TECHNIQUES FOR PARALLEL POLAR CODE DECODING USING A PERMUTATION MATRIX

TECHNICAL FIELD

[0001] The present disclosure relates to wireless communications, and more specifically to techniques for parallelization of polar code decoders using one or more permutation matrices.

BACKGROUND

[0002] A wireless communications system may include one or multiple network communication devices, which may be known as a network equipment (NE), supporting wireless communications for one or multiple user communication devices, which may be otherwise known as user equipment (UE), or other suitable terminology. The wireless communications system may support wireless communications with one or multiple user communication devices by utilizing resources of the wireless communication system (e.g., time resources (e.g., symbols, slots, subframes, frames, or the like) or frequency resources (e.g., subcarriers, carriers, or the like). Additionally, the wireless communications system may support wireless communications across various radio access technologies (RATs) including third generation (3G) radio access technology, fourth generation (4G) radio access technology, fifth generation (5G) radio access technology, among other suitable radio access technologies beyond 5G (e.g., 5G-Advanced (5G-A), sixth generation (6G), etc.).

SUMMARY

[0003] An article "a" before an element is unrestricted and understood to refer to "at least one" of those elements or "one or more" of those elements. The terms "a," "at least one," "one or more," and "at least one of one or more" may be interchangeable. As used herein, including in the claims, "or" as used in a list of items (e.g., a list of items prefaced by a phrase such as "at least one of" or "one or more of" or "one or both of) indicates an inclusive list such that, for example, a list of at least one of A, B, or C means A or B or C or AB or AC or BC or ABC (i.e., A and B and C). Also, as used herein, the phrase "based on" shall not be construed as a reference to a closed set of conditions. For example, an example step that is described as "based on condition A" may be based on both a

condition A and a condition B without departing from the scope of the present disclosure. In other words, as used herein, the phrase "based on" shall be construed in the same manner as the phrase "based at least in part on." Further, as used herein, including in the claims, a "set" may include one or more elements.

[0004] A UE for wireless communication is described. The UE may be configured to, capable of, or operable to receive a polar codeword comprising a set of bits; generate a permuted vector based at least in part on the polar codeword and a permutation matrix; determine a plurality of polar subcodes based on the permuted vector; identify, for each polar subcode of the plurality of polar subcodes, a set of kernels, where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and decode the plurality of polar subcodes using a plurality of parallel polar code component decoders, where each parallel polar code component decoder includes one or more fast decode modules to decode a corresponding kernel of the set of kernels.

[0005] A processor for wireless communication is described. The processor may be configured to, capable of, or operable to receive, at a UE, a polar codeword comprising a set of bits; generate a permuted vector based at least in part on the polar codeword and a permutation matrix; determine a plurality of polar subcodes based on the permuted vector; identify, for each polar subcode of the plurality of polar subcodes, a set of kernels, where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and decode the plurality of polar subcodes using a plurality of parallel polar code component decoders, where each parallel polar code component decoder includes one or more fast decode modules to decode a corresponding kernel of the set of kernels.

[0006] A method performed or performable by a UE for wireless communication is described. The method may include receiving, at the UE, a polar codeword comprising a set of bits; generating a permuted vector based at least in part on the polar codeword and a permutation matrix; determining a plurality of polar subcodes based on the permuted vector; identifying, for each polar subcode of the plurality of polar subcodes, a set of kernels, where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and decoding the plurality of polar subcodes using a plurality of parallel polar code component decoders, where each parallel polar code component decoder includes one or more fast decode modules to decode a corresponding kernel of the set of kernels.

[0007] A base station for wireless communication is described. The base station may be configured to, capable of, or operable to receive a polar codeword comprising a set of bits; generate a permuted vector based at least in part on the polar codeword and a permutation matrix; determine a plurality of polar subcodes based on the permuted vector; identify, for each polar subcode of the plurality of polar subcodes, a set of kernels, where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and decode the plurality of polar subcodes using a plurality of parallel polar code component decoders, where each parallel polar code component decoder includes one or more fast decode modules to decode a corresponding kernel of the set of kernels.

