

# DECODING DEVICE AND METHOD AND SIGNAL TRANSMISSION SYSTEM

## TECHNICAL FIELD

[0001] The present invention relates to the field of data processing technologies,  
5 and in particular, to a decoding device and a decoding method that are applied to an  
LDPC code, and a signal transmission system and a data storage system that use an  
LDPC code.

## BACKGROUND

[0002] A low-density parity-check (full name in English: Low Density Parity  
10 Check; abbreviation in English: LDPC) code was first proposed by Gallager in 1963,  
and is a linear block code including an information bit and a parity bit. Assuming that  
the LDPC code has a code length of  $n$  and a size (a length) of the information bit is  $k$ ,  
a size of the parity bit is  $n-k$ , and a ratio  $R=k/n$  is referred to as a code rate. An LDPC  
code encoding device processes a to-be-transmitted signal according to a generator  
15 matrix (also referred to as a  $G$  matrix) to generate a code word. Generally, some code  
elements of this code word include the to-be-transmitted signal and are also referred  
to as information bits, and some code elements include parity bits used for check.  
Subsequently, the LDPC code encoding device sends the code word to an LDPC code  
decoding device through a communications channel. Then, the LDPC code decoding  
20 device decodes the code word, that is, verifies whether the code word passes check  
decision. If the code word fails to pass the decision, which indicates that an error  
occurs because the code word is interfered by noise in a process of transmission  
through the communications channel, the LDPC code decoding device needs to  
perform an iterative update on the code word by using a check matrix  $H$ , until an  
25 updated code word can pass the decision.

[0003] In a storage controller, generally, a logic/chip or the like is used to

implement an LDPC code decoding device. Therefore, in a specific implementation process, data internally processed by the decoding device is quantized, calculated, and stored, and the quantization process causes a decrease in an error correction capability of the LDPC code decoding device.

5

## SUMMARY

**[0004]** The present application provides a decoding device, where the decoding device can perform processing in a case in which a code element quantization range is exceeded in a code element update process and improve an LDPC code error correction capability of the decoding device.

10 **[0005]** According to an aspect, an embodiment of the present application provides a decoding device, where the decoding device is configured to decode an LDPC code and includes: an acquiring unit, configured to acquire a first code word, for example,  $-9, +7, -12, +4, +7, +10, -11$ , where the first code word is generated by an LDPC code encoding device according to a generator matrix and includes  $M$  first code  
15 elements,  $M$  is a positive integer and  $M$  in the previous example is 7; a processing unit, configured to acquire  $P$  first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix, where for example, when  $n$  is set to 1, three elements on the first row of the check matrix are 1, and therefore,  $P$  is 3; acquire  $P$  first update values corresponding to the  $P$  first code elements; obtain, according to the  $P$  first update  
20 values,  $P$  second update values corresponding to the  $P$  first code elements; determine whether each second update value of the  $P$  second update values exceeds a code element quantization range; keep the  $a^{\text{th}}$  first code element not updated when it is determined that the  $a^{\text{th}}$  second update value of the  $P$  second update values exceeds the code element quantization range; update the  $b^{\text{th}}$  first code element by using the  $b^{\text{th}}$   
25 second update value when it is determined that the  $b^{\text{th}}$  second update value of the  $P$  second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation,  $P$  second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generate a second code word according to the  $P$  second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, where the  
30 check matrix corresponds to the generator matrix and includes  $N$  rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element; and a determining unit, configured to acquire the

second code word and determine whether the second code word is decoded successfully. After acquiring the to-be-decoded first code word, the decoding device needs to perform, on the first code word, processing corresponding to the  $n^{\text{th}}$  row of the check matrix, may update a code element in the first code word in the processing process, and if an updated code element exceeds the code element quantization range, needs to abandon this update. That is, in this aspect, if an update corresponding to a code element exceeds the code element quantization range, the update corresponding to the code element is abandoned.

**[0006]** Optionally, in this aspect, the determining whether each second update value of the P second update values exceeds a code element quantization range may be concurrently determining whether each second update value of the P second update values exceeds the code element quantization range, or may be sequentially determining, from left to right according to a sequence of the P first code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, whether each second update value of the P second update values exceeds the code element quantization range.

**[0007]** Optionally, the decoding device in this aspect is implemented by a circuit, and the acquiring unit, the processing unit, and the determining unit that are included in the decoding device are also implemented by a circuit.

**[0008]** Optionally, in this aspect, the generating a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix includes: generating the second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix and another first code element that is in the first code word and unrelated to the  $n^{\text{th}}$  row of the check matrix.

**[0009]** Optionally, in this aspect, the generating a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix includes: generating the second code word according to P second code elements corresponding to each row of the check matrix.

**[0010]** In a possible design, the processing unit mentioned above includes a storage module; the storage module is configured to store the P first code elements, and is further configured to store the P second update values; the processing unit reads, when determining that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range, the  $a^{\text{th}}$  first code element stored in the storage module; and the processing unit reads, when determining that the  $b^{\text{th}}$

second update value of the P second update values does not exceed the code element quantization range, the  $b^{\text{th}}$  second update value stored in the storage module, so as to update the  $b^{\text{th}}$  first code element. If the foregoing case occurs in which an updated code element exceeds the code element quantization range, a code element before the  
5 update needs to be read to abandon this update. Therefore, the storage module is configured to store the P second update values and the P first code elements, so as to implement that a needed value can be acquired from the storage module no matter whether the updated code element exceeds the code element quantization range, and ensure that the decoding device can flexibly respond to a case in which a second  
10 update value exceeds or does not exceed the code element quantization range in a decoding process.

**[0011]** According to another aspect, an embodiment of the present application provides a decoding device, where the decoding device is configured to decode an LDPC code and includes: an acquiring unit, configured to acquire a first code word,  
15 for example,  $-9, +7, -12, +4, +7, +10, -11$ , where the first code word is generated by an LDPC code encoding device according to a generator matrix and includes M first code elements, M is a positive integer and M in the previous example is 7; a processing unit, configured to acquire P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix, where for example, when three elements on the first row of the  
20 check matrix are 1, P is 3; acquire P first update values corresponding to the P first code elements; obtain, according to the P first update values, P second update values corresponding to the P first code elements; determine whether each second update value of the P second update values exceeds a code element quantization range; keep the P first code elements not updated when it is determined that the  $a^{\text{th}}$  second update  
25 value of the P second update values exceeds the code element quantization range; update the P first code elements by using the P second update values when it is determined that each second update value of the P second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix;  
30 and generate a second code word according to the P second code elements, where the check matrix corresponds to the generator matrix and includes N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , each second update value is a sum of each first update value and a corresponding first code element, and the P second code elements includes at least one of the P first code elements; and a determining unit, configured to acquire the

second code word and determine whether the second code word is decoded successfully. After acquiring the to-be-decoded first code word, the decoding device needs to perform, on the code word, processing corresponding to the  $n^{\text{th}}$  row of the check matrix, and may update a code element in the first code word in the processing process. That is, in this aspect, if an update corresponding to a code element exceeds the code element quantization range, updates corresponding to the P first code elements by using all P second update values corresponding to this row of the check matrix are abandoned.

**[0012]** Optionally, in this aspect, the determining whether each second update value of the P second update values exceeds a code element quantization range may be concurrently determining whether each second update value of the P second update values exceeds the code element quantization range, or may be sequentially determining, from left to right according to a sequence of the P first code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, whether each second update value of the P second update values exceeds the code element quantization range.

**[0013]** Optionally, the decoding device in this aspect is implemented by a circuit, and the acquiring unit, the processing unit, and the determining unit that are included in the decoding device are also implemented by a circuit.

**[0014]** Optionally, in this aspect, the generating a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix includes: generating the second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix and another first code element that is in the first code word and unrelated to the  $n^{\text{th}}$  row of the check matrix.

**[0015]** Optionally, in this aspect, the generating a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix includes: generating the second code word according to P second code elements corresponding to each row of the check matrix.

**[0016]** In a possible design, the processing unit mentioned above includes a storage module; the storage module is configured to store the P first code elements, and is further configured to store the P second update values; the processing unit reads, when determining that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range, the P first code elements stored in the storage module; and the processing unit reads, when determining that each second update value of the P second update values does not exceed the code element

quantization range, the P second update values stored in the storage module, so as to update the P first code elements. If the foregoing case occurs in which an updated code element exceeds the code element quantization range, all P first code elements before the update need to be read to abandon this update. Therefore, the storage module is configured to store the P second update values and the P first code elements, so as to implement that a needed value can be acquired from the storage module no matter whether the updated code element exceeds the code element quantization range, and ensure that the decoding device can flexibly respond to a case in which a second update value exceeds or does not exceed the code element quantization range in a decoding process.

**[0017]** According to another aspect, an embodiment of the present application provides a decoding method, where the decoding method is used to decode an LDPC code and includes: acquiring a first code word, for example,  $-9, +7, -12, +4, +7, +10, -11$ , where the first code word is generated by an LDPC code encoding device according to a generator matrix and includes M first code elements, M is a positive integer and M in the previous example is 3; acquiring P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix, where for example, when three elements on the first row of the check matrix are 1, P is 3; acquiring P first update values corresponding to the P first code elements; obtaining, according to the P first update values, P second update values corresponding to the P first code elements; determining whether each second update value of the P second update values exceeds a code element quantization range; keeping the  $a^{\text{th}}$  first code element not updated when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range; updating the  $b^{\text{th}}$  first code element by using the  $b^{\text{th}}$  second update value when it is determined that the  $b^{\text{th}}$  second update value of the P second update values does not exceed the code element quantization range; obtaining, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generating a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, where the check matrix corresponds to the generator matrix and includes N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element; and determining whether the second code word is decoded successfully. In the decoding process, after the to-be-decoded first code word is acquired, processing corresponding to the  $n^{\text{th}}$  row of

the check matrix needs to be performed on the code word, a code element in the first code word may be updated in the processing process, and if an updated code element exceeds the code element quantization range, this update needs to be abandoned. That is, in this aspect, if an update corresponding to a code element exceeds the code element quantization range, the update corresponding to the code element is abandoned.

**[0018]** Optionally, in this aspect, the determining whether each second update value of the P second update values exceeds a code element quantization range may be concurrently determining whether each second update value of the P second update values exceeds the code element quantization range, or may be sequentially determining, from left to right according to a sequence of the P first code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, whether each second update value of the P second update values exceeds the code element quantization range.

**[0019]** Optionally, in this aspect, the generating a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix includes: generating the second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix and another first code element that is in the first code word and unrelated to the  $n^{\text{th}}$  row of the check matrix.

**[0020]** Optionally, in this aspect, the generating a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix includes: generating the second code word according to P second code elements corresponding to each row of the check matrix.

**[0021]** According to another aspect, an embodiment of the present application provides a decoding method, where the decoding method is used to decode an LDPC code and includes: acquiring a first code word, for example,  $-9, +7, -12, +4, +7, +10, -11$ , where the first code word is generated by an LDPC code encoding device according to a generator matrix and includes M first code elements, M is a positive integer and M in the previous example is 7; acquiring P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix, where for example, when three elements on the first row of the check matrix are 1, P is 3; acquiring P first update values corresponding to the P first code elements; obtaining, according to the P first update values, P second update values corresponding to the P first code elements; determining whether each second update value of the P second update values exceeds a code element quantization range; keeping the P first code elements not updated

when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range; updating the P first code elements by using the P second update values when it is determined that each second update value of the P second update values does not exceed the code element quantization range;

5 obtaining, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generating a second code word according to the P second code elements, where the check matrix corresponds to the generator matrix and includes N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element;

10 element; and determining whether the second code word is decoded successfully. In the decoding process, after the to-be-decoded first code word is acquired, processing corresponding to the  $n^{\text{th}}$  row of the check matrix needs to be performed on the code word, a code element in the first code word may be updated in the processing process. That is, in this aspect, if an update corresponding to a code element exceeds the code

15 element quantization range, updates corresponding to the P first code elements by using all P second update values corresponding to this row of the check matrix are abandoned.

