data precision is the first data precision; and

means for selecting one of a second set of one or more execution units within the processor that each execute instructions with a second data precision using the graphics data when the indicated data precision is the second data precision, the second data precision being different than the first data precision.

32. The device of claim 31, wherein the first data precision comprises a full data precision, and wherein the second data precision comprises a half data precision.

33. The device of claim 29, wherein the execution units include a first set of one or more execution units within the processor that each execute instructions with the first data precision using the graphics data, and further include a second set of one or more execution units within the processor that each execute instructions with a second data precision different than the first data precision using the graphics data.

34. The device of claim 33, wherein:
 the means for selecting one of the plurality of execution units within the processor based on the indicated data precision comprises means for selecting one of the execution units in the first set; and

the means for using the selected execution unit to execute the graphics instruction comprises means for using the selected execution unit in the first set to execute the graphics instruction with the indicated data precision using the graphics data associated with the graphics instruction.

35. The device of claim 34, further comprising:
 means for receiving a second graphics instruction for execution within the processor;
 means for receiving an indication of the second data precision for execution of the second graphics instruction;
 means for receiving a second conversion instruction that, when executed by the processor, converts graphics data associated with the second graphics instruction to the indicated second data precision, the second conversion instruction being different than the second graphics instruction;
 means for selecting one of the execution units in the second set based on the indicated second data precision; and
 means for using the selected execution unit in the second set to execute the second graphics instruction with the indicated second data precision using the graphics data associated with the second graphics instruction.

36. The device of claim 29, wherein the means for receiving the indication of the data precision for execution of the graphics instruction comprises means for decoding the graphics instruction to determine the data precision.

37. The device of claim 29, wherein the graphics data associated with the graphics instruction comprises at least one of vertex graphics data and pixel graphics data.

38. A device comprising:
 a programmable streaming processor; and
 at least one memory module coupled to the programmable streaming processor,
 wherein the programmable streaming processor comprises:

a controller configured to receive a graphics instruction for execution from the at least one memory module, to receive an indication of a data precision for execution of the graphics instruction, wherein the indication of the data precision is contained within the graphics instruction and wherein the graphics instruction is a first executable instruction generated by a compiler that compiles graphics application instructions, and to receive a conversion instruction that, when executed by the processor, converts graphics data, associated with the graphics instruction, to converted graphics data, wherein the graphics data has a first data precision and the converted graphics data has the indicated data precision, and wherein the conversion instruction is different than the graphics instruction and wherein the conversion instruction is generated by the compiler; and

a plurality of execution units that are configured to execute instructions,
 wherein the controller is configured to select one of the execution units based on the indicated data precision and cause the selected execution unit to execute the graphics instruction with the indicated data precision using the converted graphics data associated with the graphics instruction.

39. The device of claim 38, further comprising at least one graphics engine coupled to the processor.

40. The device of claim 38, wherein the plurality of execution units includes a first execution unit configured to execute instructions with the indicated data precision and a second execution unit configured to execute instructions with a second data precision that is different from the indicated data precision, and wherein the controller is configured to select the first execution unit to execute the graphics instruction with the indicated data precision using the graphics data.

41. The device of claim 38, wherein the plurality of execution units includes one or more full-precision execution units and at least four half-precision execution units.

42. The device of claim 41, wherein when the indicated data precision for execution of the graphics instruction comprises a full precision, the controller is configured to select one of the full-precision execution units to execute the graphics instruction using the graphics data.

43. The device of claim 41, wherein when the indicated data precision for execution of the graphics instruction comprises a half precision, the controller is configured to select one of the half-precision execution units to execute the graphics instruction using the graphics data.

44. The device of claim 41, wherein the processor further comprises:

at least one full-precision register bank to store computation results when the at least one full-precision execution unit executes instructions; and

at least four half-precision register banks to store computation results when the at least four half-precision execution units execute instructions.

45. The device of claim 38, wherein the plurality of execution units includes at least one full-precision execution unit and at least one half-precision execution unit, and wherein when the indicated data precision for execution of the graphics instruction comprises a half precision, the controller is configured to shut down power to the at least one full-precision execution unit and cause the at least one half-precision execution unit to execute the graphics instruction using the graphics data.

46. The device of claim 38, wherein the processor comprises a shader processor.

47. The device of claim 38, wherein the device comprises a wireless communication device handset.

48. The device of claim 38, wherein the device comprises one or more integrated circuit devices.

49. A method, comprising:

analyzing, by a compiler executed by a processor, a plurality of application instructions for a graphics application; for each application instruction that specifies a first data precision level for its execution, generating, by the compiler, one or more corresponding compiled instructions that each indicate the first data precision level for its execution, wherein the first precision level comprises a full data precision level; and

generating, by the compiler, one or more conversion instructions to convert graphics data from a second, different data precision level to the first data precision level when the one or more compiled instructions are executed.

50. The method of claim 49, wherein the second data precision level comprises a half data precision level.

51. The method of claim 49, wherein generating the one or more compiled instructions comprises generating one or more compiled instructions that each indicate a full data precision level when a corresponding application instruction specifies the full data precision level for its execution.

52. The method of claim 49, wherein generating the one or more compiled instructions comprises generating one or more compiled instructions that each indicate a half data precision level when a corresponding application instruction specifies the half data precision level for its execution.

53. The method of claim 49, wherein the one or more compiled instructions each include a predefined field that includes information indicating the first data precision level when the corresponding application instruction specifies the first data precision level for its execution.

54. The method of claim 49, further comprising storing the one or more compiled instructions in memory for subsequent execution.

55. A non-transitory computer-readable storage medium comprising instructions for causing a processor to:

analyze, by a compiler executed by the processor, a plurality of application instructions for a graphics application; for each application instruction that specifies a first data precision level for its execution, generate, by the compiler, one or more corresponding compiled instructions that each indicate the first data precision level for its execution, wherein the first precision level comprises a full data precision level; and

generate, by the compiler, one or more conversion instructions to convert graphics data from a second, different data precision level to the first data precision level when the one or more compiled instructions are executed.

56. The non-transitory computer-readable storage medium of claim 55, wherein the second data precision level comprises a half data precision level.

57. The non-transitory computer-readable storage medium of claim 55, wherein the instructions for causing the processor to generate the one or more compiled instructions comprise instructions for causing the processor to generate the one or more compiled instructions that each indicate a full data precision level when a corresponding application instruction specifies the full data precision level for its execution.

58. The non-transitory computer-readable storage medium of claim 55, wherein the instructions for causing the processor to generate the one or more compiled instructions comprise instructions for causing the processor to generate the one or more compiled instructions that each indicate a half data precision level when a corresponding application instruction specifies the half data precision level for its execution.

59. The non-transitory computer-readable storage medium of claim 55, wherein the one or more compiled instructions each include a predefined field that includes information indicating the first data precision level when the corresponding application instruction specifies the first data precision level for its execution.

60. The non-transitory computer-readable storage medium of claim 55, further comprising instructions for causing the processor to store the one or more compiled instructions in memory for subsequent execution.

61. An apparatus comprising:

means for analyzing a plurality of graphics application instructions;

for each graphics application instruction that specifies a first data precision level for its execution, means for generating one or more corresponding compiled instructions that each indicate the first data precision level for its execution, wherein the first precision level comprises a full data precision level; and

means for generating one or more conversion instructions to convert graphics data from a second, different data

precision level to the first data precision level when the one or more compiled instructions are executed.

62. The apparatus of claim 61, wherein the second data precision level comprises a half data precision level.

63. The apparatus of claim 61, wherein the means for generating the one or more compiled instructions comprises means for generating the one or more compiled instructions that each indicate a full data precision level when a corresponding graphics application instruction specifies the full data precision level for its execution.

64. The apparatus of claim 61, wherein the means for generating the one or more compiled instructions comprises means for generating the one or more compiled instructions that each indicate a half data precision level when a corresponding graphics application instruction specifies the half data precision level for its execution.

65. The apparatus of claim 61, wherein the one or more compiled instructions each include a predefined field that includes information indicating the first data precision level when the corresponding graphics application instruction specifies the first data precision level for its execution.

66. The apparatus of claim 61, further comprising means for storing the one or more compiled instructions in memory for subsequent execution.

67. A non-transitory computer-readable data storage medium comprising:

one or more first executable instructions generated by a compiler, wherein the one or more first executable instructions, when executed by a programmable streaming processor, support one or more functions of a graphics application, wherein each of the first executable instructions indicates a first data precision level for its execution;

one or more second executable instructions generated by a compiler, wherein the one or more second executable instructions, when executed by the programmable streaming processor, support one or more functions of the graphics application, wherein each of the second executable instructions indicates a second data precision level different from the first data precision level for its execution, wherein the first precision level comprises a full data precision level; and

one or more third executable instructions generated by a compiler, wherein the one or more third executable instructions, when executed by the programmable streaming processor, support one or more functions of the graphics application, wherein each of the third executable instructions converts graphics data from the second data precision level to the first data precision level when the one or more first executable instructions are executed by a programmable streaming processor.

68. The non-transitory computer-readable data storage medium of claim 67, and wherein the second data precision level comprises a half data precision level.

* * * * *

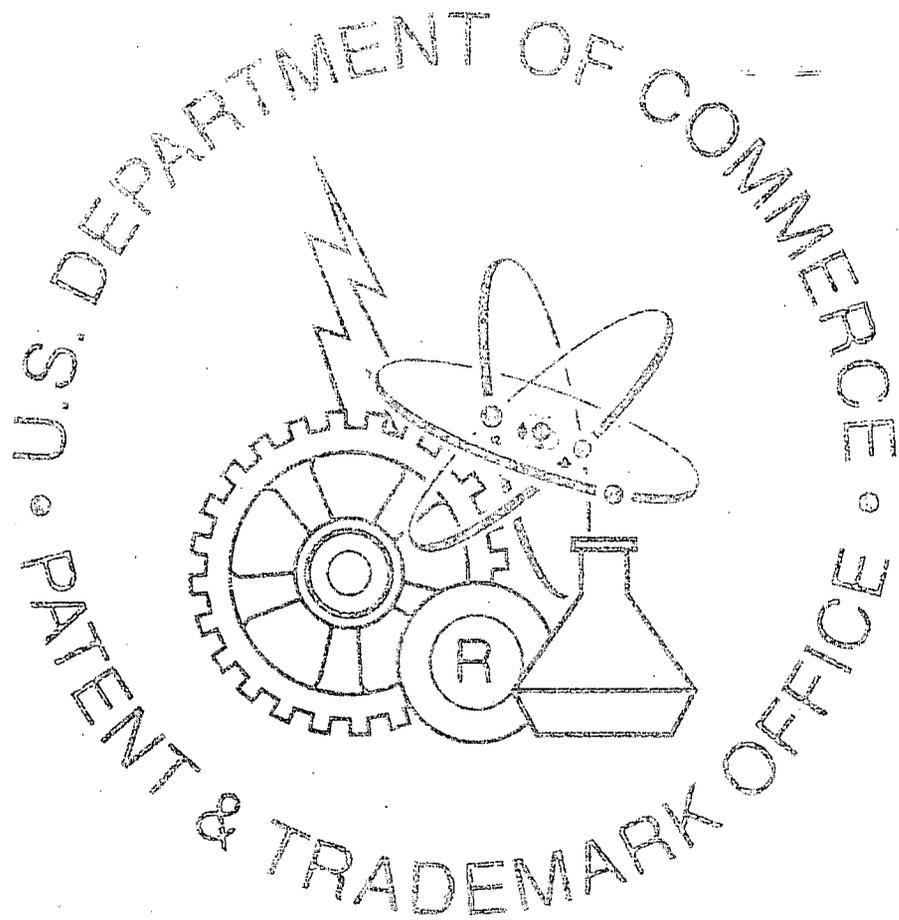


EXHIBIT B

United States
Marine et al.

8,698,558 B2
Apr. 15, 2014

U 7629935



THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

**UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office**

April 06, 2017

**THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM
THE RECORDS OF THIS OFFICE OF:**

**U.S. PATENT: 8,698,558
ISSUE DATE: April 15, 2014**

5,000,000	84	12,000
5,200,000	82	12,000
5,400,000	80	12,000
5,600,000	78	12,000
5,800,000	76	12,000
6,000,000	74	12,000
6,200,000	72	12,000
6,400,000	70	12,000
6,600,000	68	12,000
6,800,000	66	12,000
7,000,000	64	12,000
7,200,000	62	12,000
7,400,000	60	12,000
7,600,000	58	12,000
7,800,000	56	12,000
8,000,000	54	12,000
8,200,000	52	12,000
8,400,000	50	12,000
8,600,000	48	12,000
8,800,000	46	12,000
9,000,000	44	12,000
9,200,000	42	12,000
9,400,000	40	12,000
9,600,000	38	12,000
9,800,000	36	12,000
10,000,000	34	12,000

**By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office**



R. Glover
**R GLOVER
Certifying Officer**



US008698558B2

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

(10) **Patent No.:** **US 8,698,558 B2**
(45) **Date of Patent:** **Apr. 15, 2014**

(54) **LOW-VOLTAGE POWER-EFFICIENT ENVELOPE TRACKER**

(75) **Inventors:** **Lennart K Mathe, San Diego, CA (US); Thomas Domenick Marra, San Diego, CA (US); Todd R Sutton, Del Mar, CA (US)**

(73) **Assignee:** **QUALCOMM Incorporated, San Diego, CA (US)**

(*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 38 days.

(21) **Appl. No.:** **13/167,659**

(22) **Filed:** **Jun. 23, 2011**

(65) **Prior Publication Data**

US 2012/0326783 A1 Dec. 27, 2012

(51) **Int. Cl.**
H03F 3/217 (2006.01)

(52) **U.S. Cl.**
USPC 330/251; 330/136; 330/297

(58) **Field of Classification Search**
USPC 330/10, 136, 207 A, 251, 297
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,905,407 A *	5/1999	Midya	330/10
6,300,826 B1	10/2001	Mathe et al.	
6,661,217 B2	12/2003	Kimball et al.	
6,792,252 B2	9/2004	Kimball et al.	
6,838,931 B2 *	1/2005	Midya et al.	330/10
7,061,313 B2	6/2006	Kimball et al.	
7,068,984 B2	6/2006	Mathe et al.	
7,368,985 B2	5/2008	Kusunoki	

7,679,433 B1	3/2010	Li	
7,755,431 B2 *	7/2010	Sun	330/297
7,932,780 B2 *	4/2011	Elia	330/136
8,030,995 B2 *	10/2011	Okubo et al.	330/297
8,237,499 B2 *	8/2012	Chen et al.	330/136
2005/0046474 A1	3/2005	Matsumoto et al.	
2005/0215209 A1	9/2005	Tanabe et al.	
2008/0278136 A1	11/2008	Murtojarvi	
2010/0001793 A1	1/2010	Van Zeijl et al.	
2011/0095827 A1	4/2011	Tanaka et al.	
2012/0293253 A1 *	11/2012	Khlat et al.	330/127

OTHER PUBLICATIONS

Choi, et al., "Envelope Tracking Power Amplifier Robust to Battery Depletion," 2010 IEEE.
MTT-S International Microwave Symposium Digest (MTT), May 2010.

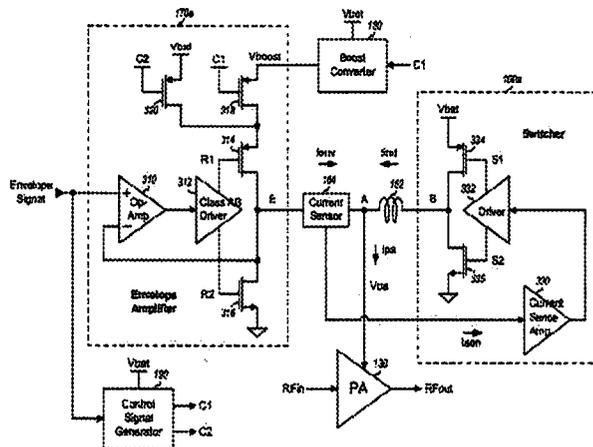
(Continued)

Primary Examiner — Khanh V Nguyen
(74) *Attorney, Agent, or Firm* — William M. Hooks

(57) **ABSTRACT**

Techniques for efficiently generating a power supply are described. In one design, an apparatus includes an envelope amplifier and a boost converter. The boost converter generates a boosted supply voltage having a higher voltage than a first supply voltage (e.g., a battery voltage). The envelope amplifier generates a second supply voltage based on an envelope signal and the boosted supply voltage (and also possibly the first supply voltage). A power amplifier operates based on the second supply voltage. In another design, an apparatus includes a switcher, an envelope amplifier, and a power amplifier. The switcher receives a first supply voltage and provides a first supply current. The envelope amplifier provides a second supply current based on an envelope signal. The power amplifier receives a total supply current including the first and second supply currents. In one design, the switcher detects the second supply current and adds an offset to generate a larger first supply current than without the offset.

20 Claims, 6 Drawing Sheets



(56)

References Cited

OTHER PUBLICATIONS

Choi, J et al., "A Polar Transmitter With CMOS Programmable Hysteretic-Controlled Hybrid Switching Supply Modulator for Multi standard Applications", IEEE Transactions on Microwave Theory and Techniques, IEEE Service Center, Piscataway, NJ, US, vol. 57, No. 7, Jul. 1, 2009, pp. 1675-1686, XP011258456.

Ertl, H et al., "Basic Considerations and Topologies of Switched-Mode Assisted Linear Power Amplifiers", IEEE Transactions on Industrial Electronics, IEEE Service Center, Piscataway, NJ, USA, vol. 44, No. 1, Feb. 1, 1997, XP011023224.

International Search Report and Written Opinion—PCT/US2012/043915—ISA/EPO—Nov. 26, 2012.

Kang D., et al., "A Multimode/Multiband Power Amplifier With a Boosted Supply Modulator", IEEE Transactions on Microwave Theory and Techniques, IEEE Service Center, Piscataway, NJ, US, vol. 58, No. 10, Oct. 1, 2010, pp. 2598-2608, XP011317521, ISSN: 0018-9480.

Kang, D et al., "LTE Power Amplifier for envelope tracking polar transmitters", Microwave Conference (EUMC), 2010, European,

IEEE, Piscataway, NJ, USA, Sep. 28, 2010, pp. 628-631, XP031786114.

Kim D., et al., "High efficiency and wideband envelope tracking power amplifier with sweet spot tracking", Radio Frequency Integrated Circuits Symposium (RFIC), 2010 IEEE, IEEE, Piscataway, NJ, USA, May 23, 2010, pp. 255-258, XP031684103, ISBN: 978-1-4244-6240-7.

Li, Y et al., "High Efficiency Wide Bandwidth Power Supplies for GSM and EDGE RF Power Amplifiers", Conference Proceedings/IEEE International Symposium on Circuits and Systems (ISCAS): May 23-26, 2005, International Conference Center, Kobe, Japan, IEEE Service Center, Piscataway, NJ, May 23, 2005, pp. 1314-1317, XP010815779.

Partial International Search Report—PCT/US2012/043915—International Search Authority European Patent Office Oct. 4, 2012.

Stauth, J.T., et al., "Optimum Bias Calculation for Parallel Hybrid Switching-Linear Regulators", Applied Power Electronics Conference, APEC 2007—Twenty Second Annual IEEE, IEEE, PI, Feb. 1, 2007, pp. 569-574, XP031085267.

* cited by examiner

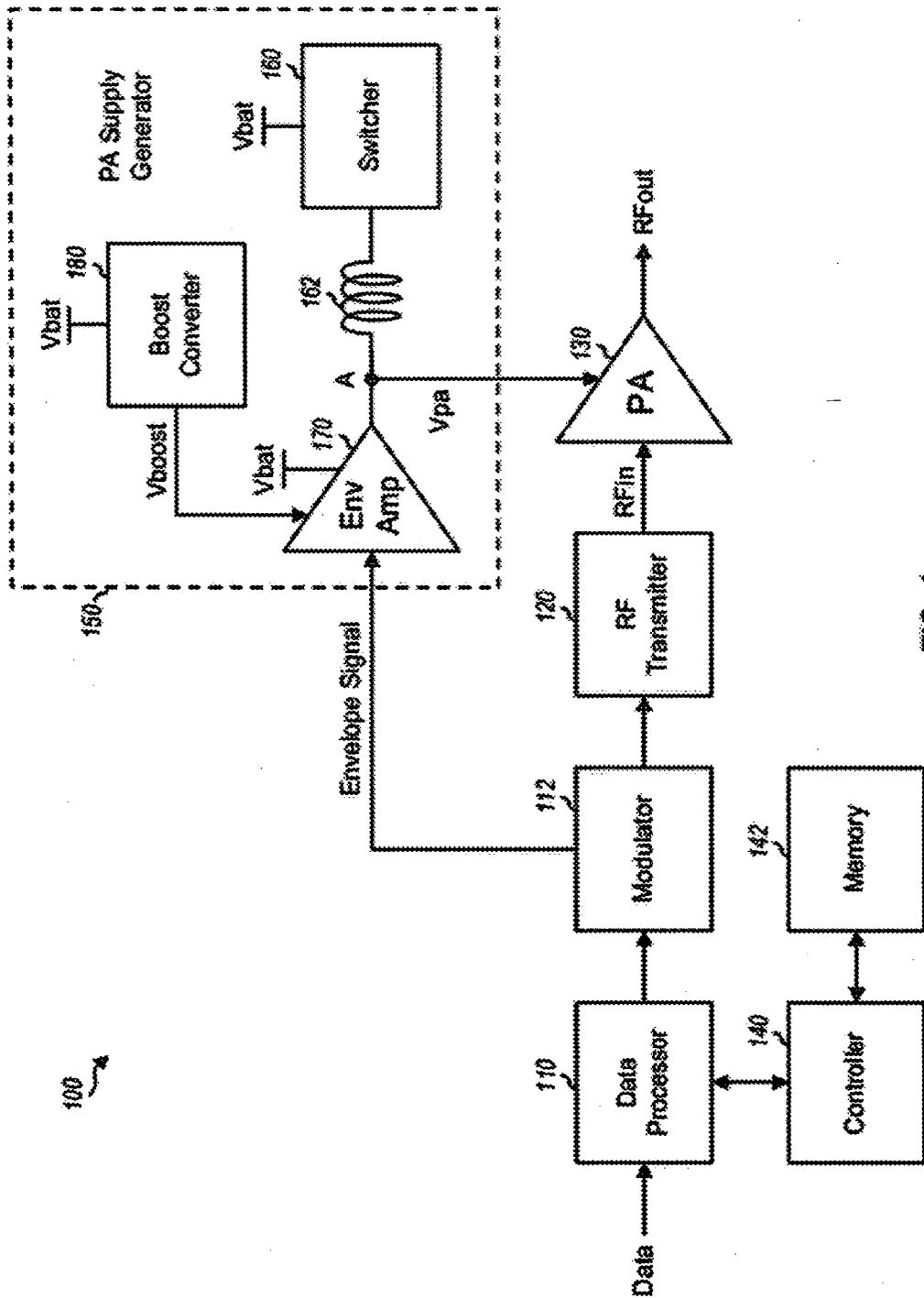
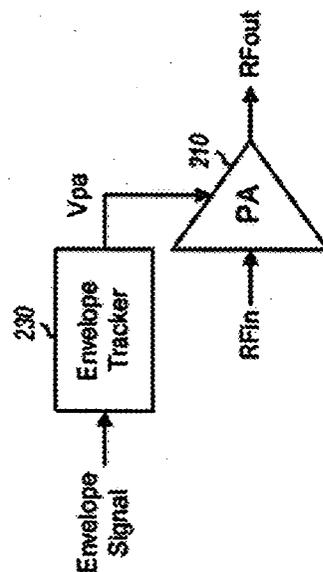
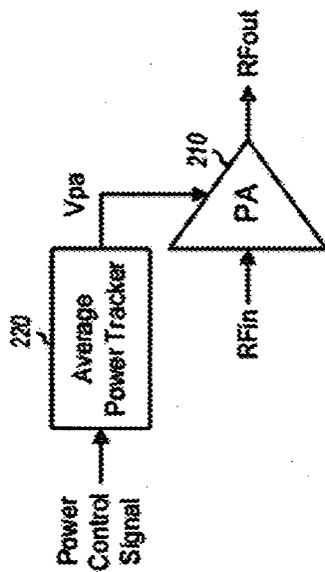
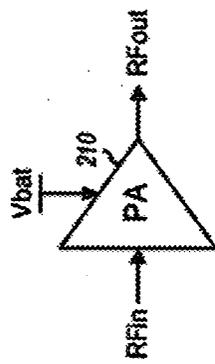
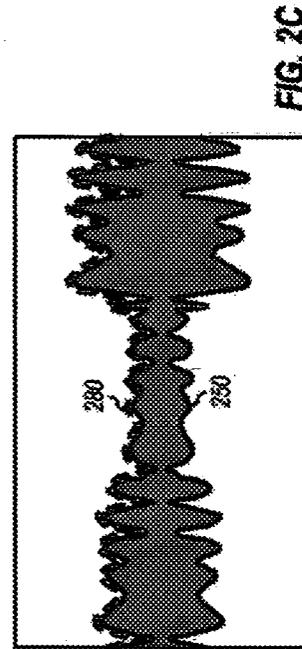
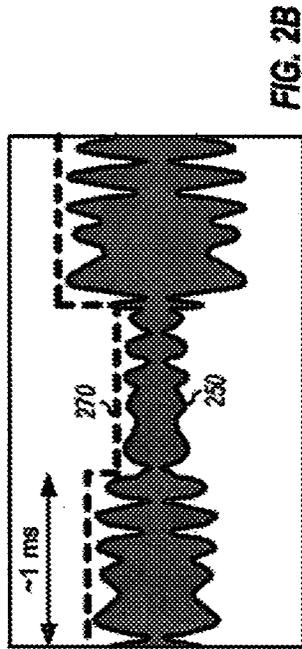
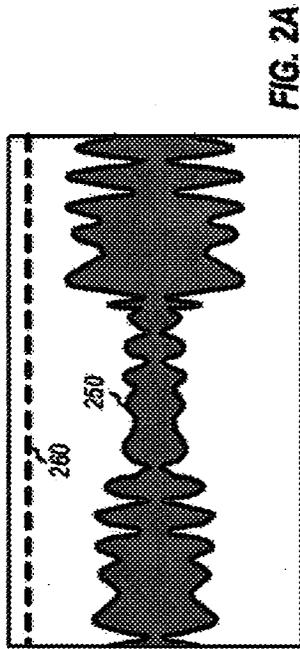


FIG. 1



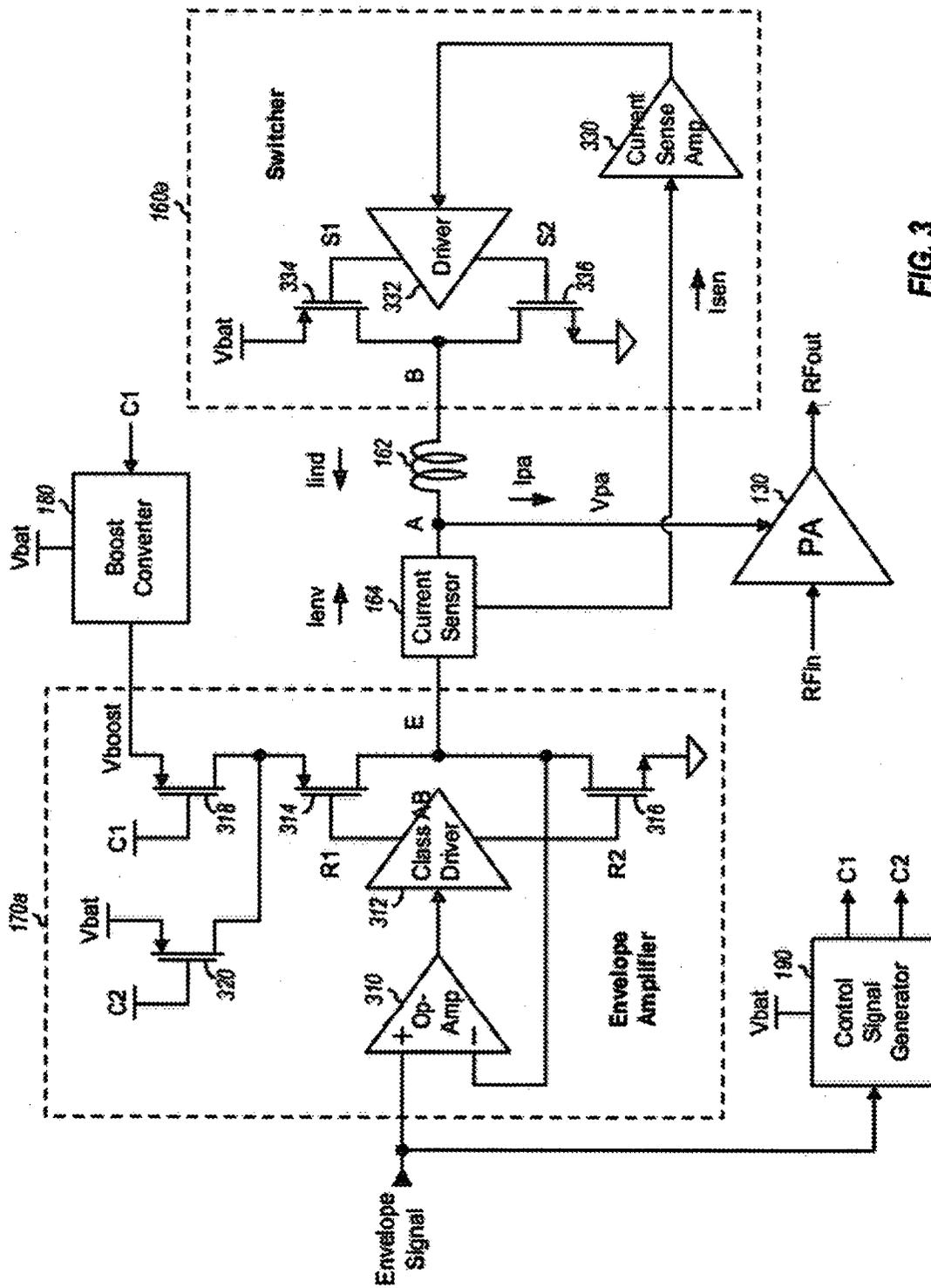


FIG. 3

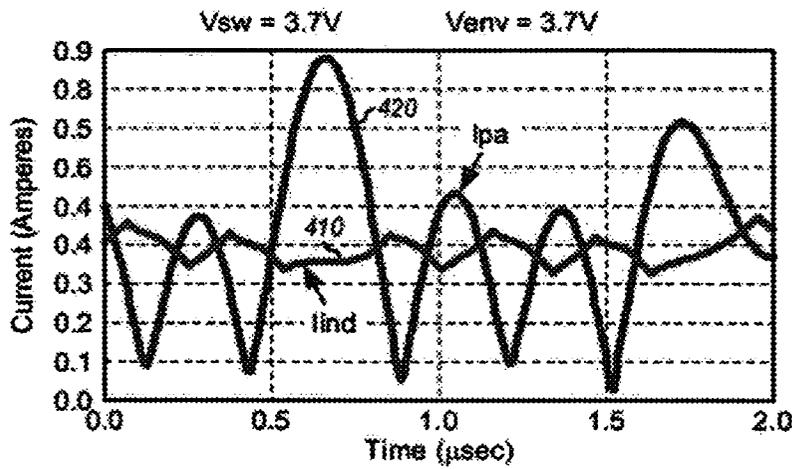


FIG. 4A

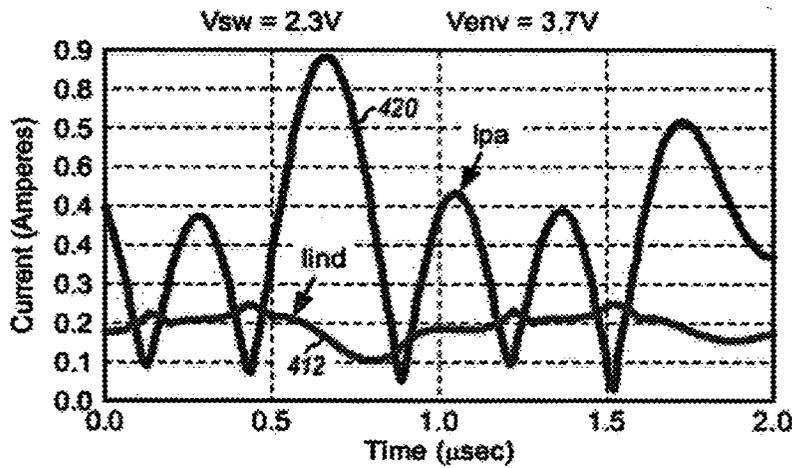


FIG. 4B

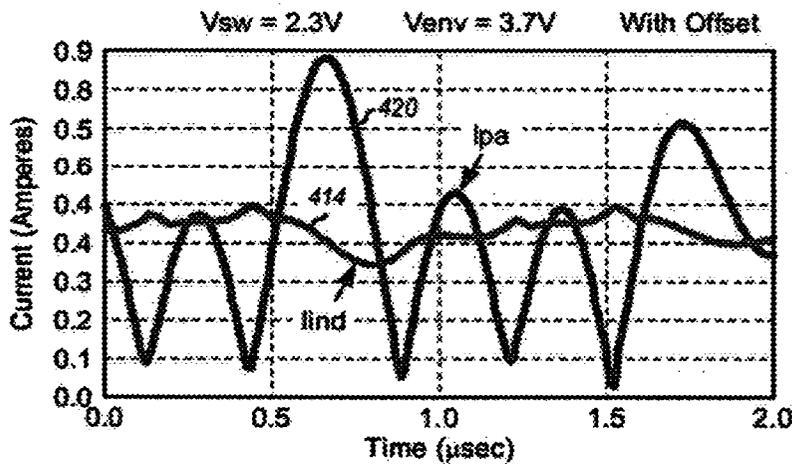


FIG. 4C

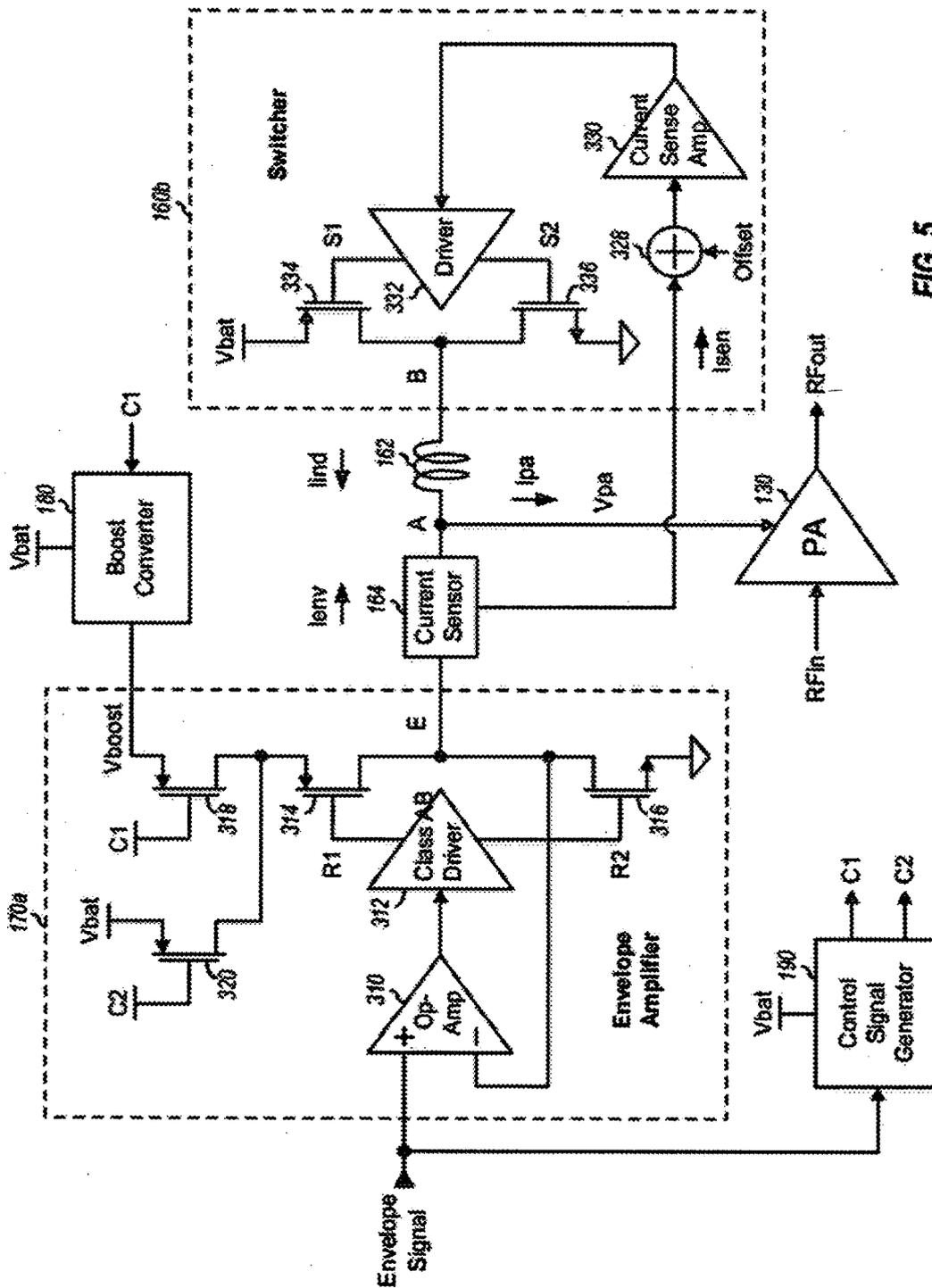


FIG. 5

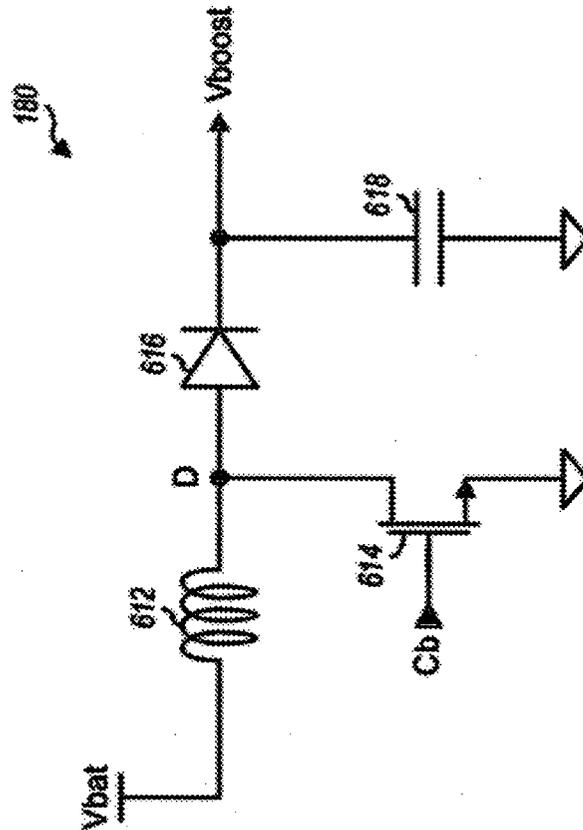


FIG. 6

1

LOW-VOLTAGE POWER-EFFICIENT ENVELOPE TRACKER

BACKGROUND

I. Field

The present disclosure relates generally to electronics, and more specifically to techniques for generating a power supply for an amplifier and/or other circuits.

II. Background

In a communication system, a transmitter may process (e.g., encode and modulate) data to generate output samples. The transmitter may further condition (e.g., convert to analog, filter, frequency upconvert, and amplify) the output samples to generate an output radio frequency (RF) signal. The transmitter may then transmit the output RF signal via a communication channel to a receiver. The receiver may receive the transmitted RF signal and perform the complementary processing on the received RF signal to recover the transmitted data.

The transmitter typically includes a power amplifier (PA) to provide high transmit power for the output RF signal. The power amplifier should be able to provide high output power and have high power-added efficiency (PAE). Furthermore, the power amplifier may be required to have good performance and high PAE even with a low battery voltage.

SUMMARY

Techniques for efficiently generating a power supply for a power amplifier and/or other circuits are described herein. In one exemplary design, an apparatus (e.g., an integrated circuit, a wireless device, a circuit module, etc.) may include an envelope amplifier and a boost converter. The boost converter may receive a first supply voltage (e.g., a battery voltage) and generate a boosted supply voltage having a higher voltage than the first supply voltage. The envelope amplifier may receive an envelope signal and the boosted supply voltage and may generate a second supply voltage based on the envelope signal and the boosted supply voltage. The apparatus may further include a power amplifier, which may operate based on the second supply voltage from the envelope amplifier. In one design, the envelope amplifier may further receive the first supply voltage and may generate the second supply voltage based on either the first supply voltage or the boosted supply voltage. For example, the envelope amplifier may generate the second supply voltage (i) based on the boosted supply voltage if the envelope signal exceeds a first threshold and/or if the first supply voltage is below a second threshold or (ii) based on the first supply voltage otherwise.

In another exemplary design, an apparatus may include a switcher, an envelope amplifier, and a power amplifier. The switcher may receive a first supply voltage (e.g., a battery voltage) and provide a first supply current. The envelope amplifier may receive an envelope signal and provide a second supply current based on the envelope signal. The power amplifier may receive a total supply current comprising the first supply current and the second supply current. The first supply current may include direct current (DC) and low frequency components. The second supply current may include higher frequency components. The apparatus may further include a boost converter, which may receive the first supply voltage and provide a boosted supply voltage. The envelope amplifier may then operate based on either the first supply voltage or the boosted supply voltage.

In yet another exemplary design, an apparatus may include a switcher that may sense an input current and generate a

2

switching signal to charge and discharge an inductor providing a supply current. The switcher may add an offset to the input current to generate a larger supply current than without the offset. The apparatus may further include an envelope amplifier, a boost converter, and a power amplifier, which may operate as described above.

Various aspects and features of the disclosure are described in further detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a block diagram of a wireless communication device.

FIGS. 2A, 2B and 2C show diagrams of operating a power amplifier based on a battery voltage, an average power tracker, and an envelope tracker, respectively.

FIG. 3 shows a schematic diagram of a switcher and an envelope amplifier.

FIGS. 4A, 4B and 4C show plots of PA supply current and inductor current versus time for different supply voltages for the switcher and the envelope amplifier.

FIG. 5 shows a schematic diagram of a switcher with offset in a current sensing path.

FIG. 6 shows a schematic diagram of a boost converter.

DETAILED DESCRIPTION

The word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any design described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other designs.

Techniques for generating a power supply for an amplifier and/or other circuits are described herein. The techniques may be used for various types of amplifiers such as power amplifiers, driver amplifiers, etc. The techniques may also be used for various electronic devices such as wireless communication devices, cellular phones, personal digital assistants (PDAs), handheld devices, wireless modems, laptop computers, cordless phones, Bluetooth devices, consumer electronic devices, etc. For clarity, the use of the techniques to generate a power supply for a power amplifier in a wireless communication device is described below.

FIG. 1 shows a block diagram of a design of a wireless communication device 100. For clarity, only a transmitter portion of wireless device 100 is shown in FIG. 1, and a receiver portion is not shown. Within wireless device 100, a data processor 110 may receive data to be transmitted, process (e.g., encode, interleave, and symbol map) the data, and provide data symbols. Data processor 110 may also process pilot and provide pilot symbols. Data processor 110 may also process the data symbols and pilot symbols for code division multiple access (CDMA), time division multiple access (TDMA), frequency division multiple access (FDMA), orthogonal FDMA (OFDMA), single-carrier FDMA (SC-FDMA), and/or some other multiplexing scheme and may provide output symbols.

A modulator 112 may receive the output symbols from data processor 110, perform quadrature modulation, polar modulation, or some other type of modulation, and provide output samples. Modulator 112 may also determine the envelope of the output samples, e.g., by computing the magnitude of each output sample and averaging the magnitude across output samples. Modulator 112 may provide an envelope signal indicative of the envelope of the output samples.

An RF transmitter 120 may process (e.g., convert to analog, amplify, filter, and frequency upconvert) the output samples from modulator 112 and provide an input RF signal

Exhibit B

Page 35

(RFin). A power amplifier (PA) 130 may amplify the input RF signal to obtain the desired output power level and provide an output RF signal (RFout), which may be transmitted via an antenna (not shown in FIG. 1). RF transmitter 120 may also include circuits to generate the envelope signal, instead of using modulator 112 to generate the envelope signal.

PA supply generator 150 may receive the envelope signal from modulator 112 and may generate a power supply voltage (V_{pa}) for power amplifier 130. PA supply generator 150 may also be referred to as an envelope tracker. In the design shown in FIG. 1, PA supply generator 150 includes a switcher 160, an envelope amplifier (Env Amp) 170, a boost converter 180, and an inductor 162. Switcher 160 may also be referred to as a switching-mode power supply (SMPS). Switcher 160 receives a battery voltage (V_{bat}) and provides a first supply current (I_{ind}) comprising DC and low frequency components at node A. Inductor 162 stores current from switcher 160 and provides the stored current to node A on alternating cycles. Boost converter 180 receives the V_{bat} voltage and generates a boosted supply voltage (V_{boost}) that is higher than the V_{bat} voltage. Envelope amplifier 170 receives the envelope signal at its signal input, receives the V_{bat} voltage and the V_{boost} voltage at its two power supply inputs, and provides a second supply current (I_{env}) comprising high frequency components at node A. The PA supply current (I_{pa}) provided to power amplifier 130 includes the I_{ind} current from switcher 160 and the I_{env} current from envelope amplifier 170. Envelope amplifier 170 also provides the proper PA supply voltage (V_{pa}) at Node A for power amplifier 130. The various circuits in PA supply generator 150 are described in further detail below.

A controller 140 may control the operation of various units within wireless device 100. A memory 142 may store program codes and data for controller 140 and/or other units within wireless device 100. Data processor 110, modulator 112, controller 140, and memory 142 may be implemented on one or more application specific integrated circuits (ASICs) and/or other ICs.

FIG. 1 shows an exemplary design of wireless device 100. Wireless device 100 may also be implemented in other manners and may include different circuits than those shown in FIG. 1. All or a portion of RF transmitter 120, power amplifier 130, and PA supply generator 150 may be implemented on one or more analog integrated circuits (ICs), RF ICs (RFICs), mixed-signal ICs, etc.

It may be desirable to operate wireless device 100 with a low battery voltage in order to reduce power consumption, extend battery life, and/or obtain other advantages. New battery technology may be able to provide energy down to 2.5 volts (V) and below in the near future. However, a power amplifier may need to operate with a PA supply voltage (e.g., 3.2V) that is higher than the battery voltage. A boost converter may be used to boost the battery voltage to generate the higher PA supply voltage. However, the use of the boost converter to directly supply the PA supply voltage may increase cost and power consumption, both of which are undesirable.

