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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 May 2004 (06.05.2004)

PCT

(10) International Publication Number
WO 2004/038946 A2

(51) International Patent Classification[7]: H04B 1/707

(21) International Application Number:
PCT/US2003/032605

(22) International Filing Date: 15 October 2003 (15.10.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/274,916 22 October 2002 (22.10.2002) US

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200 Mission College Blvd., Mailstop SC4-202, Santa Clara, CA 95052 -8119 (US).

(72) Inventors: **TENE, Eitan**; 21 Vinshel Street, 69413 Tel-Aviv (IL). **SOKOLOV, Dotan**; 11 Shlonski Street, 43592 Ra'anana (IL).

(74) Agent: **COHEN, Mark S.,**; Eitan, Pearl, Latzer & Cohen Zedek, LLP., 10 Rockefeller Plaza, Suite 1001, New York, NY 10020 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 2004/038946 A2

(54) Title: A METHOD TO REDUCE THE NUMBER OF BITS PER SOFT BIT

(57) Abstract: A method and apparatus for reducing an amount of bits of data at at least two stages of data transfer in a communications reception data path between a rake receiver and a channel decoder.

A METHOD TO REDUCE THE NUMBER OF BITS PER SOFT BIT

BACKGROUND OF THE INVENTION

A code division multiple access (CDMA) or a wide-band CDMA (WBCDMA) receiver may comprise a rake receiver, which may include multiple receiving elements, called fingers, which de-spread a received signal. A rake combiner may receive data or information from the fingers to form soft bits, which may then be stored in memory for processing.

The transfer of information in the reception path, after the rake combiner, may comprise processing bits. Along a reception data path in a WBCDMA system, for example, different processing techniques may be performed. For example, the data transmitted may be rearranged to a random order by one or more interleavers. Accordingly, at some point in the reception path, the reception data may be reordered to the correct, original order by using one or more deinterleavers. Frames and blocks of data may be formed and the data may be processed by different processing techniques, such as but not limited to, channel decoding.

The number of bits associated with the data (soft bits) at the rake combiner may be some large number n. The large number of bits may be required for data precision and dynamic range during the different processing procedures. However, processing a large number of bits may be costly and time-consuming in terms of data management, memory storage, and processing hardware and software, for example.

BRIEF DESCRIPTION OF THE DRAWINGS

The subject matter regarded as the invention is particularly pointed out and distinctly claimed in the concluding portion of the specification. The invention, however, both as to organization and method of operation, together with objects, features and advantages thereof, may best be understood by reference to the following detailed description when read with the accompanied drawings in which:

Fig. 1 is a simplified flow chart of a method for reducing an amount of bits of data at one or more stages of data transfer in a communications reception data path, in accordance with an embodiment of the invention;

Fig. 2 is a simplified flow chart of an example of the method of Fig. 1, used with a communications reception data path that may comprise slots, frames and code blocks, in accordance with an embodiment of the invention;

Fig. 3 is a simplified block diagram of an apparatus wherein bits of data are reduced at different stages of data transfer in a communications reception data path, corresponding to the exemplary embodiment of Fig. 2, in accordance with an embodiment of the invention; and

Fig. 4 is a simplified block diagram of a communications system wherein bits of data are reduced at different stages of data transfer in a communications reception data path, in accordance with an embodiment of the invention.

It will be appreciated that for simplicity and clarity of illustration, elements shown in the figures have not necessarily been drawn to scale. For example, the dimensions of some of the elements may be exaggerated relative to other elements for clarity. Further, where considered appropriate, reference numerals may be repeated among the figures to indicate corresponding or analogous elements.

DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the invention. However, it will be understood by those skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known methods, procedures, components and circuits have not been described in detail so as not to obscure the present invention.

Some portions of the detailed description that follows are presented in terms of algorithms and symbolic representations of operations on data bits or binary digital signals within a computer memory. These algorithmic descriptions and representations may be the techniques used by those skilled in the data processing arts to convey the substance of their work to others skilled in the art.

Reference is now made to Fig. 4, which illustrates a communications system 60, in accordance with an embodiment of the invention. Communications system 60 may comprise a base station 70 and a mobile station 71 communicable with one another via a wireless link 69 in accordance with a mutual protocol, such as but not limited to, code division multiple access (CDMA), wideband CDMA (WCDMA) or CDMA2000. Mobile station 71 may comprise a processor 61 adapted to reduce an amount of bits of data at one or more stages of data transfer in a communications reception data path 62. Examples of methods for reducing the amount of bits of data (soft bits), which methods may be implemented by processor 61, are described hereinbelow. A storage medium 59, such as but not limited to, a volatile or non-volatile memory array, e.g., a flash memory, read-only memory (ROM), or electrically erasable read-only memory (EEPROM), may be provided for storing instructions that enable processor 61 to reduce the amount of bits of data in communications reception data path 62.

The following description is for the example of a communications system comprising base station 70 and mobile station 71, where mobile station 71 comprises storage medium 59, processor 61 and communications data path 62 and processor 61 implements methods according to some embodiments of the invention. However, it will be appreciated that some embodiments of the invention are equally applicable to a communications system comprising a mobile station and a base station, where it is the base station that comprises storage medium 59, processor 61 and communications data path 62.

