US009270599B1

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

(10) **Patent No.: US 9,270,599 B1**
(45) **Date of Patent: Feb. 23, 2016**

(54) **DYNAMIC COMMUNICATION LINK SCALING**

(71) Applicant: **Cisco Technology, Inc.**, San Jose, CA (US)

(72) Inventors: **Srividhya Nagarajan**, San Jose, CA (US); **Lalit Kumar**, Fremont, CA (US); **Amit Singh**, Fremont, CA (US); **David Walker**, San Jose, CA (US); **Deepak Mayya**, Fremont, CA (US)

(73) Assignee: **Cisco Technology, Inc.**, San Jose, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 300 days.

(21) Appl. No.: **13/860,961**

(22) Filed: **Apr. 11, 2013**

(51) **Int. Cl.**
H04L 12/28 (2006.01)
H04L 12/803 (2013.01)

(52) **U.S. Cl.**
CPC ***H04L 47/122*** (2013.01)

(58) **Field of Classification Search**
CPC H04L 47/10; H04L 47/35; H04L 47/30
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

8,201,006 B2 6/2012 Bobrek et al.
8,817,817 B2 * 8/2014 Koenen H04L 12/10 370/468
2005/0105545 A1 * 5/2005 Thousand H04L 12/10 370/442
2012/0066531 A1 * 3/2012 Shafai H04W 52/0206 713/323
2012/0213223 A1 * 8/2012 Ortacdag H04L 49/201 370/390
2013/0077623 A1 * 3/2013 Han H04L 47/25 370/389

OTHER PUBLICATIONS

Cisco | Intel., "IEEE 802.3az Energy Efficient Ethernet: Build Greener Networks," White Paper, Oct. 2011, pp. 1-9.

* cited by examiner

Primary Examiner — Andrew Lai
Assistant Examiner — Zhiren Qin
(74) *Attorney, Agent, or Firm* — Edell, Shapiro & Finnan, LLC

(57) **ABSTRACT**

Traffic between a first device and a second device across a plurality of communication lanes of a communication link is stalled. A first device side of the communication lane is powered-down on at least one of the plurality of communication lanes to deactivate the at least one communication lane. A power-down request is sent from the first device to the second device to power-down a second device-side of the at least one communication lane. Traffic between the first device and the second device is restarted over the active communication lanes.

27 Claims, 7 Drawing Sheets

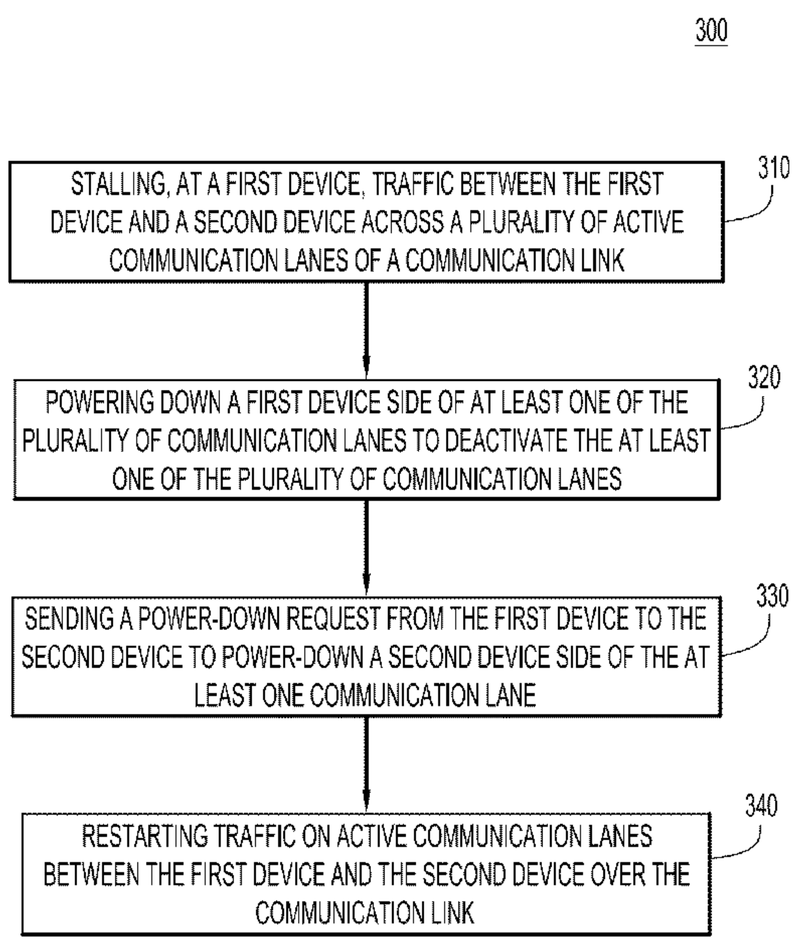
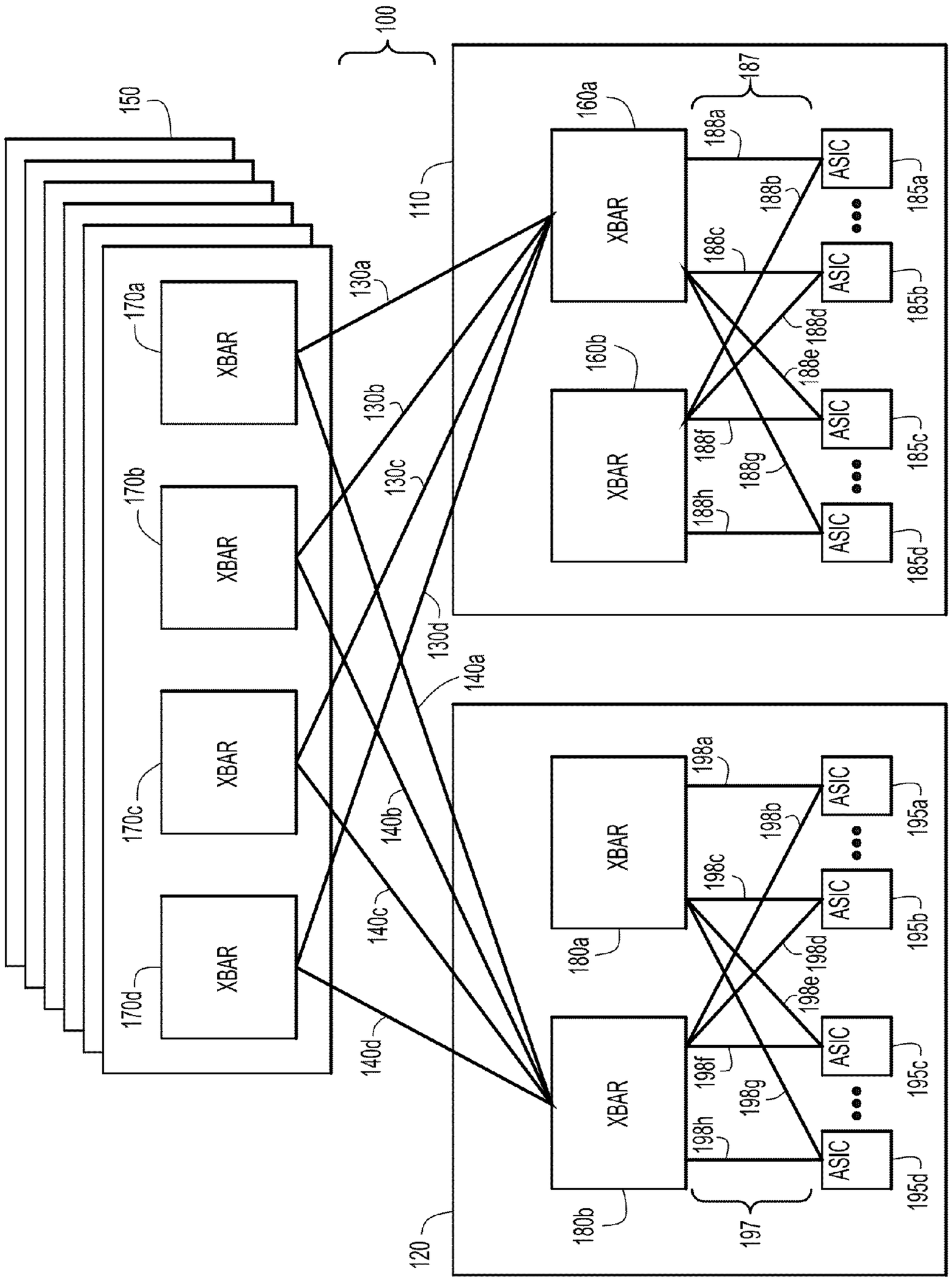


FIG.1


100
150
110
120
130a
130b
130c
130d
140a
140b
140c
140d
160a
160b
170a
170b
170c
170d
180a
180b
185a
185b
185c
185d
187
188a
188b
188c
188d
188e
188f
188g
188h
195a
195b
195c
195d
197
198a
198b
198c
198d
198e
198f
198g
198h
XBAR
ASIC