[0008] A processor for wireless communication by a base station is described. The processor may be configured to, capable of, or operable to receive, at a base station, a polar codeword comprising a set of bits; generate a permuted vector based at least in part on the polar codeword and a permutation matrix; determine a plurality of polar subcodes based on the permuted vector; identify, for each polar subcode of the plurality of polar subcodes, a set of kernels, where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and decode the plurality of polar subcodes using a plurality of parallel polar code component decoders, where each parallel polar code component decoder includes one or more fast decode modules to decode a corresponding kernel of the set of kernels.

[0009] A method performed or performable by a base station for wireless communication is described. The method may include receiving, at the base station, a polar codeword comprising a set of bits; generating a permuted vector based at least in part on the polar codeword and a permutation matrix; determining a plurality of polar subcodes based on the permuted vector; identifying, for each polar subcode of the plurality of polar subcodes, a set of kernels, where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and decoding the plurality of polar subcodes using a plurality of parallel polar code component decoders, where each parallel polar code component decoder includes one or more fast decode modules to decode a corresponding kernel of the set of kernels.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Figure 1 illustrates an example of a wireless communications system in accordance with aspects of the present disclosure.

[0011] Figure 2 illustrates an example of protocol stacks in accordance with aspects of the present disclosure.

[0012] Figure 3 illustrates an example of use cases and target requirements in accordance with aspects of the present disclosure.

[0013] Figure 4 illustrates an example of channel combining in accordance with aspects of the present disclosure.

[0014] Figure 5 illustrates an example of a channel polarization effects in accordance with aspects of the present disclosure.

[0015] Figure 6A illustrates an example of a binary tree traversal in accordance with aspects of the present disclosure.

[0016] Figure 6B illustrates an example of messaging during the binary tree traversal in accordance with aspects of the present disclosure.

[0017] Figure 7 illustrates an example of a rate-0 or repetition (SR0/REP) node in accordance with aspects of the present disclosure.

[0018] Figure 8 illustrates an example of a permutation-based decoder with parallel component decoders in accordance with aspects of the present disclosure.

[0019] Figure 9 illustrates an example of a stage-shuffling permutation of the factor graph of the polar code in accordance with aspects of the present disclosure.

[0020] Figure 10 illustrates an example of a permutation-based decoder in accordance with aspects of the present disclosure.

[0021] Figure 11 illustrates an example of polar decoding using multiple permutations in accordance with aspects of the present disclosure.

[0022] Figure 12 illustrates an example of a UE in accordance with aspects of the present disclosure.

[0023] Figure 13 illustrates an example of a processor in accordance with aspects of the present disclosure.

[0024] Figure 14 illustrates an example of a NE in accordance with aspects of the present disclosure.

[0025] Figure 15 illustrates a flowchart of a method performed by a UE in accordance with aspects of the present disclosure.

[0026] Figure 16 illustrates a flowchart of a method performed by an NE in accordance with aspects of the present disclosure.

DETAILED DESCRIPTION

[0027] A wireless communications system, including one or more network entities or UEs may support various radio access technologies, including but not limited to 5G, 6G, and radio access technologies beyond 6G. The one or more network entities or UEs may be capable of or configured to support enhanced ultra-reliable low latency communication (eURLLC) and enhanced mobile broadband (eMBB), which may be associated with applications or services requiring low end-to-end transmission latency, ultra-reliability, packet size flexibility and availability as well as high throughput (e.g., around 1Tbit/s). In some cases, eURLLC may enable connectivity for services and applications associated with various industries, such as factory automation, tactile internet, autonomous driving, and so on.

[0028] The one or more network entities or UEs may be capable of or configured to support channel coding (e.g., encoding, decoding) to support efficient eURLLC and eMBB. In some cases, the channel coding may involve using polar codes, for example, due to the performance and low complexity characteristics of using the polar codes for channel coding (e.g., encoding, decoding). In some cases, the one or more network entities or UEs may be susceptible to coding delays (e.g., decoding delays) associated with using polar codes. In some cases, due to the lack of parallelization for decoding polar codes (e.g., when sequential cyclic redundancy check (CRC)-aided successive cancellation list (CRC-SCL) decoding is used), the one or more network entities or UEs may be susceptible coding delays (e.g., decoding delays). Some fast decoders may support high throughput gains but at the expense of adjusting a polar encoder framework. Additionally, some fast and simplified polar code SCL decoders may be based on a tree traversal technique.