**[0022]** Optionally, in this aspect, the determining whether each second update value of the P second update values exceeds a code element quantization range may

20 be concurrently determining whether each second update value of the P second update values exceeds the code element quantization range, or may be sequentially determining, from left to right according to a sequence of the P first code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, whether each second update value of the P second update values exceeds the code element quantization range.

25 **[0023]** Optionally, in this aspect, the generating a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix includes: generating the second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix and another first code element that is in the first code word and unrelated to the  $n^{\text{th}}$  row of the check matrix.

30 **[0024]** Optionally, in this aspect, the generating a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix includes: generating the second code word according to P second code elements corresponding to each row of the check matrix.

**[0025]** According to another aspect, an embodiment of the present application

provides a signal transmission system, including an LDPC code encoding device and the decoding device mentioned in any one of the foregoing aspects, where the LDPC code encoding device is configured to generate a first code word according to a generator matrix and transmit the first code word to the decoding device through a communications channel. In a process of using the signal transmission system, impact of noise on a to-be-transmitted signal during transmission in the communications channel is taken into account, and therefore, the LDPC code encoding device is used to encode the to-be-transmitted signal into an LDPC code, and the LDPC code is decoded on the decoding device side, so as to ensure that a decoding result of the decoding device includes the original to-be-transmitted signal. In this signal transmission system, the decoding device mentioned in any one of the foregoing aspects is used, which improves accuracy of the decoding device in decoding an LDPC code and also improves reliability of a signal transmitted in the signal transmission system.

**[0026]** According to another aspect, an embodiment of the present application provides a data storage system, including an LDPC code encoding device, the decoding device mentioned in any one of the foregoing aspects and a storage medium, where the LDPC code encoding device is configured to generate a first code word according to a generator matrix and store the first code word in the storage medium, and the decoding device is configured to acquire the first code word from the storage medium. In a process of using the data storage system, a signal change caused when a to-be-stored signal is stored in the storage medium for an excessively long period or affected by another factor is taken into account, and therefore, the LDPC code encoding device is used to encode the to-be-stored signal into an LDPC code and then store the LDPC code in the storage medium, the LDPC code is read and decoded on the decoding device side, so as to ensure that a decoding result of the decoding device includes the original to-be-stored signal. In this data storage system, the decoding device mentioned in any one of the foregoing aspects is used, which improves accuracy of the decoding device in decoding an LDPC code and also improves reliability of a signal stored in the data storage system.

**[0027]** In comparison with the prior art, in a code word update process in solutions provided in the present application, a decoding device performs processing in a case in which a value of an updated code element exceeds a code element quantization range, which prevents the decoding device from directly quantizing the

updated code element that exceeds the code element quantization range, and improves an error correction capability of the decoding device in a decoding process.

## **BRIEF DESCRIPTION OF DRAWINGS**

**[0028]** To describe the technical solutions in the embodiments of the present invention or in the prior art more clearly, the following briefly introduces the accompanying drawings required for describing the embodiments. Apparently, the accompanying drawings in the following description show some embodiments of the present invention, and persons of ordinary skill in the art may still derive other drawings from these accompanying drawings without creative efforts.

**[0029]** FIG. 1 is a diagram of an LDPC code encoding and decoding architecture according to an embodiment of the present invention;

**[0030]** FIG. 2 is a diagram of another LDPC code encoding and decoding architecture according to an embodiment of the present invention;

**[0031]** FIG. 3 is a schematic diagram of a composite structure of a decoding device according to device embodiment 1 of the present invention;

**[0032]** FIG. 4 is a schematic diagram of a composite structure of a decoding device according to device embodiment 2 of the present invention;

**[0033]** FIG. 5 is a schematic diagram of a composite structure of a decoding device according to device embodiment 3 of the present invention;

**[0034]** FIG. 6 is a Tanner diagram corresponding to a check matrix applied in decoding devices according to device embodiment 3 and device embodiment 4 of the present invention;

**[0035]** FIG. 7 is a schematic diagram of a composite structure of a decoding device according to device embodiment 4 of the present invention;

**[0036]** FIG. 8 is a schematic flowchart of a decoding method according to method embodiment 1 of the present invention;

**[0037]** FIG. 9 is a schematic flowchart of a decoding method according to method embodiment 2 of the present invention;

**[0038]** FIG. 10 is a schematic flowchart of a decoding method according to method embodiment 3 of the present invention; and

**[0039]** FIG. 11 is emulation results of a decoding device according to the present invention and a conventional decoding device.

## DESCRIPTION OF EMBODIMENTS

[0040] To make the objectives, technical solutions, and advantages of the embodiments of the present invention clearer, the following clearly and completely describes the technical solutions in the embodiments of the present invention with reference to the accompanying drawings in the embodiments of the present invention. Apparently, the described embodiments are some but not all of the embodiments of the present invention. All other embodiments obtained by persons of ordinary skill in the art based on the embodiments of the present invention without creative efforts shall fall within the protection scope of the present invention.

[0041] Throughout this specification, the term "low-density parity-check code" (full name in English: Low Density Parity Check Code) refers to a type of linear error-correcting code and is usually used for transmitting a signal through a communications channel or for storing data in a storage medium in the communications field.

[0042] Throughout this specification, the term "code word" (full name in English: Code Word) refers to a signal including a group of code elements (full name in English: Code Element) transmitted through a communications channel or stored in a storage device. For example, in a case in which a code word of an LDPC code has a length of 7, the code word includes 7 code elements.

[0043] Throughout this specification, the term "code element quantization range" may also be referred to as a variable node (Variable Node) quantization range. In a process of decoding an LDPC code, each code element of a to-be-decoded code corresponds to one variable node, and each row of a check matrix corresponds to one check node (Check Node). For details, refer to a Tanner diagram in FIG. 6. A code element quantization range is also a size of address space allocated by an LDPC code decoding device to each code element. For example, if address space allocated to each code element is 6 bits, a code element quantization range is from  $-31$  to  $+31$ .

### **LDPC code encoding and decoding architecture in embodiments of the present invention**

[0044] FIG. 1 depicts an LDPC code encoding and decoding architecture

according to an embodiment of the present invention. Generally, for a group of to-be-transmitted signals (carrying actual information, also referred to as an information bit), an LDPC code encoding device first generates one code word for the to-be-transmitted signals according to a generator matrix, also referred to as a G matrix, and then the LDPC code encoding device sends the code word to an LDPC code decoding device through a communications channel. Generally, the code word is affected by noise in a process of transmission through the communications channel, and therefore, an error may occur in the code word in the process of the transmission through the communications channel. Therefore, the LDPC code decoding device has an error correction and checking capability for the code word transmitted through the communications channel. For example, a group of to-be-transmitted signals is 1, 0, 1, 1. A code word generated by the LDPC code encoding device according to the to-be-transmitted signals and the generator matrix is 1, 0, 1, 1, 0, 0, 1. After the LDPC code encoding device transmits the code word to the LDPC code decoding device by using an electric potential signal, electric potential signals received by the LDPC code decoding device are -9 volt, +7 volt, -12 volt, +4 volt, +7 volt, +10 volt, -11 volt. In the present invention, the electric potential signal sent to the LDPC code decoding device through a communications channel is also referred to as a code word, where a positive potential corresponds to 0 and a negative potential corresponds to 1. Therefore, after being transmitted through the communications channel, the fourth code element 1 turns into +4 volt corresponding to 0, that is, an error occurs in a process of transmitting the fourth code element of the code word through the communications channel, and the LDPC code decoding device needs to correct the error according to a check matrix corresponding to the generator matrix used by the LDPC code encoding device, so as to acquire the correct code word 1, 0, 1, 1, 0, 0, 1.

**[0045]** FIG. 2 depicts a use scenario of another LDPC code encoding and decoding architecture according to an embodiment of the present invention. According to the foregoing example, after receiving to-be-stored signals 1, 0, 1, 1, the LDPC code encoding device generates a code word 1, 0, 1, 1, 0, 0, 1 according to the to-be-stored signals and a generator matrix, and stores the code word in a storage medium. When the LDPC code decoding device accesses the storage medium, the storage medium usually records data by using an electric potential signal. For

example, electric potential signals acquired from the storage medium by the LDPC code decoding device are -9 volt, +7 volt, -12 volt, +4 volt, +7 volt, +10 volt, -11 volt. Due to aging or another factor of the storage medium, an error occurs in the fourth code element in a storage process, and the LDPC code decoding device needs to correct the error according to a check matrix corresponding to the generator matrix used by the LDPC code encoding device, so as to acquire the correct code word 1, 0, 1, 1, 0, 0, 1.

**[0046]** In practice, generally, the LDPC code encoding device and the LDPC code decoding device perform execution by using a dedicated hardware device, for example, an application-specific integrated circuit (full name in English: Application Specific Integrated Circuit; abbreviation in English: ASIC), a field programmable gate array (full name in English: Field Programmable Gate Array; abbreviation in English: FPGA), or the like. An electric potential signal transmitted to the LDPC code decoding device through the communications channel is usually a floating point number. In addition, due to existence of noise, a value range of the electric potential signal cannot be determined. Therefore, generally, in design, the LDPC code decoding device quantizes potential values of a code word that is acquired by the LDPC code decoding device or used in an error correction process. For example, when 6 bits are used for quantization, a value range is -31 to +31, a value greater than +31 is set to +31, and a value less than -31 is set to -31. There are multiple quantization methods in practice and a quantization result is not necessarily an integer, but may also include an integer part and a fractional part. However, according to a quantity of bits of an allocated quantization value in design, the quantization result has a fixed value range, also referred to as a quantization range. Generally, a variable node and a check node each have their respective quantization ranges. In this document of the present application, a code element quantization range is also a variable node quantization range.

### **Device embodiment 1**

**[0047]** Device embodiment 1 provides a decoding device 200, where the decoding device 200 is configured to decode an LDPC code. FIG. 3 shows a schematic diagram of a composite structure of the decoding device 200. The device 200 includes: an

acquiring unit 202, a processing unit 204, and a determining unit 206.

**[0048]** The acquiring unit 202 is configured to acquire a first code word, where the first code word is generated by an LDPC code encoding device according to a generator matrix and includes M first code elements and M is a positive integer. For details, refer to the LDPC code encoding and decoding architecture mentioned above. The code word acquired by the acquiring unit 202 may be transmitted through a communications channel, or may be read from a storage medium. The first code element herein is any code element in the first code word.

**[0049]** The processing unit 204 is configured to acquire P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquire P first update values corresponding to the P first code elements; obtain, according to the P first update values, P second update values corresponding to the P first code elements; sequentially determine whether each second update value of the P second update values exceeds a code element quantization range; keep the  $a^{\text{th}}$  first code element not updated when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range; update the  $b^{\text{th}}$  first code element by using the  $b^{\text{th}}$  second update value when it is determined that the  $b^{\text{th}}$  second update value of the P second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generate a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, where the check matrix corresponds to the generator matrix and includes N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element.

