Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 3000228 A1 2017/04/27

(21) 3 000 228

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2016/10/20
(87) Date publication PCT/PCT Publication Date: 2017/04/27
(85) Entrée phase nationale/National Entry: 2018/03/27
(86) N° demande PCT/PCT Application No.: US 2016/057951
(87) N° publication PCT/PCT Publication No.: 2017/070371
(30) Priorités/Priorities: 2015/10/23 (US62/245,731); 2016/06/10 (US62/348,619); 2016/10/19 (US15/298,071)

(51) Cl.Int./Int.Cl. *G06F 13/16* (2006.01)

(71) Demandeur/Applicant:
QUALCOMM INCORPORATED, US

(72) Inventeurs/Inventors:
MISHRA, LALAN JEE, US;
WIETFELDT, RICHARD, US;
O'SHEA, HELENA DEIRDRE, US;
CHEN, ZHENQI, US;
ROETHIG, WOLFGANG, US

(74) Agent: SMART & BIGGAR

(54) Titre : DISPOSITIFS FRONTAUX RADIOFREQUENCE AVEC ECRITURE MASQUEE
(54) Title: RADIO FREQUENCY FRONT END DEVICES WITH MASKED WRITE

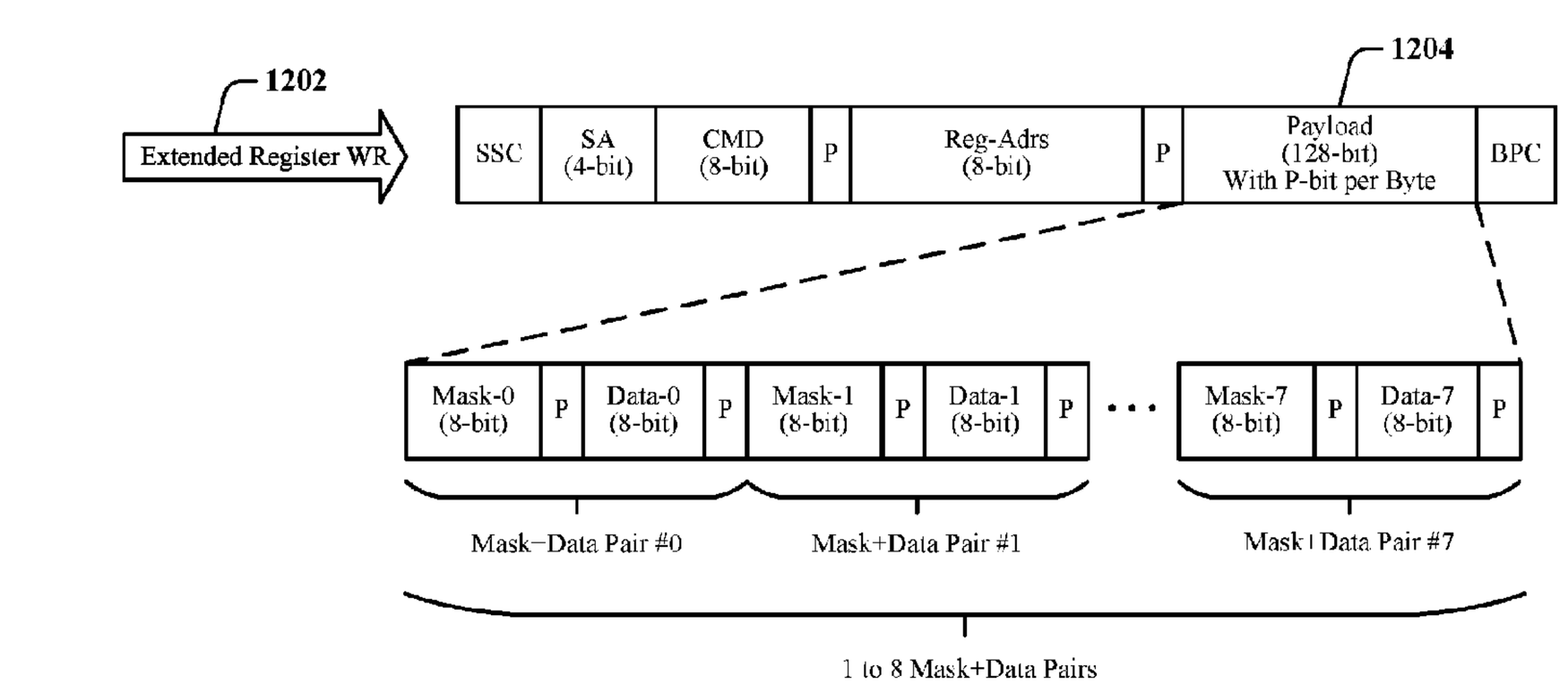

FIG. 12

(57) Abrégé/Abstract:
Methods and apparatuses are described that facilitate the communication of data between a transmitter and a receiver across a serial bus interface. In one configuration, a transmitter generates a datagram based on a 16-bit address and a mask-and-data pair burst length, the 16-bit address including a most significant byte (MSB) and a least significant byte (LSB), compares the MSB to a receiver base address maintained in a shadow register, compares the mask-and-data pair burst length to a receiver masked-write burst length maintained in the shadow register, and sends the datagram to the receiver via the bus interface when: the MSB is equal to the receiver base address maintained in the shadow register, and the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 April 2017 (27.04.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/070371 A3

(51) International Patent Classification:
G06F 13/16 (2006.01)

(21) International Application Number:
PCT/US2016/057951

(22) International Filing Date:
20 October 2016 (20.10.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

62/245,731	23 October 2015 (23.10.2015)	US
62/348,619	10 June 2016 (10.06.2016)	US
15/298,071	19 October 2016 (19.10.2016)	US

(71) Applicant: **QUALCOMM INCORPORATED** [US/US]; ATTN: International IP Administration, 5775 Morehouse Drive, San Diego, CA 92121-1714 (US).

(72) Inventors: **MISHRA, Lalan, Jee**; 5775 Morehouse Drive, San Diego, CA 92121-1714 (US). **WIETFELDT, Richard**; 5775 Morehouse Drive, San Diego, CA 92121-1714 (US). **O'SHEA, Helena, Deirdre**; 5775 Morehouse Drive, San Diego, CA 92121-1714 (US). **CHEN, Zhenqi**; 5775 Morehouse Drive, San Diego, CA 92121-1714 (US). **ROETHIG, Wolfgang**; 5775 Morehouse Drive, San Diego, CA 92121-1714 (US).

(74) Agent: **LOZA, Julio**; Loza & Loza, LLP, 305 North Second Avenue #127, Upland, CA 91786 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

[Continued on next page]

(54) Title: RADIO FREQUENCY FRONT END DEVICES WITH MASKED WRITE

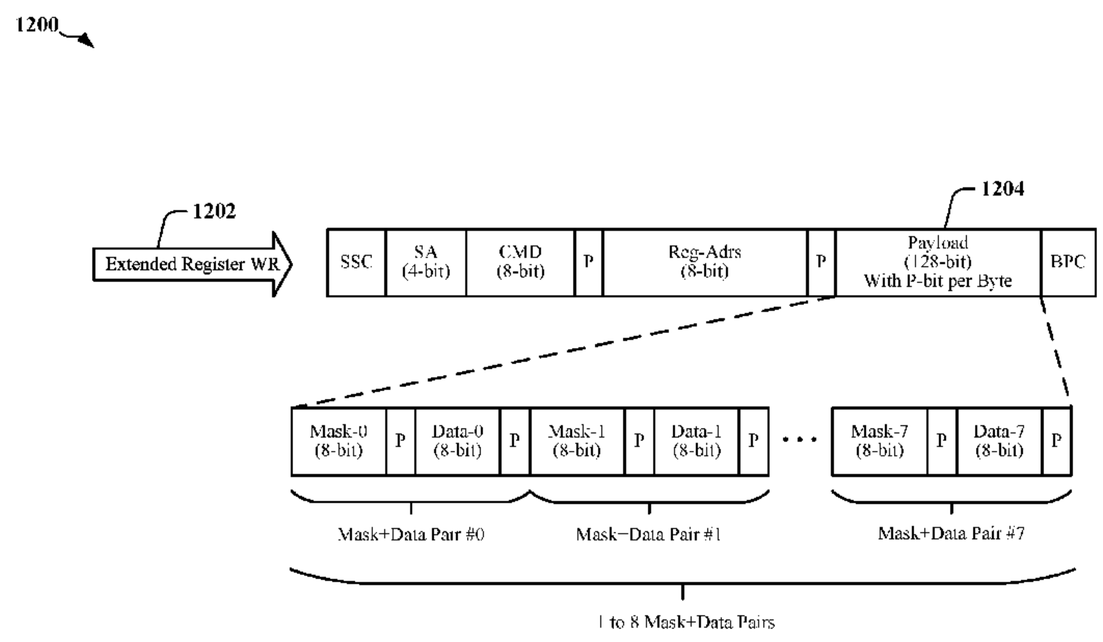

FIG. 12

(57) Abstract: Methods and apparatuses are described that facilitate the communication of data between a transmitter and a receiver across a serial bus interface. In one configuration, a transmitter generates a datagram based on a 16-bit address and a mask-and-data pair burst length, the 16-bit address including a most significant byte (MSB) and a least significant byte (LSB), compares the MSB to a receiver base address maintained in a shadow register, compares the mask-and-data pair burst length to a receiver masked-write burst length maintained in the shadow register, and sends the datagram to the receiver via the bus interface when: the MSB is equal to the receiver base address maintained in the shadow register, and the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register.

WO 2017/070371 A3

WO 2017/070371 A3

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(88) Date of publication of the international search report:
8 June 2017

RADIO FREQUENCY FRONT END DEVICES WITH MASKED WRITE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to and the benefit of Provisional Application No. 62/245,731 filed in the U.S. Patent and Trademark Office on October 23, 2015, Provisional Application No. 62/348,619 filed in the U.S. Patent and Trademark Office on June 10, 2016, and Non-Provisional Application No. 15/298,071 filed in the U.S. Patent and Trademark Office on October 19, 2016, the entire contents of which are incorporated herein by reference.

BACKGROUND

Field

[0002] The present disclosure relates generally to data transfer, and more particularly, to radio frequency front end (RFFE) devices having a masked write operation.

Background

[0003] As the mobile device market mushroomed with the development of multifunctional smartphones, cellular communication complexity has increased accordingly. It is now conventional for a radio front end of a mobile device to cover as many as ten or more frequency bands. The radio front end thus requires multiple power amplifiers, diplexers, low-noise amplifiers, antenna switches, filters, and other radio frequency (RF) front end devices to accommodate the radio signaling complexity. These various RF front end devices are in turn controlled by a host or master device such as a radio frequency integrated circuit (RFIC). As the RF front end complexity increased, the need for a standardized protocol to control the many different devices lead to the development of the Mobile Industry Processor Interface (MIPI) RF Front End Control Interface (RFFE) standard.

[0004] The RFFE standard specifies a serial bus that includes a clock line and a bidirectional data line. Through the RFFE bus, an RFFE master device may read from, and write to, registers in a plurality of RFFE slave devices so as to control the RF front end devices. The read and write commands are organized in the RFFE standard into

protocol messages that may each include an initial sequence start condition (SSC), a command frame, a data payload, and a final bus park cycle (BPC). The protocol messages include register commands, extended register commands, and extended register long commands. The protocol messages may further include broadcast commands. The register, extended register, and extended register long commands (three types of commands) can all be either read or write commands. With regard to the three types of commands, the registers in each of the RFFE slave devices are organized into a 16-bit wide address space (0x0000 – 0xFFFF in hexadecimal). Each of the three types of commands includes a command frame that addresses a specific RFFE slave device as well as the register address. A command frame in the register command (register command frame) is directed to the registers in the first five bits of an address space (0x00 – 0x1F) such that only five register address bits are needed. The register command frame is followed by an 8-bit data payload frame. In contrast, an extended register command frame includes eight register address bits and may be followed by up to 16 bytes of data. Finally, an extended register long command frame includes a full 16-bit register address so it can uniquely identify any register in the addressed RFFE slave device. The extended register long command frame may be followed by up to eight bytes of data.

[0005] Each of the commands begins with a unique sequence start condition (SSC) that is then followed by a corresponding command frame, some number of data frames, and finally a bus park cycle (BPC) to signal the end of the command. The latency involved with transmitting any of the commands thus depends on the number of bits in its various frames as well as the clocking speed for the RFFE clock line. Under the RFFE protocol, each bit of a transmitted frame corresponds to a period of the clock since the transmission is single data rate (SDR), which corresponds to one bit per clock cycle. For example, an SDR results from transmitting a bit responsive to each rising edge (or to just the falling edges) of the clock. The maximum clocking speed is 52 MHz in the RFFE v2 specification. This clocking rate has increased relative to previous versions of the RFFE protocol and is associated with increased power consumption.

[0006] Each of the three types of RFFE commands – extended register, extended register long, and register – may be either a read or a write command. In general, each write command writes a full byte to each specified register. However, it may be the

case that the RFFE master device does not need to change all eight bits in a RFFE slave device register. Furthermore, in many devices, more than one master or radio access technology (RAT) component may share control bit(s) in the same RFFE slave device register. To avoid contaminating the bits corresponding to the "other" source that writes to the same register, a "partial write" operation may be desired. In such a partial write operation, the RFFE master device must first perform a read operation on the selected slave device register using the appropriate one of the three command types. The RFFE master device then knows the current state of all the bits in the corresponding RFFE slave device register. The RFFE master device may then issue an RFFE write command using the appropriate one of the three command types in which the data payload for the corresponding slave device register has the bits it is changing while all the remaining bits stay in their current state as determined by the previous read operation. The need for a read operation prior to the partial write operation increases latency that may violate the latency requirements of certain Radio Access Technologies being implemented in the corresponding RF frontend.

[0007] Accordingly, there is a need in the art for RFFE messaging with decreased latency for partial write operations.

SUMMARY

[0008] Embodiments disclosed herein provide systems, methods, and apparatuses that facilitate the communication of data between a transmitter and a receiver across a serial bus interface. A radio frequency front end front end (RFFE) network is provided in which a master device may issue masked write commands to its slave devices that do not require any read operations to determine the value of unchanged bits in addressed slave device registers. Each masked write command may include a mask field or bit index field that identifies bit position(s) of bit(s) to be changed in an addressed slave device register.

[0009] In an aspect of the disclosure, a method performed at a transmitter for sending data to a receiver via a bus interface, includes generating a datagram based on a 16-bit address and a mask-and-data pair burst length, the 16-bit address including a most significant byte (MSB) and a least significant byte (LSB), comparing the MSB to a receiver base address maintained in a shadow register, comparing the mask-and-data

pair burst length to a receiver masked-write burst length maintained in the shadow register, and sending the datagram to the receiver via the bus interface when: the MSB is equal to the receiver base address maintained in the shadow register, and the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register. The datagram sent to the receiver does not include the MSB and the mask-and-data pair burst length.

[0010] In an aspect, the comparing the MSB to the receiver base address maintained in the shadow register includes detecting whether the MSB is equal to the receiver base address maintained in the shadow register. When the MSB is not equal to the receiver base address maintained in the shadow register, the method includes setting a base address at the receiver to be equal to the MSB, and updating the receiver base address maintained in the shadow register to the MSB. The base address at the receiver is set by sending a write access command to the receiver prior to sending the datagram.

[0011] In an aspect, the comparing the mask-and-data pair burst length to the receiver masked-write burst length maintained in the shadow register includes detecting whether the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register. When the mask-and-data pair burst length is not equal to the receiver masked-write burst length maintained in the shadow register, the method includes setting a masked-write burst length at the receiver to be equal to the mask-and-data pair burst length, and updating the receiver masked-write burst length maintained in the shadow register to the mask-and-data pair burst length. The masked-write burst length at the receiver is set by sending a write access command to the receiver prior to sending the datagram.

[0012] In another aspect of the disclosure, a transmitter for sending data to a receiver, includes a bus interface and a processing circuit. The processing circuit is configured to generate a datagram based on a 16-bit address and a mask-and-data pair burst length, the 16-bit address including a most significant byte (MSB) and a least significant byte (LSB), compare the MSB to a receiver base address maintained in a shadow register, compare the mask-and-data pair burst length to a receiver masked-write burst length maintained in the shadow register, and send the datagram to a receiver via the bus interface when: the MSB is equal to the receiver base address maintained in the shadow register, and the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register. The

datagram sent to the receiver does not include the MSB and the mask-and-data pair burst length.

[0013] In a further aspect of the disclosure, a transmitter for sending data to a receiver, includes means for generating a datagram based on a 16-bit address and a mask-and-data pair burst length, the 16-bit address including a most significant byte (MSB) and a least significant byte (LSB), means for comparing the MSB to a receiver base address maintained in a shadow register, means for comparing the mask-and-data pair burst length to a receiver masked-write burst length maintained in the shadow register, and means for sending the datagram to the receiver via the bus interface when: the MSB is equal to the receiver base address maintained in the shadow register, and the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register. The datagram sent to the receiver does not include the MSB and the mask-and-data pair burst length.

[0014] In an aspect of the disclosure, a method performed at a transmitter for sending data to a receiver, includes generating a mask field in a datagram to be transmitted through an interface to the receiver, the mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register, generating a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register, and transmitting the datagram through the interface, wherein the datagram is addressed to the RFFE register of the receiver.

[0015] In an aspect, the mask field further indicates a remaining set of bits to be left unchanged in the RFFE register. In an aspect, the mask field is a bit index field identifying a bit position to be changed in the RFFE register of the receiver and the data field is a bit value field providing a bit value for the bit position identified in the bit index field.

[0016] In an aspect, the method further includes generating a command field in the datagram, the command field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0017] In an aspect, the method further includes generating a command field in the datagram, the command field indicating that the datagram is a masked write command, and generating a mode field in the datagram, the mode field indicating whether the datagram is an extended register masked write command, an extended register long

masked write command, a register masked write command, or an extended register short masked write command.

[0018] In another aspect of the disclosure, a transmitter for sending data to a receiver, includes a bus interface and a processing circuit. The processing circuit is configured to generate a mask field in a datagram to be transmitted through the bus interface to the receiver, the mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register, generate a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register, and transmit the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver.

[0019] In an aspect, the processing circuit is further configured to generate a command field in the datagram, the command field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0020] In an aspect, the processing circuit is further configured to generate a command field in the datagram, the command field indicating that the datagram is a masked write command, and generate a mode field in the datagram, the mode field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0021] In a further aspect of the disclosure, a transmitter for sending data to a receiver, includes means for generating a mask field in a datagram to be transmitted through an interface to the receiver, the mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register, means for generating a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register, and means for transmitting the datagram through the interface, wherein the datagram is addressed to the RFFE register of the receiver.

[0022] In an aspect, the transmitter further includes means for generating a command field in the datagram, the command field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0023] In an aspect, the transmitter further includes means for generating a command field in the datagram, the command field indicating that the datagram is a masked write command, and means for generating a mode field in the datagram, the mode field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0024] In an aspect of the disclosure, a method performed at a receiver for receiving data from a transmitter, includes receiving a datagram through an interface from the transmitter, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the receiver, reading a mask field in the datagram, the mask field identifying at least one bit to be changed in the RFFE register, reading a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register, and changing the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field.

[0025] In an aspect, the mask field further indicates a remaining set of bits to be left unchanged in the RFFE register. In an aspect, the mask field is a bit index field identifying a bit position to be changed in the RFFE register of the receiver, and the data field is a bit value field providing a bit value for the bit position identified in the bit index field.

[0026] In an aspect, the method further includes reading a command field in the datagram, the command field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0027] In an aspect, the method further includes reading a command field in the datagram, the command field indicating that the datagram is a masked write command, and reading a mode field in the datagram, the mode field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0028] In another aspect of the disclosure, a receiver for receiving data from a transmitter, includes a bus interface and a processing circuit. The processing circuit is configured to receive a datagram through the bus interface from the transmitter, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the

receiver, read a mask field in the datagram, the mask field identifying at least one bit to be changed in the RFFE register, read a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register, and change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field.

[0029] In an aspect, the processing circuit is further configured to read a command field in the datagram, the command field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0030] In an aspect, the processing circuit is further configured to read a command field in the datagram, the command field indicating that the datagram is a masked write command, and read a mode field in the datagram, the mode field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0031] In a further aspect of the disclosure, a receiver for receiving data from a transmitter, includes means for receiving a datagram through an interface from the transmitter, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the receiver, means for reading a mask field in the datagram, the mask field identifying at least one bit to be changed in the RFFE register, means for reading a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register, and means for changing the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field.

[0032] In an aspect, the receiver further includes means for reading a command field in the datagram, the command field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0033] In an aspect, the receiver further includes means for reading a command field in the datagram, the command field indicating that the datagram is a masked write command, and means for reading a mode field in the datagram, the mode field indicating whether the datagram is an extended register masked write command, an

extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0034] In an aspect of the disclosure, a method performed at a transmitter for sending data to a receiver through a bus interface, includes setting a configuration register to indicate whether a masked-write operation is enabled with respect to a datagram to be transmitted to the receiver, generating a command field in the datagram, the command field indicating whether the datagram is an extended register write command or an extended register long write command, generating a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register and a data field providing a value of the at least one bit to be changed in the RFFE register, and transmitting the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver.

[0035] In an aspect, the configuration register includes eight register bits, and the setting the configuration register includes setting a third register bit of the eight register bits to a value of 1 to indicate that the masked-write operation is enabled and to a value of 0 to indicate that the masked-write operation is disabled, and when the masked-write operation is enabled, setting a fourth register bit of the eight register bits to a value of 1 to indicate that the masked-write operation is enabled with respect to the extended register long write command and to a value of 0 to indicate that the masked-write operation is enabled with respect to the extended register write command.

[0036] In another aspect of the disclosure, a transmitter for sending data to a receiver, includes a bus interface and a processing circuit. The processing circuit is configured to set a configuration register to indicate whether a masked-write operation is enabled with respect to a datagram to be transmitted to the receiver, generate a command field in the datagram, the command field indicating whether the datagram is an extended register write command or an extended register long write command, generate a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register and a data field providing a value of the at least

one bit to be changed in the RFFE register, and transmit the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver.

[0037] In a further aspect of the disclosure, a transmitter for sending data to a receiver through a bus interface, includes means for setting a configuration register to indicate whether a masked-write operation is enabled with respect to a datagram to be transmitted to the receiver, means for generating a command field in the datagram, the command field indicating whether the datagram is an extended register write command or an extended register long write command, means for generating a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register and a data field providing a value of the at least one bit to be changed in the RFFE register, and means for transmitting the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver.

[0038] In an aspect of the disclosure, a method performed at a receiver for receiving data from a transmitter through a bus interface, includes reading a configuration register to detect whether a masked-write operation is enabled with respect to a datagram to be received from the transmitter, receiving the datagram through the bus interface from the transmitter, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the receiver, reading a command field in the datagram, the command field indicating whether the datagram is an extended register write command or an extended register long write command, reading a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in the RFFE register and a data field providing a value of the at least one bit to be changed in the RFFE register, and changing the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair.

[0039] In an aspect, the configuration register includes eight register bits, and the reading the configuration register includes detecting that the masked-write operation is enabled when a third register bit of the eight register bits is set to a value of 1 and detecting that the masked-write operation is disabled when the third register bit is set to a value of 0. When the masked-write operation is enabled, the reading the

configuration register further includes detecting that the masked-write operation is enabled with respect to the extended register long write command when a fourth register bit of the eight register bits is set to a value of 1 and detecting that the masked-write operation is enabled with respect to the extended register write command when the fourth register bit is set to a value of 0.

[0040] In another aspect of the disclosure, a receiver for receiving data from a transmitter, includes a bus interface and a processing circuit. The processing circuit is configured to read a configuration register to detect whether a masked-write operation is enabled with respect to a datagram to be received from the transmitter, receive the datagram through the bus interface from the transmitter, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the receiver, read a command field in the datagram, the command field indicating whether the datagram is an extended register write command or an extended register long write command, read a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in the RFFE register and a data field providing a value of the at least one bit to be changed in the RFFE register, and change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair.

[0041] In a further aspect of the disclosure, a receiver for receiving data from a transmitter through a bus interface, includes means for reading a configuration register to detect whether a masked-write operation is enabled with respect to a datagram to be received from the transmitter, means for receiving the datagram through the bus interface from the transmitter, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the receiver, means for reading a command field in the datagram, the command field indicating whether the datagram is an extended register write command or an extended register long write command, means for reading a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in the RFFE register and a data field providing a value of the at least one bit to be changed in the RFFE register, and means for changing the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair.

[0042] In an aspect of the disclosure, a method performed at a transmitter for sending data to a receiver through a bus interface, includes enabling a masked-write operation by setting a single bit within a configuration register at the receiver to a first value, disabling the masked-write operation by setting the single bit within the configuration register at the receiver to a second value, generating a datagram to be transmitted to the receiver via the bus interface, the datagram providing an address value, generating a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register and a data field providing a value of the at least one bit to be changed in the RFFE register, enabling a page segmented access operation by setting another single bit within the configuration register at the receiver to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the datagram when the page segmented access operation is enabled, disabling the page segmented access operation by setting the other single bit within the configuration register at the receiver to a second value, wherein the address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled, and transmitting the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver. The datagram may be an extended register write datagram or an extended register write long datagram.

[0043] In another aspect of the disclosure, a transmitter for sending data to a receiver, includes a bus interface and a processing circuit. The processing circuit is configured to enable a masked-write operation by setting a single bit within a configuration register at the receiver to a first value, disable the masked-write operation by setting the single bit within the configuration register at the receiver to a second value, generate a datagram to be transmitted to the receiver via the bus interface, the datagram providing an address value, generate a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register and a data field providing a value of the at least one bit to be changed in the RFFE register, enable a page segmented access operation by setting another single bit within the configuration

register at the receiver to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the datagram when the page segmented access operation is enabled, disable the page segmented access operation by setting the other single bit within the configuration register at the receiver to a second value, wherein the address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled, and transmit the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver. The datagram may be an extended register write datagram or an extended register write long datagram.

[0044] In an aspect of the disclosure, a method performed at a receiver for receiving data from a transmitter through a bus interface, includes receiving a first datagram from the transmitter for setting a single bit within a configuration register at the receiver, detecting that a masked-write operation is enabled when the single bit within the configuration register is set to a first value, detecting that the masked-write operation is disabled when the single bit within the configuration register at the receiver is set to a second value, receiving a second datagram from the transmitter, the second datagram providing an address value, reading a payload field in the second datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register of the receiver and a data field providing a value of the at least one bit to be changed in the RFFE register, receiving a third datagram from the transmitter for setting another single bit within the configuration register at the receiver, detecting that a page segmented access operation is enabled when the other single bit within the configuration register at the receiver is set to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the datagram when the page segmented access operation is enabled, detecting that the page segmented access operation is disabled when the other single bit within the configuration register at the receiver is set to a second value, wherein the address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled, and changing the at least one bit in the

RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair.

[0045] In another aspect of the disclosure, a receiver for receiving data from a transmitter, includes a bus interface and a processing circuit. The processing circuit is configured to receive a first datagram from the transmitter for setting a single bit within a configuration register at the receiver, detect that a masked-write operation is enabled when the single bit within the configuration register is set to a first value, detect that the masked-write operation is disabled when the single bit within the configuration register at the receiver is set to a second value, receive a second datagram from the transmitter, the second datagram providing an address value, read a payload field in the second datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register of the receiver and a data field providing a value of the at least one bit to be changed in the RFFE register, receive a third datagram from the transmitter for setting another single bit within the configuration register at the receiver, detect that a page segmented access operation is enabled when the other single bit within the configuration register at the receiver is set to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the datagram when the page segmented access operation is enabled, detect that the page segmented access operation is disabled when the other single bit within the configuration register at the receiver is set to a second value, wherein the address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled, and change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair.

BRIEF DESCRIPTION OF THE DRAWINGS

[0046] FIG. 1 illustrates an apparatus that includes a RF front end (RFFE) that may be adapted according to certain aspects disclosed herein.

[0047] FIG. 2 is a block diagram illustrating a device that employs a RFFE bus to couple various front end devices.