[0029] With tree traversal, the polar code is partitioned into a tree-like data structure that represents all possible combinations of bits or codewords. The tree-like data structure includes multiple nodes, at various levels of the tree. The tree traversal technique navigates (i.e., "explores") through the tree-like data structure and, at each node, the decoder performs computations based on the received signal and the code properties to determine the likelihood of different codewords or paths through the tree. Tree traversal allows for flexibility in exploring different paths or hypotheses in the decoding process, while avoiding redundant computations. However, tree traversal algorithms can have high computational complexity and may have a relatively slow converge speed (i.e., compared to other decoders) which may lead to higher decoding latency.

[0030] The fast SCL decoders described herein may be based on the implementation of parallel decoders at the intermediate levels of the successive cancellation (SC) decoding tree for some special nodes (or special kernels) with specific patterns of information and frozen bits. Several special nodes have been identified such as Rate-1, Rate-0, repetition (REP) and single parity check (SPC) nodes, which enabled high decoding latency and throughput gains. Using parallel decoders provides latency and throughput gains, however, the implementation of parallel decoders alone may not be sufficient to attain 6G key performance indicators (KPIs), therefore the present disclosure described additional (e.g., more aggressive) techniques to attain the 6G KPIs.

[0031] Various aspects of the present disclosure relate to techniques for parallelization of a fast simplified successive cancellation (FSSC) decoder. The one or more network entities or UEs may be capable of or configured to receive a polar codeword and permutate the received polar codeword in order to create partitions of special nodes/kernels that can be fast decoded using parallel component decoders without having to perform one or more tree traversal techniques.

[0032] To permute the polar codeword, a receiver of a network entity or a UE may apply one or more permutation matrices. The network entity or the UE may dynamically determine a permutation matrix. In some implementations, the network entity or the UE may sample (e.g., select) a permutation matrix from a set of permutation matrices. These permutation matrices could be applied by the network entity or the UE to the received codeword or a polar code factor graph, or a combination thereof. A high-level gain evaluation shows that the techniques described herein reduce decoding latency and

improve throughput gains without altering the error-correction performance of the polar code.

[0033] As used herein, a “fast decoder" refers to a category of decoding algorithm with low computational complexity and fast execution time to facilitate the real-time, or near-real-time, decoding of sophisticated encoding schemes. Examples of fast decoders include the fast successive cancellation decoder, the FSSC decoder, and the like. A component decoder refers to a decoding algorithm used as part of a larger decoding process, e.g., by operating on a partitioned section (or component) of the codeword. Accordingly, a fast component decoder refers to a fast decoder that also operates as a component decoder in the decoding process.

[0034] As used herein, a “factor graph” refers to the representation of the factorization of a function of several variables. A factor graph may be used to represent probabilistic modeling and inference tasks, e.g., in channel decoding, thereby capturing the relationships between random variables and the factors that connect them. One type of factor graph is the Forney-style factor graph (FFG), designed to represent the factorization of the joint probability distribution of random variables according to the underlying probabilistic model. In various embodiments, an FFG includes variable nodes representing random variables and factor nodes representing factors or functions that relate variables. These nodes are connected by edges that denote dependencies.

[0035] Aspects of the present disclosure are described in the context of a wireless communications system.

[0036] **Figure 1** illustrates an example of a wireless communications system 100 in accordance with aspects of the present disclosure. The wireless communications system 100 may include one or more NE 102, one or more UE 104, and a core network (CN) 106. The wireless communications system 100 may support various radio access technologies. In some implementations, the wireless communications system 100 may be a 4G network, such as a Long-Term Evolution (LTE) network or an LTE-Advanced (LTE-A) network. In some other implementations, the wireless communications system 100 may be a New Radio (NR) network, such as a 5G network, a 5G-Advanced (5G-A) network, or a 5G ultrawideband (5G-UWB) network.

