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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 September 2017 (08.09.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/151175 A1

(51) International Patent Classification:
H03M 13/29 (2006.01) *H03M 13/39* (2006.01)

(21) International Application Number:
PCT/US2016/039077

(22) International Filing Date:
23 June 2016 (23.06.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/302,038 1 March 2016 (01.03.2016) US

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventors: **DAVYDOV, Alexei**; Lenin Ave 28/11-40, Nizhny Novgorod, NIZ, 603132 (RU). **CHATTERJEE, Debdeep**; 725 Mariposa Ave, Apt 301, Mountainn View, California 94041 (US). **ERMOLAEV, Grigory**; Monchegorskaya str. 7a, Nizhny Novgorod, NIZ, 603000 (RU).

(74) Agent: **LEDELL, Brian**; Ledell Ansari, LLP, c/o CPA Global, 900 Second Avenue South, Suite 600, Minneapolis, Minnesota 55402 (US).

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

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

[Continued on next page]

(54) Title: TURBO CODING WITH SUPPORT FOR LOW CODING RATES

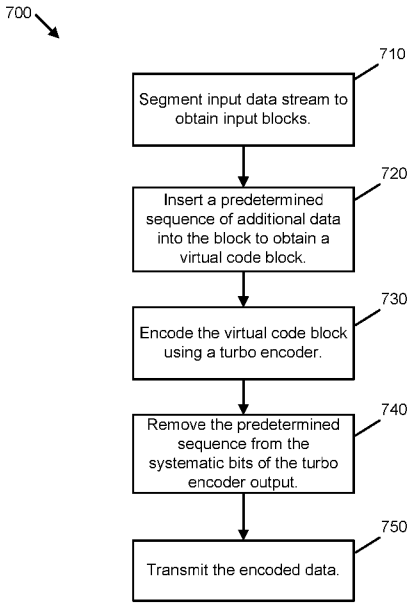


Fig. 7

(57) Abstract: A turbo encoder designed to operate at one coding rate may be operated at a lower coding rate by modifying the size of each input block of data. In one implementation, the size may be increased by inserting predetermined bit sequences (i.e., additional bits) into the original block of data to create a "virtual code block" that is larger than the input block of data. The virtual code block may be input to the turbo coder to obtain an encoded output block that includes a systematic portion and parity portion. The systematic portion may be modified to remove the predetermined bit sequence and the resulting output block transmitted over a channel (e.g., a noisy radio channel).

WO 2017/151175 A1

Published:

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

TURBO CODING WITH SUPPORT FOR LOW CODING RATES

RELATED APPLICATIONS

The present application claims the benefit of U.S. Provisional Patent Application No. 62/302,038, which was filed on March 1, 2016, the contents of which are hereby incorporated by reference as though fully set forth herein.

BACKGROUND

Turbo codes are a class of high-performance forward error correction (FEC) codes that are used to control errors in data transmission over unreliable or noisy communication channels. A central idea of FEC codes is that the sender encodes a message in a redundant way using an error correcting coding technique (e.g., a turbo code). The redundancy allows the receiver to detect, and potentially to correct, a limited number of errors that may occur in the message. FEC codes give the receiver the opportunity to correct errors without needing a reverse channel to request the retransmission of data.

The amount of redundancy in the encoded data is known as the "code rate" or "information rate," which may be defined as the proportion of the encoded data that is useful (non-redundant) data. That is, if the code rate is 1/3, for every one bit of useful information, the encoder may generate three total bits of data, of which two bits are redundant.

One application of turbo codes is in the radio link of cellular communication networks. For example, in Third Generation Partnership Project (3GPP) standardized networks, turbo codes may be used. In certain applications, such as Narrowband (NB) Internet-of-Things (IoT) applications, it may be desirable to use turbo codes that use a low coding rate, such as 1/3. In the 3GPP Long Term Evolution (LTE)-Advanced (LTE-A) architecture, for example, turbo coders that operate at a coding rate of 1/3 may be used. In some applications, however, it may be desirable to use coding rates even lower than 1/3.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments described herein will be readily understood by the following detailed description in conjunction with the accompanying drawings. To facilitate this description, like reference numerals may designate like structural elements. Embodiments are illustrated by way of example and not by way of limitation in the figures of the accompanying drawings.

Figs. 1A and 1B are block diagrams illustrating example implementations of turbo coders;

Fig. 2 is a block diagram conceptually illustrating an example of rate matching of encoded data streams from a turbo coder;

Fig. 3 is a diagram illustrating a turbo coded sequence;

Fig. 4 is a diagram illustrating a system, consistent with aspects described herein, for performing turbo coding to generate low coding rate;

Fig. 5 is a diagram graphically illustrating an example of turbo coding of an input data block;

Fig. 6 is a diagram graphically illustrating an example of decoding of the encoded block from Fig. 5;

Fig. 7 is a flowchart illustrating an example process for encoding a data stream at a transmitting device;

Fig. 8 is a flowchart illustrating an example process for decoding a data stream a receiving device;

Figs. 9A-9D are diagrams illustrating example embodiments for adding additional data, to an input data block, to obtain a virtual code block; and

Fig. 10 is a block diagram example components of another electronic device.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The following detailed description refers to the accompanying drawings. The same reference numbers in different drawings may identify the same or similar elements. It is to be understood that other embodiments may be utilized and structural or logical changes may be made without departing from the scope of the present disclosure. Therefore, the following detailed description is not to be taken in a limiting sense, and the scope of embodiments is defined by the appended claims and their equivalents.

Techniques described herein relate to the supporting of low coding rates, such as a coding rate of 1/6, using a turbo coding component that is designed to operate at a higher coding rate, such as 1/3. The techniques described herein may be particularly useful for low bandwidth applications that communicate over a noisy channel. For example, the techniques described herein may be useful for NB IoT devices that communicate using a wireless cellular network.

The techniques described herein may improve the performance, at low coding rates, using turbo coding. Relative to existing techniques for implementing a low coding rate turbo code, the techniques described herein may be more efficient. The more efficient coding may result in lower radio uplink transmission times, which may thus lead to less power consumption and increased battery life of IoT/Machine-Type-Communication (MTC) devices. The performance improvement provided by the low rate channel coding can be particularly useful in

certain network modes, such as the 3GPP extended coverage mode. In some implementations, the techniques described herein may also be beneficial in downlink transmissions (e.g., to increase spectral efficiency).

As described herein, an input block of data, that is to be encoded, may be modified to increase the size of the block of data. The increased size may be performed by inserting predetermined bit sequences (i.e., additional bits) into the original block of data to create a "virtual code block" that is larger than the input block of data. The virtual code block may be input to a turbo coder to obtain an encoded output block. In turbo coding, the output block may include three portions, known as the systematic portion, a first parity portion, and a second parity portion. The systematic portion may be modified to remove the predetermined bit sequence, and the resulting output block may be transmitted over a channel (e.g., a noisy radio channel).

At the receiver, the systematic portion of the received data may again be modified to reinsert the predetermined bit sequence. The encoded data, with the reinserted predetermined bit sequence, may be decoded to obtain the original block of data.

Fig. 1A is a block diagram illustrating an example implementation of a turbo coder 100. Turbo coder 100 may receive input bits ("Input Bits") that correspond to the data that is to be encoded. The input bits may be received as blocks having predetermined lengths. That is, a data stream may be broken up into fixed length blocks. Turbo coder 100 may have three outputs: a systematic output and two parity outputs.

As shown, turbo coder 100 includes two component encoders 110 ("Systematic Convolution Encoder 1") and 120 ("Systematic Convolution Encoder 2") that are separated by an interleaver 130. As shown, only one of the systematic outputs from the two component encoders may be used (i.e., a "Systematic Bits" output is not shown for Systematic Convolution Encoder 2). As shown in Fig. 1A, the output of turbo encoder 100 may thus include three sections: Systematic Bits, Parity Bits 1, and Parity Bits 2. These three sections may be serialized and eventually transmitted over the channel (e.g., a noisy radio channel). The coding rate of turbo coder 100 may be 1/3.

Fig. 1B is a block diagram illustrating one particular embodiment of turbo coder 100. As shown in Fig. 1B, turbo coder 100 may be a systematic convolutional turbo coder, such as a turbo coder described in section 5.1.3.2.1 of 3GPP Technical Specification (TS) 36.212 v. 12.5.0. Turbo coder 100, as shown in Fig. 1B, may be a systematic parallel concatenated convolutional coder with two 8-state constituent encoders and one turbo code internal interleaver. Each constituent encoder is independently terminated by tail bits. The dotted

lines (line 125) in Fig. 1B may indicate trellis termination information that may not be normally output from turbo coder 100. In turbo coder 100, as shown in Fig. 1B, c_k may correspond to the input bits (as shown in Fig. 1A), and may represent an input block of length k. x_k may correspond to the Systematic Bits (as shown in Fig. 1A), of length k bits. Similarly, z_k and $z_{k'}$ may correspond to Parity Bits 1 and Parity Bits 2 (as shown in Fig. 1A), respectively, each of length k bits. $x_{k'}$ may correspond to the trellis termination information. Each of encoders 110 and 120 may include a number of delay elements ("D") and adders ("+").

The output of turbo coder 100 may correspond to the systematic bits and two sets of parity bits, as well as 12 tail bits due to trellis information. Post-encoding rate matching may be performed on the output bits, on a per code block basis, and may include interleaving the three information bit streams (i.e., Systematic Bits, Parity Bits 1, and Parity Bits 2) followed by collection of the bit streams and output using a circular buffer.

Fig. 2 is a block diagram conceptually illustrating an example of rate matching of the encoded data streams. As shown, turbo encoder 100 may produce the three output data streams: Systematic Bits, Parity Bits 1, and Parity Bits 2. The parity bits may be processed by interleaver 210. Interleaver 210 may operate to perform interleaving and multiplexing operations. The interleaving and multiplexing operations may, for example, include sub-block interleaving as described in in section 5.1.4.2.1 of 3GPP TS 36.212 v. 12.5.0. The Systematic Bits and the interleaved parity bits may be input to bit collection and selection component 220, which may operate to buffer and output a single encoded and serialized output stream.

In one implementation, bit collection and selection component 220 may implement a circular buffer. The circular buffer may be implemented as described in in section 5.1.4.2.2 of 3GPP TS 36.212 v. 12.5.0. Input streams corresponding to the systematic bits, parity bits 1, and parity bits 2 may be received and stored in pre-designated locations (e.g., as described in section 5.1.4.2.2 of 3GPP TS 36.212 v. 12.5.0) of the circular buffer. When reading the output sequence, for transmission over the channel, the circular buffer may be sequentially read, and potentially "re-read" by wrapping around from the end of the buffer to the beginning of the circular buffer.

To achieve a coding rate lower than 1/3 using turbo coder 100, existing techniques may read the circular buffer in a wrap-around manner to thereby repeat the encoded bits in the output bitstream that is transmitted over the transport channel. For example, to achieve a coding rate of 1/6, the circular buffer may be read twice.