The communications reception data path 62 may comprise, without limitation, a data path between a rake receiver 63 and a channel decoder 64. The data may be collected into different size groups along the communications reception data path 62. For example, the data output from one or more rake combiners 66 may be collected into one or more slots 65. The data from the one or more slots 65 may be collected into one or more frames 67. The data from the one or more frames 67 may be collected into one or more code blocks 68. The size of the group of data may increase from one stage to a subsequent stage. For example, the size of code block 68 may be larger than the size of frame 67 and the size of frame 67 may be larger than the size of slot 65.

Processor 61 may comprise a separate unit in communication with any of the components of communications reception data path 62. Alternatively, processor 61 may comprise therewithin some or all of the components of communications reception data path 62. Still alternatively, any of the components of communications reception data path 62 may comprise therewithin portions or all of processor 61.

Reference is now made to Fig. 1, which illustrates a method for reducing an amount of bits of data (also referred to as soft bits) at one or more stages of data transfer in a communications reception data path, in accordance with an embodiment of the invention. The method may be implemented by system 60 of Fig. 4. The amount of bits of data (soft bits) may be reduced at one, two or more stages of the data transfer (e.g., reception), for example, in the communications reception data path (e.g., between rake receiver 63 and channel decoder 64 of Fig. 4), as indicated at reference numeral 100 in Fig. 1.The method may reduce the amount of bits transferred from one or more rake combiners 66 to one or more slots 66, and/or to one or more frames 67, and/or to one or more code blocks 68 (reference numeral 101). Accordingly, the method may comprise collecting bits of data into groups of data at one or two or more stages of data transfer in the communications reception data path, wherein the group of data increases in size from one stage to a subsequent stage (e.g., from slots to frames to code blocks), and reducing an amount of the bits of data at one or more stages of data transfer in the communications reception data path.

The reducing may comprise performing normalization and/or quantization of bits of one or more stages of data transfer (reference numeral 102). As will be described more in detail with reference to Fig. 2, the method of reducing the number of bits may

comprise calculating a scaling factor (reference numeral 103), and may also comprise calculating a correcting scaling factor (reference numeral 104). The correcting scaling factor may comprise dividing the scaling factor of the current stage of data transfer by one or more scaling factors associated with a preceding stage of data transfer reception. The bits of the stages of data transfer may then be multiplied by the correcting scaling factor (reference numeral 105) per slot. The data in the current stage may not all be necessarily multiplied by the same correcting scaling factor. The bits may then be quantized (reduced in number) such that there are fewer bits in the particular stage of data transfer than a preceding stage of data transfer (reference numeral 106).

Reference is now made to Fig. 2, which illustrates an example of the method of Fig. 1, which may be implemented in the communications reception data path 62 shown in Fig. 4, in accordance with an embodiment of the invention. Fig. 3 is a simplified block diagram of elements of communications reception data path 62 and of elements which processor 61 may comprise to reduce the amount of bits of data at one or more stages of data transfer in communications reception data path 62.

Slots 65 (Fig. 4) may be collected from the output of rake combiner or combiners 66 (reference numeral 200, Fig. 2). The number of bits in the soft bits of the rake combiner 66 may be 16, for example, although the present invention is not limited to this value.

In a first stage, the slot 65 may be normalized and/or quantized (reference numeral 201). Examples of normalization techniques and methods of calculating scaling factors are described more in detail further hereinbelow.

The normalization of the slot soft bits may comprise, without limitation, calculation of a slot scaling factor, multiplication of the slot soft bits by the slot scaling factor, followed by quantization of the soft bits to a number of bits fewer than the bits of the rake combiner. The number of bits in the soft bits of the slots may be 8, for example, although the present invention is not limited to this value.

As seen in Fig. 3, the communications system may comprise, for example, M channels, and for each channel, the number of bits in the slots may be reduced as described above. The data in the slots may be concatenated and transferred to frames (reference numeral 20 in Fig.3).

Referring again to Fig. 2, in a second stage of reducing the number of bits, the frames may be normalized and/or quantized (reference numeral 202). A frame scaling factor may be calculated from the slot scaling factors (calculated from the input root-mean-square (RMS) values of the slots). A slot correcting scaling factor may then be calculated for one or more of the slots combining the frame. For example, the slot correcting scaling factor may be calculated by dividing the frame scaling factor by the slot scaling factor. The soft bits in the frame may then be multiplied by the slot correcting scaling factor of the particular slot. In other words, a soft bit in the frame may be connected to the slot it came from and multiplied by that slot's slot correcting scaling factor. The soft bits of the frames may then be quantized to a number of bits fewer than the bits of the slots. The number of bits in the soft bits of the frames may be 5, for example, although the present invention is not limited to this value.

As seen in Fig. 3, after the frame normalization and quantization, the data may be deinterleaved, wherein the order of the bits may be changed (reference numeral 22). This may be followed by various processing procedures, such as but not limited to, physical channel concatenation 24, deleting 26 of DTX (discontinuous transmission) indication bits, and channel demultiplexing 28. Then for individual channels, there may be further processing procedures, such as but not limited to, radio frame collection 30, deinterleaving 32, wherein the order of the bits may be changed, and deleting 34 of DTX indication bits. The data may then be transferred to code blocks (reference numeral 36).

Referring again to Fig. 2, in a third stage of reducing the number of bits, the code blocks may be normalized and/or quantized (reference numeral 203). A code block scaling factor may be calculated from the frame scaling factors (e.g., calculated from the frames RMS). A frame correcting scaling factor may then be calculated for one or more of the frames in the code block. For example, the frame correcting scaling factor may be calculated by dividing the code block scaling factor by one or more of the frame scaling factors. The soft bits in the code block may then be multiplied by the frame correcting scaling factor of the particular frame, wherein the soft bit may be multiplied by the correcting scaling factor from its own frame. The frame correcting scaling factors may be taken in a cyclic order while multiplying them with the code block soft bits The soft bits of the code blocks may then be quantized to a number of bits fewer than or equal to

the bits of the frames. Soft bits from different frames may be placed in the same level of reference.