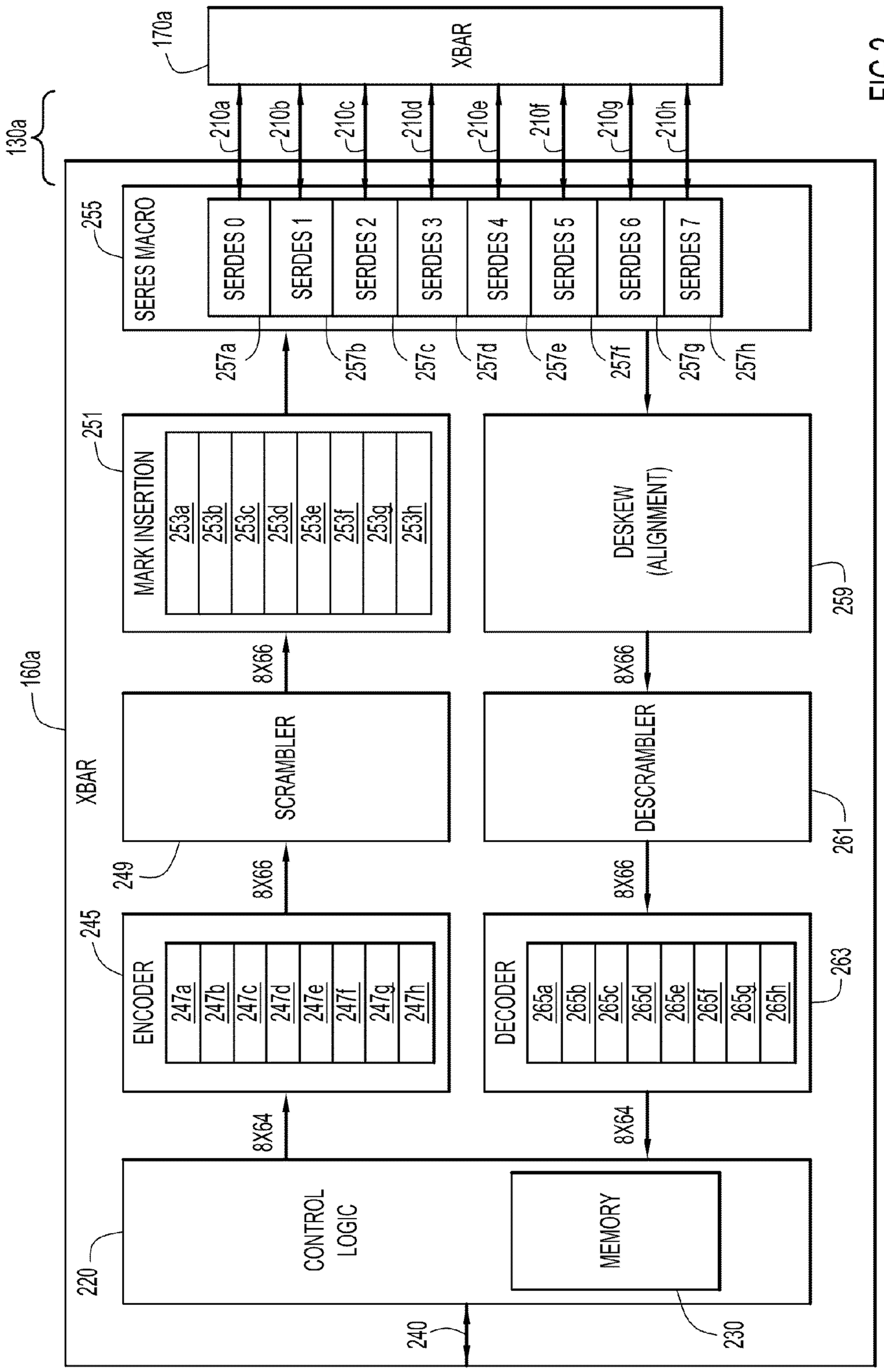

170a
XBAR
130a
210a
210b
210c
210d
210e
210f
210g
210h
255
SERES MACRO
SERDES 0
SERDES 1
SERDES 2
SERDES 3
SERDES 4
SERDES 5
SERDES 6
SERDES 7
257a
257b
257c
257d
257e
257f
257g
257h
251
MARK INSERTION
253a
253b
253c
253d
253e
253f
253g
253h
DESKEW
(ALIGNMENT)
259
160a
XBAR
8X66
SCRAMBLER
249
8X66
DESCRAMBLER
261
245
ENCODER
247a
247b
247c
247d
247e
247f
247g
247h
DECODER
263
265a
265b
265c
265d
265e
265f
265g
265h
8X64
8X64
220
CONTROL
LOGIC
MEMORY
240
230

FIG.2

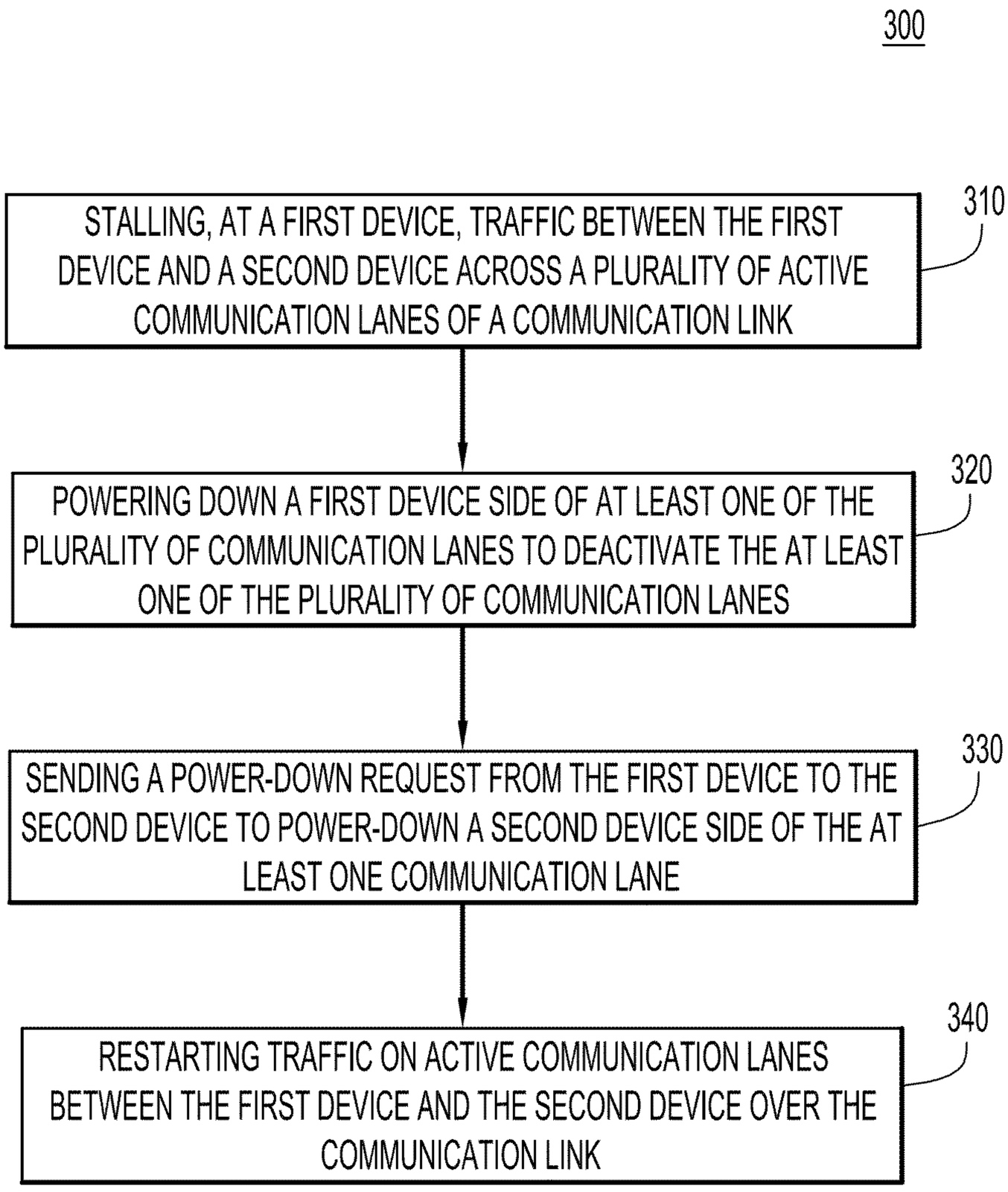

300
STALLING, AT A FIRST DEVICE, TRAFFIC BETWEEN THE FIRST DEVICE AND A SECOND DEVICE ACROSS A PLURALITY OF ACTIVE COMMUNICATION LANES OF A COMMUNICATION LINK
310
POWERING DOWN A FIRST DEVICE SIDE OF AT LEAST ONE OF THE PLURALITY OF COMMUNICATION LANES TO DEACTIVATE THE AT LEAST ONE OF THE PLURALITY OF COMMUNICATION LANES
320
SENDING A POWER-DOWN REQUEST FROM THE FIRST DEVICE TO THE SECOND DEVICE TO POWER-DOWN A SECOND DEVICE SIDE OF THE AT LEAST ONE COMMUNICATION LANE
330
RESTARTING TRAFFIC ON ACTIVE COMMUNICATION LANES BETWEEN THE FIRST DEVICE AND THE SECOND DEVICE OVER THE COMMUNICATION LINK
340