Fig. 3 is a diagram illustrating a turbo coded sequence, using turbo encoder 100, in which a coding rate of 1/6 may obtained by reading two iterations from bit collection and selection component 220 (e.g., by iterating twice through the circular buffer). As shown, block 305 may represent the input data block. Block 310 may represent the encoded output stream with coding rate equal to 1/3. That is, block 310 may represent the output of turbo encoder 100. Bit stream 320 may represent the turbo coded output stream, after repetition based on wrap-around of the circular buffer implemented by bit collection and selection component 220. In this case, the coding rate may be decreased to 1/6. However, the repetition-based encoding, as shown in Fig. 3, to decrease the coding rate, can be relatively inefficient. In particular, it would be desirable to use a coding scheme that requires fewer bits to achieve the same coding rate.

Fig. 4 is a diagram illustrating a system 400, consistent with aspects described herein, for performing turbo coding to generate a lower coding rate (e.g., 1/6). The lower coding rate may be obtained based on generation of a virtual input code block that is larger than the actual input code block. The encoding technique discussed herein may be more efficient than the use of repetition encoding, as described with respect to Fig. 3, to obtain a lower coding rate.

As shown in Fig. 4, system 400 may include a coding section 410 and a decoding section 450. Coding and decoding sections 410 and 450, respectively, may correspond to software and/or hardware implemented in User Equipment (UE) and/or base stations (e.g., evolved NodeBs) in a cellular network. The UE and base station may communicate over a noisy channel (e.g., via a radio link). In other implementations, coding section 410 and decoding section 450 may be implemented in other environments, such as between any two devices that communicate via a noisy channel, including radio and non-radio links. Additionally, although coding section 410 may be described as being implemented by a UE and decoding section 450 may be primarily described herein as being implemented by a base station, in some implementations, the UE may implement decoding section 450 and the base station may implement coding section 410. Still further, in some implementations, both communication devices (e.g., both the UE and base station) may each implement both of coding section 410 and decoding section 450.

As shown in Fig. 4, coding section 410 may include block expansion component 415, encoder 420, encoded block compression component 425, and bit collection and selection component 430. Decoding section 450 may include block expansion component 455 and decoder 460. Channel 470, such as a wireless (e.g., radio) link, is also illustrated in Fig.

4. As previously mentioned, FEC codes, such as turbo codes, may be particularly useful in the context of lossy (e.g., noisy) communication mediums.

Block expansion component 415 may operate to expand the input data block (i.e., a portion of an input data stream that is to be transmitted over channel 470). The expanded version of the input data block, which may be called a "virtual input code block" herein, may be larger than the original input block. For example, the virtual input block size can be selected as:

$$b = a * (1/R - 1) /2 \qquad \text{(equation 1)};$$

where R is the desired coding rate, a is the size (length) of the original input data block, and b is the size of the virtual input code block. Thus, the additional number of bits to add to the input data block, to obtain the virtual input code block, is of length b-a. In one implementation, the additional bits to add to the input data block, and the spacing of the additional bits relative to the original input data block, may be deterministically determined in a manner that is known by both coding section 410 and decoding section 450. For example, the additional bits may be added to the beginning of the input data block, added to the end of the input data block, or interleaved within the input data block. Additional potential embodiments for combining the additional bits to the input data block, to obtain the virtual input code block, are described below with reference to Figs. 9A-9D.

Referring back to Fig. 4, encoder 420 may include a turbo coder, such as a turbo coder implemented as described above with respect to turbo coder 100. The output of encoder 420 may thus include systematic bits and two sets of parity bits. Encoded block compression component 425 may receive the output of encoder 420 (e.g., the systematic bits, parity bits 1, and parity bits 2), and may operate on the systematic bits to remove the additional bits from the systematic bits. Alternatively, encoded block compression component 425 may replace the systematic bits with the input data block. In either situation, the resultant systematic bits may be identical the systematic bits that would be obtained if block expansion were not used. The length of parity bits 1 and parity bits 2, however, may be longer than if block expansion were not used.

Bit collection and selection component 430 may operate similarly to bit collection and selection component 220. However, repetitive reading of the circular buffer, which may be used in conventional systems in order to obtain a lower coding rate, is not needed in the implementation illustrated in Fig. 4.

Decoding section 450 may include block expansion component 455 and decoder 460. Block expansion component 455 may operate to identify the systematic data of the

received input block and to re-insert the additional bits. Block expansion component 455 may thus operate similarly to block expansion component 415. Decoder 460 may perform a turbo coding decoding operation that complements the encoding performed by encoder 420. The output of decoder 460, during a successful operation, will thus be the original input block.

Fig. 5 is a diagram graphically illustrating an example of turbo coding of an input data block by coding section 410. Input data block 510, of length *a* bits (e.g., four bits), may be input to block expansion component 415. After operation of block expansion component 415, virtual input code block 520, of length *b* bits, may be obtained. Virtual input code block 520 may include the bits from input code block 510 and may also include the additional bits. In this example, the additional bits are added in a predetermined sequence in which two bits are added after each bit in original data block 510. The additional bits may be, for example, all "zero" bits, all "one" bits, or a known combination of zero and one bits (e.g., alternating zero and one bits).

Virtual input code block 520 may be processed by coder 420 to obtain coded block 530. As previously mentioned, the processing by coder 420 may include turbo coding using systematic convolution encoders. Coded block 530, as illustrated, may include systematic bits and two sets of parity bits. The systematic bits may be unchanged from virtual input code block 520. Coded block 530 may be processed by encoded block compression component 425 to remove the additional bits from the systematic bits, and to thus obtain the original input data block. The final coded sequence, potentially after processing by bit collection and selection component 430, is shown as coded block 540. For the situation in which encoder 420 implements a turbo coder with a coding rate of 1/3, the final coding rate of coded block 550 may be lower, such as 1/6. The lower coding rate may be achieved without performing output repetition using wrap around reading of the circular buffer.

Fig. 6 is a diagram graphically illustrating an example of decoding of the coded block from Fig. 5. The decoding may be performed, by decoding section 450, after reception of the coded block over the channel. As shown, the received block 610 may be obtained at decoding section 450. Block expansion component 455 may identify the systematic bits of received bit sequence 610 (at 620) and add the additional bits back to the systematic bits using the same pattern that was used by block expansion component 415, to obtain block 630. Block 630 may then be decoded by decoder 460 to obtain the substantive (output) data block 640. If the coding/decoding operation was successful, data block 640 will match input data block 510.

Fig. 7 is a flowchart illustrating an example process 700 for encoding and transmitting a data stream at a transmitting device. Process 700 may be performed by, for example, a UE or a base station.

To begin, process 700 may include segmenting an input data stream (e.g., the data to be transmitted over a noisy channel) to obtain an input block (block 710). Each input block may be of a fixed length.

Process 700 may further include inserting a predetermined sequence of additional data into the block to obtain a virtual code block (block 720). As previously mentioned, the predetermined sequence of additional data may be added to the beginning of the input data block, added to the end of the input data block, or interleaved within the input data block. The amount of additional data (i.e., the length of the predetermined sequence) relative to the size of the input data block, may determine the coding rate of the encoded data. In one implementation, the relationship between the length of the input data block, the coding rate, and the amount of additional data may be defined by equation (1).

Process 700 may further include encoding the virtual code block using a turbo encoder (block 730). For example, the virtual code block may be input to turbo encoder 100, which may output systematic bits, and first and second parity bits. The output of turbo encoder 100 may be referred to as an encoded code block.

Process 700 may further include removing the predetermined sequence of additional data from the systematic bits of the output of the turbo encoder (block 740). "Removing," as used herein, may also refer to replacing the systematic bits with the original input block or otherwise using the systematic bits that do not include the predetermined sequence of additional data. The first and second parity bits may, however, be based on the virtual code block. The systematic bits, with the predetermined sequence of additional data removed, along with the first and second parity bits, correspond to the output data that is to be transmitted over the channel.

Process 700 may further include transmitting the encoded data (block 750). The encoded data may be transmitted over the noisy channel, such as a radio link.

Fig. 8 is a flowchart illustrating an example process 800 for decoding a data stream at a receiving device. Process 800 may be performed by, for example, a UE or a base station.

Process 800 may include receiving/processing an encoded block (block 810). Process 800 may further include adding the predetermined sequence to the encoded block (block 820). In one implementation, the addition of the predetermined sequence to the

encoded block may include identifying the systematic bits and adding the predetermined sequence, to the systematic bits, in the same manner that the predetermined sequence was added at the transmitting device.

Process 800 may further include decoding the data block (block 830). The decoding process may be based on the use of a turbo coding decode process, such as the decode process for turbo coder 100.

Figs. 9A-9D are diagrams illustrating example embodiments for adding additional data, to an input data block, to obtain a virtual code block. In Figs. 9A-9D, the values for the additional data may be set to any particular value, pattern of values, or pseudorandom deterministic sequence. For example, the additional data may be added as all one bits, all zero bits, a predetermined pattern, a predetermined recurring pattern, and/or based on a pseudorandom sequence.

As shown in Fig. 9A, the additional bits may be added to the input data block, to obtain virtual code block 910, in a manner in which the additional bits are equally spaced with respect to the bits of the input data block. In the example shown, two additional bits are added after each bit of the input data block.

As shown in Fig. 9B, the additional bits may be inserted at the beginning of the data block to obtain virtual code block 920. Similarly, as shown in Fig. 9C, the additional bits may be inserted at the end of the data block to obtain virtual code block 930.

As shown in Fig. 9D, the additional bits may be inserted in a pseudorandom pattern or based on a predetermined insertion pattern. For example, as shown in the example of Fig. 9D, two additional bits are followed by the first bit of the input data block, which is followed by one additional bit, which is followed by the second bit of the input data block, which is followed by three additional bits, which is followed by the third bit of data block, etc. In one implementation, a pseudorandom insertion pattern may be used in which the seed or initial value for the pseudorandom pattern may be based a value such as a value based on a current index of the frame or subframe.

Embodiments described herein may be implemented into a system using any suitably configured hardware and/or software. Fig. 10 illustrates, for one embodiment, example components of an electronic device 1000. In embodiments, the electronic device 1000 may be a mobile device or a RAN node (e.g., an eNodeB). In some embodiments, the electronic device 1000 may include application circuitry 1002, baseband circuitry 1004, Radio Frequency (RF) circuitry 1006, front-end module (FEM) circuitry 1008 and one or more antennas 1060, coupled together at least as shown. In other embodiments, any of said circuitries can be included in

different devices. In one implementation, the implementations described previously may be implemented by baseband circuitry 1004.