The bits of the code blocks may then undergo further processing, as seen in Fig. 3. The processing may comprise, without limitation, inverse rate matching 38, channel decoding 40 (reference numeral 204 in Fig. 2), track block separation and/or code block segmentation 42, and CRC (cyclic redundancy checking) 44.

With the present invention, it may be possible to store two or three or more soft bits within one word, for example, although the invention is not limited to these values.

Different normalization techniques may be used to carry out the invention. The following are some illustrative examples, but the invention is not limited to these examples.

The normalization may be divided into calculating the scaling factor and normalizing over the input soft bits. The scaling factor may be calculated from the input soft bits and from the scaling factors of the previous stage.

Calculating the scaling factor from the input data bits may be performed as follows:

$Input_RMS = \sqrt{\frac{\sum_{i=0}^{N-1}(InputSB[i])^2}{N}}$; N is the number of soft bits in the input.

$scaling_factor = \frac{Desired_RMS}{Input_RMS}$; *Desired RMS* may be constant, for example, 0.5.

This calculation may be done in the slot normalization mentioned hereinabove.

Calculating the scaling factor from the previous stage may be done by using a max/min algorithm as follows:

First pass: $Input_RMS = \sqrt{\frac{\sum RMS_In[k]^2}{M}}$;

M is the number of elements from the previous stage (for example, the number of slots in a frame in the frame normalization and quantization stage, or the number of TTI (transmission time intervals) in the code block normalization and quantization stage.)

RMS In[*k*] is the *Input RMS* of the k slot in the frame normalization and quantization stage or the k frame in the code block normalization and quantization stage.

$MaxRMS = \max_{k\in\{0,..,M-1\}}(RMS_In[k])$; $MinRMS = \min_{k\in\{0,..,M-1\}}(RMS_In[k])$;

$$if\left(\frac{MaxRMS}{MinRMS} > Max2MinRatio\right)$$

Second pass:

$$Input_RMS = \sqrt{\frac{\sum RMS_In[k]^2}{L}};$$ for every k such that

$$RMS_In[k] < Input_RMS * Threshold;$$

L is the number of k for which the equation is true.

Threshold may be constant, for example, 1.3. The *Threshold* may be used to multiply the *Input RMS* calculated in the first pass.

Max2MinRatio may be constant, for example, 7.

This point of the calculation may be reached either from the second pass or from the first pass if no second pass was performed.

$$scaling_factor = \frac{Desired_RMS}{Input_RMS};$$

$$Correct_scaling_factor[k] = \frac{scaling_factor}{SF[k]} \forall k \in \{0,\ldots,M-1\};$$

$SF[k]$ is the scaling factor of the k slot in the frame normalization and quantization stage or the k frame in the code block normalization and quantization stage.

The second pass may be performed in the frame normalization. In the code block normalization, this algorithm may be performed without the second pass.

One purpose of the second pass in the max/min algorithm may be to detect a case where there may be a large difference between powerful soft bits and weak soft bits. This may give more weight to the weak soft bits. This may be done in the frame normalization and quantization stage while taking the RMS of the slots as a measurement for the mean strength of the soft bits in the slot.

This normalization may comprise the following:

Slot normalization: $NormalizedSB[i] = InputSB[i] * scaling_factor; \forall i \in \{0,\ldots,N-1\}$.

Frame normalization:

For (j = 0; j < SlotsInFrame; j++)

For (l = 0; l < SBinSlot; l++)

$$NormalizedSB[j,l] = InputSB[j,l] * correcting_scaling_factor[j]$$

The slots may be arranged in sequential order in the frame before the second deinterleaver (reference numeral 22 in Fig. 3).

The code block normalization may comprise the following:

j = 0

For (l = 0; l < SBinCodeBlock; l++)

$$NormalizedSB[l] = InputSB[l] * correcting_scaling_factor[j]$$
$$j = (j+1) \bmod TTI$$

The frames in the code block frame may be mixed after the 1st deinterleaver (reference numeral 32 in Fig. 3). The order may be that the 1st soft bit may be from frame #0, the 2nd soft bit may be from frame #1, the 3rd soft bit may be from frame #2 and so on. The order of the frames may be comprise the inter column permutation pattern in the 1st deinterleaver.

A summary of the normalization and quantization flow may be as follows:

Slots: $RMS_{Slot}[k] = \sqrt{\frac{\sum SB[j]^2}{N}}$ $SF_{Slot}[k] = \frac{Desired_RMS_{Slot}}{RMS_{Slot}[k]}$

Frames: $RMS_{Frame}[l] = \sqrt{\frac{\sum_k RMS_{Slot}[k]^2}{M}}$ Calculating may use two passes (first pass and second pass). M is the number of slots for which $RMS_{Slot}[k] < RMS_{Frame}[l] * Threshold$, where $RMS_{Frame}[l]$ was calculated in the first pass.

$$SF_{Frame}[l] = \frac{Desired_RMS_{Frame}}{RMS_{Frame}[l]}$$

$$CorrectingSF_{Slot}[k] = \frac{SF_{Frame}[l]}{SF_{Slot}[k]}$$

BlockCode:

$$RMS_{CodeBlock} = \sqrt{\frac{\sum_l RMS_{Frame}[l]^2}{TTI}}$$

$$SF_{CodeBlock} = \frac{Desired_RMS_{CodeBlock}}{RMS_{CodeBlock}}$$

$$CorrectingSF_{Frame}[l] = \frac{SF_{CodeBlock}}{SF_{Frame}[l]}$$

Different quantization techniques may be used to carry out the invention. The following are some illustrative examples, but the invention is not limited to these examples.