[0048] FIG. 3 illustrates an example of a system architecture for an apparatus employing a data link between IC devices according to certain aspects disclosed herein.

[0049] FIG. 4 is a diagram illustrating reserved command fields in an RFFE protocol.

[0050] FIG. 5 is a diagram illustrating four masked write commands that include N-bit mask fields in accordance with an aspect of the disclosure.

[0051] FIG. 6 is a diagram illustrating a modification of the masked write commands of FIG. 5 in which a single reserved command field is employed in accordance with an aspect of the disclosure.

[0052] FIG. 7 is a diagram illustrating four masked write commands that include a bit index identifying the bit position of a bit to be changed in accordance with an aspect of the disclosure.

[0053] FIG. 8 is a diagram illustrating a modification of the masked write commands of FIG. 7 in which a single reserved command field is employed in accordance with an aspect of the disclosure.

[0054] FIG. 9 is a diagram illustrating an example packet structure supporting a 16-bit address space and N-pairs of mask-and-data bytes.

[0055] FIG. 10 is a diagram illustrating an example datagram in a transmit buffer.

[0056] FIG. 11 is a diagram illustrating an example operation for a datagram in a transmit buffer.

[0057] FIG. 12 is a diagram illustrating an example packet structure of an extended register write command supporting a masked-write operation.

[0058] FIG. 13 is a diagram illustrating an example packet structure of an extended register write long command supporting a masked-write operation.

[0059] FIG. 14 is a diagram illustrating a bit structure of a configuration register.

[0060] FIG. 15 is a diagram of an RFFE register space.

[0061] FIG. 16 is a diagram of an RFFE register space having a configuration register and a page-address register.

[0062] FIG. 17 illustrates a table defining another example bit structure of a configuration register and a diagram depicting a function of the configuration register bits.

[0063] FIG. 18 is a diagram illustrating page segmented access.

[0064] FIG. 19 is a block diagram illustrating an example of an apparatus employing a processing circuit that may be adapted according to certain aspects disclosed herein.

[0065] FIG. 20 is a flow chart of a method for sending data to a receiver in accordance with certain aspects disclosed herein.

[0066] FIG. 21 is a flow chart of another method for sending data to a receiver in accordance with certain aspects disclosed herein.

[0067] FIG. 22 is a flow chart of a further method for sending data to a receiver in accordance with certain aspects disclosed herein.

[0068] FIG. 23 is a flow chart of another method for sending data to a receiver in accordance with certain aspects disclosed herein.

[0069] FIG. 24 is a diagram illustrating an example of a hardware implementation for a transmitting apparatus and employing a processing circuit adapted according to certain aspects disclosed herein.

[0070] FIG. 25 is a flow chart of a method for receiving data from a transmitter in accordance with certain aspects disclosed herein.

[0071] FIG. 26 is a flow chart of another method for receiving data from a transmitter in accordance with certain aspects disclosed herein.

[0072] FIG. 27 is a flow chart of a further method for receiving data from a transmitter in accordance with certain aspects disclosed herein.

[0073] FIG. 28 is a diagram illustrating an example of a hardware implementation for a receiving apparatus and employing a processing circuit adapted according to certain aspects disclosed herein.

DETAILED DESCRIPTION

[0074] Various aspects are now described with reference to the drawings. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of one or more aspects. It may be evident, however, that such aspect(s) may be practiced without these specific details.

[0075] As used in this application, the terms “component,” “module,” “system” and the like are intended to include a computer-related entity, such as, but not limited to hardware, firmware, a combination of hardware and software, software, or software in

execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program and/or a computer. By way of illustration, both an application running on a computing device and the computing device can be a component. One or more components can reside within a process and/or thread of execution and a component may be localized on one computing device and/or distributed between two or more computing devices. In addition, these components can execute from various computer readable media having various data structures stored thereon. The components may communicate by way of local and/or remote processes such as in accordance with a signal having one or more data packets, such as data from one component interacting with another component in a local system, distributed system, and/or across a network such as the Internet with other systems by way of the signal.

[0076] Moreover, the term "or" is intended to mean an inclusive "or" rather than an exclusive "or." That is, unless specified otherwise, or clear from the context, the phrase "X employs A or B" is intended to mean any of the natural inclusive permutations. That is, the phrase "X employs A or B" is satisfied by any of the following instances: X employs A; X employs B; or X employs both A and B. In addition, the articles "a" and "an" as used in this application and the appended claims should generally be construed to mean "one or more" unless specified otherwise or clear from the context to be directed to a singular form.

Exemplary Apparatus with Multiple IC Device Subcomponents

[0077] Certain aspects of the invention may be applicable to communications links deployed between electronic devices that include subcomponents of an apparatus such as a telephone, a mobile computing device, an appliance, automobile electronics, avionics systems, etc. FIG. 1 depicts an apparatus 100 that may employ a communication link between IC devices. In one example, the apparatus 100 may be a mobile communication device. The apparatus 100 may include a processing circuit having two or more IC devices 104, 106 that may be coupled using a first communication link. One IC device may be an RF front end device 106 that enables the apparatus to communicate through one or more antennas 108 with a radio access network, a core access network, the Internet and/or another network. The RF front end

device 106 may include a plurality of devices coupled by a second communication link, which may include an RFFE bus.

[0078] The processing circuit 102 may include one or more application-specific IC (ASIC) devices 104. In one example, an ASIC device 104 may include and/or be coupled to one or more processing devices 112, logic circuits, one or more modems 110, and processor readable storage such as a memory device 114 that may maintain instructions and data that may be executed by a processor on the processing circuit 102. The processing circuit 102 may be controlled by one or more of an operating system and an application programming interface (API) layer that supports and enables execution of software modules residing in storage media. The memory device 114 may include read-only memory (ROM) or random-access memory (RAM), electrically erasable programmable ROM (EEPROM), flash cards, or any memory device that can be used in processing systems and computing platforms. The processing circuit 102 may include or have access to a local database or parameter storage that can maintain operational parameters and other information used to configure and operate apparatus 100. The local database may be implemented using one or more of a database module, flash memory, magnetic media, EEPROM, optical media, tape, soft or hard disk, or the like. The processing circuit may also be operably coupled to external devices such as the antennas 108, a display 120, operator controls, such as a button 124 and/or an integrated or external keypad 122, among other components.

Overview of the RFFE Bus

[0079] FIG. 2 is a block diagram 200 illustrating an example of a device 202 that employs an RFFE bus 208 to couple various front end devices 212–217. A modem 204 including an RFFE interface 210 may also be coupled to the RFFE bus 208. In various examples, the device 202 may be implemented with one or more baseband processors 206, one or more other communication links 220, and various other buses, devices and/or different functionalities. In the example, the modem 204 may communicate with a baseband processor 206, and the device 202 may be embodied in one or more of a mobile computing device, a cellular phone, a smart phone, a session initiation protocol (SIP) phone, a laptop, a notebook, a netbook, a smartbook, a personal digital assistant (PDA), a satellite radio, a global positioning system (GPS) device, a smart home device, intelligent lighting, a multimedia device, a video device, a digital audio player

(e.g., MP3 player), a camera, a game console, an entertainment device, a vehicle component, avionics systems, a wearable computing device (e.g., a smart watch, a health or fitness tracker, eyewear, etc.), an appliance, a sensor, a security device, a vending machine, a smart meter, a drone, a multicopter, or any other similar functioning device.

[0080] The RFFE bus 208 may be coupled to an RF integrated circuit (RFIC) 212, which may include one or more controllers, and/or processors that configure and control certain aspects of the RF front end. The RFFE bus 208 may couple the RFIC 212 to a switch 213, an RF tuner 214, a power amplifier (PA) 215, a low noise amplifier (LNA) 216, and a power management module 217.

[0081] In an example, the baseband processor 206 may be a master device. The master device/baseband processor 206 may drive the RFFE bus 208 to control the various front end devices 212–217. During transmission, the baseband processor 206 may control the RFFE interface 210 to select the power amplifier 215 for a corresponding transmission band. In addition, the baseband processor 206 may control the switch 213 so that the resulting transmission may propagate from an appropriate antenna. During reception, the baseband processor 206 may control the RFFE interface 210 to receive from the low noise amplifier 216 depending on the corresponding transmission band. It will be appreciated that numerous other components may be controlled through the RFFE bus 208 in this fashion such that the device 202 is merely representative and not limiting. Moreover, other devices such as the RFIC 212 may serve as an RFFE master device in alternative embodiments.

[0082] FIG. 3 is a block schematic diagram illustrating an example of an architecture for a device 300 that may employ an RFFE bus 330 to connect bus master devices 320_1–320_N and slave devices 302 and 322_1–322_N. The RFFE bus 330 may be configured according to application needs, and access to multiple buses 330 may be provided to certain of the devices 320_1–320_N, 302, and 322_1–322_N. In operation, one of the bus master devices 320_1–320_N may gain control of the bus and transmit a slave identifier (slave address) to identify one of the slave devices 302 and 322_1–322_N to engage in a communication transaction. Bus master devices 320_1–320_N may read data and/or status from slave devices 302 and 322_1–322_N, and may write data to memory or may configure the slave devices 302 and 322_1–322_N. Configuration may involve

writing to one or more registers or other storage on the slave devices 302 and 322_1–322_N.

[0083] In the example illustrated in FIG. 3, a first slave device 302 coupled to the RFFE bus 330 may respond to one or more bus master devices 320_1–320_N, which may read data from, or write data to the first slave device 302. In one example, the first slave device 302 may include or control a power amplifier (see the PA 215 in FIG. 2), and one or more bus master devices 320_1–320_N may from time-to-time configure a gain setting at the first slave device 302.

[0084] The first slave device 302 may include RFFE registers 306 and/or other storage devices 324, a processing circuit and/or control logic 312, a transceiver 310 and an interface including a number of line driver/receiver circuits 314a, 314b as needed to couple the first slave device 302 to the RFFE bus 330, e.g., via a serial clock line (SCLK) 316 and a serial data line (SDATA) 318. The processing circuit and/or control logic 312 may include a processor such as a state machine, sequencer, signal processor or general-purpose processor. The interface may be implemented using the state machine. Alternatively, the interface may be implemented in software on a suitable processor if included in the first slave device 302. The transceiver 310 may include one or more receivers 310a, one or more transmitters 310c and certain common circuits 310b, including timing, logic and storage circuits and/or devices. In some instances, the transceiver 310 may include encoders and decoders, clock and data recovery circuits, and the like. A transmit clock (TXCLK) signal 328 may be provided to the transmitter 310c, where the TXCLK signal 328 can be used to determine data transmission rates.

[0085] The RFFE bus 330 is typically implemented as a serial bus in which data is converted from parallel to serial form by a transmitter, which transmits the encoded data as a serial bitstream. A receiver processes the received serial bitstream using a serial-to-parallel convertor to deserialize the data. The serial bus may include two or more wires, and a clock signal may be transmitted on one wire with serialized data being transmitted on one or more other wires. In some instances, data may be encoded in symbols, where each bit of a symbol controls the signaling state of a wire of the RFFE bus 330.

[0086] To control the slave devices 302 and 322_1–322_N, a master device (e.g., one of master devices 320_1–320_N) either writes or reads to RFFE registers within the slave devices, e.g., the RFFE registers 306 within the first slave device 302. The RFFE

registers 306 may be arranged according to an RFFE register address space that ranges from a zeroth (0) address to a 65535 address. In other words, each slave device may include up to 65,536 registers. To address such a number of registers, 16 register address bits for each of the slave devices 302 and 322_1–322_N are required. The master device may read from or write to the registers 306 in each slave device using one of the three types of commands discussed above (register command, extended register command, or extended register long command). For example, the register command addresses only the first 32 registers 306 in the address space for each of the slave devices 302 and 322_1–322_N. In this fashion, the register command requires only five register address bits. In contrast, the extended register command may initially access up to the first 256 registers in each of the slave devices 302 and 322_1–322_N. A corresponding 8-bit register address for the extended register command acts as a pointer in that the data payload for the extended register command may include up to 16 bytes. A corresponding read or write operation for an extended register command may thus extend across 16 registers starting from the register identified by the 8-bit register address. The extended register long command includes a 16-bit register address which may act as a pointer to any of the possible 65,536 registers in each slave device. The data payload for an extended register long command may include up to eight bytes so that the corresponding read or write operation for the extended register long command may extend across eight registers starting from the register identified by the 16-bit address. In an aspect of the disclosure, up to 15 slave devices may be coupled to one RFFE bus. If a front end includes more than 15 slave devices, additional RFFE busses may be provided.

Exemplary Masked Write Operating Environment for Radio Frequency Front End (RFFE) Devices

[0087] In an aspect of the disclosure, it is not uncommon for a register to be shared by two slave devices or components. For example, a pair of LNAs may each be configured by four of the eight bits in a shared slave register. The sharing of a register by two constrained slave devices may be referred to as a "highly integrated" register mapping. Configuring just one of the slave devices requires a partial write operation in that only four bits of an 8-bit slave register are written to rather than all eight bits. The remaining bits for the other slave device sharing the register must be left unchanged. A

master device may include a "shadow register" that may be loaded with the contents of a corresponding slave device through a read operation. The master device may write to the slave device by using the contents of the shadow register and merely changing the corresponding bits for the particular slave device and leaving the remaining bits for the shared register unaffected. Such a partial write operation may be referred to as a "read-modify-write" operation in that it involves a read of the slave device register, a modification of just the selected bits, and a write operation to the entire eight bits in the slave register using the modified bits and unmodified bits from the corresponding shadow register. The use of a shadow register not only necessitates a prior read in case of a multi-master configuration but also adds to silicon area due to the additional register space requirement.

[0088] FIG. 4 is a diagram illustrating reserved command fields in an RFFE protocol. To reduce the latency of conventional RFFE commands over the RFFE bus 208 for partial write operations (read-modify-write operations), new command frames are provided herein that invoke a masked write mode of transmission. To provide these new command frames, the reserved command frames established by the RFFE protocol are exploited. In that regard, the RFFE protocol reserved at least 12 command frames 400 as shown in FIG. 4 ranging from a reserved command frame at hexadecimal 10 to a reserved command frame at hexadecimal 1B. Each reserved command frame begins with a sequence start condition (SSC) followed by a four-bit slave device address (SA (4)) as shown in FIG. 4. Each reserved command is eight bits long. For example, the reserved command at hexadecimal 10 comprises the eight bits 00010000. All of the reserved commands are followed by a parity bit P followed by address (Reg-Adrs) and data frames for reserved purposes.

[0089] FIG. 5 is a diagram illustrating four masked write commands that include N-bit mask fields in accordance with an aspect of the disclosure. To signal the use of a masked write operation, four of the reserved command frames (designated as command frames CF1 through CF4) may be used to identify masked write RFFE commands 500 as shown in FIG. 5. In these masked write commands 500, a mask N-bit field 510 identifies the masked bits that will be left unchanged by the masked write operation and the unmasked bits that will be changed by the masked write operation. N is the number of bits in the corresponding registers. The following discussion will assume that N = 8, but it will be appreciated that other register widths may be used in alternative

implementations. A data N-bit field 512 provides the binary value of the unmasked bits. For example, an extended register masked write command (Extended Register WR) 502 begins with an SSC followed by a 4-bit slave device address (Slave Address (4-bit)). An 8-bit command frame CF1 taken from one of the reserved command frames 400 discussed with regard to FIG. 4 identifies command 502 to the receiving slave device interface. An 8-bit address (Reg-Adrs (8-bit)) identifies the address of the register in the corresponding slave device for the extended register mask write operation. An idle symbol (bus park cycle) completes command 502.

[0090] An extended register long masked write command (Extended Register WR Long) 504 also begins with an SSC and the four-bit slave address (Slave Address (4-bit)), but is followed by a unique reserved command frame CF2. Command frame CF2 is followed by a sixteen-bit register address (Reg-Adrs (16-bit)), mask N-bit field 510, data N-bit field 512, and the idle symbol. A register masked write command (Register WR) 506 also begins with an SSC and the slave address field (Slave Address (4-bit)) followed by a unique reserved command frame CF3. The reserved command frame CF3 is followed by a 5-bit register address (Reg-Adrs (5-bit)), mask N-bit field 510, data N-bit field 512, and the idle symbol. Finally, an extended register short masked write command (Extended Register WR Short) 508 is analogous to the extended register long masked write command 502 except that it uses a unique reserved command code CF4 and a 6-bit, 7-bit, or 9-15 bit register address (Reg-Adrs (9-15 bit)). The number of register bits may be established during a device initialization phase.

[0091] FIG. 6 is a diagram illustrating a modification of the masked write commands of FIG. 5 in which a single reserved command field is employed in accordance with an aspect of the disclosure. Rather than use four of the reserved command frames as in FIG. 5, FIG. 6 depicts use of a single reserved command frame for generic masked write commands 600. All of the commands 600 begin with an SSC and a slave address (Slave Address (4-bit)) and end with an idle symbol. A generic extended register masked write command (Extended Register WR) 602 uses a reserved command frame CF1 that is followed by a 2-bit mode field 614 having, for example, a value of (0, 0) to signify that an extended register masked write operation is intended. Analogous to command 502, command 602 includes an 8-bit register address, mask N-bit field 610, data N-bit field 612, and an idle symbol. A generic extended register long masked write command (Extended Register WR Long) 604 uses the same reserved

command field CF1. Command 604 also includes the 2-bit mode field 614 having a value of, for example, (0, 1) to signify that an extended register long masked write operation is intended. Analogous to command 604, command 604 includes a 16-bit register address, mask N-bit field 610, data N-bit field 612, and an idle symbol. A generic register masked write command (Register WR) 606 also includes the reserved command field CF1 and a 2-bit mode field 614 having a value of, for example, (1, 0) to identify that a register masked write operation is intended at the following 5-bit register address using mask N-bit field 610 and data N-bit field 612. Finally, a generic register short masked write command (Extended Register WR Short) 608 is analogous to the generic extended register long masked command 508 except that it uses a 9-15 bit register address (Reg-Adrs (9-15 bit)).

[0092] FIG. 7 is a diagram illustrating four masked write commands that include a bit index identifying the bit position of a bit to be changed in accordance with an aspect of the disclosure. Should only one bit need to be changed in the addressed register, mask N-bit field 510 may be replaced by a log2(N) bit index 710 as shown in FIG. 7 that uniquely identifies the bit position that will be written to in the corresponding N-bit wide slave device register. A bit value field 712 identifies what the bit value should be for the bit position identified by the bit index 710. It will be appreciated that the bit index 710 and the bit value field 712 may be used in an embodiment in which each masked command includes a unique reserved command field analogously as discussed with regard to FIG. 5. Thus, an extended register indexed masked write command (Extended Register WR) 702 is analogous to the command 502 except that fields 510 and 512 are replaced by fields 710 and 712, respectively. Similarly, an extended register long indexed masked write command (Extended Register WR Long) 704 is analogous to the command 504, a register indexed masked write command (Register WR) 706 is analogous to the command 506, and an extended register short indexed masked write command (Extended Register WR Short) 708 is analogous to the command 508, but for the replacements of the fields 510 and 512 with the fields 710 and 712, respectively.

[0093] FIG. 8 is a diagram illustrating a modification of the masked write commands of FIG. 7 in which a single reserved command field is employed in accordance with an aspect of the disclosure. The number of reserved commands may be reduced even further as shown in FIG. 8 for generic indexed masked write

commands 800 that use a common reserved command field CF1. To identify the type of indexed masked write operation, commands 800 each include a 2-bit mode field 814 as discussed with regard to FIG. 6. Thus, a generic extended register indexed masked write command (Extended Register WR) 802 is analogous to the command 602 except that fields 610 and 612 are replaced by fields 810 and 812, respectively. Similarly, a generic extended register long indexed masked write command (Extended Register WR Long) 804 is analogous to the command 604, a generic register indexed masked write command (Register WR) 806 is analogous to the command 606, and a generic extended register short indexed masked write command (Extended Register WR Short) 808 is analogous to the command 608, but for the replacements of the fields 610 and 612 with the fields 810 and 812, respectively. In an aspect of the disclosure, a master device interface and a slave device interface are configured to implement the masked write operations discussed herein. This is quite advantageous because the conventional need for a read-modify-write sequence (i.e., read the contents of a register prior to a partial write operation) on the RFFE is eliminated. In this fashion, the disclosed masked write operations advantageously reduce bus communication latency.

[0094] The techniques discussed above either makes use of multiple command frames (a.k.a. command codes) or a single command-frame in conjunction with mode bits. While the above techniques may be preferred in some implementations, an additional technique will be described below that has the added benefits of address-paging and burst-write. Although the technique is rooted in RFFE enhancement, its application is not specifically limited to an RFFE bus, but may be applicable to other bus architectures as well.

[0095] As described above, an RFFE masked-write command may provide one N-bit mask field and one N-bit control data field per datagram. Each datagram has a fixed overhead of 15 clock cycles (SSC: 1-cycle, USID: 4-cycles, command code: 8-cycles, parity: 1-cycle, BPC: 1-cycle). However, for applications where multiple mask-and-data bits are to be sent, use of multiple datagrams may not be the most optimal way to transfer data due to an associated overhead. Moreover, the number of unused reserved RFFE command codes are limited. Thus, use of multiple reserved command codes to indicate burst transfer may not be appropriate since there may not be enough command codes available, even to accommodate, for example, eight bursts of masked-write commands.

[0096] Therefore, there is a need for a new technique to enable write access in the entire RFFE UDR space that provides support for a burst mode while using one command code. Accordingly, aspects of the present disclosure provide for a page addressing scheme and burst writing scheme with mask-and-data byte pairs.

[0097] In the page addressing scheme, a slave device has a 1-byte base-address-register. For example, the base-address register may contain a value of 0x00 in hexadecimal. A master device maintains a copy of the slave device's base-address-register at the master device in a shadow register. The master device prepares a datagram based on a 16-bit address, i.e., 1-byte most significant byte (MSB) and 1-byte least significant byte (LSB), but will only send the 1-byte LSB in the masked-write datagram. Prior to sending the masked-write datagram, the master device will compare the 1-byte MSB of the 16-bit address to a current copy of the slave device's base-address-register in the shadow register. If the value of the slave device's base-address-register does not match the 1-byte MSB of the 16-bit address, the master device will first set (or update) the base address on the slave device (to change the page) in order to match the 1-byte MSB of the 16-bit address. The master device may use a register-write access command or any other type of write access command (as preferred) to perform the slave device base address change before sending the masked-write datagram. The page is only to be changed when a page mismatch is detected prior to datagram transmission. The master device may further update the shadow register with the updated slave device base address.

[0098] In the burst writing scheme with mask-and-data byte pair, the slave device has a 1-byte masked-write burst length register. For example, the burst length register may have a value of 0x01 in hexadecimal. The master device maintains a copy of the slave device's masked-write burst length register at the master device in the shadow register. The master device prepares the datagram based on a mask-and-data pair burst length (e.g., number of mask-and-data byte pairs) specified, but will not send the burst length in the masked-write datagram. Prior to sending the masked-write datagram, the master device will compare the specified burst length with a current copy of the slave device's masked-write burst length in the shadow register. If the value of the slave device's masked-write burst length does not match the specified burst length, the master device will first set (or update) the masked-write burst length on the slave device in order to match the specified burst length. The master device may use a register-write

access command or any other type of write access command (as preferred) to perform the slave device masked-write burst length change before sending the masked-write datagram. The masked-write burst length is only to be changed when a burst length mismatch is detected prior to datagram transmission. The master device may further update the shadow register with the updated masked-write burst length. Accordingly, when the 1-byte MSB of the 16-bit address matches the base address of the slave device and the specified burst length matches the masked-write burst length of the slave device, the master device may send the masked-write datagram to the slave device.

[0099] FIG. 9 is a diagram illustrating an example packet structure 900 supporting a 16-bit address space and N-pairs of mask-and-data bytes. Referring to FIG. 9, a datagram header 902 may include a slave address having four bits (SA(4)), a command frame having eight bits (CMD(8)), and a parity bit P. A master device may prepare a datagram based on a 16-bit address 904 having 1-byte most significant byte (MSB) 906 and 1-byte least significant byte (LSB) 908. The master device will only send the 1-byte LSB 908 in the masked-write datagram. Thus, the 1-byte MSB 906 will not be transmitted as part of the masked-write datagram. Moreover, the master device may prepare the datagram based on a specified mask-and-data pair burst length 910. The mask-and-data pair burst length 910 indicates the number of mask-and-data byte pairs (e.g., Mask+Data pair #0, Mask+Data pair #1, …, Mask+Data pair #N) in the payload. The mask-and-data pair burst length 910 will not be transmitted as part of the masked-write datagram.

[0100] FIG. 10 is a diagram illustrating an example datagram 1000 in a transmit buffer. In the page addressing scheme, a slave device 1022 has a 1-byte base-address-register 1024. A master device 1012 maintains a copy of the slave device's base-address-register 1024 at the master device in a shadow register 1014. The master device 1012 prepares a datagram 1000 based on a 16-bit address 1004 having 1-byte most significant byte (MSB) 1006 and 1-byte least significant byte (LSB) 1008, but will only send the 1-byte LSB 1008 in the masked-write datagram 1000. Prior to sending the masked-write datagram 1000, the master device 1012 will compare the 1-byte MSB 1006 to a current copy of the slave device's base-address-register 1024 in the shadow register 1014. If the slave device's base-address-register 1024 does not match the 1-byte MSB 1006, the master device 1012 will first set (or update) the base address 1024 on the slave device 1022 (to change the page) in order to match the 1-byte MSB 1006.

The master device 1012 may further update the shadow register 1014 with the updated slave device base address 1024.

[0101] In the burst writing scheme with mask-and-data byte pair, the slave device 1022 has a 1-byte masked-write burst length register 1026. The master device 1012 maintains a copy of the slave device's masked-write burst length register 1026 at the master device 1012 in a shadow register 1016. The master device 1012 prepares the datagram 1000 based on a mask-and-data pair burst length 1010 (e.g., number of mask-and-data byte pairs) specified, but will not send the burst length 1010 in the masked-write datagram 1000. Prior to sending the masked-write datagram 1000, the master device will compare the specified burst length 1010 with a current copy of the slave device's masked-write burst length 1026 in the shadow register 1016. If the slave device's masked-write burst length 1026 does not match the specified burst length 1010, the master device 1012 will first set (or update) the masked-write burst length 1026 on the slave device 1022 in order to match the specified burst length 1010. The master device 1012 may further update the shadow register 1016 with the updated masked-write burst length 1026. Accordingly, when the 1-byte MSB 1006 matches the base address 1024 of the slave device 1022 and the specified burst length 1010 matches the masked-write burst length 1026 of the slave device 1022, the master device 1012 may send the masked-write datagram 1000 to the slave device 1022.

[0102] FIG. 11 is a diagram illustrating an example operation for a datagram 1100 in a transmit buffer. In the page addressing scheme, a slave device 1122 has a 1-byte base-address-register 1124. A master device 1112 maintains a copy of the slave device's base-address-register 1124 at the master device in a shadow register 1114. The master device 1112 prepares a datagram 1100 based on a 16-bit address 1104 having 1-byte most significant byte (MSB) 1106 and 1-byte least significant byte (LSB) 1108, but will only send the 1-byte LSB 1108 in the masked-write datagram 1100. Prior to sending the masked-write datagram 1100, the master device 1112 will compare 1130 the 1-byte MSB 1106 to a current copy of the slave device's base-address-register 1124 in the shadow register 1114. If the slave device's base-address-register 1124 in the shadow register 1114 is equal to the 1-byte MSB 1006 (see 1132), then there is no need to set (or update) the base address 1124 on the slave device 1122. If the slave device's base-address-register 1124 does not equal the 1-byte MSB 1106 (see 1134), the master device 1112 will set (or update) the base address 1124 on the slave device 1222 (to

change the page) in order to match the 1-byte MSB 1106. The master device 1112 may use a register-write access command 1136 or any other type of write access command (as preferred) to perform the slave device base address change before sending the masked-write datagram. The page is only to be changed when a page mismatch is detected prior to datagram transmission. The master device 1112 may further update the shadow register 1114 with the updated slave device base address 1124.