FIG.3

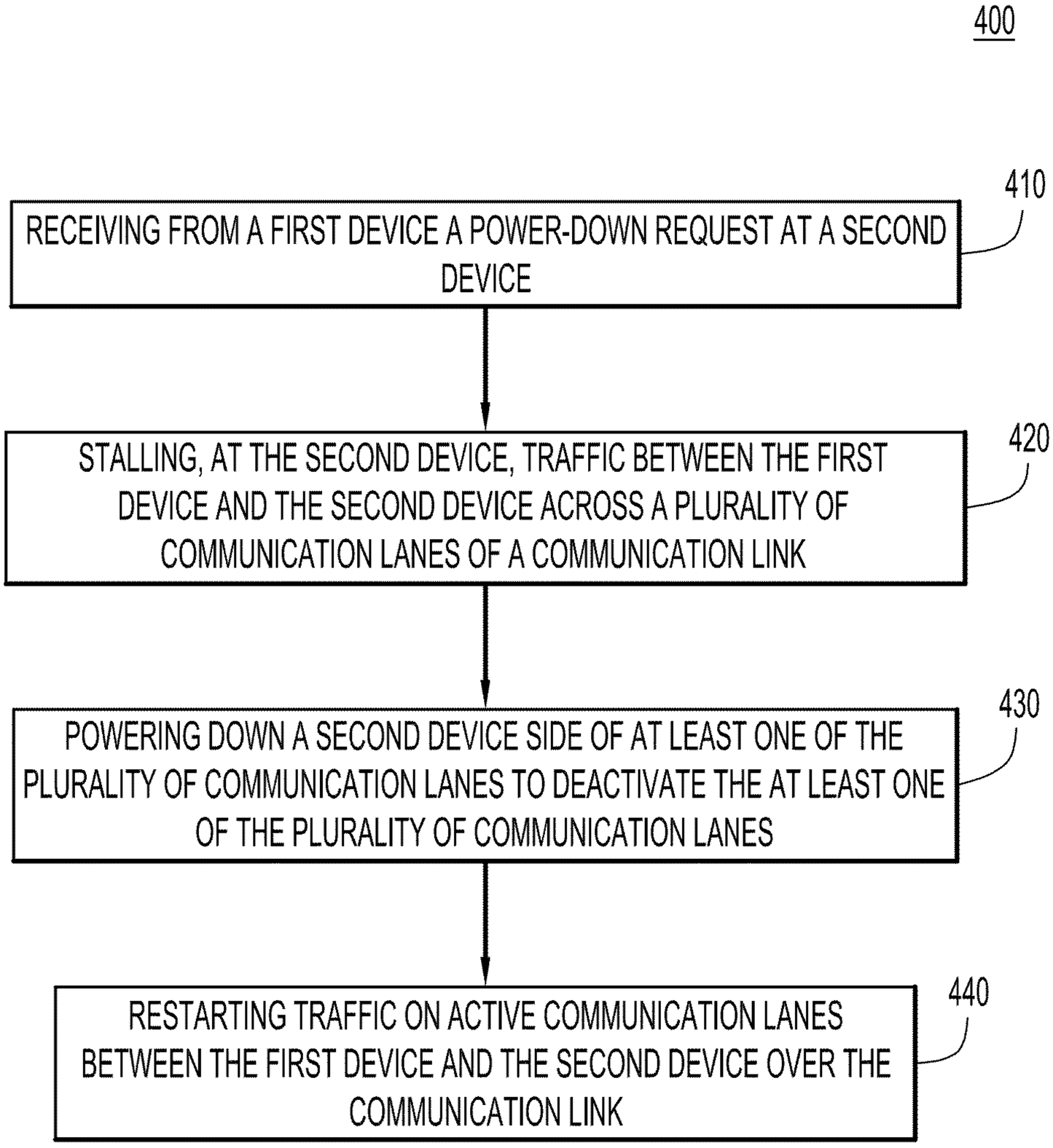

400
RECEIVING FROM A FIRST DEVICE A POWER-DOWN REQUEST AT A SECOND DEVICE
410
STALLING, AT THE SECOND DEVICE, TRAFFIC BETWEEN THE FIRST DEVICE AND THE SECOND DEVICE ACROSS A PLURALITY OF COMMUNICATION LANES OF A COMMUNICATION LINK
420
POWERING DOWN A SECOND DEVICE SIDE OF AT LEAST ONE OF THE PLURALITY OF COMMUNICATION LANES TO DEACTIVATE THE AT LEAST ONE OF THE PLURALITY OF COMMUNICATION LANES
430
RESTARTING TRAFFIC ON ACTIVE COMMUNICATION LANES BETWEEN THE FIRST DEVICE AND THE SECOND DEVICE OVER THE COMMUNICATION LINK
440

FIG.4

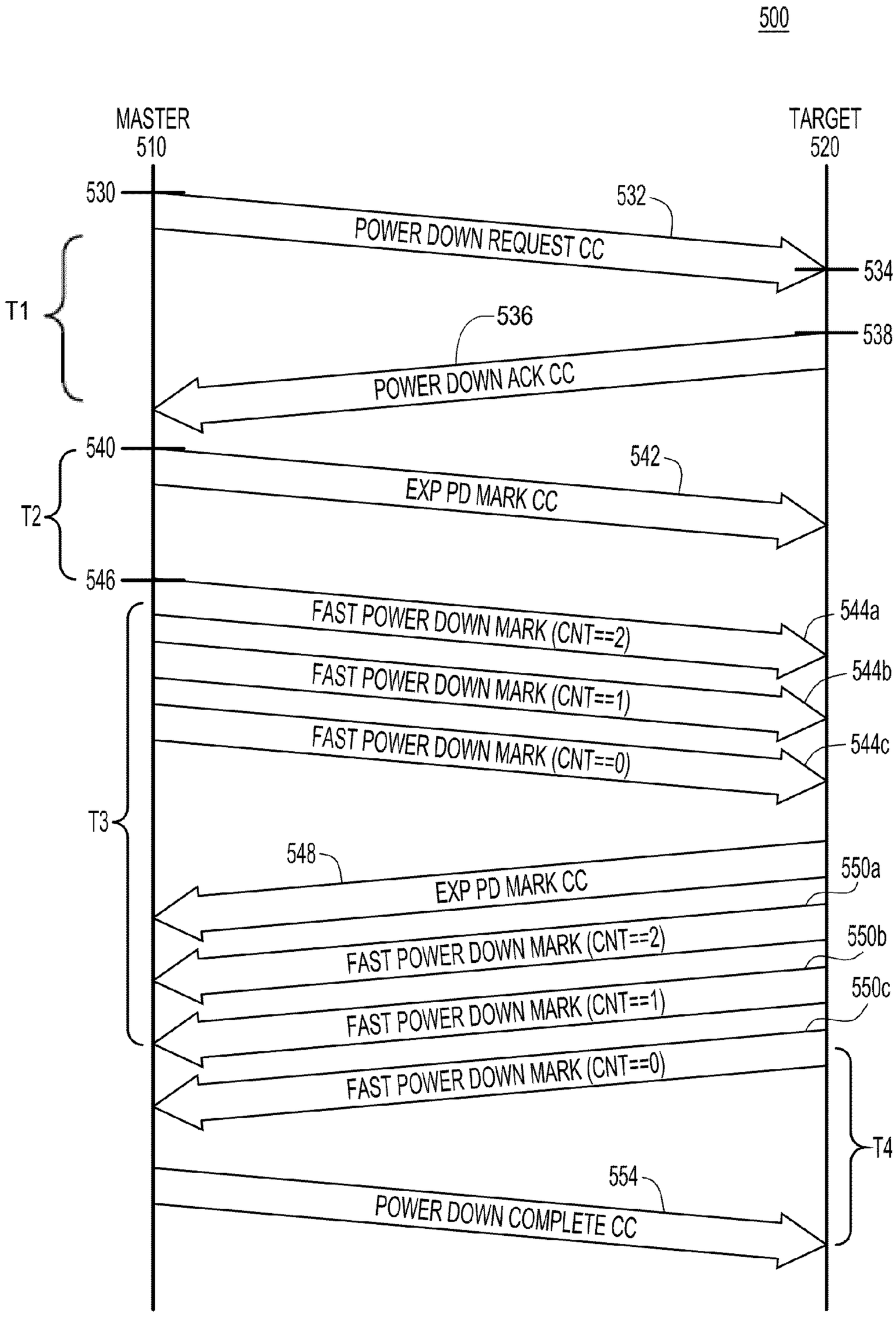

500
MASTER
510
TARGET
520
530
532
POWER DOWN REQUEST CC
534
T1
536
538
POWER DOWN ACK CC
540
542
EXP PD MARK CC
T2
546
FAST POWER DOWN MARK (CNT==2)
544a
FAST POWER DOWN MARK (CNT==1)
544b
FAST POWER DOWN MARK (CNT==0)
544c
T3
548
EXP PD MARK CC
550a
FAST POWER DOWN MARK (CNT==2)
550b
FAST POWER DOWN MARK (CNT==1)
550c
FAST POWER DOWN MARK (CNT==0)
T4
554
POWER DOWN COMPLETE CC