The quantization may be performed using quantization levels, which comprise a nonsymmetrical uniform sequence including zero. For example, quantization levels for 3 bits may be: {-4, -3, -2, -1, 0, 1, 2, 3}. The quantization limits may comprise the following:

$$NumOfLevels = 2^{NumOfQuanBits}$$

$$QuanStep = \frac{2}{NumOfLevels}$$

$$QuanLimits[k] = -1 + (k + 0.5) * QuanStep$$

In the example of 3 bits, the quantization limits may be: {-0.875, -0.625, -0.375, -0.125, 0.125, 0.375, 0.625, 0.875}.

The quantization may be done in the following way:

The quantization level k *QuanLevel[k]* may be taken for the normalized bit j *NormBit[j]* if:

(*NormBit[j]* < *QuanLimit[k]*) and (*NormBit[j]* > *QuanLimit[k-1]*) with the exceptions

QuanLevel[1] may be taken if *NormBit[j]* < *QuanLimit[1]* and

QuanLevel[NumOfLevels] may be taken if *NormBit[j]* > *QuanLimit[NumOfLevels – 1]*.

The quantization in the slot quantization may comprise a reduction to 8 bits, in the frame quantization to 5 bits and in the block code quantization to 5 bits, the invention not being limited to these values.

While certain features of the invention have been illustrated and described herein, many modifications, substitutions, changes, and equivalents will now occur to those of ordinary skill in the art. It is, therefore, to be understood that the appended claims are intended to cover all such modifications and changes as fall within the true spirit of the invention.

What is claimed is:

1. A method comprising:

reducing an amount of bits of data at at least two stages of data transfer in a communications reception data path between a rake receiver and a channel decoder.

2. The method according to claim 1, wherein said reducing comprises reducing the amount of bits of data at more than two stages of said data transfer.

3. The method according to claim 1, wherein said reducing comprises reducing the amount of bits of data transferred from at least one rake combiner to at least one slot.

4. The method according to claim 1, wherein said reducing comprises reducing the amount of bits of data transferred from at least one slot to at least one frame.

5. The method according to claim 1, wherein said reducing comprises reducing the amount of bits of data transferred from at least one frame to at least one code block.

6. The method according to claim 1, wherein said reducing comprises performing at least one of normalization and quantization of bits of said at least two stages of data transfer.

7. The method according to claim 6, wherein said performing comprises calculating a scaling factor.

8. The method according to claim 7, wherein said performing further comprises calculating a correcting scaling factor.

9. The method according to claim 8, wherein calculating said correcting scaling factor comprises dividing said scaling factor by a scaling factor associated with a preceding stage of data transfer.

10. The method according to claim 8, further comprising multiplying said bits of data by said correcting scaling factor.

11. The method according to claim 6, wherein said performing comprises quantizing said bits such that there are fewer bits in one stage of data transfer than a preceding stage of data transfer.

12. The method according to claim 7, wherein said performing comprises normalizing bits of said at least two stages of data transfer by calculating said scaling factor as a root-mean-square function of said bits of said at least two stages of data transfer.

13. The method according to claim 1, further comprising concatenating data transferred from a first stage of data transfer to a second stage of data transfer.

14. The method according to claim 1, further comprising deinterleaving data transferred from a first stage of data transfer to a second stage of data transfer.

15. Apparatus comprising:

a processor adapted to reduce an amount of bits of data at at least two stages of data transfer in a communications reception data path between a rake receiver and a channel decoder.

16. The apparatus according to claim 16, wherein said processor is adapted to perform at least one of normalization and quantization of bits of said at least two stages of data transfer.

17. A communications system comprising:

a communications reception data path comprising a rake receiver and a channel decoder;

a processor adapted to reduce an amount of bits of data at at least two stages of data transfer in said communications reception data path; and

a flash memory.

18. The system according to claim 18 wherein said processor is adapted to perform at least one of normalization and quantization of bits of said at least two stages of data transfer.

19. The system according to claim 18 wherein said communications reception data path comprises at least one of a slot, a frame and a code block.

20. An article comprising:

a storage medium for storing instructions that enable a processor to reduce an amount of bits of data at at least two stages of data transfer in a communications reception data path between a rake receiver and a channel decoder.

21. The article according to claim 21, wherein said instructions comprises instructions to perform at least one of normalization and quantization of bits of said at least two stages of data transfer.

22. A method comprising:

collecting bits of data into groups of data at at least two stages of data transfer in a communications reception data path, wherein the group of data increases in size from one stage to a subsequent stage, and reducing an amount of the bits of data at at least one of said stages of data transfer in said communications reception data path.

23. The method according to claim 22, wherein said collecting comprises collecting the bits of data into at least one slot, at least one frame and at least one code block, wherein said at least one code block is larger in size than said at least one frame and said at least one frame is larger in size than said at least one slot.

24. The method according to claim 23, wherein said reducing comprises reducing the amount of bits of data transferred from said at least one slot to said at least one frame.