[0103] In the burst writing scheme with mask-and-data byte pair, the slave device 1122 has a 1-byte masked-write burst length register 1126. The master device 1112 maintains a copy of the slave device's masked-write burst length register 1126 at the master device 1112 in a shadow register 1116. The master device 1112 prepares the datagram 1100 based on a mask-and-data pair burst length 1110 (e.g., number of mask-and-data byte pairs) specified, but will not send the burst length 1110 in the masked-write datagram 1100. Prior to sending the masked-write datagram 1100, the master device will compare 1140 the specified burst length 1110 with a current copy of the slave device's masked-write burst length 1126 in the shadow register 1116. If the slave device's masked-write burst length 1126 in the shadow register 1116 is equal to the specified burst length 1110 (see 1142), then there is no need to set (or update) the masked-write burst length 1126 on the slave device 1122. If the slave device's masked-write burst length 1126 does not equal the specified burst length 1110 (see 1144), the master device 1112 will set (or update) the masked-write burst length 1126 on the slave device 1122 in order to match the specified burst length 1110. The master device may use a register-write access command 1146 or any other type of write access command (as preferred) to perform the slave device masked-write burst length change before sending the masked-write datagram. The masked-write burst length is only to be changed when a burst length mismatch is detected prior to datagram transmission. The master device 1112 may further update the shadow register 1116 with the updated masked-write burst length 1126. Accordingly, when the 1-byte MSB 1106 matches the base address 1124 of the slave device 1122 and the specified burst length 1110 matches the masked-write burst length 1126 of the slave device 1122, the master device 1112 may send the masked-write datagram 1100 to the slave device 1122.

[0104] In an aspect of the disclosure, a masked-write operation may be performed using an extended register write datagram and/or an extended register long write datagram. Payloads of the datagrams may be used to transmit a number of mask-and-

data pairs. Such an operation may be hereinafter referred to as a custom masked-write operation. In an aspect, a normal write datagram may be distinguished from the custom masked-write datagram by defining two bits in a configuration register, as will be described below. Utilization of a write datagram payload for a masked-write purpose is illustrated in FIGs. 12 and 13.

[0105] FIG. 12 is a diagram illustrating an example packet structure 1200 of an extended register write command 1202 supporting a masked-write operation. FIG. 13 is a diagram illustrating an example packet structure 1300 of an extended register long write command 1302 supporting a masked-write operation.

[0106] Referring to FIGs. 12 and 13, a custom masked-write operation may make use of the extended register write command 1202 and the extended register long write command 1302 with a condition that the number of bytes in a payload section is specified as an even number to allow transmission of an integer number of mask-and-data byte pairs. A first mask byte (i.e., Mask-0) is located at a first even location (0-th byte) following an address byte in the payload. The first mask byte is followed by a corresponding first data byte (i.e., Data-0), which is located at a first odd location after the address byte. Thus, following the address byte, mask bytes (e.g., Mask-0, Mask-1, Mask-2, etc.) may occupy even locations, while data bytes (e.g., Data-0, Data-1, Data-2, etc.) may occupy odd locations, in the payload.

[0107] A maximum number of mask-and-data byte pairs that can be transmitted in a write command is dependent on a maximum number of bytes allowed in a payload. For example, in the extended register write command 1202, a maximum allowed byte count in the payload 1204 is 16 bytes (128 bits). Thus, the extended register write command 1202 may support the transmission of a maximum of 8 mask-and-data byte pairs in one datagram. As shown in FIG. 12, the payload 1204 may include 1 to 8 mask-and-data byte pairs (e.g., Mask+Data pair #0, Mask+Data pair #1, …, Mask+Data pair #7). In another example, in the extended register long write command 1302, a maximum allowed byte count in the payload 1304 is 8 bytes (64 bits). Thus, the extended register long write command 1302 may support the transmission of a maximum of 4 mask-and-data byte pairs in one datagram. As shown in FIG. 13, the payload 1304 may include 1 to 4 mask-and-data byte pairs (e.g., Mask+Data pair #0, Mask+Data pair #1, …, Mask+Data pair #3).

[0108] In an aspect of the disclosure, an 8-bit configuration register may be utilized to provide a control function interface to facilitate the enabling and disabling of the masked-write operation using the extended register write command and/or the extended register long write command. The configuration register may be located within a user defined register space: 0x01 to 0x1C in hexadecimal. For example, register location 0x18 may be used as the configuration register. However, in alternative aspects, it is contemplated that the configuration register may be at any location within an entire register space.

[0109] FIG. 14 is a diagram illustrating an example bit structure 1400 of a configuration register 1402. As shown, the configuration register 1402 includes eight configuration register bits D7, D6, D5, D4, D3, D2, and D1. In an aspect of the disclosure, a third register bit D5 1404 and a fourth register bit D4 1406 may be used to distinguish between when the extended register write and extended register long write commands are to be used in a normal manner and when the extended register write and extended register long write commands are to be used for a custom masked-write operation.

[0110] For example, when the third register bit D5 1404 is set to a value of 1, then the custom masked-write operation is enabled, and the extended register write command or the extended register long write command is to be used for a masked-write operation. However, when the third register bit D5 1404 is set to a value of 0, then the custom masked-write operation is disabled, and both the extended register write command and the extended register long write command are to be used in the normal manner. Moreover, when the fourth register bit D4 1406 is set to a value of 1, then the extended register long write command (e.g., extended register long write command 1302) is to be specifically used for the custom masked write operation if the third register bit D5 1404 is set to the value of 1. When the fourth register bit D4 1406 is set to a value of 0, then the extended register write command (e.g., extended register write command 1202) is to be specifically used for the custom masked-write operation if the third register bit D5 1404 is set to the value of 1.

[0111] FIG. 15 is a diagram of an RFFE register space 1500. The RFFE register space 1500 may extend from register 0x0000 to register 0xFFFF in hexadecimal.

[0112] An association of commands in terms of register space accessibility is shown in FIG. 15. The reach of an extended register operation may be limited to the

space between the 0x00 register and the 0xFF register. However, a complex RFFE slave may contain multiple pages (each having 0x00 to 0xFF 1-byte locations) within the 64K register space, and therefore, enable extended register operation to access the entire 64K register space and reduce bus latency. To achieve this, the 64K register space may be segmented into 256 pages (pages 0x00 to 0xFF), each containing 256 register locations. An 8-bit register address in a datagram combined with a page address allows any register access within the 64K space. The page address may be stored at a known register location and may be combined as an address-MSB with the datagram-supplied 8-bit register address (address-LSB). This may be the basis for page segmented access for an extended register operation.

[0113] FIG. 16 is a diagram of an RFFE register space 1600 having a configuration register and a page-address register. To facilitate the enabling and disabling of various features, an 8-bit configuration register may be used. The configuration register and a page-address register may use two specific registers in the register space that are register-mode accessible. For example, as shown in FIG. 16, the configuration register may be defined at location 0x18 and the page-address register may be defined at location 0x19 in the register space. Both the 0x18 and 0x19 locations are in a user defined space.

[0114] FIG. 17 illustrates a table 1700 defining another example bit structure of a configuration register and a diagram 1750 depicting a function of the configuration register bits. A configuration register containing bit locations D7 to D0 may be defined at register location 0x18. Referring to table 1700 and diagram 1750, page segmented access (PSA) may be enabled or disabled by enabling (e.g., setting to "1") or disabling (e.g., setting to "0") the configuration bit at bit location D2. A double data rate (DDR) mode may be enabled or disabled by enabling or disabling the configuration bit at bit location D1. Additionally, custom masked-write (CMW) may be enabled or disabled by enabling or disabling the configuration bit at bit location D0. For D0, D1, and D2, a configuration bit value of "1" implies that a corresponding function is enabled while a configuration bit value of "0" implies that the corresponding function is disabled.

[0115] FIG. 18 is a diagram 1800 illustrating page segmented access. Standard extended register operations are based on an 8-bit register address. This may limit the applicability of these modes of register access to the first 256 locations of a register space (0x00 to 0xFF). Accordingly, page segmented access (PSA) for an extended

register operation may be an alternative to standard extended register long operations in terms of accessing an entire 64K register space while using only an 8-bit register address in the datagram. Because only an 8-bit register address is used, page segmented access also allows for a maximum payload of 16 bytes per datagram, which is more efficient than the conventional extended register long operations using a 16-bit address and having a maximum payload of 8 bytes per datagram.

[0116] 64K register space access may be enabled for an extended register mode by using a page segmented address register to serve as a register address-MSB location. From a chip level design point of view, PSA operation may be orthogonal to a masked-write operation and a double data rate (DDR) mode operation. Page segmented access (PSA) for the extended register mode may be enabled using a 1-byte register holding the register address-MSB and a single configuration bit included in the configuration register.

[0117] PSA for extended register operations may be applicable to both read and write operations. PSA may use a value located at register location 0x19 as the register address-MSB and concatenate the register address-MSB with an 8-bit address (register address-LSB) supplied in the extended register operation datagram. A single bit in the configuration register may enable/disable PSA.

[0118] Page segmented access using the content of register location 0x19 and an address-LSB retrieved from an extended register operation datagram is shown in FIG. 18. The page address register 1802 at register location 0x19 may contain an 8-bit MSB value for the register address in the 0x0000 to 0xFFFF register space. The value from the register location 0x19 may be used as an address-MSB and combined with the 8-bit address 1804 (address-LSB) received from the extended register operation datagram. Accordingly, the entire 64K register space may be accessed using only an 8-bit register address 1804 in the extended register operation datagram. The value at register location 0x19 has no effect on the extended register operation if the page segment access (PSA) mode is disabled.

[0119] In an aspect of the disclosure, page segmented access (PSA) for extended register operations allows full access to an entire 16-bit address space of any RFFE device. Enabling this feature provides a number of advantages over extended register long-based operations. For example, with only an 8-bit address in the datagram, an entire 64K register space becomes available. Also, since the extended register

command can have up to 16 bytes of payload in contrast to the extended register long command, which can only have up to 8 bytes of payload, PSA provides improved throughput and reduces latency.

[0120] PSA may be enabled or disabled by enabling (e.g., setting to "1") or disabling (e.g., setting to "0") a single configuration bit (e.g., configuration bit at bit location D2) within a configuration register located at register location 0x18. When enabled, an 8-bit page address stored at register location 0x19 may be serve as a register address-MSB and may be attached to an 8-bit address (serving as an register address-LSB) supplied within an extended register datagram.

[0121] Referring back to FIG. 12, the extended register write command 1202 supporting a masked-write operation may be limited to the first 256 locations (register locations 0x00 to 0xFF) of the register space if page segmented access (PSA) is not enabled. However, when PSA is enabled, the extended register write command 1202 supporting the masked-write operation may have full access to the entire 64K register space. As described above, full access to the entire 64K register space is facilitated by using an 8-bit page address stored at register location 0x19 as a register address-MSB, and attaching the register address-MSB to an 8-bit address (used as a register address-LSB) supplied within the extended register write command 1202.

Example of Hardware Implementation

[0122] FIG. 19 is a conceptual diagram illustrating a simplified example of a hardware implementation for an apparatus 1900 employing a processing circuit 1902 that may be configured to perform one or more functions disclosed herein. In accordance with various aspects of the disclosure, an element, or any portion of an element, or any combination of elements as disclosed herein may be implemented using the processing circuit 1902. The processing circuit 1902 may include one or more processors 1904 that are controlled by some combination of hardware and software modules. Examples of processors 1904 include microprocessors, microcontrollers, digital signal processors (DSPs), ASICs, field programmable gate arrays (FPGAs), programmable logic devices (PLDs), state machines, sequencers, gated logic, discrete hardware circuits, and other suitable hardware configured to perform the various functionality described throughout this disclosure. The one or more processors 1904 may include specialized processors that perform specific functions, and that may be

configured, augmented or controlled by one of the software modules 1916. The one or more processors 1904 may be configured through a combination of software modules 1916 loaded during initialization, and further configured by loading or unloading one or more software modules 1916 during operation.

[0123] In the illustrated example, the processing circuit 1902 may be implemented with a bus architecture, represented generally by the bus 1910. The bus 1910 may include any number of interconnecting buses and bridges depending on the specific application of the processing circuit 1902 and the overall design constraints. The bus 1910 links together various circuits including the one or more processors 1904, and storage 1906. Storage 1906 may include memory devices and mass storage devices, and may be referred to herein as computer-readable media and/or processor-readable media. The bus 1910 may also link various other circuits such as timing sources, timers, peripherals, voltage regulators, and power management circuits. A bus interface 1908 may provide an interface between the bus 1910 and one or more line interface circuits 1912. A line interface circuit 1912 may be provided for each networking technology supported by the processing circuit. In some instances, multiple networking technologies may share some or all of the circuitry or processing modules found in a line interface circuit 1912. Each line interface circuit 1912 provides a means for communicating with various other apparatus over a transmission medium. Depending upon the nature of the apparatus 1900, a user interface 1918 (e.g., keypad, display, speaker, microphone, joystick) may also be provided, and may be communicatively coupled to the bus 1910 directly or through the bus interface 1908.

[0124] A processor 1904 may be responsible for managing the bus 1910 and for general processing that may include the execution of software stored in a computer-readable medium that may include the storage 1906. In this respect, the processing circuit 1902, including the processor 1904, may be used to implement any of the methods, functions and techniques disclosed herein. The storage 1906 may be used for storing data that is manipulated by the processor 1904 when executing software, and the software may be configured to implement any one of the methods disclosed herein.

[0125] One or more processors 1904 in the processing circuit 1902 may execute software. Software shall be construed broadly to mean instructions, instruction sets, code, code segments, program code, programs, subprograms, software modules, applications, software applications, software packages, routines, subroutines, objects,

executables, threads of execution, procedures, functions, algorithms, etc., whether referred to as software, firmware, middleware, microcode, hardware description language, or otherwise. The software may reside in computer-readable form in the storage 1906 or in an external computer readable medium. The external computer-readable medium and/or storage 1906 may include a non-transitory computer-readable medium. A non-transitory computer-readable medium includes, by way of example, a magnetic storage device (e.g., hard disk, floppy disk, magnetic strip), an optical disk (e.g., a compact disc (CD) or a digital versatile disc (DVD)), a smart card, a flash memory device (e.g., a "flash drive," a card, a stick, or a key drive), a random access memory (RAM), a read only memory (ROM), a programmable ROM (PROM), an erasable PROM (EPROM), an electrically erasable PROM (EEPROM), a register, a removable disk, and any other suitable medium for storing software and/or instructions that may be accessed and read by a computer. The computer-readable medium and/or storage 1906 may also include, by way of example, a carrier wave, a transmission line, and any other suitable medium for transmitting software and/or instructions that may be accessed and read by a computer. Computer-readable medium and/or the storage 1906 may reside in the processing circuit 1902, in the processor 1904, external to the processing circuit 1902, or be distributed across multiple entities including the processing circuit 1902. The computer-readable medium and/or storage 1906 may be embodied in a computer program product. By way of example, a computer program product may include a computer-readable medium in packaging materials. Those skilled in the art will recognize how best to implement the described functionality presented throughout this disclosure depending on the particular application and the overall design constraints imposed on the overall system.

[0126] The storage 1906 may maintain software maintained and/or organized in loadable code segments, modules, applications, programs, etc., which may be referred to herein as software modules 1916. Each of the software modules 1916 may include instructions and data that, when installed or loaded on the processing circuit 1902 and executed by the one or more processors 1904, contribute to a run-time image 1914 that controls the operation of the one or more processors 1904. When executed, certain instructions may cause the processing circuit 1902 to perform functions in accordance with certain methods, algorithms and processes described herein.

[0127] Some of the software modules 1916 may be loaded during initialization of the processing circuit 1902, and these software modules 1916 may configure the processing circuit 1902 to enable performance of the various functions disclosed herein. For example, some software modules 1916 may configure internal devices and/or logic circuits 1922 of the processor 1904, and may manage access to external devices such as the line interface circuit 1912, the bus interface 1908, the user interface 1918, timers, mathematical coprocessors, and so on. The software modules 1916 may include a control program and/or an operating system that interacts with interrupt handlers and device drivers, and that controls access to various resources provided by the processing circuit 1902. The resources may include memory, processing time, access to the line interface circuit 1912, the user interface 1918, and so on.

[0128] One or more processors 1904 of the processing circuit 1902 may be multifunctional, whereby some of the software modules 1916 are loaded and configured to perform different functions or different instances of the same function. The one or more processors 1904 may additionally be adapted to manage background tasks initiated in response to inputs from the user interface 1918, the line interface circuit 1912, and device drivers, for example. To support the performance of multiple functions, the one or more processors 1904 may be configured to provide a multitasking environment, whereby each of a plurality of functions is implemented as a set of tasks serviced by the one or more processors 1904 as needed or desired. In one example, the multitasking environment may be implemented using a timesharing program 1920 that passes control of a processor 1904 between different tasks, whereby each task returns control of the one or more processors 1904 to the timesharing program 1920 upon completion of any outstanding operations and/or in response to an input such as an interrupt. When a task has control of the one or more processors 1904, the processing circuit is effectively specialized for the purposes addressed by the function associated with the controlling task. The timesharing program 1920 may include an operating system, a main loop that transfers control on a round-robin basis, a function that allocates control of the one or more processors 1904 in accordance with a prioritization of the functions, and/or an interrupt driven main loop that responds to external events by providing control of the one or more processors 1904 to a handling function.

Exemplary Methods and Device for Sending a Datagram from a Transmitter to a Receiver

[0129] FIG. 20 is a flow chart 2000 of a method for sending data to a receiver via a bus interface. The method may be performed at a device operating as a transmitter (e.g., bus master).

[0130] The device may generate a datagram based on a 16-bit address and a mask-and-data pair burst length 2002. The 16-bit address includes a most significant byte (MSB) and a least significant byte (LSB).

[0131] The device then compares the MSB to a receiver base address (segment or value) maintained in a shadow register 2004. This comparison includes detecting whether the MSB is equal to the receiver base address maintained in the shadow register. If the MSB is not equal to the receiver base address maintained in the shadow register, then the device sets a base address at the receiver to be equal to the MSB. The device may set the base address at the receiver by sending a write access command to the receiver prior to sending the datagram. The device further updates the receiver base address maintained in the shadow register to the MSB.

[0132] The device may further compare the mask-and-data pair burst length to a receiver masked-write burst length (segment or value) maintained in the shadow register 2006. This comparison includes detecting whether the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register. If the mask-and-data pair burst length is not equal to the receiver masked-write burst length maintained in the shadow register, then the device sets a masked-write burst length at the receiver to be equal to the mask-and-data pair burst length. The device may set the masked-write burst length at the receiver by sending a write access command to the receiver prior to sending the datagram. The device further updates the receiver masked-write burst length maintained in the shadow register to the mask-and-data pair burst length.

[0133] Finally, when the MSB is equal to the receiver base address maintained in the shadow register and the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register, the device sends the datagram via the bus interface to the receiver 2008. The datagram sent to the receiver does not include the MSB and the mask-and-data pair burst length.

[0134] FIG. 21 is a flow chart 2100 of another method for sending data to a receiver via a bus interface. The method may be performed at a device operating as a transmitter (e.g., bus master).

[0135] The device may generate a command field in a datagram to be transmitted through the bus interface to the receiver 2102. The command field may indicate a type of masked write command the datagram is, such as whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0136] Alternatively, the device may generate a command field and a mode field in the datagram 2104. As such, the command field may indicate that the datagram is a masked write command and the mode field may indicate the masked write command type, such as whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0137] The device may generate a mask field in the datagram 2106. The mask field identifies at least one bit to be changed in a radio frequency front end (RFFE) register. The mask field also indicates a remaining set of bits to be left unchanged in the RFFE register. The device may also generate a data field in the datagram 2108. The data field provides a value of the at least one bit to be changed in the RFFE register. In an aspect of the disclosure, the mask field is a bit index field identifying a bit position to be changed in the RFFE register, and the data field is a bit value field providing a bit value for the bit position identified in the bit index field.

[0138] Finally, the device may transmit the datagram through the interface, wherein the datagram is addressed to the RFFE register of the receiver 2110.

[0139] FIG. 22 is a flow chart 2200 of a further method for sending data to a receiver via a bus interface. The method may be performed at a device operating as a transmitter (e.g., bus master).

[0140] The device may set a configuration register to indicate whether a masked-write operation is enabled with respect to a datagram to be transmitted to the receiver 2202. The configuration register may include eight register bits. Accordingly, the device may set a third register bit (e.g., register bit D5 1404) of the eight register bits to a value of 1 to indicate that the masked-write operation is enabled. Alternatively, the

device may set the third register bit (e.g., register bit D5 1404) to a value of 0 to indicate that the masked-write operation is disabled.

[0141] In an aspect, the datagram may be either an extended register write command or an extended register long write command. Therefore, when the masked-write operation is enabled, the device may set a fourth register bit (e.g., register bit D4 1406) in the configuration register to a value of 1 to indicate that the masked-write operation is enabled with respect to the extended register long write command. When the masked-write operation is enabled, the device may also set the fourth register bit (e.g., register bit D4 1406) to a value of 0 to indicate that the masked-write operation is enabled with respect to the extended register write command.

[0142] The device may generate a command field in the datagram 2204. The command field may indicate whether the datagram is the extended register write command or the extended register long write command.

[0143] The device may also generate a payload field in the datagram 2206. The payload field may include a number of mask-and-data pairs when the masked-write operation is enabled. Each mask-and-data pair may include a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register and a data field providing a value of the at least one bit to be changed in the RFFE register.

[0144] The device transmits the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver 2208.

[0145] FIG. 23 is a flow chart 2300 of another method for sending data to a receiver via a bus interface. The method may be performed at a device operating as a transmitter (e.g., bus master).

[0146] The device may enable a masked-write operation by setting a single bit within a configuration register at the receiver to a first value 2302. Additionally and/or alternatively, the device may disable the masked-write operation by setting the single bit within the configuration register at the receiver to a second value. For example, the masked-write operation may be enabled by performing a write operation to the receiver's configuration register (e.g., register at location 0x18) in order to set a bit D0 to a value of "1". In another example, the masked-write operation may be disabled by performing a write operation to the receiver's configuration register (e.g., register at location 0x18) in order to set the bit D0 to a value of "0".

[0147] The device may generate a datagram to be transmitted to the receiver via the bus interface 2304. The datagram includes or provides an address value (e.g., register address 1804 in FIG. 18). The datagram may be an extended register write datagram or an extended register write long datagram.

[0148] The device may also generate a payload field in the datagram 2306. The payload field includes a number of mask-and-data pairs when the masked-write operation is enabled. Each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register and a data field providing a value of the at least one bit to be changed in the RFFE register.

[0149] The device may enable a page segmented access operation by setting another single bit within the configuration register at the receiver to a first value 2308. For example, the page segmented access operation may be enabled by performing a write operation to the receiver's configuration register (e.g., register at location 0x18) in order to set a bit D2 to a value of "1". An address of the RFFE register is a combination of an address value located at a page address register (e.g., register location 0x19) at the receiver and the address value provided by the datagram when the page segmented access operation is enabled.

[0150] The device may disable the page segmented access operation by setting the other single bit within the configuration register at the receiver to a second value 2310. For example, the page segmented access operation may be disabled by performing a write operation to the receiver's configuration register (e.g., register at location 0x18) in order to set a bit D2 to a value of "0". The address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled.

[0151] The device may transmit the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver 2312.

[0152] FIG. 24 is a diagram illustrating a simplified example of a hardware implementation for a transmitting apparatus 2400 employing a processing circuit 2402. Examples of operations performed by the transmitting apparatus 2400 include the operations described above with respect to the flow charts of FIGs. 20 to 23. The processing circuit typically has a processor 2416 that may include one or more of a microprocessor, microcontroller, digital signal processor, a sequencer and a state machine. The processing circuit 2402 may be implemented with a bus architecture, represented generally by the bus 2420. The bus 2420 may include any number of

interconnecting buses and bridges depending on the specific application of the processing circuit 2402 and the overall design constraints. The bus 2420 links together various circuits including one or more processors and/or hardware modules, represented by the processor 2416, the modules or circuits 2404, 2406, 2408, 2410, bus interface circuits 2412 configurable to support communication over connectors or wires 2414 and the computer-readable storage medium 2418. The bus 2420 may also link various other circuits such as timing sources, peripherals, voltage regulators, and power management circuits, which are well known in the art, and therefore, will not be described any further.

[0153] The processor 2416 is responsible for general processing, including the execution of software/instructions stored on the computer-readable storage medium 2418. The software/instructions, when executed by the processor 2416, causes the processing circuit 2402 to perform the various functions described supra for any particular apparatus. The computer-readable storage medium may also be used for storing data that is manipulated by the processor 2416 when executing software, including data decoded from symbols transmitted over the connectors or wires 2414, which may be configured as data lanes and clock lanes. The processing circuit 2402 further includes at least one of the modules/circuits 2404, 2406, 2408, and 2410. The modules/circuits 2404, 2406, 2408, and 2410 may be software modules running in the processor 2416, resident/stored in the computer-readable storage medium 2418, one or more hardware modules coupled to the processor 2416, or some combination thereof. The modules/circuits 2404, 2406, 2408, and/or 2410 may include microcontroller instructions, state machine configuration parameters, or some combination thereof.

[0154] In one configuration, the apparatus 2400 for communication includes a datagram generating/sending module/circuit 2404 that is configured to, generate a datagram based on a 16-bit address and a mask-and-data pair burst length, the 16-bit address including a most significant byte (MSB) and a least significant byte (LSB), and via the bus interface module/circuit 2412, send the datagram to a receiver when: the MSB is equal to a receiver base address maintained in a shadow register, and the mask-and-data pair burst length is equal to a receiver masked-write burst length maintained in the shadow register. The apparatus 2400 further includes an address comparing module/circuit 2406 that is configured to, compare the MSB to the receiver base address maintained in the shadow register. The apparatus 2400 further includes a burst

length comparing module/circuit 2408 that is configured to, compare the mask-and-data pair burst length to the receiver masked-write burst length maintained in the shadow register. The apparatus 2400 further includes a register setting module/circuit 2410 that is configured to, set a configuration register to indicate whether a masked-write operation is enabled with respect to a datagram to be transmitted through the bus interface module/circuit 2412 to the receiver.

[0155] In another configuration, the datagram generating/sending module/circuit 2404 is configured to, generate a command field in a datagram to be transmitted through the bus interface module/circuit 2412 to the receiver, generate a mode field in the datagram, generate a payload field in the datagram, generate a mask field in the datagram, generate a data field in the datagram, and transmit the datagram through the bus interface module/circuit 2412, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the receiver.

[0156] In a further configuration, the datagram generating/sending module/circuit 2404 is configured to enable a masked-write operation by setting a single bit within a configuration register at the receiver to a first value, disable the masked-write operation by setting the single bit within the configuration register at the receiver to a second value, generate a datagram to be transmitted to the receiver via the bus interface, the datagram providing an address value, generating a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register and a data field providing a value of the at least one bit to be changed in the RFFE register, enable a page segmented access operation by setting another single bit within the configuration register at the receiver to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the datagram when the page segmented access operation is enabled, disable the page segmented access operation by setting the other single bit within the configuration register at the receiver to a second value, wherein the address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled, and transmit the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver.

Exemplary Method and Device for Receiving a Datagram at a Receiver from a Transmitter

[0157] FIG. 25 is a flow chart 2500 of a method for receiving data from a transmitter via a bus interface. The method may be performed at a device operating as a receiver (e.g., bus slave).

[0158] The device may receive a datagram through the bus interface from the transmitter 2502. The datagram is addressed to a radio frequency front end (RFFE) register of the receiver.

[0159] The device may read a command field in the datagram 2504. The command field may indicate a type of masked write command the datagram is, such as whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0160] Alternatively, the device may read a command field and a mode field in the datagram 2506. As such, the command field may indicate that the datagram is a masked write command and the mode field may indicate the masked write command type, such as whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

[0161] The device may read a mask field in the datagram 2508. The mask field identifies at least one bit to be changed in the RFFE register. The mask field also indicates a remaining set of bits to be left unchanged in the RFFE register. The device may also read a data field in the datagram 2510. The data field provides a value of the at least one bit to be changed in the RFFE register. In an aspect of the disclosure, the mask field is a bit index field identifying a bit position to be changed in the RFFE register, and the data field is a bit value field providing a bit value for the bit position identified in the bit index field.

