

(51) International Patent Classification:  
*H04N 7/24* (2006.01)(21) International Application Number:  
PCT/KR2009/005798(22) International Filing Date:  
9 October 2009 (09.10.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
10-2008-0099907 11 October 2008 (11.10.2008) KR(71) Applicant (for all designated States except US): **INDUSTRY-UNIVERSITY COOPERATION FOUNDATION HANYANG UNIVERSITY** [KR/KR]; c/o Hanyang University, 17, Haengdang-dong, Seongdong-gu, Seoul 133-791 (KR).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **KIM, Hyun, Gyu** [KR/KR]; C-501, Dongmyeong Hometown, 118-27, Eungam 1-dong, Enpyeong-gu, Seoul 122-011 (KR). **LEE, Sin, Wook** [KR/KR]; 1484-7 #402 Daekyoung Velloche, Seocho 3-dong, Seocho-gu, Seoul 137-073 (KR). **SHIN, Hwa, seon** [KR/KR]; 604-406, MokryeonMaeul SKApt., 147, Yatap-dong, Bundang-gu, Seongnam-si, Gyeonggi-do 463-947 (KR). **LEE, Sang, Heon** [KR/KR]; 1417-1803, Castle Goldpark, Hwanggeum 1-dong, Suseong-gu, Daegu 706-934 (KR). **CHOI, Byeon, Ho** [KR/KR]; 101-302, Dongbu Apt., Jukjeon1-dong, Suji-gu, Yongin-si, Gyeonggi-do 448-985 (KR). **LEE, Chung, Ku** [KR/KR]; 16-406, Donga Apt., Bupyeong 1-dong, Bupyeong-gu, Incheon 403-762 (KR). **JANG, Euee, Seon** [KR/KR]; 101-2301, Gangbyeon Kunyoung Apt., Seongsu 1-ga 1-dong, Seongdong-gu, Seoul 133-794 (KR).(74) Agent: **JIMYUNG PATENT FIRM**; 6F, Jungwoo Bldg., 1689-1, Seocho 4-dong, Seocho-gu, Seoul 137-882 (KR).

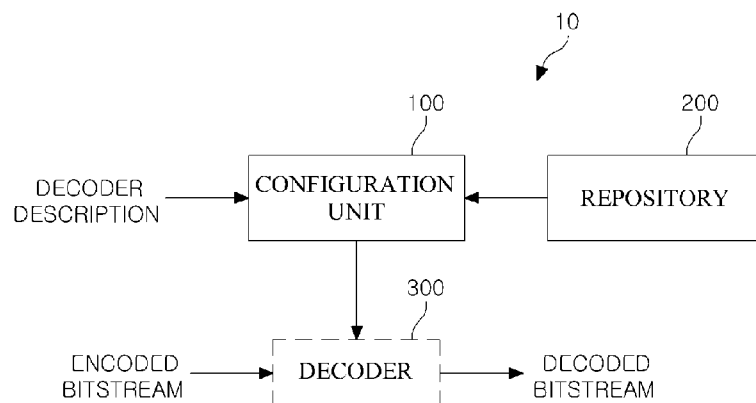
(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG,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

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

[Continued on next page]

(54) Title: DECODING APPARATUS, DECODING METHOD, ENCODING METHOD, AND COMPUTER-READABLE STORAGE MEDIUM

[Fig. 1]



(57) Abstract: Provided are a decoding apparatus and method. The decoding apparatus includes a repository and a configuration unit. The repository stores a plurality of functional units in which an input or output port is defined as a token. The configuration unit receives an identifier of at least one of the functional units and decoder description including connection relationship information which is composed of the token, selects pertinent functional units from the repository, and connects the selected functional units by using the token to configure a variable decoder. The token corresponds to data which are inputted or outputted between the functional units, and is defined as a name or an identifier and the data, and a first functional unit of the functional units has the input port which is defined as a first token and presence of a storage function for a first data corresponding to the first token.



---

**Published:**

- *without international search report and to be republished  
upon receipt of that report (Rule 48.2(g))*

## Description

### **Title of Invention: DECODING APPARATUS, DECODING METHOD, ENCODING METHOD, AND COMPUTER-READABLE STORAGE MEDIUM**

#### **Technical Field**

- [1] The following disclosure relates to a decoding apparatus and method, and in particular, to a decoding apparatus, decoding method and computer-readable storage medium, which decode signals encoded in various encoding schemes by using one device, and to an encoding method.

[2]

#### **Background Art**

- [3] In Moving Picture Experts Group (MPEG), establishing is begun of the standard for a video coding description framework, called Reconfigurable Video Coding (RVC). RVC is a technology for a codec framework that divides all sorts of video coding and decoding technologies by module and allows the divided technologies to be performed, and is a technology that increases flexibility for the use of a current video coding technology and may be easily applied to video coding technologies that is to be developed in the future.
- [4] The basic concept of RVC prepares a plurality of Functional Units (FUs) that are divided by function of performing each operation in decoding (further, coding), and performs decoding by combining suitable functional units in accordance with each coding scheme.
- [5] In relation to the functional units and a Decoding Descriptor (DD) defining the connection relationships between these, because a related art standard defines the functional units based on input/output signals (for example, input/output and control-context), it has considerable complexity. As an example, definition is differently made for each functional unit. For example, although specific inputs are basically identical, one functional unit refers to as YUV, but another functional unit refers to as YV12.
- [6] Consequently, it is difficult to actually use the functional units and to verify suitability for the connection relationships between these. That is, if the connection relationships between the functional units are verified, the possibility of connection between the functional units may be known after comparing all data characteristics, for example, the size (or type) of data, serial/parallel transmission schemes and internal algorithms.
- [7] In addition to the function or operation of the functional unit, the sizes or sequences of input/output signals are clarified. In this way, as implementation-dependent contents

are included, it is difficult to analyze standard documents, and quantity increases.

- [8] In defining the functional units, moreover, many limitations exist in the related art. According to the definition of the functional unit using the related art, for example, in a case where the functional unit normally operates when continuously or periodically receiving data (for example, when receiving data per macroblock (MB) or block), a parser functional unit (or a functional unit of transferring a corresponding signal) of parsing a encoded bitstream should continuously or periodically transmit necessary data to each of the functional units. Even when transmission data are periodically transmitted (for example, transmitted per Group Of Picture (GOP) or Video Object Plane (VOP)) like the widths and heights of pictures or data are not largely changed with time, the parser functional unit should continuously or periodically retransmit data for the widths and heights of pictures, and thus the parser functional unit has an unnecessary load.

[9]

## **Disclosure of Invention**

### **Technical Problem**

- [10] Accordingly, the present disclosure provides an RVC decoding apparatus and method, which enable to simply and clearly define the connection relationships between functional units in an RVC technology, increasing a standardization speed and enabling practical application.
- [11] The present disclosure also provides a decoding apparatus and method, which dynamically reconfigure a decoding logic in run time to be suitable for each of videos that are encoded in various encoding schemes.
- [12] The present disclosure also provides a decoding apparatus and method, which enable simply and clearly to write a decoding descriptor including connection information, and thus increase flexibility for the writing of the decoding descriptor and decrease a degree of difficulty for it, enabling even non-experts to easily implement their encoding/decoding schemes and improving the transmission efficiency of content.
- [13] The present disclosure also provides a decoding apparatus and method, which can verify the connection configuration of the functional units.
- [14] The present disclosure also provides an RVC system, which enables implementation users to implement functional units optimized for their technology characteristics by removing unnecessary restrictions in the implementation of the functional units, enhancing decoding performance.
- [15] The present disclosure also provides a decoding apparatus and method, which newly define functional units and thus increase efficiency for the operation of each of the functional units, improving efficiency for decoding.

- [16] The objects of the present invention are not limited to the above-described object, and the objects and advantages of the present invention other than the above-described object can be understood by description below and will be more apparent with reference to the embodiments of the present invention.