[0037] In other implementations, the wireless communications system 100 may be a combination of a 4G network and a 5G network, or other suitable radio access technology

(RAT) including Institute of Electrical and Electronics Engineers (IEEE) 802.11 (Wi-Fi), IEEE 802.16 (WiMAX), IEEE 802.20. The wireless communications system 100 may support radio access technologies beyond 5G, for example, 6G. Additionally, the wireless communications system 100 may support technologies, such as time division multiple access (TDMA), frequency division multiple access (FDMA), or code division multiple access (CDMA), etc.

[0038] The one or more NE 102 may be dispersed throughout a geographic region to form the wireless communications system 100. One or more of the NE 102 described herein may be or include or may be referred to as a network node, a base station, a network element, a network function, a network entity, a radio access network (RAN), a NodeB, an eNodeB (eNB), a next-generation NodeB (gNB), or other suitable terminology. An NE 102 and a UE 104 may communicate via a communication link, which may be a wireless or wired connection. For example, an NE 102 and a UE 104 may perform wireless communication (e.g., receive signaling, transmit signaling) over a Uu interface.

[0039] An NE 102 may provide a geographic coverage area for which the NE 102 may support services for one or more UEs 104 within the geographic coverage area. For example, an NE 102 and a UE 104 may support wireless communication of signals related to services (e.g., voice, video, packet data, messaging, broadcast, etc.) according to one or multiple radio access technologies. In some implementations, an NE 102 may be moveable, for example, a satellite associated with a non-terrestrial network (NTN). In some implementations, different geographic coverage areas associated with the same or different radio access technologies may overlap, but the different geographic coverage areas may be associated with different NE 102.

[0040] The one or more UE 104 may be dispersed throughout a geographic region of the wireless communications system 100. A UE 104 may include or may be referred to as a remote unit, a mobile device, a wireless device, a remote device, a subscriber device, a transmitter device, a receiver device, or some other suitable terminology. In some implementations, the UE 104 may be referred to as a unit, a station, a terminal, or a client, among other examples. Additionally, or alternatively, the UE 104 may be referred to as an internet-of-things (IoT) device, an internet-of-everything (IoE) device, or machine-type communication (MTC) device, among other examples.

[0041] A UE 104 may be able to support wireless communication directly with other UEs 104 over a communication link. For example, a UE 104 may support wireless communication directly with another UE 104 over a device-to-device (D2D) communication link. In some implementations, such as vehicle-to-vehicle (V2V) deployments, vehicle-to-everything (V2X) deployments, or cellular-V2X deployments, the communication link may be referred to as a sidelink. For example, a UE 104 may support wireless communication directly with another UE 104 over a PC5 interface.

[0042] An NE 102 may support communications with the CN 106, or with another NE 102, or both. For example, an NE 102 may interface with other NE 102 or the CN 106 through one or more backhaul links (e.g., S1, N2, N3, or network interface). In some implementations, the NE 102 may communicate with each other directly. In some other implementations, the NE 102 may communicate with each other or indirectly (e.g., via the CN 106). In some implementations, one or more NE 102 may include subcomponents, such as an access network entity, which may be an example of an access node controller (ANC). An ANC may communicate with the one or more UEs 104 through one or more other access network transmission entities, which may be referred to as a radio heads, smart radio heads, or transmission-reception points (TRPs).

[0043] The CN 106 may support user authentication, access authorization, tracking, connectivity, and other access, routing, or mobility functions. The CN 106 may be an evolved packet core (EPC), or a 5G core (5GC), which may include a control plane entity that manages access and mobility (e.g., a mobility management entity (MME), an access and mobility management functions (AMF)) and a user plane entity that routes packets or interconnects to external networks (e.g., a serving gateway (S-GW), a Packet Data Network (PDN) gateway (P-GW), or a user plane function (UPF)). In some implementations, the control plane entity may manage non-access stratum (NAS) functions, such as mobility, authentication, and bearer management (e.g., data bearers, signal bearers, etc.) for the one or more UEs 104 served by the one or more NE 102 associated with the CN 106.