**[0050]** The determining unit 206 is configured to acquire the second code word generated by the determining processing unit 204 and determine whether the second code word is decoded successfully.

$$\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{vmatrix}$$

**[0051]** In this embodiment, for example, the foregoing check matrix is used by the decoding device, that is, N is equal to 3, and the first code word acquired by the acquiring unit 202 is -9, +7, -12, +4, +7, +10, -11, that is, M is equal to 7. The

processing unit 204 generates, according to a code element that is in the first code word and related to the first row of the check matrix, an update value that is corresponding to the code element that is in the code word and related to the first row of the check matrix, that is,  $n$  is equal to 1. It should be noted that, during actual operation, the processing unit 204 may perform processing according to a sequence from the first row, the second row, to the third row of the check matrix, or may perform processing according to another sequence, where a value of  $n$  herein may be any positive integer less than or equal to a quantity of rows of the check matrix. After acquiring code elements related to the first row of the check matrix, namely, the 1<sup>st</sup> code element  $-9$ , the 2<sup>nd</sup> code element  $+7$ , and the 4<sup>th</sup> code element  $+4$ , where the first code elements include  $-9$ ,  $+7$ , and  $+4$ , the processing unit 204 generates an update value  $+4$  corresponding to the 1<sup>st</sup> code element in the first code word, an update value  $-4$  corresponding to the 2<sup>nd</sup> code element, and an update value  $-7$  corresponding to the 4<sup>th</sup> code element, that is,  $P$  is equal to 3, and the first update values include  $+4$ ,  $-4$ , and  $-7$ . A symbol of a first update value corresponding to each first code element of the  $P$  first code elements is a product of symbols of other first code elements except this first code element, and an absolute value of the first update value corresponding to each first code element of the  $P$  first code elements is an absolute value of a first code element with a smallest absolute value (if this first code element itself is the first code element with the smallest absolute value, the absolute value of the first update value corresponding to this first code element is set to an absolute value of a first code element with a second smallest absolute value). For example, if first code elements include a code element A, a code element B, and a code element C, a symbol of an update value corresponding to the code element A is a product of symbols of the code element B and the code element C, and an absolute value of the update value corresponding to the code element A is a smaller absolute value of absolute values of the code element B and the code element C.

**[0052]** After obtaining the update values corresponding to the code elements that are in the first code word and related to the first row of the check matrix, the processing unit 204 adds each first update value to its corresponding first code element to obtain a second update value, that is, second update values include  $-5$  (a sum of  $+4$  and  $-9$ ),  $+3$  (a sum of  $+7$  and  $-4$ ), and  $-3$  (a sum of  $+4$  and  $-7$ ).

**[0053]** After acquiring the second update values, the processing unit 204 sequentially determines whether each second update value exceeds the code element

quantization range; if the processing unit 204 determines that the  $a^{\text{th}}$  second update value exceeds the code element quantization range, the processing unit 204 does not update the  $a^{\text{th}}$  first code element by using the  $a^{\text{th}}$  second update value, that is, keeps the  $a^{\text{th}}$  first code element not updated; and if the processing unit 204 determines that the  $b^{\text{th}}$  second update value does not exceed the code element quantization range, the processing unit 204 updates the  $b^{\text{th}}$  first code element by using the  $b^{\text{th}}$  second update value.

**[0054]** According to the foregoing example, because  $P$  is equal to 3 in this example, a value of  $a$  may be 1, 2, or 3 and corresponds to a second update value  $-5$ ,  $+3$ , or  $-3$ , respectively. For example,  $a$  is set to 1, and the processing unit 204 determines whether the second update value  $-5$  exceeds the code element quantization range; if the second update value  $-5$  exceeds the code element quantization range,  $-9$  is not updated by using  $-5$ , that is, the code element  $-9$  keeps unchanged; if the second update value  $-5$  does not exceed the code element quantization range,  $-9$  is updated by using  $-5$ , that is, the code element  $-9$  is updated to  $-5$ . The processing unit 204 sequentially processes the first code elements  $-9$ ,  $+7$ , and  $+4$  according to the method. For example, the code element quantization range in this embodiment is from  $-15$  to  $+15$ . Then, none of the second update values exceeds the code element quantization range. The processing unit 204 generates second code elements after completing the processing corresponding to the first row of the check matrix. The second code elements include  $-5$ ,  $+3$ , and  $-3$ . The generation of each code element of the second code elements is to maintain a corresponding first code element (if a corresponding second update value exceeds the code element quantization range) or to update a corresponding first code element by using a corresponding second update value (if the corresponding second update value does not exceed the code element quantization range).

**[0055]** The processing unit 204 generates the second code word according to the second code elements. Mainly two methods for generating the second code word are shown as follows:

**[0056]** Method 1: A second code word is generated by updating, to a second code element, a code element (that is, the first code element) that is in the first code word

-9, +7, -12, +4, +7, +10, -11 and related to the  $n^{\text{th}}$  ( $n$  is equal to 1 in this example) row of the check matrix, and a first code element that is in the first code word and unrelated to the  $n^{\text{th}}$  row of the check matrix is reserved, that is, the second code word is -5, +3, -12, -3, +7, +10, -11.

5    **[0057]**    Method 2: A first temporary code word is generated by updating, to a second code element, a first code element that is in the first code word and related to the  $n^{\text{th}}$  ( $n$  is equal to 1 in this example) row of the check matrix, and a first code element that is in the first code word and unrelated to the first row of the check matrix is reserved, that is, the first temporary code word is -5, +3, -12, -3, +7, +10, -11.

10   **[0058]**    A second temporary code word is generated by updating code elements that are in the first temporary code word and related to the second row of the check matrix. Code elements -12, +4, and +7 that are in the first code word and related to the second row of the check matrix are first acquired. Third update values including +4, -7, and -4 that are corresponding to the code elements that are in the first code  
15   word and related to the second row of the check matrix are generated. Fourth update values are generated by adding +4, -7, and -4 to the code elements -12, -3, and +7 that are in the first temporary code word and related to the second row of the check matrix, where the fourth update values include -8, -10, and +3. Whether any one of -8, -10, and +3 exceeds the code element quantization range is determined. If any  
20   one of the fourth update values exceeds the code element quantization range, the corresponding code element in the first temporary code word is not updated by using the fourth update value. If none of the fourth update values exceeds the code element quantization range, the first temporary code word -5, +3, -12, -3, +7, +10, -11 is updated by using the fourth update values -8, -10, and +3, so as to obtain a second  
25   temporary code word -5, +3, -8, -10, +3, +10, -11.

30   **[0059]**    A second code word is generated by updating code elements that are in the second temporary code word and related to the third row of the check matrix. Code elements +4, +10, and -11 that are in the first code word and related to the third row of the check matrix are first acquired. Fifth update values including -10, -4, and +4 that are corresponding to the code elements that are in the first code word and related to the third row of the check matrix are generated. Sixth update values including -20, +6, and -7 are generated by adding -10, -4, and +4 to the code elements -10, +10,

and  $-11$  that are in the second temporary code word and related to the third row of the check matrix. Whether any one of  $-20$ ,  $+6$ , and  $-7$  exceeds the code element quantization range is determined. If any one of the sixth update values exceeds the code element quantization range, the corresponding code element in the second temporary code word is not updated by using the sixth update value. If none of the sixth update values exceeds the code element quantization range, the second temporary code word  $-5, +3, -8, -10, +3, +10, -11$  is updated by using the sixth update values  $-20, +6$ , and  $-7$ . A second code word  $-5, +3, -8, -10, +3, +6, -7$  is obtained in this example because  $-20$  exceeds the code element quantization range.

**[0060]** It should be noted that, in method 2, the second code word is obtained by processing the first code word according to a sequence from the first row, the second row, to the third row of the check matrix, but in an actual use process of method 2, the first code word may be processed according to another sequence of the check matrix. For example, the first row may be first processed to obtain the first temporary code word, then the third row is processed to obtain the second temporary code word, and then the second row is processed to obtain the second code word. There may be multiple processing sequences. It is only required that, in the process of generating the second code word by using the first code word, each row of the check matrix is processed correspondingly and the processing corresponding to each row of the check matrix is performed only once.

**[0061]** Optionally, the processing unit 204 includes a storage module, configured to store the  $P$  first code elements, and is further configured to store the  $P$  second update values. When determining that the  $a^{\text{th}}$  second update value of the  $P$  second update values exceeds the code element quantization range, the processing unit 204 reads the  $a^{\text{th}}$  first code element stored in the storage module; that is, if a second update value exceeds the code element quantization range, the processing unit 204 does not update a corresponding first code element by using the second update value, and therefore reads, from the storage module, the first code element corresponding to the second update value, so as to acquire the second code elements. When determining that the  $b^{\text{th}}$  second update value of the  $P$  second update values does not exceed the code element quantization range, the processing unit 204 reads the  $b^{\text{th}}$  second update value stored in the storage module, so as to update the  $b^{\text{th}}$  first code element; that is, if

a second update value does not exceed the code element quantization range, the processing unit 204 updates a corresponding first code element by using the second update value, and therefore reads the second update value from the storage module and replaces the corresponding first code element, so as to acquire the second code elements.

**[0062]** According to the foregoing example, the storage module stores the first code elements  $-9$ ,  $+7$ , and  $+4$ , and the storage module further stores the second update values  $-5$ ,  $+3$ , and  $-3$ . If the processing unit 204 determines that a second update value exceeds the code element quantization range, the processing unit 204 reads a corresponding first code element; if the processing unit 204 determines that a second update value does not exceed the code element quantization range, the processing unit 204 reads a corresponding second update value. In an example in which the processing unit 204 generates the second code word by using method 1, the processing unit 204 determines that none of the second update values  $-5$ ,  $+3$ , and  $-3$  exceeds the code element quantization range, and therefore reads the second update values to update the first code word and generate a second code word  $-5$ ,  $+3$ ,  $-12$ ,  $-3$ ,  $+7$ ,  $+10$ ,  $-11$ .

**[0063]** The processing unit 204 sends the second code word to the determining unit 206, where the determining unit 206 is configured to determine whether the second code word is decoded successfully.

**[0064]** In a case in which the processing unit 204 generates the second code word by using method 1 mentioned above, if the determining unit 206 determines that the second code word  $-5$ ,  $+3$ ,  $-12$ ,  $-3$ ,  $+7$ ,  $+10$ ,  $-11$  is decoded successfully, the decoding device 200 continues to perform an update corresponding to another row of the check matrix and generate a second code word corresponding to the row of the check matrix. Herein, the second row is used as an example. After an update corresponding to the second row of the check matrix is complete, an obtained second code word is  $-5$ ,  $+3$ ,  $-8$ ,  $-10$ ,  $+3$ ,  $+10$ ,  $-11$ . If the determining unit 206 determines that the second code word  $-5$ ,  $+3$ ,  $-8$ ,  $-10$ ,  $+3$ ,  $+10$ ,  $-11$  is decoded successfully, the decoding device 200 continues to perform an update corresponding to another row of the check matrix and generate a second code word corresponding to the row of the check matrix.

Herein, the third row is used as an example. After an update corresponding to the third row of the check matrix is complete, an obtained second code word is  $-5, +3, -8, -10, +3, +6, -7$ . If the determining unit 206 determines that the second code word  $-5, +3, -8, -10, +3, +6, -7$  is decoded successfully, the decoding device 200 uses, as a decoding result, a result  $1, 0, 1, 1, 0, 0, 1$  of mapping  $-5, +3, -8, -10, +3, +6, -7$ . Generally, in a case in which the processing unit 204 generates the second code word by using method 1 mentioned above, the decoding device 200 needs to determine that the second code word can pass decision corresponding to a quantity of consecutive rows in the check matrix, and then a result of mapping the second code word corresponding to the last row of the check matrix is a decoding result. For example, the decoding device 200 performs processing according to a sequence from the first row, the second row, to the third row of the check matrix and needs to pass decision of three consecutive rows of the check matrix. If decision of a second code word that is output by the second row of the check matrix fails, a second code word that is output by the third row is decoded successfully, and second code words that are output by both the first row and the second row in a next round of update succeed in decoding, a result of mapping the second code word corresponding to the second row of the check matrix in the next round of update is a decoding result. Each round of update includes processing corresponding to each row of the check matrix, and before processing corresponding to a row of the check matrix in a round of update, an excitation for a code element corresponding to the row in a process of processing the row of the check matrix in a previous round of update needs to be removed.