[17]

### **Solution to Problem**

- [18] In one general aspect, a decoding apparatus includes: a repository storing a plurality of functional units in which an input or output port is defined as a token; and a configuration unit receiving decoder description which includes an identifier of at least one of the functional units and connection relationship information which is composed of the token, selecting pertinent functional units from the repository, and connecting the selected functional units by using the token to configure a variable decoder, wherein the token corresponds to data which are inputted or outputted between the functional units, and is defined as a name or an identifier and the data, and a first functional unit of the functional units has the input port which is defined as a first token and presence of a storage function for a first data corresponding to the first token.
- [19] The first functional unit may store the current first data which is inputted to the input port, and may operate by regarding that the stored first data is continuously or periodically inputted until a next first data is inputted.
- [20] The first functional unit may pre-store a second data as a default value, and may operate by regarding that the second data is continuously or periodically inputted until the first data is inputted.
- [21] When a branch unit having an input port which is defined as a second token and at least two output ports which are defined as the second token are stored in the repository, and two or more second functional units in which an input port is defined as the second token are used to configure the variable decoder, the configuration unit may select the branch unit from the repository, and may connect each output port of the branch unit and an input port of each of the second functional units by using the second token.
- [22] In another general aspect, a decoding method includes: receiving an encoded bitstream and decoder description which includes an identifier of at least one functional unit and a token corresponding to data which are inputted or outputted between the functional units; selecting functional units corresponding to the identifier of the functional unit from a repository, and connecting the selected functional units by using the token to configure a variable decoder for decoding the encoded bitstream; and decoding the encoded bitstream through the variable decoder, wherein the con-

- figuring of a variable decoder includes selecting a first functional unit having an input port which is defined as a first token and presence of a storage function for a first data corresponding to the first token, and connecting the first functional unit with the selected functional unit by using the first token.
- [23] The decoding of the encoded bitstream may include storing, by the first functional unit, the current first data which is inputted to the input port, and operating by regarding that the stored first data is continuously or periodically inputted until a next first data is inputted.
- [24] The decoding of the encoded bitstream may include pre-storing, by the first functional unit, a second data as a default value, and operating by regarding that the second data is continuously or periodically inputted until the first data is inputted.
- [25] When a branch unit having an input port which is defined as a second token and at least two output ports which are defined as the second token are stored in the repository, and two or more second functional units in which an input port is defined as the second token are used to configure the variable decoder, the configuring of a variable decoder may further include selecting the branch unit from the repository, and connecting each output port of the branch unit and an input port of each of the second functional units by using the second token.
- [26] In another general aspect, an encoding method includes: encoding a bitstream; extracting a functional unit for performing decoding, which corresponds to the encoding, from a functional unit table; setting connection relationship between the extracted functional unit and a functional unit; generating decoder description in which the set connection relationship is described; and merging and transmitting the encoded bitstream and the decoder description, wherein a name or identifier of the functional unit, an input port of the functional unit which is defined as a token and presence of a storage function for data corresponding to the token and an output port which is defined as the token are listed by functional unit in the functional unit table, and the token corresponds to data which are inputted or outputted between the functional units, and is defined as a name or an identifier and the data.
- [27] The setting of connection relationship may include: extracting a token capable of an input or output of the extracted functional unit on the basis of a token pool; checking whether the extracted token is usable in common between two functional units to be connected; and defining connection relationship between the functional units by using names or identifiers of the two functional units to be connected, when the extracted token is usable in common.
- [28] The setting of connection relationship may include: extracting another functional unit for replacing at least one of the two functional units, when the token is not usable in common between two functional units; and checking whether the replaced functional

unit and a functional unit to be connected with the replaced functional unit use in common the token, wherein the extracting of another functional unit and the checking of common use of the token are repeated until the token is checked as being usable in common between the two functional units.

- [29] In another general aspect, a computer-readable storage medium storing a program which connects a plurality of functional units, which are included to accord a serial or parallel transmission scheme and a size and transfer sequence of data which are transferred between the functional units, to configure a decoder, the program executing: receiving decoder description which comprises an identifier of the functional unit and connection relation information which is composed of a token corresponding to data which are inputted or outputted between the functional units; and selecting functional units corresponding to the identifier, and connecting the selected functional units by using the token to configure the decoder, according to the decoder description, wherein the configuring of the decoder includes selecting a first functional unit having an input port which is defined as a first token and presence of a storage function for a first data corresponding to the first token, and connecting the first functional unit with the selected functional unit by using the first token.

- [30] Other features and aspects will be apparent from the following detailed description, the drawings, and the claims.

### **Advantageous Effects of Invention**

- [31] A decoding apparatus and method according to exemplary embodiments can decode videos encoded in various encoding schemes through one decoding apparatus.
- [32] The decoding apparatus and method according to exemplary embodiments define the connection relationships between functional units by using tokens, define each of the tokens by using only a name (or an identifier) and description, define the functional units through the tokens, and thus can simply and clearly define the connection relationships between the functional units, thereby enabling to develop a decoding apparatus and method to which a practical application-enabled RVC technology. Accordingly, the decoding apparatus and method according to exemplary embodiments can contribute to standardization.
- [33] The decoding apparatus and method according to exemplary embodiments easily enable to write decoder description to derive the quick distribution of an RVC technology and decrease the capacity of the decoder description, improving transmission efficiency in content transmission.
- [34] The decoding apparatus and method according to exemplary embodiments can dynamically reconfigure a reconfiguration decoder corresponding to each encoding video for a short time, and moreover, can concentrate and unify information through tokens,

a token pool and an FU table.

- [35] The decoding apparatus and method according to exemplary embodiments can provide verification conditions for all three stages including an encoder level, a decoder level and a standardization level. First, the decoding apparatus and method according to exemplary embodiments can verify connection with the token information of a decoder description when writing the decoder description, in the encoder level. Moreover, the decoding apparatus and method according to exemplary embodiments include token information even in the syntax parser description of the decoder description, enabling a decoder to prepare connection.
- [36] In the decoder level, the decoding apparatus and method according to exemplary embodiments can easily verify the connection between functional units when connecting the functional units, in the analysis of decoder description.
- [37] In the standardization level, the decoding apparatus and method according to exemplary embodiments can verify the suitability of connection relationship (i.e., connectable network) when defining a new functional unit. When connection is made with only the token of a token pool, suitability is best. When defining a new token, a case of using only the new token may occur. Accordingly, because of avoiding new definition if possible and trying to reuse, technologies are accumulated. Furthermore, a necessary conformance test can be performed in a stage necessary for standardization.
- [38] As a result, the decoding apparatus and method according to exemplary embodiments enhance efficiency for the operations of the functional units, improving efficiency for decoding.

[39]

### **Brief Description of Drawings**

- [40] FIG. 1 is a block diagram for describing a decoding apparatus according to an exemplary embodiment.
- [41] FIGS. 2 to 7 are exemplary diagrams for describing functional units.
- [42] FIG. 8 is a flow chart for describing a decoding method according to an exemplary embodiment.
- [43] FIGS. 9 to 11 are exemplary diagrams for describing the operations of functional units.
- [44] FIG. 12 is a conceptual view for describing a decoding apparatus and method according to an exemplary embodiment.
- [45] FIG. 13 is a conceptual view for describing a decoding apparatus and method according to another exemplary embodiment.
- [46] FIG. 14 is a conceptual view for describing a decoding apparatus and method according to another exemplary embodiment.



[47]

[48]

### **Best Mode for Carrying out the Invention**

[49] Hereinafter, exemplary embodiments will be described in detail with reference to the accompanying drawings. Throughout the drawings and the detailed description, unless otherwise described, the same drawing reference numerals will be understood to refer to the same elements, features, and structures. The relative size and depiction of these elements may be exaggerated for clarity, illustration, and convenience. The following detailed description is provided to assist the reader in gaining a comprehensive understanding of the methods, apparatuses, and/or systems described herein. Accordingly, various changes, modifications, and equivalents of the methods, apparatuses, and/or systems described herein will be suggested to those of ordinary skill in the art. Also, descriptions of well-known functions and constructions may be omitted for increased clarity and conciseness.

[50] In specification, encoded/decoded content is exemplified as video, but the kind of content is not limited thereto. It is apparent that various types of multimedia contents such as video and voice may be applied.

[51] The basic technical spirit of the present invention does not define the connection relationships between functional units in regard to input/output signals or input/output ports but defines the connection relationships as tokens in regard to the flow of signals.

[52] Moreover, tokens are defined with only their own names and descriptions corresponding to the minimum basic definition for data signals.

[53] Hereinafter, a decoding apparatus and method according to an exemplary embodiment will be described below with reference to FIGS. 1 to 8. FIG. 1 is a block diagram for describing a decoding apparatus according to an exemplary embodiment. FIGS. 2 to 7 are exemplary diagrams for describing functional units. FIG. 8 is a flow chart for describing a decoding method according to an exemplary embodiment.

[54] Referring to FIGS. 1 to 8, a decoding apparatus according to an exemplary embodiment may include a configuration unit 100 and a repository 200 (for example, a Video Tool Library (VTL)). The configuration unit 100 receives Decoder Description (DD) for configuring a decoder 300 that decodes an encoded bitstream in operation S310, and selects at least one functional unit from the repository 200. The configuration unit 100 connects the selected functional units to configure the decoder 300 suitable for decoding an encoded signal. Since the decoder 300 has variable characteristic in which it is reconfigured according to the decoder description, it will be referred to as a variable decoder for clearly discriminating with a related art decoder hereinafter.