Application circuitry 1002 may include one or more application processors. For example, the application circuitry 1002 may include circuitry such as, but not limited to, one or more single-core or multi-core processors. The processor(s) may include any combination of general-purpose processors and dedicated processors (e.g., graphics processors, application processors, etc.). The processors may be coupled with and/or may include memory/storage and may be configured to execute instructions stored in the memory/storage to enable various applications and/or operating systems to run on the system. The memory/storage may include, for example, computer-readable medium 1003, which may be a non-transitory computer-readable medium. Application circuitry 1002 may, in some embodiments, connect to or include one or more sensors, such as environmental sensors, cameras, etc.

Baseband circuitry 1004 may include circuitry such as, but not limited to, one or more single-core or multi-core processors. The baseband circuitry 1004 may include one or more baseband processors and/or control logic to process baseband signals received from a receive signal path of the RF circuitry 1006 and to generate baseband signals for a transmit signal path of the RF circuitry 1006. Baseband processing circuitry 1004 may interface with the application circuitry 1002 for generation and processing of the baseband signals and for controlling operations of the RF circuitry 1006. For example, in some embodiments, the baseband circuitry 1004 may include a second generation (2G) baseband processor 1004a, third generation (3G) baseband processor 1004b, fourth generation (4G) baseband processor 1004c, and/or other baseband processor(s) 1004d for other existing generations, generations in development or to be developed in the future (e.g., fifth generation (5G), 10G, etc.). The baseband circuitry 1004 (e.g., one or more of baseband processors 1004a-d) may handle various radio control functions that enable communication with one or more radio networks via the RF circuitry 1006. The radio control functions may include, but are not limited to, signal modulation/demodulation, encoding/decoding, radio frequency shifting, etc. In some implementations, the functionality of baseband circuitry 904 may be wholly or partially implemented by memory/storage devices configured to execute instructions stored in the memory/storage. The memory/storage may include, for example, a non-transitory computer-readable medium 1004h.

In some embodiments, modulation/demodulation circuitry of the baseband circuitry 1004 may include Fast-Fourier Transform (FFT), precoding, and/or constellation mapping/demapping functionality. In some embodiments, encoding/decoding circuitry of the baseband circuitry 1004 may include convolution, tail-biting convolution, turbo, Viterbi, and/or

Low Density Parity Check (LDPC) encoder/decoder functionality. Embodiments of modulation/demodulation and encoder/decoder functionality are not limited to these examples and may include other suitable functionality in other embodiments. In some embodiments, the baseband circuitry 1004 may include elements of a protocol stack such as, for example, elements of an evolved universal terrestrial radio access network (EUTRAN) protocol including, for example, physical (PHY), media access control (MAC), radio link control (RLC), packet data convergence protocol (PDCP), and/or radio resource control (RRC) elements. A central processing unit (CPU) 1004e of the baseband circuitry 1004 may be configured to run elements of the protocol stack for signaling of the PHY, MAC, RLC, PDCP and/or RRC layers. In some embodiments, the baseband circuitry may include one or more audio digital signal processor(s) (DSP) 1004f. The audio DSP(s) 1004f may be include elements for compression/decompression and echo cancellation and may include other suitable processing elements in other embodiments.

In some embodiments, the baseband circuitry 1004 may include elements of a protocol stack such as, for example, elements of an evolved universal terrestrial radio access network (EUTRAN) protocol including, for example, physical (PHY), media access control (MAC), radio link control (RLC), packet data convergence protocol (PDCP), and/or radio resource control (RRC) elements. A central processing unit (CPU) 1004e of the baseband circuitry 1004 may be configured to run elements of the protocol stack for signaling of the PHY, MAC, RLC, PDCP and/or RRC layers. In some embodiments, the baseband circuitry may include one or more audio digital signal processor(s) (DSP) 1004f. The audio DSP(s) 104f may be include elements for compression/decompression and echo cancellation and may include other suitable processing elements in other embodiments.

Baseband circuitry 1004 may further include memory/storage 1004g. The memory/storage 1004g may be used to load and store data and/or instructions for operations performed by the processors of the baseband circuitry 1004. Memory/storage 1004g may particularly include a non-transitory memory. Memory/storage for one embodiment may include any combination of suitable volatile memory and/or non-volatile memory. The memory/storage 1004g may include any combination of various levels of memory/storage including, but not limited to, read-only memory (ROM) having embedded software instructions (e.g., firmware), random access memory (e.g., dynamic random access memory (DRAM)), cache, buffers, etc. The memory/storage 1004g may be shared among the various processors or dedicated to particular processors.

Components of the baseband circuitry may be suitably combined in a single chip, a single chipset, or disposed on a same circuit board in some embodiments. In some

embodiments, some or all of the constituent components of the baseband circuitry 1004 and the application circuitry 1002 may be implemented together such as, for example, on a system on a chip (SOC).

In some embodiments, the baseband circuitry 1004 may provide for communication compatible with one or more radio technologies. For example, in some embodiments, the baseband circuitry 1004 may support communication with an evolved universal terrestrial radio access network (EUTRAN) and/or other wireless metropolitan area networks (WMAN), a wireless local area network (WLAN), a wireless personal area network (WPAN). Embodiments in which the baseband circuitry 1004 is configured to support radio communications of more than one wireless protocol may be referred to as multi-mode baseband circuitry.

RF circuitry 1006 may enable communication with wireless networks using modulated electromagnetic radiation through a non-solid medium. In various embodiments, the RF circuitry 1006 may include switches, filters, amplifiers, etc. to facilitate the communication with the wireless network. RF circuitry 1006 may include a receive signal path which may include circuitry to down-convert RF signals received from the FEM circuitry 1008 and provide baseband signals to the baseband circuitry 1004. RF circuitry 1006 may also include a transmit signal path which may include circuitry to up-convert baseband signals provided by the baseband circuitry 1004 and provide RF output signals to the FEM circuitry 1008 for transmission.

In some embodiments, the RF circuitry 1006 may include a receive signal path and a transmit signal path. The receive signal path of the RF circuitry 1006 may include mixer circuitry 1006a, amplifier circuitry 1006b and filter circuitry 1006c. The transmit signal path of the RF circuitry 1006 may include filter circuitry 1006c and mixer circuitry 1006a. RF circuitry 1006 may also include synthesizer circuitry 1006d for synthesizing a frequency for use by the mixer circuitry 1006a of the receive signal path and the transmit signal path. In some embodiments, the mixer circuitry 1006a of the receive signal path may be configured to down-convert RF signals received from the FEM circuitry 1008 based on the synthesized frequency provided by synthesizer circuitry 1006d. The amplifier circuitry 1006b may be configured to amplify the down-converted signals and the filter circuitry 1006c may be a low-pass filter (LPF) or band-pass filter (BPF) configured to remove unwanted signals from the down-converted signals to generate output baseband signals.

Output baseband signals may be provided to the baseband circuitry 1004 for further processing. In some embodiments, the output baseband signals may be zero-frequency baseband signals, although this is not a requirement. In some embodiments, mixer circuitry

1006a of the receive signal path may comprise passive mixers, although the scope of the embodiments is not limited in this respect.

In some embodiments, the mixer circuitry 1006a of the transmit signal path may be configured to up-convert input baseband signals based on the synthesized frequency provided by the synthesizer circuitry 1006d to generate RF output signals for the FEM circuitry 1008. The baseband signals may be provided by the baseband circuitry 1004 and may be filtered by filter circuitry 1006c. The filter circuitry 1006c may include a low-pass filter (LPF), although the scope of the embodiments is not limited in this respect.

In some embodiments, the mixer circuitry 1006a of the receive signal path and the mixer circuitry 1006a of the transmit signal path may include two or more mixers and may be arranged for quadrature downconversion and/or upconversion respectively. In some embodiments, the mixer circuitry 1006a of the receive signal path and the mixer circuitry 1006a of the transmit signal path may include two or more mixers and may be arranged for image rejection (e.g., Hartley image rejection). In some embodiments, the mixer circuitry 1006a of the receive signal path and the mixer circuitry 1006a may be arranged for direct downconversion and/or direct upconversion, respectively. In some embodiments, the mixer circuitry 1006a of the receive signal path and the mixer circuitry 1006a of the transmit signal path may be configured for super-heterodyne operation.

In some embodiments, the output baseband signals and the input baseband signals may be analog baseband signals, although the scope of the embodiments is not limited in this respect. In some alternate embodiments, the output baseband signals and the input baseband signals may be digital baseband signals. In these alternate embodiments, the RF circuitry 1006 may include analog-to-digital converter (ADC) and digital-to-analog converter (DAC) circuitry and the baseband circuitry 1004 may include a digital baseband interface to communicate with the RF circuitry 1006.

In some dual-mode embodiments, a separate radio IC circuitry may be provided for processing signals for each spectrum, although the scope of the embodiments is not limited in this respect.

In some embodiments, the synthesizer circuitry 1006d may be a fractional-N synthesizer or a fractional N/N+6 synthesizer, although the scope of the embodiments is not limited in this respect as other types of frequency synthesizers may be suitable. For example, synthesizer circuitry 1006d may be a delta-sigma synthesizer, a frequency multiplier, or a synthesizer comprising a phase-locked loop with a frequency divider.

The synthesizer circuitry 1006d may be configured to synthesize an output frequency for use by the mixer circuitry 1006a of the RF circuitry 1006 based on a frequency input and a divider control input. In some embodiments, the synthesizer circuitry 1006d may be a fractional N/N+6 synthesizer.

In some embodiments, frequency input may be provided by a voltage-controlled oscillator (VCO), although that is not a requirement. Divider control input may be provided by either the baseband circuitry 1004 or the applications processor 1002 depending on the desired output frequency. In some embodiments, a divider control input (e.g., N) may be determined from a look-up table based on a channel indicated by the applications processor 1002.

Synthesizer circuitry 1006d of the RF circuitry 1006 may include a divider, a delay-locked loop (DLL), a multiplexer and a phase accumulator. In some embodiments, the divider may be a dual modulus divider (DMD) and the phase accumulator may be a digital phase accumulator (DPA). In some embodiments, the DMD may be configured to divide the input signal by either N or N+6 (e.g., based on a carry out) to provide a fractional division ratio. In some example embodiments, the DLL may include a set of cascaded, tunable, delay elements, a phase detector, a charge pump and a D-type flip-flop. In these embodiments, the delay elements may be configured to break a VCO period up into Nd equal packets of phase, where Nd is the number of delay elements in the delay line. In this way, the DLL provides negative feedback to help ensure that the total delay through the delay line is one VCO cycle.

In some embodiments, synthesizer circuitry 1006d may be configured to generate a carrier frequency as the output frequency, while in other embodiments, the output frequency may be a multiple of the carrier frequency (e.g., twice the carrier frequency, four times the carrier frequency) and used in conjunction with quadrature generator and divider circuitry to generate multiple signals at the carrier frequency with multiple different phases with respect to each other. In some embodiments, the output frequency may be a LO frequency (fLO). In some embodiments, the RF circuitry 1006 may include an IQ/polar converter.