**[0065]** In a case in which the processing unit 204 generates the second code word by using method 2 mentioned above, if the determining unit 206 determines that the second code word  $-5, +3, -8, -10, +3, +6, -7$  is decoded successfully, a result  $1, 0, 1, 1, 0, 0, 1$  of mapping the second code word is a decoding result. If the second code word fails to be decoded, a next round of update is performed.

**[0066]** This embodiment provides a decoding device. The decoding device performs processing in a case that occurs in a code word update process and in which a value of an updated code element exceeds a code element quantization range, which prevents the decoding device from directly quantizing an updated code word that exceeds the quantization range, and improves an error correction capability of the decoding device in an LDPC code decoding process.

## Device embodiment 2

[0067] Device embodiment 2 provides a decoding device 400, where the decoding device 400 is configured to decode an LDPC code. FIG. 4 shows a schematic diagram of a composite structure of the decoding device 400. The device 400 includes: an acquiring unit 402, a processing unit 404, and a determining unit 406.

[0068] The acquiring unit 402 is configured to acquire a first code word, where the first code word is generated by an LDPC code encoding device according to a generator matrix and includes M first code elements and M is a positive integer. For details, refer to the LDPC code encoding and decoding architecture mentioned above.

The code word acquired by the acquiring unit 402 may be transmitted through a communications channel, or may be read from a storage medium. The first code element herein is any code element in the first code word.

[0069] The processing unit 404 is configured to acquire P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquire P first update values corresponding to the P first code elements; obtain, according to the P first update values, P second update values corresponding to the P first code elements; sequentially determine whether each second update value of the P second update values exceeds a code element quantization range; keep the P first code elements not updated when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range; update the P first code elements by using the P second update values when it is determined that each second update value of the P second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generate a second code word according to the P second code elements, where the check matrix corresponds to the generator matrix and includes N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element.

[0070] The determining unit 406 is configured to acquire the second code word generated by the determining processing unit 404 and determine whether the second code word is decoded successfully.

$$\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{vmatrix}$$

**[0071]** In this embodiment, for example, the foregoing check matrix is used by the decoding device, that is, N is equal to 3, and the first code word acquired by the acquiring unit 402 is -9, +7, -12, +4, +7, +10, -11, that is, M is equal to 7. The

5 processing unit 404 generates, according to a code element that is in the first code word and related to the first row of the check matrix, an update value that is corresponding to the code element that is in the code word and related to the first row of the check matrix, that is, n is equal to 1. It should be noted that, during actual operation, the processing unit 404 may perform processing according to a sequence  
10 from the first row, the second row, to the third row of the check matrix, or may perform processing according to another sequence, where a value of n herein may be any positive integer less than or equal to a quantity of rows of the check matrix. After acquiring code elements related to the first row of the check matrix, namely, the 1<sup>st</sup> code element -9, the 2<sup>nd</sup> code element +7, and the 4<sup>th</sup> code element +4, where the first  
15 code elements include -9, +7, and +4, the processing unit 404 generates an update value +4 corresponding to the first code element in the first code word, an update value -4 corresponding to the second code element, and an update value -7 corresponding to the fourth code element, that is, P is equal to 3, and the first update values include +4, -4, and -7. A symbol of a first update value corresponding to each  
20 first code element of the P first code elements is a product of symbols of other first code elements except this first code element, and an absolute value of the first update value corresponding to each first code element of the P first code elements is an absolute value of a first code element with a smallest absolute value (if this first code element itself is the first code element with the smallest absolute value, the absolute  
25 value of the first update value corresponding to this first code element is set to an absolute value of a first code element with a second smallest absolute value). For example, if first code elements include a code element A, a code element B, and a code element C, a symbol of an update value corresponding to the code element A is a product of symbols of the code element B and the code element C, and an absolute  
30 value of the update value corresponding to the code element A is a smaller absolute value of absolute values of the code element B and the code element C.

**[0072]** After obtaining the update values corresponding to the code elements that

are in the first code word and related to the first row of the check matrix, the processing unit 404 adds each first update value to its corresponding first code element to obtain a second update value, that is, second update values include  $-5$  (a sum of  $+4$  and  $-9$ ),  $+3$  (a sum of  $+7$  and  $-4$ ), and  $-3$  (a sum of  $+4$  and  $-7$ ).

5    **[0073]**    After acquiring the second update values, the processing unit 404 sequentially determines whether each second update value exceeds the code element quantization range; if the processing unit 404 determines that the  $a^{\text{th}}$  second update value exceeds the code element quantization range, the processing unit 404 does not update the  $P$  first code elements by using the  $P$  second update values, that is, keeps the  
10     $P$  first code elements not updated; and if the processing unit 404 determines that each second update value of the  $P$  second update values does not exceed the code element quantization range, the processing unit 404 updates the  $P$  first code elements by using the  $P$  second update values.

**[0074]**    According to the foregoing example, because  $P$  is equal to 3 in this  
15    example, a value of  $a$  may be 1, 2, or 3 and corresponds to a second update value  $-5$ ,  $+3$ , or  $-3$ , respectively. For example,  $a$  is set to 1, and the processing unit 404 determines whether the second update value  $-5$  exceeds the code element quantization range; if the second update value  $-5$  exceeds the code element quantization range,  $-9$  is not updated by using  $-5$ ,  $-4$  is not updated by using  $+3$ , and  
20     $-7$  is not updated by using  $-3$ ; similarly, if another second update value exceeds the code element quantization range, updating the first code elements by using any one of the second update values is abandoned; if none of the second update values  $-5$ ,  $+3$ , and  $-3$  exceeds the code element quantization range,  $-9$  is updated by using  $-5$ ,  $-4$  is updated by using  $+3$ , and  $-7$  is updated by using  $-3$ . For example, the code element  
25    quantization range in this embodiment is from  $-15$  to  $+15$ . Then, none of the second update values exceeds the code element quantization range. The processing unit 404 generates second code elements after completing the processing corresponding to the first row of the check matrix. The second code elements include  $-5$ ,  $+3$ , and  $-3$ . The generation of each code element of the second code elements is to maintain a  
30    corresponding first code element (if any one of the second update values exceeds the code element quantization range) or to update a corresponding first code element by using a corresponding second update value (if none of the second update values exceeds the code element quantization range).

**[0075]**    The processing unit 404 generates the second code word according to the

second code elements. Mainly two methods for generating the second code word are shown as follows:

**[0076]** Method 1: A second code word is generated by updating, to a second code element, a code element (that is, the first code element) that is in the first code word  
5 -9, +7, -12, +4, +7, +10, -11 and related to the  $n^{\text{th}}$  ( $n$  is equal to 1 in this example) row of the check matrix, and a first code element that is in the first code word and unrelated to the  $n^{\text{th}}$  row of the check matrix is reserved, that is, the second code word is -5, +3, -12, -3, +7, +10, -11.

**[0077]** Method 2: A first temporary code word is generated by updating, to a  
10 second code element, a first code element that is in the first code word and related to the  $n^{\text{th}}$  ( $n$  is equal to 1 in this example) row of the check matrix, and a first code element that is in the first code word and unrelated to the first row of the check matrix is reserved, that is, the first temporary code word is -5, +3, -12, -3, +7, +10, -11.

**[0078]** A second temporary code word is generated by updating code elements  
15 that are in the first temporary code word and related to the second row of the check matrix. Code elements -12, +4, and +7 that are in the first code word and related to the second row of the check matrix are first acquired. Third update values including +4, -7, and -4 that are corresponding to the code elements that are in the first code word and related to the second row of the check matrix are generated. Fourth update  
20 values are generated by adding +4, -7, and -4 to the code elements -12, -3, and +7 that are in the first temporary code word and related to the second row of the check matrix, where the fourth update values include -8, -10, and +3. Whether any one of -8, -10, and +3 exceeds the code element quantization range is determined. If any one of the fourth update values exceeds the code element quantization range, the first  
25 temporary code word is not updated by using the fourth update values. If none of the fourth update values exceeds the code element quantization range, the first temporary code word -5, +3, -12, -3, +7, +10, -11 is updated by using the fourth update values -8, -10, and +3, so as to obtain a second temporary code word -5, +3, -8, -10, +3, +10, -11.

**[0079]** A second code word is generated by updating code elements that are in the  
30 second temporary code word and related to the third row of the check matrix. Code elements +4, +10, and -11 that are in the first code word and related to the third row of the check matrix are first acquired. Fifth update values including -10, -4, and +4 that are corresponding to the code elements that are in the first code word and related

to the third row of the check matrix are generated. Sixth update values including  $-20$ ,  $+6$ , and  $-7$  are generated by adding  $-10$ ,  $-4$ , and  $+4$  to the code elements  $-10$ ,  $+10$ , and  $-11$  that are in the second temporary code word and related to the third row of the check matrix. Whether any one of  $-20$ ,  $+6$ , and  $-7$  exceeds the code element quantization range is determined. If any one of the sixth update values exceeds the code element quantization range, the second temporary code word is not updated by using the sixth update values. If none of the sixth update values exceeds the code element quantization range, the second temporary code word  $-5$ ,  $+3$ ,  $-8$ ,  $-10$ ,  $+3$ ,  $+10$ ,  $-11$  is updated by using the sixth update values  $-20$ ,  $+6$ , and  $-7$ . A second code word  $-5$ ,  $+3$ ,  $-8$ ,  $-10$ ,  $+3$ ,  $+10$ ,  $-11$  is obtained in this example because the sixth update value  $-20$  exceeds the code element quantization range.

**[0080]** It should be noted that, in method 2, the second code word is obtained by processing the first code word according to a sequence from the first row, the second row, to the third row of the check matrix, but in an actual use process of method 2, the first code word may be processed according to another sequence of the check matrix. For example, the first row may be first processed to obtain the first temporary code word, then the third row is processed to obtain the second temporary code word, and then the second row is processed to obtain the second code word. There may be multiple processing sequences. It is only required that, in the process of generating the second code word by using the first code word, each row of the check matrix is processed correspondingly and the processing corresponding to each row of the check matrix is performed only once.

**[0081]** Optionally, the processing unit 404 includes a storage module, configured to store the  $P$  first code elements, and is further configured to store the  $P$  second update values. When determining that the  $a^{\text{th}}$  second update value of the  $P$  second update values exceeds the code element quantization range, the processing unit 404 reads the  $P$  first code elements stored in the storage module; that is, if a second update value exceeds the code element quantization range, the processing unit 404 does not update the first code elements by using the second update values, and therefore reads the  $P$  first code elements from the storage module, so as to acquire the second code elements. When determining that each second update value of the  $P$  second update values does not exceed the code element quantization range, the processing unit 404 reads the  $P$  second update values stored in the storage module; that is, if none of the second update values exceeds the code element quantization range, the processing

unit 404 updates the corresponding P first code elements by using the P second update values, and therefore reads the P second update values from the storage module and replaces the P first code elements, so as to acquire the second code elements.