FIG.5

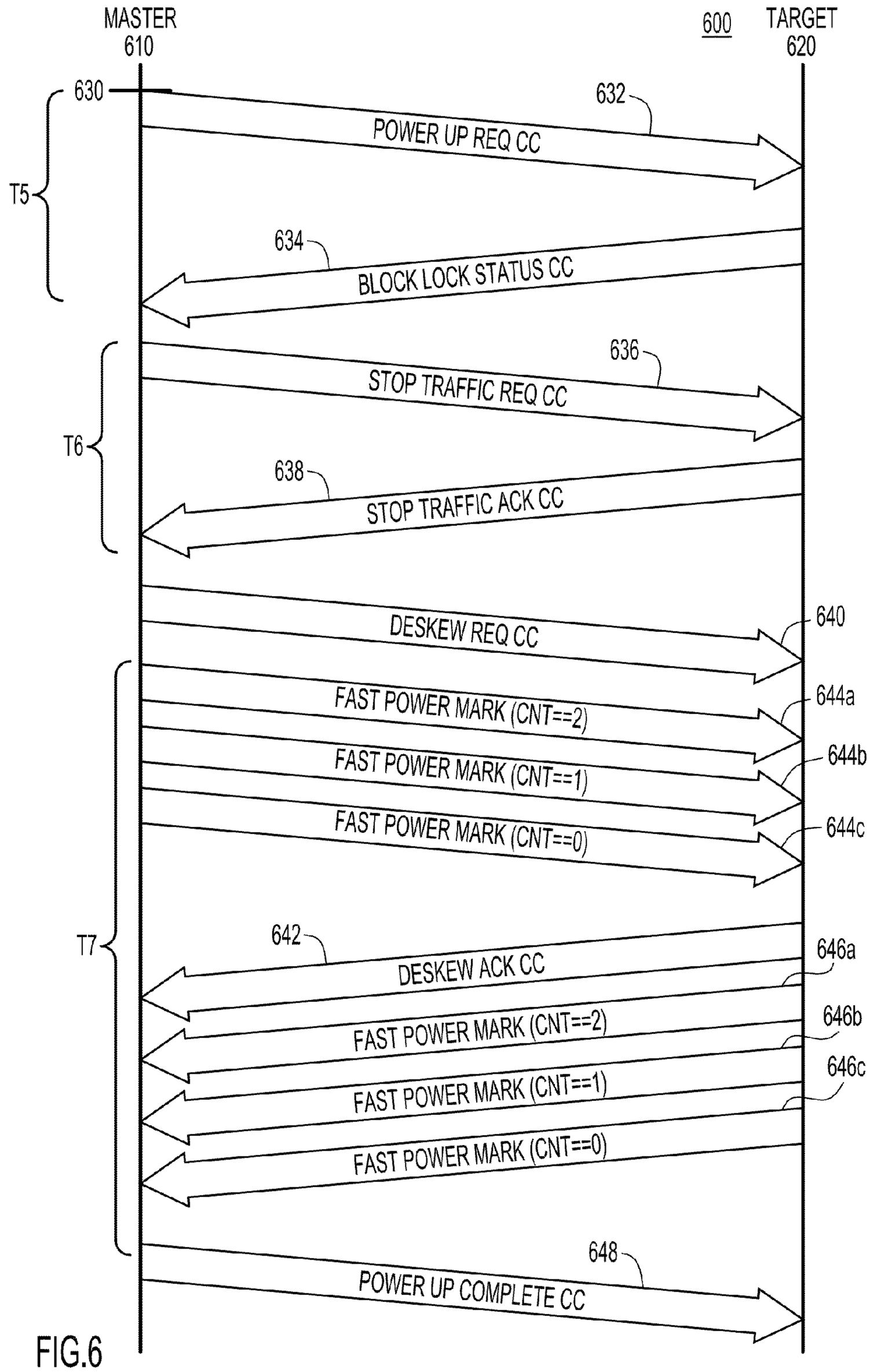

MASTER
610
600
TARGET
620
630
632
POWER UP REQ CC
T5
634
BLOCK LOCK STATUS CC
636
STOP TRAFFIC REQ CC
T6
638
STOP TRAFFIC ACK CC
DESKEW REQ CC
640
FAST POWER MARK (CNT==2)
644a
FAST POWER MARK (CNT==1)
644b
FAST POWER MARK (CNT==0)
644c
T7
642
DESKEW ACK CC
646a
FAST POWER MARK (CNT==2)
646b
FAST POWER MARK (CNT==1)
646c
FAST POWER MARK (CNT==0)
648
POWER UP COMPLETE CC

FIG.6

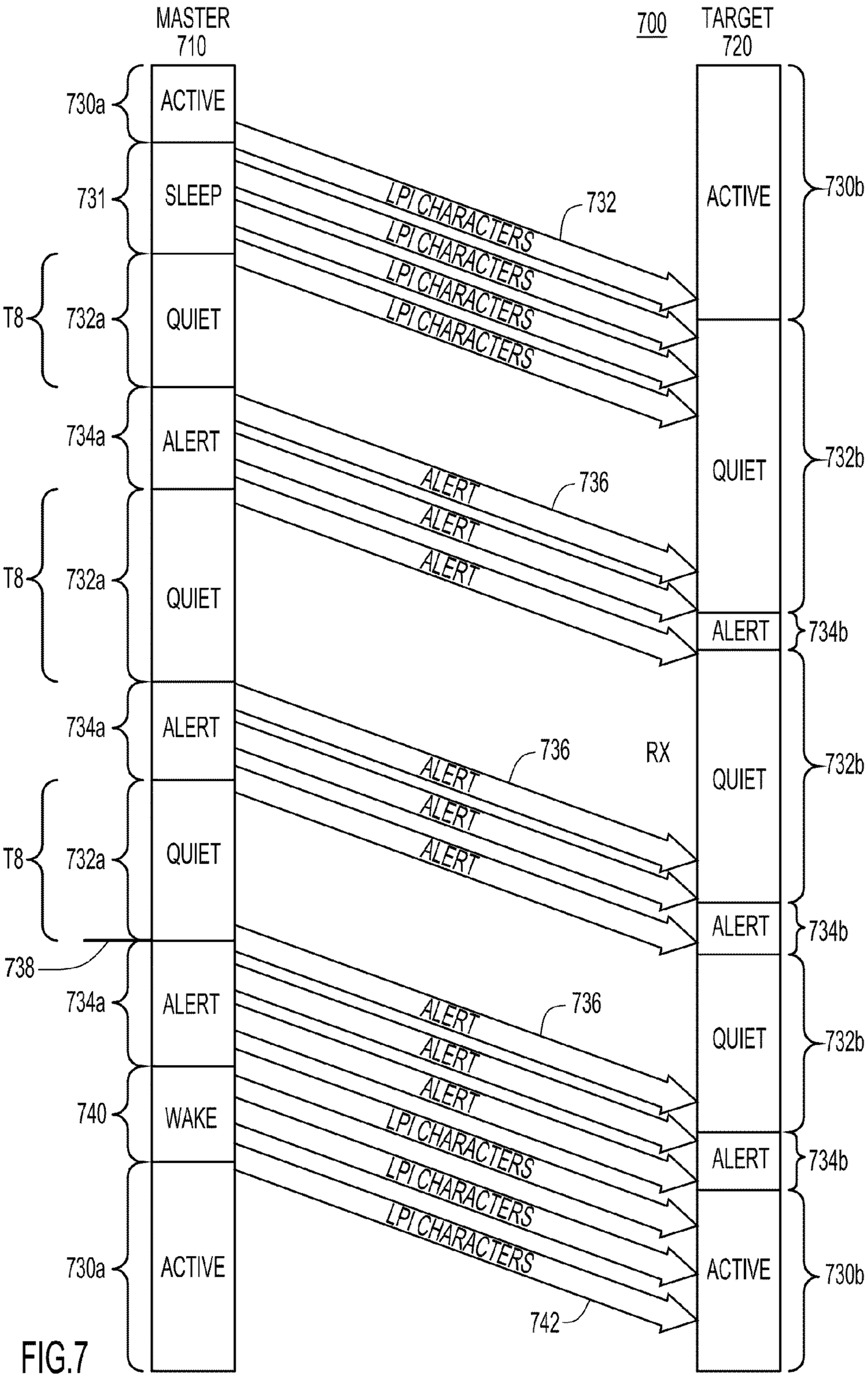

MASTER
710
700
TARGET
720
730a
ACTIVE
731
SLEEP
T8
732a
QUIET
734a
ALERT
738
740
WAKE
LPI CHARACTERS
732
ALERT
736
RX
730b
732b
734b
742

FIG.7

DYNAMIC COMMUNICATION LINK SCALING

TECHNICAL FIELD

The present disclosure relates to communication links, and in particular, multilane communication links.

BACKGROUND

In data center environments, rack units may house many server devices, such as blade servers. Each server device may be configured to host one or more physical or virtual host devices. The servers in the rack units are connected to switch devices such as Top of Rack (ToR) switch devices. The switches, in turn, are connected to other switches via a spine switch or spine fabric. Data in a communication session may be exchanged between host devices (physical and/or virtual) in the same or different rack units. For example, packets of data in the session may be sent from a host device in one rack unit to a host device in another rack unit using network or fabric links. Fabric networks provide cross-connections between multiple fabric links on the same linecard or across multiple linecards. A fabric link may include multiple fabric lanes in each fabric link.