25. The method according to claim 23, wherein said reducing comprises reducing the amount of bits of data transferred from said at least one frame to said at least one code block.

26. The method according to claim 22, wherein said reducing comprises performing at least one of normalization and quantization of bits at said at least two stages of data transfer.

27. The method according to claim 26, wherein said performing comprises quantizing said bits such that there are fewer bits in one stage of data transfer than a preceding stage of data transfer.

28. A method comprising:

reducing an amount of bits of data at at least one stage of data transfer in a communications reception data path between a rake receiver and a channel decoder.

29. The method according to claim 28, wherein said reducing comprises performing at least one of normalization and quantization of bits of said at least one stage of data transfer.

30. Apparatus comprising:

a processor adapted to reduce an amount of bits of data at at least one stage of data transfer in a communications reception data path between a rake receiver and a channel decoder.

31. The apparatus of claim 30, wherein said processor is adapted to perform at least one of normalization and quantization of bits of said at least one stage of data transfer.

32. A communications system comprising:

a communications reception data path comprising a rake receiver and a channel decoder;

a processor adapted to reduce an amount of bits of data at at least one stage of data transfer in a communications reception data path; and

a flash memory.

33. The system according to claim 32, wherein said processor is adapted to perform at least one of normalization and quantization of bits of said at least one stage of data transfer.

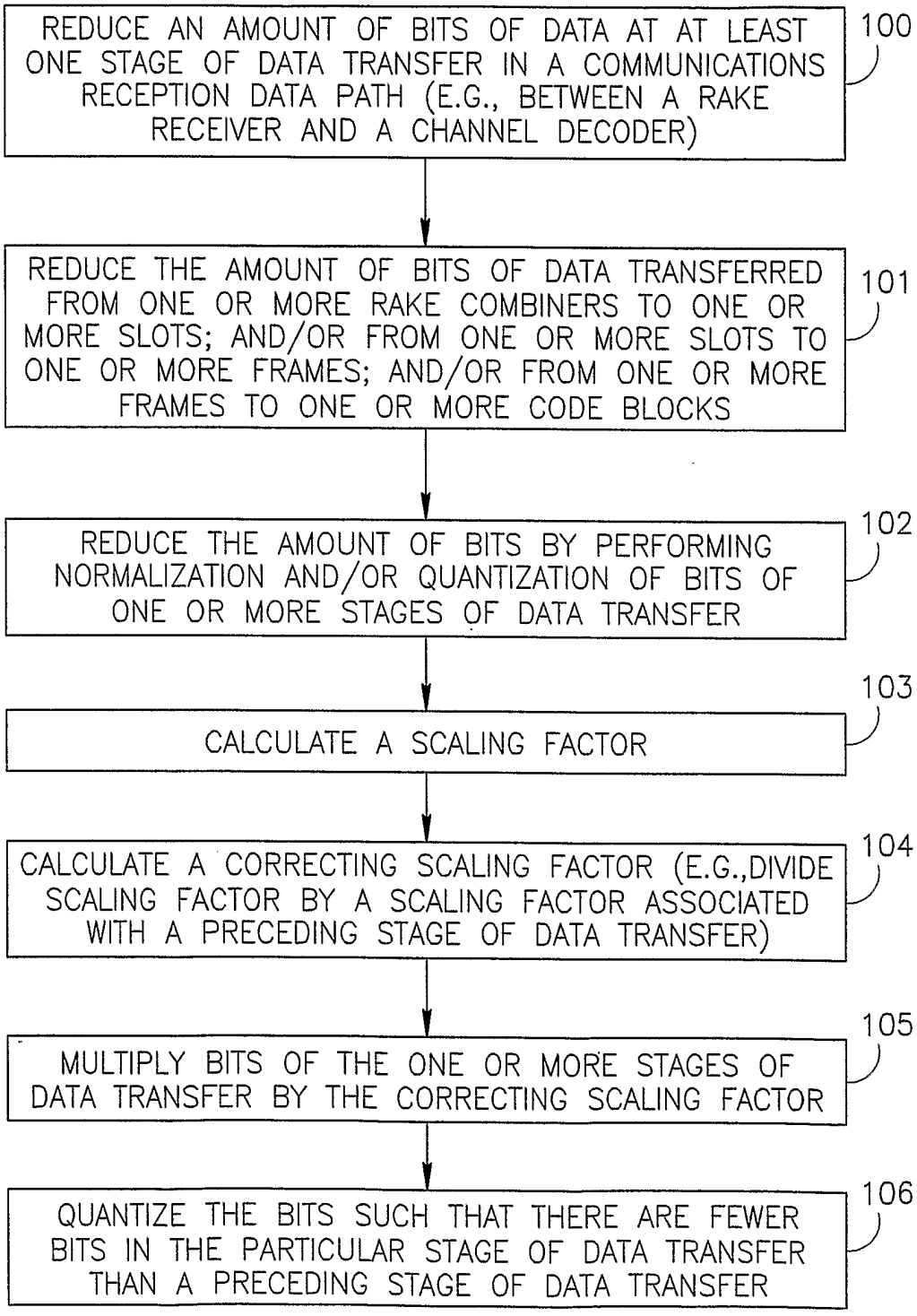
REDUCE AN AMOUNT OF BITS OF DATA AT AT LEAST ONE STAGE OF DATA TRANSFER IN A COMMUNICATIONS RECEPTION DATA PATH (E.G., BETWEEN A RAKE RECEIVER AND A CHANNEL DECODER)
100
REDUCE THE AMOUNT OF BITS OF DATA TRANSFERRED FROM ONE OR MORE RAKE COMBINERS TO ONE OR MORE SLOTS; AND/OR FROM ONE OR MORE SLOTS TO ONE OR MORE FRAMES; AND/OR FROM ONE OR MORE FRAMES TO ONE OR MORE CODE BLOCKS
101
REDUCE THE AMOUNT OF BITS BY PERFORMING NORMALIZATION AND/OR QUANTIZATION OF BITS OF ONE OR MORE STAGES OF DATA TRANSFER
102
CALCULATE A SCALING FACTOR
103
CALCULATE A CORRECTING SCALING FACTOR (E.G.,DIVIDE SCALING FACTOR BY A SCALING FACTOR ASSOCIATED WITH A PRECEDING STAGE OF DATA TRANSFER)
104
MULTIPLY BITS OF THE ONE OR MORE STAGES OF DATA TRANSFER BY THE CORRECTING SCALING FACTOR
105
QUANTIZE THE BITS SUCH THAT THERE ARE FEWER BITS IN THE PARTICULAR STAGE OF DATA TRANSFER THAN A PRECEDING STAGE OF DATA TRANSFER
106