[0162] Finally, the device may change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field 2512.

[0163] FIG. 26 is a flow chart 2600 of another method for receiving data from a transmitter via a bus interface. The method may be performed at a device operating as a receiver (e.g., bus slave).

[0164] The device may read a configuration register to detect whether a masked-write operation is enabled with respect to a datagram to be received from the transmitter 2602. The configuration register includes eight register bits. Accordingly, the device may detect that the masked-write operation is enabled when a third register bit (e.g., register bit D5 1404) of the eight register bits is set to a value of 1. The device may also detect that the masked-write operation is disabled when the third register bit (e.g., register bit D5 1404) is set to a value of 0.

[0165] In an aspect, the datagram may be either an extended register write command or an extended register long write command. Therefore, when the masked-write operation is enabled, the device may detect that the masked-write operation is enabled with respect to the extended register long write command when a fourth register bit (e.g., register bit D4 1406) in the configuration register is set to a value of 1. When the masked-write operation is enabled, the device may also detect that the masked-write operation is enabled with respect to the extended register write command when the fourth register bit (e.g., register bit D4 1406) is set to a value of 0.

[0166] The device may receive the datagram through the bus interface from the transmitter 2604, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the receiver.

[0167] The device may read a command field in the datagram 2606. The command field indicates whether the datagram is the extended register write command or the extended register long write command.

[0168] The device may also read a payload field in the datagram 2608. The payload field includes a number of mask-and-data pairs when the masked-write operation is enabled. Each mask-and-data pair includes a mask field identifying at least one bit to be changed in the RFFE register and a data field providing a value of the at least one bit to be changed in the RFFE register. Finally, the device may change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair 2610.

[0169] FIG. 27 is a flow chart 2700 of a further method for receiving data from a transmitter via a bus interface. The method may be performed at a device operating as a receiver (e.g., bus slave).

[0170] The device may receive a first datagram from the transmitter for setting a single bit within a configuration register at the receiver 2702. The device may detect

that a masked-write operation is enabled when the single bit within the configuration register is set to a first value. Alternatively, the device may detect that the masked-write operation is disabled when the single bit within the configuration register at the receiver is set to a second value 2704. For example, the device may detect that the masked-write operation is enabled when a bit D0 in the receiver's configuration register (e.g., register at location 0x18) has a value of "1" as set by the transmitter via a write operation. In another example, the device may detect that the masked-write operation is disabled when the bit D0 in the receiver's configuration register (e.g., register at location 0x18) has a value of "0" as set by the transmitter via a write operation.

[0171] The device may receive a second datagram from the transmitter 2706. The second datagram includes or provides an address value (e.g., register address 1804 in FIG. 18). The second datagram may be an extended register write datagram or an extended register write long datagram.

[0172] The device may reading a payload field in the second datagram 2708. The payload field includes a number of mask-and-data pairs when the masked-write operation is enabled. Each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register of the receiver and a data field providing a value of the at least one bit to be changed in the RFFE register.

[0173] The device may receive a third datagram from the transmitter for setting another single bit within the configuration register at the receiver 2710. The device may detect that a page segmented access operation is enabled when the other single bit within the configuration register at the receiver is set to a first value 2712. For example, the device may detect that the page segmented access operation is enabled when a bit D2 in the receiver's configuration register (e.g., register at location 0x18) has a value of "1" as set by the transmitter via a write operation. An address of the RFFE register is a combination of an address value located at a page address register (e.g., register location 0x19) at the receiver and the address value provided by the datagram when the page segmented access operation is enabled.

[0174] The device may detect that the page segmented access operation is disabled when the other single bit within the configuration register at the receiver is set to a second value 2714. For example, the device may detect that the page segmented access operation is disabled when the bit D2 in the receiver's configuration register (e.g., register at location 0x18) has a value of "0" as set by the transmitter via a write

operation. The address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled.

[0175] The device may change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair 2716.

[0176] FIG. 28 is a diagram illustrating a simplified example of a hardware implementation for a receiving apparatus 2800 employing a processing circuit 2802. Examples of operations performed by the receiving apparatus 2800 include the operations described above with respect to the flow chart of FIGs. 25 to 27. The processing circuit typically has a processor 2816 that may include one or more of a microprocessor, microcontroller, digital signal processor, a sequencer and a state machine. The processing circuit 2802 may be implemented with a bus architecture, represented generally by the bus 2820. The bus 2820 may include any number of interconnecting buses and bridges depending on the specific application of the processing circuit 2802 and the overall design constraints. The bus 2820 links together various circuits including one or more processors and/or hardware modules, represented by the processor 2816, the modules or circuits 2804, 2806, 2808, 2810, bus interface circuits 2812 configurable to support communication over connectors or wires 2814 and the computer-readable storage medium 2818. The bus 2820 may also link various other circuits such as timing sources, peripherals, voltage regulators, and power management circuits, which are well known in the art, and therefore, will not be described any further.

[0177] The processor 2816 is responsible for general processing, including the execution of software/instructions stored on the computer-readable storage medium 2818. The software/instructions, when executed by the processor 2816, causes the processing circuit 2802 to perform the various functions described supra for any particular apparatus. The computer-readable storage medium may also be used for storing data that is manipulated by the processor 2816 when executing software, including data decoded from symbols transmitted over the connectors or wires 2814, which may be configured as data lanes and clock lanes. The processing circuit 2802 further includes at least one of the modules/circuits 2804, 2806, 2808, and 2810. The modules/circuits 2804, 2806, 2808, and 2810 may be software modules running in the processor 2816, resident/stored in the computer-readable storage medium 2818, one or

more hardware modules coupled to the processor 2816, or some combination thereof. The modules/circuits 2804, 2806, 2808, and/or 2810 may include microcontroller instructions, state machine configuration parameters, or some combination thereof.

[0178] In one configuration, the apparatus 2800 for communication includes a datagram receiving module/circuit 2804 that is configured to, receive a datagram through the bus interface module/circuit 2812 from a transmitter, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the apparatus 2800. The apparatus 2800 further includes a field reading module/circuit 2806 that is configured to, read a command field in the datagram, read a payload field in the datagram, read a mode field in the datagram, read a mask field in the datagram, and read a data field in the datagram. The apparatus 2800 further includes a bit changing module/circuit 2808 that is configured to, change at least one bit in the RFFE register identified in the mask field according to a value provided in the data field. The apparatus 2800 also includes a register reading module/circuit 2810 that is configured to, read a configuration register to detect whether a masked-write operation is enabled with respect to a datagram to be received from the transmitter.

[0179] In another configuration, the datagram receiving module/circuit 2804 is configured to receive a first datagram from the transmitter for setting a single bit within a configuration register at the receiver, detect that a masked-write operation is enabled when the single bit within the configuration register is set to a first value, detect that the masked-write operation is disabled when the single bit within the configuration register at the receiver is set to a second value, receive a second datagram from the transmitter, the second datagram providing an address value, receive a third datagram from the transmitter for setting another single bit within the configuration register at the receiver, detect that a page segmented access operation is enabled when the other single bit within the configuration register at the receiver is set to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the datagram when the page segmented access operation is enabled, and detect that the page segmented access operation is disabled when the other single bit within the configuration register at the receiver is set to a second value, wherein the address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled,

[0180] In another configuration, the field reading module/circuit 2806 is configured to read a payload field in the second datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register of the receiver and a data field providing a value of the at least one bit to be changed in the RFFE register.

[0181] In another configuration, the field reading module/circuit 2806 is configured to change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair.

[0182] It is understood that the specific order or hierarchy of steps in the processes disclosed is an illustration of exemplary approaches. The specific order or hierarchy of steps in the processes may be rearranged based upon design preferences. The accompanying method claims present elements of the various steps in a sample order, and are not meant to be limited to the specific order or hierarchy presented.

[0183] The previous description is provided to enable any person skilled in the art to practice the various aspects described herein. Various modifications to these aspects will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other aspects. Thus, the claims are not intended to be limited to the aspects shown herein, but is to be accorded the full scope consistent with the language claims, wherein reference to an element in the singular is not intended to mean "one and only one" unless specifically so stated, but rather "one or more." Unless specifically stated otherwise, the term "some" refers to one or more. All structural and functional equivalents to the elements of the various aspects described throughout this disclosure that are known or later come to be known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the claims. Moreover, nothing disclosed herein is intended to be dedicated to the public regardless of whether such disclosure is explicitly recited in the claims. No claim element is to be construed as a means plus function unless the element is expressly recited using the phrase "means for."

CLAIMS

1. A method performed at a transmitter for sending data to a receiver via a bus interface, comprising:

generating a datagram based on a 16-bit address and a mask-and-data pair burst length, the 16-bit address including a most significant byte (MSB) and a least significant byte (LSB);

comparing the MSB to a receiver base address maintained in a shadow register;

comparing the mask-and-data pair burst length to a receiver masked-write burst length maintained in the shadow register; and

sending the datagram to the receiver via the bus interface when:

the MSB is equal to the receiver base address maintained in the shadow register, and

the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register.

2. The method of claim 1, wherein the datagram sent to the receiver does not include the MSB and the mask-and-data pair burst length.

3. The method of claim 1, wherein the comparing the MSB to the receiver base address maintained in the shadow register includes:

detecting whether the MSB is equal to the receiver base address maintained in the shadow register; and

when the MSB is not equal to the receiver base address maintained in the shadow register:

setting a base address at the receiver to be equal to the MSB; and

updating the receiver base address maintained in the shadow register to the MSB.

4. The method of claim 3, wherein the base address at the receiver is set by sending a write access command to the receiver prior to sending the datagram.

5. The method of claim 1, wherein the comparing the mask-and-data pair burst length to the receiver masked-write burst length maintained in the shadow register includes:

detecting whether the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register; and

when the mask-and-data pair burst length is not equal to the receiver masked-write burst length maintained in the shadow register:

setting a masked-write burst length at the receiver to be equal to the mask-and-data pair burst length, and

updating the receiver masked-write burst length maintained in the shadow register to the mask-and-data pair burst length.

6. The method of claim 5, wherein the masked-write burst length at the receiver is set by sending a write access command to the receiver prior to sending the datagram.

7. A transmitter for sending data to a receiver, comprising:

a bus interface; and

a processing circuit configured to:

generate a datagram based on a 16-bit address and a mask-and-data pair burst length, the 16-bit address including a most significant byte (MSB) and a least significant byte (LSB);

compare the MSB to a receiver base address maintained in a shadow register;

compare the mask-and-data pair burst length to a receiver masked-write burst length maintained in the shadow register; and

send the datagram to a receiver via the bus interface when:

the MSB is equal to the receiver base address maintained in the shadow register, and

the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register.

8. The transmitter of claim 7, wherein the datagram sent to the receiver does not include the MSB and the mask-and-data pair burst length.

9. The transmitter of claim 7, wherein the processing circuit is configured to compare the MSB to the receiver base address maintained in the shadow register by:

detecting whether the MSB is equal to the receiver base address maintained in the shadow register; and

when the MSB is not equal to the receiver base address maintained in the shadow register:

setting a base address at the receiver to be equal to the MSB; and

updating the receiver base address maintained in the shadow register to the MSB.

10. The transmitter of claim 9, wherein the processing circuit is configured to set the base address at the receiver by sending a write access command to the receiver prior to sending the datagram.

11. The transmitter of claim 7, wherein the processing circuit is configured to compare the mask-and-data pair burst length to the receiver masked-write burst length maintained in the shadow register by:

detecting whether the mask-and-data pair burst length is equal to the receiver masked-write burst length maintained in the shadow register; and

when the mask-and-data pair burst length is not equal to the receiver masked-write burst length maintained in the shadow register:

setting a masked-write burst length at the receiver to be equal to the mask-and-data pair burst length, and

updating the receiver masked-write burst length maintained in the shadow register to the mask-and-data pair burst length.

12. The transmitter of claim 11, wherein the processing circuit is configured to set the masked-write burst length at the receiver by sending a write access command to the receiver prior to sending the datagram.

13. A method performed at a transmitter for sending data to a receiver, comprising:

generating a mask field in a datagram to be transmitted through an interface to the receiver, the mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register;

generating a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register; and

transmitting the datagram through the interface, wherein the datagram is addressed to the RFFE register of the receiver.

14. The method of claim 13, wherein the mask field further indicates a remaining set of bits to be left unchanged in the RFFE register.

15. The method of claim 13, further comprising generating a command field in the datagram, the command field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

16. The method of claim 13, further comprising:

generating a command field in the datagram, the command field indicating that the datagram is a masked write command; and

generating a mode field in the datagram, the mode field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

17. The method of claim 13, wherein:

the mask field is a bit index field identifying a bit position to be changed in the RFFE register of the receiver; and

the data field is a bit value field providing a bit value for the bit position identified in the bit index field.

18. A transmitter for sending data to a receiver, comprising:

a bus interface; and

a processing circuit configured to:

generate a mask field in a datagram to be transmitted through the bus interface to the receiver, the mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register,

generate a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register, and

transmit the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver.

19. A method performed at a receiver for receiving data from a transmitter, comprising:

receiving a datagram through an interface from the transmitter, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the receiver;

reading a mask field in the datagram, the mask field identifying at least one bit to be changed in the RFFE register;

reading a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register; and

changing the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field.

20. The method of claim 19, wherein the mask field further indicates a remaining set of bits to be left unchanged in the RFFE register.

21. The method of claim 19, further comprising reading a command field in the datagram, the command field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

22. The method of claim 19, further comprising:

reading a command field in the datagram, the command field indicating that the datagram is a masked write command; and

reading a mode field in the datagram, the mode field indicating whether the datagram is an extended register masked write command, an extended register long masked write command, a register masked write command, or an extended register short masked write command.

23. The method of claim 19, wherein:

the mask field is a bit index field identifying a bit position to be changed in the RFFE register of the receiver; and

the data field is a bit value field providing a bit value for the bit position identified in the bit index field.

24. A receiver for receiving data from a transmitter, comprising:

a bus interface; and

a processing circuit configured to:

receive a datagram through the bus interface from the transmitter, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the receiver,

read a mask field in the datagram, the mask field identifying at least one bit to be changed in the RFFE register,

read a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register, and

change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field.

25. A method performed at a transmitter for sending data to a receiver through a bus interface, comprising:

enabling a masked-write operation by setting a single bit within a configuration register at the receiver to a first value;

generating a datagram to be transmitted to the receiver via the bus interface, the datagram providing an address value;

generating a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a

radio frequency front end (RFFE) register and a data field providing a value of the at least one bit to be changed in the RFFE register; and

transmitting the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver.

26. The method of claim 25, further comprising:

disabling the masked-write operation by setting the single bit within the configuration register at the receiver to a second value.

27. The method of claim 25, further comprising:

enabling a page segmented access operation by setting another single bit within the configuration register at the receiver to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the datagram when the page segmented access operation is enabled; and

disabling the page segmented access operation by setting the other single bit within the configuration register at the receiver to a second value, wherein the address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled.

28. A transmitter for sending data to a receiver, comprising:

a bus interface; and

a processing circuit configured to:

enable a masked-write operation by setting a single bit within a configuration register at the receiver to a first value,

generate a datagram to be transmitted to the receiver via the bus interface, the datagram providing an address value,

generate a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register and a data field providing a value of the at least one bit to be changed in the RFFE register, and

transmitting the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver.

29. The transmitter of claim 28, the processing circuit further configured to:

disable the masked-write operation by setting the single bit within the configuration register at the receiver to a second value.

30. The transmitter of claim 28, the processing circuit further configured to:

enable a page segmented access operation by setting another single bit within the configuration register at the receiver to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the datagram when the page segmented access operation is enabled; and

disable the page segmented access operation by setting the other single bit within the configuration register at the receiver to a second value, wherein the address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled.

31. A method performed at a receiver for receiving data from a transmitter through a bus interface, comprising:

receiving a first datagram from the transmitter for setting a single bit within a configuration register at the receiver;

detecting that a masked-write operation is enabled when the single bit within the configuration register is set to a first value;

receiving a second datagram from the transmitter, the second datagram providing an address value;

reading a payload field in the second datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register of the receiver and a data field providing a value of the at least one bit to be changed in the RFFE register; and

changing the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair.

32. The method of claim 31, further comprising:

detecting that the masked-write operation is disabled when the single bit within the configuration register at the receiver is set to a second value.

33. The method of claim 31, further comprising:

receiving a third datagram from the transmitter for setting another single bit within the configuration register at the receiver;

detecting that a page segmented access operation is enabled when the other single bit within the configuration register at the receiver is set to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the second datagram when the page segmented access operation is enabled; and

detecting that the page segmented access operation is disabled when the other single bit within the configuration register at the receiver is set to a second value, wherein the address of the RFFE register is the address value provided by the second datagram when the page segmented access operation is disabled.

34. A receiver for receiving data from a transmitter, comprising:

a bus interface; and

a processing circuit configured to:

receive a first datagram from the transmitter for setting a single bit within a configuration register at the receiver,

detect that a masked-write operation is enabled when the single bit within the configuration register is set to a first value,

receive a second datagram from the transmitter, the second datagram providing an address value,

read a payload field in the second datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register of the receiver and a data field providing a value of the at least one bit to be changed in the RFFE register, and

change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair.

35. The receiver of claim 34, the processing circuit further configured to:

detect that the masked-write operation is disabled when the single bit within the configuration register at the receiver is set to a second value.

36. The receiver of claim 34, the processing circuit further configured to:

receive a third datagram from the transmitter for setting another single bit within the configuration register at the receiver;

detect that a page segmented access operation is enabled when the other single bit within the configuration register at the receiver is set to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the second datagram when the page segmented access operation is enabled; and

detect that the page segmented access operation is disabled when the other single bit within the configuration register at the receiver is set to a second value, wherein the address of the RFFE register is the address value provided by the second datagram when the page segmented access operation is disabled.

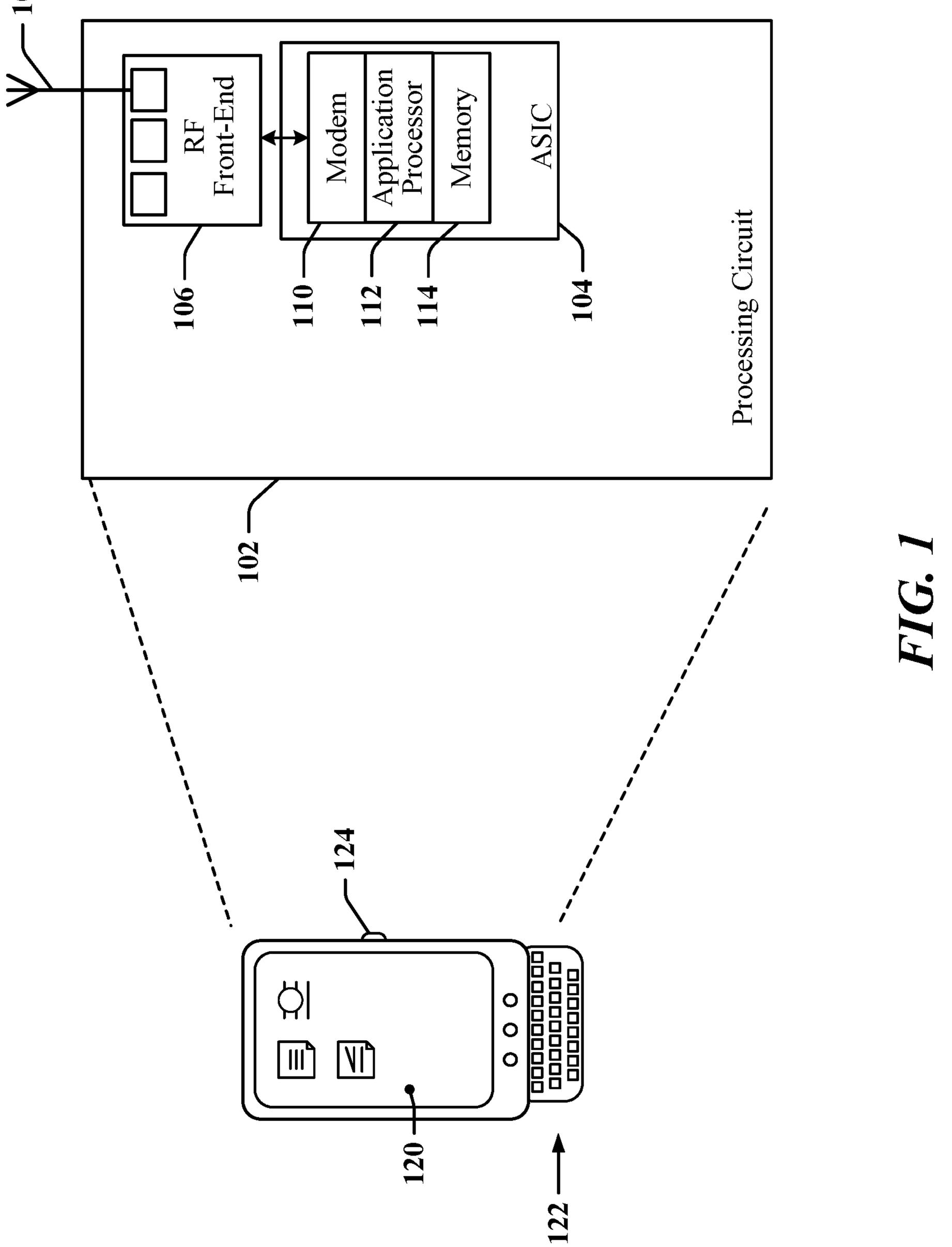
100
102
104
106
108
110
112
114
120
122
124
RF Front-End
Modem
Application Processor
Memory
ASIC
Processing Circuit

FIG. 1

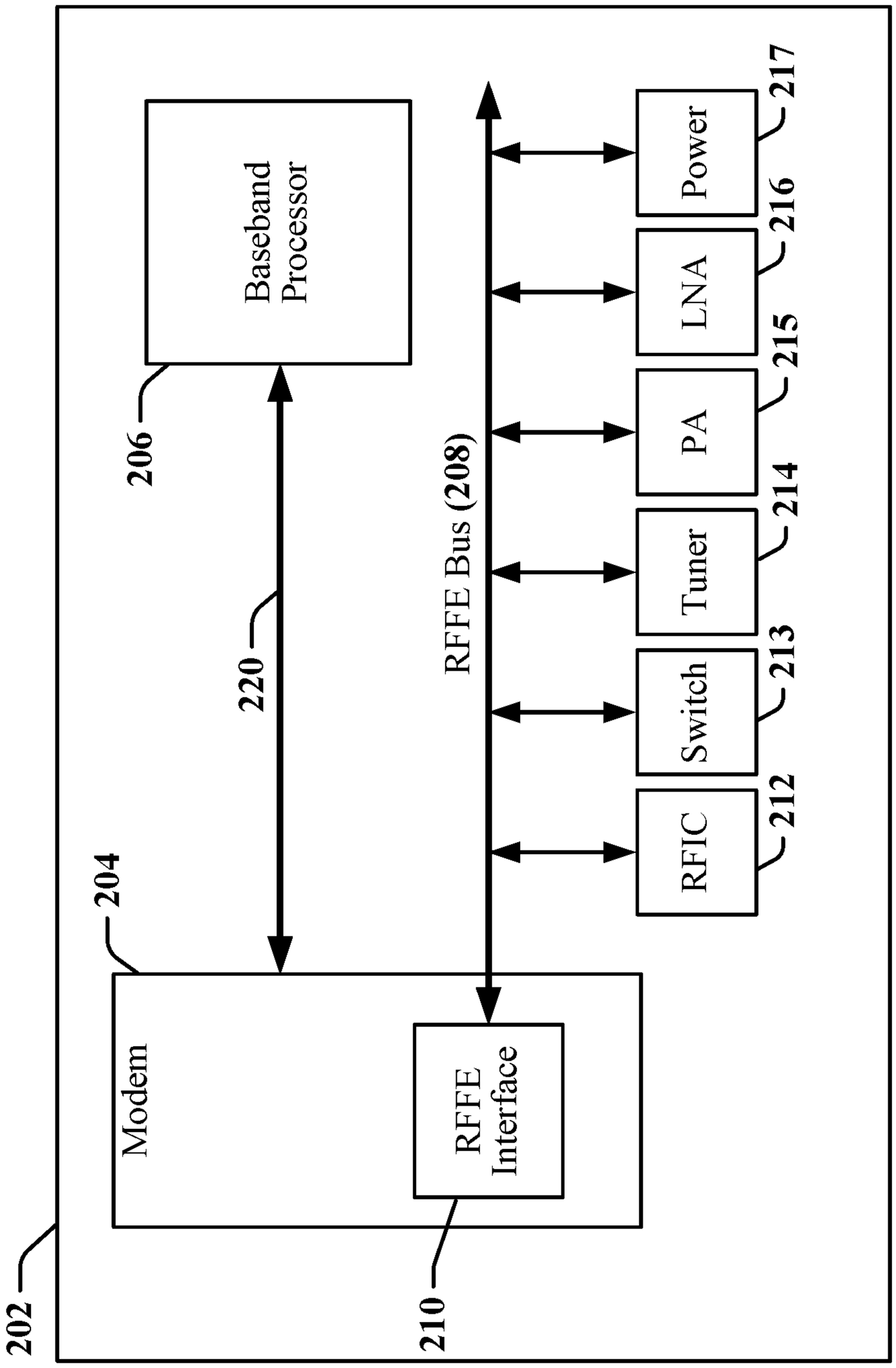
200
202
210
206
Baseband Processor
220
204
Modem
RFFE Interface
RFFE Bus (208)
RFIC
212
Switch
213
PA
215
LNA
216
Power
217
Tuner
214

FIG. 2

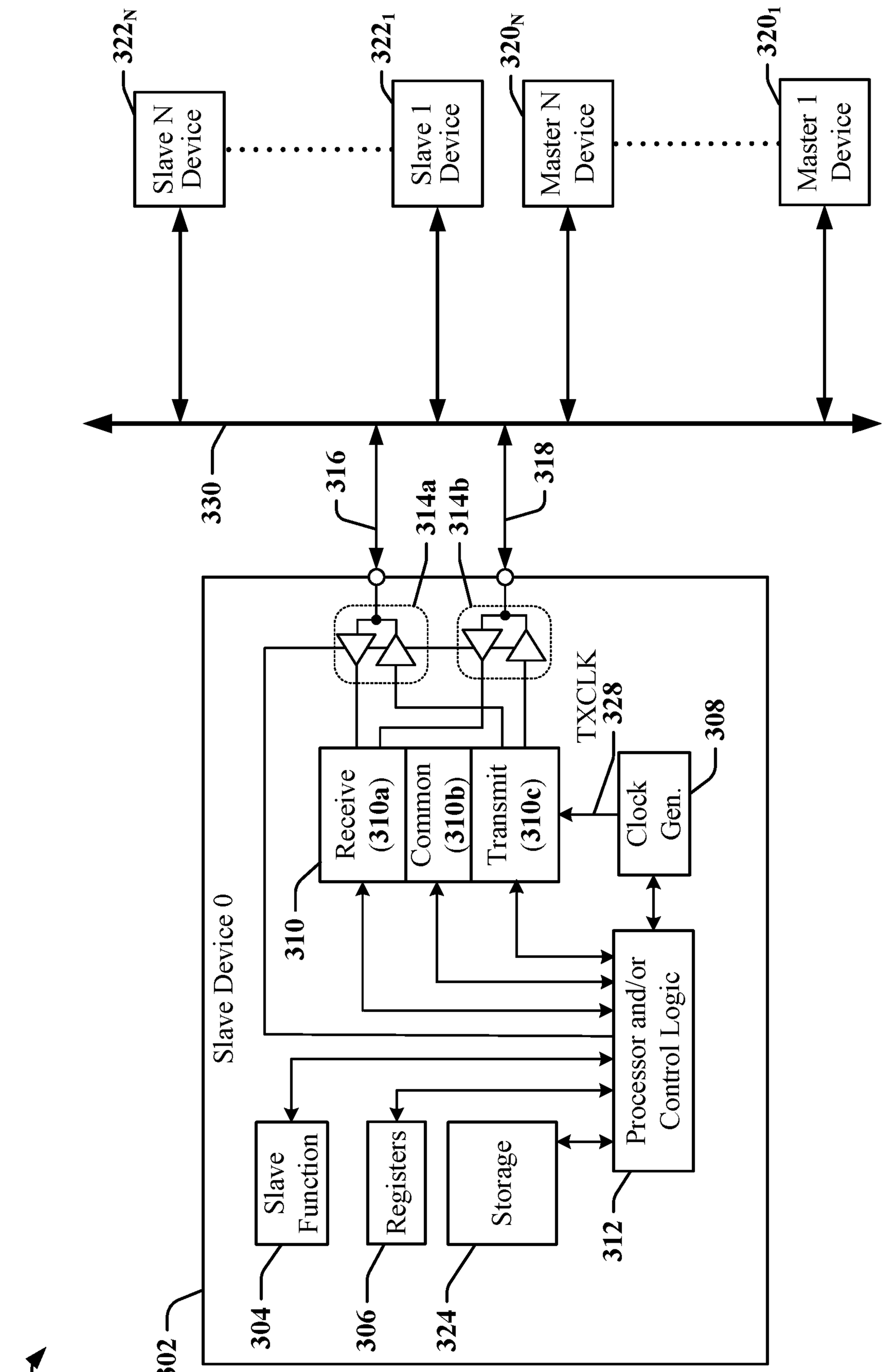
300
302
Slave Device 0
304
Slave Function
306
Registers
324
Storage
312
Processor and/or Control Logic
308
Clock Gen.
TXCLK
328
310
Receive (310a)
Common (310b)
Transmit (310c)
330
316
318
314a
314b
322_N
Slave N Device
322_1
Slave 1 Device
320_N
Master N Device
320_1
Master 1 Device