In a fabric, significant power is consumed by the serial input/output devices used to communicate over the fabric links. For example, consider a fabric with 28 fabric links, with each link consisting of 8 fabric lanes. If each lane includes a serializer/deserializer which consumes 240 mW of power in a specific period of time, 53.7 W of power are consumed for a single fabric during that time. During the period in which the fabric link is not fully utilized, each serializer/deserializer still consumes peak power.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates a multilane communication link between a first device and a second device, in which dynamic communication link scaling is employed according to the techniques presented herein.

FIG. **2** illustrates a crossbar used to control dynamic communication link scaling in a fabric link.

FIG. **3** is a flowchart illustrating a process for powering-down at least one lane of a multilane communication link from the perspective of a master device.

FIG. **4** is a flowchart illustrating a process for powering-down at least one lane of a multilane communication link from the perspective of a target device.

FIG. **5** is a ladder diagram illustrating messages sent between the master device and the target device to power-down at least one lane of a multilane communication link.

FIG. **6** is a ladder diagram illustrating messages between the master device and the target device to power-up at least one lane of a multilane communication link.

FIG. **7** is a ladder diagram illustrating a low-power mode of a communication lane in a multilane communication link.

DESCRIPTION OF EXAMPLE EMBODIMENTS

Overview

Generally, techniques are presented herein to manage traffic flow across a plurality of communication lanes between two devices, which allows a multilane communication link operating at lower capacities to conserve power by powering-down one or more of its communication lanes. Additionally, the multilane communication link may power-up one or more of the communication lanes when high performance is needed.

Traffic is sent between a first device and a second device over a plurality of active communication lanes of a communication link. A number of the active communication lanes of the communication link is altered. Thereafter, traffic is sent over the altered number of active communication lanes.

In order to alter the number of active communication lanes, traffic between the first device and the second device across a plurality of communication lanes of a communication link is stalled. A first device side of the communication lane is powered-down on at least one of the plurality of communication lanes to deactivate the at least one communication lane. A power-down request is sent from the first device to the second device to power-down a second device side of the at least one communication lane. Traffic is resumed between the first device and the second device over the altered number of active communication lanes.

EXAMPLE EMBODIMENTS

Presented herein is a method to save power on a communication link, e.g., a fabric link, by turning some of the lanes off when traffic is still actively flowing on other lanes. and when needed, powering up more lanes. This is achieved by a link level protocol over the fabric link. The device at one of the two ends of a fabric link is designated as "Master" and the other as "Target." The Master device is capable of initiating a power saving protocol to avoid any potential deadlocks.

The protocol involves an exchange of a series of control messages before lanes can be added or removed. Adding (powering up) and removing (powering down) lanes requires the scrambler/descrambler on the transmit/receive devices to work in synchronization to avoid errors. This synchronization is achieved through "Markers." Markers play an important role in ensuring the reliability of power down/power up process, as described hereinafter.

Referring to FIG. **1**, a switched fabric **100** is configured to communicate data between a first apparatus **110** and a second apparatus **120**. Specifically, the first apparatus **110** may be embodied in a first linecard and the second apparatus **120** may be embodied in a second linecard. The fabric **100** is made up of a plurality of multilane communication links or multilane fabric links **130***a-d* and **140***a-d*. Each fabric link **130***a-d*, **140***a-d* is comprised of a plurality of fabric lanes, for example, eight fabric lanes (not shown). The multilane fabric links **130***a-d*, **140***a-d* communicate between the first apparatus **110** and second apparatus **120** through fabric spine **150**. Specifically, each end of multilane fabric links **130***a-d*, **140***a-d* terminates in a crossbar ("xbar") located in one of the first apparatus **110**, second apparatus **120**, or fabric spine **150**. Specifically, multilane fabric links **130***a-d* connect between xbar **160***a* in first apparatus **110** and xbars **170***a-d* in fabric spine **150**. Similarly, multilane fabric links **140***a-d* connect between xbars **170***a-d* in fabric spine **150** and xbar **180***b* in second apparatus **120**. For example, the ends of each of the eight fabric lanes of multilane fabric link **130***a* terminate in xbars **160***a* and **170***a*, respectively. Similarly, the ends of each of the eight lanes in multilane fabric link **140***a* terminate in xbars **170***a* and **180***b*, respectively. Each of the xbars **160***a-b*, **170***a-d*, **180***a-b* is configured to dynamically scale the number of active lanes in fabric links **130***a-d* and **140***a-d*, respectively. Additional multilane fabric links (not depicted) may form further connections between xbars **160***a,b*, **170***a-d* and **180***a,b*.

Included in first linecard **110** and second linecard **120** are application specific integrated circuits ("ASICs") **185***a-d* and **190***a-d*, respectively. Also included in linecard **110** is switched fabric **187** which includes multilane fabric links **188***a-h* and allows intercommunication between ASICs **185***a-d* through xbars **160***a* and *b*. Similarly, switched fabric **197** includes multilane fabric links **198***a-h* and allows intercommunication between ASICs **195***a-d* through xbars **180***a* and *b*. As with fabric **100**, xbars **160***a,b* and **180***a-b* may be configured to dynamically scale the number of active lanes in multilane fabric links **188***a-h* and **198***a-h*, respectively.

Turning now FIG. **2**, depicted therein is a detailed view of xbar **160***a*, multilane fabric link **130***a*, and xbar **170***a*. Specifically depicted in xbar **160***a* are the portions of xbar **160***a* which may be used during dynamic communication link scaling, while other portions of xbar **160***a* may have been omitted for simplifying the description. As can be seen in FIG. **2**, multilane fabric link **130***a* is made up of eight separate fabric lanes **210***a-h*. In order to send data across multilane fabric link **130***a*, and to dynamically scale the number of fabric lanes **210***a-h* that are active for data transfers, xbar **160***a* includes control logic **220**. Control logic **220** may be embodied in a multipurpose microprocessor. Accordingly, memory **230** may comprise software instructions that when executed by the multipurpose microprocessor cause the processor to dynamically scale the number of fabric lanes **210***a-h* used to perform data transfers, thereby performing dynamic link scaling. According to other examples, control logic **220** may be embodied in an ASIC specifically designed with hardwired logic to control the transfer of data and dynamically scale the number of fabric lanes **210***a-h* which are active for a data transfer.

When data is sent from xbar **160***a* to xbar **170***a*, data received from input/output port **240** is sent to encoder **245**. Specifically, control logic **220** may first split that data stream into separate streams for each of fabric lanes **210***a-h* that will be active during the data transfer. For example, if all of fabric lanes **210***a-h* will be active during the transfer, control logic **220** will split the input data into eight separate streams. If, on the other hand, only fabric lanes **210***a* and *b* will be active during the data transfer, control logic **220** will split the input data into two separate data streams.

Encoder **245**, upon receiving the data streams from control logic **220**, will encode the data according to the transfer protocol used to send data across fabric lanes **210***a-h*. According to one example, encoder **245** encodes the data into 64-bits, plus an additional 2 bits of data which may be used to identify the type of data sent in the remaining 64 bits of the code word. Encoder **245** may comprise eight separate data stream encoders **247***a-h*, one for each of fabric link lanes **210***a-h*.

Each 64/66 bit code word is then sent from encoder **245** to scrambler **249**. Scrambler **249** may modify the code words generated by encoder **245** to ensure a balanced data stream. Specifically, scrambler **249** may modify the code words to ensure that the number of "0"s sent over each fabric lane **210***a-h* is approximately equal to the number of "1"s sent over each fabric lane **210***a-h*, thereby ensuring a direct current ("DC") balanced data stream.

The scrambled and balanced code words are sent from scrambler **249** to mark insertion logic **251**. In mark insertion logic **251**, markers are included in the data stream that allow the sending xbar, such as xbar **160***a*, to remain in alignment with the receiving xbar, such as receiving xbar **170***a*. Similar to encoder **245**, mark insertion logic **251** may include a separate mark inserter **253***a-h* for each of fabric lanes **210***a-h*. After mark insertion, the encoded and scrambled data is sent to serializer/deserializer ("serdes") macro **255**. Serdes macro **255** contains a separate serdes **257***a-h* for each of fabric lanes **210***a-h*. The serdes **257***a-h* serialize the encoded and scrambled data for transfer to xbar **170***a* over the fabric lanes **210***a-h* that are currently active.

Upon receiving data from xbar **170***a* over fabric lanes **210***a-h*, xbar **160***a* effectively reverses the process described above for sending data. The data is received over fabric lanes **210***a-h* on the presently active fabric lanes. The serdes macro **255** receives the serialized data, and the serdes **257***a-h* deserialize the data received from their respective fabric lanes. Once deserialized, the data is sent to deskew logic **259** where the markers inserted by the sending xbar **170***a* are used to ensure that xbar **160***a* is in correct alignment with xbar **170***a*.

With alignment ensured, the markers are removed from the encoded data, and the 64/66 bit code words are sent to descrambler **261**. Descrambler **261** reverses the DC-balancing performed in the scrambler of the sending xbar, such as xbar **170***a*. The unbalanced code words are subsequently sent to decoder **263** where the data is decoded. As with encoder **245**, decoder **263** may include eight separate data stream decoders **265***a-h*, one for each of fabric lanes **210***a-h*. The decoded data is then sent to control logic **220** for use by a device, such as first linecard **110** of FIG. **1**, or subsequent transfer to another device.