FIG.1

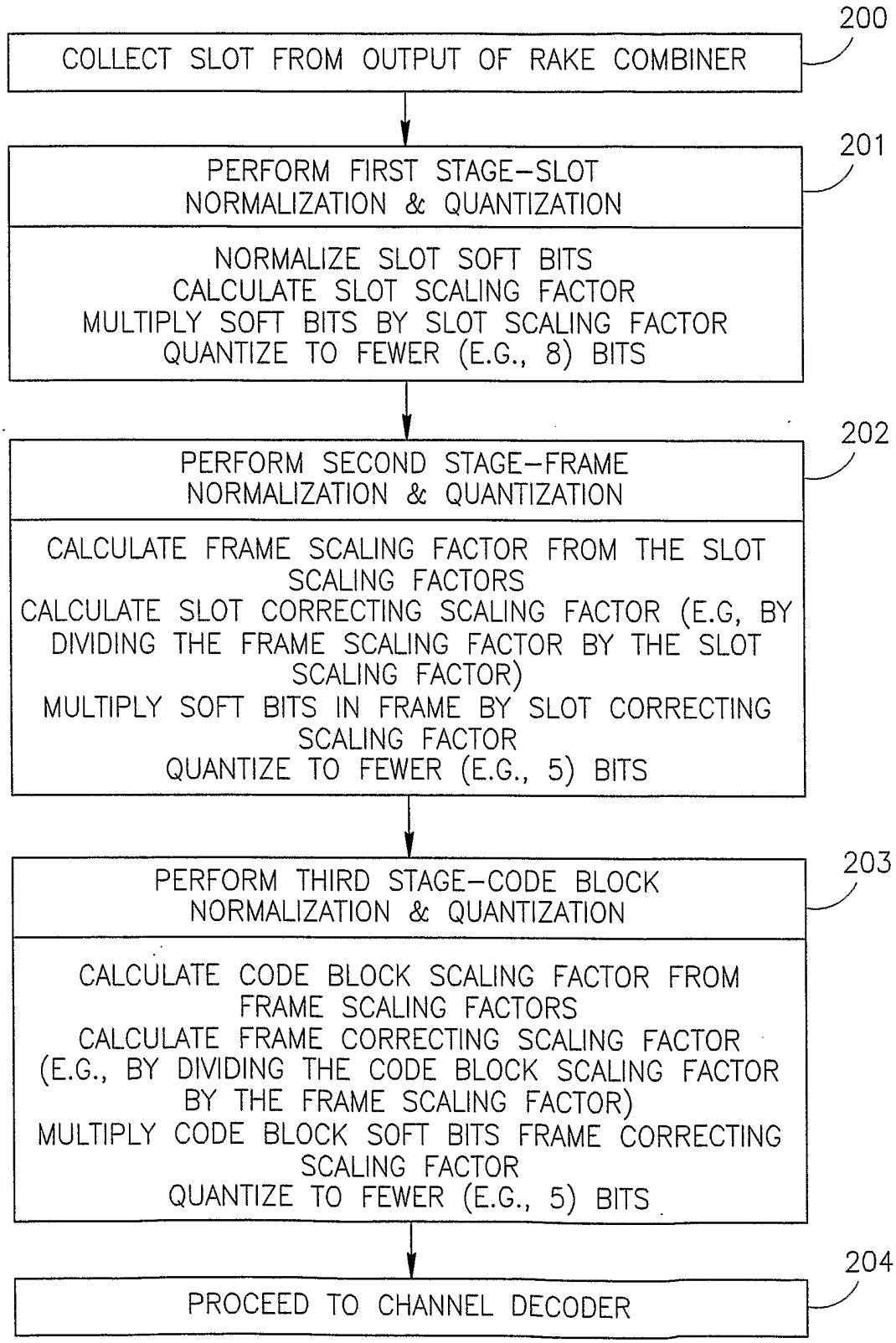
200
COLLECT SLOT FROM OUTPUT OF RAKE COMBINER
201
PERFORM FIRST STAGE–SLOT NORMALIZATION & QUANTIZATION
NORMALIZE SLOT SOFT BITS
CALCULATE SLOT SCALING FACTOR
MULTIPLY SOFT BITS BY SLOT SCALING FACTOR
QUANTIZE TO FEWER (E.G., 8) BITS
202
PERFORM SECOND STAGE–FRAME NORMALIZATION & QUANTIZATION
CALCULATE FRAME SCALING FACTOR FROM THE SLOT SCALING FACTORS
CALCULATE SLOT CORRECTING SCALING FACTOR (E.G, BY DIVIDING THE FRAME SCALING FACTOR BY THE SLOT SCALING FACTOR)
MULTIPLY SOFT BITS IN FRAME BY SLOT CORRECTING SCALING FACTOR
QUANTIZE TO FEWER (E.G., 5) BITS
203
PERFORM THIRD STAGE–CODE BLOCK NORMALIZATION & QUANTIZATION
CALCULATE CODE BLOCK SCALING FACTOR FROM FRAME SCALING FACTORS
CALCULATE FRAME CORRECTING SCALING FACTOR (E.G., BY DIVIDING THE CODE BLOCK SCALING FACTOR BY THE FRAME SCALING FACTOR)
MULTIPLY CODE BLOCK SOFT BITS FRAME CORRECTING SCALING FACTOR
QUANTIZE TO FEWER (E.G., 5) BITS
204
PROCEED TO CHANNEL DECODER