FEM circuitry 1008 may include a receive signal path which may include circuitry configured to operate on RF signals received from one or more antennas 1060, amplify the received signals and provide the amplified versions of the received signals to the RF circuitry 1006 for further processing. FEM circuitry 1008 may also include a transmit signal path which may include circuitry configured to amplify signals for transmission provided by the RF circuitry 1006 for transmission by one or more of the one or more antennas 1060.

In some embodiments, the FEM circuitry 1008 may include a TX/RX switch to switch between transmit mode and receive mode operation. The FEM circuitry may include a receive

signal path and a transmit signal path. The receive signal path of the FEM circuitry may include a low-noise amplifier (LNA) to amplify received RF signals and provide the amplified received RF signals as an output (e.g., to the RF circuitry 1006). The transmit signal path of the FEM circuitry 1008 may include a power amplifier (PA) to amplify input RF signals (e.g., provided by RF circuitry 1006), and one or more filters to generate RF signals for subsequent transmission (e.g., by one or more of the one or more antennas 1060).

In some embodiments, the electronic device 1000 may include additional elements such as, for example, memory/storage, display, camera, sensors, and/or input/output (I/O) interface. In some embodiments, the electronic device of Fig. 10 may be configured to perform one or more methods, processes, and/or techniques such as those described herein.

A number of examples, relating to implementations of the techniques described above, will next be given.

In a first example, an apparatus for a baseband processor of User Equipment (UE) may include circuitry to: process an input data stream to obtain an input block of a particular size; insert additional data into the input block to obtain a virtual code block having a size larger than the particular size of the input block; encode the virtual code block, using a turbo coding technique, to obtain an encoded code block, the encoded code block including a systematic portion and a parity portion; remove the additional data from the systematic portion of the encoded code block to obtain an output data block that includes the systematic portion with the additional data removed and the parity portion; and output the output data block for processing and transmission over a channel.

In example 2, the subject matter of example 1, wherein the circuitry is additionally to: generate additional data as a sequence of either all zero or a sequence of all one bits.

In example 3, the subject matter of example 1, or any of the examples herein, wherein the circuitry is additionally to: generate the additional data as a pseudorandom sequence of bits.

In example 4, the subject matter of examples 1, 2 or 3, or any of the examples herein, wherein inserting the additional data into the input block includes inserting the additional data at a beginning or end of the input block.

In example 5, the subject matter of examples 1, 2 or 3, or any of the examples herein, wherein inserting the additional data into the input block includes inserting the additional data pseudorandomly within the input block.

In example 6, the subject matter of example 5, or any of the examples herein, wherein the pseudorandom insertion of the additional data is based on an index value obtained based on a subframe or frame index.

In example 7, the subject matter of examples 2 or 3, or any of the examples herein, wherein inserting the additional data into the input block includes inserting the additional data at equally spaced locations within the input block.

In example 8, the subject matter of examples 1, 2, or 3, or any of the examples herein, wherein the turbo coding technique includes a systematic convolutional turbo code.

In example 9, the subject matter of examples 1, 2, or 3, or any of the examples herein, wherein the circuity is further to: determine the size of the virtual code block based on: $b = a * (1/R - 1) /2$, where b is the size of the virtual code block, a is the size the input block, and R is a coding rate.

In a tenth example, a computer-readable medium may contain program instructions for causing one or more processors, associated with a communication device, to: process an input data stream to obtain an input block of a particular size; insert additional data into the input block to obtain a virtual code block having a size larger than the particular size of the input block; encode the virtual code block, using a turbo coding technique, to obtain an encoded code block, the encoded code block including a systematic portion and a parity portion; process the encoded code block to obtain an output data block that includes the systematic portion of the encoded code block, but without the additional data, and the parity bits.

In example 11, the subject matter of example 10, or any of the examples herein, wherein the communication device includes a User Equipment (UE) device or a evolved NodeB (eNB) device.

In example 12, the subject matter of examples 10 or 11, or any of examples herein, wherein the program instructions additionally cause the one or more processors to: insert the additional data into the input block by inserting the additional data at a beginning or end of the input block.

In example 13, the subject matter of examples 10 or 11, or any of the examples herein, wherein the program instructions additionally cause the one or more processors to: insert the additional data into the input block by inserting the additional data pseudorandomly within the input block.

In example 14, the subject matter of example 13, or any of the examples herein, wherein the pseudorandom insertion of the additional data is based on an index value obtained based on a subframe or frame index.

In example 15, the subject matter of example 10, or any of the examples, wherein the turbo coding technique includes a systematic convolutional turbo code.

In example 15, the subject matter of examples 10 or 11, or any of the examples herein, wherein the program instructions additionally cause the one or more processors to: determine the size of the virtual code block based on: $b = a * (1/R - 1) /2$, where b is the size of the virtual code block, a is the size the input block, and R is a coding rate.

In a seventeenth example, a communication device for a cellular network may be to: receive an indication of a coding rate; determine, based on the indication of the coding rate, a desired size of a virtual code block; process an input data stream to obtain an input block of a particular size; generate additional data, the length of the generated additional data being determined based on the desired size of the virtual code block and the particular size of the input block; insert the generated additional data into the input block to obtain the virtual code block having the desired size; encode the virtual code block, using a turbo coding technique, to obtain an encoded code block, the encoded code block including a systematic portion and a parity portion; compress the virtual code block to remove the sequence of additional data from the systematic portion of the encoded code block and to obtain an output data block; and output the output data block for transmission over a channel.

In example 18, the subject matter of example 17, or any of the examples herein, wherein the communication device includes a User Equipment (UE) device or a evolved NodeB (eNB) device.

In example 19, the subject matter of example 17, or any of the examples herein, wherein the communication device is additionally to: generate the additional data as a sequence of either all zero bits or as a sequence of all one bits.

In example 20, the subject matter of example 17, or any of the examples herein, wherein the communication device is additionally to: generate the additional data as a pseudorandom sequence of bits.

In example 21, the subject matter of examples 17-20, or any of the examples herein, wherein inserting the additional data into the input block includes: inserting the additional data at a beginning or end of the input block; or inserting the additional data pseudorandomly, within the input block, based on a predetermined insertion pattern.

In example 22, the subject matter of examples 17-20, or any of the examples herein, wherein the turbo coding technique includes a systematic convolutional turbo code.

In example 23, the subject matter of example 17, wherein the circuity is further to: determine the size of the virtual code block based on: $b = a * (1/R - 1) /2$, where b is the size of the virtual code block, a is the size the input block, and R is a coding rate.

In a twenty-fourth example, a method of encoding data comprises: processing an input data stream to obtain an input block of a particular size; inserting additional data into the input block to obtain a virtual code block having a size larger than the particular size of the input block; encoding the virtual code block, using a turbo coding technique, to obtain an encoded code block, the encoded code block including a systematic portion and a parity portion; removing the additional data from the systematic portion of the encoded code block to obtain an output data block that includes the systematic portion with the additional data removed and the parity portion; and outputting the output data block for transmission over a channel.

In example 25, the subject matter of example 24, or any of the examples herein, further comprising: generating the additional data as a sequence of all zero or one bits.

In example 26, the subject matter of example 24, or any of the examples herein, further comprising: generating the additional data as a pseudorandom sequence of bits.

In example 27, the subject matter of examples 24, 25, or 26, or any of the examples herein, wherein inserting the additional data into the input block includes inserting the additional data at a beginning or end of the input block.

In example 28, the subject matter of examples 24, 25 or 26, or any of the examples herein, wherein inserting the additional data into the input block includes inserting the additional data pseudorandomly within the input block.

In example 29, the subject matter of example 28, or any of the examples herein, wherein the pseudorandom insertion of the additional data is based on an index value obtained based on a subframe or frame index.

In example 30, the subject matter of example 24, or any of the examples herein, wherein inserting the additional data into the input block includes inserting the additional data at equally spaced locations within the input block.

In example 31, the subject matter of example 24, or any of the examples herein, wherein the turbo coding technique includes a systematic convolutional turbo code.

In example 32, the subject matter of example 24, or any of examples herein, further comprising: determining the size of the virtual code block based on: $b = a * (1/R - 1) /2$, where b is the size of the virtual code block, a is the size the input block, and R is a coding rate.

In example 33, a device may include means for processing an input data stream to obtain an input block of a particular size; means for inserting additional data into the input block to obtain a virtual code block having a size larger than the particular size of the input block; means for encoding the virtual code block, using a turbo coding technique, to obtain an encoded code block, the encoded code block including a systematic portion and a parity portion; means

for removing the additional data from the systematic portion of the encoded code block to obtain an output data block that includes the systematic portion with the additional data removed and the parity portion; and means for outputting the output data block for transmission over a channel.

In example 34, the subject matter of claim 33, or any of the examples herein, further comprising: means for generating the additional data as either a sequence of all zero bits or as a sequence of all one bits.

In example 35, the subject matter of claim 33, or any of the examples herein, further comprising: means for generating the additional data as a pseudorandom sequence of bits.

In example 36, the subject matter of claim 33, or any of the examples herein, wherein inserting the additional data into the input block includes inserting the additional data at a beginning or end of the input block.

In a thirty-seventh example, an apparatus for a baseband processor of an evolved NodeB (eNB), may include circuitry to process an input data stream to obtain an input block of a particular size; insert additional data into the input block to obtain a virtual code block having a size larger than the particular size of the input block; encode the virtual code block, using a turbo coding technique, to obtain an encoded code block, the encoded code block including a systematic portion and a parity portion; remove the additional data from the systematic portion of the encoded code block to obtain an output data block that includes the systematic portion with the additional data removed and the parity portion; and output the output data block for processing and transmission over a channel.

In example 38, the subject matter of claim 37, or any of the examples herein, wherein the circuitry is additionally to: generate the additional data as a sequence of either all zero bits or a sequence of all one bits.

In example 39, the subject matter of claim 37, or any of the examples herein, wherein the circuitry is additionally to: generate the additional data as a pseudorandom sequence of bits.

In example 40, the subject matter of any of claims 37, 38, or 39, or any of the examples herein, wherein inserting the additional data into the input block includes inserting the additional data at a beginning or end of the input block.

In example 41, the subject matter of any of claims 37, 38, or 39, or any of the examples herein, wherein inserting the additional data into the input block includes inserting the additional data pseudorandomly within the input block.

In the preceding specification, various embodiments have been described with reference to the accompanying drawings. It will, however, be evident that various modifications and

changes may be made thereto, and additional embodiments may be implemented, without departing from the broader scope as set forth in the claims that follow. The specification and drawings are accordingly to be regarded in an illustrative rather than restrictive sense.

For example, while series of signals and/or operations have been described with regard to Figs. 7 and 8, the order of the signals may be modified in other implementations. Further, non-dependent signals may be performed in parallel.

It will be apparent that example aspects, as described above, may be implemented in many different forms of software, firmware, and hardware in the implementations illustrated in the figures. The actual software code or specialized control hardware used to implement these aspects should not be construed as limiting. Thus, the operation and behavior of the aspects were described without reference to the specific software code—it being understood that software and control hardware could be designed to implement the aspects based on the description herein.