**[0082]** According to the foregoing example, the storage module stores the first code elements  $-9$ ,  $+7$ , and  $+4$ , and the storage module further stores the second update values  $-5$ ,  $+3$ , and  $-3$ . If the processing unit 404 determines that a second update value exceeds the code element quantization range, the processing unit 404 reads the first code elements  $-9$ ,  $+7$ , and  $+4$  and abandons updating the first code elements by using any one of the second update values. If the processing unit 404 determines that none of the second update values exceeds the code element quantization range, the processing unit 404 reads the P second update values. For example, the processing unit 404 generates the second code word by using method 1. The processing unit 404 determines that none of the second update values  $-5$ ,  $+3$ , and  $-3$  exceeds the code element quantization range, the processing unit 404 reads the second update values  $-5$ ,  $+3$ , and  $-3$  to update the first code word and generate the second code word  $-5$ ,  $+3$ ,  $-12$ ,  $-3$ ,  $+7$ ,  $+10$ ,  $-11$ .

**[0083]** The processing unit 404 sends the second code word to the determining unit 406, where the determining unit 406 is configured to determine whether the second code word is decoded successfully.

**[0084]** In a case in which the processing unit 404 generates the second code word by using method 1 mentioned above, if the determining unit 406 determines that the second code word  $-5$ ,  $+3$ ,  $-12$ ,  $-3$ ,  $+7$ ,  $+10$ ,  $-11$  is decoded successfully, the decoding device 400 continues to perform an update corresponding to another row of the check matrix and generate a second code word corresponding to the row of the check matrix. Herein, the second row is used as an example. After an update corresponding to the second row of the check matrix is complete, an obtained second code word is  $-5$ ,  $+3$ ,  $-8$ ,  $-10$ ,  $+3$ ,  $+10$ ,  $-11$ . If the determining unit 406 determines that the second code word  $-5$ ,  $+3$ ,  $-8$ ,  $-10$ ,  $+3$ ,  $+10$ ,  $-11$  is decoded successfully, the decoding device 400 continues to perform an update corresponding to another row of the check matrix and generate a second code word corresponding to the row of the check matrix. Herein, the third row is used as an example. After an update corresponding to the third row of the check matrix is complete, because an update value corresponding to the code element  $-10$  exceeds the code element quantization range, the update corresponding to the third row of the check matrix is invalid and a second code word

-5, +3, -8, -10, +3, +10, -11 is obtained. If the determining unit 406 determines that the second code word -5, +3, -8, -10, +3, +10, -11 is decoded successfully, the decoding device 400 uses, as a decoding result, a result 1, 0, 1, 1, 0, 0, 1 of mapping -5, +3, -8, -10, +3, +10, -11. Generally, in a case in which the processing unit 404  
5 generates the second code word by using method 1 mentioned above, the decoding device 400 needs to determine that the second code word can pass decision corresponding to a quantity of consecutive rows in the check matrix, and then a result of mapping the second code word corresponding to the last row of the check matrix is a decoding result. For example, the decoding device 400 performs processing  
10 according to a sequence from the first row, the second row, to the third row of the check matrix and needs to pass decision of three consecutive rows of the check matrix. If decision of a second code word that is output by the second row of the check matrix fails, a second code word that is output by the third row is decoded successfully, and second code words that are output by both the first row and the  
15 second row in a next round of update succeed in decoding, a result of mapping the second code word corresponding to the second row of the check matrix in the next round of update is a decoding result. Each round of update includes processing corresponding to each row of the check matrix, and before processing corresponding to a row of the check matrix in a round of update, an excitation for a code element  
20 corresponding to the row in a process of processing the row of the check matrix in a previous round of update needs to be removed.

**[0085]** In a case in which the processing unit 404 generates the second code word by using method 2 mentioned above, if the determining unit 406 determines that the second code word -5, +3, -8, -10, +3, +10, -11 is decoded successfully, a result 1, 0,  
25 1, 1, 0, 0, 1 of mapping the second code word is a decoding result. If the second code word fails to be decoded, a next round of update is performed.

**[0086]** This embodiment provides a decoding device. The decoding device performs processing in a case that occurs in a code word update process and in which a value of an updated code element exceeds a code element quantization range, which  
30 prevents the decoding device from directly quantizing an updated code word that exceeds the quantization range, and improves an error correction capability of the decoding device in an LDPC code decoding process.

### Device embodiment 3

**[0087]** Device embodiment 3 provides a decoding device 600. FIG. 5 shows a schematic diagram of a composite structure of the decoding device 600. In this embodiment, an internal composite structure of the processing unit 204 in device embodiment 1 is detailed as a processing unit 604. An overall function of the processing unit 604 in this embodiment is the same as that of the processing unit 204 in device embodiment 1.

**[0088]** The following describes in detail an operation principle of the decoding device 600 proposed in device embodiment 3 by using an example in which a code word acquired by the decoding device 600 is -9, -7, -12, -4, +7, +10, -11, a code element quantization range is from -15 to +15, and the following check matrix is used by the decoding device 600.

$$\left| \begin{array}{ccccccc|ccccccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array} \right|$$

**[0089]** After acquiring the code word -9, +7, -12, +4, +7, +10, -11, an acquiring unit 602 first sends the code word to a determining unit 606, and the determining unit 606 maps negative values in the code word to 1 and maps positive values to 0 to obtain a mapped code word 1, 0, 1, 0, 0, 0, 1, and performs an exclusive OR operation, also referred to as a sum modulo 2 or an addition modulo 2 operation, on each code element in the mapped code word according to a check equation. According to an operation on the first row of the check matrix,  $(1+0+0) \bmod (2)=1$ , and the code word fails to be decoded. If the code word acquired by the decoding device 600 passes decision of the determining unit 606 before the code word is processed by the processing unit 604, the code word does not need subsequent processing, and the decoding device 600 directly uses a mapping result of the code word as a decoding result. The code word 1, 0, 1, 0, 0, 0, 1 needs to pass check corresponding to all three rows of the check matrix before the determining unit 606 determines that the code word is decoded successfully.

**[0090]** A variable node processing unit 6042 receives the code word -9, +7, -12, +4, +7, +10, -11 and stores -9, +7, -12, +4, +7, +10, -11 to a storage module 6043. It can be learned from the first row of the check matrix that the first, the second, and the fourth variable nodes are related to the first row of the check matrix, that is, the

three code elements  $-9$ ,  $+7$ , and  $+4$  are related to the first row of the check matrix. The variable node processing unit 6042 sends  $-9$ ,  $+7$ , and  $+4$  to a check node processing unit 6044. In practice, processing may start from any row of the check matrix. It should be noted that, if an update of a corresponding row of the check matrix or updates corresponding to all rows of the check matrix are already performed, the variable node processing unit 6042 further needs to remove impact of a previous round of update values on the code word before performing this round of update.

**[0091]** After receiving  $-9$ ,  $+7$ , and  $+4$ , the check node processing unit 6044 generates, according to the first row of the check matrix, an update value  $+4$  for returning to the first variable node, an update value  $-4$  for returning to the second variable node, and an update value  $-7$  for returning to the fourth variable node, and sends the three update values to the variable node processing unit 6042.

**[0092]** The variable node processing unit 6042 may subsequently perform either of the following two solutions:

**[0093]** Solution 1: After receiving update values, the variable node processing unit 6042 adds code elements of the first, the second, and the fourth variable nodes to their corresponding update values, and therefore, the code element corresponding to the first variable node is updated to  $-5$ , the code element corresponding to the second variable node is updated to  $+3$ , and the code element corresponding to the fourth variable node is updated to  $-3$ ; stores the updated code elements of the first, the second, and the fourth variable nodes in the storage module 6043; if an update value of a code element corresponding to any variable node in an update process exceeds the code element quantization range, that is, from  $-15$  to  $+15$ , abandons the update of the code element corresponding to the variable node, reads, from the storage module 6043, the code element before the update, continues to update a code element corresponding to a next variable node, and repeats the foregoing determining actions for the update of the code element corresponding to the next variable node, until processing of all code elements corresponding to the first variable node, the second variable node, and the fourth variable node is complete.

**[0094]** Solution 2: After receiving update values, the variable node processing unit 6042 stores all the update values; adds code elements of the first, the second, and the fourth variable nodes to their corresponding update values; if an update value of a code element corresponding to any variable node in an update process exceeds the

code element quantization range, that is, from  $-15$  to  $+15$ , abandons the update of the code element corresponding to the variable node, reads, from the storage module 6043, the code element before the update, and continues to update a code element corresponding to a next variable node; after updates of the first row of the check matrix are complete, continues to update the second row of the check matrix, that is, perform update processing corresponding to a check node 2 in a Tanner diagram shown in FIG. 6. In FIG. 6, a variable node 1 to a variable node 7 correspond to seven code elements received by the variable node processing unit 6042 respectively, and a check node 1 to a check node 3 correspond to the first row to the third row of the check matrix respectively. In FIG. 6, a processing action performed by the variable node 1 to the variable node 7 is also performed by the variable node processing unit 6042 in FIG. 5, and a processing action performed by the check node 1 to the check node 3 is also performed by the check node processing unit 6044 in FIG. 5.

**[0095]** After the check node 1 returns update values to the variable node 1, the variable node 2, and the variable node 4, if none of updated code elements corresponding to the variable node 1, the variable node 2, and the variable node 4 exceeds the code element quantization range, the variable node 3, the variable node 4, and the variable node 5 continues to send corresponding code elements to the check node 2, and the check node 2 generates corresponding update values and then returns the update values to corresponding variable nodes, and if the code element quantization range is still not exceeded, an update corresponding to the check node 3 continues. In this case, because the code element corresponding to the variable node 4 participates in the processing corresponding to the check node 1 and also participates in the processing corresponding to the check node 2 and the processing corresponding to the check node 3, in FIG. 6, if a sum of the code element  $+4$  corresponding to the variable node 4 and an update value  $-7$  returned by the check node 1 does not exceed the code element quantization range, a sum of a code element corresponding to a variable node and an update value returned by another check node may exceed the code element quantization range. In an example shown in FIG. 6, the updates returned by the check node 1, the check node 2, and the check node 3 to the variable node 4 are  $-7$ ,  $-7$ , and  $-10$  respectively. After the three update values are added to the code word  $+4$ , obtained  $-20$  exceeds the code element quantization range. Because all update values are already stored, the three update values are sorted to obtain an update value that has the greatest impact on the exceeding of the code element quantization range.

Because the code word in this example exceeds a minimum value of the code element quantization range, the update value with the greatest impact is an update value with a smallest value, that is,  $-10$ , the variable node 4 abandons the update value  $-10$  returned by the check node 3, that is, after this round of update, a code word

5 corresponding to the variable node 4 is  $+4 + (-7) + (-7)$ , and equals to  $-10$ . In solution 2, after a sum of a code element of any variable node and update values returned by multiple check nodes exceeds the code element quantization range, the update values received by the variable node are sorted; when the sum of the code element of the variable node and the update values exceeds a maximum value of the

10 code element quantization range, the variable node first abandons a largest value of the update values; and when the sum of the code element of the variable node and the update values exceeds a minimum value of the code element quantization range, the variable node first abandons a smallest value of the update values, which ensures that the abandoned update value has least impact on a decoding process.

15 **[0096]** Each time the variable node processing unit 6042 completes updating code elements corresponding to one row of the check matrix, that is, processing an update value sent by one check node, the variable node processing unit 6042 sends an updated code word to the determining unit 606, and the determining unit 606 maps the updated code word and determines whether this row of code word passes decision.