FIG.2

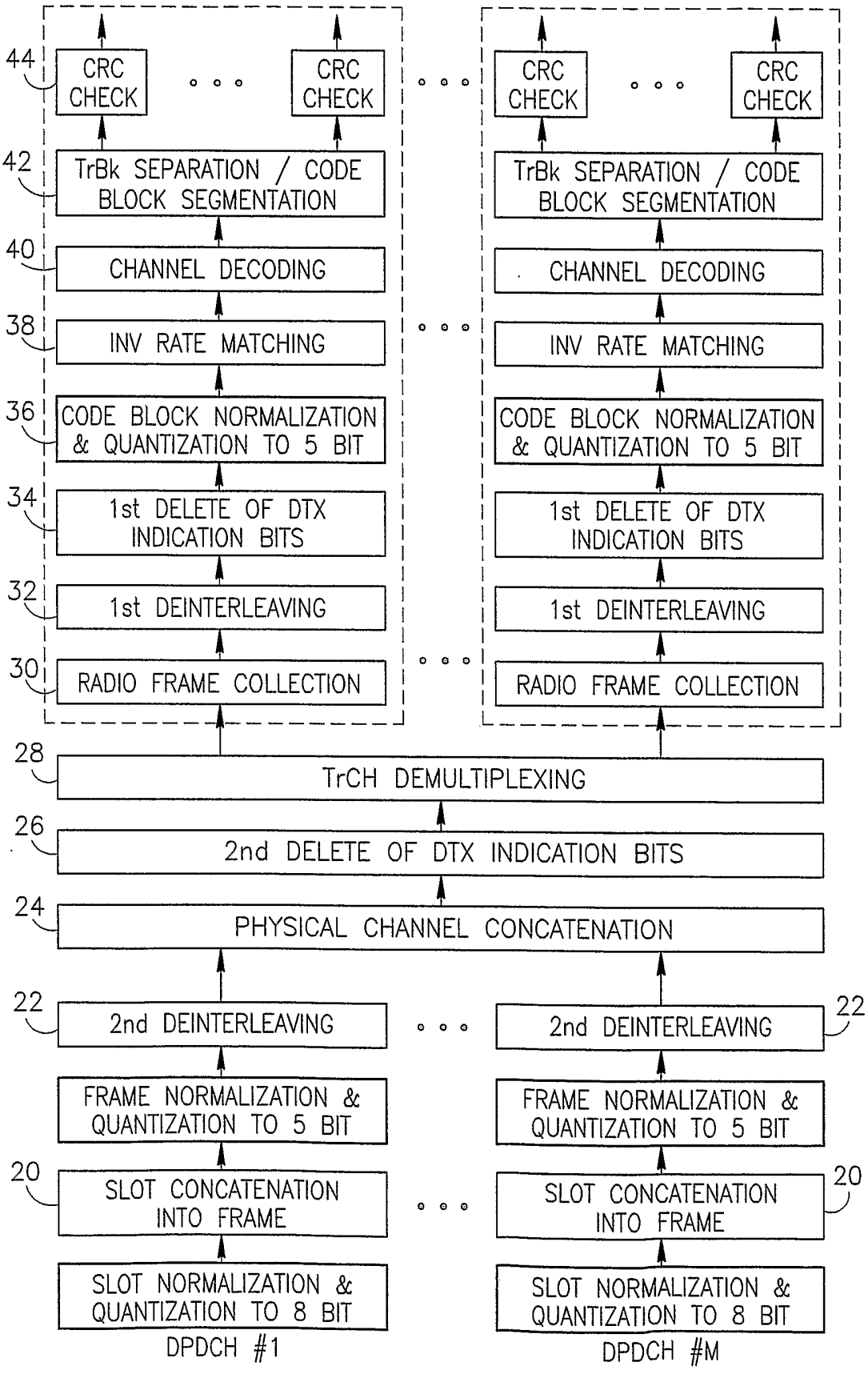
44
CRC CHECK
CRC CHECK
CRC CHECK
CRC CHECK
42
TrBk SEPARATION / CODE BLOCK SEGMENTATION
TrBk SEPARATION / CODE BLOCK SEGMENTATION
40
CHANNEL DECODING
CHANNEL DECODING
38
INV RATE MATCHING
INV RATE MATCHING
36
CODE BLOCK NORMALIZATION & QUANTIZATION TO 5 BIT
CODE BLOCK NORMALIZATION & QUANTIZATION TO 5 BIT
34
1st DELETE OF DTX INDICATION BITS
1st DELETE OF DTX INDICATION BITS
32
1st DEINTERLEAVING
1st DEINTERLEAVING
30
RADIO FRAME COLLECTION
RADIO FRAME COLLECTION
28
TrCH DEMULTIPLEXING
26
2nd DELETE OF DTX INDICATION BITS
24
PHYSICAL CHANNEL CONCATENATION
22
2nd DEINTERLEAVING
2nd DEINTERLEAVING
22
FRAME NORMALIZATION & QUANTIZATION TO 5 BIT
FRAME NORMALIZATION & QUANTIZATION TO 5 BIT
20
SLOT CONCATENATION INTO FRAME
SLOT CONCATENATION INTO FRAME
20
SLOT NORMALIZATION & QUANTIZATION TO 8 BIT
SLOT NORMALIZATION & QUANTIZATION TO 8 BIT
DPDCH #1
DPDCH #M