- [55] The decoder description includes the names of functional units (see FIG. 2) necessary for configuring the variable decoder 300 that decodes an encoded bitstream, and connection information for connecting the functional units. A plurality of functional units are stored in the repository 200.
- [56] The following description will be made with reference to FIG. 8 on an operation in which the configuration unit 100 configures the variable decoder 300.
- [57] The decoder description includes Parser Description (PD) being information necessary for configuring a parser functional unit that parses an input bitstream, and FU network description (FND) being connection information necessary for connecting the functional units. When the configuration unit 100 receives the decoder description in operation S310, it generates the parser functional unit on the basis of the parser description in operation S320. The configuration unit 100 selects one or more functional units from the repository 200 on the basis of the FND, and connects the selected functional units to configure a decoding logic that is configured with the functional units in operation S330.
- [58] The decoding apparatus 10 configures the variable decoder 300 by using the parser functional unit and the decoding logic to be included in operation S340.
- [59] In configuring the variable decoder 300, unlike the above described, the decoding apparatus 10 may use content stored in the repository 200 without generating the parser functional unit. In this case, the parser description may include only the name or identifier of the parser functional unit.
- [60] As described above, in configuring the variable decoder, the present invention may be modified in various forms without departing from the technical subject matters of the present invention, and also the above-described operations S310 to S340 have not been described based on time series limitation.
- [61] The decoding apparatus 10 and method according to an exemplary embodiment configure the variable decoder according to the decoder description that is inputted together with an encoded bitstream even when the input bitstream is encoded in all sorts of encoding schemes, and thus can dynamically a suitable decoder based on a corresponding encoding scheme, thereby decoding signals encoded in various encoding schemes through one device.
- [62] Hereinafter, functional units and tokens will be described in more detail.
- [63] First, a token denotes a signal (i.e., data) that is transferred between the functional units, and therefore one token may naturally define the connection relationship, including the input/output relationship between two functional units, between the functional units. In an exemplary embodiment, defined are the connection relationships between the functional units that configure the variable decoder 300 by the use of one or more tokens.

- [64] A token according to an exemplary embodiment is defined with only its own name (or identifier) or its own name and identifier, and description that describes only its own characteristic discriminated with other tokens.
- [65] In an exemplary embodiment, the name of a token may be an abbreviation or a variable length word that enables to instinctively recognize the kind of the token, and an identifier may be a fixed-length number or an abbreviation for easily configuring decoder description.
- [66] Moreover, the description of a token defines only the most basic characteristic (for example, a bitstream itself and a control signal) of a signal (i.e., data), and does not define the size, transfer sequence and transmission type (for example, serial or parallel transmission) of the signal.
- [67] The functional unit and the token will be described in more detail with reference to FIGS. 3 to 6.
- [68] When two tokens are connected to a token X as illustrated in FIG. 3, largely three cases will be described below according to the intent of implementation.
- [69] First, FIG. 4 illustrates different examples in which each token X is processed in series. The first figure of FIG. 4 prefers performing transmission in an abc type. The second figure of FIG. 4 prefers performing transmission in a cba type. If data transfer sequence has been defined when defining the token X, cases such as figures should be configured with different tokens, and moreover, since the functional unit A of the former and the functional unit A of the latter represent different outputs, they should be registered as different functional units.
- [70] FIG. 5 illustrates examples in which each token X is processed in series and in parallel. The first figure of FIG. 5 transmits data 'abc' in series. The second figure of FIG. 5 divides the bus of the token X by three and independently transmits data 'a' , data 'b' and data 'c' . In this case, the first figure may set the input/output of the token X at the size of the data 'a' , and the second figure may set the input/output of the token X at the size of "a+b+c". If size has been defined when defining the token X, cases such as figures should be configured with different tokens, and moreover, since the functional unit A of the former and the functional unit A of the latter represent different outputs, they should be registered as different functional units. However, like the above-described example, this also makes the description of standard documents complex, and moreover, may enforce a specific scheme to an implementation user.
- [71] Next, FIG. 6 illustrates an example in which the token of a serial concept is processed in parallel at the inside. When configuring an algorithm, this case corresponds to a case which is improved in a parallel scheme that is proposed and implemented from a serial viewpoint. The former is a case that transmits the token X according to the description of an algorithm, and the latter is a case that divides into

three tokens to implement parallelism. In the latter, however, in a case of internally defining as “ $X=X_a+X_b+X_c$ ”, i.e., a case having definition in which the token X is processed to be configured with ‘ $X_a$ ’, ‘ $X_b$ ’ and ‘ $X_c$ ’ in parallel, three operations are performed at the inside but they are identically performed when seen from the outside. If parallelism has been defined when defining the token X in the standard, cases such as figures should be configured with different tokens, and moreover, since the functional unit A of the former and the functional unit A of the latter represent different outputs, they should be registered as different functional units. However, like the above-described example, this also makes the description of standard documents complex, and moreover, may enforce a specific scheme to an implementation user.

[72] In FIG. 6, when “ $X=X_a+X_b+X_c$ ” is defined as a internal regulation, because input/output results are the same, there is no need that defines this case in detail while additionally defining a functional unit in the standard.

[73] By gathering the names (or an identifiers, or names and identifiers) and descriptions of the tokens to configure a token pool, the token pool may be used when writing decoder description and verifying a new functional unit. An example of the template of the token pool is as listed in the following Table 1, and an embodiment of the token pool may be as listed in the following Table 2.

[74] Table 1

[Table 1]

[Table ]

Template of Description of Token Pool

| ID                                                             | NAME                                            | DESCRIPTION                                                                      |
|----------------------------------------------------------------|-------------------------------------------------|----------------------------------------------------------------------------------|
| Unique numerical number which represent this Token in this row | Unique name to represent this Token in this row | Some text to describe characteristics or functionality of this Token in this row |

[75] Table 2

[Table 2]

[Table ]

Token Pool

| ID | NAME         | DESCRIPTION                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | CBP          | An array of 6 Boolean values for describing whether the individual block includes one or more DCT coefficients that are decoded as (LAST, RUN, LEVEL) symbol                                                                                                                                                                                                                                                                                                                           |
| 2  | Quant_scale  | a 5-bit integer value for describing the quantization step in the current MB, which is derived from the running QP and dQuant to calculate QP and determine dc_scaler for current MB by referring to Table 7-1 (14496-2:2004)                                                                                                                                                                                                                                                          |
| 3  | Dquant_scale | a 2-bit code which specifies the change in the quantiser scale, for I-, P-, and S(GMC)-VOPs<br>dquant_codes and corresponding values: (ISO/IEC 14496-2 Table 6-32)<br>dquant_scale = 00 → value = -1<br>dquant_scale = 01 → value = -2<br>dquant_scale = 10 → value = 1<br>dquant_scale = 11 → value = 2                                                                                                                                                                               |
| 4  | AC_pred_flag | to determine the direction of inverse DC prediction                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 5  | AltScan      | Indicates zigzag or alternate scan. AltScan flag is 0 when zigzag, 1 when alternate scan. It must set every block                                                                                                                                                                                                                                                                                                                                                                      |
| 6  | Ac_pred_dir  | A flag to communicate the direction of prediction to the IAP block but also info about the scan mode of the block to the inverse scan block. AC_PRED_DIR = -2 => NEWVOP flag. AC_PRED_DIR = -1 => An uncoded block so skip inverse AC prediction. AC_PRED_DIR = 0 => No inverse AC prediction but use zig zag inverse scan. AC_PRED_DIR = 1 => Prediction from the left and use alternate vertical scan. AC_PRED_DIR = 2 => Prediction from the top and use alternate horizontal scan. |
| 7  | RefMB        | A buffer of MB data to have a relation with current MB                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 8  | MB           | A MB buffer to save a target MB data                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 9  | MB_type      | Indicate the type of MB                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 10 | Qp           | Quantization parameter of an MB which is calculated in fuDCR.                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 11 | DC_left      | to compute the direction of DC/AC prediction for 6 (4 Y/1 Cb/1 Cr) block(s)                                                                                                                                                                                                                                                                                                                                                                                                            |
| 12 | DC_top       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 13 | DC_lefttop   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

- [76] In this way, by defining tokens with only their own names (or identifiers, or names and identifiers) and descriptions, the definition of the token is abstracted, and by including only implementation-independent content, it can be prevented that standard content is limited and used to a specific company or a specific industry. Furthermore, for functional units, the functional units are defined through a separate FU table and contents for the functional units are not described in the token pool, thereby concentrating information.
- [77] By describing a token identifier (ID) instead of a token name or both the token ID and the token name in the token pool, an identifier may be used in the writing of decoder description or the configuring of a functional unit. While using a token ID, an ID-Name map may be used for again recovering as a name. In this case, a functional such as "IDToName( )/NameToID( )" may be included.
- [78] A functional unit stored in the repository 200 has its own connection information for the configuring and verification of a variable decoder based on decoder description.
- [79] That is, each functional unit has information for the kind of an input/output signal that may be processed by it and information for the number of input/output ports. This information may be defined with only tokens. That is, as described above, a token may define the connection relationship between functional units, and naturally, as illustrated in FIG. 2, the input/output port of a functional unit 210 may be defined together.
- [80] To provide a description through an example with reference to FIG. 2 and the Table 1, the functional unit 210 has two input ports and one output port. When the functional unit 210 receives and processes signals corresponding tokens TA and TB to output a signal corresponding to a token TC, it may be defined as follows.
- [81] FU I
- [82] Input : TA, TB
- [83] Output : TC
- [84] That is, by defining the input/output relationship between functional units from a data viewpoint instead of an interface viewpoint, the complexity of connection information decreases and verification may be performed.
- [85] A functional unit as illustrated in FIG. 7 may exist. That is, the input port of the functional unit in FIG. 7 may be defined as one token TA, and the output port of the functional unit may be defined as one or more tokens TA (for example, FIG. 7 illustrates a case in which the number of output ports is two). That is, the functional unit receives data corresponding to the token TA to output the same data through one or more output ports. When the flow of data is consumptive (i.e., a case in which data are consumed when they are provided from any one branch point to any one location and are not provided to another location), the functional unit is one that provides the same data to a plurality of locations. Even in the functional unit, all input/output ports may

be defined as tokens, and the connection relationship between the functional unit and another functional unit may be defined.