20 In the previous example, after updates of code elements corresponding to variable nodes corresponding to the first row of the check matrix are complete, an updated code word is  $-5, +3, -12, -3, +7, +10, -11$ , and the determining unit 606 performs decision on the updated code word. For details, refer to how the processing unit 204 generates the second code word by using method 1 in device embodiment 1. After

25 updating the code elements corresponding to one row of the check matrix and acquiring the updated code word, the variable node processing unit 6042 may further continue to generate the second code word by using method 2 used by the processing unit 204 in device embodiment 1, and send the second code word to the determining unit 606. It should be noted that, before performing mapping and decision on the code

30 word, the determining unit 606 may further perform minimum offset algorithm processing on the code word, or may further perform normalized min-sum algorithm

processing on the code word.

[0097] This embodiment provides a decoding device. The decoding device performs processing in a case that occurs in a code word update process and in which a value of an updated code element exceeds a code element quantization range, which prevents the decoding device from directly quantizing an updated code word that exceeds the code element quantization range, and improves an error correction capability of the decoding device in a decoding process.

#### Device embodiment 4

[0098] Device embodiment 4 provides a decoding device 800. FIG. 7 shows a schematic diagram of a composite structure of the decoding device 800. In this embodiment, an internal composite structure of the processing unit 404 in device embodiment 2 is detailed as a processing unit 804. An overall function of the processing unit 804 in this embodiment is the same as that of the processing unit 404 in device embodiment 2.

[0099] The following describes in detail an operation principle of the decoding device 800 proposed in device embodiment 4 by using an example in which a code word acquired by the decoding device 800 is -9, -7, -12, -4, +7, +10, -11, a code element quantization range is from -15 to +15, and the following check matrix is used by the decoding device 800.

$$\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{vmatrix}$$

[0100] After acquiring the code word -9, +7, -12, +4, +7, +10, -11, an acquiring unit 802 first sends the code word to a determining unit 806, and the determining unit 806 maps negative values in the code word to 1 and maps positive values to 0 to obtain a mapped code word 1, 0, 1, 0, 0, 0, 1, and performs an exclusive OR operation, also referred to as a sum modulo 2 or an addition modulo 2 operation, on each code element in the mapped code word according to a check equation. According to an operation on the first row of the check matrix,  $(1+0+0) \bmod (2)=1$ , and the code word fails to be decoded. If the code word acquired by the decoding device 800 passes decision of the determining unit 806 before the code word is processed by the processing unit 804, the code word does not need subsequent processing, and the decoding device 800 directly uses a mapping result of the code word as a decoding

result. The code word 1, 0, 1, 0, 0, 0, 1 needs to pass check corresponding to all three rows of the check matrix before the determining unit 806 determines that the code word is decoded successfully.

**[0101]** A variable node processing unit 8042 receives the code word  $-9, +7, -12,$

5  $+4, +7, +10, -11$  and stores  $-9, +7, -12, +4, +7, +10, -11$  to a storage module 8043.

It can be learned from the first row of the check matrix that the first, the second, and the fourth variable nodes are related to the first row of the check matrix, that is, the three code elements  $-9, +7,$  and  $+4$  are related to the first row of the check matrix.

The variable node processing unit 8042 sends  $-9, +7,$  and  $+4$  to a check node

10 processing unit 8044. In practice, processing may start from any row of the check matrix. It should be noted that, if an update of a corresponding row of the check

matrix or updates corresponding to all rows of the check matrix are already performed, the variable node processing unit 8042 further needs to remove impact of a previous

round of update values on the code word before performing this round of update.

15 **[0102]** After receiving  $-9, +7,$  and  $+4,$  the check node processing unit 8044

generates, according to the first row of the check matrix, an update value  $+4$  for returning to the first variable node, an update value  $-4$  for returning to the second variable node, and an update value  $-7$  for returning to the fourth variable node, and sends the three update values to the variable node processing unit 8042.

20 **[0103]** The variable node processing unit 8042 subsequently performs the

following solution: after receiving update values, adding code elements of the first, the second, and the fourth variable nodes to their corresponding update values, where the code element corresponding to the first variable node is updated to  $-5,$  the code element corresponding to the second variable node is updated to  $+3,$  and the code

25 element corresponding to the fourth variable node is updated to  $-3;$  storing the

updated code elements of the first, the second, and the fourth variable nodes in the storage module 8043; if an update value of a code element corresponding to any variable node exceeds the code element quantization range in an update process,

abandoning this round of update of the code elements corresponding to the first, the

30 second, and the fourth variable nodes, and reading, from the storage module 8043, the

code elements corresponding to the first, the second, and the fourth variable nodes

before the update.

**[0104]** Each time the variable node processing unit 8042 completes updating code elements corresponding to one row of the check matrix, that is, processing an update

value sent by one check node, the variable node processing unit 8042 sends an updated code word to the determining unit 806, and the determining unit 806 maps the updated code word and determines whether this row of code word passes decision. In the previous example, after updates of code elements corresponding to variable nodes corresponding to the first row of the check matrix are complete, an updated code word is  $-5, +3, -12, -3, +7, +10, -11$ , and the determining unit 806 performs decision on the updated code word. For details, refer to how the processing unit 404 generates the second code word by using method 1 in device embodiment 2. After updating the code elements corresponding to one row of the check matrix and acquiring the updated code word, the variable node processing unit 8042 may further continue to generate the second code word by using method 2 used by the processing unit 404 in device embodiment 2, and send the second code word to the determining unit 806. It should be noted that, before performing mapping and decision on the code word, the determining unit 806 may further perform minimum offset algorithm processing on the code word, or may further perform normalized min-sum algorithm processing on the code word.

**[0105]** This embodiment provides a decoding device. The decoding device performs processing in a case that occurs in a code word update process and in which a value of an updated code element exceeds a code element quantization range, which prevents the decoding device from directly quantizing an updated code word that exceeds the code element quantization range, and improves an error correction capability of the decoding device in a decoding process.

### **Method embodiment 1**

**[0106]** This method embodiment provides a decoding method, where the decoding method is used to decode an LDPC code. This method may be performed by the decoding device 200 according to device embodiment 1 or be performed by the decoding device 600 according to device embodiment 3. FIG. 8 shows a schematic flowchart of the decoding method. The method includes:

**[0107]** Step 1002: Acquire a first code word, where the first code word is generated by an LDPC code encoding device according to a generator matrix and includes M first code elements and M is a positive integer. For details about execution of step 1002, refer to the acquiring unit 202 in device embodiment 1 or the acquiring

unit 602 in device embodiment 3.

**[0108]** Step 1004: Acquire P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquire P first update values corresponding to the P first code elements; obtain, according to the P first update values, P second update values corresponding to the P first code elements; sequentially determine whether each second update value of the P second update values exceeds a code element quantization range; keep the  $a^{\text{th}}$  first code element not updated when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range; update the  $b^{\text{th}}$  first code element by using the  $b^{\text{th}}$  second update value when it is determined that the  $b^{\text{th}}$  second update value of the P second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generate a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, where the check matrix corresponds to the generator matrix and includes N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element. For details about execution of step 1004, refer to the processing unit 204 in device embodiment 1, or the processing unit 604 and all subunits of the processing unit 604 in device embodiment 3.

**[0109]** Step 1006: Determine whether the second code word generated in step 1004 is decoded successfully. For details about execution of step 1006, refer to the determining unit 206 in device embodiment 1 or the determining unit 606 in device embodiment 3.

**[0110]** According to the decoding method provided in this embodiment, processing is performed in a case that occurs in a code word update process and in which a value of an updated code element exceeds a code element quantization range, which prevents direct quantization of an updated code word that exceeds the code element quantization range in a decoding process, and improves an LDPC code error correction capability.

## Method embodiment 2

**[0111]** This method embodiment provides a decoding method, where the decoding method is used to decode an LDPC code. This method may be performed by

the decoding device 400 according to device embodiment 2 or be performed by the decoding device 800 according to device embodiment 4. FIG. 9 shows a schematic flowchart of the decoding method. The method includes:

**[0112]** Step 1202: Acquire a first code word, where the first code word is generated by an LDPC code encoding device according to a generator matrix and includes M first code elements and M is a positive integer. For details about execution of step 1202, refer to the acquiring unit 402 in device embodiment 2 or the acquiring unit 802 in device embodiment 4.

**[0113]** Step 1204: Acquire P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquire P first update values corresponding to the P first code elements; obtain, according to the P first update values, P second update values corresponding to the P first code elements; sequentially determine whether each second update value of the P second update values exceeds a code element quantization range; keep the P first code elements not updated when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range; update the P first code elements by using the P second update values when it is determined that each second update value of the P second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generate a second code word according to the P second code elements, where the check matrix corresponds to the generator matrix and includes N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element. For details about execution of step 1204, refer to the processing unit 404 in device embodiment 2, or the processing unit 804 and all subunits of the processing unit 804 in device embodiment 4.

**[0114]** Step 1206: Determine whether the second code word generated in step 1204 is decoded successfully. For details about execution of step 1206, refer to the determining unit 406 in device embodiment 2 or the determining unit 806 in device embodiment 4.

**[0115]** According to the decoding method provided in this embodiment, processing is performed in a case that occurs in a code word update process and in which a value of an updated code element exceeds a code element quantization range, which prevents direct quantization of an updated code word that exceeds the code element quantization range in a decoding process, and improves an LDPC code error

correction capability.

### Method embodiment 3

**[0116]** This method embodiment provides a decoding method. Because an LDPC code is being decoded, this method may be performed by any one of the decoding devices according to device embodiment 1, or device embodiment 2, or device embodiment 3, or device embodiment 4. This embodiment mainly describes how a final decoding result is acquired in a process in which a processing unit 204, a processing unit 404, a processing unit 604, and a processing unit 804 acquire a second code word by using method 1 and perform decision on the second code word. FIG. 10 shows a schematic flowchart of the decoding method. The method includes:

**[0117]** Step 1402: Acquire a code word processed by an LDPC code encoding device. In this method embodiment, for example, the acquired code word is -9, +7, -12, +4, +7, +10, -11, and the following check matrix is used. In practice, if one round or multiple rounds of updates are already performed on the code word before execution of step 1402, in step 1402, impact brought by a previous round of update on the code word further needs to be removed, that is, a previous round of update values are subtracted.

$$\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{vmatrix}$$

**[0118]** Step 1404: Determine whether the code word -9, +7, -12, +4, +7, +10, -11 is decoded successfully. Negative values in code elements are mapped to 1 and positive values are mapped to 0 to obtain a mapped code word 1, 0, 1, 0, 0, 0, 1, and an exclusive OR operation, also referred to as a sum modulo 2 or an addition modulo 2 operation, is performed on each code element in the mapped code word according to a check equation. According to an operation on the  $K^{\text{th}}$  row of the check matrix,  $(1+0+0) \bmod (2)=1$ , and the code word fails to be decoded. If the decoding succeeds in step 1404, a decoding result is output.

**[0119]** Step 1406: Generate, according to code elements corresponding to the  $K^{\text{th}}$  row of the check matrix, update values corresponding to the code elements corresponding to the  $K^{\text{th}}$  row. That is, for the first code element -9, the second code element +7, and the fourth code element +4, update values corresponding to the three code elements are generated, namely, an update value +4 of the first code element, an

update value  $-4$  of the second code element, and an update value  $-7$  of the fourth code element. Herein,  $K$  may start from 1.

**[0120]** After step 1406, either step 1408 or step 1410 may be performed.

**[0121]** Step 1408: Update, according to the update values of the code elements corresponding to the  $K^{\text{th}}$  row of the check matrix, the code elements corresponding to the  $K^{\text{th}}$  row of the check matrix; and, if an update value of any code element exceeds a code element quantization range, abandon an update of the code element. In this method embodiment, a code element quantization range is set to from  $-15$  to  $+15$ , and the code word  $-9, +7, -12, +4, +7, +10, -11$  is updated to a code word  $-5, +3, -12, -3, +7, +10, -11$ .