[0044] The CN 106 may communicate with a packet data network over one or more backhaul links (e.g., via an S1, N2, N3, or another network interface). The packet data network may include an application server. In some implementations, one or more UEs 104 may communicate with the application server. A UE 104 may establish a session (e.g., a protocol data unit (PDU) session, or a PDN connection, or the like) with the CN

106 via an NE 102. The CN 106 may route traffic (e.g., control information, data, and the like) between the UE 104 and the application server using the established session (e.g., the established PDU session). The PDU session may be an example of a logical connection between the UE 104 and the CN 106 (e.g., one or more network functions of the CN 106).

[0045] In the wireless communications system 100, the NEs 102 and the UEs 104 may use resources of the wireless communications system 100 (e.g., time resources (e.g., symbols, slots, subframes, frames, or the like) or frequency resources (e.g., subcarriers, carriers)) to perform various operations (e.g., wireless communications). In some implementations, the NEs 102 and the UEs 104 may support different resource structures. For example, the NEs 102 and the UEs 104 may support different frame structures. In some implementations, such as in 4G, the NEs 102 and the UEs 104 may support a single frame structure. In some other implementations, such as in 5G and among other suitable radio access technologies, the NEs 102 and the UEs 104 may support various frame structures (i.e., multiple frame structures). The NEs 102 and the UEs 104 may support various frame structures based on one or more numerologies.

[0046] One or more numerologies may be supported in the wireless communications system 100, and a numerology may include a subcarrier spacing and a cyclic prefix. A first numerology (e.g., μ=0) may be associated with a first subcarrier spacing (e.g., 15 kHz) and a normal cyclic prefix. In some implementations, the first numerology (e.g., μ=0) associated with the first subcarrier spacing (e.g., 15 kHz) may utilize one slot per subframe. A second numerology (e.g., μ=1) may be associated with a second subcarrier spacing (e.g., 30 kHz) and a normal cyclic prefix. A third numerology (e.g., μ=2) may be associated with a third subcarrier spacing (e.g., 60 kHz) and a normal cyclic prefix or an extended cyclic prefix. A fourth numerology (e.g., μ=3) may be associated with a fourth subcarrier spacing (e.g., 120 kHz) and a normal cyclic prefix. A fifth numerology (e.g., μ=4) may be associated with a fifth subcarrier spacing (e.g., 240 kHz) and a normal cyclic prefix.

[0047] A time interval of a resource (e.g., a communication resource) may be organized according to frames (also referred to as radio frames). Each frame may have a duration, for example, a 10 millisecond (ms) duration. In some implementations, each frame may include multiple subframes. For example, each frame may include 10 subframes, and each subframe may have a duration, for example, a 1 ms duration. In

some implementations, each frame may have the same duration. In some implementations, each subframe of a frame may have the same duration.

[0048] Additionally or alternatively, a time interval of a resource (e.g., a communication resource) may be organized according to slots. For example, a subframe may include a number (e.g., quantity) of slots. The number of slots in each subframe may also depend on the one or more numerologies supported in the wireless communications system 100. For instance, the first, second, third, fourth, and fifth numerologies (i.e., μ=0, μ=1, μ=2, μ=3, μ=4) associated with respective subcarrier spacings of 15 kHz, 30 kHz, 60 kHz, 120 kHz, and 240 kHz may utilize a single slot per subframe, two slots per subframe, four slots per subframe, eight slots per subframe, and 16 slots per subframe, respectively.

[0049] Each slot may include a number (e.g., quantity) of symbols (e.g., orthogonal frequency domain multiplexing (OFDM) symbols). In some implementations, the number (e.g., quantity) of slots for a subframe may depend on a numerology. For a normal cyclic prefix, a slot may include 14 symbols. For an extended cyclic prefix (e.g., applicable for 60 kHz subcarrier spacing), a slot may include 12 symbols. The relationship between the number of symbols per slot, the number of slots per subframe, and the number of slots per frame for a normal cyclic prefix and an extended cyclic prefix may depend on a numerology. It should be understood that reference to a first numerology (e.g., μ=0) associated with a first subcarrier spacing (e.g., 15 kHz) may be used interchangeably between subframes and slots.