- [86] By configuring connection information for each functional unit (i.e., input/output relationship information) with only a token, the configuration unit 100 may quickly configure the variable decoder 300 on the basis of decoder description and may verify whether the connection between the functional units is suitable in configuring. A detailed description associated with this will be made below with reference to FIG. 9.
- [87] Since the possibility of connection between a specific functional unit and another functional unit may be verified beforehand owing to the definition of a token, the functional unit of another codec may be used. Furthermore, if a new specific codec is defined and only the structure of a token based on it is defined, the suitability of the codec may be verified. A conformance test may be made in a codec level. For example, a bitstream is generated by encoding through a self-algorithm, and decoder description suitable for the bitstream is transferred to a decoder. The self-algorithm may add some functional units to the existing codec or may be subtracted some functional units from the existing codec. This may be performed at any time when only decoder description is analyzed and the suitability of a token used in it is determined. This characteristic has a meaning that may flexibly meet a new codec to be emerged in the future.
- [88] Functional units may have a case in which signals inputted to the respective input ports of the functional units are sequence-dependent. For example, when a token  $T_A$  is a flag, a token  $T_B$  is a pixel signal and the functional unit 210 is implemented so that it may receive a token  $T_B$  to operate after the token  $T_A$  is first inputted, connection information for the functional unit 210 includes token precedence information that represents operation characteristic such as " $T_A > T_B$ ", and thus definition may be clearly made with only a token even in a case where the operation characteristic of a functional unit is sequence-dependent of an input signal. Herein, '>' represents precedence between two or more tokens, and is used to denote a case of being relatively first inputted. To provide a description through another example, when a functional unit is implemented so that it may receive tokens  $T_{B1}$  and  $T_{B2}$  to operate after token  $T_A$  is first inputted, the token precedence of the functional unit may be defined as " $T_A > (T_{B1}, T_{B2})$ ". Herein, ',' denotes that tokens  $T_{B1}$  and  $T_{B2}$  should be inputted together at a specific moment.
- [89] To provide a description though another example, when token precedence is defined as " $T_A > (T_{B1}, T_{B2}^*)$ ", this means that a token  $T_A$  should be inputted before tokens  $T_{B1}$  and  $T_{B2}$  but the token  $T_{B2}$  may not be inputted. That is, '\*' denotes not being used or connected depending on cases. When not being used or connected depending on cases, a separate default value may be designated and clarified on the basis of the default

value of a corresponding token. That is, any signal may not be inputted (or outputted) to an input port (or from an output port) that is defined as a token in which “\*” is clarified. Although this case, depending on cases, a functional unit may be defined as that in which a default value is designated and thereby the signal of the default value is inputted (or outputted) to an input port (or from an output port), and thus the functional unit may operate according to the definition.

[90] Grammatical contents representing the token precedence are not limited thereto, and may be further added.

[91] In other words, when each functional unit operates according to the sequence of a token that is inputted to an input port, the operation of the functional unit is defined as the sequence of the token, and the sequence of the token is indicated through the name or identifier of the token.

[92] The above-described description on the connection information included in the functional unit may be charted as a functional unit table like the following Table 3.

[93] Table 3

[Table 3]

[Table ]

Function Unit Table

| Number | Name  | Description                      |                                                    |                                 |
|--------|-------|----------------------------------|----------------------------------------------------|---------------------------------|
|        |       | Input                            | Output                                             | Token Precedence                |
| 1      | FU I  | T <sub>A</sub>                   | T <sub>C</sub>                                     | T <sub>A</sub> > T <sub>B</sub> |
|        |       | T <sub>B</sub>                   |                                                    |                                 |
| 2      | FU II | T <sub>C</sub>                   | T <sub>D</sub>                                     |                                 |
|        |       |                                  | T <sub>E</sub>                                     |                                 |
| ...    | ...   | ...                              | ...                                                | ...                             |
| 10     | FU X  | T <sub>D</sub> or T <sub>E</sub> | T <sub>F</sub> or T <sub>G</sub> or T <sub>H</sub> |                                 |
| ...    | ...   | ...                              | ...                                                | ...                             |

[94] As listed in the Table 3, the input port and output port of each functional unit and the characteristic of an input/output signal (i.e., data) may be simply defined as a token. Herein, the FU table represents including even token precedence as an example, but it is not limited thereto. The token precedence may not be included.

[95] Two or more tokens may be inputted to the one input port of a functional unit, and also two or more tokens may be outputted from one output port. In this case, like the description of the FU X described in the Table 2, an operation may be defined as a group “(T<sub>D</sub> or T<sub>E</sub>), (T<sub>F</sub> or T<sub>G</sub> or T<sub>H</sub>)” .



- [96] By adding to the Table 3, the input port of a functional unit may be divided into a token and the presence of a storage function for data corresponding to the token and be thereby defined. In this case, a functional unit table may be charted for the input port of a functional unit like the following Table 4.

[97] Table 4

[Table 4]

[Table ]

Function Unit Table

| Number | Name | Description              |                |                |
|--------|------|--------------------------|----------------|----------------|
|        |      | Input                    |                | Output         |
|        |      | With memory              | Without memory |                |
| 1      | FU I | T <sub>A</sub> (Default) | T <sub>B</sub> | T <sub>C</sub> |

- [98] To provide a more detailed description with reference to FIG. 2, any one of the input ports of the functional unit 210 is defined as a token T<sub>A</sub> and that in which there is a storage function for a first data corresponding to the token T<sub>A</sub>. Another one of the input ports of the functional unit 210 is defined as a token T<sub>B</sub> and that in which there is no storage function for a second data corresponding to the token T<sub>B</sub>. The output port of the functional unit 210 is defined as a token T<sub>C</sub>.
- [99] To provide a description on the operation of the defined functional unit 210, the functional unit 210 receives the first data corresponding to the token T<sub>A</sub> and the second data corresponding to the token T<sub>B</sub> to operate, and outputs a third data corresponding to a token T<sub>C</sub>. If the tokens T<sub>A</sub> and T<sub>B</sub> are defined as having no storage functional, the functional unit 210 may stop operating and wait until before the first data corresponding to the token T<sub>A</sub> is inputted although the second data corresponding to the token T<sub>B</sub> is inputted, according to system characteristic (for example, a system in which the flow of data is consumptive). According to the Table 4, on the other hand, since one input port is defined as the token T<sub>A</sub> and having a storage function for the first data, the functional unit 210 stores the first data corresponding to the token T<sub>A</sub> that is currently inputted to the input port, and operates by regarding that a first data is continuously or periodically inputted until a next first data is inputted. Moreover, since another input port is defined as the token T<sub>B</sub> and having no storage function for the second data, the functional unit 210 does not store the first data corresponding to the token T<sub>A</sub> that is inputted to the input port and receives the second data in real time to operate.
- [100] To provide a detailed description though an example, the first data corresponding to the token T<sub>A</sub> may be data for the width and height of a picture that is provided peri-

odically (for example, per Group Of Picture (GOP) or Video Object Plane (VOP)), and the second data corresponding to the token  $T_B$  may be data for the motion vector of each macroblock that is continuously provided.