Even though particular combinations of features are recited in the claims and/or disclosed in the specification, these combinations are not intended to be limiting. In fact, many of these features may be combined in ways not specifically recited in the claims and/or disclosed in the specification.

No element, act, or instruction used in the present application should be construed as critical or essential unless explicitly described as such. An instance of the use of the term "and," as used herein, does not necessarily preclude the interpretation that the phrase "and/or" was intended in that instance. Similarly, an instance of the use of the term "or," as used herein, does not necessarily preclude the interpretation that the phrase "and/or" was intended in that instance. Also, as used herein, the article "a" is intended to include one or more items, and may be used interchangeably with the phrase "one or more." Where only one item is intended, the terms "one," "single," "only," or similar language is used.

WHAT IS CLAIMED IS:

1. An apparatus for a baseband processor of User Equipment (UE), the apparatus comprising circuitry to:

process an input data stream to obtain an input block of a particular size;

insert additional data into the input block to obtain a virtual code block having a size larger than the particular size of the input block;

encode the virtual code block, using a turbo coding technique, to obtain an encoded code block, the encoded code block including a systematic portion and a parity portion;

remove the additional data from the systematic portion of the encoded code block to obtain an output data block that includes the systematic portion with the additional data removed and the parity portion; and

output the output data block for processing and transmission over a channel.

2. The apparatus of claim 1, wherein the circuitry is additionally to:

generate the additional data as a sequence of either all zero bits or a sequence of all one bits.

3. The apparatus of claim 1, wherein the circuitry is additionally to:

generate the additional data as a pseudorandom sequence of bits.

4. The apparatus of any of claims 1, 2 or 3, wherein inserting the additional data into the input block includes inserting the additional data at a beginning or end of the input block.

5. The apparatus of any of claims 1, 2 or 3, wherein inserting the additional data into the input block includes inserting the additional data pseudorandomly within the input block.

6. The apparatus of claim 5, wherein the pseudorandom insertion of the additional data is based on an index value obtained based on a subframe or frame index.

7. The apparatus of any of claims 2 or 3, wherein inserting the additional data into the input block includes inserting the additional data at equally spaced locations within the input block.

8. The apparatus of any of claims 1, 2, or 3, wherein the turbo coding technique includes a systematic convolutional turbo code.

9. The apparatus of any of claims 1, 2, or 3, wherein the circuity is further to:

determine the size of the virtual code block based on:

$b = a * (1/R - 1) /2$

where b is the size of the virtual code block, a is the size the input block, and R is a coding rate.

10. A computer-readable medium containing program instructions for causing one or more processors, associated with a communication device, to:

process an input data stream to obtain an input block of a particular size;

insert additional data into the input block to obtain a virtual code block having a size larger than the particular size of the input block;

encode the virtual code block, using a turbo coding technique, to obtain an encoded code block, the encoded code block including a systematic portion and a parity portion;

process the encoded code block to obtain an output data block that includes the systematic portion of the encoded code block, but without the additional data, and the parity bits; and

output the output data block for processing and transmission over a channel.

11. The computer-readable medium of claim 10, wherein the communication device includes a User Equipment (UE) device or a evolved NodeB (eNB) device.

12. The computer-readable medium of claim 10 or 11, wherein the program instructions additionally cause the one or more processors to:

insert additional data into the input block by inserting the additional data at a beginning or end of the input block.

13. The computer-readable medium of claim 10 or 11, wherein the program instructions additionally cause the one or more processors to:

insert the additional data into the input block by inserting the additional data pseudorandomly within the input block.

14. The computer-readable medium of claim 13, wherein the pseudorandom insertion of the additional data is based on an index value obtained based on a subframe or frame index.

15. The computer-readable medium of claim 10, wherein the turbo coding technique includes a systematic convolutional turbo code.

16. The computer-readable medium of any of claims 10 or 11, wherein the program instructions additionally cause the one or more processors to:

determine the size of the virtual code block based on:

$b = a * (1/R - 1) /2$

where b is the size of the virtual code block, a is the size the input block, and R is a coding rate.

17. A communication device for a cellular network, the communication device to:

receive an indication of a coding rate;

determine, based on the indication of the coding rate, a desired size of a virtual code block;

process an input data stream to obtain an input block of a particular size;

generate additional data, the length of the generated additional data being determined based on the desired size of the virtual code block and the particular size of the input block;

insert the generated additional data into the input block to obtain the virtual code block having the desired size;

encode the virtual code block, using a turbo coding technique, to obtain an encoded code block, the encoded code block including a systematic portion and a parity portion;

compress the virtual code block to remove the sequence of additional data from the systematic portion of the encoded code block and to obtain an output data block; and

output the output data block for transmission over a channel.

18. The communication device of claim 17, wherein the communication device includes a User Equipment (UE) device or a evolved NodeB (eNB) device.

19. The communication device of claim 17, wherein the communication device is additionally to:

generate the additional data as a sequence of either all zero bit or as a sequence of all one bits.

20. The communication device of claim 17, wherein the communication device is additionally to:

generate the additional data as a pseudorandom sequence of bits.

21. The communication device of any one of claims 17-20, wherein inserting the additional data into the input block includes:

inserting the additional data at a beginning or end of the input block; or

inserting the additional data pseudorandomly, within the input block, based on a predetermined insertion pattern.

22. The communication device of any one of claims 17-20, wherein the turbo coding technique includes a systematic convolutional turbo code.

23. The communication device of claim 17, wherein the circuity is further to:

determine the size of the virtual code block based on:

$b = a * (1/R - 1) /2$

where b is the size of the virtual code block, a is the size the input block, and R is a coding rate.

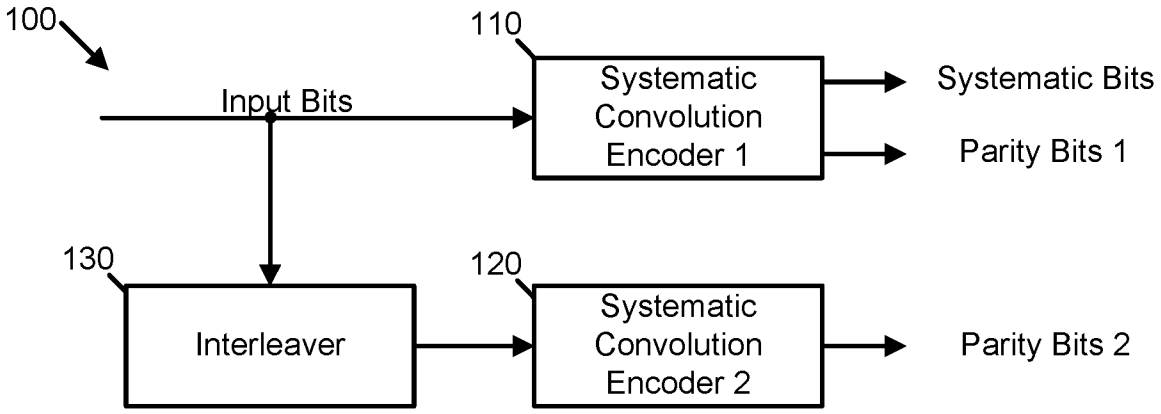

100
110
Input Bits
Systematic Convolution Encoder 1
Systematic Bits
Parity Bits 1
130
120
Interleaver
Systematic Convolution Encoder 2
Parity Bits 2