PA supply generator 150 can efficiently generate the PA supply voltage with envelope tracking to avoid the disadvantages of using a boost converter to directly provide the PA supply voltage. Switcher 160 may provide the bulk of the power for power amplifier 130 and may be connected directly to the battery voltage. Boost converter 180 may provide power to only envelope amplifier 170. PA supply generator 150 can generate the PA supply voltage to track the envelope of the RFin signal provided to power amplifier 130, so that just the proper amount of PA supply voltage is supplied to power amplifier 130.

FIG. 2A shows a diagram of using a battery voltage for a power amplifier 210. The RFout signal (which follows the RFin signal) has a time-varying envelope and is shown by a plot 250. The battery voltage is shown by a plot 260 and is higher than the largest amplitude of the envelope in order to avoid clipping of the RFout signal from power amplifier 210. The difference between the battery voltage and the envelope of the RFout signal represents wasted power that is dissipated by power amplifier 210 instead of delivered to an output load.

FIG. 2B shows a diagram of generating a PA supply voltage (V_{pa}) for power amplifier 210 with an average power tracker (APT) 220. APT 220 receives a power control signal indicating the largest amplitude of the envelope of the RFout signal in each time interval. APT 220 generates the PA supply voltage (which is shown by a plot 270) for power amplifier 210 based on the power control signal. The difference between the PA supply voltage and the envelope of the RFout signal represents wasted power. APT 220 can reduce wasted power since it can generate the PA supply voltage to track the largest amplitude of the envelope in each time interval.

FIG. 2C shows a diagram of generating a PA supply voltage for power amplifier 210 with an envelope tracker 230. Envelope tracker 230 receives an envelope signal indicative of the envelope of the RFout signal and generates the PA supply voltage (which is shown by a plot 280) for power amplifier 210 based on the envelope signal. The PA supply voltage closely tracks the envelope of the RFout signal over time. Hence, the difference between the PA supply voltage and the envelope of the RFout signal is small, which results in less wasted power. The power amplifier is operated in saturation for all envelope amplitudes in order to maximize PA efficiency.

PA supply generator 150 in FIG. 1 can implement envelope tracker 230 in FIG. 2C with high efficiency. This is achieved by a combination of (i) an efficient switcher 160 to generate a first supply current (I_{ind}) with a switch mode power supply and (ii) a linear envelope amplifier 170 to generate a second supply current (I_{env}).

FIG. 3 shows a schematic diagram of a switcher 160a and an envelope amplifier 170a, which are one design of switcher 160 and envelope amplifier 170, respectively, in FIG. 1. Within envelope amplifier 170a, an operational amplifier (op-amp) 310 has its non-inverting input receiving the envelope signal, its inverting input coupled to an output of envelope amplifier 170a (which is node E), and its output coupled to an input of a class AB driver 312. Driver 312 has its first output (R1) coupled to the gate of a P-channel metal oxide semiconductor (PMOS) transistor 314 and its second output (R2) coupled to the gate of an N-channel MOS (NMOS) transistor 316. NMOS transistor 316 has its drain coupled to node E and its source coupled to circuit ground. PMOS transistor 314 has its drain coupled to node E and its source coupled to the drains of PMOS transistors 318 and 320. PMOS transistor 318 has its gate receiving a C1 control signal and its source receiving the V_{boost} voltage. PMOS transistor 320 has its gate receiving a C2 control signal and its source receiving the V_{bat} voltage.

A current sensor 164 is coupled between node E and node A and senses the I_{env} current provided by envelope amplifier 170a. Sensor 164 passes most of the I_{env} current to node A and provides a small sensed current (I_{sen}) to switcher 160a. The I_{sen} current is a small fraction of the I_{env} current from envelope amplifier 170a.

Within switcher 160a, a current sense amplifier 330 has its input coupled to current sensor 164 and its output coupled to an input of a switcher driver 332. Driver 332 has its first output (S1) coupled to the gate of a PMOS transistor 334 and

its second output (S2) coupled to the gate of an NMOS transistor 336. NMOS transistor 336 has its drain coupled to an output of switcher 160a (which is node B) and its source coupled to circuit ground. PMOS transistor 334 has its drain coupled to node B and its source receiving the Vbat voltage. Inductor 162 is coupled between nodes A and B.

Switcher 160a operates as follows. Switcher 160a is in an On state when current sensor 164 senses a high output current from envelope amplifier 170a and provides a low sensed voltage to driver 332. Driver 332 then provides a low voltage to the gate of PMOS transistor 334 and a low voltage to the gate of NMOS transistor 336. PMOS transistor 334 is turned on and couples the Vbat voltage to inductor 162, which stores energy from the Vbat voltage. The current through inductor 162 rises during the On state, with the rate of the rise being dependent on (i) the difference between the Vbat voltage and the Vpa voltage at node A and (ii) the inductance of inductor 162. Conversely, switcher 160a is in an Off state when current sensor 164 senses a low output current from envelope amplifier 170a and provides a high sensed voltage to driver 332. Driver 332 then provides a high voltage to the gate of PMOS transistor 334 and a high voltage to the gate of NMOS transistor 336. NMOS transistor 336 is turned on, and inductor 162 is coupled between node A and circuit ground. The current through inductor 162 falls during the Off state, with the rate of the fall being dependent on the Vpa voltage at node A and the inductance of inductor 162. The Vbat voltage thus provides current to power amplifier 130 via inductor 162 during the On state, and inductor 120 provides its stored energy to power amplifier 130 during the Off state.

In one design, envelope amplifier 170a operates based on the Vboost voltage only when needed and based on the Vbat voltage the remaining time in order to improve efficiency. For example, envelope amplifier 170a may provide approximately 85% of the power based on the Vbat voltage and only approximately 15% of the power based on the Vboost voltage. When a high Vpa voltage is needed for power amplifier 130 due to a large envelope on the RFout signal, the C1 control signal is at logic low, and the C2 control signal is at logic high. In this case, boost converter 180 is enabled and generates the Vboost voltage, PMOS transistor 318 is turned on and provides the Vboost voltage to the source of PMOS transistor 314, and PMOS transistor 320 is turned off. Conversely, when a high Vpa voltage is not needed for power amplifier 130, the C1 control signal is at logic high, and the C2 control signal is at logic low. In this case, boost converter 180 is disabled, PMOS transistor 318 is turned off, and PMOS transistor 320 is turned on and provides the Vbat voltage to the source of PMOS transistor 314.

Envelope amplifier 170a operates as follows. When the envelope signal increases, the output of op-amp 310 increases, the R1 output of driver 312 decreases and the R2 output of driver 312 decreases until NMOS transistor 316 is almost turned off, and the output of envelope amplifier 170a increases. The converse is true when the envelope signal decreases. The negative feedback from the output of envelope amplifier 170a to the inverting input of op-amp 310 results in envelope amplifier 170a having unity gain. Hence, the output of envelope amplifier 170a follows the envelope signal, and the Vpa voltage is approximately equal to the envelope signal. Driver 312 may be implemented with a class AB amplifier to improve efficiency, so that large output currents can be supplied even though the bias current in transistors 314 and 316 is very low.

A control signal generator 190 receives the envelope signal and the Vbat voltage and generates the C1 and C2 control signals. The C1 control signal is complementary to the C2

control signal. In one design, generator 190 generates the C1 and C2 control signals to select the Vboost voltage for envelope amplifier 170 when the magnitude of the envelope signal exceeds a first threshold. The first threshold may be a fixed threshold or may be determined based on the Vbat voltage. In another design, generator 190 generates the C1 and C2 control signals to select the Vboost voltage for envelope amplifier 170 when the magnitude of the envelope signal exceeds the first threshold and the Vbat voltage is below a second threshold. Generator 190 may also generate the C1 and C2 signals based on other signals, other voltages, and/or other criteria.

FIG. 3 shows an exemplary design of switcher 160 and envelope amplifier 170 in FIG. 1. Switcher 160 and envelope amplifier 170 may also be implemented in other manners. For example, envelope amplifier 170 may be implemented as described in U.S. Pat. No. 6,300,826, entitled "Apparatus and Method for Efficiently Amplifying Wideband Envelope Signals," issued Oct. 9, 2001.

Switcher 160a has high efficiency and delivers a majority of the supply current for power amplifier 130. Envelope amplifier 170a operates as a linear stage and has relatively high bandwidth (e.g., in the MHz range). Switcher 160a operates to reduce the output current from envelope amplifier 170a, which improves overall efficiency.

It may be desirable to support operation of wireless device 100 with a low battery voltage (e.g., below 2.5V). This may be achieved by operating switcher 160 based on the Vbat voltage and operating envelope amplifier 170 based on the higher Vboost voltage. However, efficiency may be improved by operating envelope amplifier 170 based on the Vboost voltage only when needed for large amplitude envelope and based on the Vbat voltage the remaining time, as shown in FIG. 3 and described above.

FIG. 4A shows plots of an example of the PA supply current (Ipa) and the inductor current (Iind) from inductor 162 versus time for a case in which switcher 160a has a supply voltage (Vsw) of 3.7V and envelope amplifier 170a has a supply voltage (Venv) of 3.7V. The Iind current is the current through inductor 162 and is shown by a plot 410. The Ipa current is the current provided to power amplifier 130 and is shown by a plot 420. The Ipa current includes the Iind current as well as the Ienv current from envelope amplifier 170a. Envelope amplifier 170a provides output current whenever the Ipa current is higher than the Iind current. The efficiency of switcher 160a and envelope amplifier 170a is approximately 80% in one exemplary design.

FIG. 4B shows plots of the PA supply current (Ipa) and the inductor current (Iind) versus time for a case in which switcher 160a has a supply voltage of 2.3V and envelope amplifier 170a has a supply voltage of 3.7V. The Iind current is shown by a plot 412, and the Ipa current is shown by plot 420. When the supply voltage of switcher 160a is reduced to 2.3V, inductor 162 charges more slowly, which results in a lower average Iind current as compared to the case in which the supply voltage of switcher 160a is at 3.7V in FIG. 4A. The lower Iind current causes envelope amplifier 170a to provide more of the Ipa current. This reduces the overall efficiency to approximately 65% in one exemplary design because envelope amplifier 170a is less efficient than switcher 160a. The drop in efficiency may be ameliorated by increasing the Iind current from the switcher.

FIG. 5 shows a schematic diagram of a switcher 160b, which is another design of switcher 160 in FIG. 1. Switcher 160b includes current sense amplifier 330, driver 332, and MOS transistors 334 and 336, which are coupled as described above for switcher 160a in FIG. 3. Switcher 160b further includes a current summer 328 having a first input coupled to

current sensor 164, a second input receiving an offset (e.g., an offset current), and an output coupled to the input of current sense amplifier 330. Summer 328 may be implemented with a summing circuit (e.g., an amplifier), a summing node, etc.

Switcher 160b operates as follows. Summer 328 receives the Isen current from current sensor 164, adds an offset current, and provides a summed current that is lower than the Isen current by the offset current. The remaining circuits within switcher 160b operate as described above for switcher 160a in FIG. 3. Summer 328 intentionally reduces the Isen current provided to current sense amplifier 330, so that switcher 160 is turned On for a longer time period and can provide a larger Iind current, which is part of the Ipa current provided to power amplifier 130. The offset provided to summer 328 determines the amount by which the Iind current is increased by switcher 160b relative to the Iind current provided by switcher 160a in FIG. 3.

In general, a progressively larger offset may be used to generate a progressively larger inductor current than without the offset. In one design, the offset may be a fixed value selected to provide good performance, e.g., good efficiency. In another design, the offset may be determined based on the battery voltage. For example, a progressively larger offset may be used for a progressively lower battery voltage. The offset may also be determined based on the envelope signal and/or other information.

An offset to increase the inductor current may be added via summer 328, as shown in FIG. 5. An offset may also be added by increasing the pulse width of an output signal from current sense amplifier via any suitable mechanism.

FIG. 4C shows plots of the PA supply current (Ipa) and the inductor current (Iind) versus time for a case in which switcher 160b in FIG. 5 has a supply voltage of 2.3V and envelope amplifier 170a has a supply voltage of 3.7V. The Iind current is shown by a plot 414, and the Ipa current is shown by plot 420. When the supply voltage of switcher 160b is reduced to 2.3V, inductor 162 charges more slowly, which results in a lower Iind current as shown in FIG. 4B. The offset added by summer 328 in FIG. 5 reduces the sensed current provided to current sense amplifier 330 and results in switcher 160b being turned On longer. Hence, switcher 160b with offset in FIG. 5 can provide a higher Iind current than switcher 160a without offset in FIG. 3. The overall efficiency for switcher 160b and envelope amplifier 170a is improved to approximately 78% in one exemplary design.

FIG. 6 shows a schematic diagram of a design of boost converter 180 in FIGS. 1, 3 and 5. Within boost converter 180, an inductor 612 has one end receiving the Vbat voltage and the other end coupled to node D. An NMOS transistor 614 has its source coupled to circuit ground, its gate receiving a Cb control signal, and its drain coupled to node D. A diode 616 has its anode coupled to node D and its cathode coupled to the output of boost converter 180. A capacitor 618 has one end coupled to circuit ground and the other end coupled to the output of boost converter 180.

Boost converter 180 operates as follows. In an On state, NMOS transistor 614 is closed, inductor 612 is coupled between the Vbat voltage and circuit ground, and the current via inductor 612 increases. In an Off state, NMOS transistor 614 is opened, and the current from inductor 612 flows via diode 616 to capacitor 618 and a load at the output of boost converter 180 (not shown in FIG. 6). The Vboost voltage may be expressed as:

$$V_{boost} = V_{bat} \cdot \frac{1}{1 - \text{Duty_Cycle}} \quad \text{Eq (1)}$$

where Duty_Cycle is the duty cycle in which NMOS transistor 614 is turned on. The duty cycle may be selected to obtain the desired Vboost voltage and to ensure proper operation of boost converter 180.

The techniques described herein enable an envelope tracker to operate at a lower battery voltage (e.g., 2.5V or lower). The envelope tracker includes switcher 160 and envelope amplifier 170 for the design shown in FIG. 1. In one design of supporting operation with a lower battery voltage, as shown in FIG. 3, switcher 160 is connected to the Vbat voltage and envelope amplifier 170 is connected to either the Vbat voltage or the Vboost voltage. Switcher 160 provides power most of the time, and envelope amplifier 170 provides power during peaks in the envelope of the RFout signal. The overall efficiency of the envelope tracker is reduced by the efficiency of boost converter 180 (which may be approximately 85%) only during the time in which envelope amplifier 170 provides power.

In another design of supporting operation with a lower battery voltage, the entire envelope tracker is operated based on the Vboost voltage from boost converter 180. In this design, boost converter 180 provides high current required by power amplifier 130 (which may be more than one Ampere), and efficiency is reduced by the efficiency of boost converter 180 (which may be approximately 85%).

In yet another design of supporting operation with a lower battery voltage, a field effect transistor (FET) switch is used to connect the envelope tracker to (i) the Vbat voltage when the Vbat voltage is greater than a Vthresh voltage or (ii) the Vboost voltage when the Vbat voltage is less than the Vthresh voltage. Efficiency would then be reduced by losses in the FET switch. However, better efficiency may be obtained for envelope amplifier 170 due to a lower input voltage.

In one exemplary design, an apparatus (e.g., an integrated circuit, a wireless device, a circuit module, etc.) may comprise an envelope amplifier and a boost converter, e.g., as shown in FIGS. 1 and 3. The boost converter may receive a first supply voltage and generate a boosted supply voltage having a higher voltage than the first supply voltage. The first supply voltage may be a battery voltage, a line-in voltage, or some other voltage available to the apparatus. The envelope amplifier may receive an envelope signal and the boosted supply voltage and may generate a second supply voltage (e.g., the Vpa voltage in FIG. 3) based on the envelope signal and the boosted supply voltage. The apparatus may further comprise a power amplifier, which may operate based on the second supply voltage from the envelope amplifier. The power amplifier may receive and amplify an input RF signal and provide an output RF signal.

In one design, the envelope amplifier may further receive the first supply voltage and may generate the second supply voltage based on the first supply voltage or the boosted supply voltage. For example, the envelope amplifier may generate the second supply voltage (i) based on the boosted supply voltage if the envelope signal exceeds a first threshold, or if the first supply voltage is below a second threshold, or both or (ii) based on the first supply voltage otherwise.

In one design, the envelope amplifier may include an op-amp, a driver, a PMOS transistor, and an NMOS transistor, e.g., op-amp 310, driver 312, PMOS transistor 314, and NMOS transistor 316 in FIG. 3. The op-amp may receive the envelope signal and provide an amplified signal. The driver

may receive the amplified signal and provide a first control signal (R1) and a second control signal (R2). The PMOS transistor may have a gate receiving the first control signal, a source receiving the boosted supply voltage or the first supply voltage, and a drain providing the second supply voltage. The NMOS transistor may have a gate receiving the second control signal, a drain providing the second supply voltage, and a source coupled to circuit ground. The envelope amplifier may further comprise second and third PMOS transistors (e.g., PMOS transistors 318 and 320). The second PMOS transistor may have a gate receiving a third control signal (C1), a source receiving the boosted supply voltage, and a drain coupled to the source of the PMOS transistor. The third PMOS transistor may have a gate receiving a fourth control signal (C2), a source receiving the first supply voltage, and a drain coupled to the source of the PMOS transistor.

In another exemplary design, an apparatus (e.g., an integrated circuit, a wireless device, a circuit module, etc.) may comprise a switcher, an envelope amplifier, and a power amplifier, e.g., as shown in FIGS. 1 and 3. The switcher may receive a first supply voltage (e.g., a battery voltage) and provide a first supply current (e.g., the I_{ind} current in FIG. 3). The envelope amplifier may receive an envelope signal and provide a second supply current (e.g., the I_{env} current) based on the envelope signal. The power amplifier may receive a total supply current (e.g., the I_{pa} current) comprising the first supply current and the second supply current. The first supply current may comprise DC and low frequency components. The second supply current may comprise higher frequency components. The apparatus may further comprise a boost converter, which may receive the first supply voltage and provide a boosted supply voltage having a higher voltage than the first supply voltage. The envelope amplifier may operate based on the first supply voltage or the boosted supply voltage.

In one design, the switcher may comprise a current sense amplifier, a driver, a PMOS transistor, and an NMOS transistor, e.g., current sense amplifier 330, driver 332, PMOS transistor 334, and NMOS transistor 336 in FIG. 3. The current sense amplifier may sense the first supply current, or the second supply current (e.g., as shown in FIG. 3), or the total supply current and may provide a sensed signal. The driver may receive the sensed signal and provide a first control signal (S1) and a second control signal (S2). The PMOS transistor may have a gate receiving the first control signal, a source receiving the first supply voltage, and a drain providing a switching signal for an inductor providing the first supply current. The NMOS transistor may have a gate receiving the second control signal, a drain providing the switching signal, and a source coupled to circuit ground. The inductor (e.g., inductor 162) may be coupled to the drains of the PMOS transistor and the NMOS transistor, may receive the switching signal at one end, and may provide the first supply current at the other end.

In yet another exemplary design, an apparatus (e.g., an integrated circuit, a wireless device, a circuit module, etc.) may comprise a switcher, e.g., switcher 160b in FIG. 5. The switcher may sense an input current (e.g., the I_{env} current in FIG. 5) and generate a switching signal to charge and discharge an inductor providing a supply current (e.g., the I_{ind} current). The switcher may add an offset to the input current to generate a larger supply current than without the offset. The switcher may operate based on a first supply voltage (e.g., a battery voltage). In one design, the offset may be determined based on the first supply voltage. For example, a larger offset may be used for a smaller first supply voltage, and vice versa.

In one design, the switcher may comprise a summer, a current sense amplifier, and a driver, e.g., summer 328, current sense amplifier 330, and driver 332 in FIG. 5. The summer may sum the input current and an offset current and provide a summed current. The current sense amplifier may receive the summed current and provide a sensed signal. The driver may receive the sensed signal and provide at least one control signal used to generate the switching signal. In one design, the at least one control signal may comprise a first control signal (S1) and a second control signal (S2), and the switcher may further comprise a PMOS transistor and an NMOS transistor, e.g., PMOS transistor 334 and NMOS transistor 336 in FIG. 5. The PMOS transistor may have a gate receiving the first control signal, a source receiving the first supply voltage, and a drain providing the switching signal. The NMOS transistor may have a gate receiving the second control signal, a drain providing the switching signal, and a source coupled to circuit ground.

In one design, the apparatus may further comprise an envelope amplifier, a boost converter, and a power amplifier. The envelope amplifier may receive an envelope signal and provide a second supply current (e.g., the I_{env} current in FIG. 5) based on the envelope signal. The boost converter may receive the first supply voltage and provide a boosted supply voltage. The envelope amplifier may operate based on the first supply voltage or the boosted supply voltage. The power amplifier may receive a total supply current (e.g., the I_{pa} current) comprising the supply current from the switcher and the second supply current from the envelope amplifier.

The circuits (e.g., the envelope amplifier, the switcher, the boost converter, etc.) described herein may be implemented on an IC, an analog IC, an RF IC (RFIC), a mixed-signal IC, an ASIC, a printed circuit board (PCB), an electronic device, etc. The circuits may be fabricated with various IC process technologies such as complementary metal oxide semiconductor (CMOS), NMOS, PMOS, bipolar junction transistor (BJT), bipolar-CMOS (BiCMOS), silicon germanium (SiGe), gallium arsenide (GaAs), etc.

An apparatus implementing any of the circuits described herein may be a stand-alone device or may be part of a larger device. A device may be (i) a stand-alone IC, (ii) a set of one or more ICs that may include memory ICs for storing data and/or instructions, (iii) an RFIC such as an RF receiver (RFR) or an RF transmitter/receiver (RTR), (iv) an ASIC such as a mobile station modem (MSM), (v) a module that may be embedded within other devices, (vi) a receiver, cellular phone, wireless device, handset, or mobile unit, (vii) etc.

The previous description of the disclosure is provided to enable any person skilled in the art to make or use the disclosure. Various modifications to the disclosure will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other variations without departing from the scope of the disclosure. Thus, the disclosure is not intended to be limited to the examples and designs described herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

What is claimed is:

1. An apparatus comprising:

- a boost converter operative to receive a first supply voltage and generate a boosted supply voltage having a higher voltage than the first supply voltage; and
- an envelope amplifier operative to receive an envelope signal and the boosted supply voltage and generate a second supply voltage based on the envelope signal and the boosted supply voltage, wherein the envelope amplifier is operative to further receive the first supply voltage and generate the second supply voltage based on the first

Exhibit B

supply voltage and generate the second supply voltage based on the first supply voltage or the boosted supply voltage, and further wherein the envelope amplifier comprises

an operational amplifier (op-amp) operative to receive the envelope signal and provide an amplified signal, a driver operative to receive the amplified signal and provide a first control signal and a second control signal, a P-channel metal oxide semiconductor (PMOS) transistor having a gate receiving the first control signal, a source receiving the boosted supply voltage or the first supply voltage, and a drain providing the second supply voltage, and
 an N-channel metal oxide semiconductor (NMOS) transistor having a gate receiving the second control signal, a drain providing the second supply voltage, and a source coupled to circuit ground.

2. The apparatus of claim 1, wherein the envelope amplifier is operative to generate the second supply voltage based on the boosted supply voltage if the envelope signal exceeds a first threshold, or if the first supply voltage is below a second threshold, or both.

3. The apparatus of claim 1, wherein the envelope amplifier further comprises

a second PMOS transistor having a gate receiving a third control signal, a source receiving the boosted supply voltage, and a drain coupled to the source of the PMOS transistor, and

a third PMOS transistor having a gate receiving a fourth control signal, a source receiving the first supply voltage, and a drain coupled to the source of the PMOS transistor.

4. The apparatus of claim 1, further comprising:

a power amplifier operative to receive the second supply voltage from the envelope amplifier and to receive and amplify an input radio frequency (RF) signal and provide an output RF signal.

5. The apparatus of claim 1, wherein the first supply voltage is a battery voltage for the apparatus.

6. An apparatus for wireless communication, comprising: a power amplifier operative to receive and amplify an input radio frequency (RF) signal and provide an output RF signal; and

a supply generator operative to receive an envelope signal and a first supply voltage, to generate a boosted supply voltage having a higher voltage than the first supply voltage, and to generate a second supply voltage for the power amplifier based on the envelope signal and the boosted supply voltage, wherein the supply generator incorporates an operational amplifier (op-amp) operative to receive the envelope signal and provide an amplified signal, a driver operative to receive the amplified signal and provide a first control signal and a second control signal, a P-channel metal oxide semiconductor (PMOS) transistor having a gate receiving a first control signal, a source receiving the boosted supply voltage or the first supply voltage, and a drain providing the second supply voltage, and an N-channel metal oxide semiconductor (NMOS) transistor having a gate receiving the second control signal, a drain providing the second supply voltage, and a source coupled to circuit ground.

7. The apparatus of claim 6, wherein the supply generator is operative to generate the second supply voltage based on the envelope signal and either the boosted supply voltage or the first supply voltage.

8. A method of generating supply voltages, comprising: generating a boosted supply voltage based on a first supply voltage, the boosted supply voltage having a higher voltage than the first supply voltage; and

generating a second supply voltage based on an envelope signal and the boosted supply voltage, wherein the second supply voltage is generated by an envelope amplifier that produces the second supply voltage using an operational amplifier (op-amp) that receives the envelope signal and provides an amplified signal, a driver that receives the amplified signal and provides a first control signal and a second control signal, a P-channel metal oxide semiconductor (PMOS) transistor that receives the first control signal, a source that receives the boosted supply voltage or the first supply voltage, and a drain providing the second supply voltage and an N-channel metal oxide semiconductor (NMOS) transistor that receives the second control signal at a gate and provides a second supply voltage through a drain, and a source for circuit grounding.

9. The method of claim 8, wherein the generating the second supply voltage comprises generating the second supply voltage based on the envelope signal and either the boosted supply voltage or the first supply voltage.

10. An apparatus for generating supply voltages, comprising:

means for generating a boosted supply voltage based on a first supply voltage, the boosted supply voltage having a higher voltage than the first supply voltage; and

means for generating a second supply voltage based on the envelope signal and the boosted supply voltage, wherein the means for generating the second supply voltage incorporates an envelope amplifier that produces the second supply voltage using an operational amplifier (op-amp) that receives the envelope signal and provides an amplified signal, a driver that receives the amplified signal and provides a first control signal and a second control signal, a P-channel metal oxide semiconductor (PMOS) transistor that receives the first control signal, a source that receives the boosted supply voltage or the first supply voltage, and a drain providing the second supply voltage and an N-channel metal oxide semiconductor (NMOS) transistor that receives the second control signal at a gate and provides a second supply voltage through a drain, and a source for circuit grounding.

11. The apparatus of claim 10, wherein the means for generating the second supply voltage comprises means for generating the second supply voltage based on an envelope signal and either the boosted supply voltage or the first supply voltage.

12. An apparatus comprising:

a switcher operative to receive a first supply voltage and provide a first supply current;

an envelope amplifier operative to receive an envelope signal and provide a second supply current based on the envelope signal; and

a power amplifier operative to receive an envelope signal and provide a second supply current based on the envelope signal; and

a power amplifier operative to receive a total supply current comprising the first supply current and the second supply current, wherein the switcher comprises

a current sense amplifier operative to sense the first supply current, or the second supply current, or the total supply current and provide a sensed signal,

a driver operative to receive the sensed signal and provide a first control signal and a second control signal,

13

a P-channel metal oxide semiconductor (PMOS) transistor having a gate receiving the first control signal, a source receiving the first supply voltage, and a drain providing a switching signal for an inductor providing the first supply current, and
 an N-channel metal oxide semiconductor (NMOS) transistor having a gate receiving the second control signal, a drain providing the switching signal, and a source coupled to circuit ground.

13. The apparatus of claim 12, further comprising:
 a boost converter operative to receive the first supply voltage and provide a boosted supply voltage having a higher voltage than the first supply voltage, wherein the envelope amplifier operates based on the first supply voltage or the boosted supply voltage.

14. The apparatus of claim 12, wherein the first supply current comprises direct current (DC) and low frequency components, and wherein the second supply current comprises higher frequency components.

15. An apparatus comprising:
 an inductor operative to receive a switching signal and provide a supply current; and
 a switcher operative to sense an input current and generate the switching signal to charge and discharge the inductor to provide the supply current, the switcher adding an offset to the input current to generate a larger supply current via the inductor than without the offset, wherein the switcher comprises
 a summer operative to sum the input current and an offset current and provide a summed current,
 a current sense amplifier operative to receive the summed current and provide a sensed signal, and
 a driver operative to receive the sensed signal and provide at least one control signal used to generate the switching signal for the inductor.

14

16. The apparatus of claim 15, wherein the switcher operates based on a first supply voltage, and wherein the offset is determined based on the first supply voltage.

17. The apparatus of claim 15, wherein the at least one control signal comprises a first control signal and a second control signal, and wherein the switcher further comprises
 a P-channel metal oxide semiconductor (PMOS) transistor having a gate receiving the first control signal, a source receiving a first supply voltage, and a drain providing the switching signal, and
 an N-channel metal oxide semiconductor (NMOS) transistor having a gate receiving the second control signal, a drain providing the switching signal, and a source coupled to circuit ground.

18. The apparatus of claim 15, further comprising:
 an envelope amplifier operative to receive an envelope signal and provide a second supply current based on the envelope signal, wherein a total supply current comprises the supply current from the switcher and the second supply current from the envelope amplifier.

19. The apparatus of claim 18, further comprising:
 a boost converter operative to receive the first supply voltage and provide a boosted supply voltage having a higher voltage than the first supply voltage, wherein the envelope amplifier operates based on the first supply voltage or the boosted supply voltage.

20. The apparatus of claim 15, further comprising:
 a power amplifier operative to receive the supply current from the inductor and to receive and amplify an input radio frequency (RF) signal and provide an output RF signal.

* * * * *

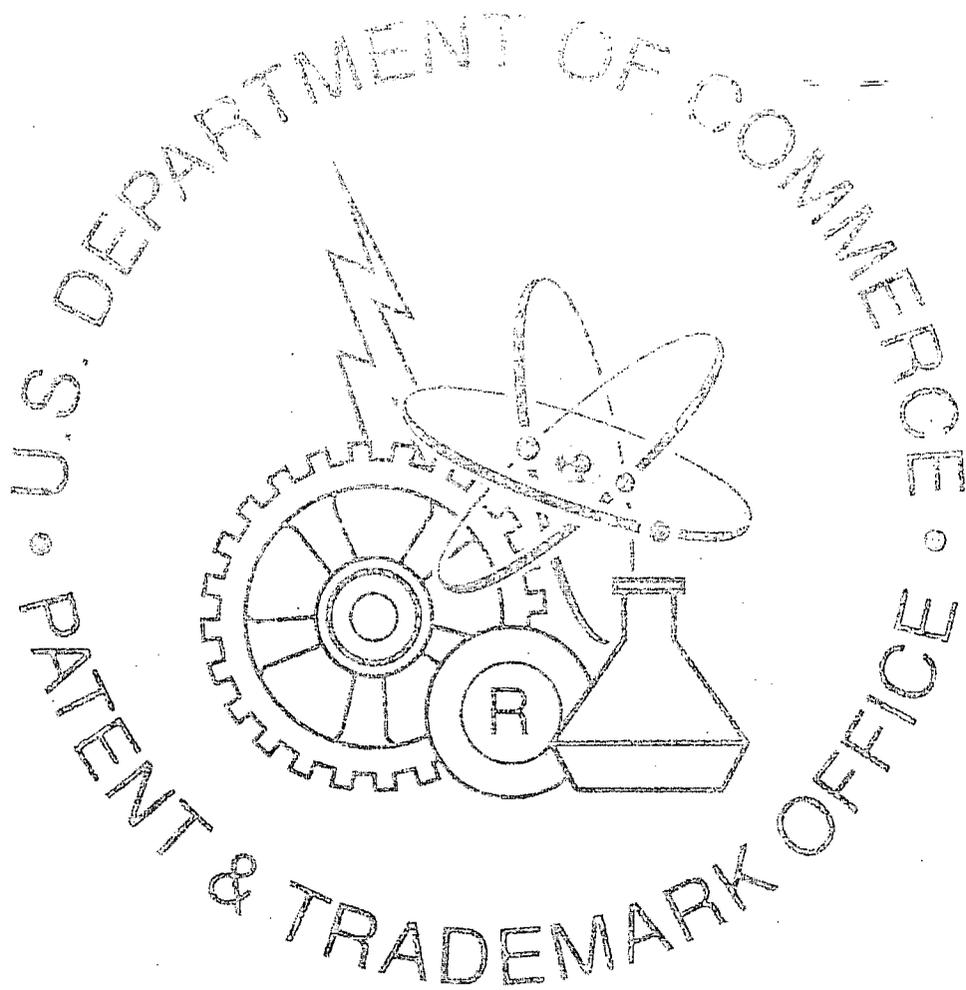


EXHIBIT C

(12) United States
Data et al.

8,487,658
Jul. 16, 2013

U 7629935



THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

**UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office**

April 06, 2017

**THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM
THE RECORDS OF THIS OFFICE OF:**

**U.S. PATENT: 8,487,658
ISSUE DATE: July 16, 2013**

U.S. PATENT DOCUMENTS

1,790,881 A	7/20/00	Method for
5,476,518 B1	7/20/07	Method for
6,974,979 B1	12/29/01	Method for
7,009,063 B2	8/2/04	Method for
7,408,289 B2	6/2/08	Method for
8,071,210 A6	12/28/07	Method for
8,040,227 B2	11/2/09	Method for

**By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office**



T. Lawrence
**T. LAWRENCE
Certifying Officer**



US008487658B2

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

(10) **Patent No.:** **US 8,487,658 B2**
(45) **Date of Patent:** **Jul. 16, 2013**

(54) **COMPACT AND ROBUST LEVEL SHIFTER LAYOUT DESIGN**

(75) **Inventors:** **Animesh Datta**, San Diego, CA (US);
William James Goodall, III, Cary, NC (US)

(73) **Assignee:** **QUALCOMM Incorporated**, San Diego, CA (US)

(*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 79 days.

(21) **Appl. No.:** **13/180,598**

(22) **Filed:** **Jul. 12, 2011**

(65) **Prior Publication Data**
US 2013/0015882 A1 Jan. 17, 2013

(51) **Int. Cl.**
H03K 19/00 (2006.01)
H01L 25/00 (2006.01)

(52) **U.S. Cl.**
USPC **326/101; 326/80; 326/63; 257/206**

(58) **Field of Classification Search**
None
See application file for complete search history.

(56) **References Cited**
U.S. PATENT DOCUMENTS

5,780,881	A	7/1998	Matsuda et al.	
6,414,518	B1	7/2002	Patel et al.	
6,974,978	B1*	12/2005	Possley	257/204
7,095,063	B2	8/2006	Cohn et al.	
7,408,269	B2	8/2008	Joshi et al.	
2003/0231046	A1	12/2003	Giacomini et al.	
2004/0225985	A1	11/2004	Kashiwagi et al.	

2008/0143418	A1	6/2008	Lu et al.	
2008/0265936	A1	10/2008	Vora	
2009/0108904	A1	4/2009	Shiffer, II	
2010/0214002	A1	8/2010	Miyoshi et al.	
2010/0238744	A1	9/2010	Yano	
2011/0001538	A1	1/2011	Alam	
2011/0031944	A1*	2/2011	Stirk et al.	323/234
2011/0266631	A1*	11/2011	Morino et al.	257/371

FOREIGN PATENT DOCUMENTS

JP	2008067411	A	3/2008
WO	2006025025	A1	3/2006

OTHER PUBLICATIONS

International Search Report and Written Opinion—PCT/US2012/046562—ISA/EPO—Sep. 24, 2012.

* cited by examiner

Primary Examiner — Jany Richardson

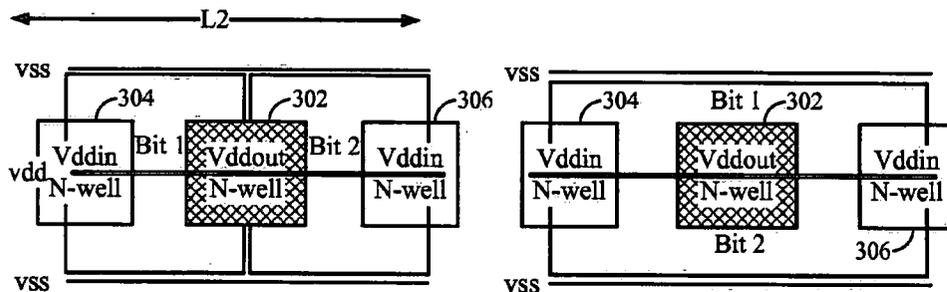
(74) *Attorney, Agent, or Firm* — Sam Talpalatsky; Nicholas J. Pauley; Joseph Augusta

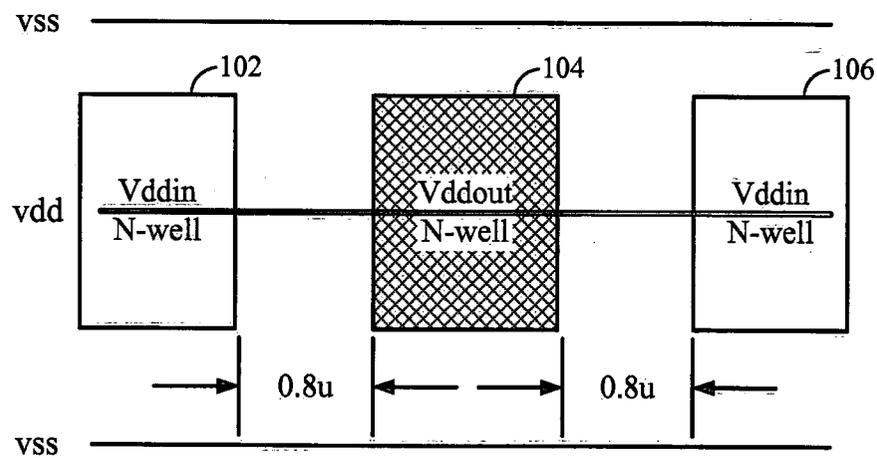
(57) **ABSTRACT**

Method and apparatus for voltage level shifters (VLS) design in bulk CMOS technology. A multi-voltage circuit or VLS that operate with different voltage levels and that provides area and power savings for multi-bit implementation of level shifter design. A two-bit VLS to shift bits from a first voltage level logic to a second voltage level logic. The VLS formed with a first N-well in a substrate. The VLS formed with a second N-well in the substrate, adjacent to a side of the first N-well. The VLS formed with a third N-well in the substrate, adjacent to a side of the first N-well and opposite the second N-well. A first one-bit VLS circuit having a portion formed on the first N-well and a portion formed on the second N-well. A second bit VLS circuit having a portion formed on the first N-well and a portion formed on the third N-well.

23 Claims, 5 Drawing Sheets

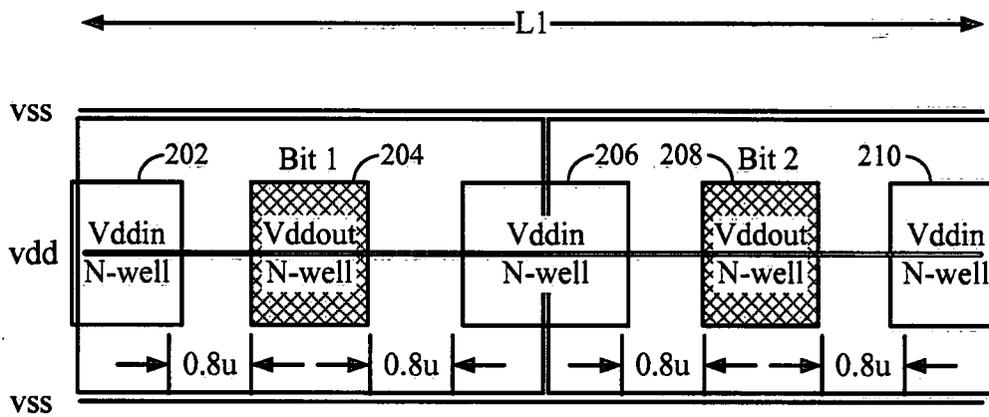
Compact physical design of 2-bit shifter layout





Layout architecture of 1-bit level shifter

FIG. 1



Conventional multi-bit level shifter layout architecture

FIG. 2

Compact physical design of 2-bit shifter layout

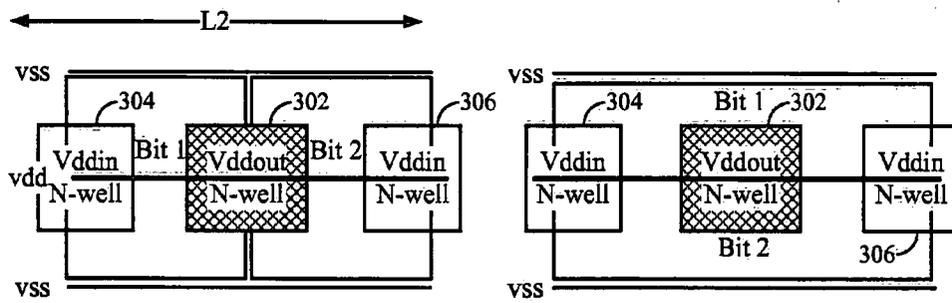
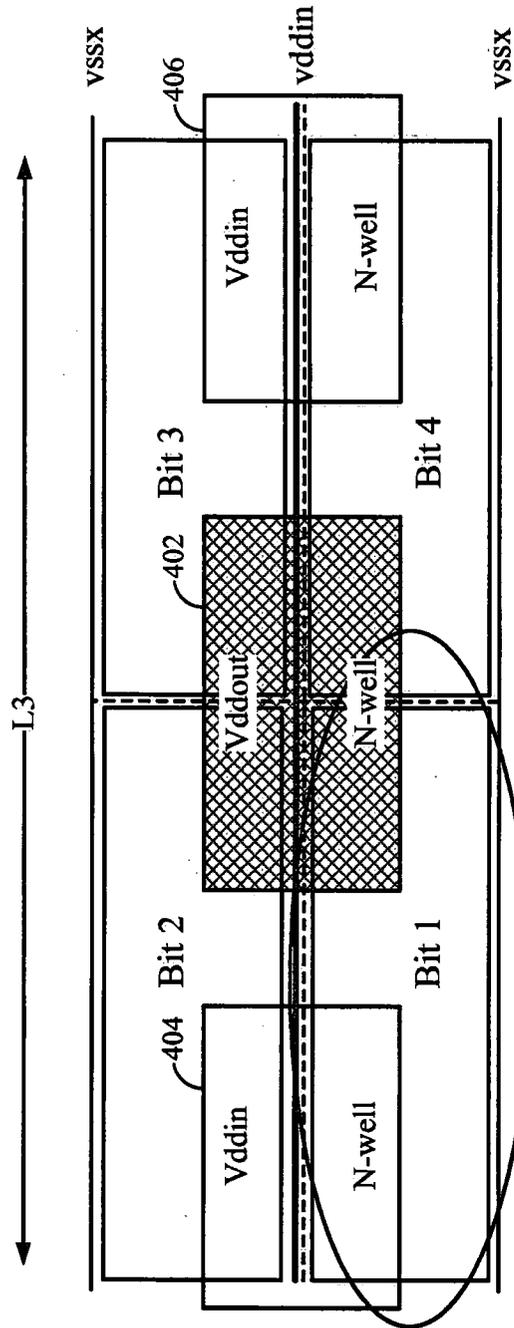


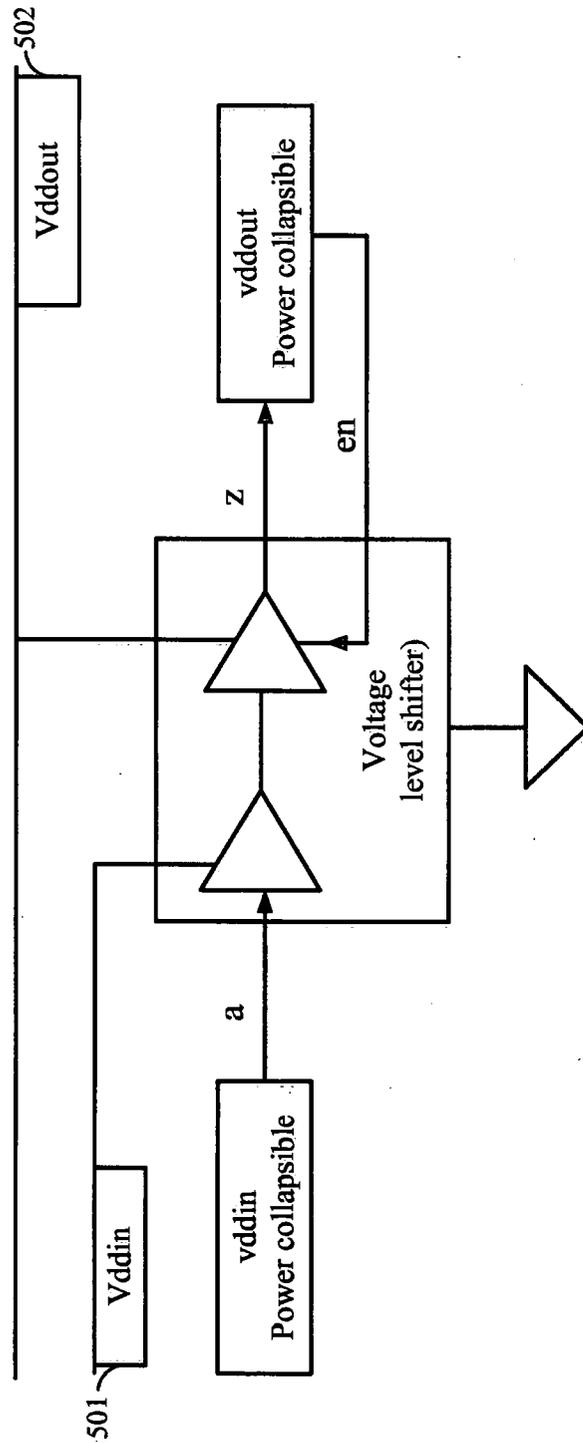
FIG. 3A

FIG. 3B



Compact 4-bit level shifter layout architecture

FIG. 4



PRIOR ART
FIG. 5

1

COMPACT AND ROBUST LEVEL SHIFTER LAYOUT DESIGN

FIELD OF DISCLOSURE

The field of invention relates to a semiconductor device and methods of manufacturing a semiconductor device handling a plurality of voltage, specifically multi-voltage circuits for shifting the voltage level between voltage domains.