- [101] If the functional unit 210 does not store the first data corresponding to the token  $T_A$ , a parser functional unit retransmits data for the width and height of a picture per macroblock. When the data for the width and height of the picture are not almost changed, the parser functional unit has unnecessary load in which it should repetitively retransmit the same data for each operation of the functional unit 210. Accordingly, the input port of the functional unit 210 is defined as the presence of a storage function for a token and data corresponding to the token, and when the input port of the functional unit 210 has the storage function, the load can decrease.
- [102] When the first data corresponding to the token  $T_A$  is inputted to an input port having a storage function, since the functional unit 210 stores the first data, the first data may not yet be inputted. The functional unit 210 may stop operating and wait until before the first data corresponding to the token  $T_A$  is inputted although the second data corresponding to the token  $T_B$  is inputted, according to system characteristic (for example, a system in which the flow of data is consumptive). Accordingly, the functional unit 210 may pre-store a fourth data as a default value. In this case, the functional unit 210 may regard that the fourth data is inputted to the input port and operate. Furthermore, when the first data is inputted, the functional unit 210 stores the first data and may operate by regarding that a first data is being inputted until a next first data is inputted. That is, the input/output port of each functional unit may be defined through a token. For the efficient operation of the each functional unit, the input port of the each functional unit may be defined as a token and the presence of a storage function for data corresponding to the token.
- [103] As described above, since the each functional unit operates like the defined in description, different token inputs may be discriminated. A token and the operation of a functional unit will be additionally described below with reference to FIGS. 9 to 11.
- [104] Tokens are identical, which means equivalence. When not equivalence, tokens should be different from one another.
- [105] In FIG. 9, a mixer 1 FU receives a token  $T_A$  to again output a token  $T_A$ . When the token  $T_A$  is a video input, the mixer 1 FU may perform image processing such as filtering or add a caption. Tokens  $T_A$  perform certain operations, but the tokens  $T_A$  are equivalent in regard to input/output data. That is, since the types of input/output data are the same although the contents of the input/output data are different, the tokens  $T_A$  are equivalent because they do not get out of abstract meanings that are defined. A mixer 2 FU receives two tokens  $T_A$  as inputs to output one token  $T_A$ . This corresponds to a case in which the mixer 2 FU receives different audios to perform a mixing

operation when the token  $T_A$  has an audio type. The audio mixing operation enables to listen window sound, Compact Disk (CD), microphone and line inputs through one speaker, in personal computers (PCs). In this case, it is apparent that the tokens  $T_A$  of inputs/outputs are equivalent. On the other hand, in a case where video is applied to a mixer 3 FU as an example, the mixer 3 FU receives a color key through a token  $T_C$  input and applies the color key to the input of a token  $T_B$  to perform transparency processing. In this case, the input of the token  $T_B$  is overlaid with the input of the token  $T_A$  and may be outputted as the type of a token  $T_A$ . Herein, since the token  $T_A$  and the token  $T_B$  perform different functions such as a main and a sub, they are not equivalent. If the mixer 3 FU defines a token  $T_B$  as a token  $T_A$  like the mixer 2 FU, the token  $T_B$  may be included in the main or the sub when it is connected to the token  $T_A$  output of the mixer 2 FU. An output different from an output, desired by the mixer 3 FU, may arise.

- [106] The internal operation of a functional unit is not known to the outside, and only characteristic for using the functional unit is known to the outside through a token and token precedence. Considering this characteristic, the mixer 3 FU should define different tokens in locations where different inputs are desired and thereby divide undesired inputs. In other words, the functional unit should perform all functions for token input/output that is clarified by it.
- [107] Referring to FIG. 10, an FU A has three outputs, which are designated as the type of a token group such as “( $T_A$  or  $T_B$ ), ( $T_A$  or  $T_C$ ), ( $T_A$  or  $T_D$ )”. Because an FU B, an FU C and an FU D use tokens  $T_A$  as inputs, when the FU A is connected to the FU B, the FU C and the FU D through the tokens  $T_A$ , equivalent output is made for all three outputs.
- [108] Referring to FIG. 11, unlike FIG. 10, because an FU B, an FU C and an FU D respectively use a token  $T_B$ , a token  $T_C$  and a token  $T_D$ , an FU A is respectively connected to the FU B, the FU C and the FU D through the token  $T_B$ , the token  $T_C$  and the token  $T_D$ . As described above, because each functional unit should perform all functions for token input/output that is clarified by it, the FU A performs different outputs instead of the equivalent output of the token  $T_A$ , for each output. In other words, although the inside of the FU A is not known, the FU A may perform equivalent output and sometimes perform output by changing a format to another format. For this, the FU A may be used. As an actual example, the FU A basically outputs RGB, but it may divide RGB into R, G and B to output them according to the FU B, the FU C and the FU D that are connected to it.
- [109] To provide a brief description, as listed in the Table 1, a token is defined with only an identifier and description in a token pool, and consequently, the connection information of a functional unit may be simply defined with only a token as listed in the Table 3 or the Table 4.

- [110] As listed in the Table 3 or the Table 4, the connection information of a plurality of functional units may be recorded in one lookup table, and each functional unit may store its own connection information.
- [111] In a case of using a token according to an exemplary embodiment, the writing of decoder description becomes very easy, and the configuring of the variable decoder 300 may be quickened and simplified.
- [112] For example, when intending to configure the variable decoder in FIG. 9, only the following content is described in the FND of decoder description.
- [113] <Example of DD>
- [114] Parser FU - FU III (TA)
- [115] Parser FU - FU II (TB)
- [116] FU III - FU V (TC)
- [117] FU II - FU V (TD)
- [118] That is, a DD writer does not care that each functional unit inputs/outputs a signal having a certain size and sequence in which input sequence and in which transmission scheme (for example, serial or parallel transmission), and merely cares the processing function and connection information of the each functional unit.
- [119] Considering a broadcasting environment that may become an example of dynamic reconfiguration, a bitstream transmitted from a broadcasting station may not be limited to the product and device of a specific company. In other words, the same decoder description and a bitstream may be transmitted irrespective of a bitstream that is written based on a serial scheme and a bitstream for a parallel scheme. If decoder description is written to be limited to a serial scheme or a parallel scheme, it should be differently transmitted depending on cases, and consequently, a case incapable of viewing sometimes may occur in a specific product. Accordingly, only the minimum element should be defined in decoder description, and implementation-dependent characteristic is not reflected in a token that is designated by the decoder description. Through this, decoder description identically operates in a design-time stage for producing a VTL and a run time that is actually used by a user.
- [120] The configuration unit 100, which analyzes decoder description to configure the variable decoder 300, sequentially selects each functional unit according to the described in decoder description and sets connection between the functional units, thereby configuring the variable decoder 300 in a simple scheme.
- [121] Particularly, the configuration unit 100 can easily verify whether connection information, described in decoder description, is suitable. For example, when "[FU III - FU V (TC)]" is described in the FND of decoder description, the configuration unit 100 may check whether a token TC is in the output-enabled signal (i.e., a token) of an FU III and the input-enabled signal of an FU V on the basis of the connection in-

formation (for example, content described in the Table 2) between the FU III and the FU V to verify whether connection is suitable, appropriately meeting a result of the verification.

- [122] When the token TC is not in the output-enabled signal (i.e., a token) of an FU III or the input-enabled signal of an FU V, the configuration unit 100 checks that the FND of decoder description is wrongly described to output an error message, and may stop configuring the variable decoder 300. The configuration unit 100 may include a verification unit (not shown) for this.
- [123] As described above, by defining the connection information of a functional unit with only a token, a functional unit or an FU table itself is easily understood, and by unifying the connection medium between a token pool and functional units with a token name (or an identifier), the disaccord of information is prevented in advance and the efficiency of information concentration is improved. Furthermore, by allowing the repetitive description of a token for a functional unit, a corresponding function may have generic characteristic. Through this, a base that may concentrate at the technical feature of a functional unit and integrate similar functional units can be prepared.
- [124] In an exemplary embodiment, as a premise condition, the implementation of a functional unit is made in one company or one party that shares the reference of FU implementation.
- [125] That is, since a DD writer or the configuration unit 100 never considers the detailed processing operation of a functional unit, the detailed format of an input/output signal (i.e., data), a transmission scheme (for example, serial or parallel transmission) and the internal sequence of a signal, an FU producer should implement a functional unit while autonomously defining the detailed format of a signal (i.e., data) that is inputted/outputted between functional units to be stored together in one repository, a transmission scheme (for example, serial or parallel transmission), the internal sequence of a signal, connectable functional units and the processing algorithm of each functional unit.
- [126] Due to these, the producer of a functional unit secures flexibility for contents other than a token, and thus can produce various types of functional units through various schemes. Accordingly, technologies for functional units can further progress.
- [127] According to an exemplary embodiment, an FU producer may determine the interface scheme between functional units without being limited by a signal transfer sequence, the classification of the serial/parallel scheme and input/output scheme of a signal and size. For example, since a signal transfer sequence, the classification of the serial/parallel scheme and input/output scheme of a signal and size in a case of configuring a functional unit in hardware may differ from those of a case of configuring a functional unit in software, each FU producer may determine a processing standard

and a detailed signal suitable for an FU implementation scheme that is selected by it. Alternatively, each FU producer can make the implementation of a high-performance FU flexible by using its own special know-how.