Because serdes **257***a-h* use a significant amount of power even when not actually sending or receiving traffic, by powering-down a portion of serdes **257***a-h* that are not necessary to maintain sufficient communication performance, significant power savings may be achieved. Accordingly, control logic **220** is also configured to power-up and power-down one or more fabric lanes **210***a-h*, as well as their respective serdes **257***a-h*. Xbar **160***a* may serve as the master device, with control logic **220** initiating the procedure used to dynamically scale the number of active lanes in the multilane fabric link **130***a*, with receiving xbar **170***a* serving as the target device, responding to the process initiated by sending xbar **160***a*. Example processes for powering-up and powering-down fabric lanes **210***a-h* are described below with reference to FIGS. **3-8**.

Turning now to FIG. **3**, depicted therein is a flowchart of a process **300** for dynamically scaling the number of lanes in a multilane fabric link between a first device and a second device. In general, the process includes sending traffic between a first device and a second device over a plurality of active communication lanes of a communication link; altering a number of the active communication lanes of the communication link; and sending traffic over the altered number of active communication lanes. As used herein, "traffic" refers to communications between a first device and the second device unrelated to the management of the dynamic link scaling described herein. Said differently, "traffic" refers to the communications between two devices, such as two xbars, that do not serve to dynamically scale the number of communication lanes in a multilane communication link between the two devices. For example, packet data sent over the multilane fabric during normal operation may be considered traffic. On the other hand, the communications indicated by reference numerals **532-554** and **632-648**, described below with reference to FIGS. **5** and **6**, are communications used to carry out, i.e., manage, the dynamic link scaling. Accordingly messages **532-554** and **632-648** of FIGS. **5** and **6**, respectively, are not considered "traffic."

Specifically, FIG. **3** is directed toward the process steps that would be carried out by one of the devices, in this case the device that serves as the master device, in order to alter the number of the active communication lanes of the communication link. The process begins in step **310** where traffic

between a first device and a second device are stalled across a plurality of active communication lanes of a communication link. The stalling of the traffic may be in response to a determination that the traffic being handled by the plurality of communication lanes could be handled by fewer than the current number of active communication lanes. According to other examples, the stalling make take place as a regularly scheduled process. For example, if it is known that certain times of the day historically have lower traffic demands, the first device may regularly initiate the powering-down of one or more communication lanes at a specific time of the day. The term "stalled" is used herein to mean temporarily halted or interrupted.

With traffic stalled at the first device, a first device side of at least one of the plurality of communication lanes is powered-down to deactivate the at least one of the plurality of communication lanes in step **320**. A lane is said to be deactivated or inactive when it is deactivated. In step **330**, a power-down request is sent from the first device to the second device in order to power-down the second device side of the communication lane. Finally, in step **340**, traffic is resumed between the first device and the second device over the altered number of active communication lanes.

Turning to FIG. **4**, depicted therein is a process **400** for dynamically scaling the number of communication lanes in a multilane fabric link between a first device and a second device. The process contains similarities to process **300** of FIG. **3**, but the steps depicted in FIG. **4** may be carried out by a target or slave device of the power-down process. The process begins in step **410** when a power-down request is received at a second device. The second device may be serving as a slave device for the power-down process, and therefore, the power-down request may have been sent by a master device. If the first device is serving as the master device, the power-down request may have been sent by the first device. According to other examples, the power-down request may have been received from a third device, separate from the devices communicating over the multilane fabric link.

In step **420**, traffic across a plurality of communication lanes between the first device and the second device are stalled. In step **430**, a second device side of the at least one of the communication lanes is powered-down to deactivate the at least one of the plurality of communication lanes. Finally, in step **440**, traffic between the first device and the second device is restarted on the active communication lanes of the communication link.

While FIGS. **3** and **4** generally depict dynamically scaling the number of lanes in a multilane fabric link from the perspective of a master device and a target device, respectively, FIG. **5** provides a more detailed example through ladder diagram **500** illustrating example messages sent between a master device **510** and a target device **520**.

Prior to the sending of any messages in FIG. **5**, a determination is made as to which of devices **510** and **520** will serve as the master **510** and the target **520**. This determination may be made when communication between device **510** and device **520** is initiated. At some point during the communications between master **510** and target **520**, a determination is made that one or more of the communication lanes between master **510** and target **520** should be powered-down. This determination may come from either the master **510** or the target **520**, or from a third device not illustrated in FIG. **5**. Once the determination is made that at least one of the communication lanes between the master **510** and the target **520** should be powered-down, traffic being sent from the master **510** to the target **520** is stalled. The traffic may be stalled at a packet boundary to ensure continuity of the traffic between master **510** and target **520**.

Having stalled traffic between master **510** and target **520** on the master device side of the communication link, at **530** a power-down request **532** is sent from the master **510** to the target **520**. The power-down request **532** may comprise a specific code word or series of bits that the target device **520** will recognize, not as link traffic, but as a power-down request **532**. In addition to sending power-down request **532**, master device **510** may start a timer T**1** which will measure the duration until a response is received from the target **520**. If timer T**1** reaches a predetermined value, the master **510** may send another power-down request message or abort the power-down process.

At **534** the power-down request message **532** is received at target device **520**. In response to receiving power-down request message **532**, target device **520** stalls traffic from target device **520** to master device **510**. The traffic from the target device **520** to the master device **510** may also be stalled at a packet boundary to maintain the continuity of the data. Having stalled the traffic, target device **520** sends power-down acknowledgement message **536** at **538**.

Power-down acknowledgement **536** is received at the master device **510** at **540**, and the power-down acknowledgment serves as an indication to master **510** that target **520** received the power-down request. Power-down acknowledgement **536** may also serve as an indication that communications from target device **520** (except those necessary for the power-down process) have been stalled.

Master **510** may further check to ensure that no messages are received from or sent to target **520** subsequent to receiving power-down acknowledgement message **536**. Upon receipt of power-down acknowledgment **536**, master **510** sends expected powered-down mark (message) **542**. The expected power-down mark **542** is an indication to target device **520** that the power-down process is proceeding, and to expect power-down marks **544***a-c*. Both expected powered-down mark **542** and power down marks **544***a-c* may be included in the communications by mark insertion logic **251** of FIG. **2**.

After sending expected power-down mark **542**, master device **510** may wait a period of time T**2** before continuing with the power-down procedure. Time T**2** may serve to ensure any traffic that may have been delayed is received before any lanes of the communication link are rendered inactive. The master device **510** may also simply wait time period T**2** to ensure that target device **520** has sufficient time to receive expected power-down mark **542**.

At the conclusion of time T**2**, master device **510** sends power-down mark **544***a* at **546**. Master device **510** may also send additional power-down marks, such as marks **544***b* and **544***c*. By sending multiple power-down marks, the master **510** increases the reliability of the process, as the target device only needs to receive a single power-down mark to continue the power-down process.

The power-down marks **544***a-c* may be embodied as unscrambled, direct current balanced, predefined code words. The master device **510** sends power-down marks **544***a-c* at a periodic programmable interval. Power-down marks **544***a-c* a may include a countdown value so that the target **520** knows how many more power-down marks will be received before power-down happens. For example, as shown in FIG. **5**, power-down mark **544***a* has a count (Cnt) value of 2, power-down mark **544***b* has a count value of 1, and power-down mark **544***c* has a count value of 0. The period of power-down marks **544***a-c* a may be made short so as to expedite the

power-up/down process. Accordingly, the power-down marks **544***a-c* a may be referred to as fast power markers.

Having sent all power-down marks **544***a-c*, master **510** reconfigures the master side of the multilane fabric link for operation with fewer communication lanes. For example, if a scrambler is used by master **510** to ensure sufficient transitions in the data transmitted over the plurality of communication lanes, the scrambler will be reconfigured to no longer include the lanes that will be powered-down when dividing the data. Also, after sending the power-down marks **544***a-c*, master device **510** may start a timer T**3**. If a predetermined time passes without receiving a response from target **520**, master **510** may abort the power-down process and return to its previous state of operation or resend the power down marks.

Due to the countdown of the power-down marks **544***a-c*, target device **520** will reconfigure the target side of the multilane fabric link to operate correctly once one or more of the plurality of communication lanes are powered-down at the same time that master **510** reconfigures to operate without the powered-down lanes. While FIG. **5** shows three power-down marks, because the frequency is known for the sending of the marks, and each mark is identified by its countdown value, target **520** can synchronize with master **510** even if only a single power down mark is received. For example, if only power-down mark **544***b* is received by target **520**, because target **520** knows the frequency at which the power-down marks **544***a-c* are sent, and it knows that power-down mark **544***b* is the next to last mark, target **520** can anticipate when power-down mark **544***c* should have been received, and can reconfigure accordingly.

When reconfiguring target **520**, if a descrambler is used to descramble the data received over the plurality of communication lanes, the descrambler will be reconfigured to no longer descramble data from the lanes to be powered-down. Target device **520** will send expected power-down message **548** which is an indication that power-down marks **550***a-c* will be subsequently sent to master **510**. Power-down marks **550***a-c* are then sent. Target device **520** may start a timer T**4**. If a predetermined time passes without receiving a response from master **510**, target **520** may abort the power-down process and return to its previous state of operation or resend the power-down marks.