FIG. 3

Hex	Description	SSC	SA (4)	Command Frame								Reg-Adrs	Data
											P		
10	(Reserved)			0	0	0	1	0	0	0	0 P	Undefined	
11	(Reserved)			0	0	0	1	0	0	0	1 P	Undefined	
12	(Reserved)			0	0	0	1	0	0	1	0 P	Undefined	
13	(Reserved)			0	0	0	1	0	0	1	1 P	Undefined	
14	(Reserved)			0	0	0	1	0	1	0	0 P	Undefined	
15	(Reserved)			0	0	0	1	0	1	0	1 P	Undefined	
16	(Reserved)			0	0	0	1	0	1	1	0 P	Undefined	
17	(Reserved)			0	0	0	1	0	1	1	1 P	Undefined	
18	(Reserved)			0	0	0	1	1	0	0	0 P	Undefined	
19	(Reserved)			0	0	0	1	1	0	0	1 P	Undefined	
1A	(Reserved)			0	0	0	1	1	0	1	0 P	Undefined	
1B	(Reserved)			0	0	0	1	1	0	1	1 P	Undefined	

400

FIG. 4

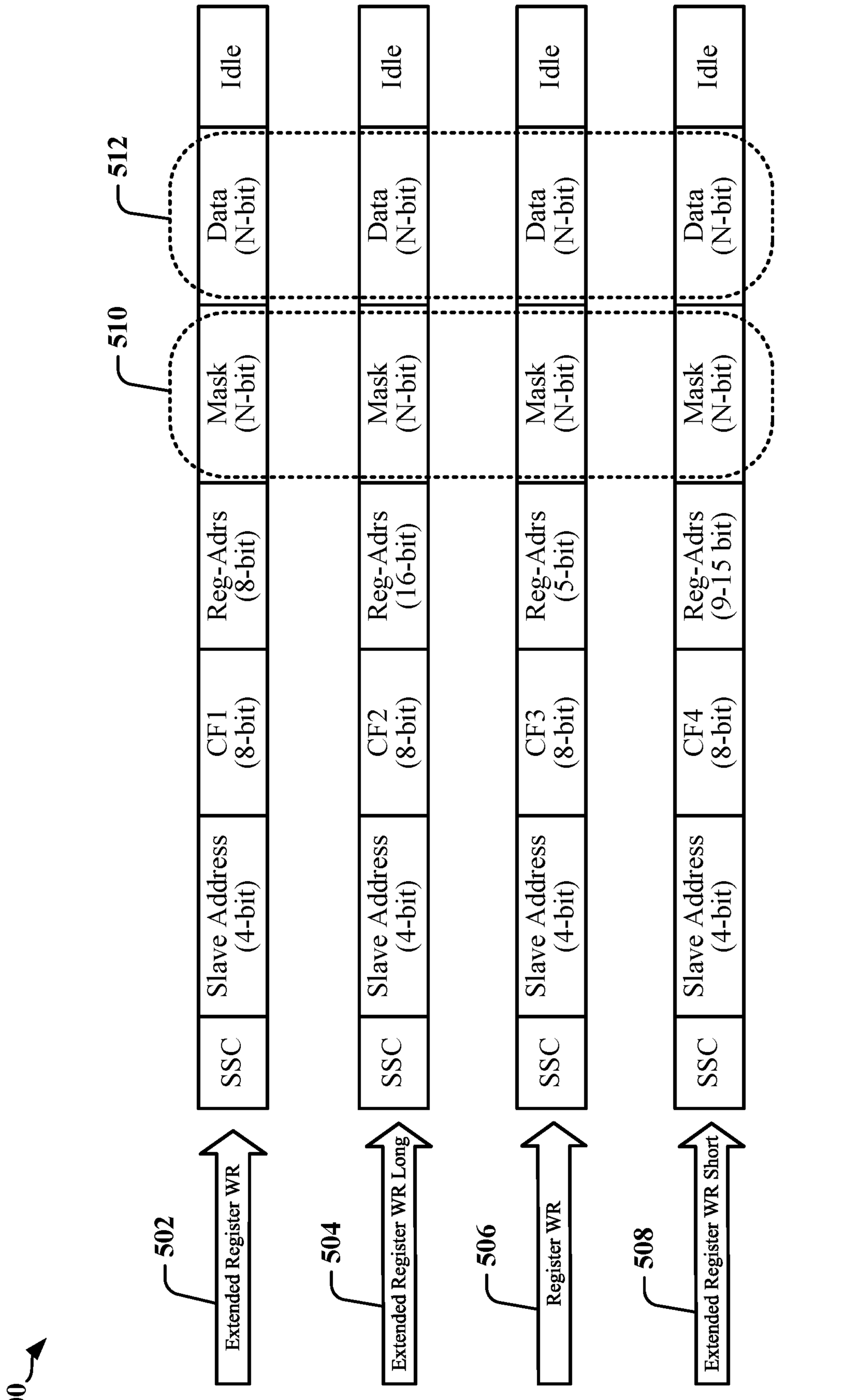
500
502
Extended Register WR
504
Extended Register WR Long
506
Register WR
508
Extended Register WR Short
510
512
SSC
Slave Address (4-bit)
Reg-Adrs (8-bit)
Mask (N-bit)
CF1 (8-bit)
Data (N-bit)
Idle
Reg-Adrs (16-bit)
CF2 (8-bit)
Reg-Adrs (5-bit)
CF3 (8-bit)
Reg-Adrs (9-15 bit)
CF4 (8-bit)

FIG. 5

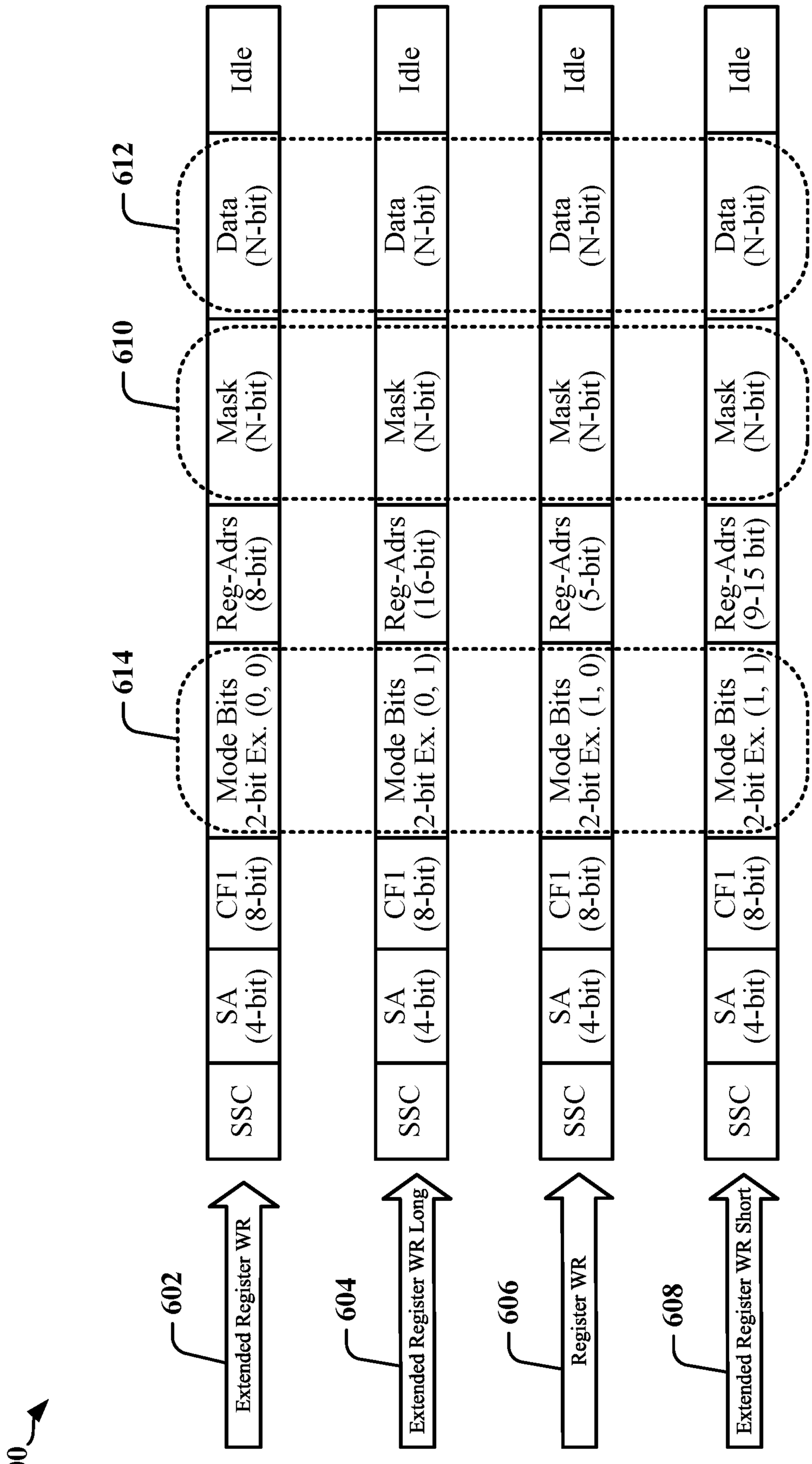
600
602
Extended Register WR
604
Extended Register WR Long
606
Register WR
608
Extended Register WR Short
610
612
614
Idle
Data (N-bit)
Mask (N-bit)
Reg-Adrs (8-bit)
Reg-Adrs (16-bit)
Reg-Adrs (5-bit)
Reg-Adrs (9-15 bit)
Mode Bits 2-bit Ex. (0, 0)
Mode Bits 2-bit Ex. (0, 1)
Mode Bits 2-bit Ex. (1, 0)
Mode Bits 2-bit Ex. (1, 1)
CF1 (8-bit)
SA (4-bit)
SSC

FIG. 6

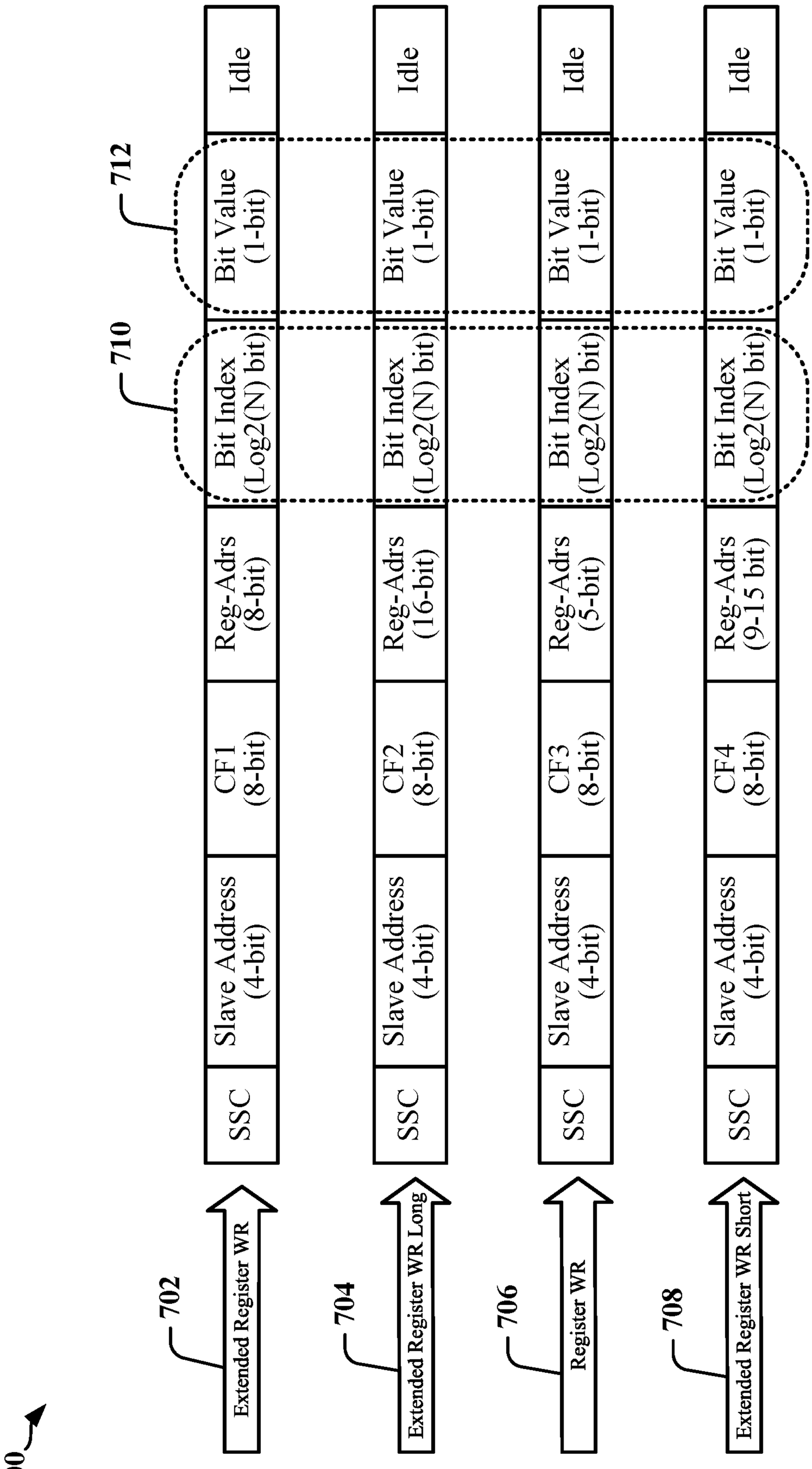
700
702
Extended Register WR
704
Extended Register WR Long
706
Register WR
708
Extended Register WR Short
710
712
SSC
Slave Address (4-bit)
Reg-Adrs (8-bit)
Bit Index (Log2(N) bit)
Bit Value (1-bit)
CF1 (8-bit)
Reg-Adrs (16-bit)
CF2 (8-bit)
Reg-Adrs (5-bit)
CF3 (8-bit)
Reg-Adrs (9-15 bit)
CF4 (8-bit)
Idle

FIG. 7

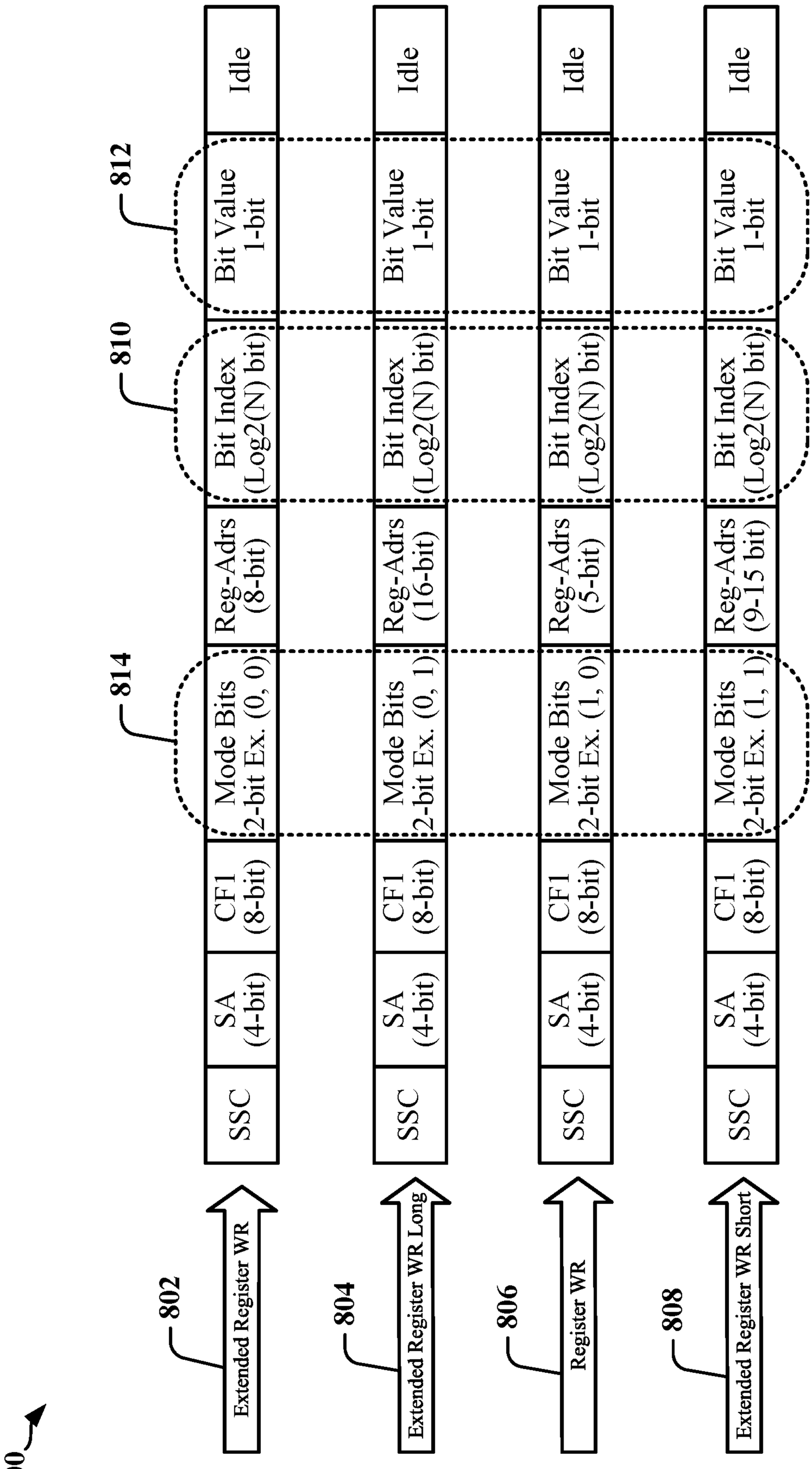
800
802
Extended Register WR
804
Extended Register WR Long
806
Register WR
808
Extended Register WR Short
810
812
814
SSC
SA (4-bit)
CF1 (8-bit)
Mode Bits 2-bit Ex. (0, 0)
Mode Bits 2-bit Ex. (0, 1)
Mode Bits 2-bit Ex. (1, 0)
Mode Bits 2-bit Ex. (1, 1)
Reg-Adrs (8-bit)
Reg-Adrs (16-bit)
Reg-Adrs (5-bit)
Reg-Adrs (9-15 bit)
Bit Index (Log2(N) bit)
Bit Value 1-bit
Idle

FIG. 8

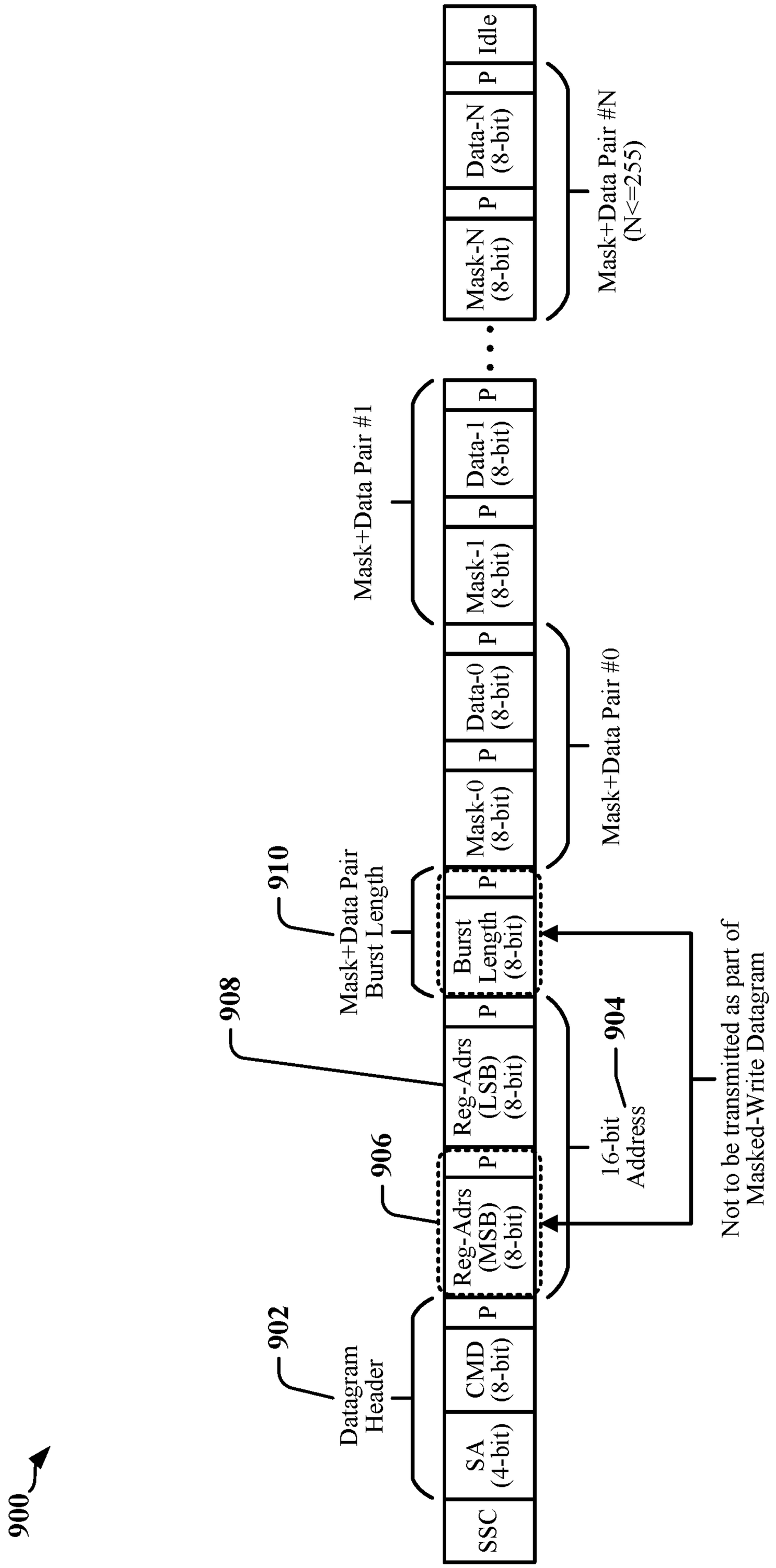
900
902
Datagram Header
SSC
SA (4-bit)
CMD (8-bit)
P
904
16-bit Address
906
Reg-Adrs (MSB) (8-bit)
P
908
Reg-Adrs (LSB) (8-bit)
P
910
Mask+Data Pair Burst Length
Burst Length (8-bit)
P
Not to be transmitted as part of Masked-Write Datagram
Mask+Data Pair #0
Mask-0 (8-bit)
P
Data-0 (8-bit)
P
Mask+Data Pair #1
Mask-1 (8-bit)
P
Data-1 (8-bit)
P
Mask+Data Pair #N (N<=255)
Mask-N (8-bit)
P
Data-N (8-bit)
P
Idle

FIG. 9

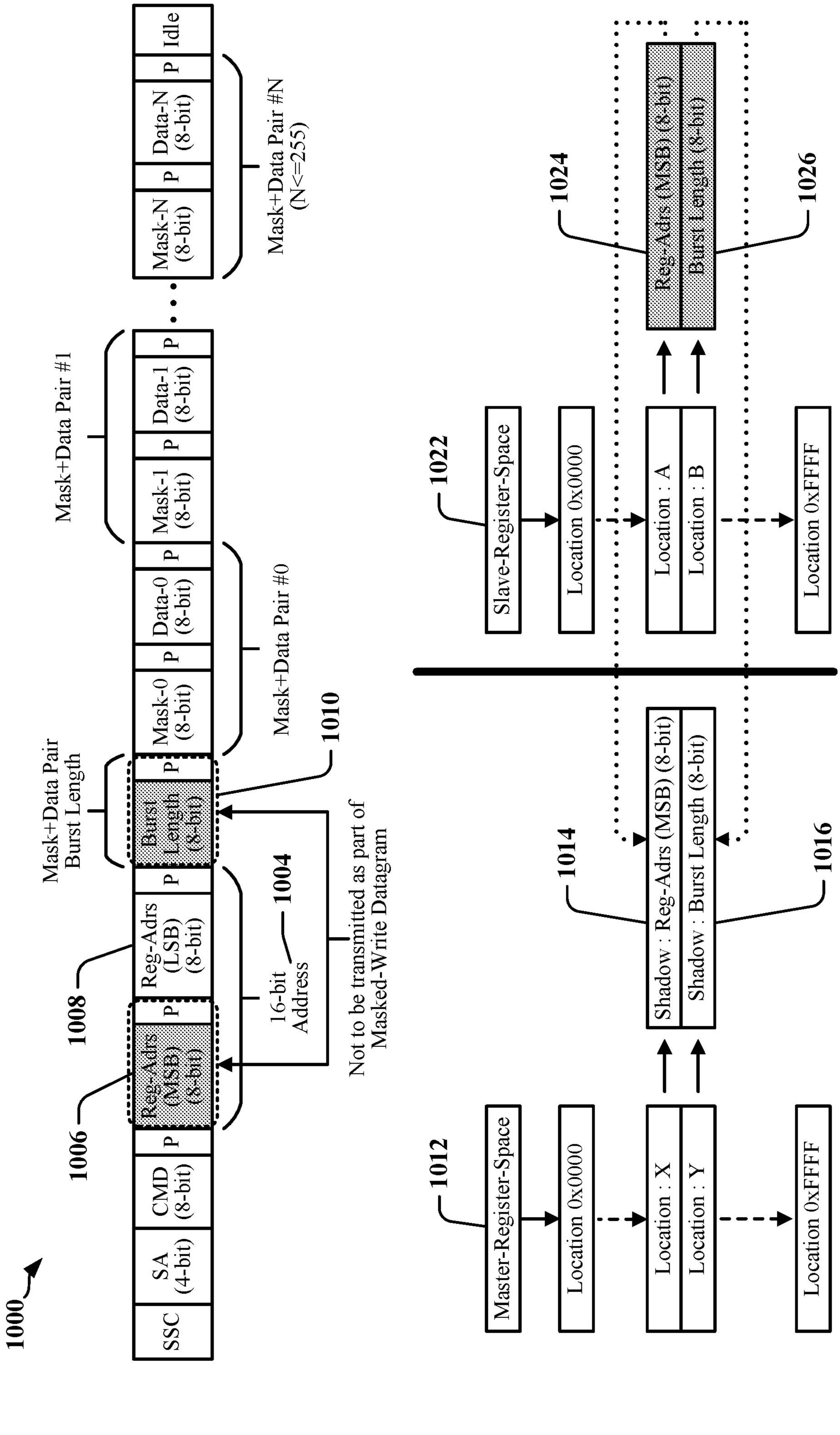
1000
SSC
SA (4-bit)
CMD (8-bit)
P
Reg-Adrs (MSB) (8-bit)
1006
Mask+Data Pair Burst Length
1008
Reg-Adrs (LSB) (8-bit)
P
Burst Length (8-bit)
1010
16-bit Address
1004
P
Mask-0 (8-bit)
Data-0 (8-bit)
P
Mask-1 (8-bit)
Data-1 (8-bit)
P
Mask+Data Pair #0
Mask+Data Pair #1
Mask-N (8-bit)
P
Data-N (8-bit)
P
Idle
Mask+Data Pair #N (N<=255)
Not to be transmitted as part of Masked-Write Datagram
1012
Master-Register-Space
Location 0x0000
Location : X
Location : Y
Location 0xFFFF
1014
Shadow : Reg-Adrs (MSB) (8-bit)
Shadow : Burst Length (8-bit)
1016
1022
Slave-Register-Space
Location 0x0000
Location : A
Location : B
Location 0xFFFF
1024
Reg-Adrs (MSB) (8-bit)
Burst Length (8-bit)
1026