FIG.3

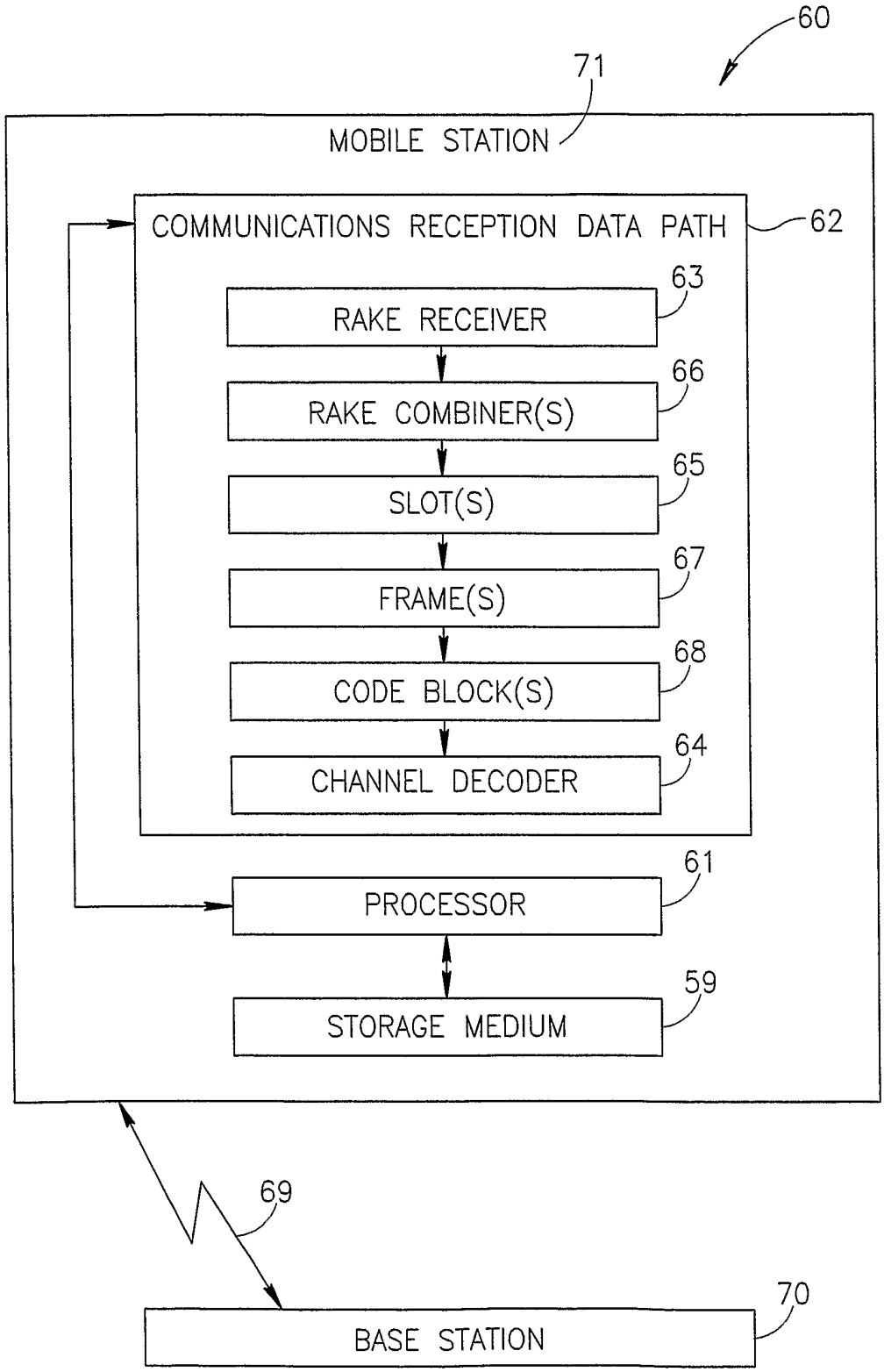
60
71
MOBILE STATION
COMMUNICATIONS RECEPTION DATA PATH
62
63
RAKE RECEIVER
66
RAKE COMBINER(S)
65
SLOT(S)
67
FRAME(S)
68
CODE BLOCK(S)
64
CHANNEL DECODER
61
PROCESSOR
59
STORAGE MEDIUM
69
70
BASE STATION

FIG.4