BACKGROUND

Integrated circuit devices containing several types of functional circuit are sometimes required to handle a plurality of voltage levels. Such devices are often known as multi-voltage level devices. Multi-voltage level devices contain a high-voltage circuit driven by a relatively high voltage power supply and a low-voltage circuit driven by a relatively low-voltage power supply. Multi-voltage circuits include but are not limited to voltage level shifters (VLS), isolation cell, retention registers, always on logic and similar components.

Power consumption of integrated circuits may be reduced and efficiencies may be increased by reducing operating voltages of the integrated circuits. Some circuits are more amenable to lower operating voltages than others. Where integrated circuits within a system operate at lower voltages, conflicts or contention may arise between the circuits. These conflicts and contention can be alleviated by level shifting the operating voltage of part of the circuits to higher voltage. But level shifting may introduce delays.

Technology scaling reduces the delay of circuit elements, enhancing the operating frequency of an integrated circuit (IC) device. The density and number of transistors on an IC are increased by scaling the feature size. By utilizing this growing number of available transistors in each new technology, novel circuit techniques can be employed further enhancing the performance of the ICs beyond the levels made possible by simply shrinking.

SUMMARY

The described features generally relate to one or more improved systems, methods and/or apparatuses for compact and robust level shifter layout design.

Further scope of the applicability of the described methods and apparatuses will become apparent from the following detailed description, claims, and drawings. The detailed description and specific examples, while indicating specific examples of the disclosure and claims, are given by way of illustration only, since various changes and modifications within the spirit and scope of the description will become apparent to those skilled in the art.

Embodiments of the present invention do not rely on particular transistor level circuit implementation of level shifters and may be applied to any possible level shifter circuit styles. Embodiment of this invention is not limited to only level shifter circuits, and is applicable to any generic multi-voltage circuit's layout design. Embodiments of the present invention seek to provide a VLS that operate for different voltage levels and that provides area and power savings for multi-bit implementation of level shifter design.

Accordingly an embodiment can include a two-bit multi-voltage circuit to shift each of two bits from a first voltage level logic to a second voltage level logic, comprising: a first N-well formed in a substrate; a second N-well formed in the substrate, adjacent to a side of the first N-well; and a third N-well formed in the substrate, adjacent to a side of the first

2

N-well opposite the first N-well; a first one-bit VLS circuit having a portion formed on the first N-well and a portion formed on the second N-well; and a second bit VLS circuit having a portion formed on the first N-well and a portion formed on the third N-well.

Another embodiment can include a four-bit multi-voltage circuit to shift each of four bits from a first voltage level logic to a second voltage level logic, comprising: a first N-well formed in a substrate; a second N-well formed in the substrate, adjacent to a side of the first N-well; a third N-well formed in the substrate, adjacent to a side of the first N-well; a first one-bit VLS circuit having a portion formed on the first N-well and a portion formed on the second N-well; a second bit VLS circuit having a portion formed on the first N-well and a portion formed on the second N-well; a third one-bit VLS circuit having a portion formed on the first N-well and a portion formed on the third N-well; and a fourth one-bit VLS circuit having a portion formed on the first N-well and a portion formed on the third N-well.

Another embodiment can include a method for reducing die area in a two-bit multi-voltage circuit to shift each of two bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, and a third N-well formed in the substrate, adjacent to a side of the first N-well opposite the second N-well, comprising: forming a first one-bit VLS circuit having a portion on the first N-well and a portion formed on the second N-well; and forming a second bit VLS circuit having a portion on the first N-well and a portion formed on the third N-well.

Another embodiment can include an apparatus for reducing die area in a two-bit multi-voltage circuit to shift each of two bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, and a third N-well formed in the substrate, adjacent to a side of the first N-well opposite the second N-well, the apparatus comprising: logic configured to form a first one-bit VLS circuit having a portion on the first N-well and a portion formed on the second N-well; and logic configured to form a second bit VLS circuit having a portion on the first N-well and a portion formed on the third N-well.

Another embodiment can include an apparatus for reducing die area in a two-bit multi-voltage circuit to shift each of two bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, and a third N-well formed in the substrate, adjacent to a side of the first N-well opposite the second N-well, the apparatus comprising: means for forming a first one-bit VLS circuit having a portion on the first N-well and a portion formed on the second N-well; and means for forming a second bit VLS circuit having a portion on the first N-well and a portion formed on the third N-well.

Another embodiment can include a method for reducing die area in a four-bit multi-voltage circuit to shift each of four bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, a third N-well formed in the substrate, adjacent to a side of the first N-well, comprising: forming a first one-bit VLS circuit having a portion on the first N-well and a portion formed on the second N-well; forming a second bit VLS circuit having a portion on the first N-well and a portion formed on the second N-well; forming a third one-bit VLS circuit having a portion on the first N-well and a portion formed on the third

3

N-well; and forming a fourth one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well.

Another embodiment can include an apparatus for reducing die area in a four-bit multi-voltage circuit to shift each of four bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, a third N-well formed in the substrate, adjacent to a side of the first N-well, the apparatus comprising: logic configured to form a first one-bit VLS circuit having a portion on the first N-well and a portion formed the second N-well; logic configured to form a second bit VLS circuit having a portion on the first N-well and a portion formed on the second N-well; logic configured to form a third one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well; and logic configured to form a fourth one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well.

Another embodiment can include apparatus for reducing die area in a four-bit multi-voltage circuit to shift each of four bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, a third N-well formed in the substrate, adjacent to a side of the first N-well, the apparatus comprising: means for forming a first one-bit VLS circuit having a portion on the first N-well and a portion formed the second N-well; means for forming a second bit VLS circuit having a portion on the first N-well and a portion formed on the second N-well; means for forming a third one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well; and means for forming a fourth one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings are presented to aid in the description of embodiments of the invention and are provided solely for illustration of the embodiments and not limitation thereof.

FIG. 1 is a conventional 1-bit voltage level shifter.

FIG. 2 is a conventional 2-bit voltage level shifter.

FIG. 3A is a 2-bit voltage level shifter according to an embodiment of the present invention.

FIG. 3B is a 2-bit voltage level shifter according to another embodiment of the present invention.

FIG. 4 is a 4-bit voltage level shifter according to another embodiment of the present invention.

FIG. 5 illustrates a generic 1-bit voltage level shifter functional circuit, whose physical design or layout can be implemented as 1-bit level shifter in any of the embodiments.

DETAILED DESCRIPTION

Aspects of the invention are disclosed in the following description and related drawings directed to specific embodiments of the invention. Alternate embodiments may be devised without departing from the scope of the invention. Additionally, well-known elements of the invention will not be described in detail or will be omitted so as not to obscure the relevant details of the invention. Embodiments of the disclosure may be suitably employed in any device which includes active integrated circuitry including memory and on-chip circuitry for test and characterization.

The foregoing disclosed devices and methods are typically designed and are configured into GDSII and GERBER com-

4

puter files, stored on a computer readable media. These files are in turn provided to fabrication handlers who fabricate devices based on these files. The resulting products are semiconductor wafers that are then cut into semiconductor die and packaged into a semiconductor chip. The chips are then employed in devices described above.

The word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments. Likewise, the term "embodiments of the invention" does not require that all embodiments of the invention include the discussed feature, advantage or mode of operation.

The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of embodiments of the invention. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises", "comprising", "includes" and/or "including", when used herein, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

Further, many embodiments are described in terms of sequences of actions to be performed by, for example, elements of a computing device. It will be recognized that various actions described herein can be performed by specific circuits (e.g., application specific integrated circuits (ASICs)), by program instructions being executed by one or more processors, or by a combination of both. Additionally, these sequence of actions described herein can be considered to be embodied entirely within any form of computer readable storage medium having stored therein a corresponding set of computer instructions that upon execution would cause an associated processor to perform the functionality described herein. Thus, the various aspects of the invention may be embodied in a number of different forms, all of which have been contemplated to be within the scope of the claimed subject matter. In addition, for each of the embodiments described herein, the corresponding form of any such embodiments may be described herein as, for example, "logic configured to" perform the described action.

The power density of IC chips is increasing to support more features and various operating modes in portable electronic devices, especially for deep submicron technology. Deep submicron technology uses transistors of smaller size with faster switching rates (e.g., 45 nm and smaller nodes). In the IC chips of a portable electronic device, such as mobile and cellular, having dynamic supply voltage (V_{DD}) and frequency scaling can be a technique for active power (P) reduction due to square dependence of V_{DD} (i.e., $P \propto V_{DD}^2$). Therefore, IC chips employ different voltage domains for different circuit blocks. Reasons include optimizing trade-offs among, for example, speed, noise tolerance and power consumption to account for different circuit blocks having different priorities.

However, decreasing the operational voltage level of one circuit in a system can create compatibility problems where some other integrated circuit or other device is designed to operate at predetermined incompatible specific voltage level, or is accessible only via a bus that operates optimally at a different (e.g., higher) voltage logic level. For example, some circuits within a chip may operate at low voltage core-logic level to reduce power consumption and to interface with other chips operating at the same low voltage, while other circuits in the same chip may operate at higher voltage levels to

interface with a higher logic voltage chip or bus or to operate an electro-mechanical device. Also, there are many existing integrated circuits that cannot have their operating voltage altered; yet, newer lower voltage circuits must interface with them. For example, if core logic voltage were reduced from a nominal 1.2 volts to 0.7 V, the logic value represented by 0.7 V would ordinarily be insufficient to properly drive another transistor circuit operating from a 1.2 V power supply. The 0.7 V logic input to a 1.2 volt CMOS circuit would cause a prolonged transitional (i.e., conducting) state potentially resulting in damaging currents in the CMOS circuits tied to a 1.2 volt supply. The rise, fall, and propagation times of signals would be detrimentally affected by the difference between core logic voltage and the circuits operating at a higher logic voltage. Therefore, to lower the voltage of integrated circuits and to consume less power, while still enabling their interaction with existing hardware components operating at a different voltage, some form of voltage level-shifting interface circuit (e.g., level-shifting buffer circuit) is required.

Consequently, many complementary metal oxide semiconductor (CMOS) integrated circuits require more than one power supply per chip. For instance, a split rail design is utilized when the internal or core-logic voltage, V_{DDin} , operates at a different (e.g., lower) voltage level than the input/output (I/O) interface voltage or output driver voltage, V_{DDout} . The integrated circuit core voltage, V_{DDin} , applied to a given circuit may be fixed or variable depending on the integrated circuit technology, design factors, and by the performance requirements and the power supply and heat dissipation characteristics of the chip.

As a result, a signal traversing from one voltage domain to another must pass through a multi-voltage circuit or voltage level shifter (VLS) cell to maintain its logical value. Multi-voltage circuits include but are not limited to VLS, isolation cell, retention registers, always on logic and other similar components. To reduce chip power consumption and increase battery life, portable electronic device chip-sets employ a large number of VLS cells. However, this necessitates very compact level shifters design to limit the die area overhead. To reduce chip power dissipation, VLS cells need to consume lower static power and keep robust functionally. This requires reliable operation across wider range of input and output voltages without consuming extra power.

VLS cells are known to convert a signal from one voltage domain to a signal suitable for another voltage domain. A conventional VLS cell converts signals between an input domain (e.g., V_{DDin}) and an output domain (e.g., V_{DDout}). In addition, a conventional VLS cell can prevent excessive leakage to improve battery life and allow reliable functionality across a wide range of voltage domains. The attached related art FIG. 1 shows typical layout (Physical Design) structure of 1-bit (with single input signal) VLS cell which is can be placed next to another cell of input voltage domain (V_{DDin}). The FIG. 1 conventional one-bit VLS circuit requires three N-wells, two V_{DDin} N-well 102 and 106 and a V_{DDout} N-well 104 for bit 0. This allows circuits that work at different voltages to properly interface with each other without additional leakage power.

Since the adjacent N-wells in the FIG. 1 VLS have alternating different voltages, design rules require these adjacent N-wells have a minimum spacing for correct functional operation. For example, as shown in FIG. 1, the minimum spacing between the two N-wells 102 and 104 is 0.8 μm . In addition the minimum spacing between the other two N-wells 104 and 106 is also 0.8 μm .

The attached related art FIG. 5 depicts a functional circuit diagram of a conventional 1-bit VLS layout. A conventional

VLS cell can employ two stage complementary metal-oxide-semiconductor (CMOS) circuits, with a first stage operating at a first voltage 501, as shown by V_{DDin} in FIG. 5, and a second stage operating at a second voltage 502, as shown by V_{DDout} . When their threshold voltages and device strengths are properly adjusted, they can perform voltage level shifting as desired. However, the conventional VLS may occupy large layout areas because a first N-well for a CMOS transistor in the first stage is coupled to a first voltage, while a second N-well for a CMOS transistor in the second state is coupled to a second voltage, therefore the first and second N-wells have to be separated and have to maintain a certain distance, which is determined by the technology being used.

The attached related art FIG. 2 shows a conventional N-well arrangement for a conventional two-bit V_{DDin} (e.g., 0.7 V) to V_{DDout} (e.g., 1.2 V) VLS circuit. The FIG. 2 conventional two-bit VLS circuit requires five N-wells, namely one shared V_{DDin} N-well 206, a V_{DDin} N-well 202 and a V_{DDout} N-well 204 for bit 0, and V_{DDin} N-well 208 and V_{DDin} N-well 210 for bit 1. Similar to FIG. 1, this example illustrates that a minimum spacing between each N-wells 202, 204, 206, 208 and 210 to be at least 0.8 μm .

Because the adjacent N-wells in the FIG. 2 VLS have alternating different voltages, design rules require these adjacent N-wells have a minimum spacing. The related art FIG. 2 shows, for example, a minimum spacing of 0.8 μm . Notably, this minimum spacing requirement between VLS adjacent N-wells does not fully scale with the feature size. For example, if an IC implemented in 65 nm technology is scaled down to 32 nm (i.e., scaled down by approximately half), the minimum spacing between adjacent N-wells of the IC's VLS circuits does not likewise scale by half.

Therefore, due to non-shrinking latch-up design rules (N-well to N-well spacing) and presence of multiple V_{DD} domains, physical design of a VLS circuit consumes large die area. In addition, due to presence of three separate N-well regions in a VLS circuit, the physical area of level shifter cell does not shrink proportionally (expected area scaling is ~50%) when technology nodes get smaller as shown in table 1. This becomes even more apparent in 32 nm and smaller process node.

TABLE 1

Area scaling of level shifter layout design		
Technology node	Cell area (sq. μm)	Area scaling from previous node
65 nm	21.6	NA
45 nm	12.3	57%
32 nm	7.6	61%

Physical design or layout structure of level shifter circuit incurs significant area overhead compared to regular CMOS logic. In particular when multiple level shifter instances are placed together they result in multiple N-well voltage islands as shown in FIG. 2. Larger layout footprint increases length of both the internal and external interconnects length increasing dynamic power as well as die area. There is no known method technique to escape from fixed area overhead in the physical design of the level shifter. In 32 nm and smaller technology nodes the presence of strong layout proximity effects further increases area overhead of level shifter cells due to presence of protective layout structures.

Accordingly there is a need for VLS to have a reduced area and is cost efficient to fabricate. Embodiments of the present invention seek to provide a VLS that operate for different voltage levels and that provides area and power savings for

multi-bit implementation of level shifter design. Embodiments of the present invention do not rely on particular transistor level circuit implementation of level shifters and may be applied to any possible level shifter circuit styles. Embodiments of the present invention are not limited to level shifter circuits, and are applicable to any generic layout design for multi-voltage circuits. The present invention can be extended to any circuit requiring two or more different voltage domains and hence multiple N-well islands in the design.

Embodiments of the present invention relate to a physical design methodology for compact layout of VLS cells in multiple bits. The present invention utilizes the presence of separate voltage N-well islands inside a cell. Several key physical designs are used to improve design robustness and lower the power dissipation. By reducing the N-well to N-well spacing occurrence, the present invention can allow for a more compact layout of VLS cells in multiple bits. The present invention can allow for improved design robustness and low voltage performance, which includes better V_{DDmin} .

FIGS. 3A and 3B shows two embodiments of the present invention. These embodiments use area efficient physical design for a 2-bit VLS with a pair of input and output signals. FIGS. 3A and 3B illustrate example of large area saving which is achieved by merging two identical 1-bit level shifter layouts differently than a conventional 1-bit layout, as illustrated in FIG. 1. In the embodiments illustrated in FIGS. 3A and 3B, the 1-bit component layouts are not self-sufficient and cannot be used solely as a compact 1-bit level shifter layout design. Therefore, the embodiments illustrated in FIGS. 3A and 3B are examples for a 2-bit VLS, which include two combined component 1-bit level shifter layouts. In FIG. 3A, the component 1-bit level shifter layouts is composed of one N-well separation between the N-wells and two standard cell row, which has two V_{SS} and one V_{DD} power rail. In FIG. 3B, the component 1-bit level shifter layouts is composed of two N-well separations between the N-wells and one standard cell row, which has one V_{SS} and one V_{DD} power rail. As shown in table 2, in both embodiments, there can be a forty percent area savings as compared to two separate 1-bit VLS cells as illustrated in FIG. 2.

This embodiment, as illustrated at FIG. 3A forms a two-bit VLS circuit using three (3) N-wells, instead of the five (5) required in the conventional arrangements shown at FIG. 2. The overall width, L2, is approximately half ($\frac{1}{2}$) of the L1 width of the conventional arrangement shown in FIG. 2. The FIG. 3A embodiment uses a center N-well 302, biased at V_{DDout} , and one N-well biased at V_{DDin} on either side of the center N-well 304, namely N-well 304 at the left and N-well 306 at the right. The spacing SP1 between the differently biased N-wells 304 and 302 is the same as between the differently biased N-wells 302 and 306. The spacing SP1 may be substantially the same as the spacing between adjacent N-wells in the conventional arrangement of the related art FIG. 2.

The FIG. 3A embodiment employs the center N-well 302, biased at V_{DDout} , for the V_{DDout} portion of both the bit 1 and the bit 2 VLS sections of the two-bit VLS circuit. The FIG. 3A embodiment employs the left N-well 304 for only the bit 1 VLS section, and the right N-well 306 for only the bit 2 VLS section. FIG. 3B shows another embodiment, having the same N-well arrangement as the FIG. 3A embodiment, but employing all three N-wells 302, 304 and 306 for both the bit 1 VLS section and the bit 2 VLS section.

In the embodiments shown in FIG. 3A and FIG. 3B, large area saving is achieved by merging two identical 1-bit level shifter layouts, as compared to the conventional 1-bit layout as depicted in FIG. 1. However, the embodiments shown in

FIG. 3A and FIG. 3B, the 1-bit component layouts are not self-sufficient and cannot be used as compact 1-bit level shifter layout design. To summarize the component 1-bit level shifter layouts, from FIG. 3A comprises of only one N-well separation between them and two standard cell row, which has two V_{SS} and one V_{DD} power rail. The component 1-bit level shifter layouts, from FIG. 3B comprises of two N-well separations between them and of only one standard cell row, which has one V_{SS} and one V_{DD} power rail. In the embodiments of FIG. 3A and FIG. 3B, there can be a forty percent area savings as compared to two separate 1-bit cells as depicted FIG. 2.

In another embodiment, FIG. 4 illustrates a physical design method of a 4-bit VLS. As shown in table 2, with the FIG. 4 embodiment, there can be a fifty percent area savings as compared to four separate 1-bit VLS cells as illustrated by having two 2-bit VLS, as shown in FIG. 2, side-by-side. In this embodiment, the component 1-bit level shifter layouts are composed of only one N-well separation between the N-wells and of only 1 standard cell row, which has one V_{SS} and one V_{DD} power rail per bit. Each component 1-bit level shifter is merged both vertically and horizontally together to form a design rule checker clean physical design of a 4-bit VLS. The 1-bit component layout is not design rule checker clean and cannot be used as compact 1-bit level shifter layout design by itself.

Additional advantages of the embodiment depicted in FIG. 4 are that in a multi-bit design, shared clamp or isolation circuitry, if needed can further reduce area and power saving. Also, symmetrical design and placement of critical devices can ensure better design robustness and low voltage performance from device variation (e.g., in a V_{DDmin} point of view). Smaller length of interconnects between different bits reduces the capacitive loading and hence the dynamic power dissipation of this embodiment compared to conventional physical design implementation.

With reference to FIG. 4, the VLS is essentially a superposed combination of the FIGS. 3A and 3B embodiments, achieving a four-bit VLS with the same general arrangement of three N-wells. The FIG. 4 embodiment employs the center N-well 402 for all four one-bit sections of the VLS, i.e., for each of the bit 1, bit 2, bit 3 and bit 4 VLS sections. The left N-well 404 is employed for both the bit 1 and bit 2 VLS sections, similar to the left N-well 304 of the FIG. 3B embodiment being employed for the bit 1 and bit 2 sections of its two-bit VLS. Likewise, the right N-well 406 of the FIG. 4 embodiment is employed for both the bit 3 and bit 4 VLS sections of its four-bit VLS, similar to the right N-well 306 of the FIG. 3B embodiment being employed, as described above, for the bit 1 and bit 2 sections of its two-bit VLS. The spacing SP2 between the differently biased N-wells 402 and 404 may be the same as between the differently biased N-wells 404 and 406, and may be substantially the same as SP1 of the FIGS. 3A and 3B embodiments. The N-wells 402, 404 and 406 may be somewhat larger in area than the N-wells 302, 304 and 306 of the FIGS. 3A and 3B embodiments.

The overall length L3 may be approximately twice L2 of the FIGS. 3A and 3B embodiments. The FIG. 4 embodiment may be approximately the same area size as the conventional layout shown in FIG. 2, which is a two-bit VLS comprising of five N-wells.

TABLE 2

Area saving with compact multi-bit level shifter layout methodology in recent semiconductor manufacturing technologies		
Cell Type	45 nm area saving	32 nm area saving
2-bit	40%	45%
4-bit	52%	55%

The present invention allows for a compact hierarchical physical design methodology to improve multi-rail VLS standard cell design. The present invention does not rely on any particular circuit implementation of level shifter and can be applied to all possible level shifter circuit styles. The present invention can consist of stitching together to produce an error free design from several symmetrical components in different orientation that are not individually self-sufficient in design. The present invention is not limited to voltage level shifter cells, and can be applied to a number of different cell families that involves multiple N-well islands. The present invention is not limited to any particular standard cell architecture and can easily be adapted to different cell architecture and style. The present invention can be scalable to future process technology nodes with smaller geometries and larger context sensitivity.

The present invention can be successfully employed in two different standard cell architecture, such as, but not limited to 45 nm and 32 nm technology nodes. The present invention can demonstrate significant area (e.g., over 50%) and power savings for multi-bit implementation of VLS design technologies, including but not limited to 45 nm, 32 nm and smaller process technologies.

Those of skill in the art will appreciate that information and signals may be represented using any of a variety of different technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description may be represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof.

Further, those of skill in the art will appreciate that the various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present invention.

The methods, sequences and/or algorithms described in connection with the embodiments disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary storage medium is coupled to the processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor.

Accordingly, an embodiment of the invention can include a computer readable media embodying a method for compact and robust voltage level shifters design. Accordingly, the invention is not limited to illustrated examples and any means for performing the functionality described herein are included in embodiments of the invention.

While the foregoing disclosure shows illustrative embodiments of the invention, it should be noted that various changes and modifications could be made herein without departing from the scope of the invention as defined by the appended claims. The functions, steps and/or actions of the method claims in accordance with the embodiments of the invention described herein need not be performed in any particular order. Furthermore, although elements of the invention may be described or claimed in the singular, the plural is contemplated unless limitation to the singular is explicitly stated.

What is claimed is:

1. A multi-voltage circuit to shift each of two bits from a first voltage level logic to a second voltage level logic, comprising:

- a first N-well formed in a substrate;
 - a second N-well formed in the substrate, adjacent to a side of the first N-well;
 - a third N-well formed in the substrate, adjacent to a side of the first N-well opposite the second N-well;
 - a first one-bit voltage level shift (VLS) circuit having a portion formed on the first N-well and a portion formed on the second N-well; and
 - a second one-bit VLS circuit having a portion formed on the first N-well and a portion formed on the third N-well;
- the first one-bit VLS circuit and the second one-bit VLS circuit to provide a pair of output signals at the first N-well.

2. The multi-voltage circuit of claim 1, wherein the multi-voltage circuit is a voltage level shifter, an isolation cell, a retention register, or an always on logic component integrated in at least one semiconductor die.

3. The multi-voltage circuit of claim 1, wherein there is significant die area reduction and switching power saving, due to reduced length of interconnects inside cell and reduced length of top level connection, as compared to two separate 1-bit conventional multi-voltage circuit cells side-by-side.

4. The multi-voltage circuit of claim 1, wherein the first, second and third N-wells are arranged in a row with the first N-well at a center position.

5. The multi-voltage circuit of claim 1, wherein the first N-well is biased at the second voltage level, and the second and third N-wells are biased at the first voltage level.

6. The multi-voltage circuit of claim 5, wherein multi-voltage has one N-well separation between the N-wells and is comprising of two standard cell row, which includes two V_{SS} and one V_{DD} power rail.

7. The multi-voltage circuit of claim 1, wherein the first N-well is biased at the first voltage level, and the second N-well and third N-well is biased at the second voltage level.

8. The multi-voltage circuit of claim 7, wherein multi-voltage has N-well separations between the N-wells and is comprising of one standard cell row, which includes one V_{SS} and one V_{DD} power rail.

9. A four-bit multi-voltage circuit to shift each of four bits from a first voltage level logic to a second voltage level logic, comprising:

- a first N-well formed in a substrate;
- a second N-well formed in the substrate, adjacent to a side of the first N-well;
- a third N-well formed in the substrate, adjacent to a side of the first N-well opposite the second N-well;

11

a first one-bit voltage level shift (VLS) circuit having a portion formed on the first N-well and a portion formed the second N-well;

a second one-bit VLS circuit having a portion formed on the first N-well and a portion formed on the second N-well;

a third one-bit VLS circuit having a portion formed on the first N-well and a portion formed the third N-well; and
a fourth one-bit VLS circuit having a portion formed on the N-well first and a portion formed the third N-well.

10. The four-bit multi-voltage circuit of claim 9, wherein the multi-voltage circuit is a voltage level shifter, an isolation cell, a retention register, or an always on logic component.

11. The four-bit multi-voltage circuit of claim 9, wherein there is more than a fifty percent area reduction and significant switching power savings as compared to four separate 1-bit multi-voltage circuit cells in a square layout formation.

12. The four-bit multi-voltage circuit of claim 9, wherein the first, second and third N-wells are arranged in a row with the first N-well at a center position.

13. The four-bit multi-voltage circuit of claim 9, wherein the first N-well is biased at the second voltage level, and the second and third N-wells are biased at the first voltage level.

14. The four-bit multi-voltage circuit of claim 9, wherein the first N-well is biased at the first voltage level, and the second and third N-wells are biased at the second voltage level.

15. The four-bit multi-voltage circuit of claim 9, wherein each 1-bit level shifter layouts in the multi-voltage circuit is composed of only one N-well separation between the N-wells and of only one standard cell row, which includes one V_{SS} and one V_{DD} power rail per bit.

16. A method for reducing die area, and switching power in a two-bit multi-voltage circuit to shift each of two bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, and a third N-well formed in the substrate, adjacent to a side of the first N-well opposite the second N-well, comprising:

forming a first one-bit voltage level shift (VLS) circuit having a portion on the first N-well and a portion formed on the second N-well; and

forming a second one-bit VLS circuit having a portion on the first N-well and a portion formed on the third N-well; the first one-bit VLS circuit and the second one-bit VLS circuit to provide a pair of output signals at the first N-well.

17. An apparatus for reducing die area, and switching power in a two-bit multi-voltage circuit to shift each of two bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, and a third N-well formed in the substrate, adjacent to a side of the first N-well opposite the second N-well, the apparatus comprising:

logic configured to form a first one-bit voltage level shift (VLS) circuit having a portion on the first N-well and a portion formed on the second N-well; and

logic configured to form a second one-bit VLS circuit having a portion on the first N-well and a portion formed on the third N-well;

the first one-bit VLS circuit and the second one-bit VLS circuit to provide a pair of output signals at the first N-well.

18. An apparatus for reducing die area in a two-bit multi-voltage circuit to shift each of two bits from a first voltage level logic to a second voltage level logic, wherein a first

12

N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, and a third N-well formed in the substrate, adjacent to a side of the first N-well opposite the second N-well, the apparatus comprising:

means for forming a first one-bit voltage level shift (VLS) circuit having a portion on the first N-well and a portion formed on the second N-well; and

means for forming a second one-bit VLS circuit having a portion on the first N-well and a portion formed on the third N-well;

the first one-bit VLS circuit and the second one-bit VLS circuit to provide a pair of output signals at the first N-well.

19. An apparatus for reducing die area in a two-bit multi-voltage circuit to shift each of two bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, and a third N-well formed in the substrate, adjacent to a side of the first N-well opposite the second N-well, the apparatus comprising:

step for forming a first one-bit voltage level shift (VLS) circuit having a portion on the first N-well and a portion formed on the second N-well; and

step for forming a second one-bit VLS circuit having a portion on the first N-well and a portion formed on the third N-well;

the first one-bit VLS circuit and the second one-bit VLS circuit to provide a pair of output signals at the first N-well.

20. A method for reducing die area, and switching power in a four-bit multi-voltage circuit to shift each of four bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, a third N-well formed in the substrate, adjacent to a side of the first N-well, comprising:

forming a first one-bit voltage level shift (VLS) circuit having a portion on the first N-well and a portion formed the second N-well;

forming a second one-bit VLS circuit having a portion on the first N-well and a portion formed on the second N-well;

forming a third one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well; and
forming a fourth one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well.

21. An apparatus for reducing die area, and switching power in a four-bit multi-voltage circuit to shift each of four bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, a third N-well formed in the substrate, adjacent to a side of the first N-well, the apparatus comprising:

logic configured to form a first one-bit voltage level shift (VLS) circuit having a portion on the first N-well and a portion formed the second N-well;

logic configured to form a second one-bit VLS circuit having a portion on the first N-well and a portion formed on the second N-well;

logic configured to form a third one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well; and

logic configured to form a fourth one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well.

13

22. An apparatus for reducing die area, and switching power in a four-bit multi-voltage circuit to shift each of four bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, a third N-well formed in the substrate, adjacent to a side of the first N-well, the apparatus comprising:

means for forming a first one-bit voltage level shift (VLS) circuit having a portion on the first N-well and a portion formed the second N-well;

means for forming a second one-bit VLS circuit having a portion on the first N-well and a portion formed on the second N-well;

means for forming a third one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well; and

means for forming a fourth one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well.

14

23. An apparatus for reducing die area, and switching power in a four-bit multi-voltage circuit to shift each of four bits from a first voltage level logic to a second voltage level logic, wherein a first N-well formed in a substrate, a second N-well formed in the substrate, adjacent to a side of the first N-well, a third N-well formed in the substrate, adjacent to a side of the first N-well, the apparatus comprising:

step for forming a first one-bit voltage level shift (VLS) circuit having a portion on the first N-well and a portion formed the second N-well;

step for forming a second one-bit VLS circuit having a portion on the first N-well and a portion formed on the second N-well;

step for forming a third one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well; and

step for forming a fourth one-bit VLS circuit having a portion on the first N-well and a portion formed the third N-well.

* * * * *



EXHIBIT D



United States
Government

3894982

8,838,949 B2
Sep. 16, 2014

U 7629935

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

**UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office**

April 06, 2017

**THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM
THE RECORDS OF THIS OFFICE OF:**

**U.S. PATENT: 8,838,949
ISSUE DATE: September 16, 2014**

**By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office**

**T. LAWRENCE
Certifying Officer**





US008838949B2

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

(10) **Patent No.:** **US 8,838,949 B2**
(45) **Date of Patent:** **Sep. 16, 2014**

(54) **DIRECT SCATTER LOADING OF EXECUTABLE SOFTWARE IMAGE FROM A PRIMARY PROCESSOR TO ONE OR MORE SECONDARY PROCESSOR IN A MULTI-PROCESSOR SYSTEM**

(58) **Field of Classification Search**
CPC G06F 9/4405; G06F 9/445; G06F 15/177
USPC 713/1, 2, 100; 712/E9.003, 30
See application file for complete search history.

(75) **Inventors:** Nitin Gupta, San Diego, CA (US);
Daniel H. Kim, San Diego, CA (US);
Igor Malamant, San Diego, CA (US);
Steve Haehnichen, San Diego, CA (US)

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,978,589 A 11/1999 Yoon
6,079,017 A 6/2000 Han et al.
7,447,846 B2 11/2008 Yeh

(Continued)

FOREIGN PATENT DOCUMENTS

EP 2034416 A1 3/2009
JP S63233460 A 9/1988—

(Continued)

OTHER PUBLICATIONS

International Search Report and Written Opinion—PCT/US2011/029484—ISA/EPO—May 30, 2011.

Primary Examiner — M Elamin

(74) *Attorney, Agent, or Firm* — Peter Michael Kamarchik; Nicholas J. Pauley; Joseph Augusta

(57) **ABSTRACT**

In a multi-processor system, an executable software image including an image header and a segmented data image is scatter loaded from a first processor to a second processor. The image header contains the target locations for the data image segments to be scatter loaded into memory of the second processor. Once the image header has been processed, the data segments may be directly loaded into the memory of the second processor without further CPU involvement from the second processor.

23 Claims, 5 Drawing Sheets

(*) **Notice:** Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 362 days.

(21) **Appl. No.:** 13/052,516

(22) **Filed:** Mar. 21, 2011

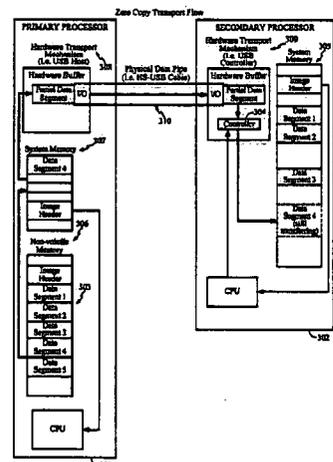
(65) **Prior Publication Data**
US 2012/0072710 A1 Mar. 22, 2012

Related U.S. Application Data

(60) Provisional application No. 61/324,035, filed on Apr. 14, 2010, provisional application No. 61/316,369, filed on Mar. 22, 2010, provisional application No. 61/324,122, filed on Apr. 14, 2010, provisional application No. 61/325,519, filed on Apr. 19, 2010.

(51) **Int. Cl.**
G06F 15/177 (2006.01)
G06F 9/445 (2006.01)
G06F 9/44 (2006.01)

(52) **U.S. Cl.**
CPC *G06F 15/177* (2013.01); *G06F 9/445* (2013.01); *G06F 9/4405* (2013.01)
USPC 713/2; 713/1; 713/100; 712/E9.003; 712/30



(56)

References Cited

U.S. PATENT DOCUMENTS

7,765,391 B2 7/2010 Uemura et al.
 2002/0138156 A1 9/2002 Wong et al.
 2009/0204751 A1 8/2009 Kushita
 2010/0077130 A1 3/2010 Kwon
 2011/0035575 A1* 2/2011 Kwon 713/2
 2012/0089814 A1 4/2012 Gupta et al.

JP H08161283 A 6/1996
 JP H09244902 A 9/1997
 JP 2000020492 A 1/2000
 JP 2004086447 A 3/2004
 JP 2004252990 A 9/2004
 JP 2005122759 A 5/2005
 JP 2007157150 A 6/2007
 KR 20070097538 A 10/2007
 WO WO2006077068 A2 7/2006
 WO 2008001671 A1 1/2008
 WO 2011119648 A1 9/2011

FOREIGN PATENT DOCUMENTS

JP H06195310 A 7/1994

* cited by examiner

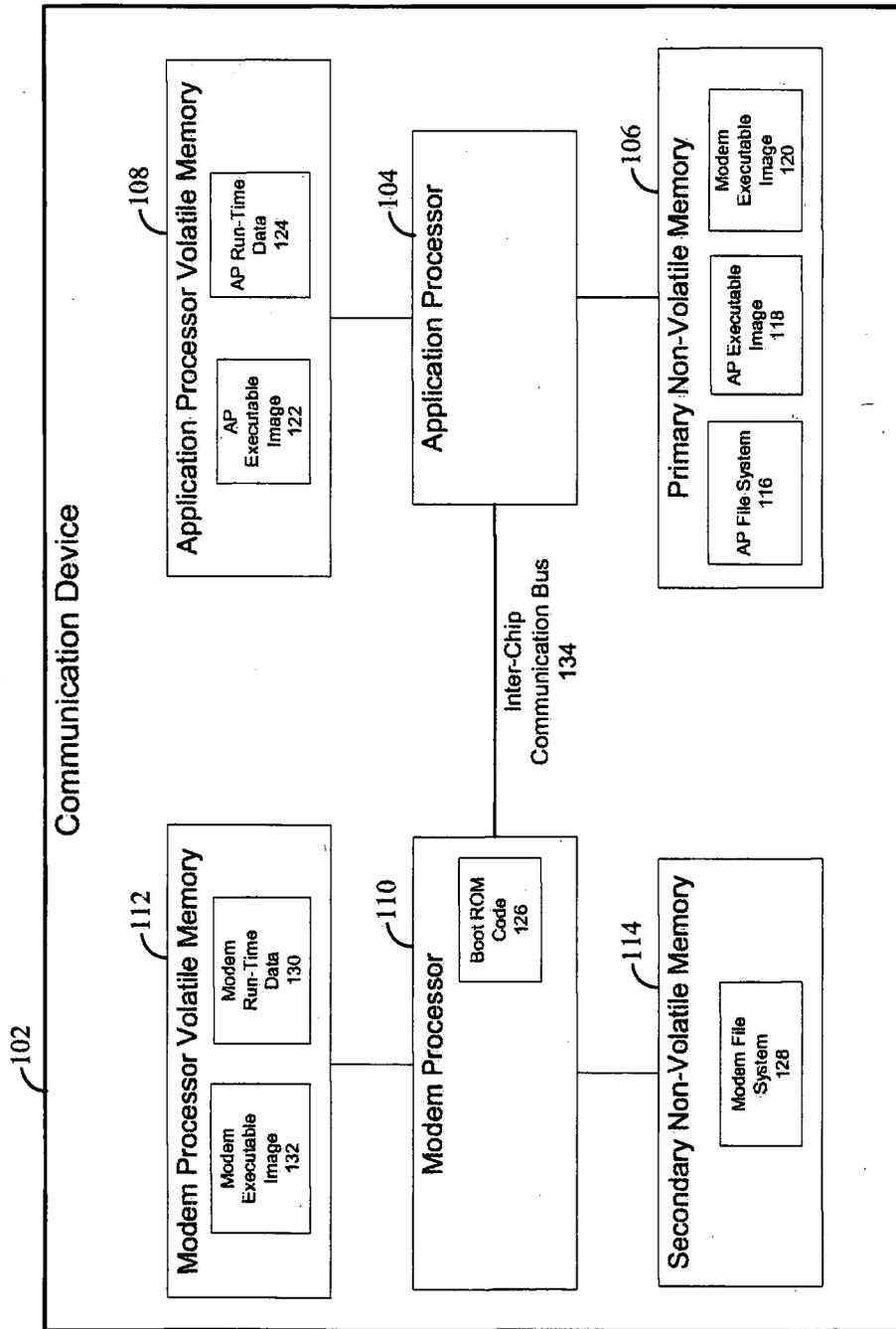


FIG. 1

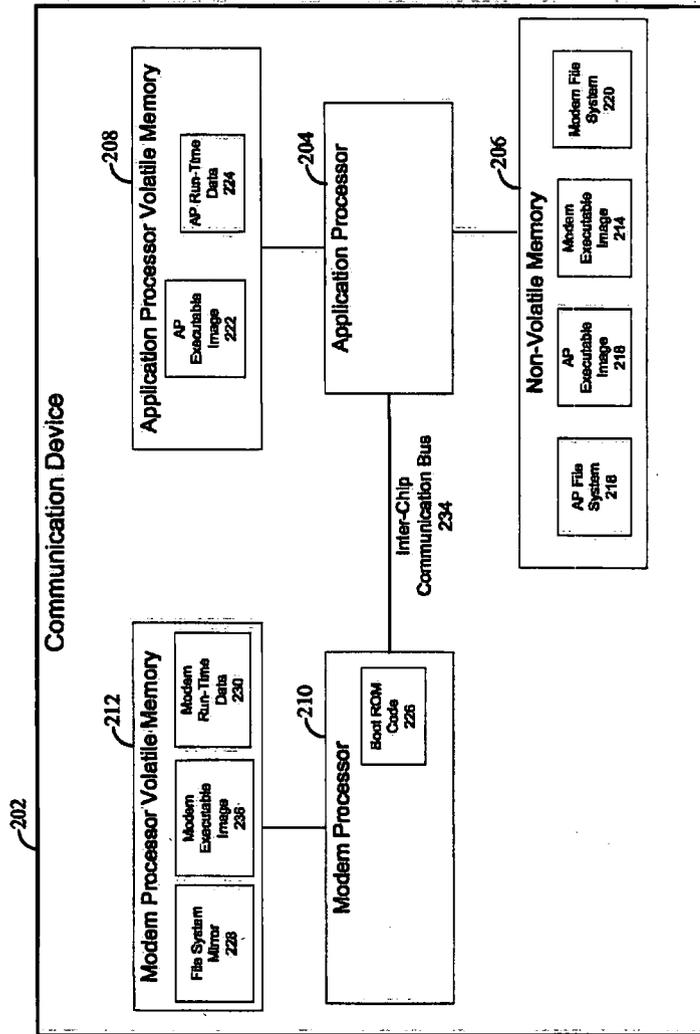


FIG. 2

Zero Copy Transport Flow

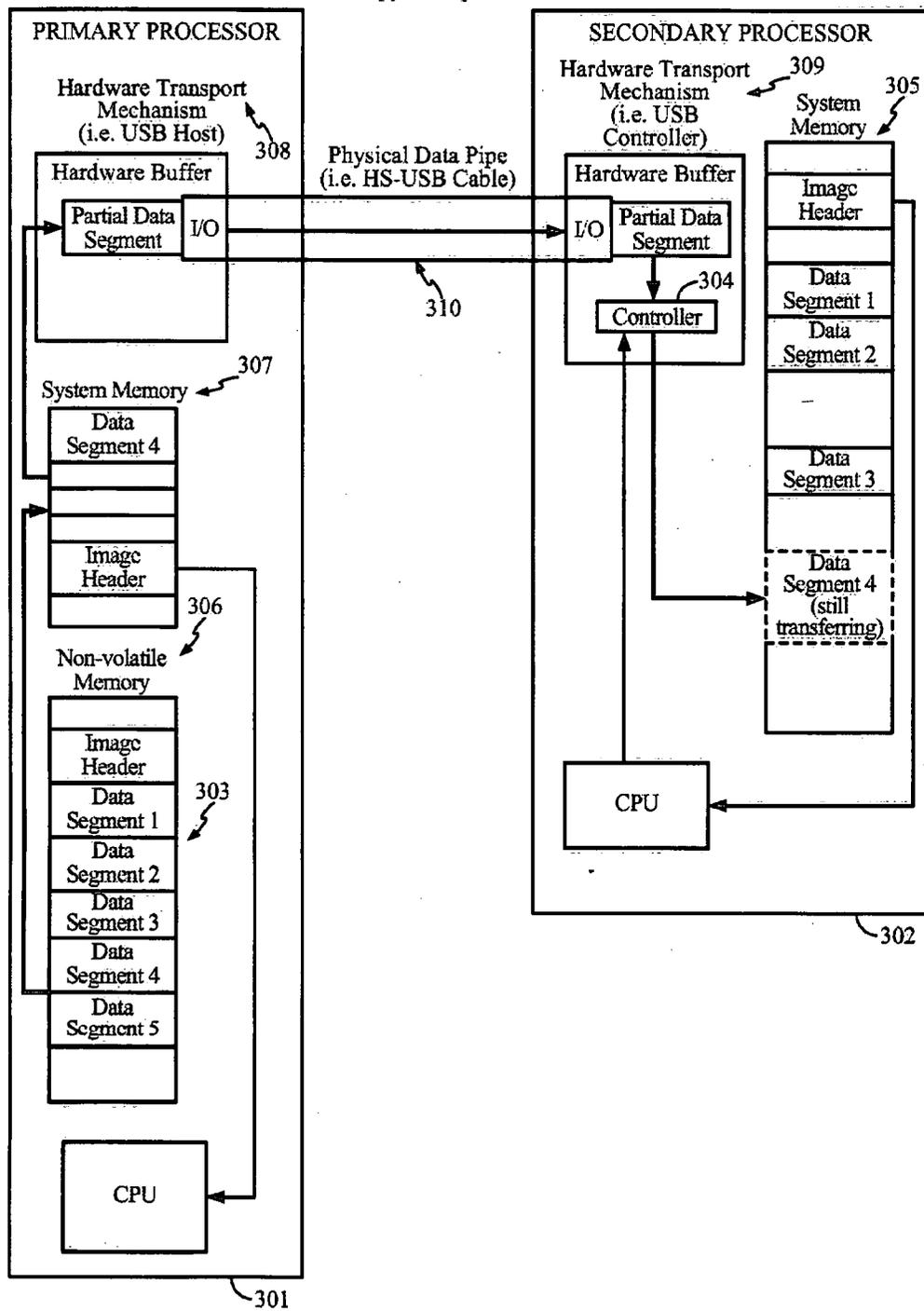


FIG. 3

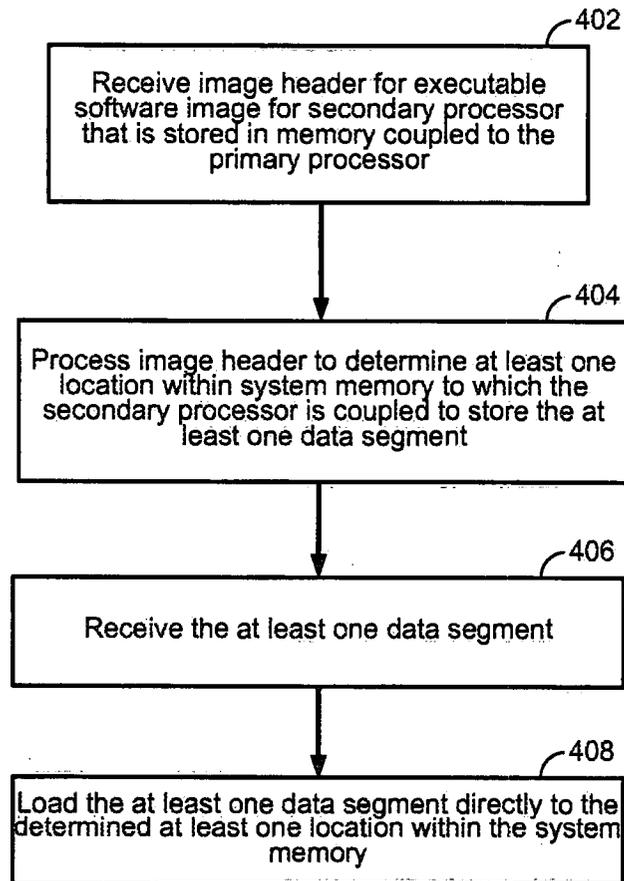


FIG. 4

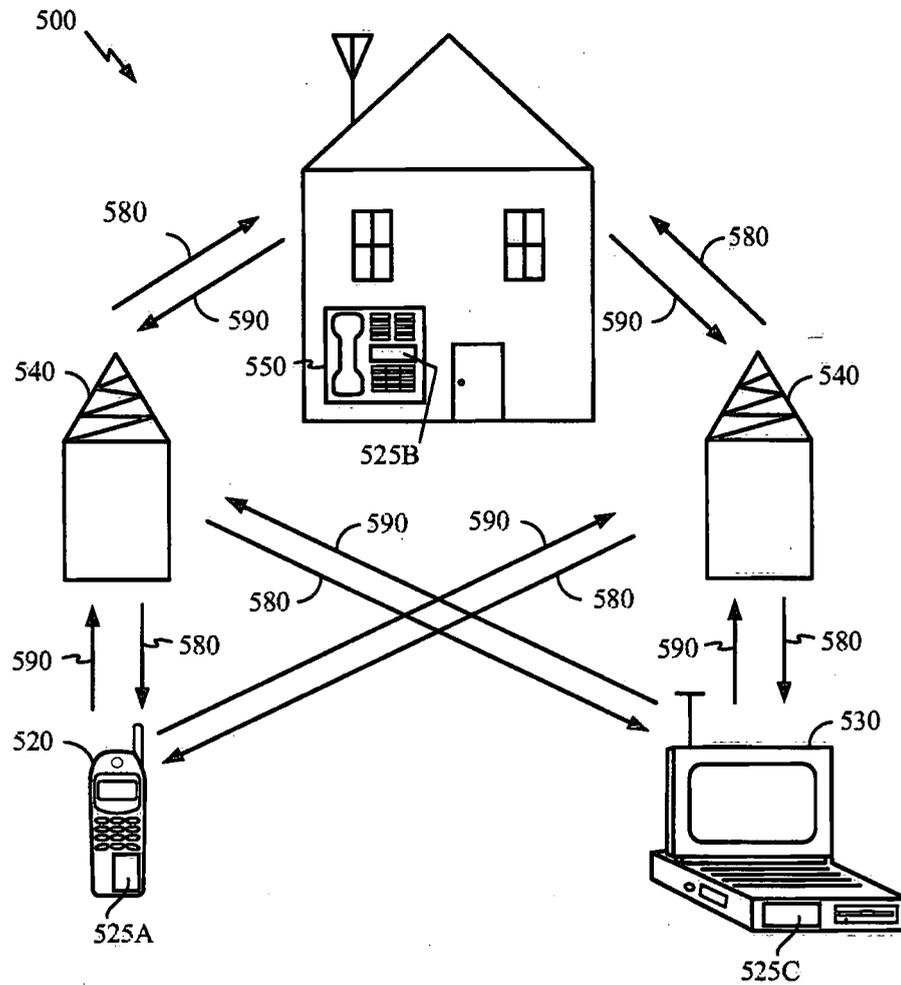


FIG. 5

**DIRECT SCATTER LOADING OF
EXECUTABLE SOFTWARE IMAGE FROM A
PRIMARY PROCESSOR TO ONE OR MORE
SECONDARY PROCESSOR IN A
MULTI-PROCESSOR SYSTEM**

**CROSS REFERENCE TO RELATED
APPLICATIONS**

This application claims the benefit of U.S. provisional patent application No. 61/316,369 filed Mar. 22, 2010, in the names of MALAMANT et al., U.S. provisional patent application No. 61/324,035 filed Apr. 14, 2010, in the names of GUPTA et al., U.S. provisional patent application No. 61/324,122 filed Apr. 14, 2010, in the names of GUPTA et al., and U.S. provisional patent application No. 61/325,519 filed Apr. 19, 2010, in the names of GUPTA et al., the disclosures of which are expressly incorporated herein by reference in their entireties.