Fig. 1A

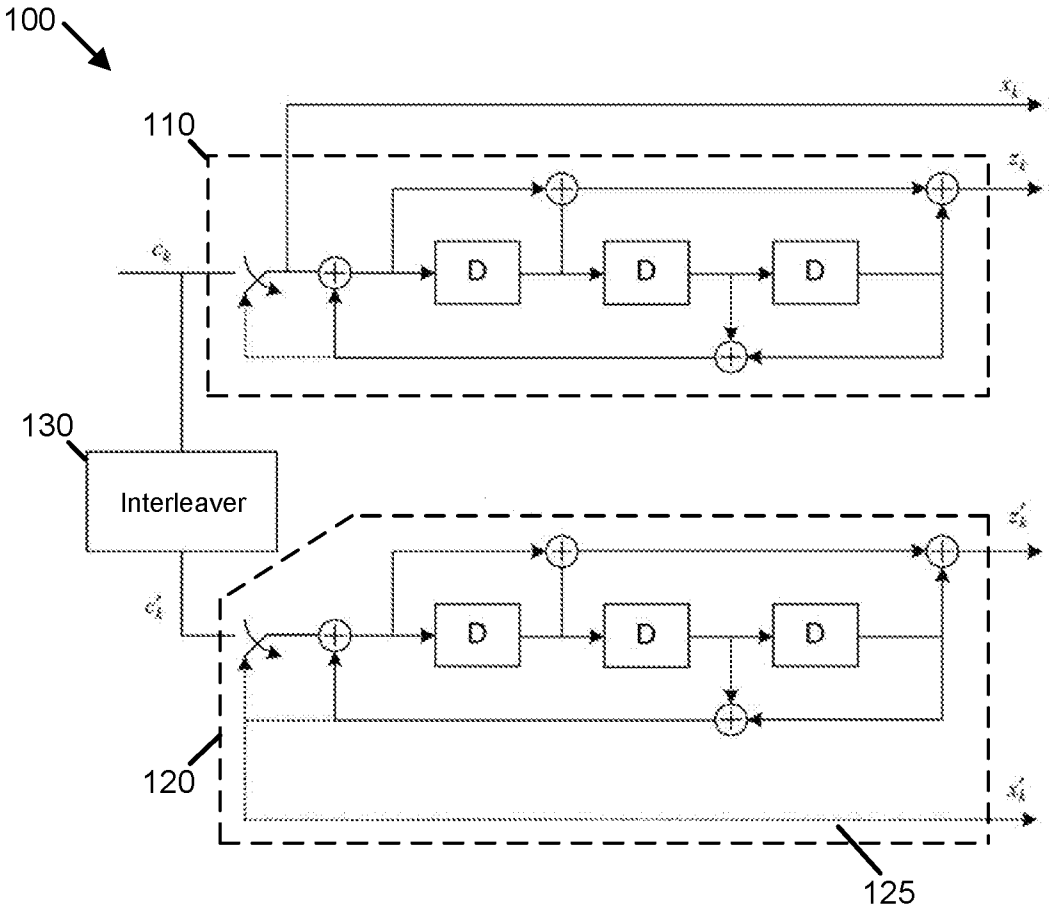

100
110
D
D
D
130
Interleaver
D
D
D
120
125

Fig. 1B

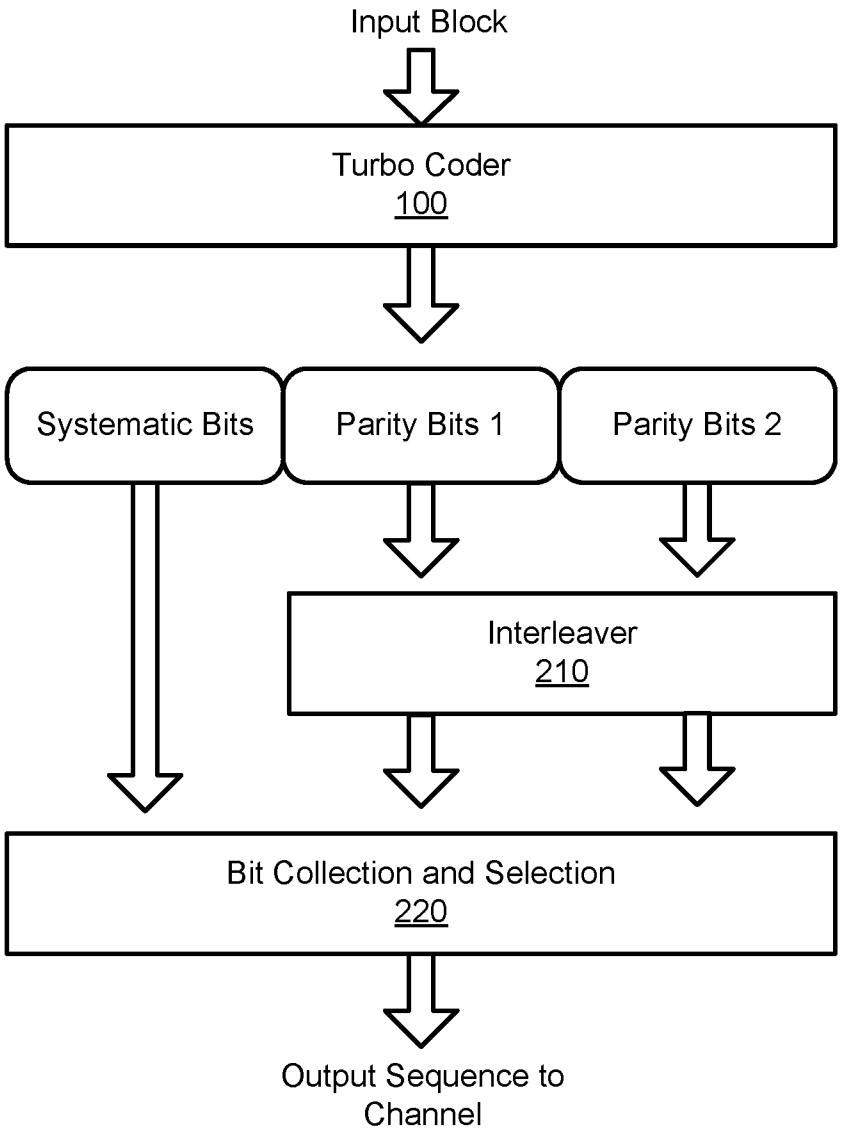

Input Block
Turbo Coder
100
Systematic Bits
Parity Bits 1
Parity Bits 2
Interleaver
210
Bit Collection and Selection
220
Output Sequence to
Channel

Fig. 2

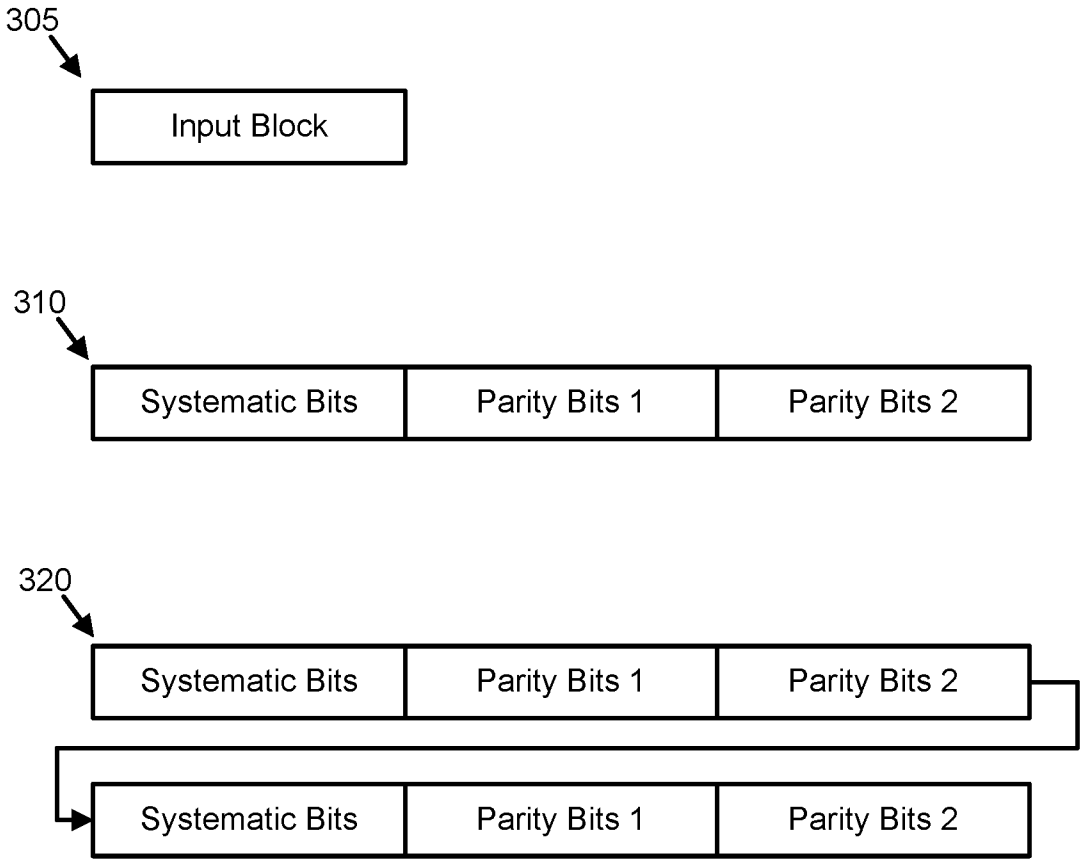

305
Input Block
310
Systematic Bits
Parity Bits 1
Parity Bits 2
320
Systematic Bits
Parity Bits 1
Parity Bits 2
Systematic Bits
Parity Bits 1
Parity Bits 2

Fig. 3

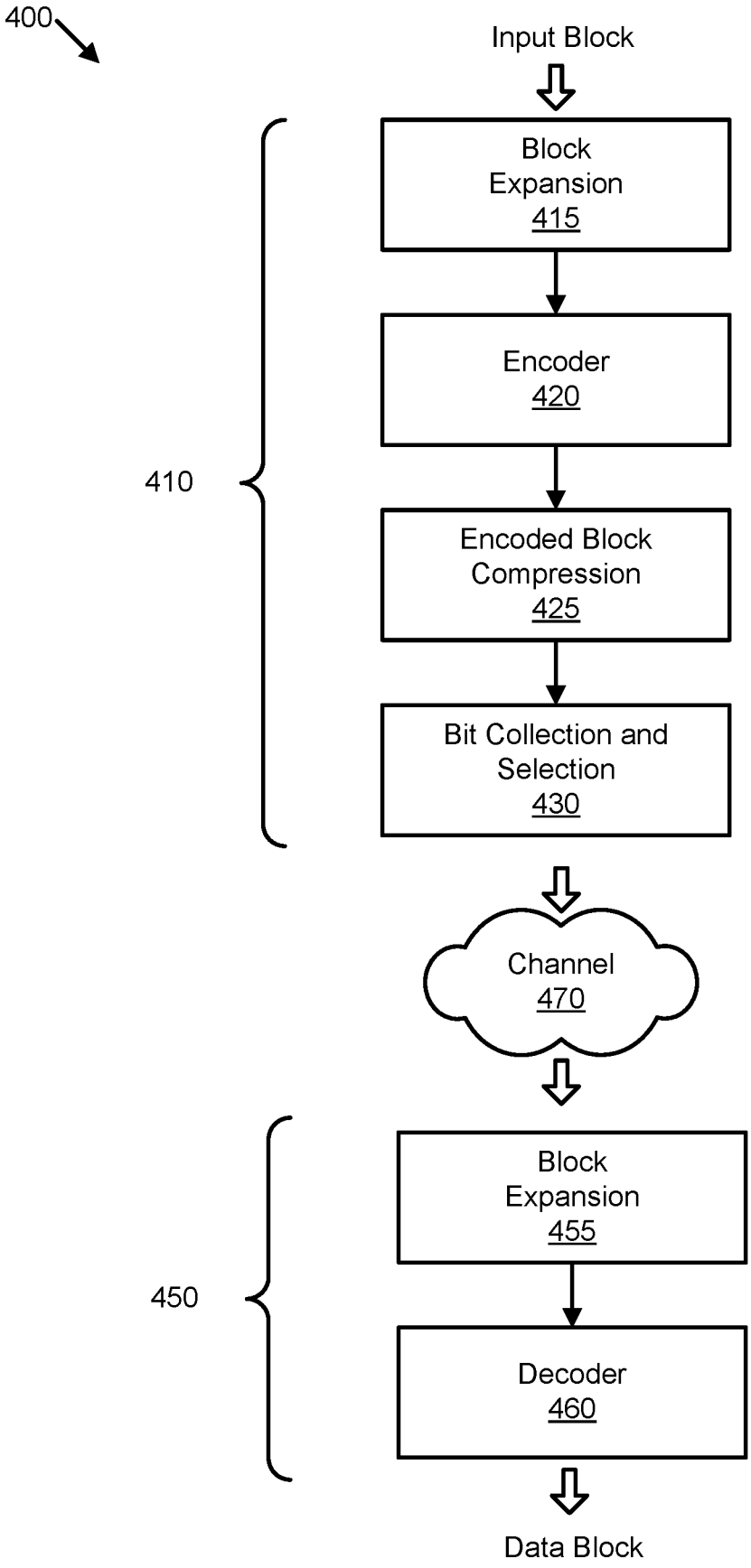

400
Input Block
Block Expansion 415
Encoder 420
410
Encoded Block Compression 425
Bit Collection and Selection 430
Channel 470
Block Expansion 455
450
Decoder 460
Data Block