Upon receipt of at least one of power-down marks **550***a-c*, master **510** sends power-down complete message **554**. Once this message is sent, traffic is resumed from master device **510**, and the serdes on the selected lanes are powered-down one lane at a time. Master **510** can send power-down complete message **554** at the appropriate time, even if only one of power-down marks **550***a-c* is received, similar to the process described above with regard to power-down marks **544***a-c*. Power-down marks **550***a-c* may have count values similar to power-down marks **544***a-c*. Upon receipt of power-down complete message **554**, target device **520** also resumes traffic, and also powers down the serdes of the selected lanes one at a time.

Powering-down the lanes may involve depowering a serdes for each of the communication lanes that is being powered-down. Because serdes use power even when not actually sending traffic, by powering-down the serdes that are not necessary to maintain sufficient communication performance, significant power savings may be achieved. As depicted in FIG. **5**, the serdes for the communication lanes are sent into a deep sleep mode. According to other examples, powering-down the serdes may comprises transitioning the serdes to a low-power mode, or stand-by mode that allows the serdes to transition back to an active mode more quickly.

According to the example of FIG. **5**, the selected lanes are not actually powered-down at the master device **510** and the target device **520** until after traffic has resumed on the lanes that are to remain active. By waiting until traffic has resumed before powering-down the lanes, the period during which traffic is stalled between the master device **510** and the target device **520** may be reduced. It is also noted that the lanes to be powered-down are powered-down one at a time at both the master **510** and the target **520**. By powering-down the lanes one at a time, noise in the active communication lanes can be reduced.

With reference now made to FIG. **6**, depicted therein is a ladder diagram **600** illustrating a process by which inactive lanes of a multilane fabric link can be powered-up. Generally, when higher performance is needed, a first device sends a power up request to the second device in order to power-up a communication lane of a communication link. A first device side of the communication link powers-up at least one of the plurality of communication lanes. The second device side of the communication link also powers-up the at least one of the plurality of communication link. A stop traffic request is sent from the first device to the second device. Then traffic between the first device and the second device across a plurality of communication lanes of a communication link is stalled. The newly powered-up lane is included in the lane alignment. Once the newly powered-up lane is included in the lane alignment, traffic between the first device and the second device is restarted over the active communication lanes.

For example, master device **610** may have previously depowered lanes according to the process of FIG. **2** and/or FIG. **5**. According to other examples, master **610** may have simply been initialized with one or more of its communication lanes in a depowered state.

At some point during the communications between master **610** and target **620**, a determination is made that one or more of the inactive communication lanes between master **610** and target **620** should be powered-up. This determination may come from either the master **610** or the target **620**, or from a third device not illustrated in FIG. **5**. Once the determination is made, at **630** master **610** sends power-up request **632** to target **620**. Also at **630**, master **610** begins powering-up the master side of the previously inactive lanes, and may also start a timer T**5**. If the timer T**5** exceeds a predetermined length of time without having received a response from target **620**, the powering-up process may be aborted or the power-up request may be resent.

The powering-up of the master side of the communication lanes may comprise powering-up a previously unpowered serdes on the master side of the communication lane. According to other examples, which will be described in more detail with reference to FIG. **7** below, powering-up of the serdes may comprise altering an operational mode of a serdes from a low power sleep mode to a normal mode of operation.

Upon receiving powering-up request **632**, target **620** will begin powering-up the target side of the previously inactive lanes. In one example, the communication lanes are powered-up before traffic is stalled in order to shorten the period of time during which traffic is not being sent between master **610** and target **620**. Once all of the lanes to be powered-up have been powered-up, and synchronized with the master side of the communication lanes, target device **620** sends block lock status message **634**. Block lock status message is an indication to master **610** that all of the previously inactive lanes have been powered-up and the master side of the lanes are synchronized with the target side of the lanes, and traffic between the master **610** and target **620** may now be stalled.

Upon receiving block lock status message **634**, master **610** stalls incoming traffic, and sends a stop traffic request (Req) **636** to target **620**. Master **610** may also start a timer T**6**. If time T**6** exceeds a predetermined length of time without having received a response from target **620**, the powering-up process may be aborted or the stop traffic request may be resent.

Upon receiving stop traffic request **636**, target **620** may stall incoming traffic, and also send traffic stop acknowledgment (Ack) **638** to master **610**. When both master **610** and target **620** stall their incoming traffic, they may do so at a packet boundary to ensure the continuity of the traffic data. Once target **620** has stalled its incoming traffic, stop traffic acknowledgment **638** is sent from target **620** to master **610**.

Upon receipt of stop traffic acknowledgement **638**, master **610** sends deskew request message **640** which is a message to target **620** indicating that target **620** should begin realigning for traffic transmission over all powered-up lanes, including the recently powered-up lanes. Master **610** may start timer T**7** having a predetermined time duration, in order to wait to see if target **620** completes its realignment process. If the predetermined period of time T**7** is reached without receiving an indication from target **620** that it has completed its realignment, master **610** may retry initiating the realignment process with target **620**, or abort the powering-up process.

Master **610** begins realigning itself to enable sending traffic over the recently powered-up lanes. For example, a scrambler may be reconfigured to direct current balance the data sent across all of the powered-up link lanes, including the recently powered-up lanes. Accordingly, master **610** may enable the scrambler for use with all communication lanes, and depower the scrambler used when fewer than all the lanes are in use. Upon receiving deskew request **640**, target **620** begins realigning for transmission over all of the powered-up lanes, including the recently powered-up lanes. The realignment may comprise realigning a single scrambler to operate over all of the powered-up lanes, or enabling a scrambler which operates when all lanes are powered-up. When the realignment process has begun at target **620**, target **620** sends deskew acknowledgement **642**.

During the realignment process, master **610** may send power marks **644***a-c* and target **620** may send power marks **646***a-c*. As with power marks **544***a-c* and **550***a-c* described above in connection with FIG. **5**, power marks **644***a-c* and **646***a-c* allow master **610** and target **620** to ensure correct timing of the alignment for the traffic that will soon be sent over the powered-up communication lanes. Similarly, power marks **644***a-c* and **646***a-c* may be included in communications through mark insertion logic **251** of FIG. **2**. While FIG. **6** depicts deskew acknowledgement **642** being sent after receipt of power marks **644***a-c*, this need not be the case. Instead, deskew acknowledgement **642** is sent in response to having received deskew request **640**, regardless of whether power marks **644***a-c* have been sent or received.

Upon receipt of power-up marks **646***a-c*, master **610** sends power-up complete message **648** and once again sends traffic over the communication link, now over all of the powered-up lanes. Similarly, upon receipt of power-up complete message **648**, target **620** begins sending and receiving traffic over all of the powered-up lanes of the communication link.

As indicated above, the serdes on both the master side and the target side of a communication lane may be completely powered-up and powered-down, or may be alternated between a full-power mode and a low-power sleep mode. Turning to FIG. **7**, depicted therein is a ladder diagram **700** illustrating how the serdes on the master **710** and target **720** side of communication link may transition from a full-power mode to a low-power sleep mode. Through the use of the low-power/full-power transitions, instead of completely powering-up and powering-down the serdes, significant time savings can be achieved during the powering-up and powering-down processes.

At the start of the message exchange depicted by ladder diagram **700**, both master **710** and target **720** operate in a full-power active state **730***a* and **730***b*, respectively. When master **710** powers-down the master side of the communication lane, for example, as described above in reference to FIG. **4**, the master side of the communication lane enters an initial sleep state **731** during which messages **732** are sent from master **710** to target **720**. Messages **732** may comprise specific code words or a specific series of characters which indicate to target **720** that the target-side of the communication lane should be placed in a quiet mode. According to one example, the process is initiated by master **710** transmitting unscrambled lower power idle (LPI) characters (0707070707070707) to the target **720**. After sending messages **732**, master **710** initiates counter T**8** and enters a quite state **732***a*. Upon receiving messages **732**, target **720** similarly enters a low-power quiet state **732***b*.

At the expiration of timer T**8**, the master-side of the communication lane enters a low power active state **734***a*. In the lower-power active state **734***a*, master **710** sends alert messages **736** to target **720**. Alert messages **736** place the target-side of the communication lane into an alert state **734***b* so that the target-side is prepared to receive messages that will place it in a full-power state, if necessary. If no such messages are received, the target-side of the communication lane returns to quiet state **732***b*. Similarly, if master **710** does not initiate powering-up of the communication lane, the master-side of the communication lane returns to quiet state **732***a*. The master **710** and target **720** repeat this process until the communication lane is to be powered-up. An analogous process may also take place over the target **720** to master **710** link as well.

At **738**, a powering-up of the communication lane is initiated. The powering-up of the communication lane may cut short timer T**8**, placing the master-side of the communication lane in an alert state **734***b* earlier than otherwise would have been the case. According to other examples, the powering-up process will simply wait until timer T**8** expires. Master **710** sends alert messages **736** as it normally would to place the target-side of the communication lane into alert state **734***b*. After sending messages **736**, the master-side of the communication lane enters wake-up mode **740** during which it sends messages **742**. Messages **742** may comprise specific code words or series of characters which indicate to target **720** that the target-side of the communication lane should enter an active state. Similar to the message sent to initiate the lower power sleep mode, the wake-up process may be initiated by the master **710** transmitting unscrambled LPI characters (0707070707070707) to the target **720**. After sending messages **742**, the master-side of the communication lane returns to full-power active mode **730***a*. Similarly, the target-side of the communication lane returns to full-power active mode **730***b*. Because the serdes for the master-side of the communication lane and the target-side of the communication lane were not fully powered-down, the transition to the full-power modes **730***a* and **730***b* takes place more quickly than it would if the serdes were fully depowered and de-synchronized.