TECHNICAL FIELD

The following description relates generally to multi-processor systems, and more specifically to multi-processor systems in which a primary processor is coupled to a non-volatile memory storing executable software image(s) of one or more other processors (referred to herein as "secondary" processors) system which are each coupled to a dedicated volatile memory, wherein the executable software images are efficiently communicated from the primary processor to the secondary processor(s) in a segmented format (e.g., using a direct scatter load process).

BACKGROUND

Processors execute software code to perform operations. Processors may require some software code, commonly referred to as boot code, to be executed for booting up. In a multi-processor system, each processor may require respective boot code for booting up. As an example, in a smartphone device that includes an application processor and a modem processor, each of the processors may have respective boot code for booting up.

A problem exists on a significant number of devices (such as smart phones) that incorporate multiple processors (e.g., a standalone application processor chip integrated with a separate modem processor chip). A flash/non-volatile memory component may be used for each of the processors, because each processor has non-volatile memory (e.g., persistent storage) of executable images and file systems. For instance, a processor's boot code may be stored to the processor's respective non-volatile memory (e.g., Flash memory, read-only memory (ROM), etc.), and upon power-up the boot code software is loaded for execution by the processor from its respective non-volatile memory. Thus, in this type of architecture the executable software, such as a processor's boot code, is not required to be loaded to the processor from another processor in the system.

Adding dedicated non-volatile memory to each processor, however, occupies more circuit board space, thereby increasing the circuit board size. Some designs may use a combined chip for Random Access Memory (RAM) and Flash memory (where RAM and Flash devices are stacked as one package to reduce size) to reduce board size. While multi-chip package solutions do reduce the needed circuit board foot print to some extent, it may increase costs.

In some multi-processor systems, software may be required to be loaded to one processor from another processor. For example, suppose a first processor in a multi-processor system is responsible for storing to its non-volatile memory boot code for one or more other processors in the system; wherein upon power-up the first processor is tasked with loading the respective boot code to the other processor(s), as opposed to such boot code residing in non-volatile memory of the other processor(s). In this type of system, the software (e.g., boot image) is downloaded from the first processor to the other processor(s) (e.g., to volatile memory of the other processor(s)), and thereafter the receiving processor(s) boots with the downloaded image.

Often, the software image to be loaded is a binary multi-segmented image. For instance, the software image may include a header followed by multiple segments of code. When software images are loaded, from an external device (e.g., from another processor) onto a target device (e.g., a target processor) there may be an intermediate step where the binary multi-segmented image is transferred into the system memory and then later transferred into target locations by the boot loader.

In a system in which the software image is loaded onto a target "secondary" processor from a first "primary" processor, one way of performing such loading is to allocate a temporary buffer into which each packet is received, and each packet would have an associated packet header information along with the payload. The payload in this case would be the actual image data. From the temporary buffer, some of the processing may be done over the payload, and then the payload would get copied over to the final destination. The temporary buffer would be some place in system memory, such as in internal random-access-memory (RAM) or double data rate (DDR) memory, for example.

Thus, where an intermediate buffer is used, the data being downloaded from a primary processor to a secondary processor is copied into the intermediate buffer. In this way, the buffer is used to receive part of the image data from the primary processor, and from the buffer the image data may be scattered into the memory (e.g., volatile memory) of the secondary processor.

The primary processor and its non-volatile memory that stores the boot image for a secondary processor may be implemented on a different chip than a chip on which the secondary processor is implemented. Thus, in order to transfer the data from the primary processor's non-volatile memory to the secondary processor (e.g., to the secondary processor's volatile memory), a packet-based communication may be employed, wherein a packet header is included in each packet communicated to the secondary processor. The packets are stored in an intermediate buffer, and some processing of the received packets is then required for that data to be stored where it needs to go (e.g., within the secondary processor's volatile memory).

SUMMARY

A multi-processor system is offered. The system includes a secondary processor having a system memory and a hardware buffer for receiving at least a portion of an executable software image. The secondary processor includes a scatter loader controller for loading the executable software image directly from the hardware buffer to the system memory. The system also includes a primary processor coupled with a memory. The memory stores the executable software image for the secondary processor. The system further includes an interface communicatively coupling the primary processor

3

and the secondary processor via which the executable software image is received by the secondary processor.

A method is also offered. The method includes receiving at a secondary processor, from a primary processor via an inter-chip communication bus, an image header for an executable software image for the secondary processor that is stored in memory coupled to the primary processor. The executable software image includes the image header and at least one data segment. The method also includes processing, by the secondary processor, the image header to determine at least one location within system memory to which the secondary processor is coupled to store the at least one data segment. The method also includes receiving at the secondary processor, from the primary processor via the inter-chip communication bus, the at least one data segment. Still further, the method includes loading, by the secondary processor, the at least one data segment directly to the determined at least one location within the system memory.

An apparatus is offered. The apparatus includes means for receiving at a secondary processor, from a primary processor via an inter-chip communication bus, an image header for an executable software image for the secondary processor that is stored in memory coupled to the primary processor. The executable software image includes the image header and at least one data segment. The apparatus also includes means for processing, by the secondary processor, the image header to determine at least one location within system memory to which the secondary processor is coupled to store the at least one data segment. The apparatus further includes means for receiving at the secondary processor, from the primary processor via the inter-chip communication bus, the at least one data segment. Still further, the apparatus includes means for loading, by the secondary processor, the at least one data segment directly to the determined at least one location within the system memory.

A multi-processor system is offered. The system includes a primary processor coupled with a first non-volatile memory. The first non-volatile memory is coupled exclusively to the primary processor and stores a file system for the primary processor and executable images for the primary processor and secondary processor. The system also includes a secondary processor coupled with a second non-volatile memory. The second non-volatile memory is coupled exclusively to the secondary processor and stores configuration parameters and file system for the secondary processor. The system further includes an interface communicatively coupling the primary processor and the secondary processor via which an executable software image is received by the secondary processor.

A multi-processor system is offered. The system includes a primary processor coupled with a first non-volatile memory. The first non-volatile memory is coupled exclusively to the primary processor and stores executable images and file systems for the primary and secondary processors. The system also includes a secondary processor. The system further includes an interface communicatively coupling the primary processor and the secondary processor via which an executable software image is received by the secondary processor.

A method is offered. The method includes sending, from a memory coupled to a primary processor, an executable software image for a secondary processor. The executable software image is sent via an interface communicatively coupling the primary processor and secondary processor. The method also includes receiving, at the secondary processor, the executable software image. The method further includes executing, at the secondary processor, the executable software image.

4

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present teachings, reference is now made to the following description taken in conjunction with the accompanying drawings.

FIG. 1 is an illustration of an exemplary device within which aspects of the present disclosure may be implemented.

FIG. 2 is an illustration of an exemplary device within which aspects of the present disclosure may be implemented.

FIG. 3 is an illustration of an operational flow for an exemplary loading process for loading an executable image from a primary processor to a secondary processor according to one aspect of the present disclosure.

FIG. 4 is a flowchart illustrating a scatter loading method according to one aspect of the present disclosure.

FIG. 5 is a block diagram showing an exemplary wireless communication system in which an embodiment of the disclosure may be advantageously employed.

DETAILED DESCRIPTION

The word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any aspect described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other aspects.

Certain aspects disclosed herein concern multi-processor systems where one primary processor is connected to a non-volatile memory storing executable images of one or more other processors (referred to herein as "secondary" processors) in the system. In such a multi-processor system each of the secondary processors may be connected to a dedicated volatile memory used for storing executable images, run-time data, and optionally a file system mirror.

Executable images are often stored in a segmented format where each segment can be loaded into a different memory region. Target memory locations of executable segments may or may not be contiguous with respect to each other. One example of a multi-segmented image format is Executable and Linking Format (ELF) which allows an executable image to be broken into multiple segments and each one of these segments may be loaded into different system memory locations.

In one exemplary aspect a direct scatter load technique is disclosed for loading a segmented image from a primary processor's non-volatile memory to a secondary processor's volatile memory. As discussed further below, the direct scatter load technique avoids use of a temporary buffer. For instance, in one aspect, rather than employing a packet-based communication in which the image is communicated via packets that each include a respective header, the raw image data is loaded from the primary processor to the secondary processor. In another aspect, headers are used which include information used to determine the target location information for the data.

Exemplary Multi-Processor Architecture with Centralized Non-Volatile Memory—with Reduced Localized Non-Volatile Memory for File System

FIG. 1 illustrates a block diagram of a first multi-processor architecture 102 in which a primary processor (application processor 104) hosts a primary (large) nonvolatile memory 106 (e.g., NAND flash memory) while a second processor (e.g., modem processor 110) has a secondary (reduced or minimal) non-volatile memory 114 (e.g., NOR flash memory).

In the communication device architecture 102, the application processor 104 is coupled to a primary non-volatile memory 106 and an application processor volatile memory

5

108 (e.g., random access memory). The modem processor 110 is coupled to a secondary non-volatile memory 114 and a modem processor volatile memory 112. An inter-processor communication bus 134 allows communications between the application processor 104 and the modem processor 110.

A modem executable image 120 for the modem processor 110 may be stored in the application processor (AP) non-volatile memory 106 together with the AP executable image 118 and the AP file system 116. The application processor 104 may load its AP executable image 118 into the application processor volatile memory 108 and store it as AP executable image 122. The application processor volatile memory 108 may also serve to store AP run-time data 124.

The modem processor 110 has the dedicated secondary reduced or minimal non-volatile memory 114 (e.g., NOR flash) for its file system 128 storage. This secondary (reduced or minimal) non-volatile memory 114 is smaller and lower cost than a flash device capable of storing both the run-time modem executable images 120 and the file system 128.

Upon system power-up, the modem processor 110 executes its primary boot loader (PBL) from the hardware boot ROM 126 (small read-only on-chip memory). The modem PBL may be adapted to download the modem executables 120 from the application processor 104. That is, the modem executable image 120 (initially stored in the primary non-volatile memory 106) is requested by the modem processor 110 from the application processor 104. The application processor 104 retrieves the modem executable image 120 and provides it to the modem processor 110 via an inter-processor communication bus 134 (e.g., inter-chip communication bus). The modem processor 110 stores the modem executable image 132 directly into the modem processor RAM (Random Access Memory) 112 to the final destination without copying the data into a temporary buffer in the modem processor RAM 112. The inter-processor communication bus 134 may be, for example, a HSIC bus (USB-based High Speed Inter-Chip), an HSI bus (MIPI High Speed Synchronous Interface), a SDIO bus (Secure Digital I/O interface), a UART bus (Universal Asynchronous Receiver/Transmitter), an SPI bus (Serial Peripheral Interface), an I2C bus (Inter-Integrated Circuit), or any other hardware interface suitable for inter-chip communication available on both the modem processor 110 and the application processor 104.

Once the modem executable image 120 is downloaded into the modem processor RAM 112 and authenticated, it is maintained as a modem executable image 132. Additionally, the modem processor volatile memory 112 may also store modem run-time data 130. The modem Boot ROM code 126 may then jump into that modem executable image 132 and start executing the main modem program from the modem processor RAM 112. Any persistent (non-volatile) data, such as radio frequency (RF) calibration and system parameters, may be stored on the modem file system 128 using the secondary (reduced or minimal) non-volatile memory 114 attached to the modem processor 110.

Exemplary Multi-Processor Architecture with Centralized Non-Volatile Memory—with No Localized Non-Volatile Memory for File Systems

FIG. 2 illustrates a block diagram of a second multi-processor architecture 202 in which a primary processor (application processor 204) hosts a primary (large) non-volatile memory 206 (e.g., NAND flash memory). The primary non-volatile memory 206 may store a modem-executable image 214 and/or a modem file system 220 for the secondary processor (modem processor 210). The secondary processor (modem processor 210) may be configured to request the modem-executable image 214 and/or modem file system 220

6

from the primary processor 204. The primary processor 204 then retrieves the requested modem-executable image 214 and/or modem file system 220 from the non-volatile memory 206 and provides it to the secondary processor 210 via an inter-processor communication bus 234.

In this architecture 202, the application processor 204 is coupled to the non-volatile memory 206 and an application processor volatile memory 208 (e.g., random access memory). The modem processor 210 is coupled to a modem processor volatile memory 212 but does not have its own non-volatile memory. The modem processor volatile memory 212 stores a file system mirror 228, a modem executable image 236, and modem run-time data 230. The inter-processor communication bus 231 allows communications between the application processor 204 and modem processor 210.

All the executable images 214 and file system 220 for the modem processor 210 may be stored in the non-volatile memory 206 together with the AP executable image 218 and the AP file system 216. The application processor 204 may load its AP executable image 218 into the application processor volatile memory 208 and store it as AP executable image 222. The application processor volatile memory 208 may also serve to store AP run-time data 224. The modem file system may be encrypted with a modem processor's private key for privacy protection and prevention of subscriber identity cloning.

Upon system power-up, the modem Boot ROM code 226 downloads both the modem executable image 214 and the modem file system 220 from the application processor 204 into the modem processor volatile memory 212. During normal operation, any read accesses to the modem file system 228 are serviced from the modem processor volatile memory 212. Any write accesses are performed in the modem processor volatile memory 212 as well. In addition, there may be a background process running on the modem processor 210 and the application processor 204 to synchronize the contents of the File System 228 in modem processor volatile memory 212 with the modem file system 220 stored on the non-volatile memory 206.

The primary and secondary processors may periodically synchronize the file system in the volatile memory for the secondary processor with the corresponding file system in the primary non-volatile memory. The first write to the modem file system 228 may start a timer (for example, a ten minute timer) in the modem processor 210. While this timer is running, all writes to the file system 228 are coalesced into the modem processor volatile memory 212. Upon expiration of the timer, the modem processor 210 copies the file system image 228 from volatile memory 212, encrypts it, and alerts the application processor 204 that new data is available. The application processor 204 reads the encrypted copy and writes it to the non-volatile memory 206 into the modem file system 220. The application processor 204 then signals the modem processor 210 that the write operation is complete. If a synchronization operation fails, a present version of the modem file system may be used. Synchronization may occur periodically (for example, every ninety seconds) or after a certain time following a write operation by the modem to its file system. To prevent corruption from circumstances such as sudden power removal, two copies of the modem file system 220 may be stored.

The modem processor 210 may also initiate a "flush" operation of the file system mirror 228 to the application processor's non-volatile memory 206. This may occur for a number of reasons, including phone power-off, as well as sending an acknowledgement message to the network to indicate acceptance and storage of incoming SMS messages.

File system read operations on the modem processor 210 are serviced from the modem processor volatile memory 212, which reflects the current state of the modem file system. Because read operations are more frequent than write operations, and write operations tend to occur in "bursts" of activity, the overall system load and power consumption may be reduced.

The application processor 204, modem processor 210, and Hoot loader have specific measures in place to ensure that there is always at least one complete file system image available in the non-volatile memory 206 at all times. This provides immunity to power-loss or surprise-reset scenarios.

Application of the concepts disclosed herein are not limited, to the exemplary system shown above but may likewise be employed with various other multi-processor systems. Zero Copy Transport flow

Aspects of the present disclosure provide techniques for efficiently loading the executable software images from the primary processor's non-volatile memory to the secondary processor's volatile memory. As mentioned above, traditional loading processes require an intermediate step where the binary multi-segmented image is buffered (e.g., transferred into the system memory) and then later scattered into target locations (e.g., by a boot loader). Aspects of the present disclosure provide techniques that alleviate the intermediate step of buffering required in traditional loading processes. Thus, aspects of the present disclosure avoid extra memory copy operations, thereby improving performance (e.g., reducing the time required to boot secondary processors in a multi-processor system).

As discussed further below, one exemplary aspect of the present disclosure employs a direct scatter load technique for loading the executable software images from the primary processor's non-volatile memory to the secondary processor's volatile memory. Certain aspects of the present disclosure also enable concurrent image transfers with post-transfer data processing, such as authentication, which may further improve efficiency, as discussed further below.

In one aspect, the host primary processor does not process or extract any information from the actual image data it simply sends the image data as "raw" data to the target, without any packet header attached to the packet. Because the target secondary processor initiates the data transfer request, it knows exactly how much data to receive. This enables the host to send data without a packet header, and the target to directly receive and store the data. In that aspect, the target requests data from the host as needed. The first data item it requests is the image header for a given image transfer. Once the target has processed the image header, it knows the location and size of each data segment in the image. The image header also specifies the destination address of the image in target memory. With this information, the target can request data from the host for each segment, and directly transfer the data to the appropriate location in target memory. The hardware controller for the inter-chip communication bus on the application processor may add its own low-level protocol headers, which would be processed and stripped by the modem processor. These low-level headers may be transparent to the software running on both processors.

In one aspect of the present disclosure, the loading process is divided into two stages, as illustrated in the exemplary flow shown in FIG. 3. FIG. 3 shows a block diagram of a primary processor 301 (which may be the application processors 104 or 204 of FIG. 1 or 2 with their non-volatile memory 106 or 206) and a secondary processor 302 (which may be the modem processor 110 or 210 of FIG. 1 or 2 with their volatile memory 112 or 212). In FIG. 3, an exemplary software image

for secondary processor 302 is stored to non-volatile memory of the primary processor 301. As shown in this example, the exemplary software image 303 is a multi-segment image that includes an image header portion and multiple data segments (shown as data segments 1-5 in this example). The primary processor 301 and secondary processor 302 may be located on different physical silicon chips (i.e. on a different chip package) or may be located on the same package.

In the first stage of the exemplary loading process of FIG. 3, the image header information is transferred to the secondary processor 302. The primary processor 301 retrieves the data image segments, beginning with the image header, from non-volatile memory of the primary processor 306. The primary processor 301 parses the image header to load individual image segments from non-volatile memory of the primary processor 306 to system memory of the primary processor 307. The image header includes information used to identify where the modem image executable data is to be eventually placed into the system memory of the secondary processor 305. The header information is used by the secondary processor 302 to program the scatter loader/direct memory access controller 304 receive address when receiving the actual executable data. Data segments are then sent from system memory 307 to the primary hardware transport mechanism 308. The segments are then sent from the hardware transport mechanism 308 of the primary processor 301 to a hardware transport mechanism 309 of the secondary processor 302 over an inter-chip communication bus 310 (e.g., a HS-USB cable.) The first segment transferred may be the image header, which contains information used by the secondary processor to locate the data segments into target locations in the system memory of the secondary processor 305. The image header may include information used to determine the target location information for the data.

In one aspect, the target locations are not predetermined, but rather are determined by software executing in the secondary processor as part of the scatter loading process. Information from the image header may be used to determine the target locations. In this aspect the secondary processor's boot loader first requests the image header from the primary processor (the primary processor CPU does not process the image header at all). The secondary processor knows how the data segments are laid out in the non-volatile memory by looking at the image header (besides the RAM address/size, the header also includes the relative locations in non-volatile memory with respect to the start of the image file for each segment). Subsequent requests for the data segments are driven by the secondary processor.

In another aspect the primary processor may indicate where to put the segments in the secondary processor's volatile memory by parsing the image header and then programming the secondary processor's controller to place the following data segments in the specified address dictated in the image header. This may involve extra hardware to allow this external control of the secondary processor's controller.

The image header generally includes a list of segment start addresses and sizes defining where each of the segments should be loaded in the secondary processor's system memory 305. Secondary processor 302 includes a hardware transport mechanism 309 (e.g., a USB controller) that includes a scatter loader controller 304. In the second stage of the loading process, the boot loader programs the inter-chip connection controller's engine to receive incoming data and scatter load it into the secondary processor's corresponding target memory regions 305 according to the header information received in the first stage.

In case of USB or HSIC bus, each segment of the image may be transferred as a single USB transfer on the inter-chip communication bus 310. Knowing the size of the segment and the destination address allows the software to program the scatter loader controller 304 of the secondary processor 302 for the transfer of the entire segment directly into the target memory location (within system memory 305) with minimum software intervention by the secondary processor 302. This may result in an increased performance on the USB/HSIC bus when the segments are significantly large (e.g., over 1 megabyte (MB)).

As shown in FIG. 3, the image segments are not necessarily placed into consecutive locations within the secondary processor's system memory 305. Instead, the segments may be spread out in different locations of the memory. The exemplary loading process of FIG. 3 enables a copy of the secondary processor's software (i.e., the image 303) to be sent from the primary processor 301 directly to the final destination of segment on the secondary processor's system memory 305.

The image header is loaded from the primary processor 301 to scatter loader controller 304 of secondary processor 302. That image header provides information as to where the data segments are to be located in the system memory 305. The scatter loader controller 304 accordingly transfers the image segments directly into their respective target locations in the secondary processor's system memory 305. That is, once the secondary processor's CPU processes the image header in its memory 305 and programs the scatter loader controller 304, the scatter loader controller 304 knows exactly where the image segments need to go within the secondary processor's system memory 305, and thus the hardware scatter loader controller 304 is then programmed accordingly to transfer the data segments directly into their target destinations. In the example of FIG. 3, the scatter loader controller 304 receives the image segments and scatters them to different locations in the system memory 305. In one aspect, the executable software image is loaded into the system memory of the secondary processor without an entire executable software image being stored in the hardware buffer of the secondary processor.

Accordingly, no extra memory copy operations occur in the secondary processor in the above aspect. Thus, conventional techniques employing a temporary buffer for the entire image, and the packet header handling, etc., are bypassed in favor of a more efficient direct loading process. Thus, the exemplary load process of FIG. 3 does not require the intermediate buffer operations traditionally required for loading a software image from a primary processor to a secondary processor. Instead of scatter loading from a temporary buffer holding the entire image, the exemplary load process of FIG. 3 allows for direct scatter load the image segments to their respective target destinations directly from the hardware to the system memory. Once the image header is processed, the executable image is directly scatter loaded into target memory, bypassing farther CPU involvement.

Conventionally, when an external interface is involved (e.g., as is used in communicating image data from a primary processor to a secondary processor), some mechanism is required to transport that data so that both processors know what the actual data is and how to read the data. Often, the data to be transferred over an external interface is packetized with each packet including a header describing the data contained within the packet. For instance, in a transmission control protocol/internet protocol (TCP/IP) system where data is being transferred over a network, overhead associated with processing of packet headers arises.

In accordance with certain aspects of the present invention (e.g., as in the example of FIG. 3), the raw image data is transported. For instance, rather than transporting each segment of image data with a packet header, the exemplary load process of FIG. 3 determines the needed information about the data from the header associated with the entire image. Thus, the image header may be initially transferred, and all the processing for determining how to store the data to system memory 305 can occur before the transfer of the segments (based on the image header), and then the segments are transferred as raw data, rather than requiring processing of a packet-header for each segment as the segments are transferred. Thus, in the example of FIG. 3, the raw image data is being communicated from the primary processor to the secondary processor, and then handled by the hardware, which may strip off any USB packet headers, etc. In this exemplary aspect, there is no CPU processing done on the actual data segments, thereby improving efficiency of the load process.

When multiple images have to be loaded into the volatile memory of the same secondary processor, the above sequence of FIG. 3 may be repeated as many times as the number of images being transferred, in accordance with one aspect of the present disclosure. In certain aspects, within the primary processor 301, transfer from non-volatile memory to system memory may happen in parallel with sending data from the primary to secondary processor.

In one aspect, upon completion of each segment's transfer, the secondary processor 302 programs the scatter loader controller 304 to transfer the next segment and starts authentication of the segment that was just transferred. This enables the scatter loader controller 304 to transfer data while the secondary processor 302 performs the authentication. Authentication here refers generally to checking the integrity and authenticity of the received data. The details of the authentication mechanism are outside the scope of this disclosure, and any suitable authentication mechanism (including those well-known in the art) may be employed as may be desired in a given implementation. The above-mentioned parallelism can also apply to other post-transfer processing that may be desired to be performed by the secondary processor 302 in a given implementation.

As soon as the last segment of the last image is transferred and authenticated, the secondary processor 302 may continue with the boot process and execute transferred images.

In one aspect, the modem (secondary) processor 110 executes a boot loader from an embedded boot read-only memory (ROM). In such an aspect, executing the boot ROM from the hardware eliminates the need for flash memory or devices on the modem side. The ROM code may be executed by the silicon itself.

FIG. 4 is a flowchart illustrating a scatter loading method according to one aspect of the present disclosure. As shown in block 402, a secondary processor receives, from a primary processor via an inter-chip communication bus, an image header for an executable software image for the secondary processor that is stored in memory coupled to the primary processor, the executable software image comprising the image header and at least one data segment. As shown in block 404, the secondary processor processes the image header to determine at least one location within system memory to which the secondary processor is coupled to store the at least one data segment. As shown in block 406, the secondary processor receives, from the primary processor via the inter-chip communication bus, the at least one data segment. As shown in block 408, the secondary processor loads the at least one data segment directly to the determined at least one location within the system memory.

11

In one aspect an apparatus includes means for receiving an executable image, means for processing an image header, means for receiving a data segment, and means for loading a data segment. These means may include a primary processor 301, secondary processor 302, inter-bus communication bus 310, memory 305 or 307, non-volatile memory 306, controller 304, or hardware transport mechanisms 308 or 309. In another aspect, the aforementioned means may be a module or any apparatus configured to perform the functions recited by the aforementioned means.

In view of the above, a secondary processor's software image may be loaded from a primary processor via interconnection bonds, like HS-USB or high speed interconnect, instead of loading the software image directly from non-volatile memory connected to the secondary processor. The secondary processor may not be directly connected to non-volatile memory. Thus, aspects of the present disclosure may reduce the time it takes to boot secondary processors in a multi-processor system where secondary processor images are transferred from the primary processor. This reduction is achieved by avoiding extra memory copy operations and enabling concurrent image transfers with background data processing, such as authentication.

FIG. 5 is a block diagram showing an exemplary wireless communication system 500 in which an embodiment of the disclosure may be advantageously employed. For purposes of illustration, FIG. 5 shows three remote units 520, 530, and 550 and two base stations 540. It will be recognized that wireless communication systems may have many more remote units and base stations. Remote units 520, 530, and 550 include IC devices 525A, 525C and 525B, that include the disclosed MRAM. It will be recognized that other devices may also include the disclosed MRAM, such as the base stations, switching devices, and network equipment. FIG. 5 shows forward link signals 580 from the base station 540 to the remote units 520, 530, and 550 and reverse link signals 590 from the remote units 520, 530, and 550 to base stations 540.

In FIG. 5, remote unit 520 is shown as a mobile telephone, remote unit 530 is shown as a portable computer, and remote unit 550 is shown as a fixed location remote unit in a wireless local loop system. For example, the remote units may be mobile phones, hand-held personal communication systems (PCS) units, portable data units such as personal data assistants, GPS enabled devices, navigation devices, set top boxes, music players, video players, entertainment units, fixed location data units such as meter reading equipment, or any other device that stores or retrieves data or computer instructions, or any combination thereof. Although FIG. 5 illustrates remote units according to the teachings of the disclosure, the disclosure is not limited to these exemplary illustrated units. Embodiments of the disclosure may be suitably employed in any device which includes MRAM.

For a firmware and/or software implementation, the methodologies may be implemented with modules (e.g., procedures, functions, and so on) that perform the functions described herein. Any machine-readable medium tangibly embodying instructions may be used in implementing the methodologies described herein. For example, software codes may be stored in a memory and executed by a processor unit. Memory may be implemented within the processor unit or external to the processor unit. As used herein the term "memory" refers to any type of long term, short term, volatile, nonvolatile, or other memory and is not to be limited to any particular type of memory or number of memories, or type of media upon which memory is stored.

12

If implemented in firmware and/or software, the functions may be stored as one or more instructions or code on a computer-readable medium. Examples include computer-readable media encoded with a data structure and computer-readable media encoded with a computer program. Computer-readable media includes physical computer storage media. A storage medium may be any available medium that can be accessed by a computer. By way of example, and not limitation, such computer-readable media can include RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to store desired program code in the form of instructions or data structures and that can be accessed by a computer; disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and blu-ray disc where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

In addition to storage on computer readable medium, instructions and/or data may be provided as signals on transmission media included in a communication apparatus. For example, a communication apparatus may include a transceiver having signals indicative of instructions and data. The instructions and data are configured to cause one or more processors to implement the functions outlined in the claims.

Although specific circuitry has been set forth, it will be appreciated by those skilled in the art that not all of the disclosed circuitry is required to practice the disclosure. Moreover, certain well known circuits have not been described, to maintain focus on the disclosure.

Although the present disclosure and its advantages have been described in detail, it should be understood that various changes, substitutions and alterations can be made herein without departing from the technology of the disclosure as defined by the appended claims. For example, relational terms, such as "above" and "below" are used with respect to a substrate or electronic device. Of course, if the substrate or electronic device is inverted, above becomes below, and vice versa. Additionally, if oriented sideways, above and below may refer to sides of a substrate or electronic device. Moreover, the scope of the present application is not intended to be limited to the particular embodiments of the process, machine, manufacture, composition of matter, means, methods and steps described in the specification. As one of ordinary skill in the art will readily appreciate from the disclosure, processes, machines, manufacture, compositions of matter, means, methods, or steps, presently existing or later to be developed that perform substantially the same function or achieve substantially the same result as the corresponding embodiments described herein may be utilized, according to the present disclosure. Accordingly, the appended claims are intended to include within their scope such processes, machines, manufacture, compositions of matter, means, methods, or steps.

What is claimed is:

1. A multi-processor system comprising:
 - a secondary processor comprising:
 - system memory and a hardware buffer for receiving an image header and at least one data segment of an executable software image, the image header and each data segment being received separately, and
 - a scatter loader controller configured:
 - to load the image header; and

13

to scatter load each received data segment based at least in part on the loaded image header, directly from the hardware buffer to the system memory; a primary processor coupled with a memory, the memory storing the executable software image for the secondary processor; and an interface communicatively coupling the primary processor and the secondary processor, the executable software image being received by the secondary processor via the interface.

2. The multi-processor system of claim 1 in which the scatter loader controller is configured to load the executable software image directly from the hardware buffer to the system memory of the secondary processor without copying data between system memory locations on the secondary processor.

3. The multi-processor system of claim 1 in which raw image data of the executable software image is received by the secondary processor via the interface.

4. The multi-processor system of claim 1 in which the secondary processor is configured to process the image header to determine at least one location within the system memory to store the at least one data segment.

5. The multi-processor system of claim 4 in which the secondary processor is configured to determine, based on the received image header, the at least one location within the system memory to store the at least one data segment before receiving the at least one data segment.

6. The multi-processor system of claim 1, in which the secondary processor further comprises a non-volatile memory storing a boot loader that initiates transfer of the executable software image for the secondary processor.

7. The multi-processor system of claim 1 in which the primary and secondary processors are located on different chips.

8. The multi-processor system of claim 1 in which the portion of the executable software image is loaded into the system memory of the secondary processor without an entire executable software image being stored in the hardware buffer.

9. The multi-processor system of claim 1 integrated into at least one of a mobile phone, a set top box, a music player, a video player, an entertainment unit, a navigation device, a computer, a hand-held personal communication systems (PCS) unit, a portable data unit, and a fixed location data unit.

10. A method comprising:
 receiving at a secondary processor, from a primary processor via an inter-chip communication bus, an image header for an executable software image for the secondary processor that is stored in memory coupled to the primary processor, the executable software image comprising the image header and at least one data segment, the image header and each data segment being received separately;
 processing, by the secondary processor, the image header to determine at least one location within system memory to which the secondary processor is coupled to store each data segment;
 receiving at the secondary processor, from the primary processor via the inter-chip communication bus, each data segment; and
 scatter loading, by the secondary processor, each data segment ready to the determined at least one location within the system memory; and each data segment being scatter loaded based at least in part on the processed image header.

14

11. The method of claim 10 further comprising booting the secondary processor using the executable software image.

12. The method of claim 10 further comprising loading the executable software image directly from a hardware buffer to the system memory of the secondary processor without copying data between system memory locations.

13. The method of claim 10 in which the processing occurs prior to the loading.

14. The method of claim 10 in which the primary and secondary processors are located on different chips.

15. The method of claim 10 further comprising performing the receiving, processing, and loading, in at least one of a mobile phone, a set top box, a music player, a video player, an entertainment unit, a navigation device, a computer, a hand-held personal communication systems (PCS) unit, a portable data unit, and a fixed location data unit.

16. An apparatus comprising:
 means for receiving at a secondary processor, from a primary processor via an inter-chip communication bus, an image header for an executable software image for the secondary processor that is stored in memory coupled to the primary processor, the executable software image comprising the image header and at least one data segment, the image header and each data segment being received separately;
 means for processing, by the secondary processor, the image header to determine at least one location within system memory to which the secondary processor is coupled to store each data segment;
 means for receiving at the secondary processor, from the primary processor via the inter-chip communication bus, each data segment; and
 means for scatter loading, by the secondary processor, each data segment directly to the determined at least one location within the system memory, and each data segment being scatter loaded based at least in part on the processed image header.

17. The apparatus of claim 16 integrated into at least one of a mobile phone, a set top box, a music player, a video player, an entertainment unit, a navigation device, a computer, a hand-held personal communication systems (PCS) unit, a portable data unit, and a fixed location data unit.

18. A multi-processor system comprising:
 a primary processor coupled with a first non-volatile memory, the first non-volatile memory coupled to the primary processor and storing a file system for the primary processor and executable images for the primary processor and secondary processor;
 a secondary processor coupled with a second non-volatile memory, the second non-volatile memory coupled to the secondary processor and storing configuration parameters and file system for the secondary processor; and
 an interface communicatively coupling the primary processor and the secondary processor, an executable software image being received by the secondary processor via the interface, the executable software image comprising an image header and at least one data segment, the image header and each data segment being received separately, and the image header being used to scatter load each received data segment directly to a system memory of the secondary processor.

19. The multi-processor system of claim 18 integrated into at least one of a mobile phone, a set top box, a music player, a video player, an entertainment unit, a navigation device, a computer, a hand-held personal communication systems (PCS) unit, a portable data unit, and a fixed location data unit.

15

20. A multi-processor system comprising:
 a primary processor coupled with a first non-volatile
 memory, the first non-volatile memory coupled to the
 primary processor and storing executable images and
 file systems for the primary and secondary processors;
 a secondary processor not directly coupled to the first non-
 volatile memory; and
 an interface communicatively coupling the primary pro-
 cessor and the secondary processor, an executable
 software image being is received by the secondary
 processor via the interface, the executable software
 image comprising an image header and at least one
 data segment, the image header and each data seg-
 ment being received separately, and the image header
 being used to scatter load each received data segment
 directly to a system memory of the secondary proces-
 sor.

21. The multi-processor system of claim 20 integrated into
 at least one of a mobile phone, a set top box, a music player,
 a video player, an entertainment unit, a navigation device, a
 computer, a hand-held personal communication systems
 (PCS) unit, a portable data unit, and a fixed location data unit.

16

22. A method comprising:
 sending, from a memory coupled to a primary processor, an
 executable software image for a secondary processor,
 via an interface communicatively coupling the primary
 processor and secondary processor, the executable soft-
 ware image comprising an image header and at least one
 data segment;
 receiving, at the secondary processor, the image header and
 each data segment of the executable software image, the
 image header and each data segment being received
 separately, and the image header being used to scatter
 load each received data segment directly to a system
 memory of the secondary processor; and
 executing, at the secondary processor, the executable soft-
 ware image.

23. The method of claim 22 further comprising performing
 the sending, receiving, and executing, in at least one of a
 mobile phone, a set top box, a music player, a video player, an
 entertainment unit, a navigation device, a computer, a hand-
 held personal communication systems (PCS) unit, a portable
 data unit, and a fixed location data unit.

* * * * *

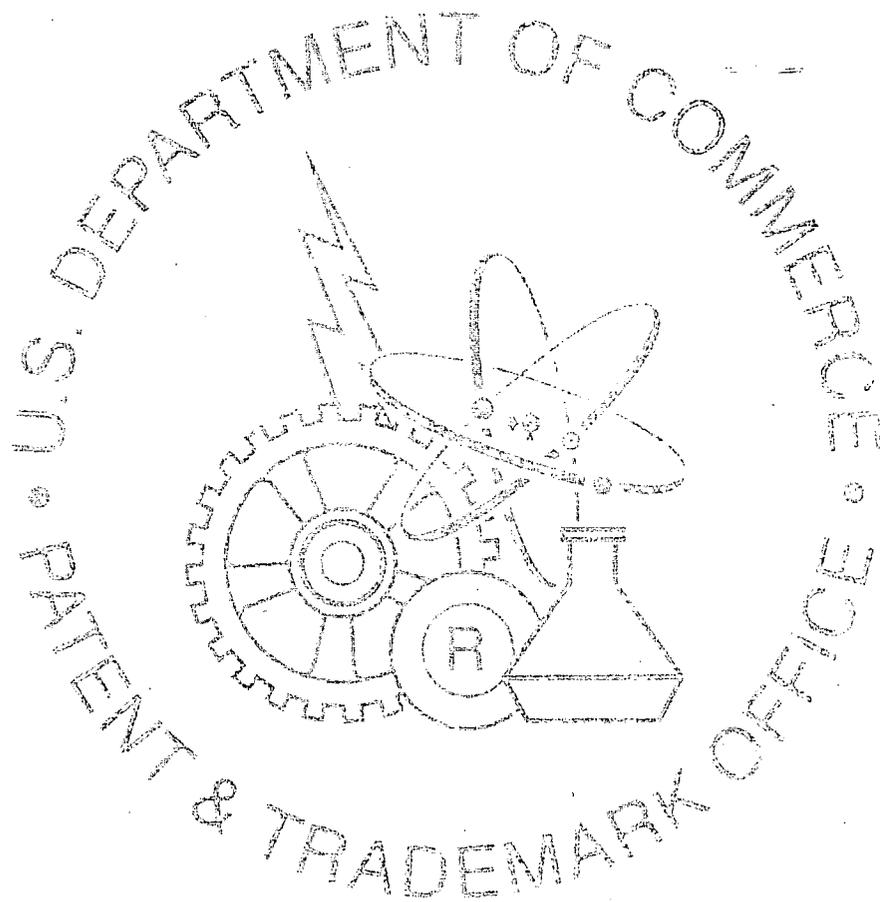


EXHIBIT E

U 7629935

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

**UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office**

April 06, 2017

**THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM
THE RECORDS OF THIS OFFICE OF:**

**U.S. PATENT: 9,535,490
ISSUE DATE: January 03, 2017**

**By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office**



**T. LAWRENCE
Certifying Officer**



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

(10) **Patent No.:** US 9,535,490 B2
(45) **Date of Patent:** Jan. 3, 2017

- (54) **POWER SAVING TECHNIQUES IN COMPUTING DEVICES**
- (71) Applicant: **QUALCOMM Incorporated**, San Diego, CA (US)
- (72) Inventors: **Vinod Harimohan Kaushik**, San Diego, CA (US); **Uppinder Singh Babbar**, San Diego, CA (US); **Andrei Danalla**, San Diego, CA (US); **Neven Klacar**, San Diego, CA (US); **Muralldhar Coimbatore Krishnamoorthy**, San Diego, CA (US); **Arunn Coimbatore Krishnamurthy**, San Diego, CA (US); **Vaibhav Kumar**, Encinitas, CA (US); **Vanitha Aravamudhan Kumar**, San Diego, CA (US); **Shailesh Maheshwari**, San Diego, CA (US); **Alok Mitra**, San Diego, CA (US); **Roshan Thomas Pius**, San Jose, CA (US); **Hariharan Sukumar**, San Diego, CA (US)
- (73) Assignee: **QUALCOMM Incorporated**, San Diego, CA (US)
- (*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 119 days.

- (52) **U.S. CL.**
CPC **G06F 1/3253** (2013.01); **G06F 1/3287** (2013.01); **G06F 13/38** (2013.01); **H04W 4/003** (2013.01);
- (58) **Field of Classification Search**
CPC **G06F 1/3202**; **G06F 1/3231**; **G06F 1/26**; **G06F 1/206**; **G06F 1/3228**; **G06F 1/08**; **G06F 1/3289**; **G06F 1/266**; **H04L 12/12**; **H04L 12/10**
See application file for complete search history.

- (56) **References Cited**
U.S. PATENT DOCUMENTS
- 5,619,681 A * 4/1997 Benhamida G06F 13/385 703/23
6,021,264 A * 2/2000 Morita G06F 13/4213 710/117
- (Continued)

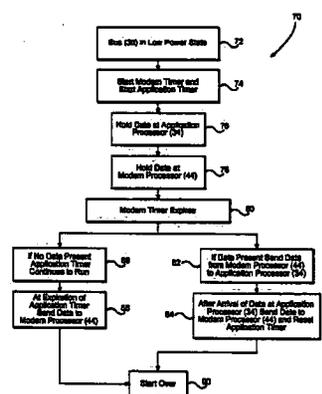
- FOREIGN PATENT DOCUMENTS**
- WO 2009039034 A1 3/2009
WO 2010125429 A1 11/2010

- OTHER PUBLICATIONS**
- Second Written Opinion for PCT/US2014/070368, mailed Nov. 9, 2015, 5 pages.
- (Continued)

Primary Examiner — Zahid Choudhury
(74) *Attorney, Agent, or Firm* — Withrow + Terranova, PLLC

- (21) Appl. No.: **14/568,694**
- (22) Filed: **Dec. 12, 2014**
- (25) **Prior Publication Data**
US 2015/0169037 A1 Jun. 18, 2015
- Related U.S. Application Data**
- (60) Provisional application No. 61/916,498, filed on Dec. 16, 2013, provisional application No. 62/019,073, filed on Jun. 30, 2014.
- (51) **Int. Cl.**
G06F 1/32 (2006.01)
H04W 52/02 (2009.01)
- (Continued)

- (57) **ABSTRACT**
- Aspects disclosed in the detailed description include power saving techniques in computing devices. In particular, as data is received by a modem processor in a computing device, the data is held until the expiration of a modem timer. The data is then passed to an application processor in the computing device over a peripheral component interconnect express (PCIe) interconnectivity bus. On receipt of
- (Continued)



the data from the modem processor, the application processor sends data held by the application processor to the modem processor over the PCIe interconnectivity bus. The application processor also has an uplink timer. If no data is received from the modem processor before expiration of the uplink timer, the application processor sends any collected data to the modem processor at expiration of the uplink timer. However, if data is received from the modem processor, the uplink timer is reset.