- [128] To provide a summary on the above described, when generating decoder description in an encoding operation, by referring to a token pool and an FU table, functional units necessary for implementing a decoder and a token for connecting these can be easily comprehended and generated. At this point, a decoder to be configured with a functional unit and a token can be easily verified. This is because the connection relationships between functional units are clearly emerged with tokens. Moreover, since decoder description is generated using the identifier of a token and the identifier of a simplified and unified functional unit, the capacity of the decoder description may be reduced. Therefore, the transmission efficiency of the decoder description can increase.
- [129] Moreover, because the configuration unit 100 connects functional units with the simple identifier of a token, it can be configured with a simple logic.
- [130] Since the connection information of a functional unit is only composed of the identifier of a token, functional units having the same characteristic may be determined as one FU regardless of an interface scheme other than a token, for example, a data transfer sequence, the classification of a serial/parallel transmission scheme and size for a token.
- [131] In a case of defining a functional unit on the basis of a data transfer sequence, the classification of a serial/parallel transmission scheme and size for a token, a functional unit of performing the same processing function should also be defined in plurality according to a data transfer sequence, the classification of a serial/parallel transmission scheme and size. Moreover, a plurality of functional units for performing the same processing function should be stored in a repository. In an exemplary embodiment, however, since the connection relationships of functional units are defined with only tokens, the connection information of the functional units are simplified and plural numbers are not required in functional units having the same characteristic, preventing the capacity of a repository from unnecessarily increasing.
- [132] In a standardization level, since a data transfer sequence, the classification of a serial/parallel transmission scheme and size for a token are not considered and only tokens are considered, all users can easily know the standard content and easily use the standardized technology.
- [133] Moreover, when defining a new functional unit is required, by first checking whether a token exists in the token pool of the Table 1, whether a usable token exists in a new functional unit is first determined. When the usable token exists, this means that the new functional unit may be connected to another functional unit, and thus the new functional unit may be immediately used. When the usable token does not exist, a new

- token is defined with only a name or an identifier and description, and the connection information of the new functional unit may be defined using the defined token and used.
- [134] In this way, by defining a token with only an identifier and description and defining a functional unit with the token, information for a token pool, an FU table, tokens and functional units may be concentrated and unified.
- [135] Hereinafter, a decoding apparatus and method according to exemplary embodiments will be described in detail through specific embodiments.
- [136] First, a decoding apparatus and method according to an exemplary embodiment will be described below with reference to FIGS. 1 to 12. FIG. 12 is a conceptual view for describing a decoding apparatus and method according to an exemplary embodiment.
- [137] Referring to FIGS. 1 to 12, the configuration unit 100 receives decoder description to configure a variable decoder 300. Herein, decoder description may include the identifier (for example, a numeral or a name for discriminating functional units) of a functional unit that is used in the variable decoder 300, FU connection-relationship information (i.e., an FND) that is composed of the identifier (for example, a numeral or a name for discriminating tokens) of a token for connecting functional units, and parser description.
- [138] First, the configuration unit 100 refers to the parser description included in the decoder description, and when a parser functional unit 205 corresponding to the parser description exists in the repository 200, the configuration unit 100 selects and positions the parser functional unit 205 from the repository 200. Herein, the parser functional unit 205 parses another encoded bitstream and provides parsed signals to other functional units. For example, the parser functional unit 205 parses an encoded bitstream into a control signal for decoding and an encoded video signal to provide the parsed signals to other functional units. Next, among a plurality of functional units 210, 215, 220, 225 and 230 that exist in the repository 200, the functional units 215, 220 and 230 corresponding to the identifiers of functional units are selected from the repository 200 and are positioned. The parser functional unit 205 and the functional units 215, 220 and 230 are connected using the identifiers of tokens.
- [139] In more detail, the configuration unit 100 selects the parser functional unit 205 having output ports that are defined as a token  $T_A$  and a token  $T_B$ , from the repository 200 and positions the selected parser functional unit 205 as illustrated in FIG. 12. Furthermore, as illustrated in FIG. 12, the configuration unit 100 positions the FU III 220 having an input port and an output port that are respectively defined as a token  $T_A$  and a token  $T_B$ , positions the FU II 215 having an input port and an output port that are respectively defined as a token  $T_B$  and a token  $T_D$ , and positions the FU V 230 having input ports that are defined as a token  $T_C$  and a token  $T_D$  and an output port that is

- defined as a token  $T_E$ . Subsequently, the configuration unit 100 connects the parser functional unit 205 and the functional units 215, 220 and 230 by using each token identifier.
- [140] In an operation where the configuration unit 100 selects the functional units 215, 220 and 230 from the repository 200, the configuration unit 100 may verify in advance whether the functional units 215, 220 and 230 may be connected by using the names or identifiers (for example,  $T_A$ ,  $T_B$ ,  $T_C$ ,  $T_D$  and/or  $T_E$ ) of tokens. As described above, since the configuration unit 100 generates decoder description on the basis of the input/output port token information and token precedence information of a functional unit necessary for implementing the variable decoder 300 based on the FU table of FIG. 2 even in an operation of generating the decoder description, it can verify beforehand whether functional units may be connected.
- [141] Next, a decoding apparatus and method according to another exemplary embodiment will be described below with reference to FIGS. 1 to 13. FIG. 13 is a conceptual view for describing a decoding apparatus and method according to another exemplary embodiment.
- [142] In this embodiment, unlike the above-described embodiment, the parser functional unit 205 does not exist in the repository 200. In this case, the configuration unit 100 generates a parser functional unit 205 by using the parser description of decoder description. The configuration unit 100 positions the generated parser functional unit 205, and selects and positions other functional units 215, 220 and 230. Subsequently, the configuration unit 100 connects the parser functional unit 205 and the functional units 215, 220 and 230 by using the names or identifiers (for example,  $T_A$ ,  $T_B$ ,  $T_C$ ,  $T_D$  and/or  $T_E$ ) of tokens to configure a decoder 300.
- [143] Next, a decoding apparatus and method according to another exemplary embodiment will be described below with reference to FIGS. 1 to 14. FIG. 14 is a conceptual view for describing a decoding apparatus and method according to another exemplary embodiment.
- [144] Referring to FIGS. 1 to 7, a parser functional unit 203 outputs data corresponding to a token  $T_A$  through one output port, but functional units 250 and 260 receive data corresponding to the token  $T_A$ . At this point, in a case where the flow of data is consumptive, data are consumed when they are provided from any one branch point to the functional unit 250, and when data are not provided to the functional unit 260, the configuration unit 100 selects and positions a branch functional unit 240 (which has been described above with reference to FIG. 7) from the repository 200. Subsequently, the configuration unit 100 connects the respective output ports of the branch functional unit 240 with the input port of the functional unit 250 and the input port of the functional unit 260 by using the token  $T_A$ , thereby configuring a virtual decoder in



FIG. 14.

- [145] Moreover, when it is defined that one of the input ports of the functional unit 260 has a storage function for the token  $T_A$  and data corresponding to the token  $T_A$ , the parser functional unit 203 can decrease load in which it should periodically or continuously provide data corresponding to the token  $T_A$ . Consequently, the configuration of the internal logic of the parser functional unit 203 can be simplified, and by decreasing undesired operations, efficient decoding can be performed.
- [146] For appropriately performing the above-described decoding, the following encoding scheme is required.
- [147] After encoding a bitstream to transmit or determining an encoding scheme, a functional unit for performing decoding corresponding to the encoding scheme should be extracted from a functional unit table. Herein, as listed in the above-described Table 4, the names or identifiers of functional units, the input port of each functional unit that is defined as a token and the presence of a storage function for data corresponding to the token and an output port that is defined as the token may be listed by functional unit in the functional unit table.
- [148] When functional units are selected/extracted, the decoding apparatus sets the connection relationship between the functional units and generates decoder description in which the connection relationship is described. Subsequently, the decoding apparatus may merge and transmit an encoded bitstream and the decoder description.
- [149] In an operation of setting the connection relationship between the functional units, the decoding apparatus extracts a token capable of the input or output of the extracted functional unit on the basis of a token pool, checks whether the extracted token may be used in common between two functional units to be connected, and defines the connection relationship between the functional units by using the names or identifiers of two functional units to be connected and the token when the extracted token may be used in common.
- [150] At this point, when the token may not be used in common between the two functional units, the decoding apparatus extracts another functional unit for replacing at least one of the two functional units, and checks whether the replaced functional unit and a functional unit to be connected with it may use a token in common. In this case, by repeating these operations until it is checked that the token may be used in common between the two functional units, the decoding apparatus can set the suitable connection relationship between functional units.
- [151] According to another exemplary embodiment, even in an encoding operation, verifying the suitability of the connection relationship between functional units can be easily performed.
- [152] The above-described decoding apparatus and method according to exemplary em-

bodiments may be implemented or executed through a program, and the program may be stored in a computer-readable storage medium.