**[0122]** Step 1410: Update, according to the update values of the code elements corresponding to the  $K^{\text{th}}$  row of the check matrix, the code elements corresponding to the  $K^{\text{th}}$  row of the check matrix; and, if an update value of any code element exceeds a code element quantization range, abandon an update of all code elements corresponding to the  $K^{\text{th}}$  row of the check matrix. Likewise, because no updated code element exceeds the code element quantization range, the code word  $-9, +7, -12, +4, +7, +10, -11$  is updated to a code word  $-5, +3, -12, -3, +7, +10, -11$ .

**[0123]** Step 1412: Determine whether the code word updated in step 1408 or step 1410 is decoded successfully. If the decoding succeeds, a counter increases by 1 (each time the decoding device acquires a new to-be-decoded code word, an initial value of the counter is 0; or each time the decoding device finds that the code word obtained in step 1412 fails to be decoded, a value of the counter returns to 0); if the decoding fails, step 1402 to step 1412 are performed again on another row of the check matrix by using the updated code word as the code word acquired in step 1402.

**[0124]** Step 1414: Determine whether the counter reaches a preset threshold; if the preset threshold is not reached, perform step 1402 to step 1412 again on another row of the check matrix by using the updated code word as the code word acquired in step 1402; if the preset threshold is reached, use, as a decoding result of the code word, a result of mapping a finally acquired updated code word and output the decoding result.

**[0125]** It should be noted that step 1404 is not required in a subsequent update process, except in the first update process of a code word. After step 1402 is performed, step 1406 is directly performed.

**[0126]** According to the decoding method provided in this embodiment,

processing is performed in a case that occurs in a code word update process of an LDPC code decoding process and in which a value of an updated code element exceeds a code element quantization range, which prevents direct quantization of an updated code word that exceeds the code element quantization range in the LDPC code decoding process, and improves an LDPC code error correction capability.

### **System embodiment 1**

[0127] This system embodiment provides a signal transmission system, including a decoding device, an LDPC encoding device, and a communications channel. The decoding device may be the decoding device 200 according to device embodiment 1 or the decoding device 600 according to device embodiment 3.

[0128] The LDPC code encoding device is configured to generate a first code word according to a generator matrix and transmit the first code word to the decoding device through the communications channel. For a subsequent process in which the decoding device processes the first code word, refer to implementation details of the decoding device 200 according to device embodiment 1 or implementation details of the decoding device 600 according to device embodiment 3.

[0129] This embodiment provides a signal transmission system. A decoding device in the signal transmission system performs processing in a case that occurs in a code word update process of an LDPC code decoding process and in which a value of an updated code element exceeds a code element quantization range, which prevents direct quantization of an updated code word that exceeds the code element quantization range in the LDPC code decoding process, improves a capability of correcting an error in an LDPC code transmitted in the signal transmission system, and also improves reliability of a signal transmitted in the signal transmission system.

### **System embodiment 2**

[0130] This system embodiment provides a signal transmission system, including a decoding device, an LDPC encoding device, and a communications channel. The decoding device may be the decoding device 400 according to device embodiment 2 or the decoding device 800 according to device embodiment 4.

[0131] The LDPC code encoding device is configured to generate a first code

word according to a generator matrix and transmit the first code word to the decoding device through the communications channel. For a subsequent process in which the decoding device processes the first code word, refer to implementation details of the decoding device 400 according to device embodiment 2 or implementation details of the decoding device 800 according to device embodiment 4.

**[0132]** This embodiment provides a signal transmission system. A decoding device in the signal transmission system performs processing in a case that occurs in a code word update process of an LDPC code decoding process and in which a value of an updated code element exceeds a code element quantization range, which prevents direct quantization of an updated code word that exceeds the code element quantization range in the LDPC code decoding process, improves a capability of correcting an error in an LDPC code transmitted in the signal transmission system, and also improves reliability of a signal transmitted in the signal transmission system.

### **System embodiment 3**

**[0133]** This system embodiment provides a data storage system, including a decoding device, an LDPC encoding device, and a storage medium. The decoding device may be the decoding device 200 according to device embodiment 1 or the decoding device 600 according to device embodiment 3.

**[0134]** The LDPC code encoding device is configured to generate a first code word according to a generator matrix and store the first code word in the storage medium, and the decoding device is configured to acquire the first code word from the storage medium. For a subsequent process in which the decoding device processes the first code word, refer to implementation details of the decoding device 200 according to device embodiment 1 or implementation details of the decoding device 600 according to device embodiment 3.

**[0135]** This embodiment provides a data storage system. A decoding device in the data storage system performs processing in a case that occurs in a code word update process of an LDPC code decoding process and in which a value of an updated code element exceeds a code element quantization range, which prevents direct quantization of an updated code word that exceeds the code element quantization range in the LDPC code decoding process, improves an LDPC code error correction

capability of the decoding device in the data storage system, and also improves reliability of data stored in the data storage system.

#### **System embodiment 4**

5     **[0136]**     This system embodiment provides a data storage system, including a decoding device, an LDPC encoding device, and a storage medium. The decoding device may be the decoding device 400 according to device embodiment 2 or the decoding device 800 according to device embodiment 4.

10    **[0137]**     The LDPC code encoding device is configured to generate a first code word according to a generator matrix and store the first code word in the storage medium, and the decoding device is configured to acquire the first code word from the storage medium. For a subsequent process in which the decoding device processes the first code word, refer to implementation details of the decoding device 400 according to device embodiment 2 or implementation details of the decoding device 800 according to device embodiment 4.

15    **[0138]**     This embodiment provides a data storage system. A decoding device in the data storage system performs processing in a case that occurs in a code word update process of an LDPC code decoding process and in which a value of an updated code element exceeds a code element quantization range, which prevents direct quantization of an updated code word that exceeds the code element quantization range in the LDPC code decoding process, improves an LDPC code error correction capability of the decoding device in the data storage system, and also improves reliability of data stored in the data storage system.

25    **[0139]**     FIG. 11 is a comparison diagram of emulation results of a conventional decoding device and a decoding device proposed in the present application. Emulation is generating, based on an LDPC FPGA emulation platform, a random sequence (that is, the to-be-transmitted signal mentioned above) by using an FPGA, then performing LDPC encoding and modulation and adding random white Gaussian noise to generate a code word (that is, the first code word mentioned above), and finally performing LDPC decoding on the code word. The FPGA emulation platform completes statistics collection on RAWBER and UBER. A full name of UBER is Uncorrected Bit Error Rate, that is, a bit error rate that cannot be corrected. A full name of RAWBER is Raw

Bit Error Rate, that is, a bit error rate of the code word generated by performing encoding and adding white noise. Therefore, a smaller value of the RAWBER indicates a lower bit error rate of the to-be-decoded code word itself in theory, and therefore, the UBER rate should also decrease accordingly. It can be seen from FIG. 11 that, in the conventional solution, when the RAWBER is less than 0.0021, the UBER no longer decreases accordingly and an LDPC code error correction capability reaches a bottleneck. However, when the LDPC code decoding device provided in the present application is used and the RAWER is less than 0.0021, the UBER still keeps decreasing accordingly, which breaks the performance bottleneck of the conventional LDPC code decoding device and effectively improves the LDPC code error correction capability.

**[0140]** In device embodiment 1, the processing unit 204 is configured to acquire P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquire P first update values corresponding to the P first code elements; obtain, according to the P first update values, P second update values corresponding to the P first code elements; determine whether each second update value of the P second update values exceeds a code element quantization range; keep the  $a^{\text{th}}$  first code element not updated when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range; update the  $b^{\text{th}}$  first code element by using the  $b^{\text{th}}$  second update value when it is determined that the  $b^{\text{th}}$  second update value of the P second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generate a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, where the check matrix corresponds to the generator matrix and includes N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element.

**[0141]** Optionally, the determining whether each second update value of the P second update values exceeds a code element quantization range may be concurrently determining whether each second update value of the P second update values exceeds the code element quantization range, that is, after acquiring each second update value of the P second update values, the processing unit 204 concurrently determines whether the P second update values exceed the code element quantization range; or

may be sequentially determining, from left to right according to a sequence of the P code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, whether each second update value of the P second update values exceeds the code element quantization range.

5   **[0142]**    Optionally, the acquiring unit 202, the processing unit 204, and the determining unit 206 that are included in the decoding device 200 each are implemented by using a circuit.

10   **[0143]**    After acquiring the second update values, the processing unit 204 determines whether each second update value exceeds the code element quantization range; if the processing unit 204 determines that the  $a^{\text{th}}$  second update value exceeds the code element quantization range, the processing unit 204 does not update the  $a^{\text{th}}$  first code element by using the  $a^{\text{th}}$  second update value, that is, keeps the  $a^{\text{th}}$  first code element not updated; if the processing unit 204 determines that the  $b^{\text{th}}$  second update value does not exceed the code element quantization range, the processing unit 204  
15   updates the  $b^{\text{th}}$  first code element by using the  $b^{\text{th}}$  second update value.

20   **[0144]**    According to the example in device embodiment 1, because P is equal to 3 in this example, a value of a may be 1, 2, or 3 and corresponds to a second update value -5, +3, or -3, respectively. For example, a is set to 1; the processing unit 204 determines whether the second update value -5 exceeds the code element  
25   quantization range; if the second update value -5 exceeds the code element quantization range, -9 is not updated by using -5, that is, the code element -9 keeps unchanged; if the second update value -5 does not exceed the code element quantization range, -9 is updated by using -5, that is, the code element -9 is updated to -5. The processor 204 may concurrently determine whether the second update  
30   values -5, +3, and -3 exceed the code element quantization range, or may determine whether the second update values -5, +3, and -3 exceed the code element quantization range according to a sequence of first code elements corresponding to a bit 1 on the first row of the check matrix, that is, first -5, then +3, and then -3.

30   **[0145]**    In device embodiment 2, the processing unit 404 is configured to acquire P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquire P first update values corresponding to the P first code elements; obtain, according to the P

first update values, P second update values corresponding to the P first code elements; determine whether each second update value of the P second update values exceeds a code element quantization range; keep the P first code elements not updated when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the  
5 code element quantization range; update the P first code elements by using the P second update values when it is determined that each second update value of the P second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generate a second code word according to the P  
10 second code elements, where the check matrix corresponds to the generator matrix and includes N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element.

**[0146]** Optionally, the determining whether each second update value of the P second update values exceeds a code element quantization range may be concurrently  
15 determining whether each second update value of the P second update values exceeds the code element quantization range, that is, after acquiring each second update value of the P second update values, the processing unit 404 concurrently determines whether the P second update values exceed the code element quantization range; or may be sequentially determining, from left to right according to a sequence of the P  
20 code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, whether each second update value of the P second update values exceeds the code element quantization range.

**[0147]** Optionally, the acquiring unit 402, the processing unit 404, and the determining unit 406 that are included in the decoding device 400 each are  
25 implemented by using a circuit.

**[0148]** After acquiring the second update values, the processing unit 404 determines whether each second update value exceeds a code element quantization range; if the processing unit 404 determines that the  $a^{\text{th}}$  second update value exceeds the code element quantization range, the processing unit 404 does not update the P  
30 first code elements by using the P second update values, that is, keeps the P first code elements not updated; if the processing unit 404 determines that each second update value of the P second update values does not exceed the code element quantization

range, the processing unit 404 updates the P first code elements by using the P second update values.