31 Claims, 13 Drawing Sheets

- (51) **Int. Cl.**
H04W 4/00 (2009.01)
G06F 13/38 (2006.01)
- (52) **U.S. Cl.**
 CPC *H04W 52/0251* (2013.01); *H04W 52/0274* (2013.01); *H04W 52/0287* (2013.01); *Y02B 60/1235* (2013.01); *Y02B 60/1282* (2013.01); *Y02B 60/50* (2013.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,151,355 A * 11/2000 Vallee G06F 13/385
 375/220
 6,272,452 B1 * 8/2001 Wu G06F 13/385
 703/24

6,472,770 B1 * 10/2002 Pohjola B60R 16/0315
 307/10.1
 6,765,901 B1 * 7/2004 Johnson H04L 12/2856
 370/352
 6,996,214 B1 * 2/2006 Beck H04M 3/4281
 379/215.01
 7,137,018 B2 11/2006 Gutman et al.
 RE39,427 E * 12/2006 O'Sullivan 455/126
 7,647,517 B2 1/2010 Tseng et al.
 8,615,671 B2 12/2013 Robles et al.
 2003/0114186 A1 * 6/2003 Goetz H04M 1/725
 455/552.1
 2008/0034106 A1 2/2008 Bakshi et al.
 2009/0185487 A1 7/2009 Adar et al.
 2013/0198538 A1 8/2013 Diab et al.
 2014/0207991 A1 7/2014 Kaushik et al.

OTHER PUBLICATIONS

Ajanovic, J., "PCI Express™ Architecture Advanced Protocols & Features," Dec. 31, 2003, PCI-SIG Developers Conference, retrieved Feb. 25, 2015 from https://www.pcisig.com/developers/main/training_materials/get_document?doc_id=f275ada7d622d20947d69cbb34b6e0a0622249c3, 49 pages.
 International Search Report and Written Opinion for PCT/US2014/070368, mailed Mar. 5, 2015, 14 pages.
 International Preliminary Report on Patentability for PCT/US2014/070368, mailed Apr. 1, 2016, 33 pages.

* cited by examiner

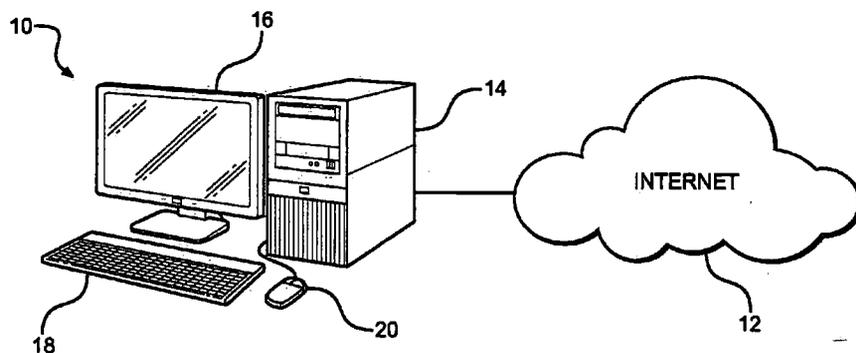


FIG. 1A

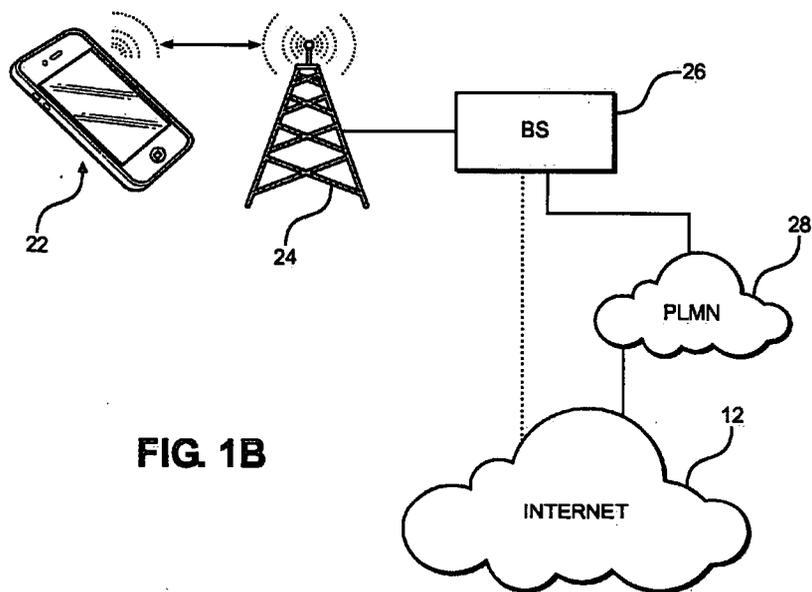


FIG. 1B

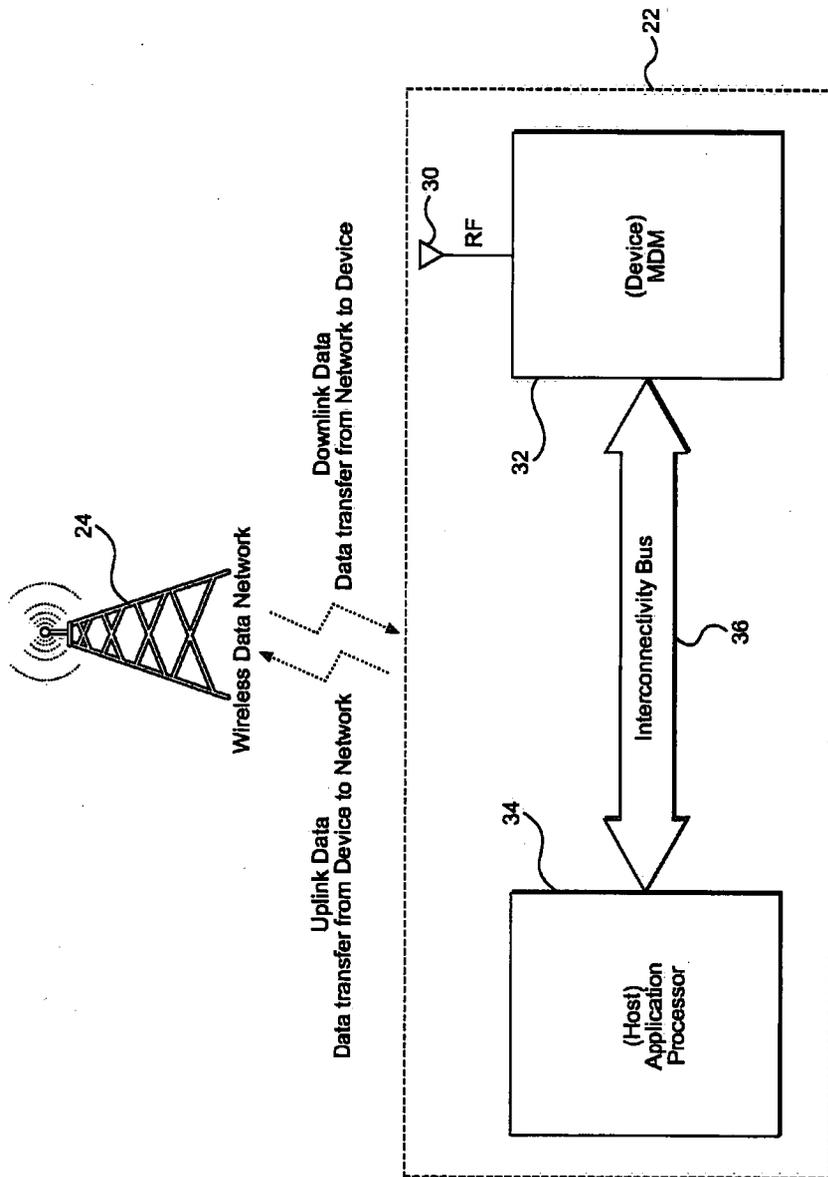


FIG. 1C

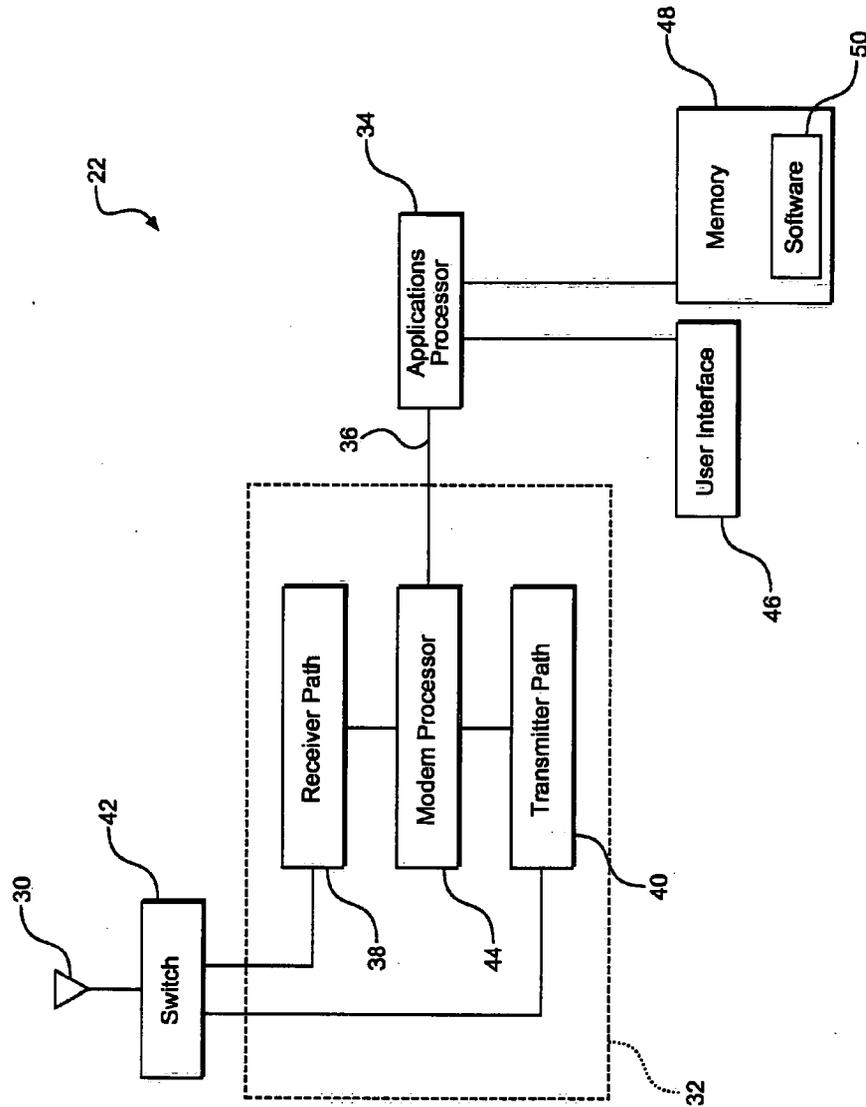


FIG. 2

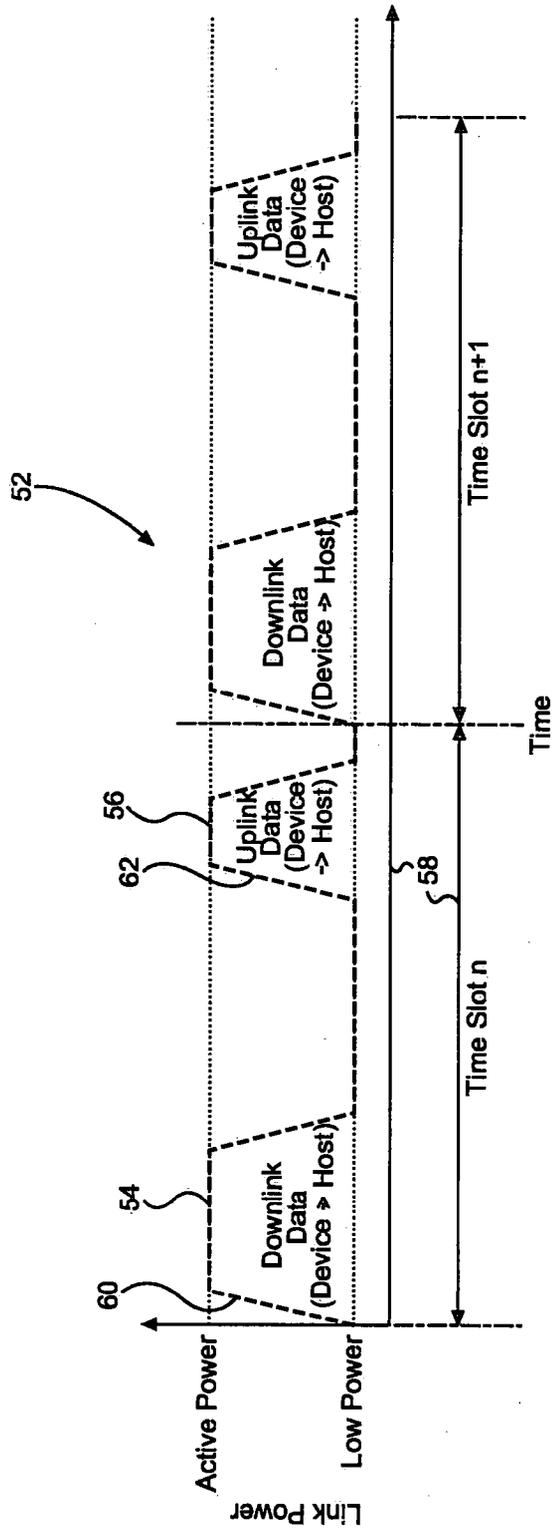


FIG. 3

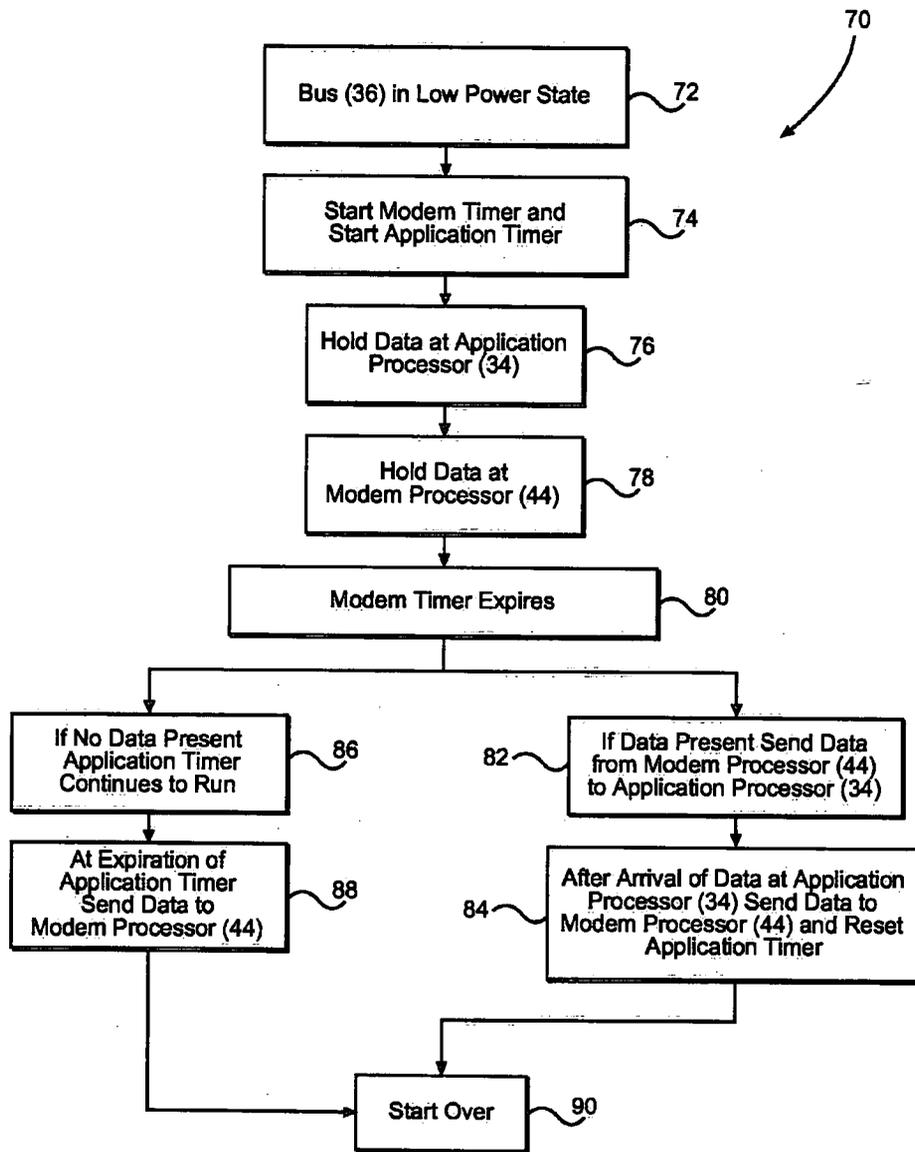


FIG. 4

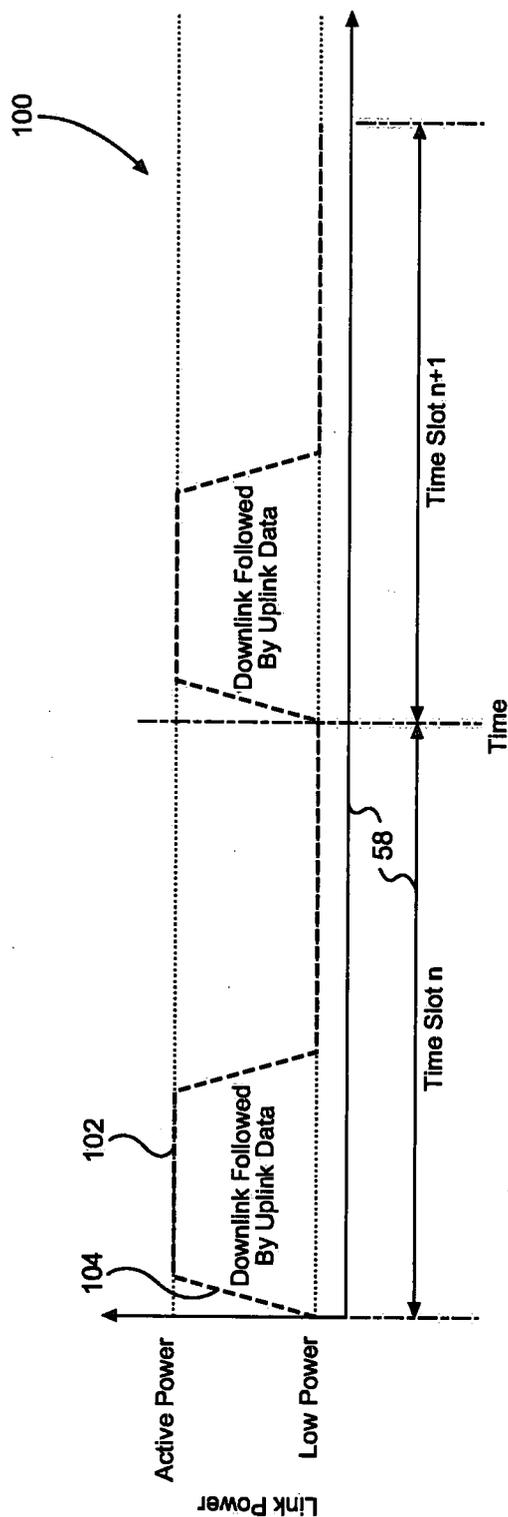


FIG. 5

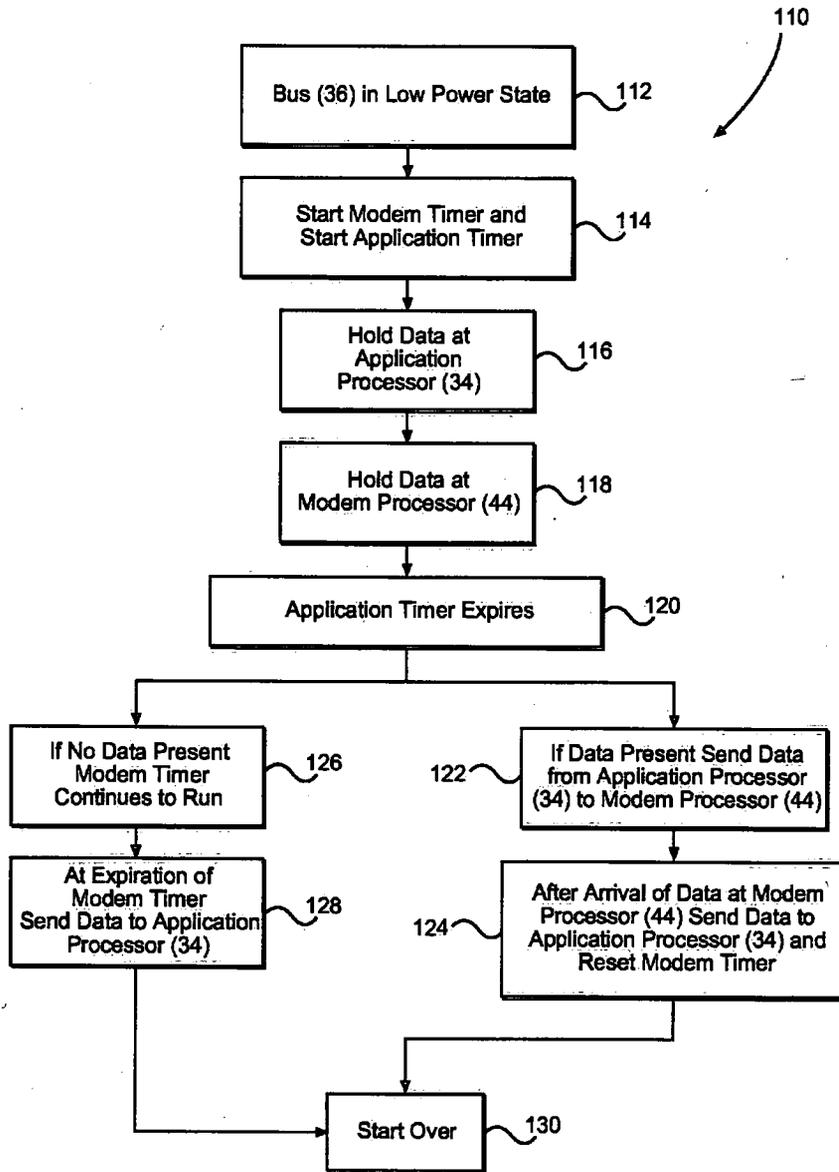


FIG. 6

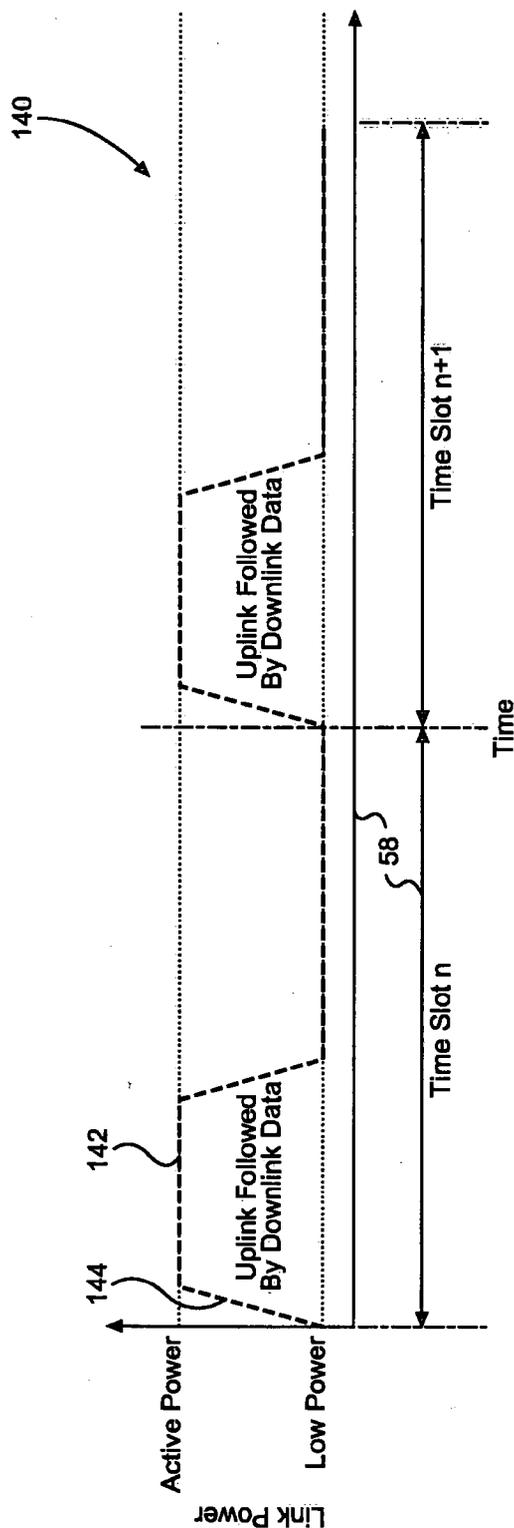


FIG. 7

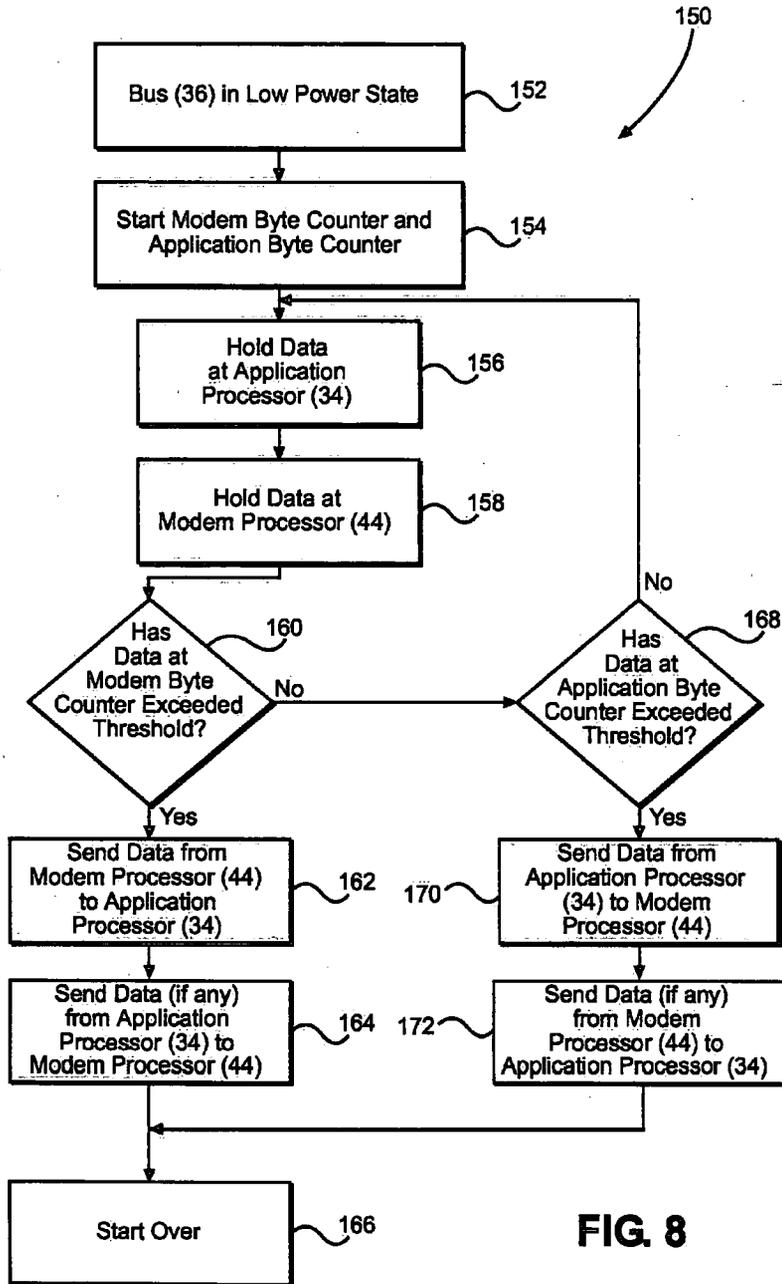


FIG. 8

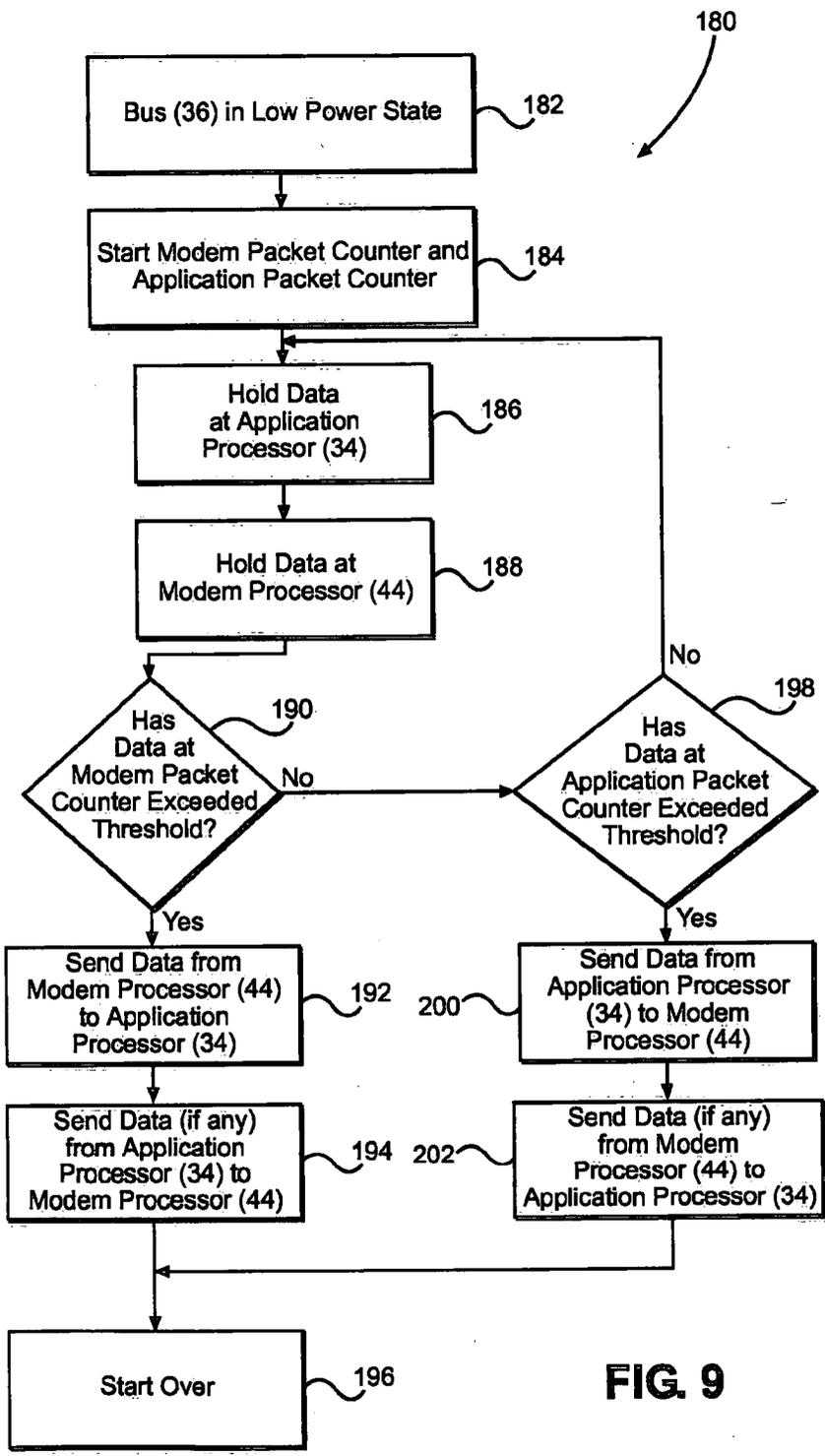


FIG. 9

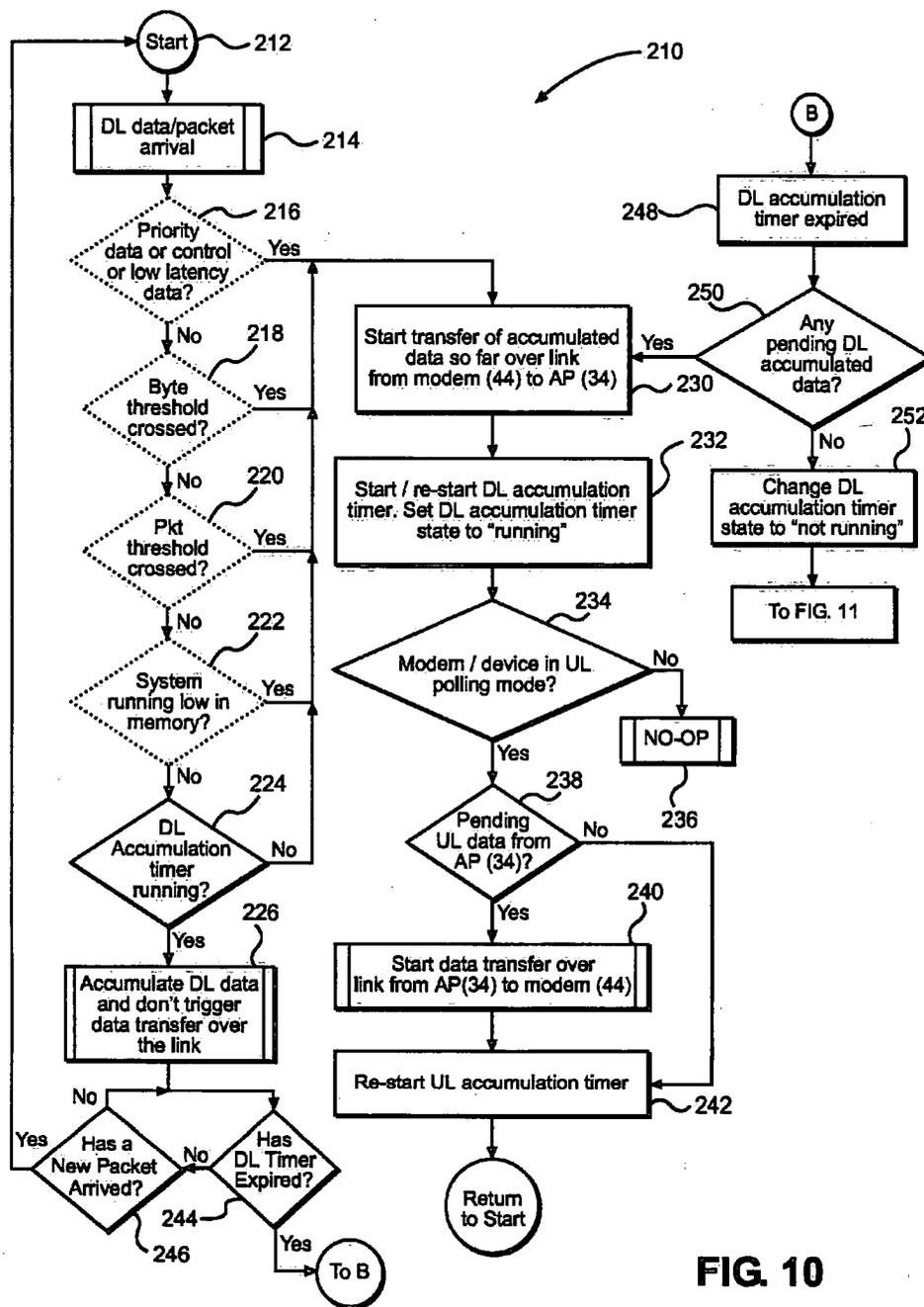


FIG. 10

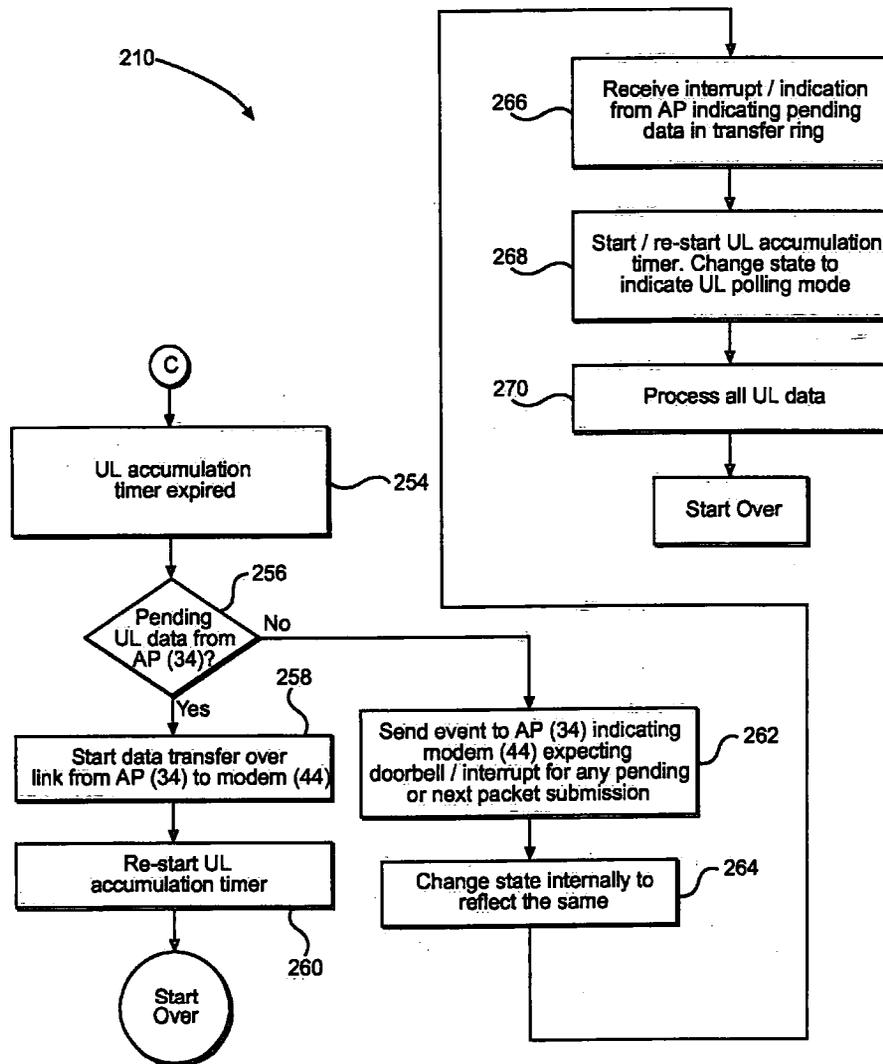


FIG. 11

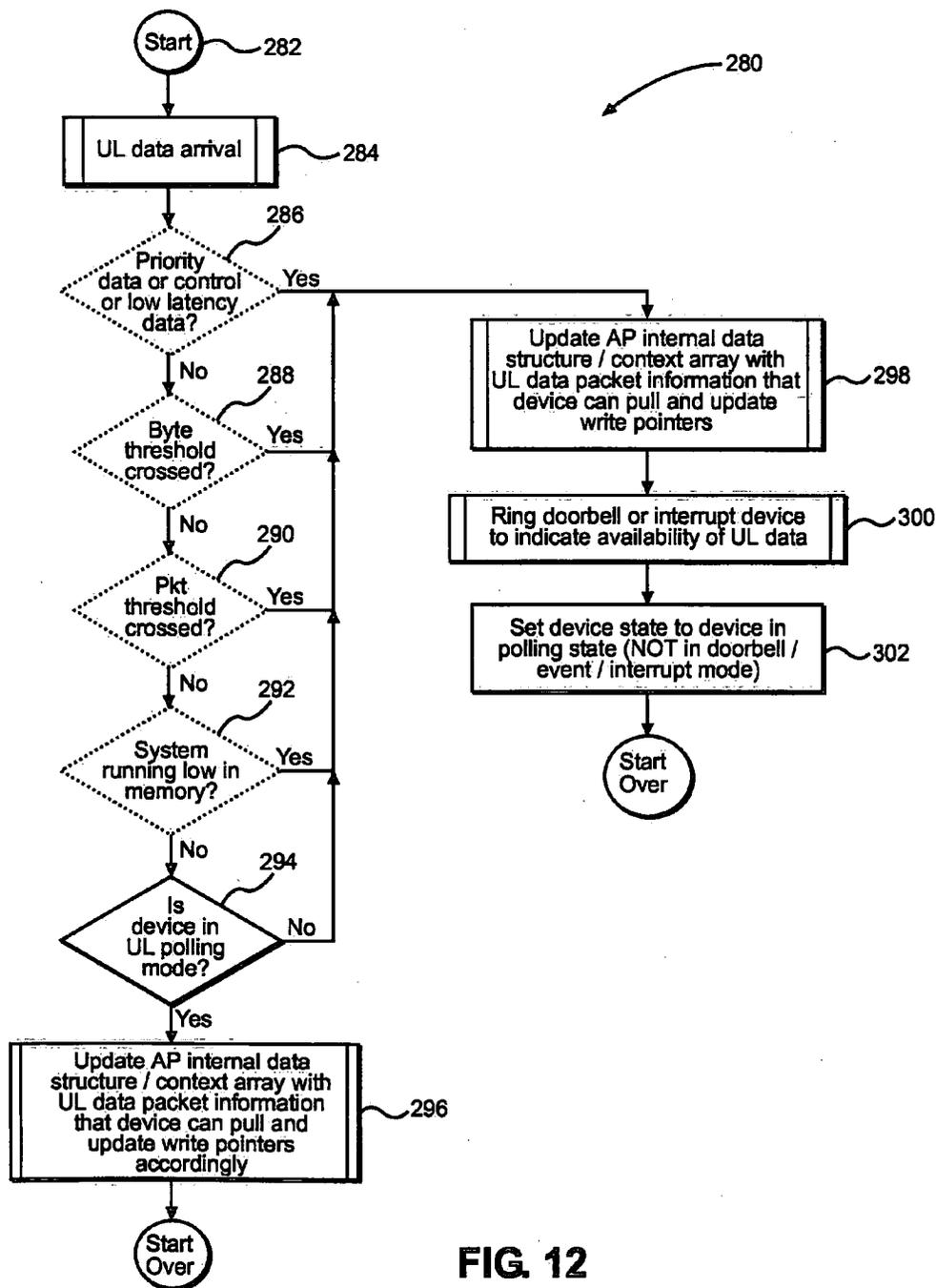


FIG. 12

POWER SAVING TECHNIQUES IN COMPUTING DEVICES

PRIORITY CLAIMS

The present application claims priority to U.S. Provisional Patent Application Ser. No. 61/916,498 filed on Dec. 16, 2013 and entitled "POWER SAVING TECHNIQUES IN COMPUTING DEVICES," which is incorporated herein by reference in its entirety.

The present application also claims priority to U.S. Provisional Patent Application Ser. No. 62/019,073 filed on Jun. 30, 2014 and entitled "POWER SAVING TECHNIQUES IN COMPUTING DEVICES," which is incorporated herein by reference in its entirety.

BACKGROUND

I. Field of the Disclosure

The technology of the disclosure relates generally to power saving techniques in computing devices.

II. Background

Computing devices are common within modern society. Ranging from small, mobile computing devices, such as a smart phone or tablet, to large server farms with numerous blades and memory banks, these devices are expected to communicate across myriad networks while providing various other base functions. While desktop devices and servers are generally immune to concerns about power consumption, mobile devices constantly struggle to find a proper balance between available functions and battery life. That is, as more functions are provided, power consumption increases, and battery life is shortened. Servers may likewise have power consumption concerns when assembled in large server farms.

Concurrent with power consumption concerns, improvements in network communications have increased data rates. For example, copper wires have been replaced with higher bandwidth fiber optic cables, and cellular networks have evolved from early Advanced Mobile Phone System (AMPS) and Global System for Mobile Communications (GSM) protocols to 4G and Long Term Evolution (LTE) protocols capable of supporting much higher data rates. As the data rates have increased, the need to be able to process these increased data rates within computing devices has also increased. Thus, earlier mobile computing devices may have had internal buses formed according to a High Speed Inter-Chip (HSIC) standard, universal serial bus (USB) standard (and particularly USB 2.0), or universal asynchronous receiver/transmitter (UART) standard. However, these buses do not support current data rates.

In response to the need for faster internal buses, the peripheral component interconnect express (PCIe) standard, as well as, later generations of USB (e.g., USB 3.0 and subsequent versions) have been adopted for some mobile computing devices. However, while PCIe and USB 3.0 can handle the high data rates currently being used, usage of such buses results in excessive power consumption and negatively impacts battery life by shortening the time between recharging events.

SUMMARY OF THE DISCLOSURE

Aspects disclosed in the detailed description include power saving techniques in computing devices. In particular, as data is received by a modem processor in a computing device, the data is held until the expiration of a modem

timer. The data is then passed to an application processor in the computing device over a peripheral component interconnect express (PCIe) interconnectivity bus. On receipt of the data from the modem processor, the application processor sends data held by the application processor to the modem processor over the PCIe interconnectivity bus. The application processor also has an uplink timer. If no data is received from the modem processor before expiration of the uplink timer, the application processor sends any collected data to the modem processor at expiration of the uplink timer. However, if data is received from the modem processor, the uplink timer is reset. By holding or accumulating the data at a source processor in this fashion, unnecessary transitions between low power states and active states on the PCIe bus are reduced and power is conserved.

In an alternate aspect, instead of initiating data transfer based on the expiration of the downlink timer (with or without expiration of the uplink timer), accumulated data transfer may be initiated based on expiration of just an uplink accumulation timer. The uplink accumulation timer may be within a host or a device associated with the interconnectivity bus.

In another alternate aspect, initiation of the data transfer may be based on reaching a predefined threshold for a byte accumulation limit counter. The byte accumulation limit counter is not mutually exclusive relative to the other counters and may operate as an override mechanism for one of the other accumulation timers. Use of such an override may be useful in situations where a sudden burst of data arrives that would exceed buffer space and/or bus bandwidth. Likewise, instead of a byte counter, a packet size counter or a "total number of packets" counter may be used to cover situations where numerous packets or a particularly large packet is delivered by the network.

In further aspects of the present disclosure, the timers may be overridden by other factors or parameters. Such an override is alluded to above with the byte accumulation limit counters and the total number of packets counter, which causes data transfers independently of the timers. Other parameters may also override the timers, such as the presence of low latency traffic (e.g., control messages), synchronizing the uplink and downlink data transfers, or low latency quality of service requirements. When such traffic is present, an interrupt or other command may be used to initiate data transfers before expiration of a timer. Still other factors may override the timers, such as an indication that a device or host is not in an automatic polling mode.