- [153] The invention can also be embodied as computer readable codes on a computer-readable storage medium. The computer-readable storage medium is any data storage device that can store data which can be thereafter read by a computer system. Examples of the computer-readable storage medium include ROMs, RAMs, CD-ROMs, DVDs, magnetic tapes, floppy disks, registers, buffers, optical data storage devices, and carrier waves (such as data transmission through the Internet). The computer-readable storage medium can also be distributed over network coupled computer systems so that the computer readable codes are stored and executed in a distributed fashion. Also, functional programs, codes, and code segments for accomplishing the present invention can be easily construed by programmers skilled in the art to which the present invention pertains.

- [154] A number of exemplary embodiments have been described above. Nevertheless, it will be understood that various modifications may be made. For example, suitable results may be achieved if the described techniques are performed in a different order and/or if components in a described system, architecture, device, or circuit are combined in a different manner and/or replaced or supplemented by other components or their equivalents. Accordingly, other implementations are within the scope of the following claims.

[155]

### **Industrial Applicability**

- [156] A decoding apparatus and method and an encoding method according to exemplary embodiments may be applied to a Reconfigurable Video Coding (RVC) technology.

[157]

[158]

## Claims

- [Claim 1] A decoding apparatus, comprising:  
a repository storing a plurality of functional units in which an input or output port is defined as a token; and  
a configuration unit receiving decoder description which comprises an identifier of at least one of the functional units and connection relationship information which is composed of the token, selecting pertinent functional units from the repository, and connecting the selected functional units by using the token to configure a variable decoder,  
wherein the token corresponds to data which are inputted or outputted between the functional units, and is defined as a name or an identifier and the data, and  
a first functional unit of the functional units has the input port which is defined as a first token and presence of a storage functional for a first data corresponding to the first token.
- [Claim 2] The decoding apparatus of claim 1, wherein the first functional unit stores the current first data which is inputted to the input port, and operates by regarding that the stored first data is continuously or periodically inputted until a next first data is inputted.
- [Claim 3] The decoding apparatus of claim 2, wherein the first functional unit pre-stores a second data as a default value, and operates by regarding that the second data is continuously or periodically inputted until the first data is inputted.
- [Claim 4] The decoding apparatus of claim 1, wherein when a branch unit having an input port which is defined as a second token and at least two output ports which are defined as the second token is stored in the repository, and two or more second functional units in which an input port is defined as the second token are used to configure the variable decoder, the configuration unit selects the branch unit from the repository, and connects each output port of the branch unit and an input port of each of the second functional units by using the second token.
- [Claim 5] The decoding apparatus of claim 1, wherein:  
the decoder description further comprises a parser description defining a parsing unit, and  
the configuration unit generates the parsing unit by using the parser description, and connects the generated parsing unit and the selected

functional unit to configure the variable decoder.

[Claim 6]

The decoding apparatus of claim 1, wherein:

the decoder description further comprises parser description defining a parsing unit, and

the configuration unit selects a parsing unit corresponding to the parser description from the repository, and connects the selected parsing unit and the selected functional unit to configure the variable decoder.

[Claim 7]

The decoding apparatus of claim 1, wherein the configuration unit comprises a verification unit verifying connection relationships between the functional units.

[Claim 8]

A decoding method, comprising:

receiving an encoded bitstream and decoder description which comprises an identifier of at least one functional unit and a token corresponding to data which are inputted or outputted between the functional units;

selecting functional units corresponding to the identifier of the functional unit from a repository, and connecting the selected functional units by using the token to configure a variable decoder for decoding the encoded bitstream; and

decoding the encoded bitstream through the variable decoder, wherein the configuring of a variable decoder comprises selecting a first functional unit having an input port which is defined as a first token and presence of a storage function for a first data corresponding to the first token, and connecting the first functional unit with the selected functional unit by using the first token.

[Claim 9]

The decoding method of claim 8, wherein the decoding of the encoded bitstream comprises storing, by the first functional unit, the current first data which is inputted to the input port, and operating by regarding that the stored first data is continuously or periodically inputted until a next first data is inputted.

[Claim 10]

The decoding method of claim 9, wherein the decoding of the encoded bitstream comprises pre-storing, by the first functional unit, a second data as a default value, and operating by regarding that the second data is continuously or periodically inputted until the first data is inputted.

[Claim 11]

The decoding method of claim 8, wherein when a branch unit having an input port which is defined as a second token and at least two output ports which are defined as the second token is stored in the repository, and two or more second functional units in which an input port is

defined as the second token are used to configure the variable decoder, the configuring of a variable decoder further comprises selecting the branch unit from the repository, and connecting each output port of the branch unit and an input port of each of the second functional units by using the second token.

[Claim 12]

An encoding method, comprising:

encoding a bitstream;

extracting a functional unit for performing decoding, which corresponds to the encoding, from a functional unit table;

setting connection relationship between the extracted functional unit and a functional unit;

generating decoder description in which the set connection relationship is described; and

merging and transmitting the encoded bitstream and the decoder description,

wherein a name or identifier of the functional unit, an input port of the functional unit which is defined as a token and presence of a storage function for data corresponding to the token and an output port which is defined as the token are listed by functional unit in the functional unit table, and

the token corresponds to data which are inputted or outputted between the functional units, and is defined as a name or an identifier and the data.

[Claim 13]

The encoding method of claim 12, wherein the setting of connection relationship comprises:

extracting a token capable of an input or output of the extracted functional unit on the basis of a token pool;

checking whether the extracted token is usable in common between two functional units to be connected; and

defining connection relationship between the functional units by using names or identifiers of the two functional units to be connected, when the extracted token is usable in common.

[Claim 14]

The encoding method of claim 13, wherein the setting of connection relationship comprises:

extracting another functional unit for replacing at least one of the two functional units, when the token is not usable in common between two functional units; and

checking whether the replaced functional unit and a functional unit to

be connected with the replaced functional unit use in common the token,

wherein the extracting of another functional unit and the checking of common use of the token are repeated until the token is checked as being usable in common between the two functional units.

[Claim 15]

A computer-readable storage medium storing a program which connects a plurality of functional units, which are included to accord a serial or parallel transmission scheme and a size and transfer sequence of data which are transferred between the functional units, to configure a decoder, the program executing:

receiving decoder description which comprises an identifier of the functional unit and connection relation information which is composed of a token corresponding to data which are inputted or outputted between the functional units; and

selecting functional units corresponding to the identifier, and

connecting the selected functional units by using the token to configure the decoder, according to the decoder description,

wherein the configuring of the decoder comprises selecting a first token and a first functional unit having an input port which is defined as presence of a storage function for a first data corresponding to the first token, and connecting the first functional unit with the selected functional unit by using the first token.

[Claim 16]

The computer-readable storage medium of claim 15, wherein when a branch unit having an input port which is defined as a second token and at least two output ports which are defined as the second token exists, and two or more second functional units in which an input port is defined as the second token are used to configure the decoder, the configuring of the decoder further comprises selecting the branch unit from a repository, and connecting each output port of the branch unit and an input port of each of the second functional units by using the second token.

[Claim 17]

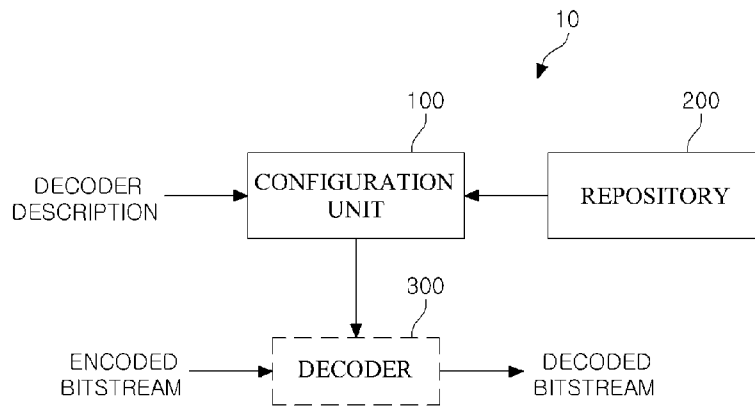
The computer-readable storage medium of claim 15, wherein the program further executes decoding an encoded bitstream through the decoder,

wherein the decoding of an encoded bitstream comprises storing, by the first functional unit, the current first data which is inputted to the input port, and operating by regarding that the stored first data is continuously or periodically inputted until a next first data is inputted.

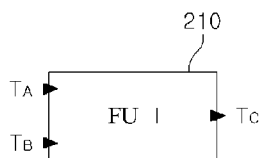
[Claim 18]

The computer-readable storage medium of claim 17, wherein the decoding of an encoded bitstream further comprises pre-storing, by the first functional unit, a second data as a default value, and operating by regarding that the second data is continuously or periodically inputted until the first data is inputted.