Fig. 4

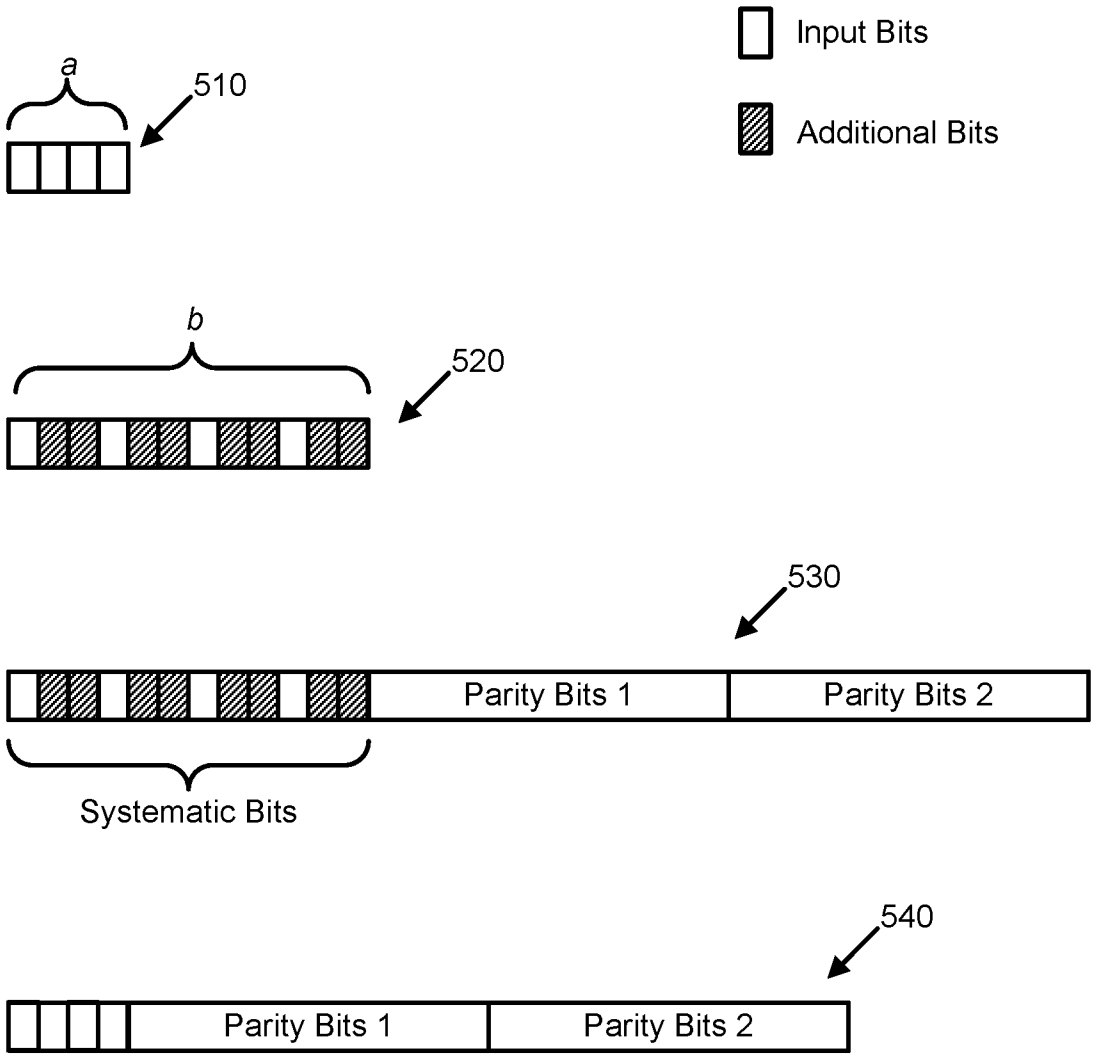

Input Bits
Additional Bits
a
510
b
520
530
Parity Bits 1
Parity Bits 2
Systematic Bits
540
Parity Bits 1
Parity Bits 2

Fig. 5

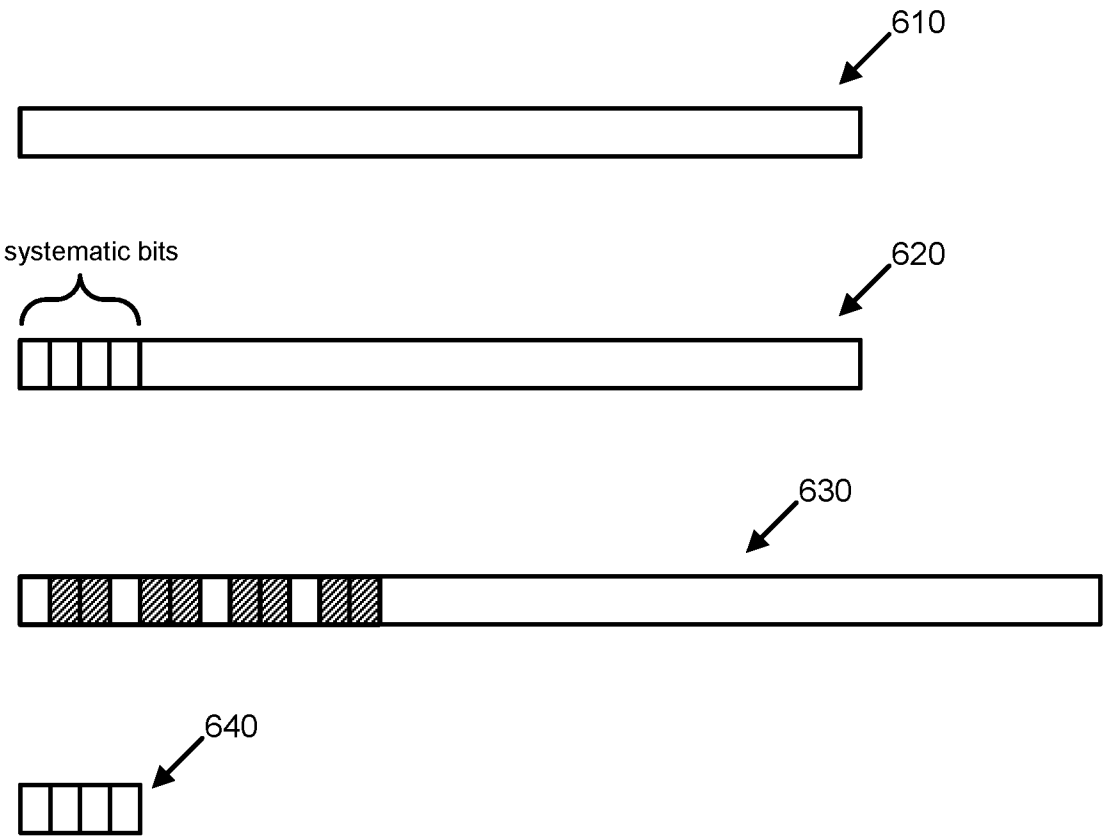


Fig. 6

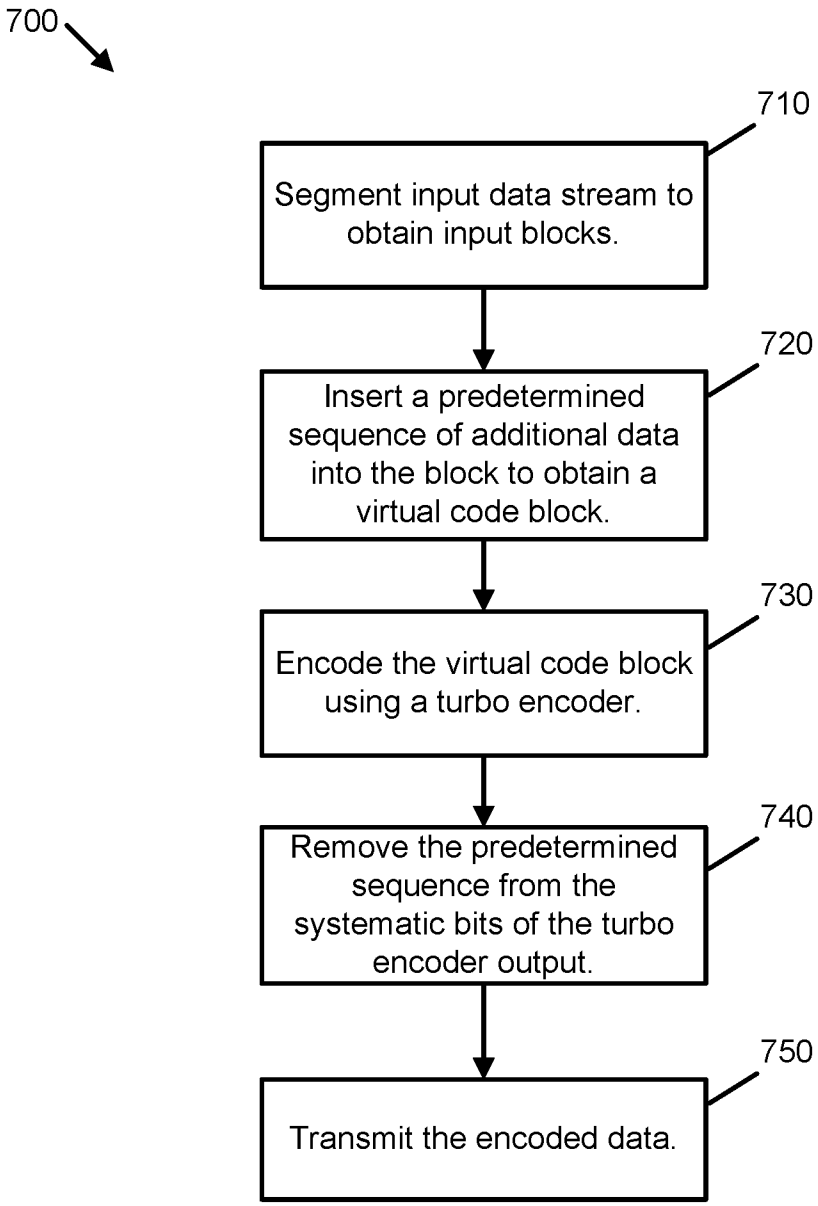

700
710
Segment input data stream to obtain input blocks.
720
Insert a predetermined sequence of additional data into the block to obtain a virtual code block.
730
Encode the virtual code block using a turbo encoder.
740
Remove the predetermined sequence from the systematic bits of the turbo encoder output.
750
Transmit the encoded data.

Fig. 7

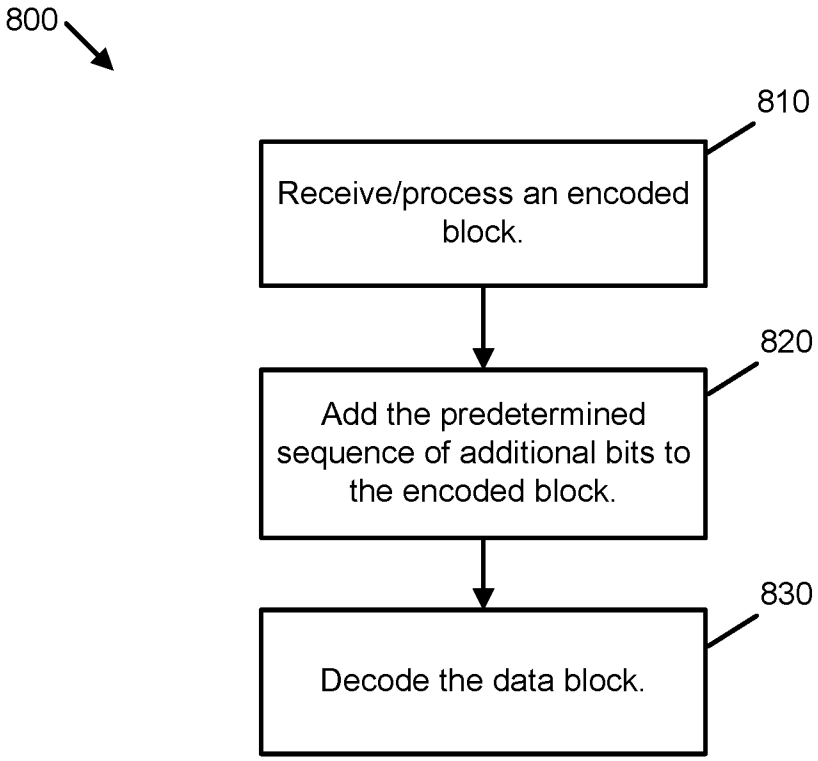

800
810
Receive/process an encoded block.
820
Add the predetermined sequence of additional bits to the encoded block.
830
Decode the data block.