In this regard in one aspect, a mobile terminal is disclosed. The mobile terminal comprises a modem timer. The mobile terminal also comprises a modem processor. The modem processor is configured to hold modem processor to application processor data until expiration of the modem timer. The mobile terminal also comprises an application processor. The mobile terminal also comprises an interconnectivity bus communicatively coupling the application processor to the modem processor. The application processor is configured to hold application processor to modem processor data until receipt of the modem processor to application processor data from the modem processor through the interconnectivity bus after which the application processor to modem processor data is sent to the modem processor through the interconnectivity bus.

In another aspect, a method of controlling power consumption in a computing device is disclosed. The method comprises holding data received by a modem processor from a remote network until expiration of a downlink timer. The method also comprises passing the data received by the

3

modem processor to an application processor over an interconnectivity bus. The method also comprises holding application data generated by an application associated with the application processor for until receipt of the data from the modem processor or expiration of an uplink timer, whichever occurs first.

In another aspect, a mobile terminal is disclosed. The mobile terminal comprises a modem processor. The mobile terminal also comprises an application timer. The mobile terminal also comprises an application processor. The application processor is configured to hold application processor to modem processor data until expiration of the application timer. The mobile terminal also comprises an interconnectivity bus communicatively coupling the application processor to the modem processor. The modem processor is configured to hold modem processor to application processor data until receipt of the application processor to modem processor data from the application processor through the interconnectivity bus after which the modem processor to application processor data is sent to the application processor through the interconnectivity bus.

In another aspect, a mobile terminal is disclosed. The mobile terminal comprises a modem byte accumulation limit counter. The mobile terminal also comprises a modem processor. The modem processor is configured to hold modem processor to application processor data until a predefined threshold of bytes has been reached by the modem byte accumulation limit counter. The mobile terminal also comprises an application processor. The mobile terminal also comprises an interconnectivity bus communicatively coupling the application processor to the modem processor. The application processor is configured to hold application processor to modem processor data until receipt of the modem processor to application processor data from the modem processor through the interconnectivity bus after which the application processor to modem processor data is sent to the modem processor through the interconnectivity bus.

With regards to another aspect, a mobile terminal is disclosed. The mobile terminal comprises a modem packet counter. The mobile terminal also comprises a modem processor. The modem processor is configured to hold modem processor to application processor data until a predefined threshold of packets has been reached by the modem packet counter. The mobile terminal also comprises an application processor. The mobile terminal also comprises an interconnectivity bus communicatively coupling the application processor to the modem processor. The application processor is configured to hold application processor to modem processor data until receipt of the modem processor to application processor data from the modem processor through the interconnectivity bus after which the application processor to modem processor data is sent to the modem processor through the interconnectivity bus.

In another aspect, a mobile terminal is disclosed. The mobile terminal comprises a modem processor. The mobile terminal also comprises an application byte counter. The mobile terminal also comprises an application processor. The application processor is configured to hold application processor to modem processor data until a predefined threshold of bytes has been reached by the application byte counter. The mobile terminal also comprises an interconnectivity bus communicatively coupling the application processor to the modem processor. The modem processor is configured to hold modem processor to application processor data until receipt of the application processor to modem processor data from the application processor through the

4

interconnectivity bus after which the modem processor to application processor data is sent to the application processor through the interconnectivity bus.

In another aspect, a mobile terminal is disclosed. The mobile terminal comprises a modem processor and an application packet counter. The mobile terminal also comprises an application processor. The application processor is configured to hold application processor to modem processor data until a predefined threshold of packets has been reached by the application packet counter. The mobile terminal comprises an interconnectivity bus communicatively coupling the application processor to the modem processor. The modem processor is configured to hold the modem processor to application processor data until receipt of the application processor to modem processor data from the application processor through the interconnectivity bus after which the modem processor to application processor data is sent to the application processor through the interconnectivity bus.

With regards to another aspect, a method is disclosed. The method comprises starting an application timer at an application processor. The method also comprises accumulating data at the application processor until expiration of the application timer. The method comprises sending the accumulated data from the application processor to a modem processor across an interconnectivity bus. The method further comprises holding modem processor data at the modem processor until receipt of the accumulated data from the application processor.

In another aspect, a mobile terminal is disclosed. The mobile terminal comprises a modem timer. The mobile terminal also comprises a modem processor. The modem processor is configured to hold modem processor to application processor data until expiration of the modem timer. The mobile terminal also comprises an application processor. The mobile terminal also comprises an interconnectivity bus communicatively coupling the application processor to the modem processor. The application processor is configured to hold application processor to modem processor data until the modem processor pulls data from the application processor after transmission of the modem processor to application processor data.

BRIEF DESCRIPTION OF THE FIGURES

FIG. 1A is a simplified view of a mobile computing device operating with remote networks;

FIG. 1B is a simplified view of a mobile terminal operating with remote networks;

FIG. 1C is an expanded block diagram view of the mobile terminal of FIG. 1B with an interconnectivity bus illustrated; FIG. 2 is a block diagram of the mobile terminal of FIG. 1B;

FIG. 3 is an exemplary time versus link power graph in a conventional computing device;

FIG. 4 is a flowchart of an exemplary process for achieving power savings in the mobile terminal of FIG. 1B;

FIG. 5 is an exemplary time versus link power graph in a mobile computing device using the process of FIG. 4;

FIG. 6 is a flowchart of another exemplary process for achieving power savings in the mobile computing device;

FIG. 7 is an exemplary time versus link power graph in the mobile computing device using the process of FIG. 6;

FIG. 8 is a flowchart of an exemplary process that uses a byte counter to control data accumulation;

FIG. 9 is a flowchart of an exemplary process that uses a packet counter to control data accumulation;

FIG. 10 is a flowchart of a consolidated accumulation process with overrides illustrated from a downlink priority perspective;

FIG. 11 is a continuation of the flowchart of FIG. 10; and
FIG. 12 is a simplified flowchart of a consolidated accumulation process with overrides illustrated from an uplink priority perspective.

DETAILED DESCRIPTION

With reference now to the drawing figures, several exemplary aspects of the present disclosure are described. The word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any aspect described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other aspects.

Aspects disclosed in the detailed description include power saving techniques in computing devices. In particular, as data is received by a modem processor in a computing device, the data is held until the expiration of a modem timer. The data is then passed to an application processor in the computing device over a peripheral component interconnect express (PCIe) interconnectivity bus. On receipt of the data from the modem processor, the application processor sends data held by the application processor to the modem processor over the PCIe interconnectivity bus. The application processor also has an uplink timer. If no data is received from the modem processor before expiration of the uplink timer, the application processor sends any collected data to the modem processor at expiration of the uplink timer. However, if data is received from the modem processor, the uplink timer is reset. By holding or accumulating the data at a source processor in this fashion, unnecessary transitions between low power states and active states on the PCIe bus are reduced and power is conserved.

In an alternate aspect, instead of initiating data transfer based on the expiration of the downlink timer (with or without expiration of the uplink timer), accumulated data transfer may be initiated based on expiration of just an uplink accumulation timer. The uplink accumulation timer may be within a host or a device associated with the interconnectivity bus.

In another alternate aspect, initiation of the data transfer may be based on reaching a predefined threshold for a byte accumulation limit counter. The byte accumulation limit counter is not mutually exclusive relative to the other counters and may operate as an override mechanism for one of the other accumulation timers. Use of such an override may be useful in situations where a sudden burst of data arrives that would exceed buffer space and/or bus bandwidth. Likewise, instead of a byte counter, a packet size counter or a "total number of packets" counter may be used to cover situations where numerous packets or a particularly large packet is delivered by the network.

In further aspects of the present disclosure, the timers may be overridden by other factors or parameters. Such an override is alluded to above with the byte accumulation limit counters and the total number of packets counter, which causes data transfers independently of the timers. Other parameters may also override the timers, such as the presence of low latency traffic (e.g., control messages), synchronizing the uplink and downlink data transfers, or low latency quality of service requirements. When such traffic is present, an interrupt or other command may be used to initiate data transfers before expiration of a timer. Still other factors may override the timers, such as an indication that a device or host is not in an automatic polling mode.

While it is contemplated that the power saving techniques of the present disclosure are used in mobile terminals, such as smart phones or tablets, the present disclosure is not so limited. Accordingly, FIGS. 1A and 1B illustrate computing devices coupled to remote networks via modems that may implement exemplary aspects of the power saving techniques of the present disclosure. In this regard, FIG. 1A illustrates a computing device 10 coupled to a network 12, which, in an exemplary aspect, is the internet. The computing device 10 may include a housing 14 with a central processing unit (CPU) (not illustrated), therein. A user may interact with the computing device 10 through a user interface formed from input/output elements such as a monitor 16 (sometimes referred to as a display), a keyboard 18, and/or a mouse 20. In some aspects, the monitor 16 may be incorporated into the housing 14. While a keyboard 18 and mouse 20 are illustrated input devices, the monitor 16 may be a touchscreen display, which may supplement or replace the keyboard 18 and mouse 20 as an input device. Other input/output devices may also be present as is well understood in conjunction with desktop or laptop style computing devices. While not illustrated in FIG. 1A, the housing 14 may also include a modem, therein. The modem may be positioned on a network interface card (NIC), as is well understood. Likewise, a router and/or an additional modem may be external to the housing 14. For example, the computing device 10 may couple to the network 12 through a router and a cable modem, as is well understood. However, even where such external routers and modems are present, the computing device 10 is likely to have an internal modem to effectuate communication with such external routers and modems.

In addition to the computing device 10, exemplary aspects of the present disclosure may also be implemented on a mobile terminal, which is a form of computing device as that term is used herein. In this regard, an exemplary aspect of a mobile terminal 22 is illustrated in FIG. 1B. The mobile terminal 22 may be a smart phone, such as a SAMSUNG GALAXY™ or APPLE IPHONE®. Instead of a smart phone, the mobile terminal 22 may be a cellular telephone, a tablet, a laptop, or other mobile computing device. The mobile terminal 22 may communicate with a remote antenna 24 associated with a base station (BS) 26. The BS 26 may communicate with the public land mobile network (PLMN) 28, the public switched telephone network (PSTN, not shown), or a network 12 (e.g., the internet), similar to the network 12 in FIG. 1A. It is also possible that the PLMN 28 communicates with the internet (e.g., the network 12) either directly or through an intervening network (e.g., the PSTN). It should be appreciated that most contemporary mobile terminals 22 allow for various types of communication with elements of the network 12. For example, streaming audio, streaming video, and/or web browsing are all common functions on most contemporary mobile terminals 22. Such functions are enabled through applications stored in the memory of the mobile terminal 22 and using the wireless transceiver of the mobile terminal 22.

To effectuate functions, such as streaming video, data arrives from the remote antenna 24 at an antenna 30 of the mobile terminal 22, as illustrated in FIG. 1C. The data is initially processed at a mobile device modem (MDM) 32 of the mobile terminal 22 and passed to an application processor 34 by an interconnectivity bus 36. In this context, the application processor 34 may be a host, and the MDM 32 may be a device as those terms are used in the PCIe standard. While exemplary aspects contemplate operating over a PCIe-compliant interconnectivity bus 36, it is possible that

the interconnectivity bus 36 may comply with High Speed Interconnect (HSIC), Universal Asynchronous Receiver/Transmitter (UART), universal serial bus (USB), or the like.

A more detailed depiction of the components of the mobile terminal 22 is provided with reference to FIG. 2. In this regard, a block diagram of some of the elements of the mobile terminal 22 of FIG. 1B is illustrated. The mobile terminal 22 may include a receiver path 38, a transmitter path 40, the antenna 30 (mentioned above with reference to FIG. 1C), a switch 42, a modem processor 44, and the application processor 34 (also introduced above in reference to FIG. 1C). Optionally, a separate control system (not shown) may also be present with a CPU as is well understood. The application processor 34 and the modem processor 44 are connected by the interconnectivity bus 36. The application processor 34 and/or the control system (if present) may interoperate with a user interface 46 and memory 48 with software 50 stored therein.

The receiver path 38 receives information bearing radio frequency (RF) signals from one or more remote transmitters provided by a base station (e.g., the BS 26 of FIG. 1B). A low noise amplifier (not shown) amplifies the signal. A filter (not shown) minimizes broadband interference in the received signal. Down conversion and digitization circuitry (not shown) down converts the filtered, received signal to an intermediate or baseband frequency signal. The baseband frequency signal is then digitized into one or more digital streams. The receiver path 38 typically uses one or more mixing frequencies generated by the frequency synthesizer. The modem processor 44 may include a base band processor (BBP) (not shown) that processes the digitized received signal to extract the information or data bits conveyed in the signal. As such, the BBP is typically implemented in one or more digital signal processors (DSPs) within the modem processor 44 or as a separate integrated circuit (IC) as needed or desired.

With continued reference to FIG. 2, on the transmit side, the modem processor 44 receives digitized data, which may represent voice, data, or control information, from the application processor 34, which it encodes for transmission. The encoded data is output to the transmitter path 40, where it is used by a modulator (not shown) to modulate a carrier signal at a desired transmit frequency. An RF power amplifier (not shown) amplifies the modulated carrier signal to a level appropriate for transmission, and delivers the amplified and modulated carrier signal to the antenna 30 through the switch 42. Collectively, the modem processor 44, the receiver path 38, and the transmitter path 40 form the MDM 32 of FIG. 1C (sometimes also referred to as a wireless modem). While the MDM 32 is specifically described with relation to the RF signals associated with a cellular signal, the present disclosure is not so limited. For example, a wireless modem using other wireless protocols may also benefit from inclusion of aspects of the present disclosure. Thus, modems operating according to standards such as BLUETOOTH®, the various IEEE 802.11 standards, Universal Mobile Telecommunications System (UMTS), High Speed Packet Access (HSPA), Long Term Evolution (LTE), and other wireless protocols may all use aspects of the present disclosure.

With continued reference to FIG. 2, a user may interact with the mobile terminal 22 via the user interface 46, such as a microphone, a speaker, a keypad, and a display. Audio information encoded in the received signal is recovered by the BBP, and converted into an analog signal suitable for driving the speaker. The keypad and display enable the user to interact with the mobile terminal 22. For example, the

keypad and display may enable the user to input numbers to be dialed, access address book information, or the like, as well as monitor call progress information. The memory 48 may have the software 50 therein as noted above, which may effectuate exemplary aspects of the present disclosure.

In conventional mobile terminals that have a PCIe interconnectivity bus (i.e., the interconnectivity bus 36), the PCIe standard allows the interconnectivity bus 36 to be placed into a sleep mode. While placing the interconnectivity bus 36 in a sleep mode generally saves power, such sleep modes do have a drawback in that they consume relatively large amounts of power as they transition out of the sleep mode. This power consumption is exacerbated because of the asynchronous nature of the PCIe interconnectivity bus 36. That is, first data may arrive at the modem processor 44 for transmission to the application processor 34 at a time different than when the second data is ready to pass from the application processor 34 to the modem processor 44. This problem is not unique to the PCIe interconnectivity bus 36.

FIG. 3 illustrates a time versus link power graph 52 that highlights how downlink data 54 may have a different transmission time than uplink data 56 within a given time slot 58. In particular, the interconnectivity bus 36 (FIG. 2) begins in a sleep or low power mode and transitions up to an active power mode by transition 60 so that the downlink data 54 may be transmitted to the application processor 34. However, the downlink data 54 may not occupy the entirety of the time slot 58, and the interconnectivity bus 36 may return to a low power state. However, subsequently, but still within the same time slot 58, the uplink data 56 from the application processor 34 is sent to the modem processor 44. Accordingly, the interconnectivity bus 36 is again transitioned from the low power state to the active power state by a second transition 62. In an exemplary aspect, the time slot 58 is approximately one millisecond long. Thus, if two transitions (i.e., 60, 62) from low power to active power occur every time slot 58, then thousands of transitions 60, 62 occur every second. Thousands of transitions 60, 62 consume substantial amounts of power and reduce the battery life of the mobile terminal 22.

Exemplary aspects of the present disclosure reduce the number of transitions (i.e., 60, 62) from low power to active power by synchronizing packet transmission from the modem processor 44 and the application processor 34, which in turn allows the link to be maintained in a low power mode more efficiently since the communication on the link is consolidated to eliminate the second power state transition. In an exemplary aspect, the data (i.e., the modem data) from the modem processor 44 transmits first, and the data (i.e., the application data) from the application processor 34 is sent after arrival of the modem data and before the interconnectivity bus 36 can return to the low power state. The synchronization is done through the use of timers at the modem processor 44 and the application processor 34. The timers may be longer than a time slot 58 of the interconnectivity bus 36.

In a first exemplary aspect, the timer on the application processor 34 is longer than the timer on the modem processor 44. The accumulation may be done on a per logical channel basis. The timer may be configurable by the application processor 34 using a mechanism suitable to the interconnectivity bus 36. For example, on a fusion device using a modem host interface (MHI) over PCIe, the timer is maintained for every inbound MHI channel and the time value used by the timers shall be configured via a MHI command message or a PCIe memory mapped input/output (MMIO) device configuration register exposed via a base

address register (BAR). The BAR is a PCIe standard defined mechanism by which a host maps the registers of a device into its virtual address map. For more information about MHI, the interested reader is referred to U.S. patent application Ser. No. 14/163,846, filed Jan. 24, 2013, which is herein incorporated by reference in its entirety. In other exemplary aspects, the timer on the modem processor 44 is longer than the timer on the application processor 34. In still other exemplary aspects, counters may be used in place of timers. The counters may be bit counters, packet counters, packet size counters, or the like. In other exemplary aspects, use of such alternate counters may be combined with the timers. In still other exemplary aspects, other override criteria may allow for data to be sent before timer or counter expiration so as to reduce latency and/or satisfy the quality of service requirements. The present disclosure steps through each of these aspects in turn, beginning with the situation where there are two timers, and the application processor 34 has a timer that is longer than the timer of the modem processor 44.

In this regard, FIG. 4 illustrates an exemplary power saving process 70. The process 70 begins with the interconnectivity bus 36 in a low-power state (block 72). The modem timer and the application timer are started (block 74). The timers may be software stores in the modem processor 44 and the application processor 34 or may be physical elements, as desired. Data is generated by the application processor 34 and data is received from the network 12 by the modem processor 44. The application data is held at the application processor 34 (block 76), and the modem data is held at the modem processor 44 (block 78) while the timers are running. As noted above, in an exemplary aspect, the time slot 58 of the interconnectivity bus 36 is one millisecond. In such an aspect, the modem timer may be approximately two to six milliseconds, and the application timer is three to seven milliseconds, or at least longer than the modem timer. The modem timer expires (block 80). If modem data is present, the modem data is released by the modem processor 44 through the interconnectivity bus 36 to the application processor 34 (block 82).

The mechanism for data transfer may be initiated and controlled by the modem processor 44 (i.e., the device). For example, on a fusion device using MHI over PCIe, the modem processor 44 may poll (read) the MHI channel Context Write Pointer to determine data buffers where downlink packets can be transferred. The application processor 34 updates the channel context data structure's Context Write Pointer field to point to the data transfer descriptors without ringing an Inbound channel doorbell. The modem processor 44 may poll for updates on the Context Write Pointer field as necessitated by downlink traffic. When the modem processor 44 runs out of buffers, i.e., a transfer ring is empty, and no buffers are present to transfer downlink data, the modem processor 44 may generate an event (e.g., an "out-of-buffer") notification to the application processor 34, followed by an interrupt. Upon receiving the event notification from the modem processor 44, the application processor 34 shall provide data buffers by updating the channel Context Write Pointer and shall ring the Inbound channel doorbell.

After arrival of the modem data at the application processor 34, the application processor 34 releases any application data that has been held at the application processor 34 and resets the application timer (block 84). Note that the application timer can run on the modem processor 44 or the application processor 34. As an alternative, the modem processor 44 may continue to pull the uplink data 56 from

the application processor 34 until it detects no further downlink data 54 activity. That is, the modem processor 44 may intersperse pulling the uplink data 56 while receiving the downlink data 54. If, however, no modem data is present at the modem processor 44 when the modem timer expires, the application timer continues (i.e., another millisecond) (block 86). At the expiration of the application timer, the application processor 34 sends any held data to the modem processor 44 through the interconnectivity bus 36 (block 88). The process then repeats by starting over (block 90).

As noted above, the uplink timer (i.e., the application timer) is, in an exemplary aspect, designed to be longer than the downlink timer (i.e., the modem timer) to increase the uplink/downlink synchronization whenever the downlink timer expires. While holding data for an extra time slot adds some latency, the brief amount added is readily absorbed by the application processor 34. Likewise, this latency is considered acceptable for the power savings. For example, by making the period of the modem timer twice the period of the time slot 58, the number of low power to active power transitions is potentially halved. Likewise, by making the period of the application timer six times the period of the time slot 58, the chance of being able to "piggyback" onto the active power state of the interconnectivity bus 36 caused by the modem data is increased, but still frequent enough that any uplink data 56 will still be sent in a timely fashion even if there is no downlink data 54 to trigger releasing the uplink data 56. Similar logic can be extended to synchronize traffic from multiple processors over the data link. In an exemplary aspect, the other processors may each have timer values higher (i.e., longer) than that of the downlink timer, and the processors can exchange their data availability information so that traffic on one processor can trigger the data transfer on other processors if there is data available to transfer.

FIG. 5 illustrates a graph 100 where the uplink data 56 follows the downlink data 54 during an active period 102 of the interconnectivity bus 36 (FIG. 2). As illustrated, there is only one transition 104 from low power to active power per time slot 58. Thus, by consolidating the data into a single active period 102, the overall time that is spent in low power may be increased, thus resulting in power savings. Additionally, power spent transitioning from a low power to active power state is reduced by the elimination of the second transition 62.

While it is conceivable that the uplink data 56 could be sent before the downlink data 54 (i.e., the application timer is shorter than the modem timer), such is generally not considered optimal because there are usually far more downlink packets than uplink packets. If this aspect is used, the application processor 34 may buffer uplink data packets into local memory prior to initiating transfer to the modem processor 44. These accumulated packets are controlled via an uplink accumulation timer. If there are plural channels, then a timer may be applied to each channel independently. When the application processor 34 is unable to use or does not have an uplink timer, the modem processor 44 may be able to instantiate an uplink timer, and upon expiry of the uplink timer, will poll data from the application processor 34. This exemplary aspect is explained in greater detail below with reference to FIGS. 6 and 7.

In this regard, FIG. 6 illustrates an exemplary power saving process 110. The process 110 begins with the interconnectivity bus 36 in a low power state (block 112). The modem timer and the application timer are started (block 114). The timers may be software stores in the modem processor 44 and the application processor 34 or may be

physical elements as desired. Data is generated by the application processor 34 and data is received from the network 12 by the modem processor 44. The application data is held at the application processor 34 (block 116), and the modem data is held at the modem processor 44 (block 118) while the timers are running. As noted above, in an exemplary aspect, the time slot 58 of the interconnectivity bus 36 is one millisecond. In such an aspect, the application timer may be approximately two milliseconds, and the modem timer is three milliseconds, or at least longer than the application timer. The application timer expires (block 120). If application data is present, the application data is released by the application processor 34 through the interconnectivity bus 36 to the modem processor 44 (block 122).

After arrival of the application data at the modem processor 44, the modem processor 44 releases any modem data that has been held at the modem processor 44 and resets the modem timer (block 124). Note that the application timer can run on the modem processor 44 or the application processor 34. Likewise, the modem timer can run on the modem processor 44 or the application processor 34.

With continued reference to FIG. 6, if no application data is present at the application processor 34 when the application timer expires, the modem timer continues (i.e., another millisecond) (block 126). At the expiration of the modem timer, the modem processor 44 sends any held data to the application processor 34 through the interconnectivity bus 36 (block 128). The process then repeats by starting over (block 130).

As noted above, in this exemplary aspect, the uplink timer (i.e., the application timer) is, in an exemplary aspect, designed to be shorter than the downlink timer (i.e., the modem timer). While holding data for an extra time slot 58 adds some latency, the brief amount added is readily absorbed by the application processor 34. Likewise, this latency is considered acceptable for the power savings. For example, by making the period of the application timer twice the period of the time slot 58, the number of low power to active power transitions is lowered. Likewise, by making the period of the modem timer six times the period of the time slot 58, the chance of being able to "piggyback" onto the active power state of the interconnectivity bus 36 caused by the application data is increased, but still frequent enough that any downlink data 54 will still be sent in a timely fashion even if there is no uplink data 56 to trigger releasing the downlink data 54. Similar logic can be extended to synchronize traffic from multiple processors over the data link. In an exemplary aspect, the other processors may each have timer values higher (i.e., longer) than that of the uplink timer and the processors can exchange their data availability information so that traffic on one processor can trigger the data transfer on other processors if there is data available to transfer.

FIG. 7 illustrates a graph 140 where the uplink data 56 precedes the downlink data 54 during an active period 142 of the interconnectivity bus 36. As illustrated, there is only one transition 144 from low power to active power per time slot 58. Thus, by consolidating the data into a single active period 142, the overall time that is spent in low power may be increased, thus resulting in power savings. Additionally, power spent transitioning from a low power to active power state is reduced by the elimination of the second transition 62.

In an exemplary aspect, the modem processor 44 may override and choose the minima from all configured values of each of the configurable parameters (like downlink or uplink accumulation timers, byte threshold, number of pack-

ets threshold, size of packet threshold, or the like) or downlink accumulation expiry timer values (e.g., from among the various channels) as the effective downlink accumulation timer expiry value. Intelligent modem processors 44 may also dynamically override or alter the downlink accumulation timer value depending on the downlink traffic pattern, and/or may adjust the downlink accumulation timer to achieve a desired quality of service (QoS) for data and/or to control traffic. A change of configuration can be triggered/controlled by the application processor 44 or any other processor in the system as well, via MHI control or QMI signaling (such as, for inter process signaling).

In addition to, or in place of, downlink and uplink timers, a byte accumulation limit counter may also be used by the modem processor 44 for downlink traffic and the application processor 34 for uplink traffic. This aspect may be advantageous in situations where there is a sudden burst of data pushed by the network or application. Note that this aspect is not mutually exclusive and may be implemented as an override mechanism for either downlink or uplink timers. For example, if the downlink accumulation timer is set relatively high to conserve power, a sudden burst of data may exceed the buffer capacity of the modem processor 44, or if allowed to accumulate in memory of the modem processor 44, this burst of data may exceed bus bandwidth allocations on the application processor 34. The application processor 34 can determine and configure the maximum byte accumulation limit based on its bus bandwidth budget, and/or buffer size reserved for downlink data transfer. The modem processor 44 can also choose an internal byte accumulation limit based on the size of downlink buffer, and/or interconnect link data throughput. With the byte accumulation limit counters, the modem processor 44 can initiate downlink data transfer to the application processor 34 prior to downlink accumulation timer expiry, if and when the buffered data size exceeds the byte accumulation limit counter. Since both the modem processor 44 and the application processor 34 may have independent recommendations for byte accumulation limit counter, the modem processor 44 may select the minima of these two values to be the effective byte accumulation limit. Similar parameters may be maintained in the application processor 34 to trigger the uplink data 56 transfer immediately (i.e., overriding the uplink accumulation timer).

Instead of, or in addition to the byte accumulation limit counter, a number of packets limit counter may be used. In an exemplary aspect, the packet number limit counter may be of similar design, and can be employed to add number of packet counter limits instead of byte limits to cover cases where a large number of packets are delivered by the network or an application. Again, such a packet limit counter may also be present or associated with the application processor 34 or the modem processor 44. Note, that the accumulation timers (uplink and/or downlink) and other configuration parameters like the number of accumulated packets threshold, accumulated bytes threshold, and the like, may be a function of LTE, HSPA, GERAN, or the like.

In still another exemplary aspect, the modem processor 44 or the application processor 34 may disable downlink or uplink accumulation in cases where there is a necessity to expedite message transfer, for example control messages (like flow control) or high QoS data traffic or low latency traffic, as determined by the modem processor 44 or the application processor 34. Latency introduced by accumulation may not be tolerable for these traffic classes.

Returning to the data accumulation based on amounts of data instead of a strict process, FIGS. 8 and 9 illustrate two

13

exemplary aspects. In this regard, FIG. 8 illustrates a process 150 illustrating a byte counter process. In particular, the process 150 begins with the interconnectivity bus 36 in a low power state (block 152). The process 150 starts a modem byte counter and an application byte counter (block 154). Data is held at the application processor 34 (block 156) and the modem processor 44 (block 158). A control system determines if the modem byte counter has exceeded a predefined threshold (block 160) based on the amount of data that has been held or accumulated at the modem processor 44.

With continued reference to FIG. 8, if the answer to block 160 is yes, then data is sent from the modem processor 44 to the application processor 34 (block 162). After receipt of the data from the modem processor 44, the application processor 34 sends data (if any) that has accumulated at the application processor 34 to the modem processor 44 (block 164). Having cleared the accumulated data at both the modem processor 44 and the application processor 34, the process starts over (block 166).

With continued reference to FIG. 8, if the answer to block 160 is no, the control system determines if the data at the application byte counter has exceeded a predefined threshold (block 168). If the answer to block 168 is no, the process 150 returns to block 156 and data continues to be held until a byte counter threshold is exceeded. If, however, the answer to block 168 is yes, then the data is sent from the application processor 34 to the modem processor 44 (block 170). After receipt of the data from the application processor 34, the modem processor 44 sends data (if any) to the application processor 34 (block 172). Having cleared the accumulated data at both the modem processor 44 and the application processor 34, the process 150 starts over (block 166).

While a byte counter may be effective in managing latency, another exemplary aspect uses a packet counter. In this regard, FIG. 9 illustrates a process 180 illustrating a byte counter process. In particular, the process 180 begins with the interconnectivity bus 36 in a low power state (block 182). The process 180 starts a modem packet counter and an application packet counter (block 184). Data is held at the application processor 34 (block 186) and the modem processor 44 (block 188). A control system determines if the modem packet counter has exceeded a predefined threshold (block 190) based on the number of packets held or accumulated at the modem processor 44.

With continued reference to FIG. 9, if the answer to block 190 is yes, then data is sent from the modem processor 44 to the application processor 34 (block 192). After receipt of the packets from the modem processor 44, the application processor 34 sends data (if any) that has accumulated at the application processor 34 to the modem processor 44 (block 194). Having cleared the accumulated packets at both the modem processor 44 and the application processor 34, the process 180 starts over (block 196).

With continued reference to FIG. 9, if the answer to block 190 is no, the control system determines if the number of packets at the application packet counter has exceeded a predefined threshold (block 198). If the answer to block 198 is no, the process 180 returns to block 186 and data continues to be held until a packet counter threshold is exceeded. If, however, the answer to block 198 is yes, then the data is sent from the application processor 34 to the modem processor 44 (block 200). After receipt of the packets from the application processor 34, the modem processor 44 sends data (if any) to the application processor 34 (block 202). Having cleared the accumulated packets at

14

both the modem processor 44 and the application processor 34, the process 180 starts over (block 196).

A similar process may be used, where instead of determining if a particular number of bytes or packets have been accumulated, the control system evaluates a size of packets or whether the system is running low in memory. Likewise, it should be appreciated that certain priority data (e.g., a control signal or other data requiring low latency) may be associated with a flag or other indicator that overrides the timers and/or counters of the present disclosure.

As noted above, it should be appreciated that the aspects of the present disclosure are not mutually exclusive and can be combined. The combinations are myriad in that a timer may be used at the application processor 34 with a byte counter at the modem processor 44 (or vice versa), the modem processor 44 works with a timer and a byte counter, while the application processor 34 just has a timer, and so on. In this regard, FIGS. 10-12 are provided that illustrate how the timers and data accumulation counters may interoperate. That is, FIGS. 10 and 11 illustrate how the downlink timer (whether in the modem processor 44 or the application processor 34) is used as the basis for data transmission (e.g., as illustrated in FIGS. 4 and 5), may be combined with the data accumulation counters, and is further modified by a high priority data override. FIG. 12 illustrates a simplified process in which the uplink timer combined with the data accumulation counters is used as the basis for data transmission (e.g., as illustrated in FIGS. 6 and 7), modified by the data overrides.

In this regard, FIGS. 10 and 11 illustrate a combined process 210 that begins at start (block 212). The process 210 continues with the arrival of downlink (DL) data (e.g., a packet) (block 214). The control system evaluates if there is any priority data, control messages, and/or other data that requires low latency (block 216). If the answer to block 216 is no, then the control system determines if a byte threshold has been crossed (i.e., are there more than the threshold worth of bytes in the accumulator) (block 218). If the answer to block 218 is no, then the control system determines if a number of packets threshold has been crossed (i.e., there are more than the threshold worth of packets in the accumulator) (block 220). If the answer to block 220 is no, then the control system determines if the system is running low in memory (block 222). If the answer to block 222 is no, then the control system ascertains if the downlink accumulation timer is running (block 224). If the answer to block 224 is yes, then the downlink data continues to accumulate and no data transfer is initiated over the link (block 226).

With continued reference to FIG. 10, if, however, the answer to block 224 is no, the downlink timer has expired, or if any of the overrides from block 216, 218, 220, or 222 has been answered affirmatively, then the process 210 starts transfer of the accumulated data (including the current packet) over the link from the modem processor 44 (also sometimes referred to as modem (44) in the Figures) to the application processor 34 (also sometimes referred to as AP (34) in the Figures) (block 230). The control system starts or restarts the downlink accumulation timer and sets the downlink accumulation timer to running (block 232). The control system determines if the modem processor 44 is in an uplink (UL) polling mode (block 234). If the answer to block 234 is no, then there is no uplink transfer (block 236). If, however, the modem processor 44 is polling the uplink device, then, based on that polling, the control system determines if there is pending uplink data from the application processor 34 (block 238). If there is pending data (i.e., the answer to block 238 is yes), then the application pro-

cessor 34 starts data transfer to the modem processor 44 (block 240). Once the data transfer is finished, or if there was no data at block 238, the control system restarts the uplink accumulation timer (block 242) and the process 210 returns to start 212.

With continued reference to FIG. 10, after block 226, the control system determines if the downlink timer has expired (block 244). If the answer to block 244 is no, the control system determines if a new packet has arrived (block 246). If the answer to block 246 is no, then the process 210 returns to block 244. If a new packet has arrived, the process 210 returns to the start 212. If the answer to block 244 is yes, the downlink timer has expired, the control system knows the downlink accumulation timer has expired (block 248). At expiration of the downlink timer, the control system determines if there is any pending accumulated downlink data (block 250). If there is data at block 250, then the data is transferred at block 230. If there is no data, then the downlink accumulation timer is set to "not running" (block 252) and the process 210 goes to FIG. 11, element C. It should be appreciated that blocks 216, 218, 220, and 222 are optional.

With reference to FIG. 11, the process 210 may continue from block 252. At this point, the uplink accumulation timer has expired (block 254). The uplink accumulation timer will expire if there is no downlink data since the uplink timer was restarted. The control system determines if there is any pending uplink data from the application processor 34 (block 256). If the answer to block 256 is yes, then the application processor 34 starts the data transfer over the link from the application processor 34 to the modem processor 44 (block 258). The control system then restarts the uplink accumulation timer (block 260). If, however, the answer to block 256 is no, there is no data, the control system sends an event to the application processor 34 indicating the modem processor 44 is expecting a doorbell/interrupt for any pending or next packet submission (block 262). That is, since there has been no data from the application processor 34 to the modem processor 44 since the previous poll time, then the modem processor 44 may go into an interrupt mode for uplink data and the modem processor 44 would expect the application processor 34 to send an interrupt whenever there was data pending at the application processor 34. The control system then changes the state internally to reflect the same (block 264).

With continued reference to FIG. 11, the modem processor 44 receives an interrupt or other indication from the application processor 34 indicating pending data in the transfer ring (block 266). The control system then restarts the uplink accumulation timer and changes the state to indicate the uplink polling mode (block 268). All the uplink data is processed (block 270) and the process 210 starts over.

In another alternate aspect, there may be situations where the buffers of the application processor 34 may be full and there is no room for data from the modem processor 44. In such an event, the application processor 34 may so inform the modem processor 44, and the modem processor 44 may send an event to the application processor 34 to provide an interrupt signal to the modem processor 44 when there are free buffers.

FIG. 12 is similar to FIGS. 10 and 11, in that it illustrates how overrides and data counters may be used in conjunction with an accumulation timer, but a process 280 of FIG. 12 assumes that the uplink timer is shorter than the downlink timer (e.g., analogous to the aspect illustrated in FIGS. 6 and 7). The process 280 begins at start (block 282). The process 280 continues with the arrival of uplink (UL) data (e.g., a

packet) (block 284). The control system evaluates if there is any priority data, control messages, and/or other data that requires low latency (block 286). If the answer to block 286 is no, then the control system determines if a byte threshold has been crossed (i.e., are there more than the threshold worth of bytes in the accumulator) (block 288). If the answer to block 288 is no, then the control system determines if a number of packets threshold has been crossed (i.e., there are more than the threshold worth of packets in the accumulator) (block 290). If the answer to block 290 is no, then the control system determines if the system is running low in memory (block 292). If the answer to block 292 is no, then the control system ascertains if the device is in an uplink polling mode (block 294). If the answer to block 294 is yes, the device is in the polling mode, then the application processor 34 updates the internal data structure/context array with uplink data packet information that the device can pull and update write pointers accordingly (block 296).

With continued reference to FIG. 12, if, however, the answer to block 294 is no, then the device is not in an uplink polling mode, or if any of the overrides from blocks 286, 288, 290, or 292 has been answered affirmatively, then the application processor 34 updates the internal data structure/context array with uplink data packet information that the device can pull and update write pointers accordingly (block 298). The application processor 34 then rings the doorbell or otherwise interrupts the device to indicate the availability of uplink data (block 300). The application processor 34 then sets the device state to the polling state (not the doorbell/event/interrupt mode) (block 302), and the process repeats.

It should be appreciated that similar processes may be performed where both timers are in the application processor 34 or the modem processor 44 or are split between the respective processors 34, 44. Likewise, once a timer has expired, data can be pulled or pushed across the interconnectivity bus 36 based on polling, setting doorbell registers, or other technique.

Those of skill in the art will further appreciate that the various illustrative logical blocks, modules, circuits, and algorithms described in connection with the aspects disclosed herein may be implemented as electronic hardware, instructions stored in memory or in another computer-readable medium and executed by a processor or other processing device, or combinations of both. The devices described herein may be employed in any circuit, hardware component, IC, or IC chip, as examples. Memory disclosed herein may be any type and size of memory and may be configured to store any type of information desired. To clearly illustrate this interchangeability, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. How such functionality is implemented depends upon the particular application, design choices, and/or design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present disclosure.

The various illustrative logical blocks, modules, and circuits described in connection with the aspects disclosed herein may be implemented or performed with a processor, a Digital Signal Processor (DSP), an Application Specific Integrated Circuit (ASIC), a Field Programmable Gate Array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A processor may be a microprocessor, but in the

alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

The aspects disclosed herein may be embodied in hardware and in instructions that are stored in hardware, and may reside, for example, in Random Access Memory (RAM), flash memory, Read Only Memory (ROM), Electrically Programmable ROM (EPROM), Electrically Erasable Programmable ROM (EEPROM), registers, a hard disk, a removable disk, a CD-ROM, or any other form of computer readable medium known in the art. An exemplary storage medium is coupled to the processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an ASIC. The ASIC may reside in a remote station. In the alternative, the processor and the storage medium may reside as discrete components in a remote station, base station, or server.

It is also noted that the operational steps described in any of the exemplary aspects herein are described to provide examples and discussion. The operations described may be performed in numerous different sequences other than the illustrated sequences. Furthermore, operations described in a single operational step may actually be performed in a number of different steps. Additionally, one or more operational steps discussed in the exemplary aspects may be combined. It is to be understood that the operational steps illustrated in the flow chart diagrams may be subject to numerous different modifications as will be readily apparent to one of skill in the art. Those of skill in the art will also understand that information and signals may be represented using any of a variety of different technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description may be represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof.

The previous description of the disclosure is provided to enable any person skilled in the art to make or use the disclosure. Various modifications to the disclosure will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other variations without departing from the spirit or scope of the disclosure. Thus, the disclosure is not intended to be limited to the examples and designs described herein, but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

What is claimed is:

1. A mobile terminal comprising:
 - a modem timer;
 - a modem processor, the modem processor configured to hold modem processor to application processor data until expiration of the modem timer;
 - an application processor;
 - an interconnectivity bus communicatively coupling the application processor to the modem processor; and
 - the application processor configured to hold application processor to modem processor data until triggered by receipt of the modem processor to application processor data from the modem processor through the interconnectivity bus after which the application processor

to modem processor data is sent to the modem processor through the interconnectivity bus responsive to the receipt of the modem processor to application processor data from the modem processor through the interconnectivity bus.

2. The mobile terminal of claim 1, wherein the interconnectivity bus comprises a peripheral component interconnect (PCI) compliant bus.

3. The mobile terminal of claim 2, wherein the PCI compliant bus comprises a PCI express (PCIe) bus.

4. The mobile terminal of claim 1, wherein the application processor includes an uplink timer and the uplink timer has a period longer than a period of the modem timer.

5. The mobile terminal of claim 4, wherein the application processor is configured to hold the application processor to modem processor data until receipt of the modem processor to application processor data from the modem processor or expiration of the uplink timer having a period longer than a period of the modem timer, whichever occurs first.

6. The mobile terminal of claim 1, wherein the modem timer is implemented in software.

7. The mobile terminal of claim 1, wherein the modem timer has a period of approximately six (6) milliseconds.

8. The mobile terminal of claim 1, wherein the modem processor comprises the modem timer.

9. The mobile terminal of claim 1, wherein the application processor comprises the modem timer.

10. The mobile terminal of claim 1, further comprising an application timer, and wherein the modem processor is configured to instruct the application processor to send an interrupt if no data is received within one time slot of the application timer.

11. The mobile terminal of claim 1, further comprising a byte accumulation limit counter associated with the modem processor, the modem processor configured to send data to the application processor if a threshold associated with the byte accumulation limit counter is exceeded.

12. The mobile terminal of claim 1, further comprising a packet number limit counter associated with the modem processor, the modem processor configured to send data to the application processor if a threshold associated with the packet number limit counter is exceeded.

13. The mobile terminal of claim 1, wherein the modem processor is configured to determine if held data comprises a control packet and send such control packet before expiration of the modem timer.

14. The mobile terminal of claim 3, wherein the modem processor further comprises an application timer, and the modem processor is configured to pull data from the application processor on receipt of the modem processor to application processor data or expiration of the application timer.

15. The mobile terminal of claim 1, further comprising a second modem processor, the second modem processor configured to exchange data availability information with the modem processor such that traffic on the modem processor can trigger data transfer for the second modem processor.

16. A method of controlling power consumption in a computing device, comprising:

- holding data received by a modem processor from a remote network until expiration of a downlink timer;
- passing the data received by the modem processor to an application processor over an interconnectivity bus; and
- holding application data generated by an application associated with the application processor until receipt

19

of the data from the modem processor or expiration of an uplink timer, whichever occurs first, wherein receipt of the data from the modem processor triggers passing the data received by the application processor to the modem processor over the interconnectivity bus before the interconnectivity bus transitions from an active power state to a low power state.

17. The method of claim 16, wherein passing the data comprises passing the data over a peripheral component interface (PCI) compliant bus.

18. The method of claim 16, wherein a period of the downlink timer comprises six (6) milliseconds.

19. The method of claim 16, wherein a period of the uplink timer comprises seven (7) milliseconds.

20. The method of claim 16, further comprising providing an override capability based on one of accumulated packet size, accumulated packet count, accumulated byte count, quality of service requirement, and control message status.

21. The method of claim 16, further comprising holding data at a second modem processor until traffic on the modem processor triggers data transfer for the second modem processor.

22. A mobile terminal comprising:

a modem processor;

an application timer;

an application processor, the application processor configured to hold application processor to modem processor data until expiration of the application timer;

an interconnectivity bus communicatively coupling the application processor to the modem processor; and

the modem processor configured to hold modem processor to application processor data until triggered by receipt of the application processor to modem processor data from the application processor through the interconnectivity bus after which the modem processor to application processor data is sent to the application processor through the interconnectivity bus responsive to the receipt of the application processor to modem processor data from the application processor through the interconnectivity bus.

23. The mobile terminal of claim 22, wherein the application processor comprises the application timer.

24. The mobile terminal of claim 22, wherein the modem processor comprises the application timer.

25. The mobile terminal of claim 22, further comprising a byte counter counting bytes at the modem processor.

26. A mobile terminal comprising:

a modem byte accumulation limit counter;

a modem processor, the modem processor configured to hold modem processor to application processor data until a predefined threshold of bytes has been reached by the modem byte accumulation limit counter;

an application processor;

an interconnectivity bus communicatively coupling the application processor to the modem processor; and

the application processor configured to hold application processor to modem processor data until triggered by receipt of the modem processor to application processor data from the modem processor through the interconnectivity bus after which the application processor to modem processor data is sent to the modem processor through the interconnectivity bus responsive to the receipt of the modem processor to application processor data from the modem processor through the interconnectivity bus.

27. A mobile terminal comprising:

a modem packet counter;

20

a modem processor, the modem processor configured to hold modem processor to application processor data until a predefined threshold of packets has been reached by the modem packet counter;

an application processor;

an interconnectivity bus communicatively coupling the application processor to the modem processor; and

the application processor configured to hold application processor to modem processor data until triggered by receipt of the modem processor to application processor data from the modem processor through the interconnectivity bus after which the application processor to modem processor data is sent to the modem processor through the interconnectivity bus responsive to the receipt of the modem processor to application processor data from the modem processor through the interconnectivity bus.

28. A mobile terminal comprising:

a modem processor;

an application byte counter;

an application processor, the application processor configured to hold application processor to modem processor data until a predefined threshold of bytes has been reached by the application byte counter;

an interconnectivity bus communicatively coupling the application processor to the modem processor; and

the modem processor configured to hold modem processor to application processor data until triggered by receipt of the application processor to modem processor data from the application processor through the interconnectivity bus after which the modem processor to application processor data is sent to the application processor through the interconnectivity bus responsive to the receipt of the application processor to modem processor data from the application processor through the interconnectivity bus.

29. A mobile terminal comprising:

a modem processor;

an application packet counter;

an application processor, the application processor configured to hold application processor to modem processor data until a predefined threshold of packets has been reached by the application packet counter;

an interconnectivity bus communicatively coupling the application processor to the modem processor; and

the modem processor configured to hold modem processor to application processor data until triggered by receipt of the application processor to modem processor data from the application processor through the interconnectivity bus after which the modem processor to application processor data is sent to the application processor through the interconnectivity bus responsive to the receipt of the modem processor to application processor data from the modem processor through the interconnectivity bus.