**[0149]** According to the example in device embodiment 1, because P is equal to 3 in this example, a value of a may be 1, 2, or 3 and corresponds to a second update value -5, +3, or -3, respectively. For example, a is set to 1; the processing unit 404 determines whether the second update value -5 exceeds the code element quantization range; if the second update value -5 exceeds the code element quantization range, -9 is not updated by using -5, -4 is not updated by using +3, and -7 is not updated by using -3; similarly, if another second update value exceeds the code element quantization range, updating the first code elements by using any one of the second update values is abandoned; if none of the second update values -5, +3, and -3 exceeds the code element quantization range, -9 is updated by using -5, -4 is updated by using +3, and -7 is updated by using -3. The processor 404 may concurrently determine whether the second update values -5, +3, and -3 exceed the code element quantization range, or may determine second update values -5, +3, and -3 according to a sequence of first code elements corresponding to a bit 1 on the first row of the check matrix, that is, first -5, then +3, and then -3, and if any one of the first code elements exceeds the code element quantization range, determining is not required for remaining first code elements.

**[0150]** In device embodiment 3, the variable node processing unit 6042 is implemented by using a circuit, and the check node processing unit 6044 is implemented by using a circuit.

**[0151]** In device embodiment 4, the variable node processing unit 8042 is implemented by using a circuit, and the check node processing unit 8044 is implemented by using a circuit.

**[0152]** In method embodiment 1, step 1004: acquire P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquire P first update values corresponding to the P first code elements; obtain, according to the P first update values, P second update values corresponding to the P first code elements; determine whether each second update value of the P second update values exceeds a code

element quantization range; keep the  $a^{\text{th}}$  first code element not updated when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range; update the  $b^{\text{th}}$  first code element by using the  $b^{\text{th}}$  second update value when it is determined that the  $b^{\text{th}}$  second update value of the P second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generate a second code word according to the P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, where the check matrix corresponds to the generator matrix and includes N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element. For details about execution of step 1004, refer to the processing unit 204 in device embodiment 1, or the processing unit 604 and all subunits of the processing unit 604 in device embodiment 3.

**[0153]** Optionally, the determining whether each second update value of the P second update values exceeds a code element quantization range may be concurrently determining whether each second update value of the P second update values exceeds the code element quantization range, that is, after acquiring each second update value of the P second update values, concurrently determining whether the P second update values exceed the code element quantization range; or may be sequentially determining, from left to right according to a sequence of the P code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, whether each second update value of the P second update values exceeds the code element quantization range.

**[0154]** In method embodiment 2, step 1204: acquire P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquire P first update values corresponding to the P first code elements; obtain, according to the P first update values, P second update values corresponding to the P first code elements; determine whether each second update value of the P second update values exceeds a code element quantization range; keep the P first code elements not updated when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range; update the P first code elements by using the P

second update values when it is determined that each second update value of the P second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generate a second code word according to the P second code elements, where the check matrix corresponds to the generator matrix and includes N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element. For details about execution of step 1204, refer to the processing unit 404 in device embodiment 2, or the processing unit 804 and all subunits of the processing unit 804 in device embodiment 4.

**[0155]** Optionally, the determining whether each second update value of the P second update values exceeds a code element quantization range may be concurrently determining whether each second update value of the P second update values exceeds the code element quantization range, that is, after acquiring each second update value of the P second update values, concurrently determining whether the P second update values exceed the code element quantization range; or may be sequentially determining, from left to right according to a sequence of the P code elements corresponding to the  $n^{\text{th}}$  row of the check matrix, whether each second update value of the P second update values exceeds the code element quantization range.

**[0156]** For parts without detailed description in an embodiment of the present application, refer to related description of another embodiment. Finally, it should be noted that the foregoing embodiments are merely intended for describing the technical solutions of the present invention, but not for limiting the present invention. Although the present invention is described in detail with reference to the foregoing embodiments, persons of ordinary skill in the art should understand that they may still make modifications to the technical solutions described in the foregoing embodiments or make equivalent replacements to some technical features thereof, without departing from the scope of the technical solutions of the embodiments of the present invention.

## CLAIMS

What is claimed is:

1. A decoding device, configured to decode a low-density parity-check LDPC code and comprising:

5 an acquiring unit, configured to acquire a first code word, wherein the first code word is generated by an LDPC code encoding device according to a generator matrix and comprises  $M$  first code elements and  $M$  is a positive integer;

a processing unit, configured to acquire  $P$  first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquire  $P$  first update values corresponding to the  $P$  first code elements; obtain, according to the  $P$  first update values,  $P$  second update values corresponding to the  $P$  first code elements; determine whether each second update value of the  $P$  second update values exceeds a code element quantization range; keep the  $a^{\text{th}}$  first code element not updated when it is determined that the  $a^{\text{th}}$  second update value of the  $P$  second update values exceeds the code element quantization range; 10 update the  $b^{\text{th}}$  first code element by using the  $b^{\text{th}}$  second update value when it is determined that the  $b^{\text{th}}$  second update value of the  $P$  second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation,  $P$  second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generate a second code word according to the  $P$  second code elements 20 corresponding to the  $n^{\text{th}}$  row of the check matrix, wherein the check matrix corresponds to the generator matrix and comprises  $N$  rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element; and

a determining unit, configured to acquire the second code word and determine 25 whether the second code word is decoded successfully.

2. The decoding device according to claim 1, wherein the determining whether each second update value of the  $P$  second update values exceeds a code element quantization range comprises:

sequentially determining whether each second update value of the  $P$  second update values exceeds the code element quantization range. 30

3. The decoding device according to claim 1 or 2, wherein the acquiring unit, the processing unit, and the determining unit each are formed by a circuit.

4. The decoding device according to any one of claims 1 to 3, wherein the processing unit comprises a storage module;

the storage module is configured to store the P first code elements, and is further configured to store the P second update values;

5 the processing unit reads, when determining that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range, the  $a^{\text{th}}$  first code element stored in the storage module; and

the processing unit reads, when determining that the  $b^{\text{th}}$  second update value of the P second update values does not exceed the code element quantization range, the  $b^{\text{th}}$  second update value stored in the storage module, so as to update the  $b^{\text{th}}$  first code element.

5. A decoding device, configured to decode a low-density parity-check LDPC code and comprising:

15 an acquiring unit, configured to acquire a first code word, wherein the first code word is generated by an LDPC code encoding device according to a generator matrix and comprises M first code elements and M is a positive integer;

a processing unit, configured to acquire P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquire P first update values corresponding to the P first code elements; obtain, according to the P first update values, P second update values corresponding to the P first code elements; determine whether each second update value of the P second update values exceeds a code element quantization range; keep the P first code elements not updated when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range; update the P first code elements by using the P second update values when it is determined that each second update value of the P second update values does not exceed the code element quantization range; obtain, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generate a second code word according to the P second code elements, wherein the check matrix corresponds to the generator matrix and comprises N rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element; and

a determining unit, configured to acquire the second code word and determine whether the second code word is decoded successfully.

6. The decoding device according to claim 5, wherein the determining whether

each second update value of the P second update values exceeds a code element quantization range comprises:

sequentially determining whether each second update value of the P second update values exceeds the code element quantization range.

5        7. The decoding device according to claim 5 or 6, wherein the acquiring unit, the processing unit, and the determining unit each are formed by a circuit.

8. The decoding device according to any one of claims 5 to 7, wherein the processing unit comprises a storage module;

10        the storage module is configured to store the P first code elements, and is further configured to store the P second update values;

the processing unit reads, when determining that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range, the P first code elements stored in the storage module; and

15        the processing unit reads, when determining that each second update value of the P second update values does not exceed the code element quantization range, the P second update values stored in the storage module, so as to update the P first code elements.

9. A decoding method, used to decode a low-density parity-check LDPC code and comprising:

20        acquiring a first code word, wherein the first code word is generated by an LDPC code encoding device according to a generator matrix and comprises M first code elements and M is a positive integer;

acquiring P first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquiring P first update values corresponding to the P first code elements; obtaining, 25 according to the P first update values, P second update values corresponding to the P first code elements; determining whether each second update value of the P second update values exceeds a code element quantization range; keeping the  $a^{\text{th}}$  first code element not updated when it is determined that the  $a^{\text{th}}$  second update value of the P second update values exceeds the code element quantization range; updating the  $b^{\text{th}}$  30 first code element by using the  $b^{\text{th}}$  second update value when it is determined that the  $b^{\text{th}}$  second update value of the P second update values does not exceed the code element quantization range; obtaining, according to the foregoing update operation, P second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generating a second code word according to the P second code elements

corresponding to the  $n^{\text{th}}$  row of the check matrix, wherein the check matrix corresponds to the generator matrix and comprises  $N$  rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element; and

5 determining whether the second code word is decoded successfully.

10. The decoding method according to claim 9, wherein the determining whether each second update value of the  $P$  second update values exceeds a code element quantization range comprises:

10 sequentially determining whether each second update value of the  $P$  second update values exceeds the code element quantization range.

11. A decoding method, used to decode a low-density parity-check LDPC code and comprising:

15 acquiring a first code word, wherein the first code word is generated by an LDPC code encoding device according to a generator matrix and comprises  $M$  first code elements and  $M$  is a positive integer;

acquiring  $P$  first code elements corresponding to the  $n^{\text{th}}$  row of a check matrix; acquiring  $P$  first update values corresponding to the  $P$  first code elements; obtaining, according to the  $P$  first update values,  $P$  second update values corresponding to the  $P$  first code elements; determining whether each second update value of the  $P$  second update values exceeds a code element quantization range; keeping the  $P$  first code elements not updated when it is determined that the  $a^{\text{th}}$  second update value of the  $P$  second update values exceeds the code element quantization range; updating the  $P$  first code elements by using the  $P$  second update values when it is determined that each second update value of the  $P$  second update values does not exceed the code element quantization range; obtaining, according to the foregoing update operation,  $P$  second code elements corresponding to the  $n^{\text{th}}$  row of the check matrix; and generating a second code word according to the  $P$  second code elements, wherein the check matrix corresponds to the generator matrix and comprises  $N$  rows,  $1 \leq n \leq N$ ,  $P \leq M$ ,  $1 \leq a \leq P$ ,  $1 \leq b \leq P$ , and each second update value is a sum of each first update value and a corresponding first code element; and

20

25

30

determining whether the second code word is decoded successfully.

12. The decoding method according to claim 11, wherein the determining whether each second update value of the  $P$  second update values exceeds a code element quantization range comprises:

sequentially determining whether each second update value of the P second update values exceeds the code element quantization range.

13. A signal transmission system, comprising an LDPC code encoding device and the decoding device according to any one of claims 1 to 4, wherein

5       the LDPC code encoding device is configured to generate a first code word according to a generator matrix and transmit the first code word to the decoding device through a communications channel.

14. A signal transmission system, comprising an LDPC code encoding device and the decoding device according to any one of claims 5 to 8, wherein

10       the LDPC code encoding device is configured to generate a first code word according to a generator matrix and transmit the first code word to the decoding device through a communications channel.

15. A data storage system, comprising an LDPC code encoding device, the decoding device according to any one of claims 1 to 4, and a storage medium, wherein

      the LDPC code encoding device is configured to generate a first code word according to a generator matrix and store the first code word in the storage medium, and the decoding device is configured to acquire the first code word from the storage medium.

20       16. A data storage system, comprising an LDPC code encoding device, the decoding device according to any one of claims 5 to 8, and a storage medium, wherein

      the LDPC code encoding device is configured to generate a first code word according to a generator matrix and store the first code word in the storage medium, and the decoding device is configured to acquire the first code word from the storage medium.