Fig. 8

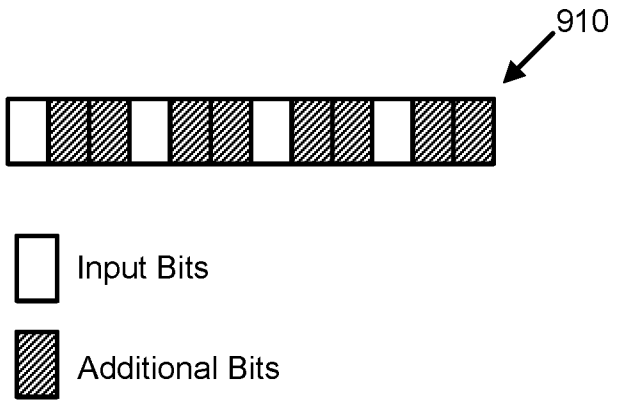

910
Input Bits
Additional Bits

Fig. 9A

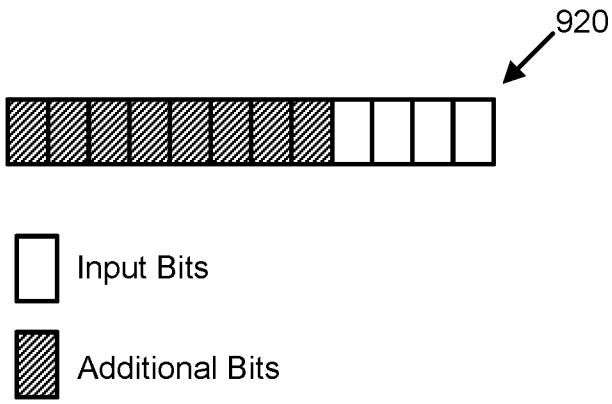

920
Input Bits
Additional Bits

Fig. 9B

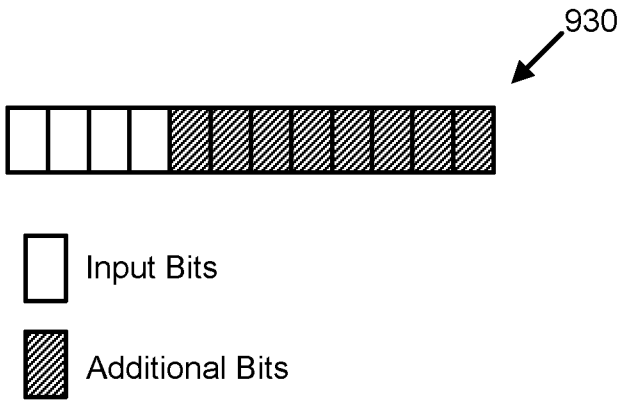

930
Input Bits
Additional Bits

Fig. 9C

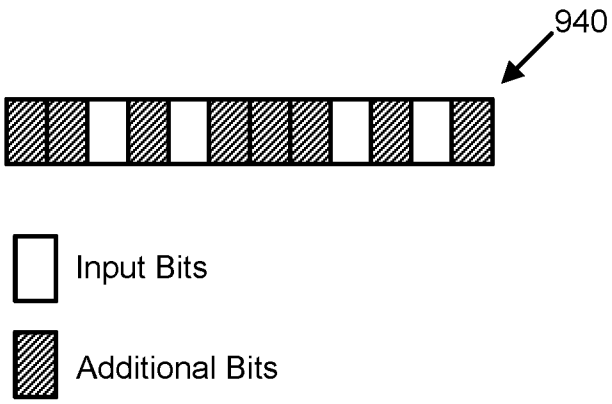

940
Input Bits
Additional Bits

Fig. 9D

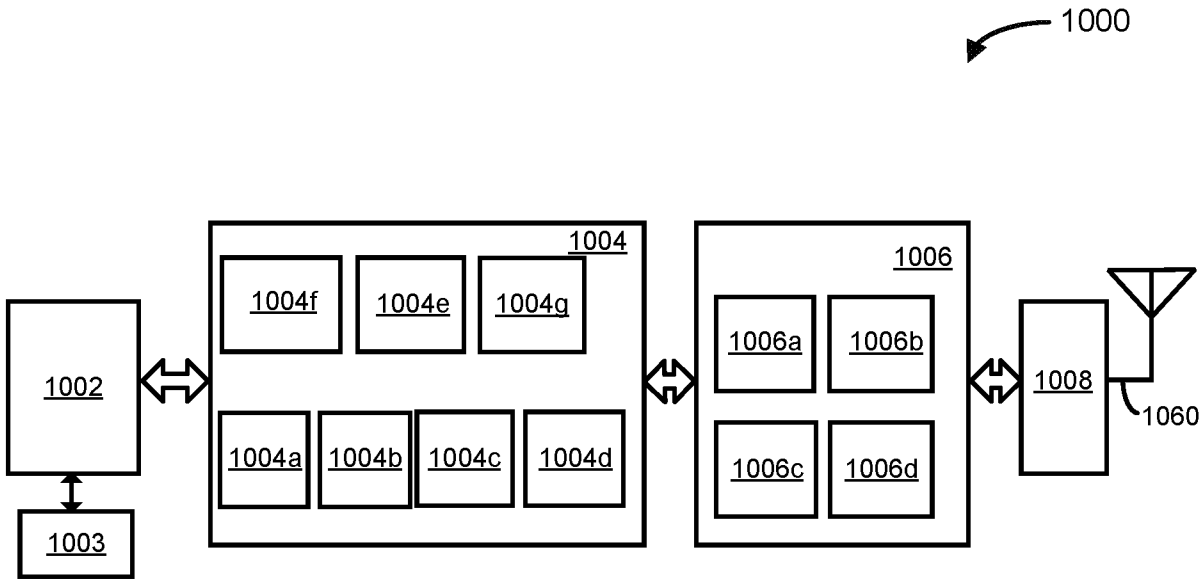

1000
1004
1004f
1004e
1004g
1004a
1004b
1004c
1004d
1006
1006a
1006b
1006c
1006d
1002
1003
1008
1060

Fig. 10

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2016/039077

A. CLASSIFICATION OF SUBJECT MATTER
INV. H03M13/29 H03M13/39
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H03M H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, COMPENDEX, INSPEC, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	EP 1 837 999 A1 (FUJITSU LTD [JP]) 26 September 2007 (2007-09-26) paragraphs [0015], [0016], [0034], [0035] figures 1,2,21,22 -----	1-6,8-16 19-21
X	WO 03/050992 A1 (ERICSSON TELEFON AB L M [SE]; SCHULIST MATTHIAS [DE]; WACHSMANN UDO [D) 19 June 2003 (2003-06-19) page 1, line 15 - line 16 page 48, line 1 - page 49, line 23 page 57, line 22 - line 25 page 58, line 1 - line 3 claim 12 figures 4a, 4b ----- -/--	1-3, 5-11, 13-16

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
24 October 2016	04/11/2016

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Burkert, Frank

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2016/039077

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	SHUNJI MIYAZAKI ET AL: "Link-Level Evaluations of Low Rate Code Extensions of Turbo Codes Using Shortened Code Techniques", PROC. IEEE 2007 VEHICULAR TECHNOLOGY CONFERENCE (FALL), IEEE, PI, 1 September 2007 (2007-09-01), pages 1017-1021, XP031147559, ISBN: 978-1-4244-0263-2 abstract page 1017, right-hand column page 1018 figures 1,2	1,2,4, 7-12,15, 16

X	FUJITSU: "Lower rate extension of channel coding to the rate 1/3", 3GPP DRAFT; R1-051439, 3RD GENERATION PARTNERSHIP PROJECT (3GPP), MOBILE COMPETENCE CENTRE ; 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS CEDEX ; FRANCE, vol. RAN WG1, no. Seoul, Korea; 20051101, 1 November 2005 (2005-11-01), XP050101029, [retrieved on 2005-11-01]	1,8-11, 15-18, 22,23
Y	pages 1,2 table 1 figures 1-3	19-21

X	KATO O ET AL: "Packet error rate analysis and its reduction by known bits insertion for turbo code in 100Mbps OFCDM system", PROC. IEEE 2003 VEHICULAR TECHNOLGY CONFERENCE (SPRING), JEJU, KOREA, APRIL 22 - 25, 2003; [IEEE VEHICULAR TECHNOLGY CONFERENCE], NEW YORK, NY : IEEE, US, vol. 3, 22 April 2003 (2003-04-22), pages 2120-2124, XP010862472, DOI: 10.1109/VETECS.2003.1207200 ISBN: 978-0-7803-7757-8 abstract page 2120, left-hand column page 2122 figure 1	1,2,7,8, 10,11,15

	-/--	

1

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SAMSUNG: "Narrowband IOT - Consideration of Uplink Channel Coding", 3GPP DRAFT; R1-155515 NARROWBAND IOT - CONSIDERATION OF UPLINK CHANNEL CODING, 3RD GENERATION PARTNERSHIP PROJECT (3GPP), MOBILE COMPETENCE CENTRE ; 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS , vol. RAN WG1, no. Malmo, Sweden; 20151005 - 20151009 4 October 2015 (2015-10-04), XP051002406, Retrieved from the Internet: URL:http://www.3gpp.org/ftp/Meetings_3GPP_SYNC/RAN1/Docs/ [retrieved on 2015-10-04] page 1 - page 2 figures 1-3 -----	1-23

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2016/039077

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1837999 A1	26-09-2007	CN 101103533 A	09-01-2008
		EP 1837999 A1	26-09-2007
		EP 2068450 A2	10-06-2009
		JP 4425282 B2	03-03-2010
		WO 2006075382 A1	20-07-2006
		WO 2006075417 A1	20-07-2006
WO 03050992 A1	19-06-2003	AU 2002231682 A1	23-06-2003
		EP 1451962 A1	01-09-2004
		WO 03050992 A1	19-06-2003