30. A method comprising:

starting an application timer at an application processor; accumulating data at the application processor until expiration of the application timer;

sending the accumulated data from the application processor to a modem processor across an interconnectivity bus; and

holding modem processor data at the modem processor until triggered by receipt of the accumulated data from the application processor,

wherein receipt of the accumulated data from the application processor triggers passing the modem processor

data to the application processor over the interconnectivity bus before the interconnectivity bus transitions from an active power state to a low power state.

31. A mobile terminal comprising:

- a modem timer; 5
- a modem processor, the modem processor configured to hold modem processor to application processor data until expiration of the modem timer;
- an application processor;
- an interconnectivity bus communicatively coupling the 10 application processor to the modem processor; and the application processor configured to hold application processor to modem processor data until the modem processor pulls data from the application processor after transmission of the modem processor to applica- 15 tion processor data,
- wherein the modem processor is further configured pull data from the application processor after transmission of the modem processor to application processor data and before the interconnectivity bus transitions from an 20 active power state to a low power state.

* * * * *

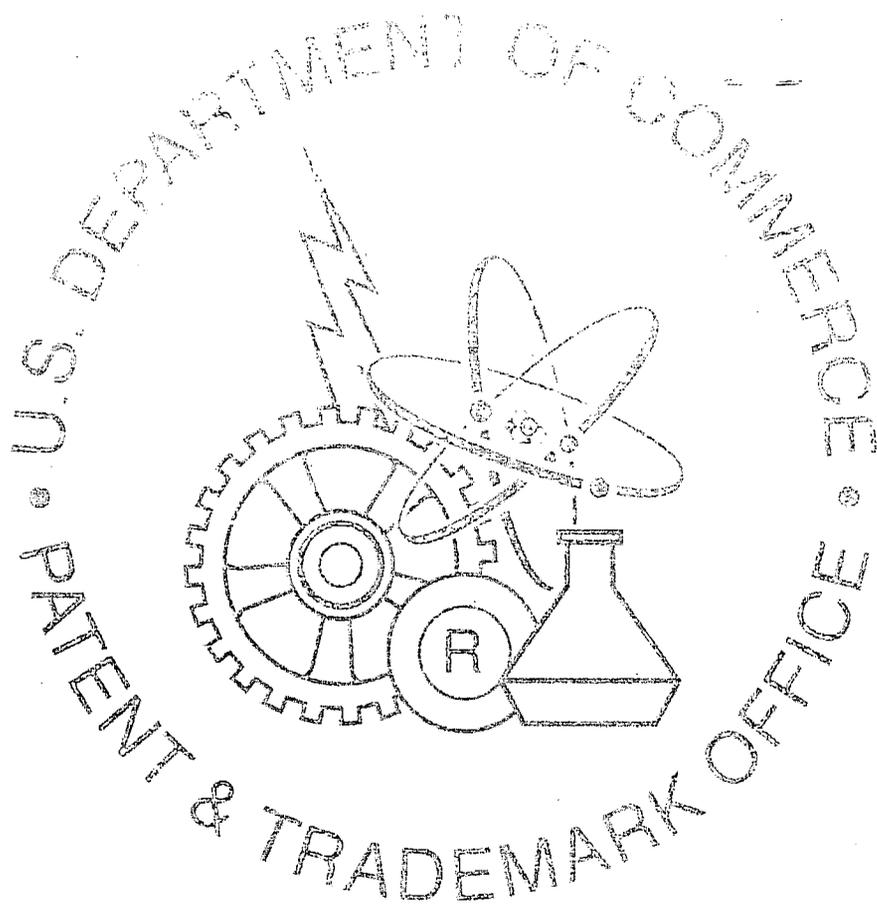


EXHIBIT F

United States
Department of Commerce

U 7629935



THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

April 06, 2017

**THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM
THE RECORDS OF THIS OFFICE OF:**

**U.S. PATENT: 9,608,675
ISSUE DATE: March 28, 2017**

**By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office**



**T. LAWRENCE
Certifying Officer**

(12) United States Patent
Dorosenco

(10) Patent No.: US 9,608,675 B2
(45) Date of Patent: Mar. 28, 2017

(54) POWER TRACKER FOR MULTIPLE TRANSMIT SIGNALS SENT SIMULTANEOUSLY

USPC 330/84, 127, 251; 370/210, 319, 338;
 375/135, 146, 219, 257, 260, 295, 297;
 455/69, 101, 522
 See application file for complete search history.

(71) Applicant: QUALCOMM Incorporated, San Diego, CA (US)

(56) References Cited

(72) Inventor: Alexander Dorosenco, San Diego, CA (US)

U.S. PATENT DOCUMENTS

(73) Assignee: QUALCOMM INCORPORATED, San Diego, CA (US)

6,009,090 A * 12/1999 Oishi et al. 370/342
 7,092,683 B2 * 8/2006 Tanaka et al. 455/108
 (Continued)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 134 days.

FOREIGN PATENT DOCUMENTS

(21) Appl. No.: 13/764,328

GB 2476393 A 6/2011
 GB 2488380 A 8/2012

(22) Filed: Feb. 11, 2013

OTHER PUBLICATIONS

(65) Prior Publication Data
 US 2014/0226748 A1 Aug. 14, 2014

International Search Report and Written Opinion—PCT/US2014/013805—ISA/EPO—Mar. 20, 2014.

(51) Int. Cl.
H04B 1/04 (2006.01)
H04W 52/52 (2009.01)
H03F 1/02 (2006.01)
H03F 3/195 (2006.01)
H03F 3/21 (2006.01)
H03F 3/24 (2006.01)
H03F 3/68 (2006.01)

Primary Examiner — Shawkat M Ali
(74) Attorney, Agent, or Firm — Haynes and Boone, LLP

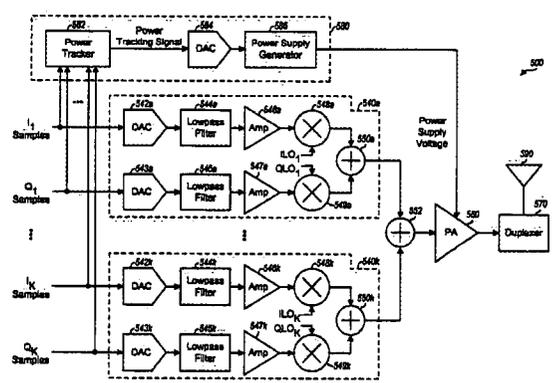
(52) U.S. Cl.
 CPC **H04B 1/04** (2013.01); **H03F 1/0227** (2013.01); **H03F 3/195** (2013.01); **H03F 3/211** (2013.01); **H03F 3/245** (2013.01); **H03F 3/68** (2013.01); **H04W 52/52** (2013.01); **H03F 2200/336** (2013.01); **H03F 2200/462** (2013.01)

(57) ABSTRACT

Techniques for generating a power tracking supply voltage for a circuit (e.g., a power amplifier) are disclosed. The circuit may process multiple transmit signals being sent simultaneously on multiple carriers at different frequencies. In one exemplary design, an apparatus includes a power tracker and a power supply generator. The power tracker determines a power tracking signal based on inphase (I) and quadrature (Q) components of a plurality of transmit signals being sent simultaneously. The power supply generator generates a power supply voltage based on the power tracking signal. The apparatus may further include a power amplifier (PA) that amplifies a modulated radio frequency (RF) signal based on the power supply voltage and provides an output RF signal.

(58) Field of Classification Search
 CPC . H04B 1/04; H04B 1/69; H04B 1/707; H04B 7/00; H04B 7/005; H03F 1/34; H03F 3/68; H03F 3/217; H03G 3/20; H04K 1/10; H04L 27/14

33 Claims, 10 Drawing Sheets



(56)

References Cited

U.S. PATENT DOCUMENTS

8,995,567 B2 * 3/2015 Rofougaran et al. 375/297
2006/0264186 A1 * 11/2006 Akizuki H03C 5/00
455/108
2008/0139140 A1 * 6/2008 Matero et al. 455/114.3
2010/0291963 A1 11/2010 Patel et al.
2011/0142156 A1 * 6/2011 Haartsen 375/271
2011/0151806 A1 * 6/2011 Kenington 455/101
2011/0193629 A1 8/2011 Hou et al.
2012/0033656 A1 2/2012 De Maaijer
2012/0039418 A1 2/2012 Vaisanen
2012/0214423 A1 8/2012 Wallace
2012/0229208 A1 * 9/2012 Wimpenny et al. 330/127
2012/0321018 A1 * 12/2012 Chen et al. 375/296
2012/0326686 A1 * 12/2012 Dai et al. 323/283
2012/0326783 A1 12/2012 Mathe et al.
2014/0111275 A1 * 4/2014 Khlal et al. 330/124 R
2014/0199949 A1 * 7/2014 Nagode et al. 455/73

* cited by examiner

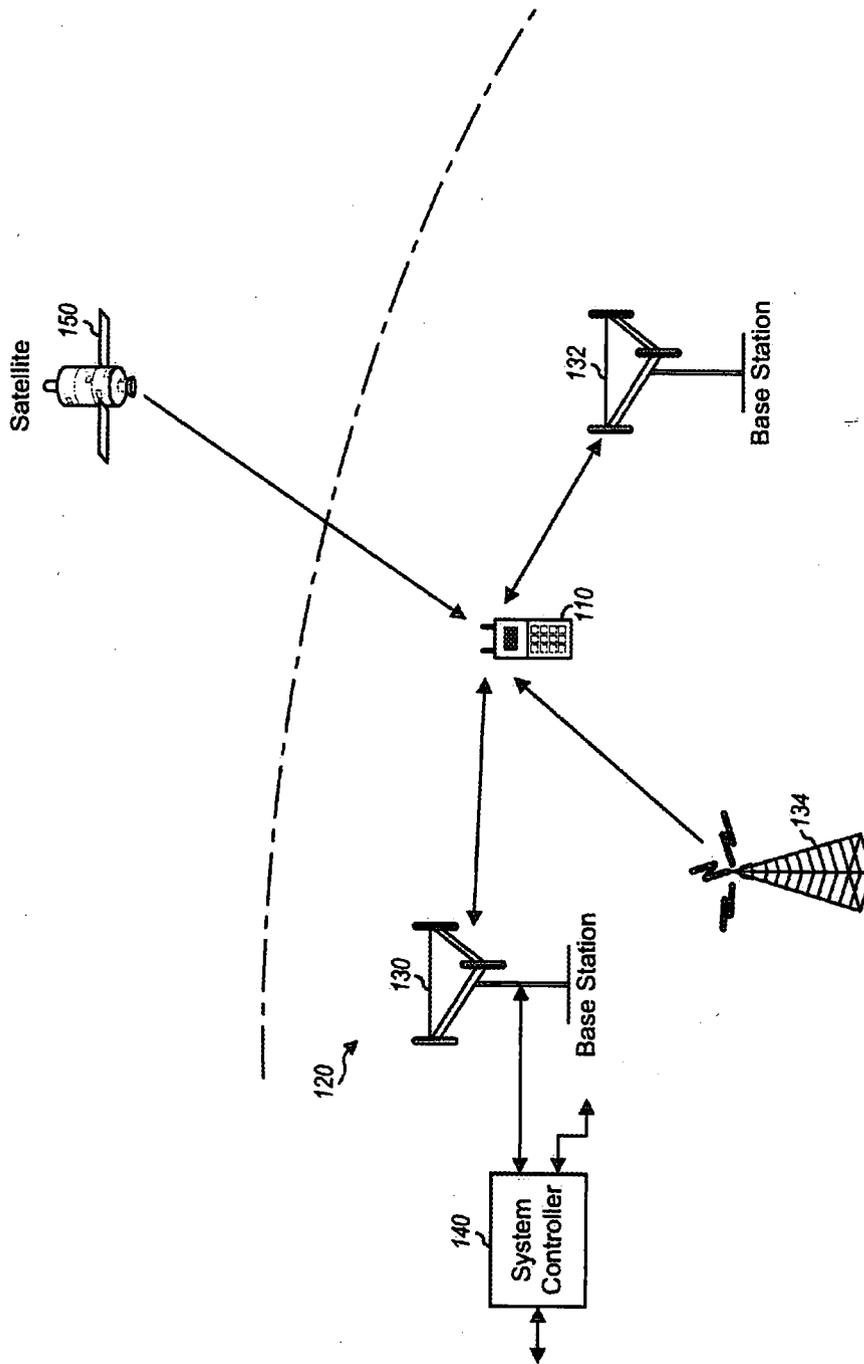
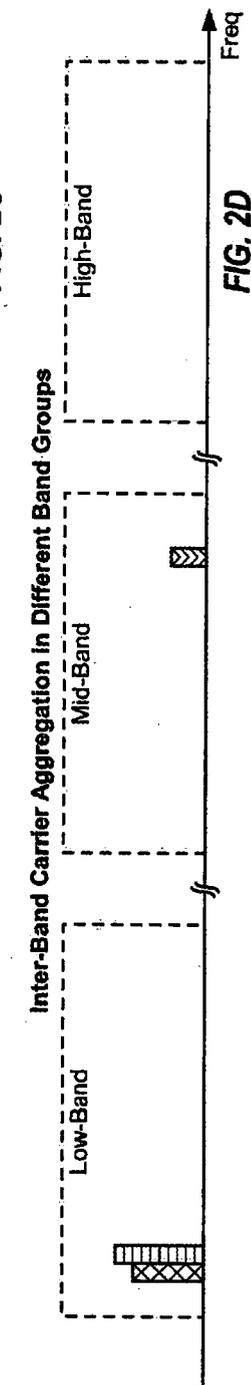
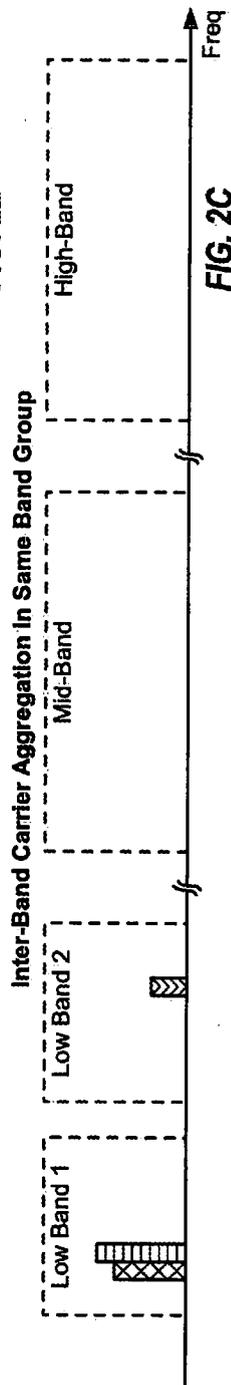
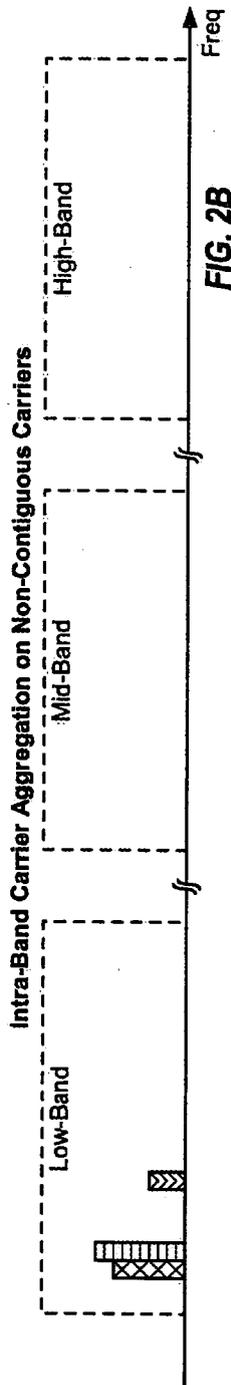
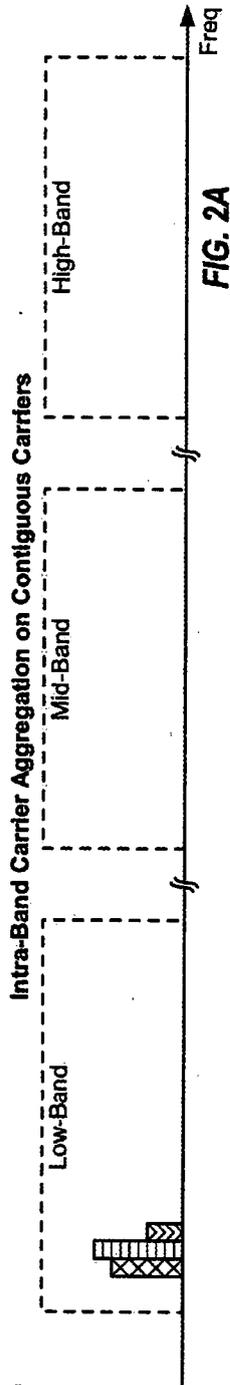


FIG. 1



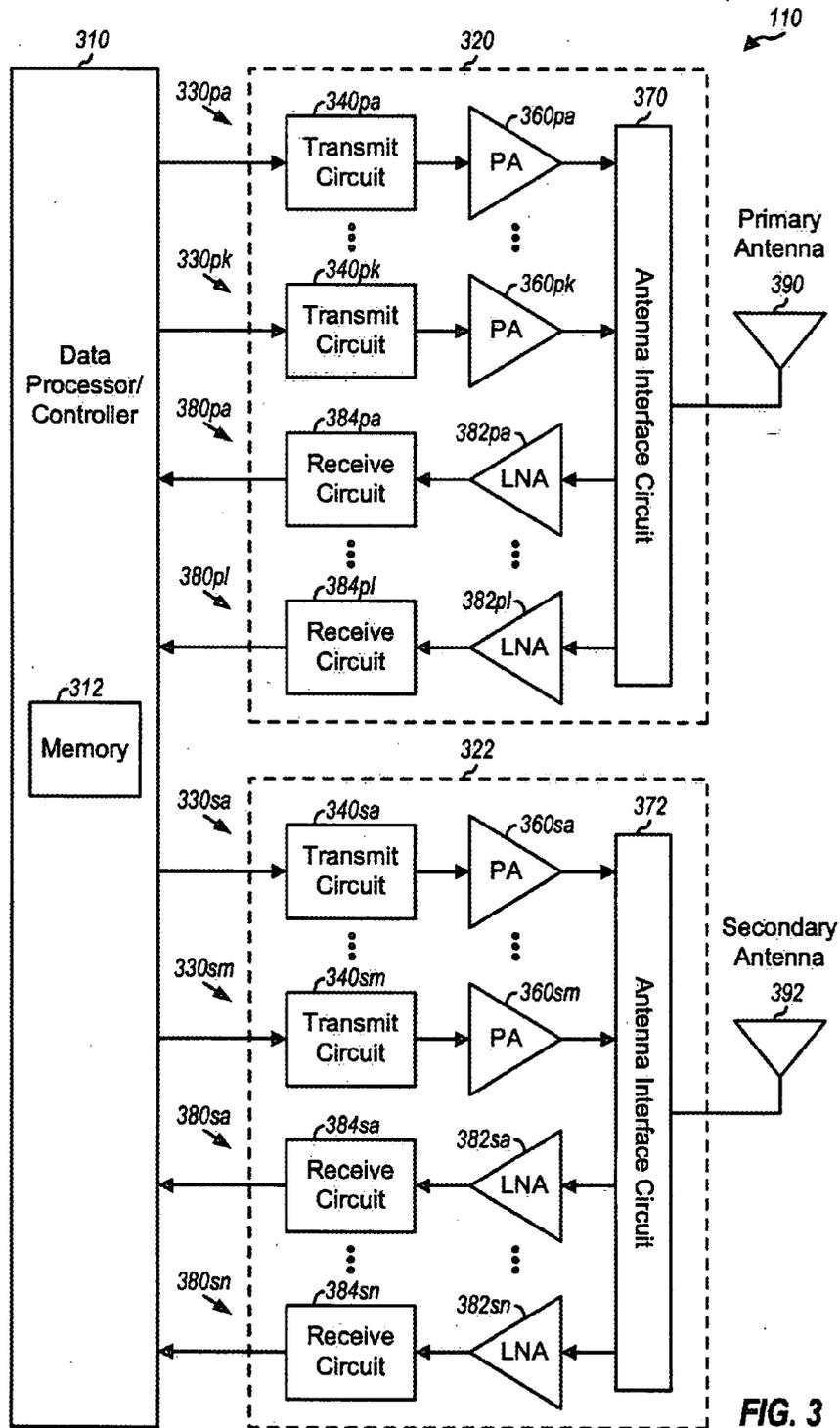


FIG. 3

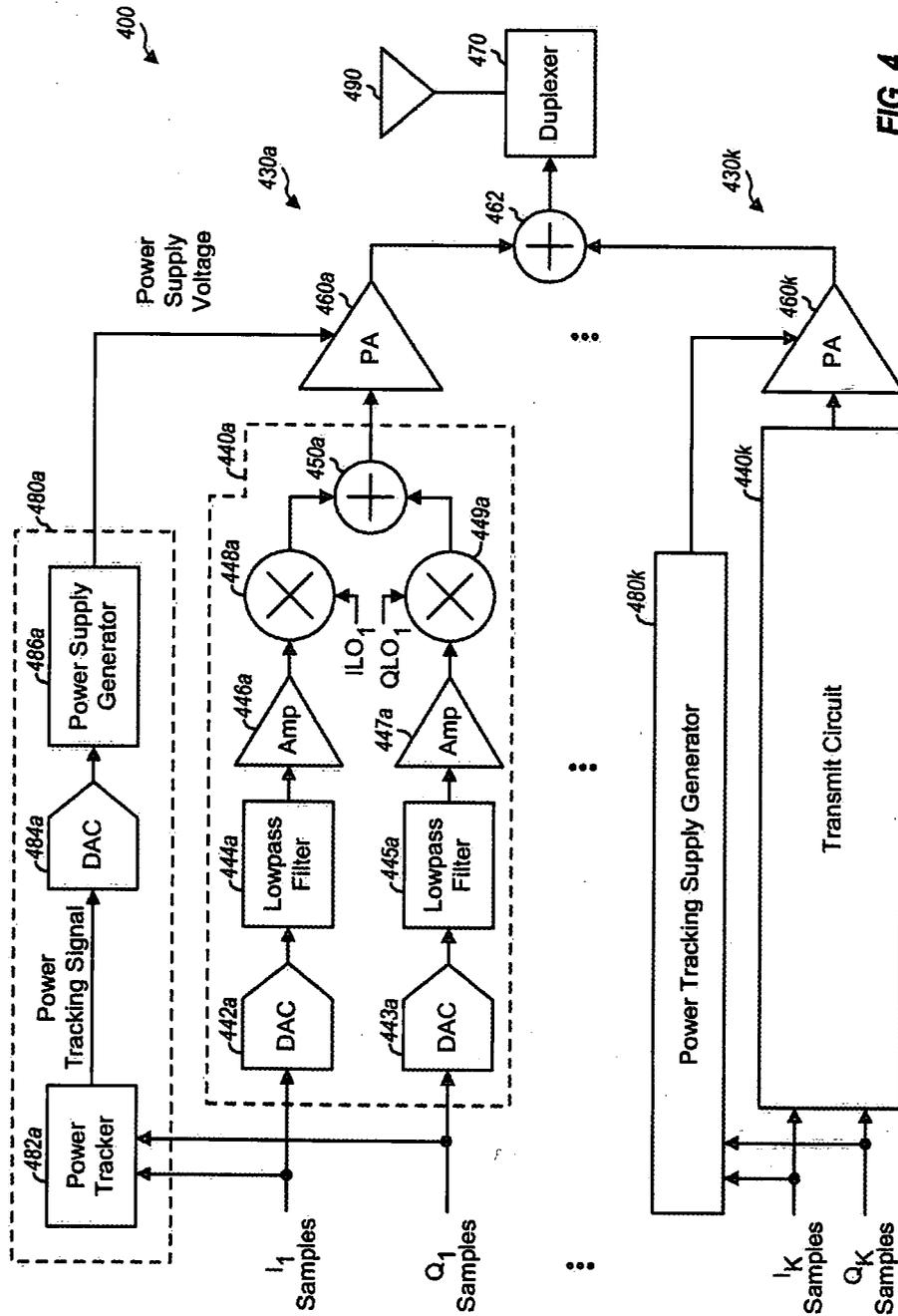


FIG. 4

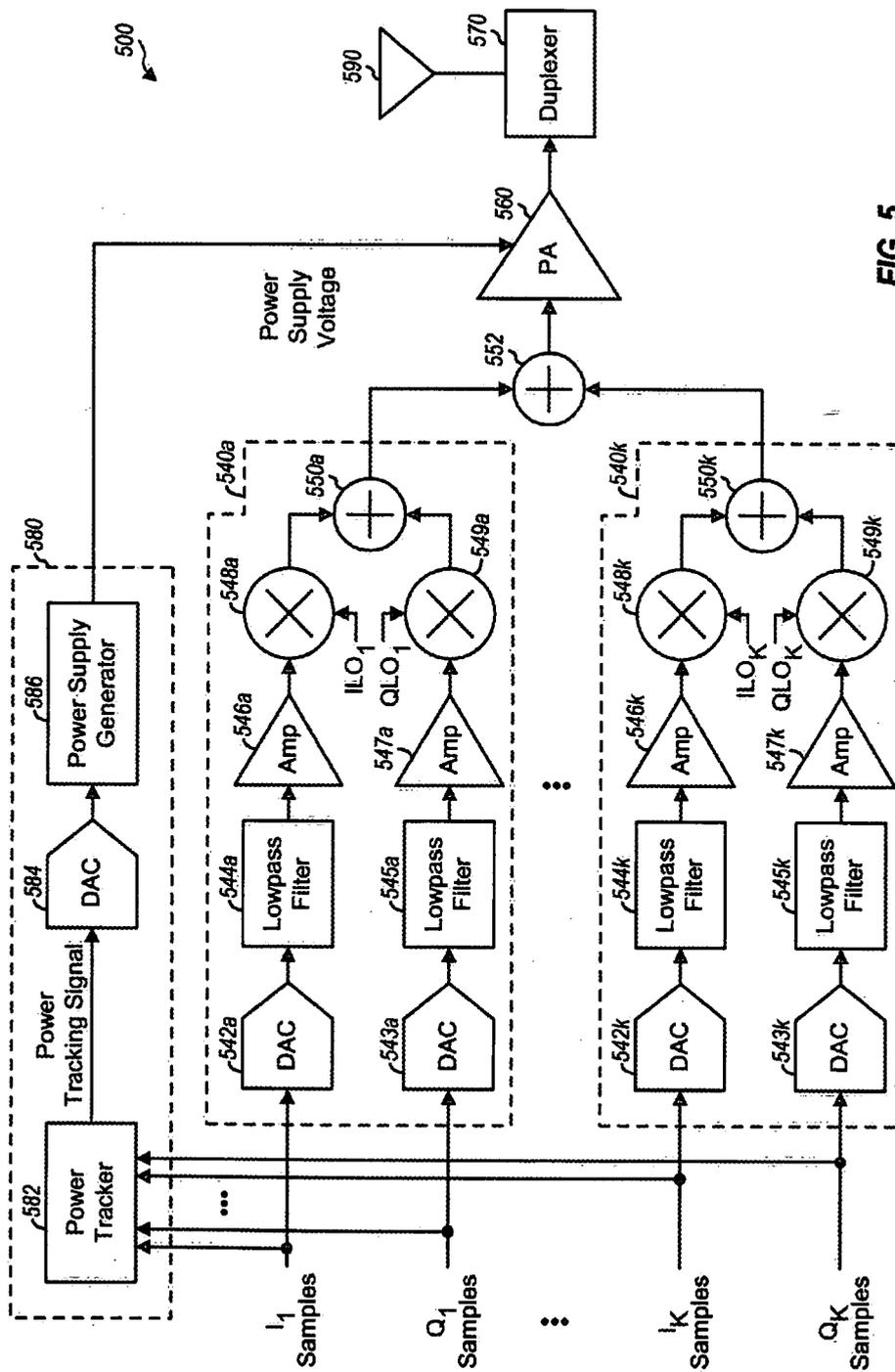


FIG. 5

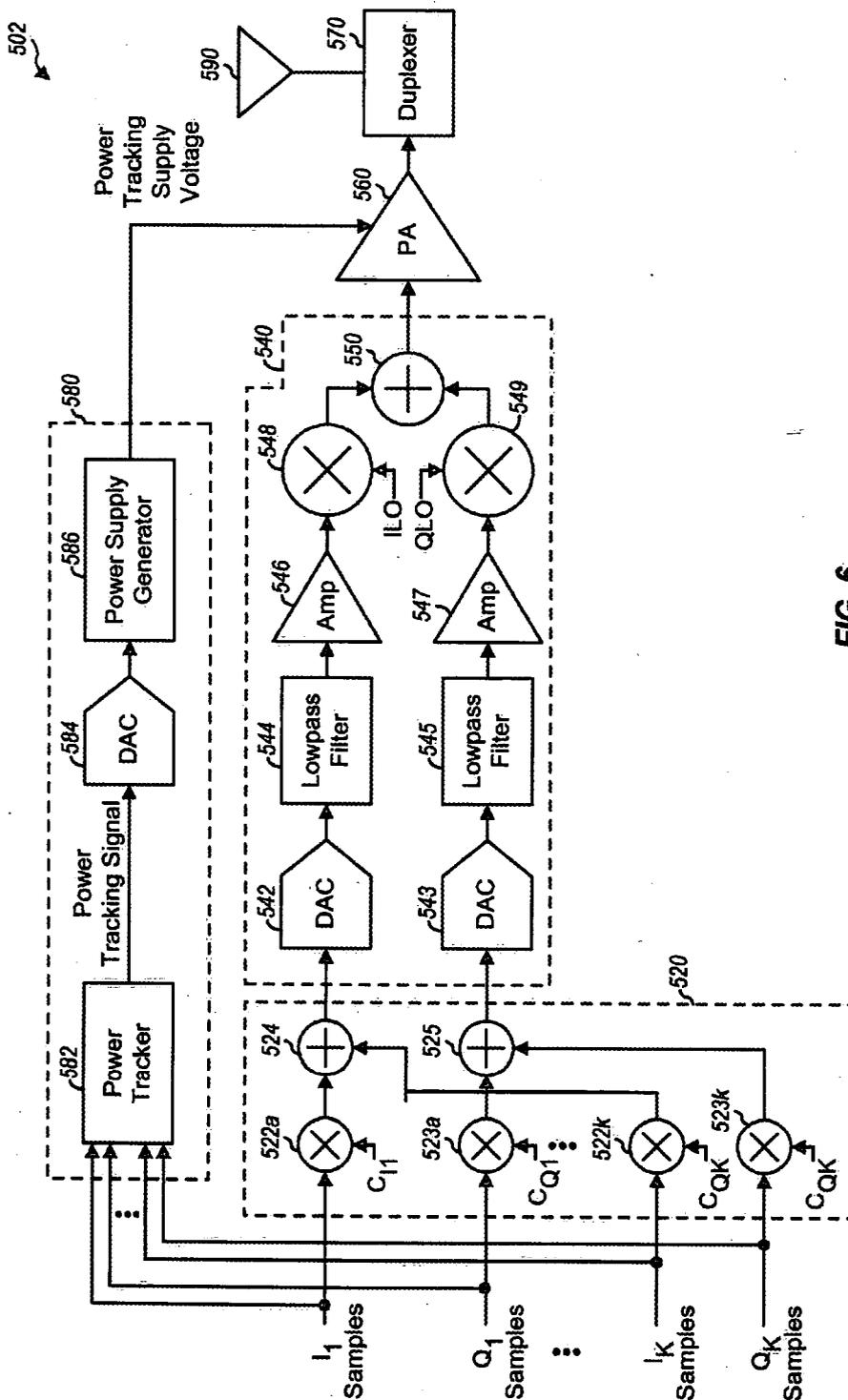
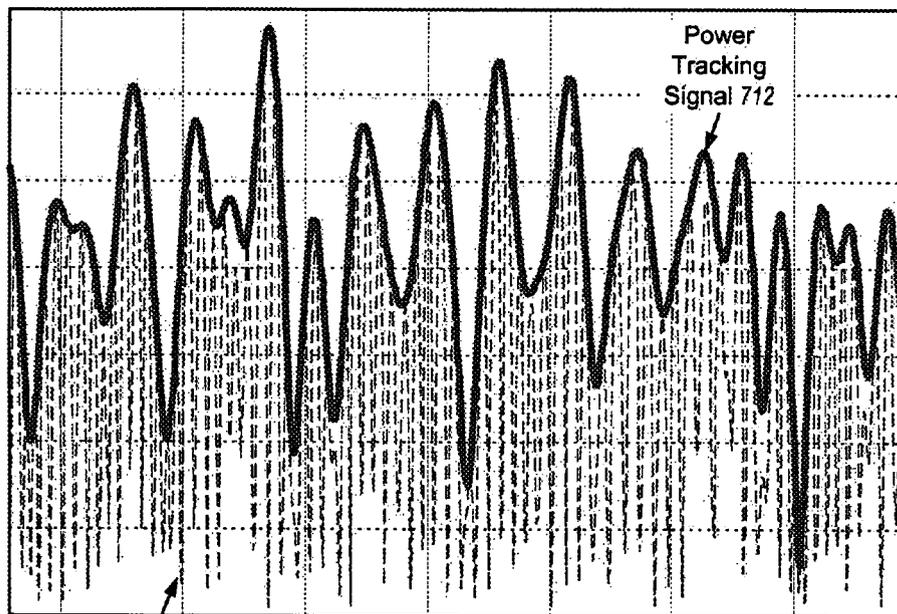
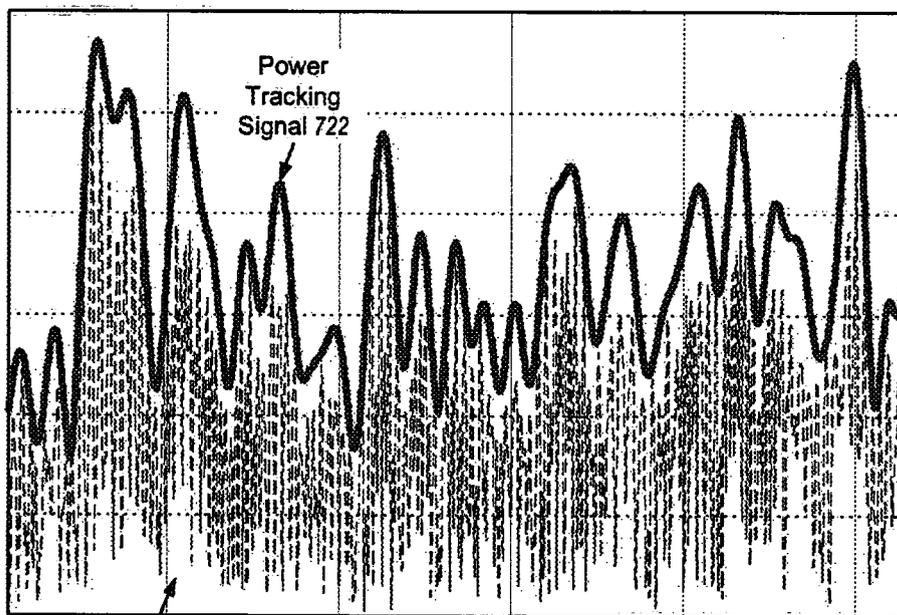


FIG. 6



Output RF Signal 710

FIG. 7A



Output RF Signal 720

FIG. 7B

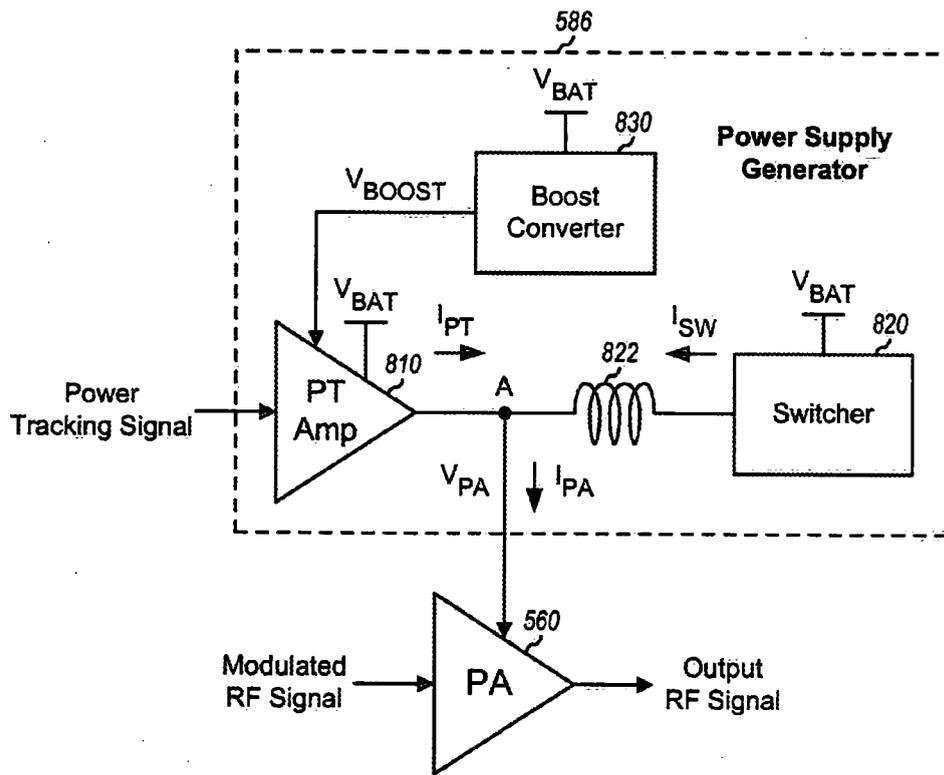


FIG. 8

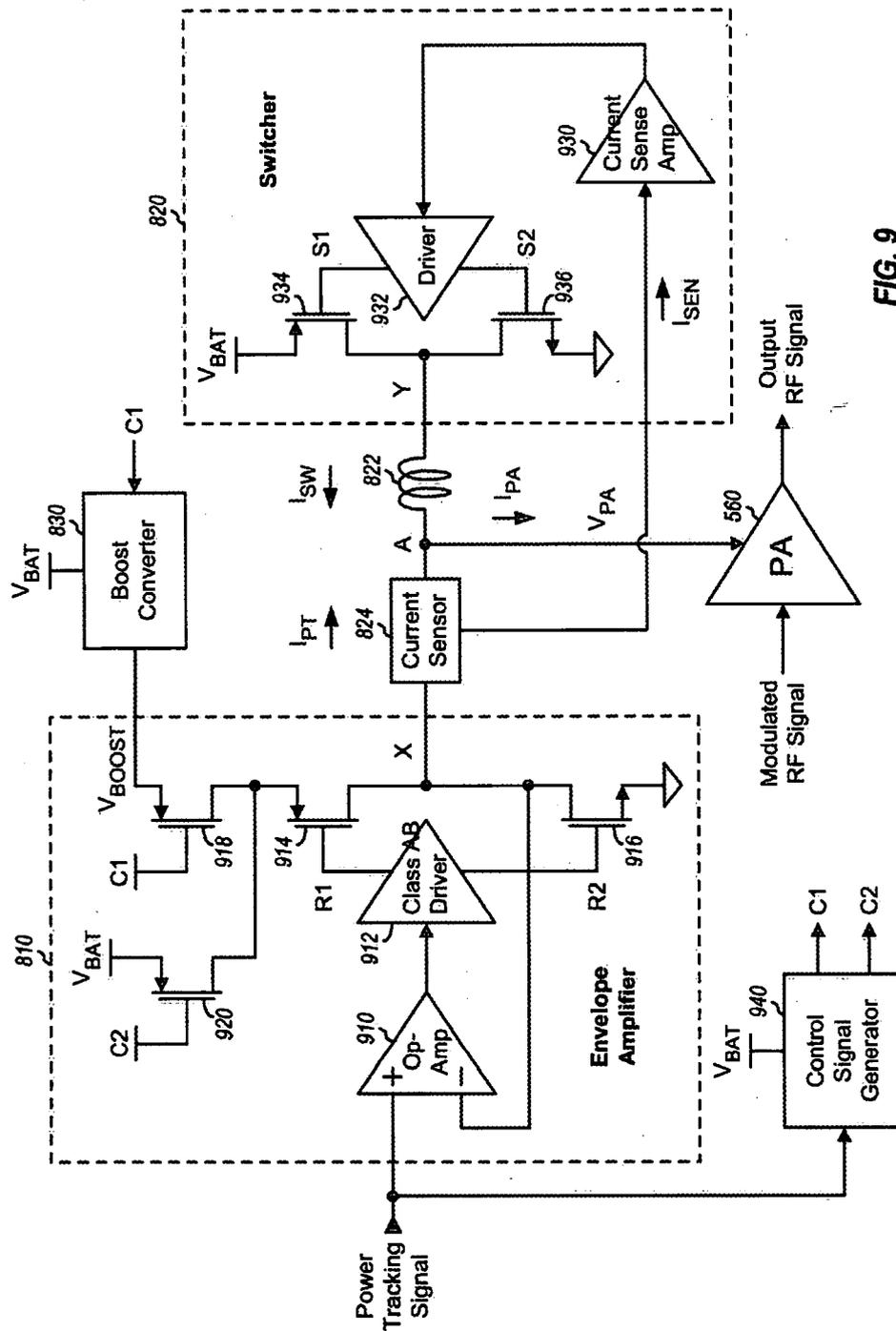


FIG. 9

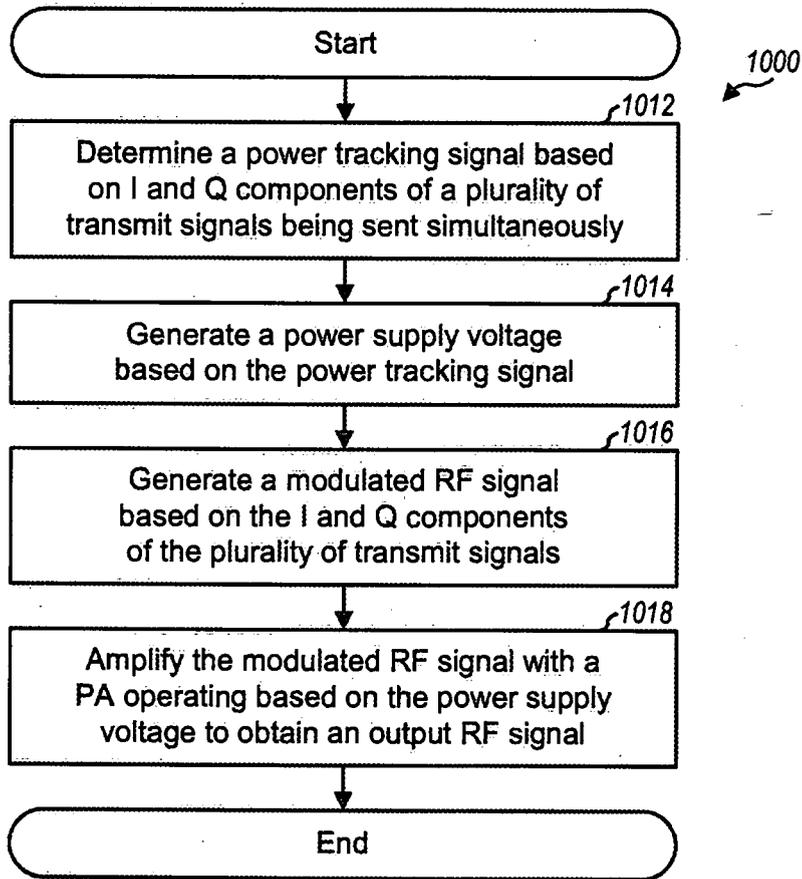


FIG. 10

**POWER TRACKER FOR MULTIPLE
TRANSMIT SIGNALS SENT
SIMULTANEOUSLY**

BACKGROUND

I. Field

The present disclosure relates generally to electronics, and more specifically to techniques for generating a power supply voltage for a circuit such as an amplifier.

II. Background

A wireless device (e.g., a cellular phone or a smartphone) in a wireless communication system may transmit and receive data for two-way communication. The wireless device may include a transmitter for data transmission and a receiver for data reception. For data transmission, the transmitter may process (e.g., encode and modulate) data to generate output samples. The transmitter may further condition (e.g., convert to analog, filter, amplify, and frequency upconvert) the output samples to generate a modulated radio frequency (RF) signal, amplify the modulated RF signal to obtain an output RF signal having the proper transmit power level, and transmit the output RF signal via an antenna to a base station. For data reception, the receiver may obtain a received RF signal via the antenna and may amplify and process the received RF signal to recover data sent by the base station.

The transmitter typically includes a power amplifier (PA) to provide high transmit power for the output RF signal. The power amplifier should be able to provide high transmit power and have high power-added efficiency (PAE).

SUMMARY

Techniques for generating a power tracking supply voltage for a circuit (e.g., a power amplifier) that processes multiple transmit signals sent simultaneously are disclosed herein. The multiple transmit signals may comprise transmissions sent simultaneously on multiple carriers at different frequencies.

In one exemplary design, an apparatus includes a power tracker and a power supply generator. The power tracker determines a power tracking signal based on inphase (I) and quadrature (Q) components of a plurality of transmit signals being sent simultaneously, as described below. The power supply generator generates a power supply voltage based on the power tracking signal. The apparatus may further include a power amplifier that amplifies a modulated RF signal based on the power supply voltage and provides an output RF signal.

Various aspects and features of the disclosure are described in further detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a wireless device communicating with a wireless system.

FIGS. 2A to 2D show four examples of carrier aggregation.

FIG. 3 shows a block diagram of the wireless device in FIG. 1.

FIG. 4 shows a transmit module comprising a separate power amplifier with separate power tracking for each transmit signal.

FIGS. 5 and 6 show two designs of a transmit module comprising a single power amplifier with power tracking for all transmit signals.

FIGS. 7A and 7B show power tracking for two and three transmit signals, respectively.

FIGS. 8 and 9 show a design of a power supply generator with power tracking.

FIG. 10 shows a process for generating a power supply voltage with power tracking.

DETAILED DESCRIPTION

The word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any design described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other designs.

Techniques for generating a power tracking supply voltage for a circuit (e.g., a power amplifier) that processes multiple transmit signals sent simultaneously are disclosed herein. The techniques may be used for various electronic devices such as wireless communication devices.