In summary, the foregoing presents techniques to save power on a fabric link by turning some of the lanes off when traffic is still actively flowing on other lanes and when needed, powering up more lanes. This is achieved by a link level protocol over the fabric link. There are numerous advantages of these techniques. The fabric link runs with expected bandwidth and not over speed, thus saving power. Different power

modes can be chosen depending on the needed power saving/response time. There is no traffic loss during powering up/down. The power down feature can be used to keep the fabric link active even if some of the serdes links are bad or not working. The power down feature can also be used to allow for programming in serdes on lanes that are down, while the fabric link is still active.

The above description is intended by way of example only.

What is claimed is:

1. A method comprising:

sending traffic between a first device and a second device over a plurality of active communication lanes of a communication link;

stalling at the first device, the traffic between the first device and the second device across the plurality of active communication lanes of the communication link;

altering a number of the active communication lanes of the communication link;

sending a power-down request from the first device to the second device to power-down a second device side of the at least one communication lane;

receiving from the second device a power-down acknowledgement indicating that traffic has been stalled at the second device;

restarting traffic on active communication lanes between the first device and the second device over the communication link;

sending a power-down complete message to the second device indicating that traffic has been restarted over the communication link;

sending traffic over the altered number of active communication lanes.

2. The method of claim **1**, wherein altering the number of active communication lanes comprises:

powering-down a first device side of at least one of the plurality of communication lanes to deactivate the at least one of the plurality of communication lanes.

3. The method of claim **2**, wherein the first device comprises a first linecard, the second device comprises a second linecard, and the communication link comprises a fabric link between the first linecard and the second linecard.

4. The method of claim **2**, further comprising:

sending to the second device a power-down mark indicating that the first device side of the at least one communication lane has been powered-down; and

receiving from the second device a power-down mark indicating that the second device side of the at least one communication lane has been powered-down.

5. The method of claim **2**, wherein powering-down the first device side of at least one of the plurality of communication lanes to deactivate the at least one of the plurality of communication lanes comprises powering down the at least one of the plurality of communication lanes after traffic has restarted over the communication link.

6. The method of claim **2**, wherein placing the deactivated at least one of the plurality of communication lanes in a low-power mode comprises alternating the state of the first device side of the deactivated at least one of the plurality of communication lanes from a quiet state to an alert state.

7. The method of claim **4**, wherein sending the power-down mark comprises sending a plurality of power-down marks to the second device; and

receiving the power-down mark comprises receiving a plurality of power-down marks from the second device.

8. The method of claim **4**, wherein powering-down the first device side of the at least one communication lane comprises disabling a scrambler at the first device; and

receiving the power-down mark comprises receiving an indication that a descrambler has been disabled at the second device.

9. The method of claim **1**, wherein stalling traffic comprises stalling traffic over each of the plurality of communication lanes.

10. The method of claim **1**, wherein sending the power-down request comprises sending the power-down request prior to powering-down the first device side of at least one of the plurality of communication lanes.

11. The method of claim **1**, wherein before stalling at the first device, the traffic between the first device and the second device across the plurality of active communication lanes of the communication link or after sending traffic over the altered number of active lanes, further comprising:

sending a power-up request to the second device;

powering-up a first device side of at least one inactive communication lane of the communication link to activate the at least one communication lane;

stalling traffic with the second device in response to sending the power-up request;

sending a realignment request to the second device;

sending a power-up complete message to the second device; and

restarting traffic with the second device in response to sending the power-up complete message.

12. The method of claim **11**, wherein powering-up the at least one first device side of the deactivated communication lane comprises synchronizing the first device side with a second device side of the deactivated communication lane.

13. The method of claim **11**, further comprising:

receiving a realignment acknowledgment from the second device.

14. The method of claim **1**, further comprises:

receiving from the first device the power-down request at the second device,

stalling, at the second device, traffic between the first device and the second device across the plurality of communication lanes of the communication link,

powering-down the second device side of at least one of the plurality of communication lanes to deactivate the at least one of the plurality of communication lanes.

15. The method of claim **14**, further comprising:

sending the power-down acknowledgement from the second device to the first device indicating that traffic has been stalled at the second device;

receiving a power-down mark from the first device indicating that a first device side of the at least one communication lane has been powered-down;

sending a power-down mark to the first device indicating that the second device side of the at least one communication lane has been powered-down.

16. The method of claim **15**, wherein stalling traffic is performed in response to receiving the power-down request; powering-down the second device side of the at least one communication lane is performed in response to receiving the power-down mark; and

sending the power-down mark is in response to powering-down the second device.

17. The method of claim **15**, wherein receiving the power-down mark comprises receiving a plurality of power-down marks from the first device; and

sending the power-down mark comprises sending a plurality of power-down marks to the first device.

18. The method of claim **1**, wherein before stalling at the first device, the traffic between the first device and the second device across the plurality of active communication lanes of

the communication link or after sending traffic over the altered number of active lanes, further comprising:

receiving a power-up request from the first device;

powering-up a second device side of at least one deactivated communication lane of the communication link;

stalling traffic with the first device in response to the powering-up of the second device side of the at least one deactivated communication lane;

receiving a realignment request from the first device;

receiving a power-up complete message from the first device.

19. The method of claim **18**, further comprising:

sending a realignment acknowledgment to the first device.

20. An apparatus comprising:

a network interface unit configured to enable communications over a network on behalf of a first device; and

a processor coupled to the network interface unit and configured to:

send and receive traffic from a second device over a plurality of active communication lanes of a communication link;

stall at the first device, the traffic between the first device and the second device across the plurality of active communication lanes of the communication link;

alter a number of the active communication lanes of the communication link;

send, via the network interface, a power-down request from the first device to the second device to power-down a second device side of the at least one communication lane;

receive from the second device a power-down acknowledgement indicating that traffic has been stalled at the second device;

restart traffic on active communication lanes between the first device and the second device over the communication link;

send a power-down complete message to the second device indicating that traffic has been restarted over the communication link;

send traffic over the altered number of active communication lanes.

21. The apparatus of claim **20**, wherein the processor is further configured to:

power-down a first device side of at least one of the plurality of communication lanes to deactivate the at least one of the plurality of communication lanes.

22. The apparatus of claim **21**, wherein the processor is further configured to:

send to the second device, through the network interface unit, a power-down mark indicating that the first device side of the at least one communication lane has been powered-down; and

receive from the second device a power-down mark indicating that the second device side of the at least one communication lane has been powered-down.

23. The apparatus of claim **21**, wherein the processor is further configured to:

send a plurality of power-down marks to the second device; and

receive a plurality of power-down marks from the second device.

24. A tangible, non-transitory computer readable medium comprising instructions that when executed by a processor cause the processor to:

send and receive traffic between a first device and a second device over a plurality of active communication lanes of a communication link;

stall at the first device, the traffic between the first device and the second device across the plurality of active communication lanes of the communication link;

alter a number of the active communication lanes of the communication link;

send a power-down request from the first device to the second device to power-down a second device side of the at least one communication lane;

receive from the second device a power-down acknowledgement indicating that traffic has been stalled at the second device;

restart traffic on active communication lanes between the first device and the second device over the communication link;

send a power-down complete message to the second device indicating that traffic has been restarted over the communication link;

send traffic over the altered number of active communication lanes.

25. The computer readable medium of claim **24**, wherein the instructions further cause the processor to:

power-down a first device side of at least one of the plurality of communication lanes to deactivate the at least one of the plurality of communication lanes.

26. The computer readable medium of claim **25**, wherein the instructions further cause the processor to:

send to the second device a power-down mark indicating that the first device side of the at least one communication lane has been powered-down; and

receive from the second device, a power-down mark indicating that the second device side of the at least one communication lane has been powered-down.

27. The computer readable medium of claim **25**, wherein the instructions further cause the processor to:

send a plurality of power-down marks to the second device; and

receive a plurality of power-down marks from the second device.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 9,270,599 B1
APPLICATION NO. : 13/860961
DATED : February 23, 2016
INVENTOR(S) : Srividhya Nagarajan et al.

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

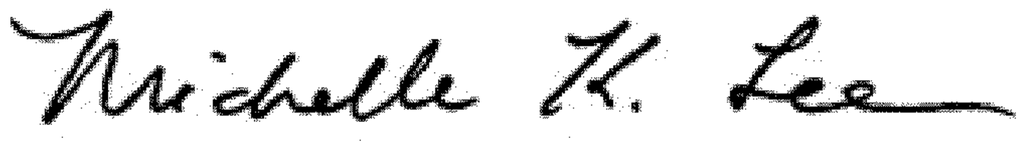
In the Claims

In Col. 11, line 30, insert --and-- after “link;”.

In Col. 13, line 40, insert --and-- after “link;”.

In Col. 14, line 31, insert --and-- after “link;”.

Signed and Sealed this
Twenty-fourth Day of May, 2016



Michelle K. Lee
Director of the United States Patent and Trademark Office