FIG. 10

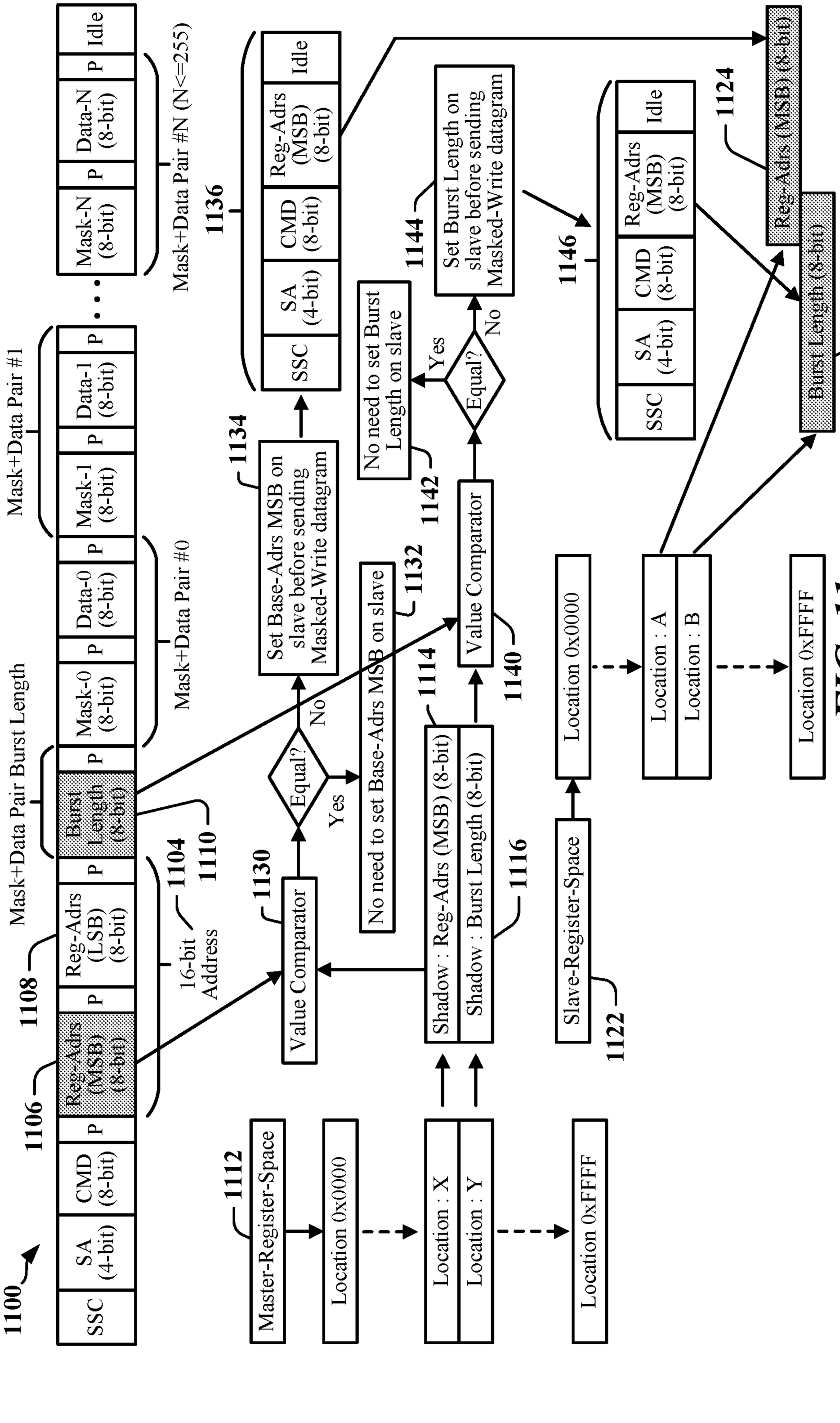
1100
SSC
SA (4-bit)
CMD (8-bit)
P
Reg-Adrs (MSB) (8-bit)
1106
1108
Reg-Adrs (LSB) (8-bit)
P
Burst Length (8-bit)
P
Data-0 (8-bit)
Mask-0 (8-bit)
P
Data-1 (8-bit)
Mask-1 (8-bit)
P
Data-N (8-bit)
Mask-N (8-bit)
P
Idle
Mask+Data Pair Burst Length
Mask+Data Pair #0
Mask+Data Pair #1
Mask+Data Pair #N (N<=255)
16-bit Address
1104
1110
1112
Master-Register-Space
Location 0x0000
Location 0xFFFF
Location : X
Location : Y
1114
Shadow : Reg-Adrs (MSB) (8-bit)
Shadow : Burst Length (8-bit)
1116
1122
Slave-Register-Space
Location 0x0000
Location 0xFFFF
Location : A
Location : B
1126
1130
Value Comparator
1132
No need to set Base-Adrs MSB on slave
1134
Set Base-Adrs MSB on slave before sending Masked-Write datagram
1136
1140
Value Comparator
1142
Equal?
Yes
No
No need to set Burst Length on slave
1144
Set Burst Length on slave before sending Masked-Write datagram
1146
1124

FIG. 11

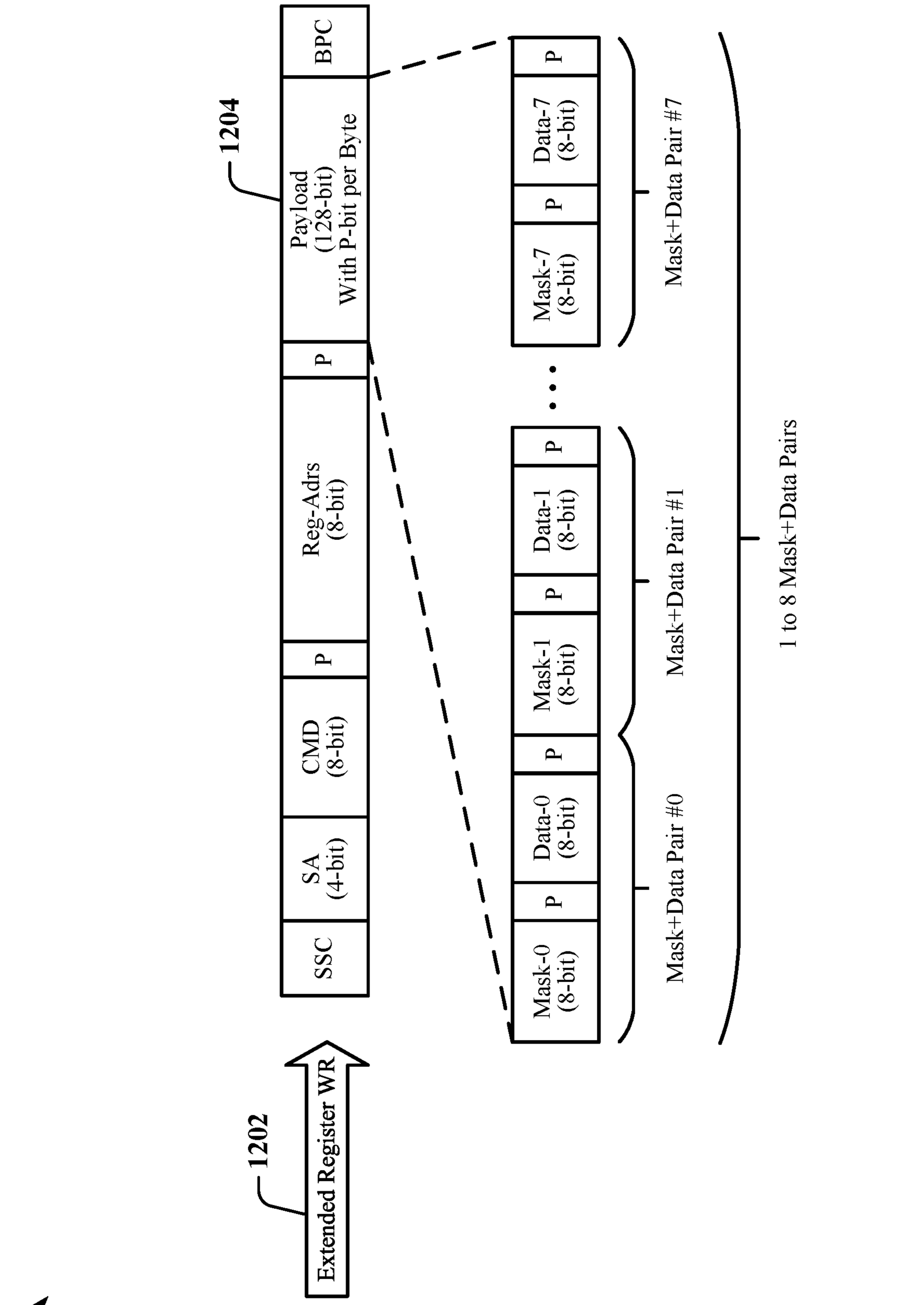
1200
1202
Extended Register WR
1204
SSC
SA (4-bit)
CMD (8-bit)
P
Reg-Adrs (8-bit)
P
Payload (128-bit) With P-bit per Byte
Mask-0 (8-bit)
P
Data-0 (8-bit)
P
Mask-1 (8-bit)
P
Data-1 (8-bit)
P
Mask-7 (8-bit)
P
Data-7 (8-bit)
Mask+Data Pair #0
Mask+Data Pair #1
Mask+Data Pair #7
1 to 8 Mask+Data Pairs

FIG. 12

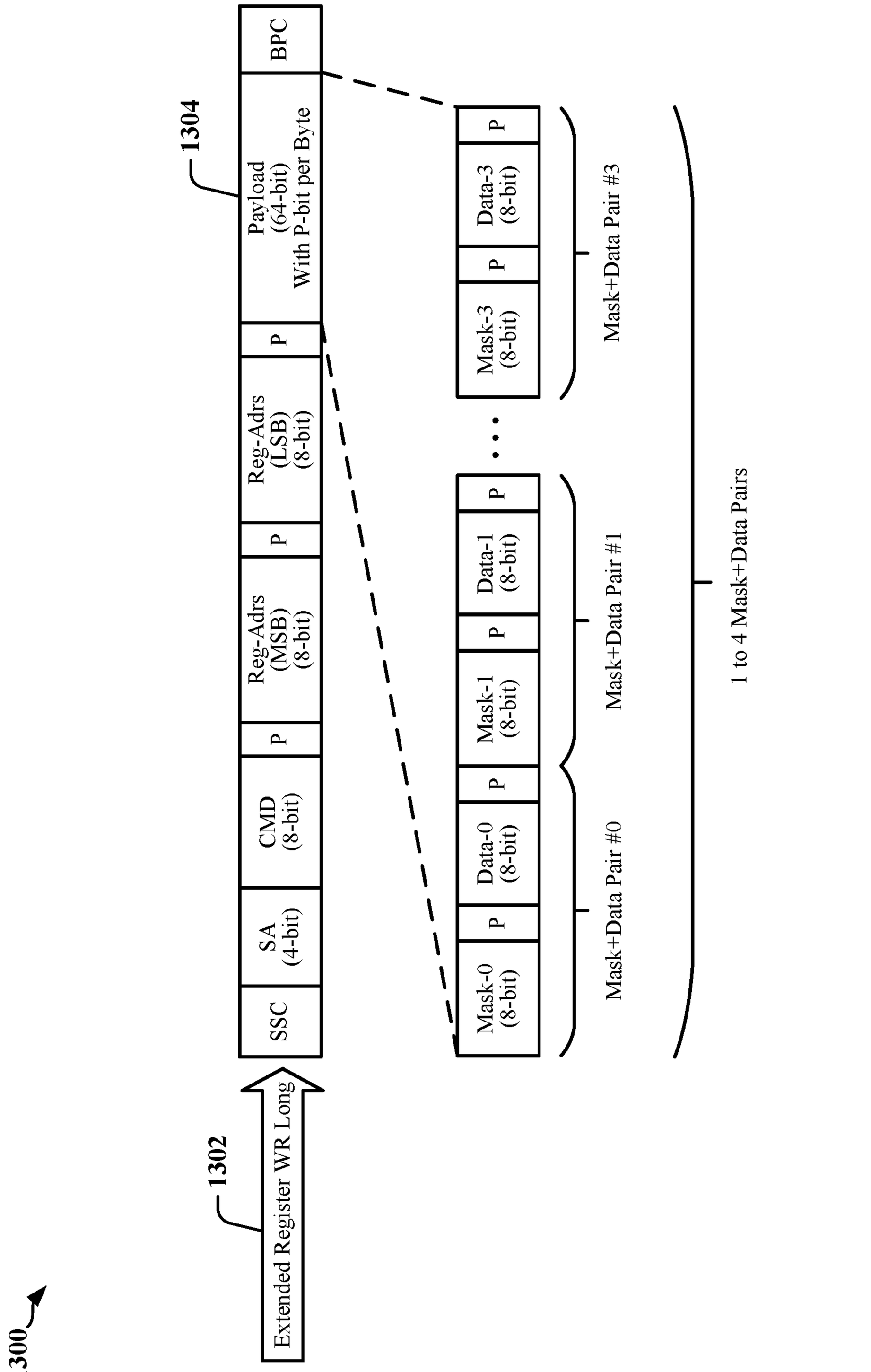
1300
1302
Extended Register WR Long
1304
SSC
SA (4-bit)
CMD (8-bit)
P
Reg-Adrs (MSB) (8-bit)
Reg-Adrs (LSB) (8-bit)
P
Payload (64-bit) With P-bit per Byte
BPC
Mask-0 (8-bit)
P
Data-0 (8-bit)
P
Mask-1 (8-bit)
P
Data-1 (8-bit)
P
Mask-3 (8-bit)
P
Data-3 (8-bit)
P
Mask+Data Pair #0
Mask+Data Pair #1
Mask+Data Pair #3
1 to 4 Mask+Data Pairs

FIG. 13

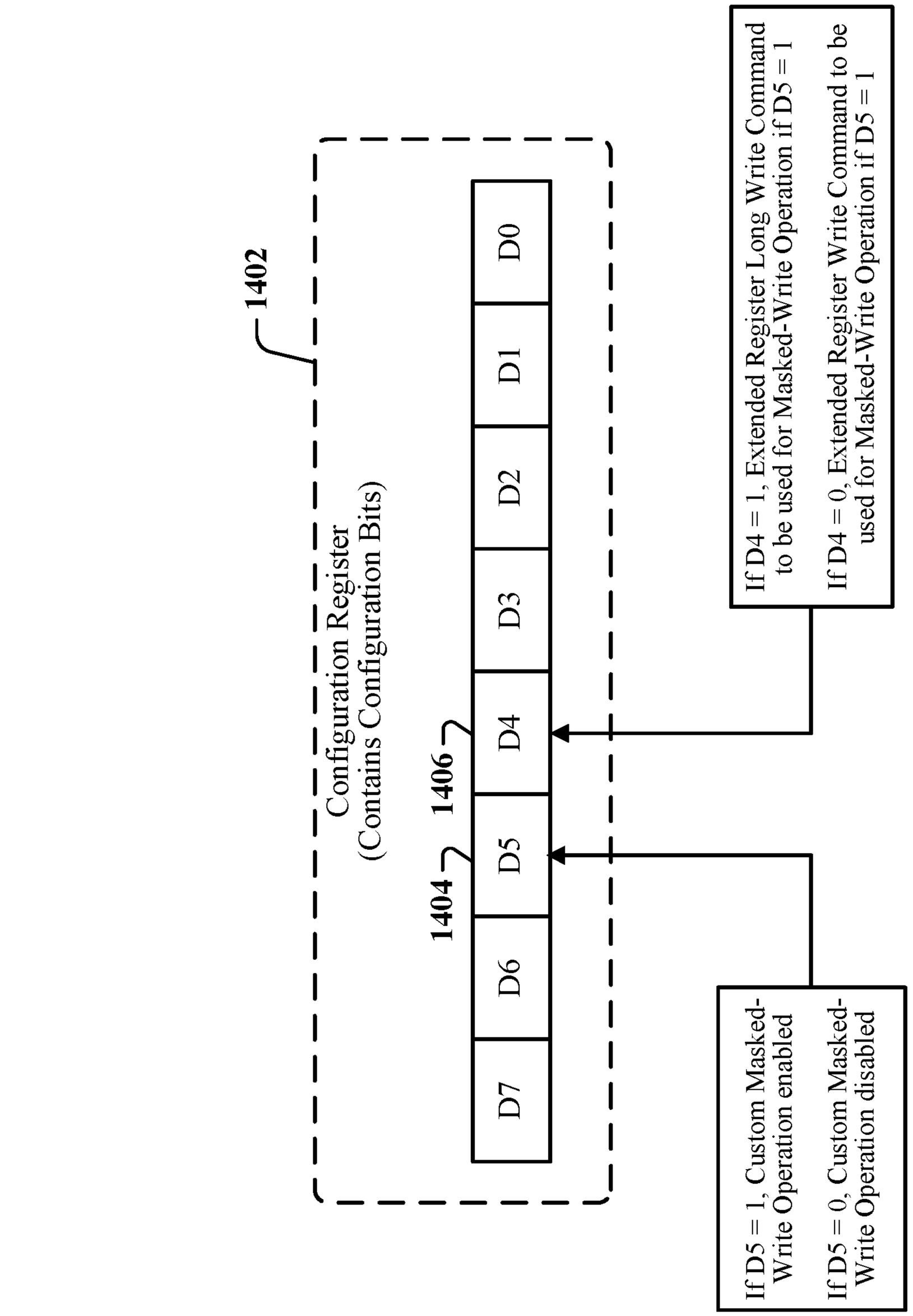
1400
1402
Configuration Register
(Contains Configuration Bits)
D7
D6
D5
D4
D3
D2
D1
D0
1404
1406
If D5 = 1, Custom Masked-Write Operation enabled
If D5 = 0, Custom Masked-Write Operation disabled
If D4 = 1, Extended Register Long Write Command to be used for Masked-Write Operation if D5 = 1
If D4 = 0, Extended Register Write Command to be used for Masked-Write Operation if D5 = 1

FIG. 14

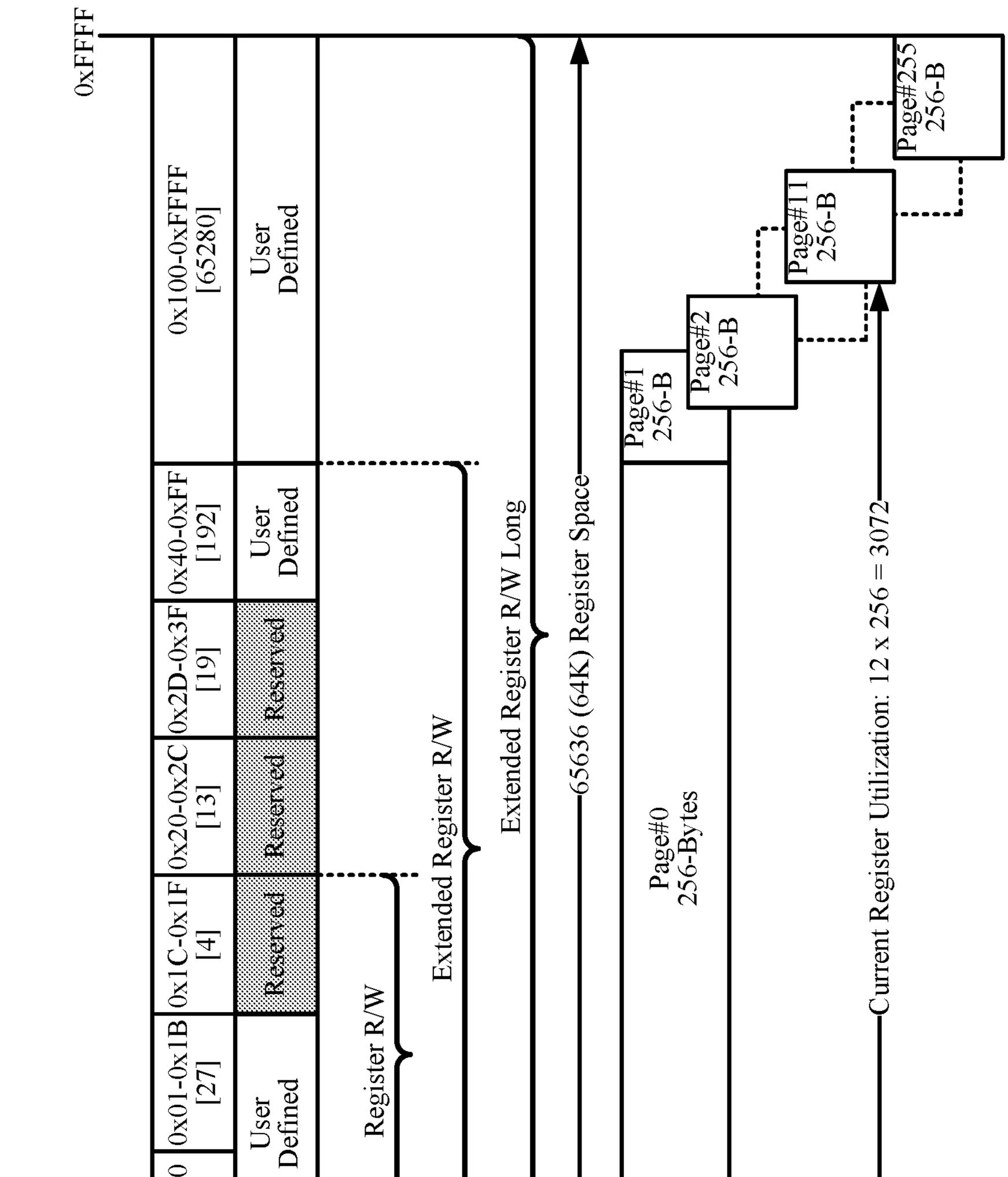
0xFFFF
0x100-0xFFFF [65280]
User Defined
0x40-0xFF [192]
User Defined
0x00 [1]
0x01-0x1B [27]
User Defined
0x1C-0x1F [4]
Reserved
0x20-0x2C [13]
Reserved
0x2D-0x3F [19]
Reserved
0x40-0x
[19
User
Defi
Register R/W
Extended Register R/W
Long
Extended Register R/W Lo
65636 (64K) Register S
Page#0
256-Bytes
Page#1
256-B
Page#2
256-B
Page#11
256-B
Page#255
256-B
3072
Current Register Utilization: 12 x 256 = 30
0x0000
1500

FIG. 15

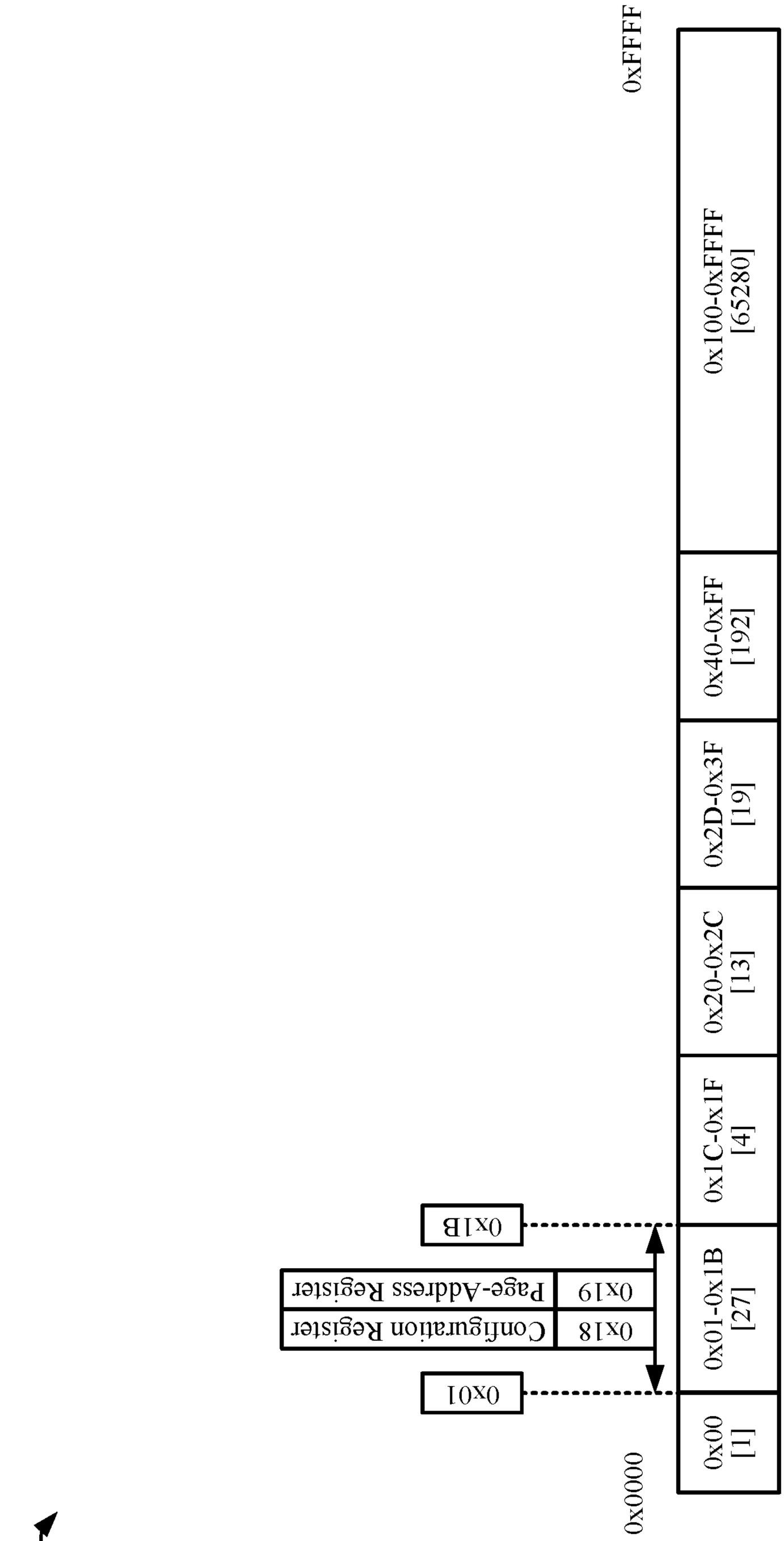
1600
0x0000
0x00 [1]
0x01-0x1B [27]
0x1C-0x1F [4]
0x20-0x2C [13]
0x2D-0x3F [19]
0x40-0xFF [192]
0x100-0xFFFF [65280]
0xFFFF
0x01
0x18
Configuration Register
0x19
Page-Address Register
0x1B