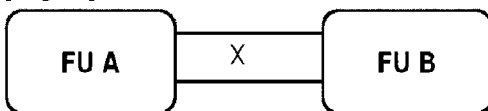
[Fig. 1]



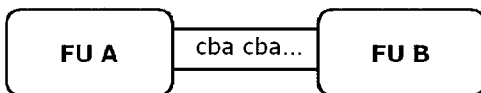
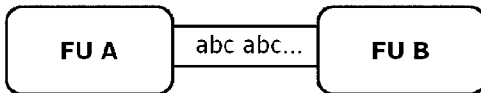
[Fig. 2]



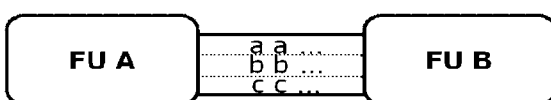
[Fig. 3]



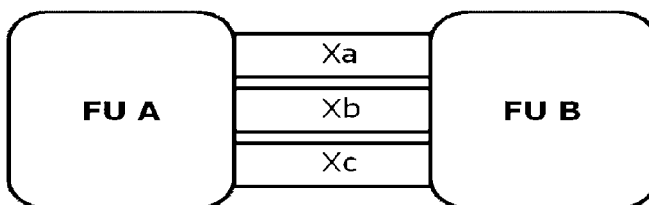
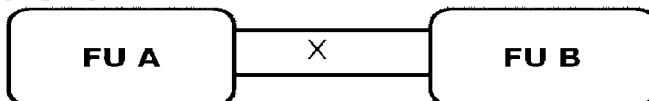
[Fig. 4]



[Fig. 5]

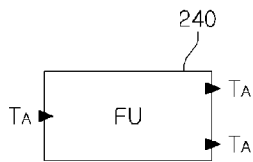


[Fig. 6]

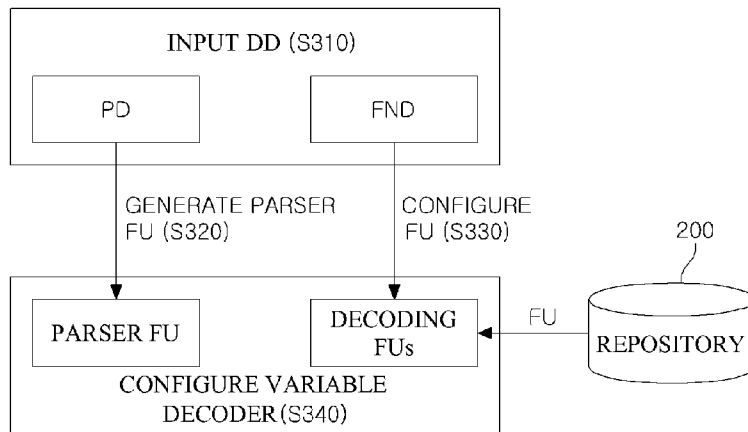




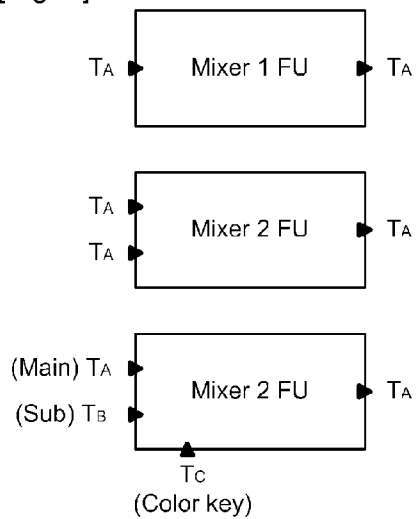
[Fig. 7]



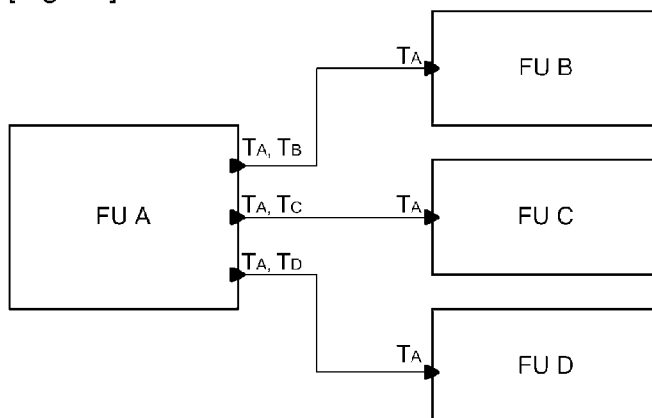
[Fig. 8]



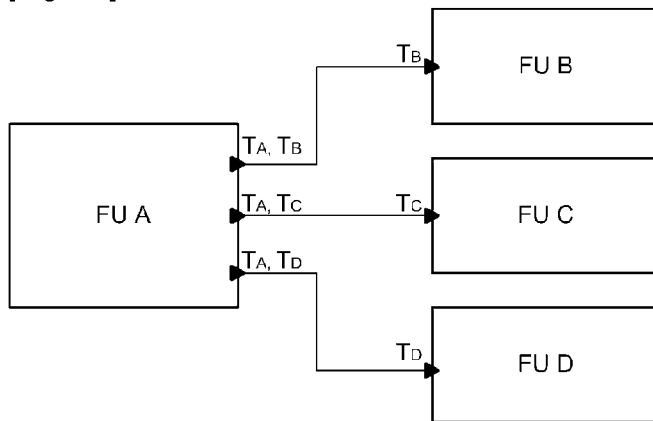
[Fig. 9]



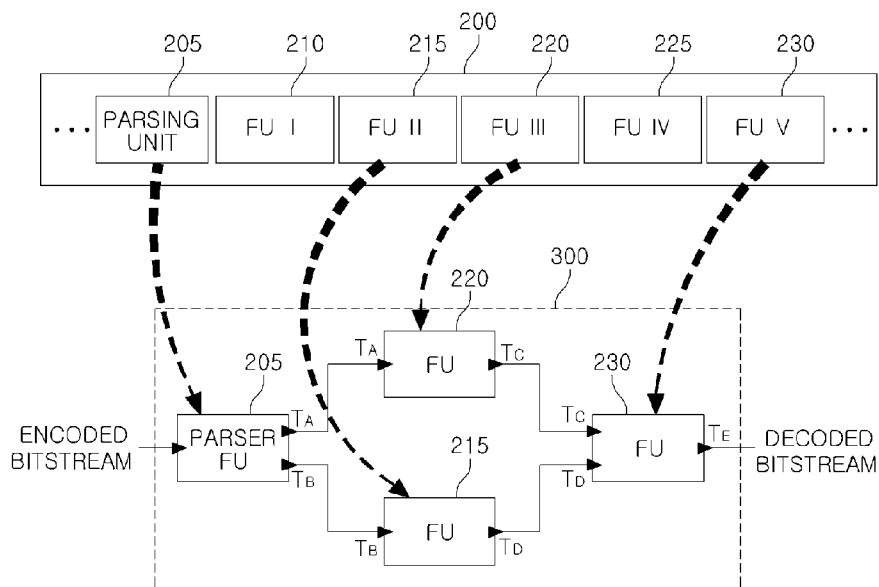
[Fig. 10]



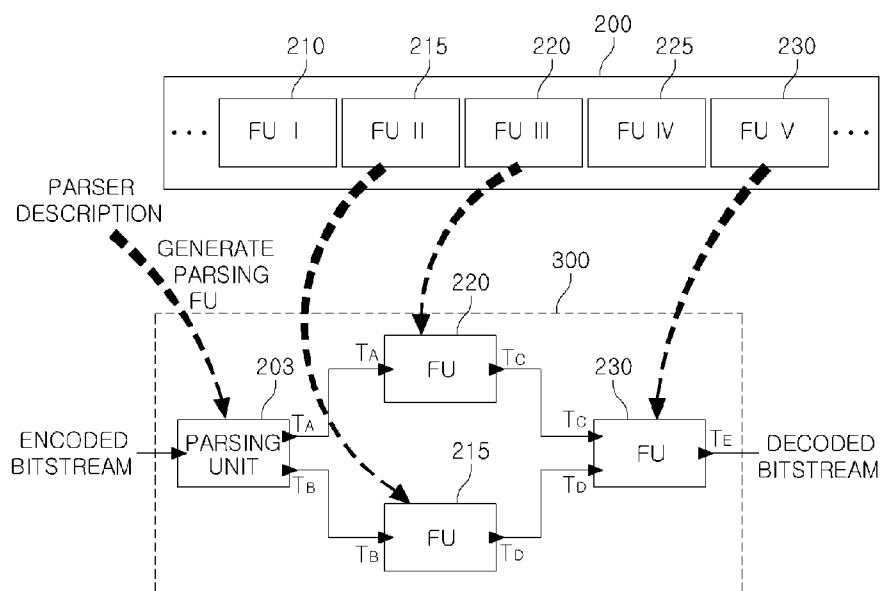
[Fig. 11]



[Fig. 12]



[Fig. 13]



[Fig. 14]