FIG. 1 shows a wireless device 110 communicating with a wireless communication system 120. Wireless system 120 may be a Long Term Evolution (LTE) system, a Code Division Multiple Access (CDMA) system, a Global System for Mobile Communications (GSM) system, a wireless local area network (WLAN) system, or some other wireless system. A CDMA system may implement Wideband CDMA (WCDMA), CDMA 1X, Time Division Synchronous CDMA (TD-SCDMA), or some other version of CDMA. For simplicity, FIG. 1 shows wireless system 120 including two base stations 130 and 132 and one system controller 140. In general, a wireless system may include any number of base stations and any set of network entities.

Wireless device 110 may also be referred to as a user equipment (UE), a mobile station, a terminal, an access terminal, a subscriber unit, a station, etc. Wireless device 110 may be a cellular phone, a smartphone, a tablet, a wireless modem, a personal digital assistant (PDA), a handheld device, a laptop computer, a smartbook, a netbook, a cordless phone, a wireless local loop (WLL) station, a Bluetooth device, etc. Wireless device 110 may be capable of communicating with wireless system 120. Wireless device 110 may also be capable of receiving signals from broadcast stations (e.g., a broadcast station 134), signals from satellites (e.g., a satellite 150) in one or more global navigation satellite systems (GNSS), etc. Wireless device 110 may support one or more radio technologies for wireless communication such as LTE, WCDMA, CDMA 1X, TD-SCDMA, GSM, 802.11, etc.

Wireless device 110 may be able to operate in low-band (LB) covering frequencies lower than 1000 megahertz (MHz), mid-band (MB) covering frequencies from 1000 MHz to 2300 MHz, and/or high-band (HB) covering frequencies higher than 2300 MHz. For example, low-band may cover 698 to 960 MHz, mid-band may cover 1475 to 2170 MHz, and high-band may cover 2300 to 2690 MHz and 3400 to 3800 MHz. Low-band, mid-band, and high-band refer to three groups of bands (or band groups), with each band group including a number of frequency bands (or simply, "bands"). Each band may cover up to 200 MHz and may include one or more carriers. Each carrier may cover up to 20 MHz in LTE. LTE Release 11 supports 35 bands, which are referred to as LTE/UMTS bands and are listed in 3GPP TS 36.101.

Wireless device 110 may support carrier aggregation, which is operation on multiple carriers. Carrier aggregation may also be referred to as multi-carrier operation. Wireless device 110 may be configured with up to 5 carriers in one or two bands in LTE Release 11.

3

In general, carrier aggregation (CA) may be categorized into two types—*intra-band CA* and *inter-band CA*. *Intra-band CA* refers to operation on multiple carriers within the same band. *Inter-band CA* refers to operation on multiple carriers in different bands.

FIG. 2A shows an example of contiguous *intra-band CA*. In the example shown in FIG. 2A, wireless device 110 is configured with three contiguous carriers in one band in low-band. Wireless device 110 may send and/or receive transmissions on the three contiguous carriers in the same band.

FIG. 2B shows an example of non-contiguous *intra-band CA*. In the example shown in FIG. 2B, wireless device 110 is configured with three non-contiguous carriers in one band in low-band. The carriers may be separated by 5 MHz, 10 MHz, or some other amount. Wireless device 110 may send and/or receive transmissions on the three non-contiguous carriers in the same band.

FIG. 2C shows an example of *inter-band CA* in the same band group. In the example shown in FIG. 2C, wireless device 110 is configured with three carriers in two bands in low-band. Wireless device 110 may send and/or receive transmissions on the three carriers in different bands in the same band group.

FIG. 2D shows an example of *inter-band CA* in different band groups. In the example shown in FIG. 2D, wireless device 110 is configured with three carriers in two bands in different band groups, which include two carriers in one band in low-band and one carrier in another band in mid-band. Wireless device 110 may send and/or receive transmissions on the three carriers in different bands in different band groups.

FIGS. 2A to 2D show four examples of carrier aggregation. Carrier aggregation may also be supported for other combinations of bands and band groups.

FIG. 3 shows a block diagram of an exemplary design of wireless device 110 in FIG. 1. In this exemplary design, wireless device 110 includes a data processor/controller 310, a transceiver 320 coupled to a primary antenna 390, and a transceiver 322 coupled to a secondary antenna 392. Transceiver 320 includes K transmitters 330_{pa} to 330_{pk}, L receivers 380_{pa} to 380_{pl}, and an antenna interface circuit 370 to support multiple bands, carrier aggregation, multiple radio technologies, etc. K and L may each be any integer value of one or greater. Transceiver 322 includes M transmitters 330_{sa} to 330_{sm}, N receivers 380_{sa} to 380_{sn}, and an antenna interface circuit 372 to support multiple bands, carrier aggregation, multiple radio technologies, receive diversity, multiple-input multiple-output (MIMO) transmission, etc. M and N may each be any integer value of one or greater.

In the exemplary design shown in FIG. 3, each transmitter 330 includes a transmit circuit 340 and a power amplifier (PA) 360. For data transmission, data processor 310 processes (e.g., encodes and symbol maps) data to be transmitted to obtain modulation symbols. Data processor 310 further processes the modulation symbols (e.g., for OFDM, SC-FDMA, CDMA, or some other modulation technique) and provides I and Q samples for each transmit signal to be sent by wireless device 110. A transmit signal is a signal comprising a transmission on one or more carriers, a transmission on one or more frequency channels, etc. Data processor 310 provides the I and Q samples for one or more transmit signals to one or more selected transmitters. The description below assumes that transmitter 330_{pa} is a transmitter selected to send one transmit signal. Within transmitter 330_{pa}, transmit circuit 340_{pa} converts I and Q samples

4

to I and Q analog output signals, respectively. Transmit circuit 340_{pa} further amplifies, filters, and upconverts the I and Q analog output signals from baseband to RF and provides a modulated RF signal. Transmit circuit 340_{pa} may include digital-to-analog converters (DACs), amplifiers, filters, mixers, matching circuits, an oscillator, a local oscillator (LO) generator, a phase-locked loop (PLL), etc. A PA 360_{pa} receives and amplifies the modulated RF signal and provides an output RF signal having the proper transmit power level. The output RF signal is routed through antenna interface circuit 370 and transmitted via antenna 390. Antenna interface circuit 370 may include one or more filters, duplexers, diplexers, switches, matching circuits, directional couplers, etc. Each remaining transmitter 330 in transceivers 320 and 322 may operate in similar manner as transmitter 330_{pa}.

In the exemplary design shown in FIG. 3, each receiver 380 includes a low noise amplifier (LNA) 382 and a receive circuit 384. For data reception, antenna 390 receives signals from base stations and/or other transmitter stations and provides a received RF signal, which is routed through antenna interface circuit 370 and provided to a selected receiver. The description below assumes that receiver 380_{pa} is the selected receiver. Within receiver 380_{pa}, an LNA 382_{pa} amplifies the received RF signal and provides an amplified RF signal. A receive circuit 384_{pa} downconverts the amplified RF signal from RF to baseband, amplifies and filters the downconverted signal, and provides an analog input signal to data processor 310. Receive circuit 384_{pa} may include mixers, filters, amplifiers, matching circuits, an oscillator, an LO generator, a PLL, etc. Each remaining receiver 380 in transceivers 320 and 322 may operate in similar manner as receiver 380_{pa}.

FIG. 3 shows an exemplary design of transmitters 330 and receivers 380. A transmitter and a receiver may also include other circuits not shown in FIG. 3, such as filters, matching circuits, etc. All or a portion of transceivers 320 and 322 may be implemented on one or more analog integrated circuits (ICs), RF ICs (RFICs), mixed-signal ICs, etc. For example, transmit circuits 340, LNAs 382, and receive circuits 384 may be implemented on one module, which may be an RFIC, etc. Antenna interface circuits 370 and 372 and PAs 360 may be implemented on another module, which may be a hybrid module, etc. The circuits in transceivers 320 and 322 may also be implemented in other manners.

Data processor/controller 310 may perform various functions for wireless device 110. For example, data processor 310 may perform processing for data being transmitted via transmitters 330 and data being received via receivers 380. Controller 310 may control the operation of transmit circuits 340, PAs 360, LNAs 382, receive circuits 384, antenna interface circuits 370 and 372, or a combination thereof. A memory 312 may store program codes and data for data processor/controller 310. Data processor/controller 310 may be implemented on one or more application specific integrated circuits (ASICs) and/or other ICs.

Wireless device 110 may send multiple transmit signals simultaneously. In one design, the multiple transmit signals may be for transmissions on multiple contiguous or non-contiguous carriers with *intra-band CA*, e.g., as shown in FIG. 2A or 2B. For example, each transmit signal may comprise a transmission sent on one carrier. In another design, the multiple transmit signals may be for transmissions on multiple frequency channels to the same wireless system. In yet another design, the multiple transmit signals may be for transmissions sent to different wireless systems (e.g., LTE and WLAN). In any case, data to be sent in each

5

transmit signal may be processed (e.g., encoded, symbol mapped, and modulated) separately to generate I and Q samples for that transmit signal. Each transmit signal may be conditioned by a respective transmit circuit 340 and amplified by a respective PA 360 to generate an output RF signal for that transmit signal.

A PA may receive a modulated RF signal and a power supply voltage and may generate an output RF signal. The output RF signal typically tracks the modulated RF signal and has a time-varying envelope. The power supply voltage should be higher than the amplitude of the output RF signal at all times in order to avoid clipping the output RF signal, which would then cause intermodulation distortion (IMD) that may degrade performance. The difference between the power supply voltage and the envelope of the output RF signal represents wasted power that is dissipated by the PA instead of delivered to an output load.

It may be desirable to generate a power supply voltage for a PA such that good performance and good efficiency can be obtained. This may be achieved by generating the power supply voltage for the PA with power tracking so that the power supply voltage can track the envelope of an output RF signal from the PA.

FIG. 4 shows a design of a transmit module 400 supporting simultaneous transmission of multiple (K) transmit signals with a separate PA and separate power tracking for each transmit signal. Transmit module 400 includes K transmitters 430a to 430k that can simultaneously process K transmit signals, with each transmitter 430 processing one transmit signal. Each transmitter 430 includes a transmit circuit 440, a PA 460, and a power tracking supply generator 480.

Transmitter 430a receives I_1 and Q_1 samples for a first transmit signal and generates a first output RF signal for the first transmit signal. The I_1 and Q_1 samples are provided to both transmit circuit 440a and voltage generator 480a. Within transmit circuit 440a, the I_1 and Q_1 samples are converted to I and Q analog signals by DACs 442a and 443a, respectively. The I analog signal is filtered by a lowpass filter 444a, amplified by an amplifier (Amp) 446a, and upconverted from baseband to RF by a mixer 448a. Similarly, the Q analog signal is filtered by a lowpass filter 445a, amplified by an amplifier 447a, and upconverted from baseband to RF by a mixer 449a. Mixers 448a and 449a perform upconversion for the first transmit signal based on I and Q LO signals (ILO_1 and QLO_1) at a center RF frequency of the first transmit signal. A summer 450a sums the I and Q upconverted signals from mixers 448a and 449a to obtain a modulated RF signal, which is provided to PA 460a.

Within voltage generator 480a, a power tracker 482a receives the I_1 and Q_1 samples for the first transmit signal, computes the power of the first transmit signal based on the I_1 and Q_1 samples, and provides a digital power tracking signal to a DAC 484a. DAC 484a converts the digital power tracking signal to analog and provides an analog power tracking signal. A power supply generator 486a receives the analog power tracking signal and generates a power supply voltage for PA 460a. PA 460a amplifies the modulated RF signal from transmit circuit 440a using the power supply voltage from supply generator 486a and provides the first output RF signal for the first transmit signal.

Each remaining transmitter 430 may similarly process I and Q samples for a respective transmit signal and may provide an output RF signal for the transmit signal. Up to K PAs 460a to 460k may provide up to K output RF signals at different RF frequencies for up to K transmit signals being

6

sent simultaneously. A summer 462 receives the output RF signals being sent simultaneously, sums the output RF signals, and provides a final output RF signal, which is routed through a duplexer 470 and transmitted via an antenna 490.

As shown in FIG. 4, power tracking may be used to improve the efficiency of PAs 460a to 460k. Each transmit signal may be processed by a respective transmitter 430 using a separate sets of mixers 448 and 449 and PA 460. Multiple transmit signals may be sent on different frequencies (e.g., different carriers) and hence may have increased envelope bandwidth. The increased envelope bandwidth may be addressed by using a separate transmitter 430 for each transmit signal. Each transmitter 430 may then handle the envelope bandwidth of one transmit signal. However, operating multiple transmitters 430 concurrently for multiple transmit signals may result in more circuits, higher power consumption, and increased cost, all of which are undesirable.

In an aspect of the present disclosure, a single PA with power tracking may be used to generate a single output RF signal for multiple transmit signals being sent simultaneously. A single power supply voltage may be generated for the PA to track the power of all transmit signals being sent simultaneously. This may reduce the number of circuit components, reduce power consumption, and provide other advantages.

FIG. 5 shows a design of a transmit module 500 supporting simultaneous transmission of multiple (K) transmit signals with a single PA and power tracking for all transmit signals. Transmit module 500 performs frequency upconversion separately for each transmit signal in the analog domain and sums the resultant upconverted RF signals for all transmit signals. Transmit module 500 includes K transmit circuits 540a to 540k that can simultaneously process K transmit signals, with each transmit circuit 540 processing one transmit signal. Transmit module 500 further includes a summer 552, a PA 560, a duplexer 570, and a power tracking supply generator 580.

Transmit circuit 540a receives I_1 and Q_1 samples for a first transmit signal and generates a first upconverted RF signal for the first transmit signal. The I_1 and Q_1 samples are provided to both transmit circuit 540a and voltage generator 580. Within transmit circuit 540a, the I_1 and Q_1 samples are converted to I and Q analog signals by DACs 542a and 543a, respectively. The I and Q analog signals are filtered by lowpass filters 544a and 545a, amplified by amplifiers 546a and 547a, upconverted from baseband to RF by mixers 548a and 549a, and summed by a summer 550a to generate the first upconverted RF signal. Mixers 548a and 549a perform upconversion for the first transmit signal based on I and Q LO signals at a center RF frequency of the first transmit signal.

Each remaining transmit circuit 540 may similarly process I and Q samples for a respective transmit signal and may provide an upconverted RF signal for the transmit signal. Up to K transmit circuits 540a to 540k may provide up to K upconverted RF signals at different RF frequencies for up to K transmit signals being sent simultaneously. A summer 552 receives the upconverted RF signals from transmit circuits 540a to 540k, sums the upconverted RF signals, and provides a modulated RF signal to PA 560.

Within voltage generator 580, a power tracker 582 receives I_1 to I_K samples and Q_1 to Q_K samples for all transmit signals being sent simultaneously. Power tracker 582 computes the overall power of all transmit signals based on the I and Q samples for these transmit signals and

provides a digital power tracking signal to a DAC 584. DAC 584 converts the digital power tracking signal to analog and provides an analog power tracking signal for all transmit signals. Although not shown in FIG. 5, a lowpass filter may receive and filter an output signal from DAC 584 and provide the analog power tracking signal. A power supply generator 586 receives the analog power tracking signal and generates a power supply voltage for PA 560.

PA 560 amplifies the modulated RF signal from summer 552 using the power supply voltage from supply generator 586. PA 560 provides an output RF signal for all transmit signals being sent simultaneously. The output RF signal is routed through duplexer 570 and transmitted via antenna 590.

FIG. 6 shows a design of a transmit module 502 supporting simultaneous transmission of multiple (K) transmit signals with a single PA and power tracking for all transmit signals. Transmit module 502 digitally upconverts each transmit signal to an intermediate frequency (IF) in the digital domain, sums the resultant upconverted IF signals for all transmit signals, and performs frequency upconversion from IF to RF for all transmit signals together in the analog domain. Transmit module 502 includes a digital modulator 520, a transmit circuit 540, PA 560, duplexer 570, and power tracking supply generator 580.

Digital modulator 520 receives I and Q samples for all transmit signals and generates a modulated IF signal for all transmit signals. Within digital modulator 520, the I_1 and Q_1 samples for the first transmit signal are upconverted to a first IF frequency by multipliers 522a and 523a, respectively, based on C_{I1} and C_{Q1} digital LO signals. The I and Q samples for each remaining transmit signal are upconverted to a different IF frequency by multipliers 522 and 523, respectively, for that transmit signal. The IF frequencies of the K transmit signals may be selected based on the final RF frequencies of the K transmit signals. A summer 524 sums the outputs of all K multipliers 522a to 522k and provides an I modulated signal. Similarly, a summer 525 sums the outputs of all K multipliers 523a to 523k and provides a Q modulated signal. The I and Q modulated signals from summers 524 and 525 form the modulated IF signal for all transmit signals.

Transmit circuit 540 receives I and Q modulated signals from digital modulator 520 and generates a modulated RF signal for all transmit signals. Within transmit circuit 540, the I and Q modulated signals are converted to I and Q analog signals by DACs 542 and 543, respectively. The I and Q analog signals are filtered by lowpass filters 544 and 545, amplified by amplifiers 546 and 547, upconverted from IF to RF by mixers 548 and 549, and summed by a summer 550 to generate the modulated RF signal. Mixers 548 and 549 perform upconversion for the modulated IF signal based on I and Q LO signals at a suitable frequency so that the K transmit signals are upconverted to their proper RF frequencies.

Power tracking voltage generator 580 receives the I_1 to I_K samples and the Q_1 to Q_K samples for all transmit signals being sent simultaneously. Voltage generator 580 generates a power supply voltage for PA 560 based on the I and Q samples. PA 560 amplifies the modulated RF signal from transmit circuit 540 using the power supply voltage from supply generator 580. PA 560 provides an output RF signal for all transmit signals being sent simultaneously. The output RF signal is routed through duplexer 570 and transmitted via antenna 590.

FIGS. 5 and 6 show two exemplary designs of a transmit module supporting simultaneous transmission of multiple

transmit signals with a single PA and power tracking for all transmit signals. Multiple transmit signals may also be sent with a single PA and power tracking in other manners. For example, polar modulation may be used instead of quadrature modulation, which is shown in FIGS. 5 and 6.

Power tracker 582 may compute the digital power tracking signal based on the I and Q samples for all transmit signals in various manners. In one design, the digital power tracking signal may be computed as follows:

$$p(t) = \sqrt{K} \sqrt{I_1^2(t) + Q_1^2(t) + \dots + I_K^2(t) + Q_K^2(t)}, \quad \text{Eq (1)}$$

where $I_k(t)$ and $Q_k(t)$ denote the I and Q samples for the k-th transmit signal in sample period t, for $k=1, \dots, K$, and $p(t)$ denotes the digital power tracking signal in sample period t.

The quantity $I_k^2(t) + Q_k^2(t)$ denotes the power of the k-th transmit signal in sample period t. In the design shown in equation (1), the powers of all transmit signals are summed to obtain an overall power. The digital power tracking signal is then obtained by taking the square root of the overall power. The scaling factor of \sqrt{K} accounts for conversion between power and voltage.

In another design, the digital power tracking signal may be computed as follows:

$$p(t) = \sqrt{I_1^2(t) + Q_1^2(t) + \dots + \sqrt{I_k^2(t) + Q_k^2(t)}}. \quad \text{Eq (2)}$$

The quantity $\sqrt{I_k^2(t) + Q_k^2(t)}$ denotes the voltage of the k-th transmit signal in sample period t. In the design shown in equation (2), the voltage of each transmit signal is first computed, and the voltages of all transmit signals are then summed to obtain the digital power tracking signal.

Equations (1) and (2) are two exemplary designs of computing the digital power tracking signal based on the I and Q samples for all transmit signals being sent simultaneously. The digital power tracking signal computed in equation (1) or (2) has a bandwidth that approximates the bandwidth of the widest transmit signal (instead of the overall bandwidth of all transmit signals being sent simultaneously). Having the bandwidth of the power tracking signal being smaller than a modulation bandwidth may allow for a more efficient power tracking circuitry and may also result in less noise being injected into PA 560 via the power supply.

The digital power tracking signal may also be computed based on the I and Q samples of the transmit signals in other manners, e.g., based on other equations or functions. In one design, the digital power tracking signal may be generated based on the I and Q samples for all transmit signals, without any filtering, e.g., as shown in equation (1) or (2). In another design, the digital power tracking signal may be filtered, e.g., with a lowpass filter having similar characteristics as lowpass filters 544 and 545 in transmit circuit 540.

In one design, the digital power tracking signal may be computed in the same manner (e.g., based on the same equation) regardless of the number of transmit signals being sent simultaneously. In another design, the digital power tracking signal may be computed in different manners (e.g., based on different equations) depending on the number of transmit signals being sent simultaneously. The digital power tracking signal may also be computed in different manners depending on other factors such as the transmit power levels of different transmit signals.

The techniques described herein for generating a power tracking supply voltage for multiple transmit signals may be used for various modulation techniques. For example, the techniques may be used to generate a power tracking supply

voltage for multiple transmit signals sent simultaneously using orthogonal frequency division multiplexing (OFDM), SC-FDMA, CDMA, or some other modulation techniques. The techniques may also be used to generate a tracking power supply voltage for any number of transmit signals being sent simultaneously.

FIG. 7A shows an example of power tracking for two transmit signals sent on two non-contiguous carriers with SC-FDMA, e.g., for non-contiguous intra-band CA shown in FIG. 2B. The two transmit signals are sent on two carriers separated by a 25 MHz gap, with each carrier having a bandwidth of 10 MHz. A plot 710 shows an output RF signal comprising the two transmit signals and provided by PA 560 in FIG. 5 or 6. A plot 712 shows a power tracking signal provided by power tracker 582 in FIG. 5 or 6. The power tracking signal is computed based on I and Q samples for the two transmit signals in accordance with equation (1). As shown in FIG. 7A, the power tracking signal closely follows the envelope of the output RF signal. Hence, good performance and high efficiency may be achieved for PA 560.

FIG. 7B shows an example of power tracking for three transmit signals sent on three non-contiguous carriers with OFDM, e.g., for non-contiguous intra-band CA. The three transmit signals are sent on three carriers, with each carrier having a bandwidth of 5 MHz and being separated by a 15 MHz gap to another carrier. A plot 720 shows an output RF signal comprising the three transmit signals and provided by PA 560 in FIG. 5 or 6. A plot 722 shows a power tracking signal provided by power tracker 582 in FIG. 5 or 6. The power tracking signal is computed based on I and Q samples for the three transmit signals in accordance with equation (1). As shown in FIG. 7B, the power tracking signal follows the envelope of the output RF signal. Hence, good performance and high efficiency may be achieved for PA 560.

It can be shown that a power tracking supply voltage may also be generated for multiple transmit signals sent on multiple carriers with CDMA. In general, the power tracking supply voltage can closely follow the envelope of the output RF signal when two transmit signals are sent simultaneously, e.g., as shown in FIG. 7A. The power tracking supply voltage can approximate the envelope of the output RF signal when more than two transmit signals are sent simultaneously, e.g., as shown in FIG. 7B.

Power supply generator 586 may generate a power supply voltage for PA 560 based on a power tracking signal in various manners. Power supply generator 586 should generate the power supply voltage in an efficient manner in order to conserve battery power of wireless device 110.

FIG. 8 shows a design of power supply generator 586 in FIGS. 5 and 6. In this design, power supply generator 586 includes a power tracking amplifier (PT Amp) 810, a switcher 820, a boost converter 830, and an inductor 822. Switcher 820 may also be referred to as a switching-mode power supply (SMPS). Switcher 820 receives a battery voltage (V_{BAT}) and provides a first supply current (I_{SW}) comprising DC and low frequency components at node A. Inductor 822 stores current from switcher 820 and provides the stored current to node A on alternating cycles. Boost converter 830 receives the V_{BAT} voltage and generates a boosted supply voltage (V_{BOOST}) that is higher than the V_{BAT} voltage. Power tracking amplifier 810 receives the analog power tracking signal at its signal input, receives the V_{BAT} voltage and the V_{BOOST} voltage at its two power supply inputs, and provides a second supply current (I_{PT}) comprising high frequency components at node A. The PA supply current (I_{PA}) provided to power amplifier 560 includes the I_{SW} current from switcher 820 and the I_{PT} current from

power tracking amplifier 810. Power tracking amplifier 810 also provides the proper PA supply voltage (V_{PA}) at Node A for PA 560. The various circuits in power supply generator 586 are described in further detail below.

FIG. 9 shows a schematic diagram of a design of power tracking amplifier 810 and switcher 820 within power supply generator 586 in FIG. 8. Within power tracking amplifier 810, an operational amplifier (op-amp) 910 has its non-inverting input receiving the power tracking signal, its inverting input coupled to an output of power tracking amplifier 810 (which is node X), and its output coupled to an input of a class AB driver 912. Driver 912 has its first output (R1) coupled to the gate of a P-channel metal oxide semiconductor (PMOS) transistor 914 and its second output (R2) coupled to the gate of an N-channel metal oxide semiconductor (NMOS) transistor 916. NMOS transistor 916 has its drain coupled to node X and its source coupled to circuit ground. PMOS transistor 914 has its drain coupled to node X and its source coupled to the drains of PMOS transistors 918 and 920. PMOS transistor 918 has its gate receiving a C1 control signal and its source receiving the V_{BOOST} voltage. PMOS transistor 920 has its gate receiving a C2 control signal and its source receiving the V_{BAT} voltage.

A current sensor 824 is coupled between node X and node A and senses the I_{PT} current provided by power tracking amplifier 810. Sensor 824 passes most of the I_{PT} current to node A and provides a small fraction of the I_{PT} current as a sensed current (I_{SEN}) to switcher 820.

Within switcher 820, a current sense amplifier 930 has its input coupled to current sensor 824 and its output coupled to an input of a switcher driver 932. Driver 932 has its first output (S1) coupled to the gate of a PMOS transistor 934 and its second output (S2) coupled to the gate of an NMOS transistor 936. NMOS transistor 936 has its drain coupled to an output of switcher 820 (which is node Y) and its source coupled to circuit ground. PMOS transistor 934 has its drain coupled to node Y and its source receiving the V_{BAT} voltage. Inductor 822 is coupled between node A and node Y.

Switcher 820 operates as follows. Switcher 820 is in an ON state when current sensor 824 senses a high output current from power tracking amplifier 810 and provides a low sensed voltage to driver 932. Driver 932 then provides a low voltage to the gate of PMOS transistor 934 and a low voltage to the gate of NMOS transistor 936. PMOS transistor 934 is turned ON and couples the V_{BAT} voltage to inductor 822, which stores energy from the V_{BAT} voltage. The current through inductor 822 rises during the ON state, with the rate of the rise being dependent on (i) the difference between the V_{BAT} voltage and the V_{PA} voltage at node A and (ii) the inductance of inductor 822. Conversely, switcher 820 is in an OFF state when current sensor 824 senses a low output current from power tracking amplifier 810 and provides a high sensed voltage to driver 932. Driver 932 then provides a high voltage to the gate of PMOS transistor 934 and a high voltage to the gate of NMOS transistor 936. NMOS transistor 936 is turned ON, and inductor 822 is coupled between node A and circuit ground. The current through inductor 822 falls during the OFF state, with the rate of the fall being dependent on the V_{PA} voltage at node A and the inductance of inductor 822. The V_{BAT} voltage thus provides current to PA 560 via inductor 822 during the ON state, and inductor 120 provides its stored energy to PA 560 during the OFF state.

Power tracking amplifier 810 operates as follows. When the power tracking signal increases, the output of op-amp 910 increases, the R1 output of driver 912 decreases and the R2 output of driver 912 decreases until NMOS transistor

916 is almost turned OFF, and the output of power tracking amplifier 810 increases. The converse is true when the power tracking signal decreases. The negative feedback from the output of power tracking amplifier 810 to the inverting input of op-amp 910 results in power tracking amplifier 810 having unity gain. Hence, the output of power tracking amplifier 810 follows the power tracking signal, and the V_{PA} voltage is approximately equal to the power tracking signal. Driver 912 may be implemented with a class AB amplifier to improve efficiency, so that large output currents can be supplied even though the bias current in transistors 914 and 916 is low.

In one design, power tracking amplifier 810 operates based on the V_{BOOST} voltage only when needed and based on the V_{BAT} voltage during the remaining time in order to improve efficiency. For example, power tracking amplifier 810 may provide approximately 85% of the power based on the V_{BAT} voltage and only approximately 15% of the power based on the V_{BOOST} voltage. When a high V_{PA} voltage is needed for PA 560 due to a large envelope of the output RF signal, the C1 control signal is at logic low, and the C2 control signal is at logic high. In this case, boost converter 830 is enabled and generates the V_{BOOST} voltage, PMOS transistor 918 is turned ON and provides the V_{BOOST} voltage to the source of PMOS transistor 914, and PMOS transistor 920 is turned OFF. Conversely, when a high V_{PA} voltage is not needed for PA 560, the C1 control signal is at logic high, and the C2 control signal is at logic low. In this case, boost converter 830 is disabled, PMOS transistor 918 is turned OFF, and PMOS transistor 920 is turned ON and provides the V_{BAT} voltage to the source of PMOS transistor 914.

A control signal generator 940 receives the power tracking signal and the V_{BAT} voltage and generates the C1 and C2 control signals. The C1 control signal is complementary to the C2 control signal. In one design, generator 940 generates the C1 and C2 control signals to select the V_{BOOST} voltage for power tracking amplifier 910 when the magnitude of the power tracking signal exceeds a first threshold. The first threshold may be a fixed threshold or may be determined based on the V_{BAT} voltage. In another design, generator 940 generates the C1 and C2 control signals to select the V_{BOOST} voltage for power tracking amplifier 910 when the magnitude of the power tracking signal exceeds the first threshold and the V_{BAT} voltage is below a second threshold. Generator 940 may also generate the C1 and C2 signals based on other signals, other voltages, and/or other criteria.

Switcher 820 has high efficiency and delivers a majority of the supply current for PA 560. Power tracking amplifier 810 operates as a linear stage and has relatively high bandwidth (e.g., in the MHz range). Switcher 820 operates to reduce the output current from power tracking amplifier 810, which improves overall efficiency.

FIG. 9 shows an exemplary design of switcher 820 and power tracking amplifier 810 in FIG. 1. Switcher 820 and power tracking amplifier 810 may also be implemented in other manners. For example, power tracking amplifier 810 may be implemented as described in U.S. Pat. No. 6,300,826, entitled "Apparatus and Method for Efficiently Amplifying Wideband Envelope Signals," issued Oct. 9, 2001.

In an exemplary design, an apparatus (e.g., an integrated circuit, a wireless device, a circuit module, etc.) may comprise a power tracker and a power supply generator. The power tracker (e.g., power tracker 582 in FIG. 5) may determine a power tracking signal based on I and Q components (e.g., I and Q samples) of a plurality of transmit signals being sent simultaneously. The power supply gen-

erator (e.g., power supply generator 586 in FIG. 5) may generate a power supply voltage based on the power tracking signal.

In one design, the power tracker may determine an overall power of the plurality of transmit signals based on the I and Q components of the plurality of transmit signals, e.g., as $I_1^2(t)+Q_1^2(t)+\dots+I_K^2(t)+Q_K^2(t)$. The power tracker may then determine the power tracking signal based on the overall power of the plurality of transmit signals, e.g., as shown in equation (1). In another design, the power tracker may determine the power of each transmit signal based on the I and Q components of that transmit signal, e.g., as $I_k^2(t)+Q_k^2(t)$ for the k-th transmit signal. The power tracker may then determine the power tracking signal based on the powers of the plurality of transmit signals, e.g., as shown in equation (2). The power tracker may determine a voltage of each transmit signal based on the power of the transmit signal, e.g., as $\sqrt{I_k^2(t)+Q_k^2(t)}$. The power tracker may then determine the power tracking signal based on voltages of the plurality of transmit signals, e.g., as shown in equation (2). The power tracker may also determine the power tracking signal based on the I and Q components of the plurality of transmit signals in other manners. In one design, the plurality of transmit signals may be sent on a plurality of carriers at different frequencies. The power tracking signal may have a bandwidth that is smaller than an overall bandwidth of the plurality of carriers.

In one design, the apparatus may comprise a plurality of transmit circuits and a summer, e.g., as shown in FIG. 5. The plurality of transmit circuits (e.g., transmit circuits 540a to 540k) may receive the I and Q components of the plurality of transmit signals and may provide a plurality of upconverted RF signals. Each transmit circuit may upconvert I and Q components of one transmit signal and provide a corresponding upconverted RF signal. The summer (e.g., summer 552) may sum the plurality of upconverted RF signals and provide a modulated RF signal. In another design, the apparatus may comprise a transmit circuit (e.g., transmit circuit 540 in FIG. 6) that may receive a modulated IF signal for the plurality of transmit signals and provide a modulated RF signal. The modulated IF signal may be generated (e.g., by digital modulator 520 in FIG. 6) based on the I and Q components of the plurality of transmit signals. In an exemplary design, the apparatus may further comprise a PA (e.g., PA 560 in FIGS. 5 and 6) that may amplify the modulated RF signal based on the power supply voltage and provide an output RF signal.

In an exemplary design, the power supply generator may comprise a power tracking amplifier (e.g., power tracking amplifier 810 in FIGS. 8 and 9) that may receive the power tracking signal and generate the power supply voltage. The power supply generator may further comprise a switcher and/or a boost converter. The switcher (e.g., switcher 820 in FIGS. 8 and 9) may sense a first current (e.g., the I_{PT} current) from the power tracking amplifier and provide a second current (e.g., the I_{SP} current) for the power supply voltage based on the sensed first current. The boost converter (e.g., boost converter 830 in FIGS. 8 and 9) may receive a battery voltage and provide a boosted voltage for the power tracking amplifier. The power tracking amplifier may operate based on the boosted voltage or the battery voltage.

FIG. 10 shows a design of a process 1000 for generating a power supply voltage with power tracking. A power tracking signal may be determined based on I and Q components of a plurality of transmit signals being sent simultaneously (block 1012). In one design of block 1012, an

overall power of the plurality of transmit signals may be determined based on the I and Q components of the plurality of transmit signals. The power tracking signal may then be determined based on the overall power of the plurality of transmit signals, e.g., as shown in equation (1). In another design of block 1012, the power of each transmit signal may be determined based on the I and Q components of the transmit signal. The power tracking signal may then be determined based on the powers of the plurality of transmit signals, e.g., as shown in equation (2).

A power supply voltage may be generated based on the power tracking signal (block 1014). In one design, the power supply voltage may be generated with an amplifier (e.g., amplifier 810 in FIG. 9) that tracks the power tracking signal. The power supply voltage may also be generated based on a switcher and/or a boost converter.

A modulated RF signal may be generated based on the I and Q components of the plurality of transmit signals (block 1016). In one design, I and Q components of each transmit signal may be upconverted to obtain a corresponding upconverted RF signal. A plurality of upconverted RF signals for the plurality of transmit signals may then be summed to obtain the modulated RF signal, e.g., as shown in FIG. 5. In another design, a modulated IF signal may be generated based on the I and Q components of the plurality of transmit signals, e.g., as shown in FIG. 6. The modulated IF signal may then be upconverted to obtain the modulated RF signal. In any case, the modulated RF signal may be amplified with a PA (e.g., PA 560 in FIGS. 5 and 6) operating based on the power supply voltage to obtain an output RF signal (block 1018).

The power tracker and power supply generator described herein may be implemented on an IC, an analog IC, an RFIC, a mixed-signal IC, an ASIC, a printed circuit board (PCB), an electronic device, etc. The power tracker and power supply generator may also be fabricated with various IC process technologies such as complementary metal oxide semiconductor (CMOS), NMOS, PMOS, bipolar junction transistor (BJT), bipolar-CMOS (BiCMOS), silicon germanium (SiGe), gallium arsenide (GaAs), etc.

An apparatus implementing the power tracker and/or power supply generator described herein may be a stand-alone device or may be part of a larger device. A device may be (i) a stand-alone IC, (ii) a set of one or more ICs that may include memory ICs for storing data and/or instructions, (iii) an RFIC such as an RF receiver (RFR) or an RF transmitter/receiver (RTR), (iv) an ASIC such as a mobile station modem (MSM), (v) a module that may be embedded within other devices, (vi) a receiver, cellular phone, wireless device, handset, or mobile unit, (vii) etc.

In one or more exemplary designs, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over as one or more instructions or code on a computer-readable medium. Computer-readable media includes both computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A storage media may be any available media that can be accessed by a computer. By way of example, and not limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to carry or store desired program code in the form of instructions or data structures and that can be accessed by a computer. Also, any connection is properly termed a com-

puter-readable medium. For example, if the software is transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and blu-ray disc where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

The previous description of the disclosure is provided to enable any person skilled in the art to make or use the disclosure. Various modifications to the disclosure will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other variations without departing from the scope of the disclosure. Thus, the disclosure is not intended to be limited to the examples and designs described herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

What is claimed is:

1. An apparatus comprising:

a power tracker configured to determine a single power tracking signal based on a plurality of inphase (I) and quadrature (Q) components of a plurality of carrier aggregated transmit signals being sent simultaneously, wherein the power tracker receives the plurality of I and Q components corresponding to the plurality of carrier aggregated transmit signals and generates the single power tracking signal based on a combination of the plurality of I and Q components, wherein the plurality of carrier aggregated transmit signals comprise Orthogonal Frequency Division Multiplexing (OFDM) or Single Carrier Frequency Division Multiple Access (SC-FDMA) signals;

a power supply generator configured to generate a single power supply voltage based on the single power tracking signal; and

a power amplifier configured to receive the single power supply voltage and the plurality of carrier aggregated transmit signals being sent simultaneously to produce a single output radio frequency (RF) signal.

2. The apparatus of claim 1, wherein the power tracker is configured to:

determine an overall power of the plurality of carrier aggregated transmit signals based on the I and Q components of the plurality of carrier aggregated transmit signals, and

determine the single power tracking signal based on the overall power of the plurality of carrier aggregated transmit signals.

3. The apparatus of claim 1, wherein the power tracker is configured to:

determine a power of each transmit signal in the plurality of carrier aggregated transmit signals based on the I and Q components of each transmit signal, and

determine the single power tracking signal based on a sum of said power of each transmit signal of the plurality of carrier aggregated transmit signals.

4. The apparatus of claim 1, wherein the power tracker is configured to:

determine a power of each transmit signal in the plurality of carrier aggregated transmit signals based on the I and Q components of each transmit signal, determine a voltage of each transmit signal based on the power of each transmit signal, and determine the single power tracking signal based on said voltage of each transmit signal of the plurality of carrier aggregated transmit signals.

5. The apparatus of claim 1, further comprising:
 a plurality of transmit circuits configured to receive the I and Q components of the plurality of carrier aggregated transmit signals and provide a plurality of upconverted RF signals, each transmit circuit configured to upconvert I and Q components of one of the plurality of carrier aggregated transmit signals and provide a corresponding upconverted RF signal, and
 a summer configured to sum the plurality of upconverted RF signals and provide the plurality of carrier aggregated transmit signals to the power amplifier.

6. The apparatus of claim 1, further comprising:
 a transmit circuit configured to receive a modulated intermediate frequency (IF) signal and provide the plurality of carrier aggregated transmit signals to the power amplifier, the modulated IF signal being generated based on the I and Q components of the plurality of carrier aggregated transmit signals.

7. The apparatus of claim 1, the power supply generator comprising:
 a power tracking amplifier configured to receive the power tracking signal and generate the power supply voltage.

8. The apparatus of claim 7, the power supply generator further comprising:
 a switcher configured to sense a first current from the power tracking amplifier and provide a second current for the power supply voltage based on the sensed first current.

9. The apparatus of claim 7, the power supply generator further comprising:
 a boost converter configured to receive a battery voltage and provide a boosted voltage for the power tracking amplifier.

10. The apparatus of claim 9, wherein the power tracking amplifier operates based on the boosted voltage or the battery voltage.

11. The apparatus of claim 1, wherein the plurality of carrier aggregated transmit signals are sent on a plurality of carriers at different frequencies.

12. The apparatus of claim 11, wherein the single power tracking signal has a bandwidth that is smaller than an overall bandwidth of the plurality of carriers.

13. The apparatus of claim 1, wherein the carrier aggregated transmit signals are intra-band carrier aggregated transmit signals.

14. The apparatus of claim 13, wherein the intra-band carrier aggregated transmit signals are contiguous.

15. The apparatus of claim 13, wherein the intra-band carrier aggregated transmit signals are non-contiguous.

16. The apparatus of claim 1, wherein the power tracker is configured to determine the single power tracking signal based on functions comprising:
 squaring each of the plurality of inphase (I) and quadrature (Q) components to produce a plurality of I^2 and Q^2 values;
 summing the plurality of I^2 and Q^2 values to produce an overall power; and
 taking the square root of the overall power.

17. The apparatus of claim 1, wherein the power tracker is configured to determine the single power tracking signal based on functions comprising:
 calculating $\sqrt{I_k^2(t)+Q_k^2(t)}$ corresponding to K inphase (I) and quadrature (Q) components to produce K voltages; and
 summing the K voltages.

18. A method comprising:
 determining a single power tracking signal based on a plurality of inphase (I) and quadrature (Q) components of a plurality of carrier aggregated transmit signals being sent simultaneously, wherein a power tracker receives the plurality of I and Q components corresponding to the plurality of carrier aggregated transmit signals and generates the single power tracking signal based on a combination of the plurality of I and Q components, wherein the plurality of carrier aggregated transmit signals comprise Orthogonal Frequency Division Multiplexing (OFDM) or Single Carrier Frequency Division Multiple Access (SC-FDMA) signals;
 generating a single power supply voltage based on the single power tracking signal; and
 receiving the single power supply voltage and the plurality of carrier aggregated transmit signals being sent simultaneously in a power amplifier and producing a single output radio frequency (RF) signal.

19. The method of claim 18, wherein the determining the single power tracking signal comprises:
 determining an overall power of the plurality of carrier aggregated transmit signals based on the I and Q components of the plurality of carrier aggregated transmit signals, and
 determining the single power tracking signal based on the overall power of the plurality of carrier aggregated transmit signals.

20. The method of claim 18, wherein the determining the single power tracking signal comprises:
 determining a power of each transmit signal in the plurality of carrier aggregated transmit signals based on the I and Q components of each transmit signal, and
 determining the single power tracking signal based on a sum of said power of each transmit signal of the plurality of carrier aggregated transmit signals.

21. The method of claim 18, further comprising:
 receiving the I and Q components of the plurality of carrier aggregated transmit signals in a plurality of transmit circuits and providing a plurality of upconverted RF signals from the plurality of transmit circuits, each transmit circuit upconverting I and Q components of one of the plurality of carrier aggregated transmit signals and providing a corresponding upconverted RF signal, and
 summing the plurality of upconverted RF signals and providing the plurality of carrier aggregated transmit signals to the power amplifier.

22. The method of claim 18, further comprising:
 receiving a modulated intermediate frequency (IF) signal in a transmit circuit and providing the plurality of carrier aggregated transmit signals to the power amplifier from the transmit circuit, the modulated IF signal being generated based on the I and Q components of the plurality of carrier aggregated transmit signals.

23. The method of claim 18, wherein the carrier aggregated transmit signals are intra-band carrier aggregated transmit signals.

17

24. The method of claim 23, wherein the intra-band carrier aggregated transmit signals are contiguous.

25. The method of claim 23 wherein the intra-band carrier aggregated transmit signals are non-contiguous.

26. The method of claim 18, wherein determining the single power tracking signal comprises:

squaring each of the plurality of inphase (I) and quadrature (Q) components to produce a plurality of I^2 and Q^2 values;

summing the plurality of I^2 and Q^2 values to produce an overall power; and

taking the square root of the overall power.

27. The method of claim 18, wherein determining the single power tracking signal comprises:

calculating $\sqrt{I_i^2(t)+Q_i^2(t)}$ corresponding to K inphase (I) and quadrature (Q) components to produce K voltages; and

summing the voltages.

28. An apparatus comprising:

means for determining a single power tracking signal based on a plurality of inphase (I) and quadrature (Q) components of a plurality of carrier aggregated transmit signals being sent simultaneously, wherein a power tracker receives the plurality of I and Q components corresponding to the plurality of carrier aggregated transmit signals and generates the single power tracking signal based on a combination of the plurality of I and Q components, wherein the plurality of carrier aggregated transmit signals comprise Orthogonal Frequency Division Multiplexing (OFDM) or Single Carrier Frequency Division Multiple Access (SC-FDMA) signals;

means for generating a single power supply voltage based on the single power tracking signal; and

means for receiving the single power supply voltage and the plurality of carrier aggregated transmit signals being sent simultaneously and producing a single output radio frequency (RF) signal.

29. The apparatus of claim 28, wherein the means for determining the single power tracking signal comprises:

means for determining an overall power of the plurality of carrier aggregated transmit signals based on the I and Q components of the plurality of carrier aggregated transmit signals, and

means for determining the single power tracking signal based on the overall power of the plurality of carrier aggregated transmit signals.

18

30. The apparatus of claim 28, wherein the means for determining the single power tracking signal comprises:

means for determining a power of each transmit signal in the plurality of carrier aggregated transmit signals based on the I and Q components of each transmit signal, and

means for determining the single power tracking signal based on a sum of said power of each transmit signal of the plurality of carrier aggregated transmit signals.

31. The apparatus of claim 28, further comprising:

means for receiving the I and Q components of the plurality of carrier aggregated transmit signals and separately upconverting the I and Q components of the plurality of carrier aggregated transmit signals to provide a plurality of upconverted RF signals, and

means for summing the plurality of upconverted RF signals and providing the plurality of carrier aggregated transmit signals to a power amplifier.

32. The apparatus of claim 28, further comprising:

means for receiving a modulated intermediate frequency (IF) signal and providing the plurality of carrier aggregated transmit signals to a power amplifier, the modulated IF signal being generated based on the I and Q components of the plurality of carrier aggregated transmit signals.

33. A non-transitory computer-readable medium comprising instructions, that when executed by a processor, cause the processor to:

determine a single power tracking signal based on a plurality of inphase (I) and quadrature (Q) components of a plurality of carrier aggregated transmit signals being sent simultaneously, wherein a power tracker receives the plurality of I and Q components corresponding to the plurality of carrier aggregated transmit signals and generates the single power tracking signal based on a combination of the plurality of I and Q components, wherein the plurality of carrier aggregated transmit signals comprise Orthogonal Frequency Division Multiplexing (OFDM) or Single Carrier Frequency Division Multiple Access (SC-FDMA) signals; generate a single power supply voltage based on the single power tracking signal; and

receive the single power supply voltage and the plurality of carrier aggregated transmit signals being sent simultaneously in a power amplifier to produce a single output radio frequency (RF) signal.

* * * * *