FIG. 16

1700

Bit and POR Value	Function	Description
D7 : [0]	Reserved	Reserved
D6 : [0]	Reserved	Reserved
D5 : [0]	Reserved	Reserved
D4 : [0]	Reserved	Reserved
D3 : [0]	Reserved	Reserved
D2 : [0]	Page Segmented Access (PSA)	1: Enable PSA; 0: Disable PSA
D1 : [0]	Double Data Rate (DDR) Mode	1: Enable DDR Mode; 0: Disable DDR Mode
D0 : [0]	Custom Masked-Write (CMW)	1: Enable CMW; 0: Disable CMW

1750

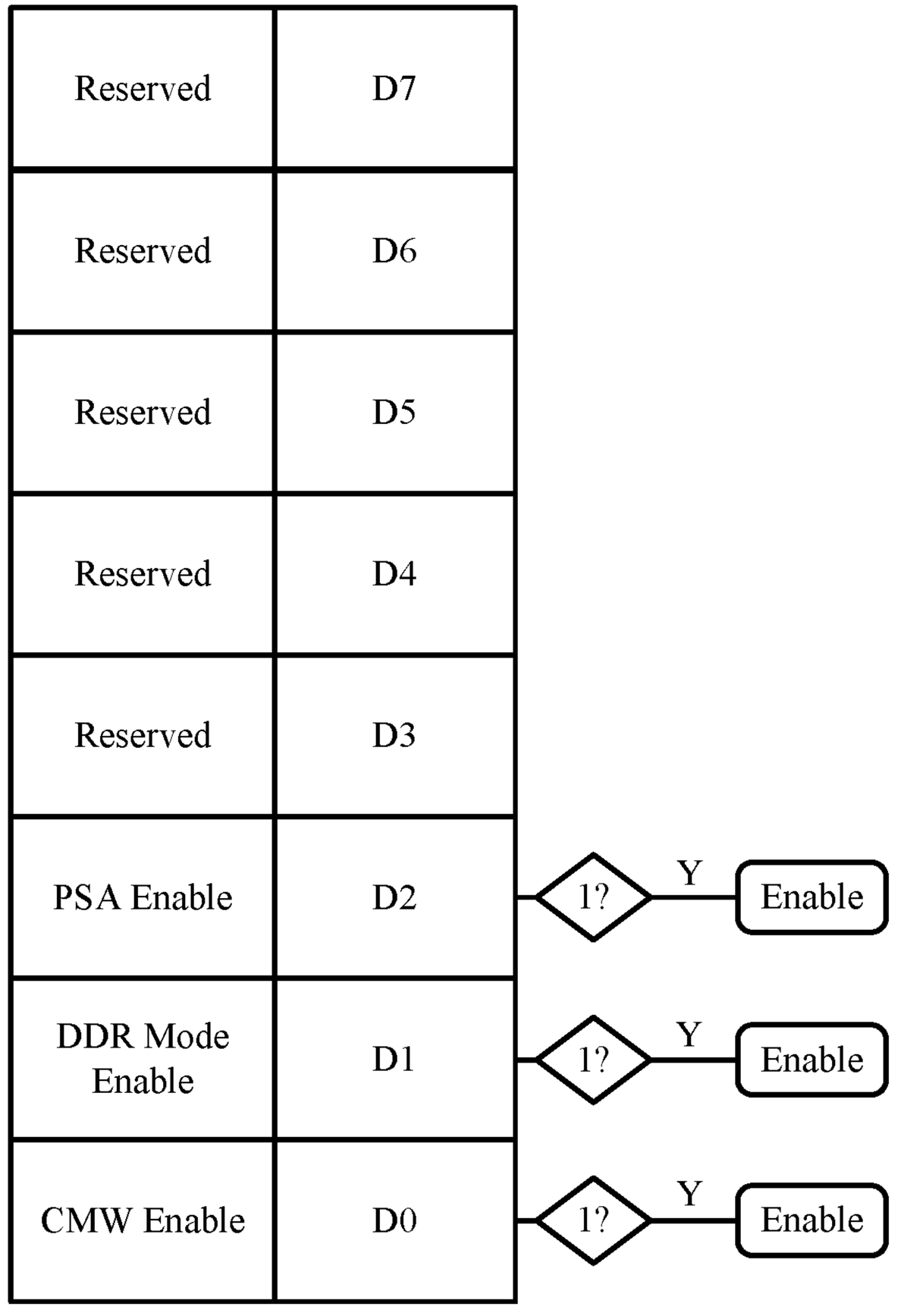

FIG. 17

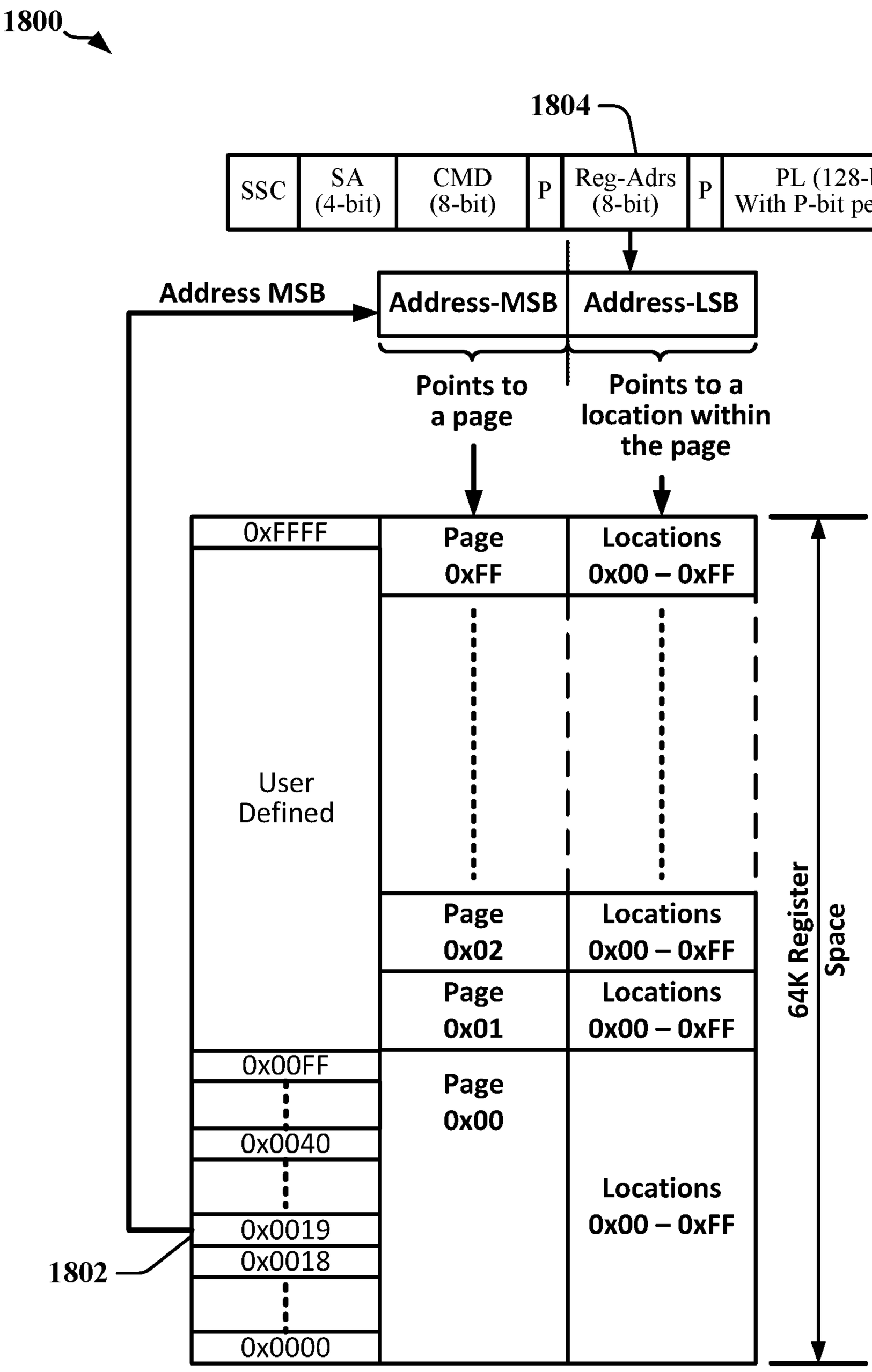
1800
1804
SSC
SA (4-bit)
CMD (8-bit)
P
Reg-Adrs (8-bit)
P
PL (128-bit) With P-bit per Byte
BPC
Address MSB
Address-MSB
Address-LSB
Points to a page
Points to a location within the page
0xFFFF
User Defined
Page 0xFF
Locations 0x00 – 0xFF
Page 0x02
Locations 0x00 – 0xFF
Page 0x01
Locations 0x00 – 0xFF
0x00FF
0x0040
0x0019
0x0018
0x0000
Page 0x00
Locations 0x00 – 0xFF
64K Register Space
1802

FIG. 18

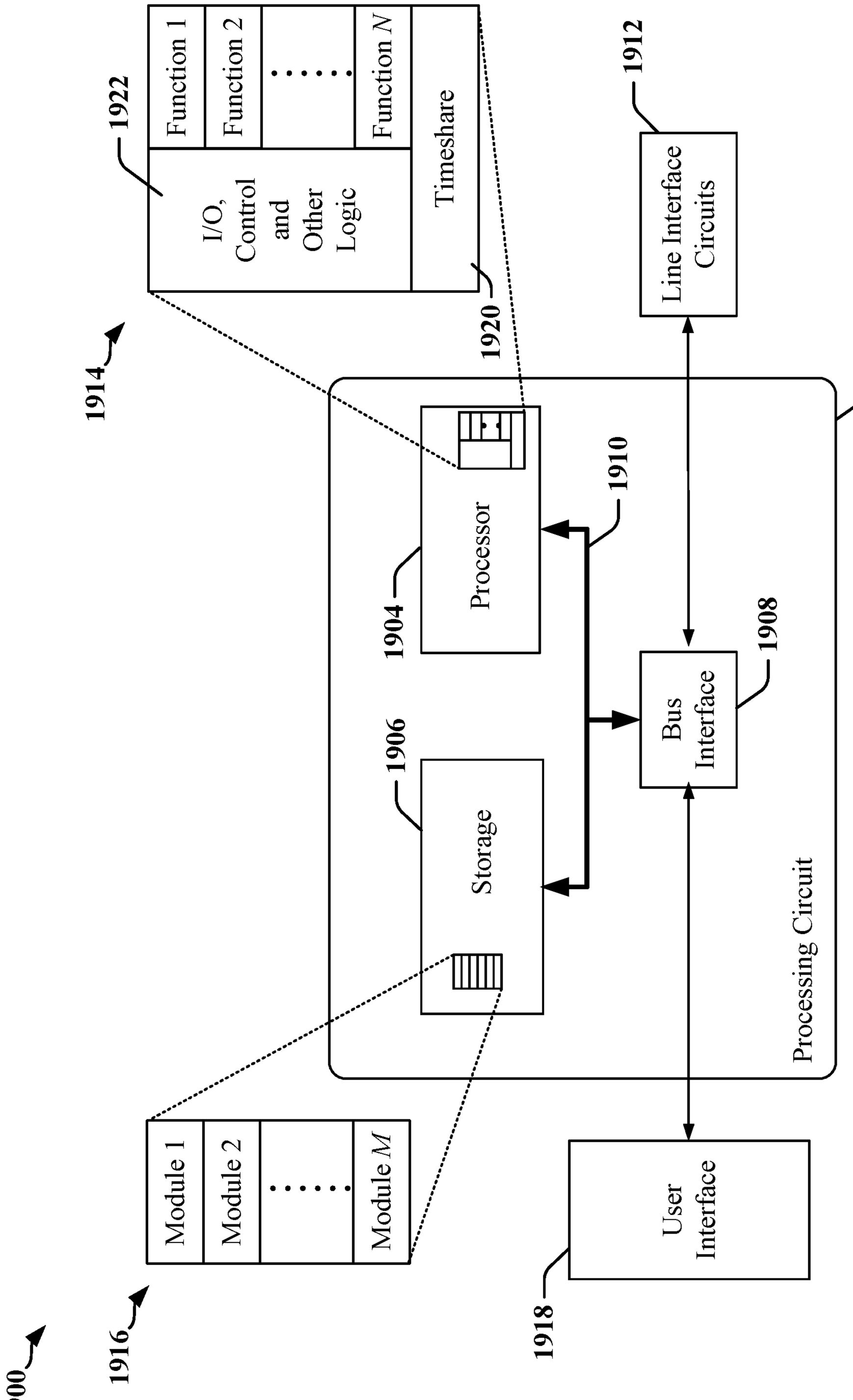
1900
1902
1904
Processor
1906
Storage
1908
Bus Interface
1910
1912
Line Interface Circuits
1914
Processing Circuit
1916
Module 1
Module 2
Module M
1918
User Interface
1920
1922
Timeshare
Function 1
Function 2
Function N
I/O, Control and Other Logic

FIG. 19

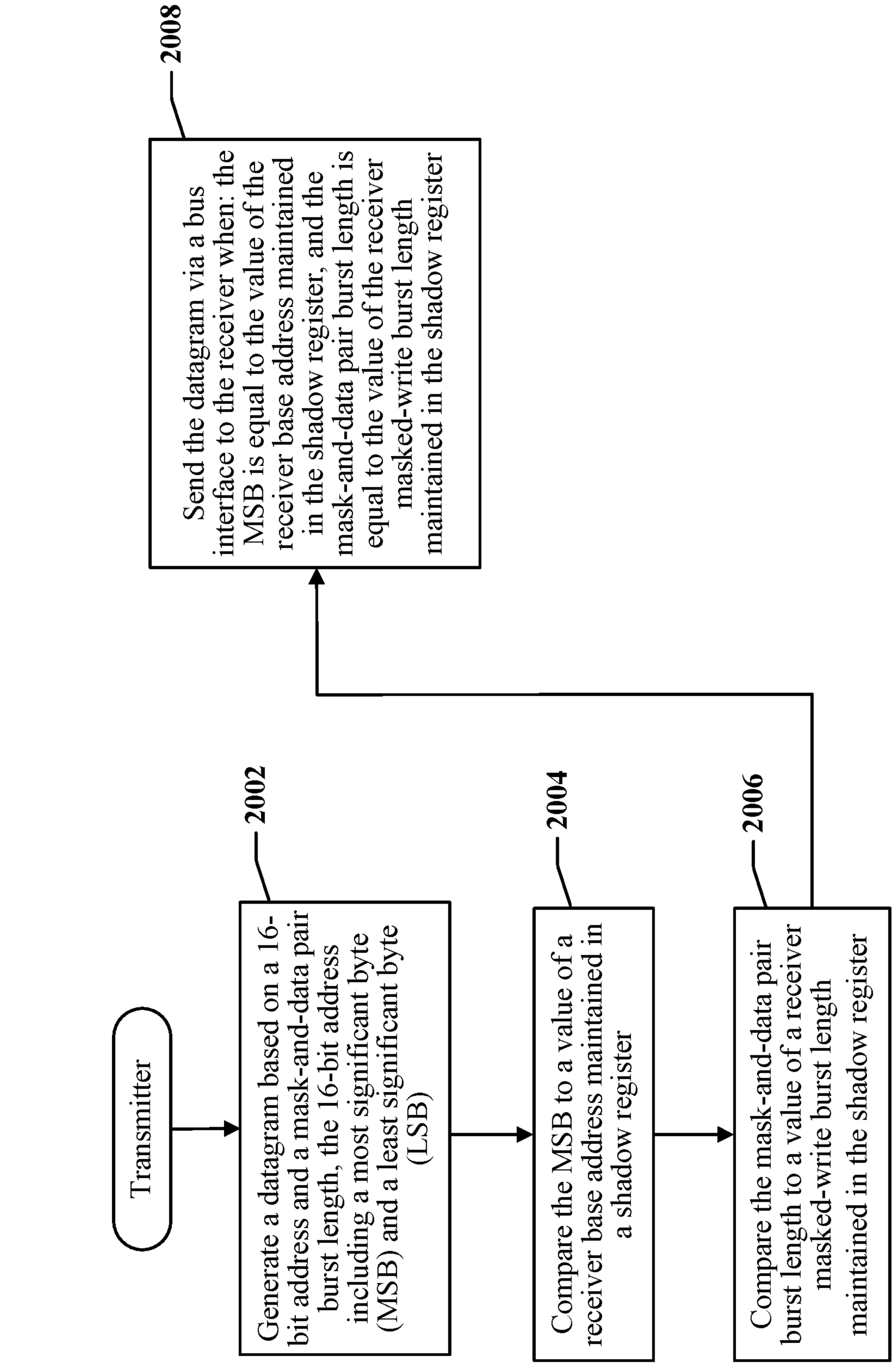
2000
Transmitter
2002
Generate a datagram based on a 16-bit address and a mask-and-data pair burst length, the 16-bit address including a most significant byte (MSB) and a least significant byte (LSB)
2004
Compare the MSB to a value of a receiver base address maintained in a shadow register
2006
Compare the mask-and-data pair burst length to a value of a receiver masked-write burst length maintained in the shadow register
Send the datagram via a bus interface to the receiver when: the MSB is equal to the value of the receiver base address maintained in the shadow register, and the mask-and-data pair burst length is equal to the value of the receiver masked-write burst length maintained in the shadow register

FIG. 20

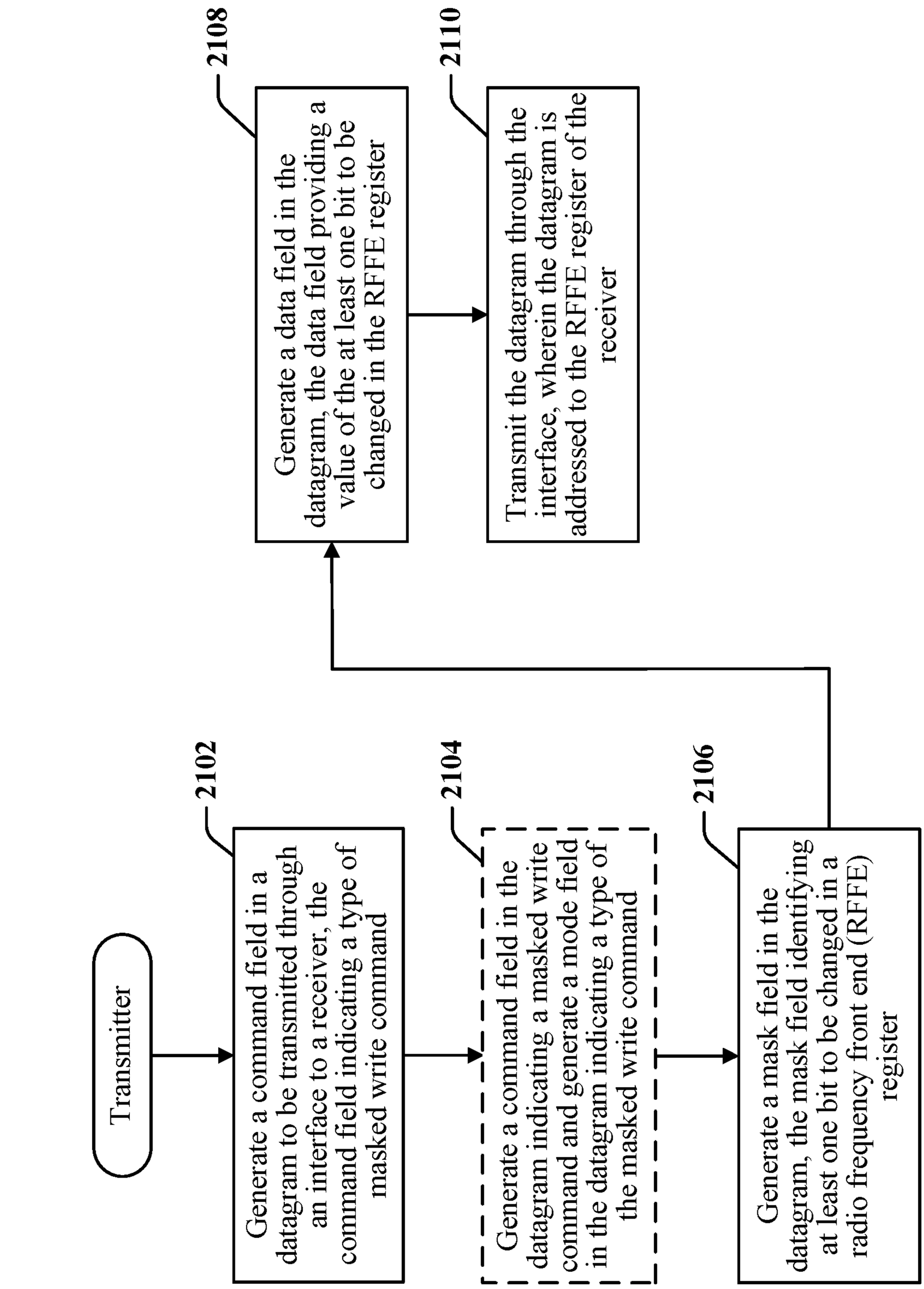
2100
Transmitter
2102
Generate a command field in a datagram to be transmitted through an interface to a receiver, the command field indicating a type of masked write command
2104
Generate a command field in the datagram indicating a masked write command and generate a mode field in the datagram indicating a type of the masked write command
2106
Generate a mask field in the datagram, the mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register
2108
Generate a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register
2110
Transmit the datagram through the interface, wherein the datagram is addressed to the RFFE register of the receiver

FIG. 21

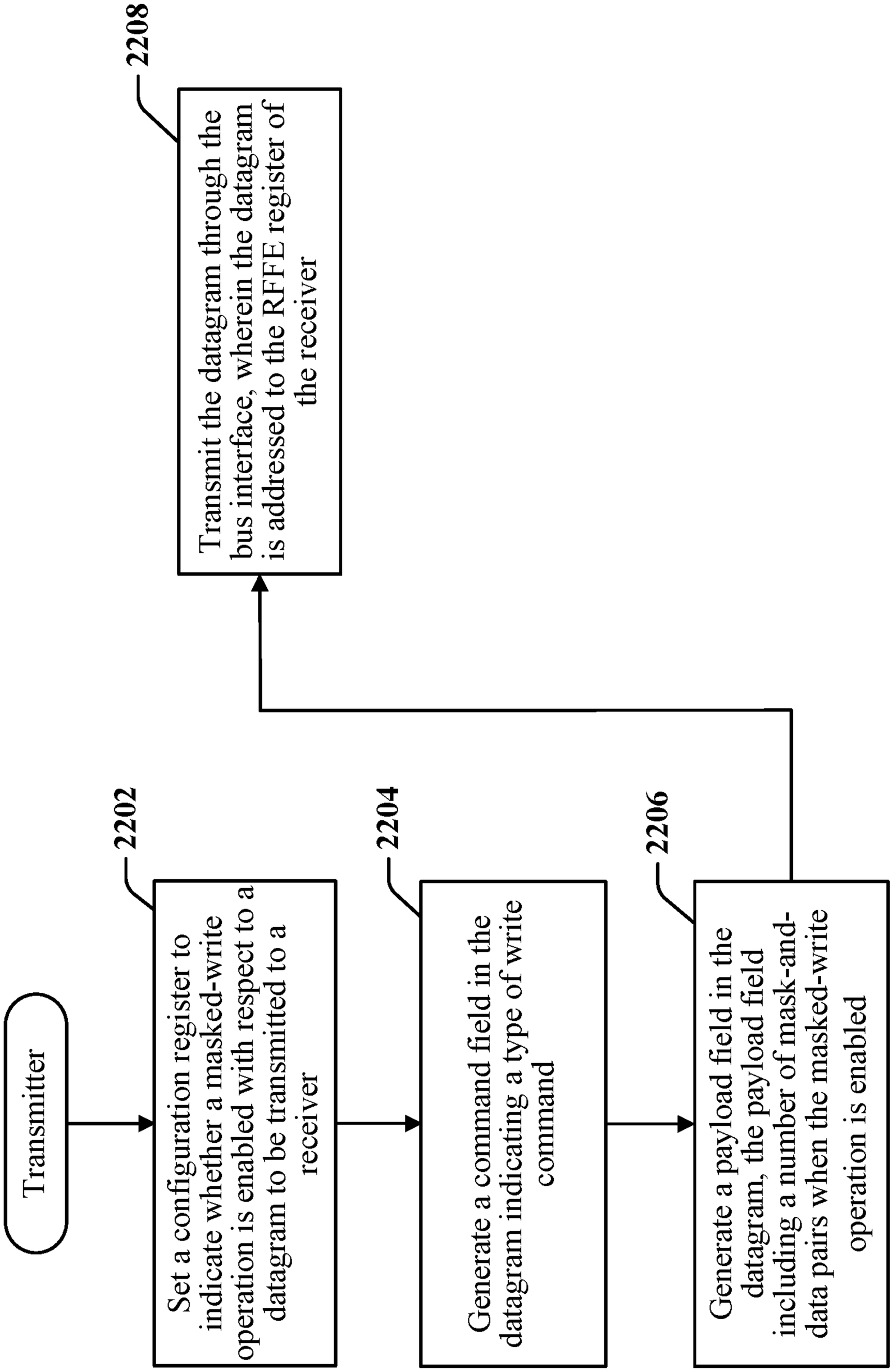
2200
Transmitter
2202
Set a configuration register to indicate whether a masked-write operation is enabled with respect to a datagram to be transmitted to a receiver
2204
Generate a command field in the datagram indicating a type of write command
2206
Generate a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled
2208
Transmit the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver

FIG. 22

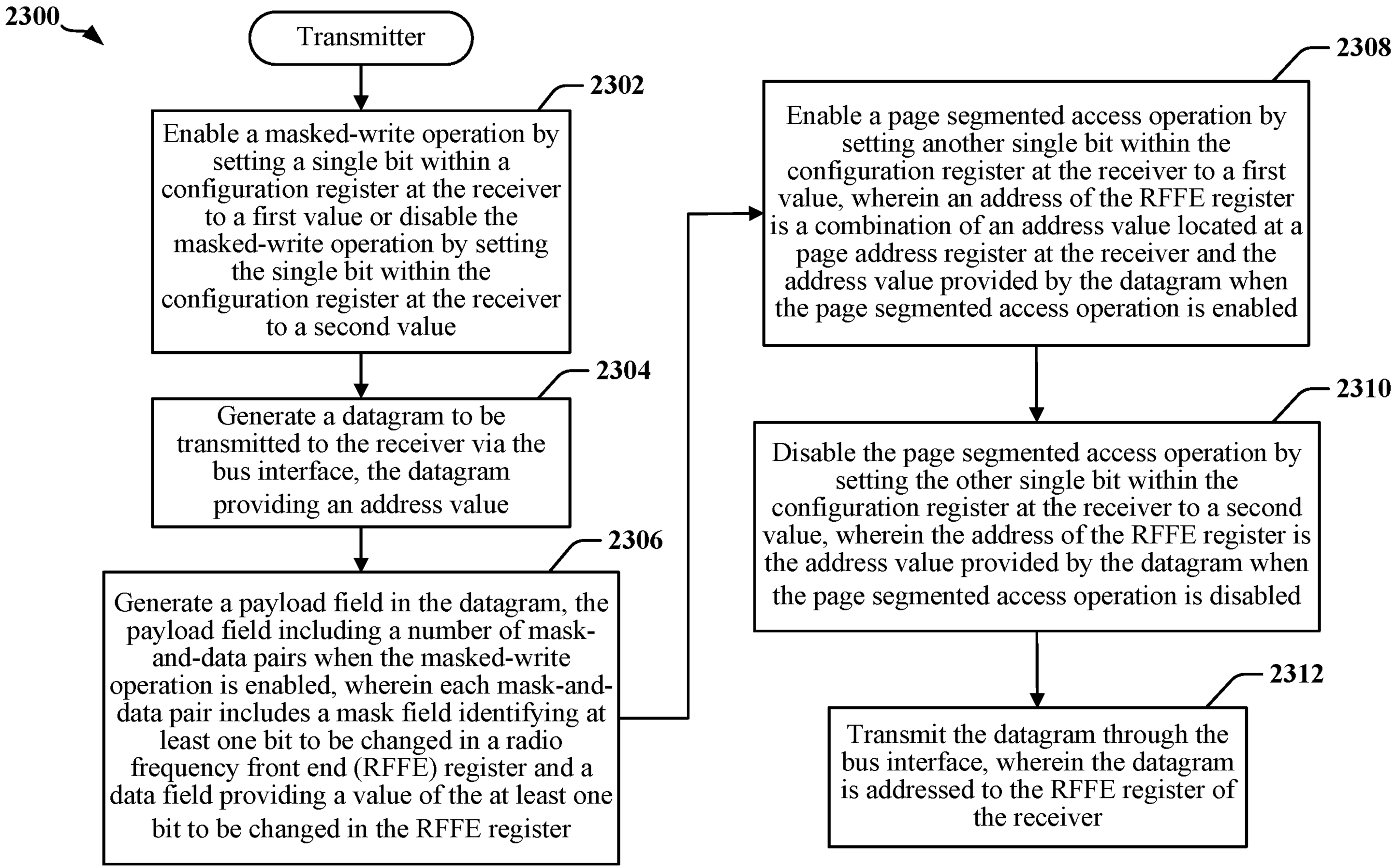
2300
Transmitter
2302
Enable a masked-write operation by setting a single bit within a configuration register at the receiver to a first value or disable the masked-write operation by setting the single bit within the configuration register at the receiver to a second value
2304
Generate a datagram to be transmitted to the receiver via the bus interface, the datagram providing an address value
2306
Generate a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register and a data field providing a value of the at least one bit to be changed in the RFFE register
2308
Enable a page segmented access operation by setting another single bit within the configuration register at the receiver to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the datagram when the page segmented access operation is enabled
2310
Disable the page segmented access operation by setting the other single bit within the configuration register at the receiver to a second value, wherein the address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled
2312
Transmit the datagram through the bus interface, wherein the datagram is addressed to the RFFE register of the receiver

FIG. 23

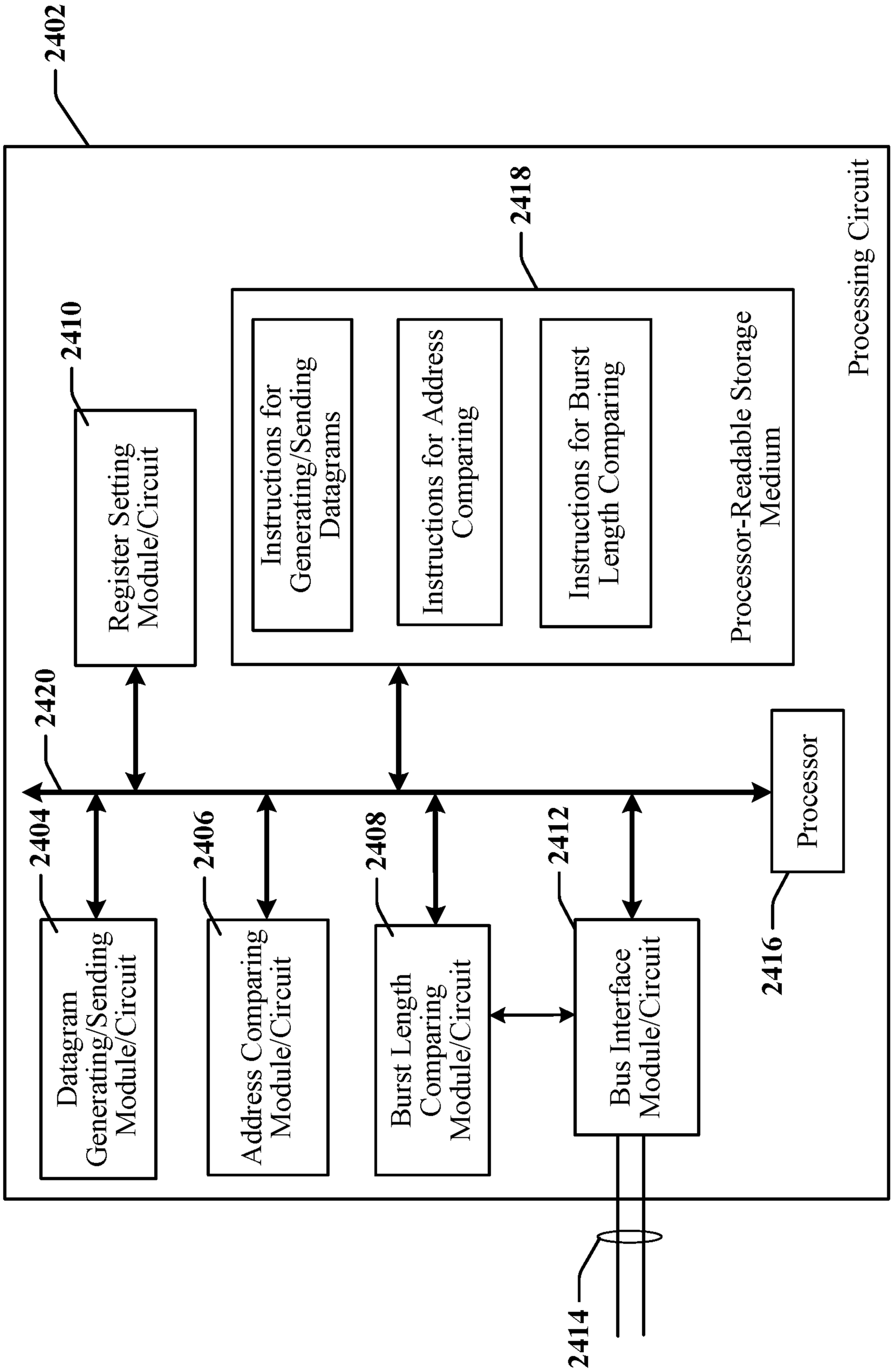
2402
Processing Circuit
2418
Processor-Readable Storage Medium
Instructions for Generating/Sending Datagrams
Instructions for Address Comparing
Instructions for Burst Length Comparing
2410
Register Setting Module/Circuit
2420
2404
Datagram Generating/Sending Module/Circuit
2406
Address Comparing Module/Circuit
2408
Burst Length Comparing Module/Circuit
2412
Bus Interface Module/Circuit
2416
Processor
2414
2400

FIG. 24

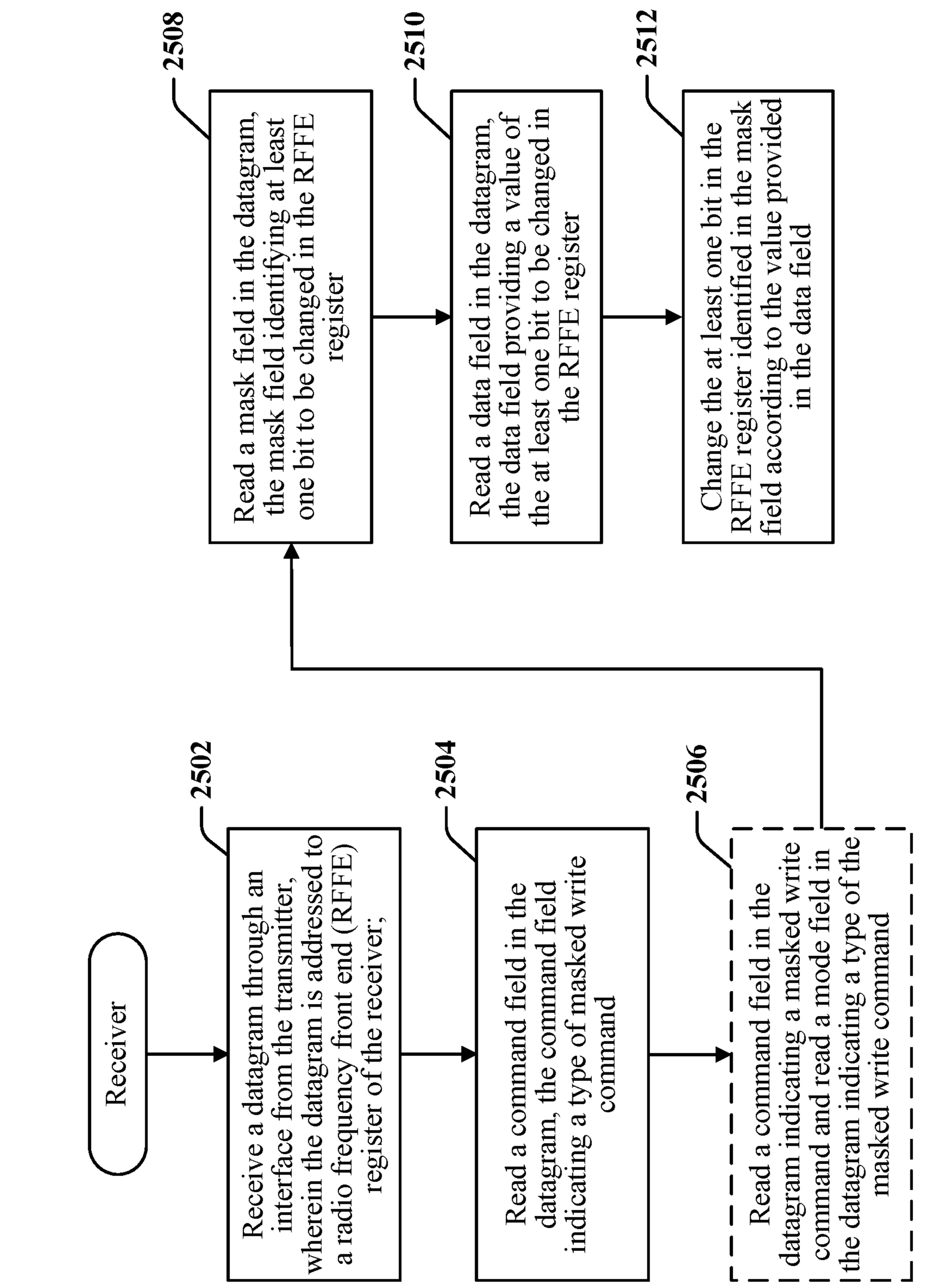
2500
Receiver
2502
Receive a datagram through an interface from the transmitter, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the receiver;
2504
Read a command field in the datagram, the command field indicating a type of masked write command
2506
Read a command field in the datagram indicating a masked write command and read a mode field in the datagram indicating a type of the masked write command
Read a mask field in the datagram, the mask field identifying at least one bit to be changed in the RFFE register
2508
Read a data field in the datagram, the data field providing a value of the at least one bit to be changed in the RFFE register
2510
Change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field
2512

FIG. 25

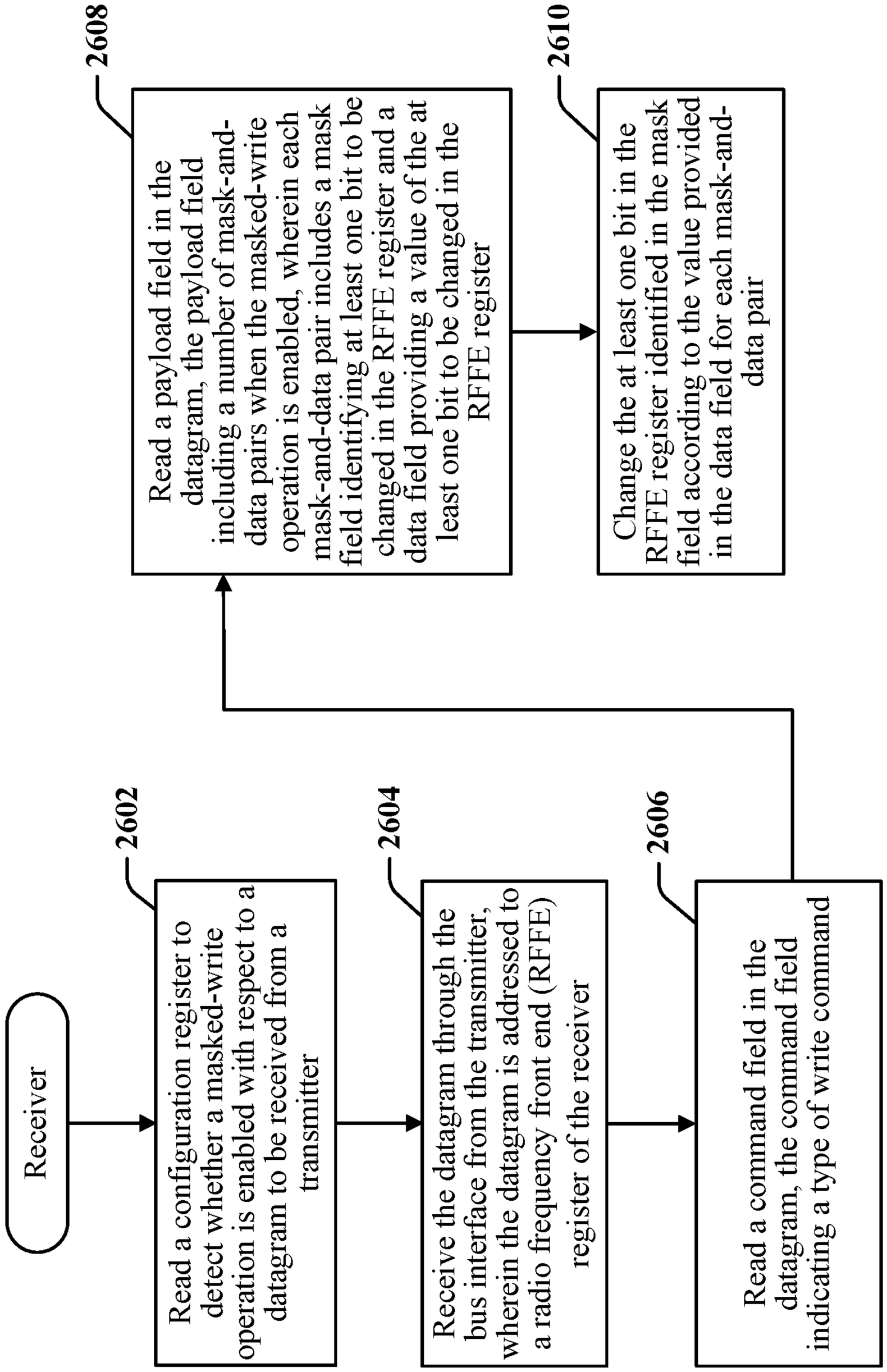
2600
Receiver
2602
Read a configuration register to detect whether a masked-write operation is enabled with respect to a datagram to be received from a transmitter
2604
Receive the datagram through the bus interface from the transmitter, wherein the datagram is addressed to a radio frequency front end (RFFE) register of the receiver
2606
Read a command field in the datagram, the command field indicating a type of write command
2608
Read a payload field in the datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in the RFFE register and a data field providing a value of the at least one bit to be changed in the RFFE register
2610
Change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair

FIG. 26

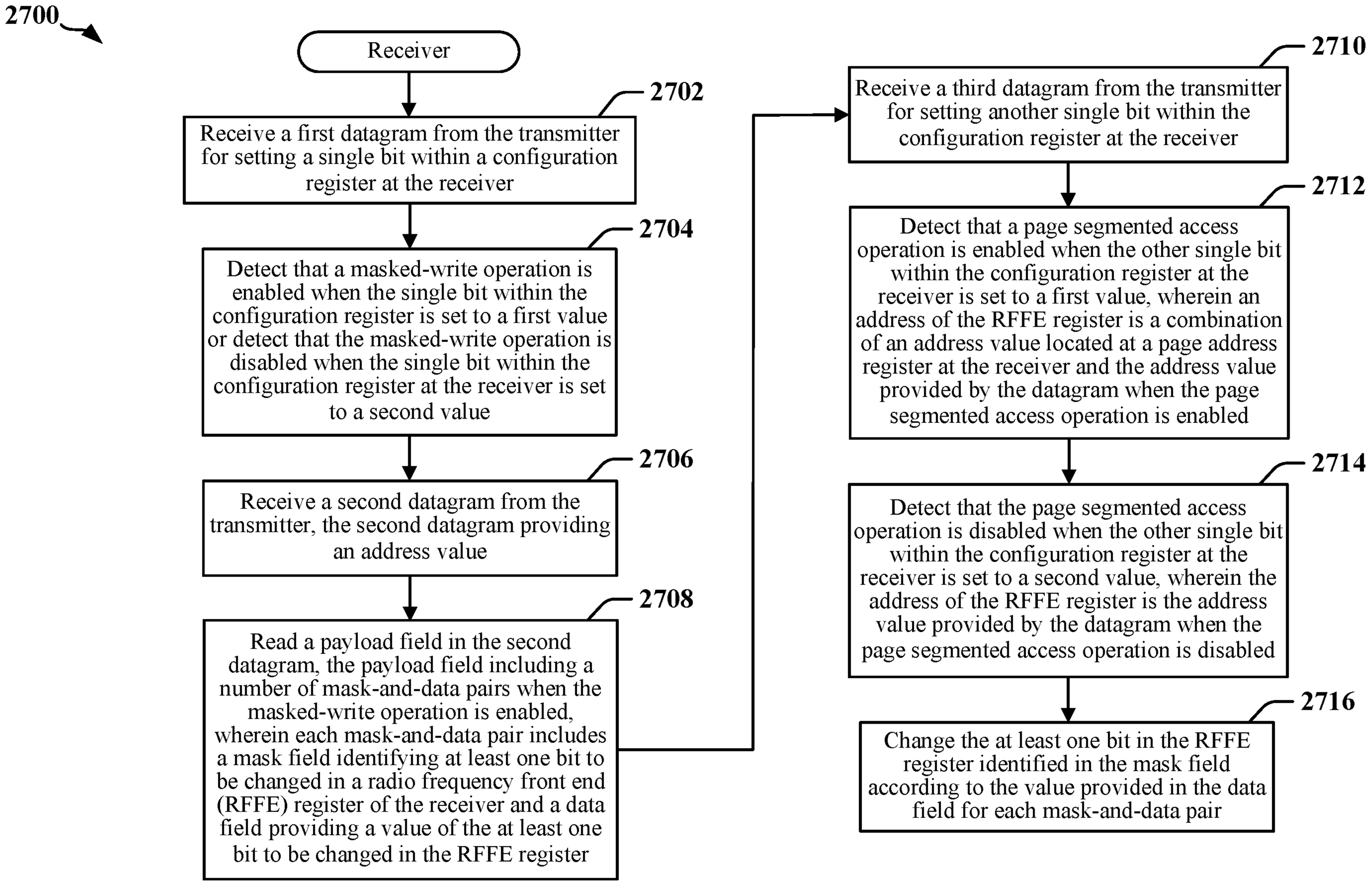
2700
Receiver
2702
Receive a first datagram from the transmitter for setting a single bit within a configuration register at the receiver
2704
Detect that a masked-write operation is enabled when the single bit within the configuration register is set to a first value or detect that the masked-write operation is disabled when the single bit within the configuration register at the receiver is set to a second value
2706
Receive a second datagram from the transmitter, the second datagram providing an address value
2708
Read a payload field in the second datagram, the payload field including a number of mask-and-data pairs when the masked-write operation is enabled, wherein each mask-and-data pair includes a mask field identifying at least one bit to be changed in a radio frequency front end (RFFE) register of the receiver and a data field providing a value of the at least one bit to be changed in the RFFE register
2710
Receive a third datagram from the transmitter for setting another single bit within the configuration register at the receiver
2712
Detect that a page segmented access operation is enabled when the other single bit within the configuration register at the receiver is set to a first value, wherein an address of the RFFE register is a combination of an address value located at a page address register at the receiver and the address value provided by the datagram when the page segmented access operation is enabled
2714
Detect that the page segmented access operation is disabled when the other single bit within the configuration register at the receiver is set to a second value, wherein the address of the RFFE register is the address value provided by the datagram when the page segmented access operation is disabled
2716
Change the at least one bit in the RFFE register identified in the mask field according to the value provided in the data field for each mask-and-data pair

FIG. 27

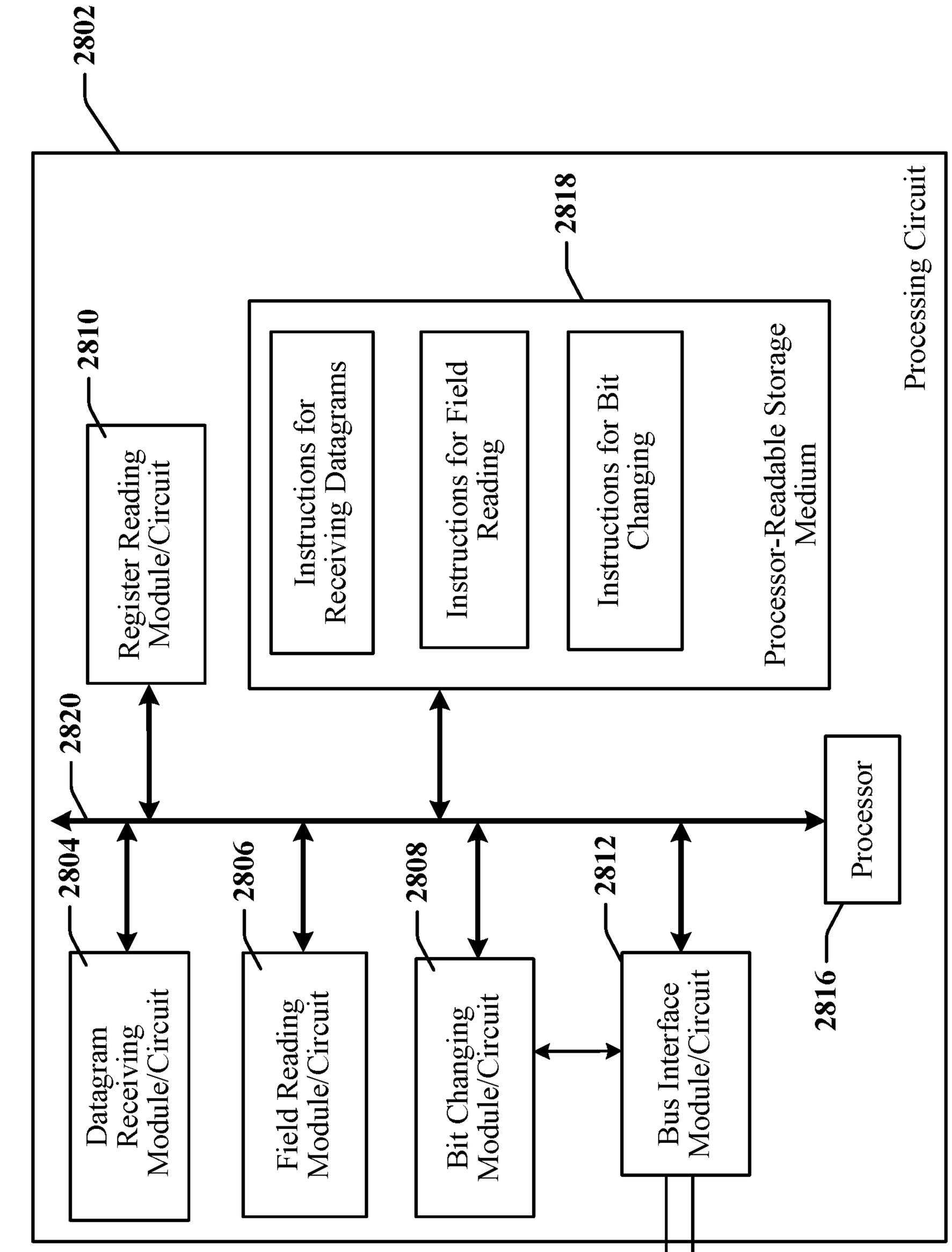
2802
Processing Circuit
2810
Register Reading Module/Circuit
2818
Instructions for Receiving Datagrams
Instructions for Field Reading
Instructions for Bit Changing
Processor-Readable Storage Medium
2820
2804
2806
2808
2812
Processor
2816
Datagram Receiving Module/Circuit
Field Reading Module/Circuit
Bit Changing Module/Circuit
Bus Interface Module/Circuit
2814
2800

FIG. 28

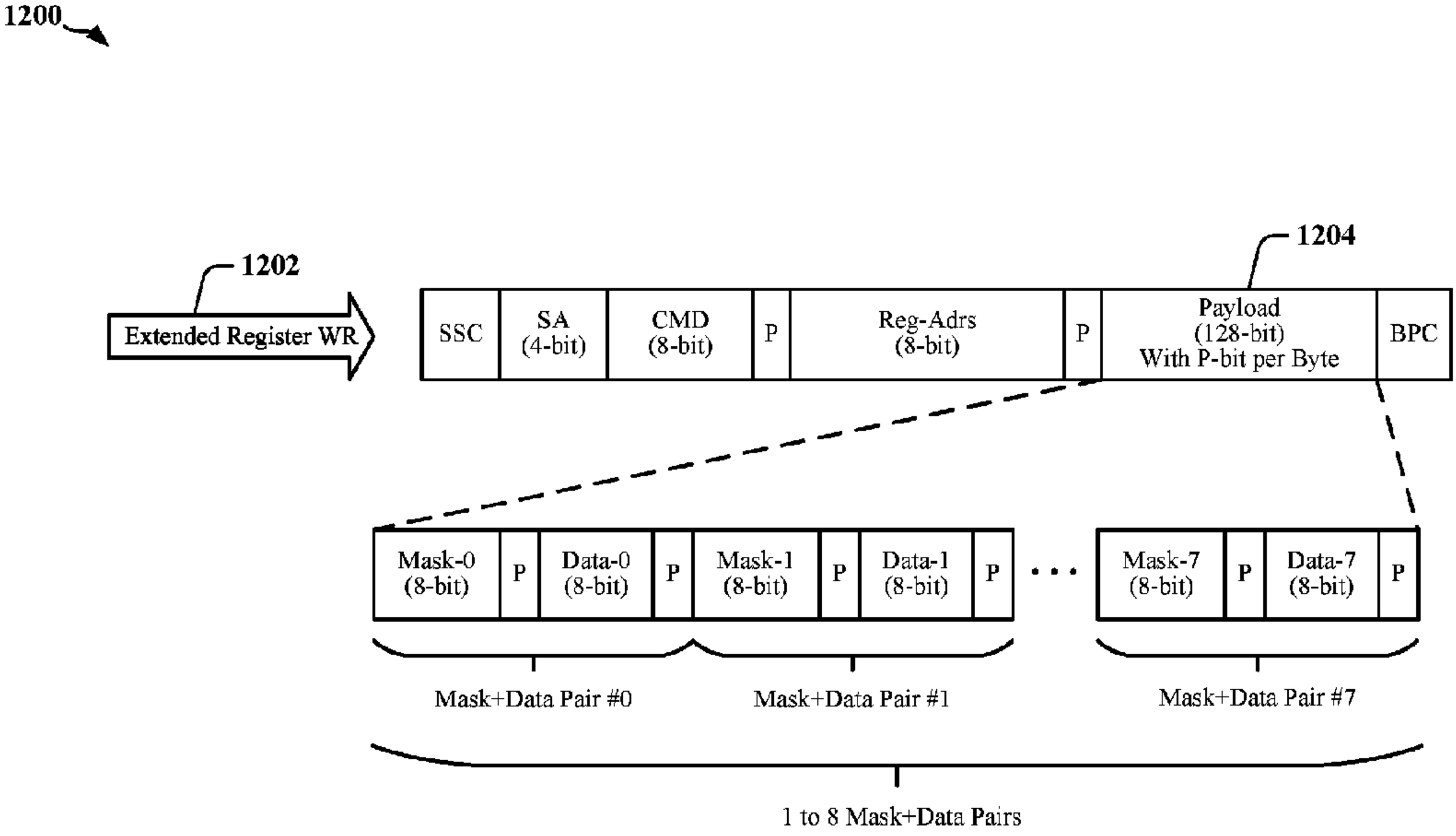
1200
1202
Extended Register WR
1204
SSC
SA (4-bit)
CMD (8-bit)
P
Reg-Adrs (8-bit)
P
Payload (128-bit) With P-bit per Byte
BPC
Mask-0 (8-bit)
P
Data-0 (8-bit)
P
Mask-1 (8-bit)
P
Data-1 (8-bit)
P
Mask-7 (8-bit)
P
Data-7 (8-bit)
P
Mask+Data Pair #0
Mask+Data Pair #1
Mask+Data Pair #7
1 to 8 Mask+Data Pairs

FIG. 12