[0050] In the wireless communications system 100, an electromagnetic (EM) spectrum may be split, based on frequency or wavelength, into various classes, frequency bands, frequency channels, etc. By way of example, the wireless communications system 100 may support one or multiple operating frequency bands, such as frequency range designations FR1 (410 MHz – 7.125 GHz), FR2 (24.25 GHz – 52.6 GHz), FR3 (7.125 GHz – 24.25 GHz), FR4 (52.6 GHz – 114.25 GHz), FR4a or FR4-1 (52.6 GHz – 71 GHz), and FR5 (114.25 GHz – 300 GHz). In some implementations, the NEs 102 and the UEs 104 may perform wireless communications over one or more of the operating frequency bands. In some implementations, FR1 may be used by the NEs 102 and the UEs 104, among other equipment or devices for cellular communications traffic (e.g., control information, data). In some implementations, FR2 may be used by the NEs 102

and the UEs 104, among other equipment or devices for short-range, high data rate capabilities.

[0051] FR1 may be associated with one or multiple numerologies (e.g., at least three numerologies). For example, FR1 may be associated with a first numerology (e.g., μ=0), which includes 15 kHz subcarrier spacing; a second numerology (e.g., μ=1), which includes 30 kHz subcarrier spacing; and a third numerology (e.g., μ=2), which includes 60 kHz subcarrier spacing. FR2 may be associated with one or multiple numerologies (e.g., at least 2 numerologies). For example, FR2 may be associated with a third numerology (e.g., μ=2), which includes 60 kHz subcarrier spacing; and a fourth numerology (e.g., μ=3), which includes 120 kHz subcarrier spacing.

[0052] Wireless communication in unlicensed spectrum (also referred to as "shared spectrum") in contrast to licensed spectrum offer some obvious cost advantages allowing communication to obviate overlaying operator's licensed spectrum and rather use license free spectrum according to local regulation in specific geographies. From the third generation partnership project (3GPP) technology perspective, the unlicensed operation can be on the Uu interface (referred to as NR-U) or also on sidelink interface (e.g., SL-U).

[0053] For initial access, a UE 104 detects a candidate cell and performs downlink (DL) synchronization. For example, the gNB (e.g., an embodiment of the NE 102) may transmit a synchronization signal and physical broadcast channel (SS/PBCH) transmission, referred to as a synchronization signal block (SSB). In various embodiments, the SSB comprises the primary synchronization signal (PSS), the secondary synchronization signal (SSS), and the master information block (MIB). The synchronization signal (i.e., comprising the PSS and SSS) is a predefined data sequence known to the UE 104 (or derivable using information already stored at the UE 104) and is in a predefined location in time relative to frame/subframe boundaries, etc. The UE 104 searches for the SSB and uses the SSB to obtain DL timing information (e.g., symbol timing) for the DL synchronization. The UE 104 may also decode system information (SI) based on the SSB. With beam-based communication, each DL beam may be associated with a respective SSB.

[0054] After performing DL synchronization and acquiring essential system information, such as the MIB and the system information block type 1 (SIB1), the UE 104 performs uplink (UL) synchronization and resource request by performing a random-

access procedure, referred to as "RACH procedure" by selecting and transmitting a preamble on the physical random access channel (PRACH). The PRACH preamble is transmitted during a random access channel (RACH) occasion, i.e., a predetermined set of time-frequency resources that are available for the reception of the PRACH preamble. With beam-based communication, the UE 104 may select a certain DL beam and transmit the PRACH preamble on a corresponding UL beam. In such embodiments, there may be a mapping between SSB and RACH occasion, allowing the network to determine which beam the UE 104 has selected.

[0055] Regarding random access, two types of RACH procedure are supported in a 3GPP wireless communication network: A) a 4-step random-access (RA) type initiated by the sending of a RACH message 1 (Msg1) and 2-step RA type with RACH message A (MsgA). Both types of RACH procedure support contention-based random access (CBRA) and contention-free random access (CFRA).

[0056] The UE 104 selects the RA type at the initiation of the RACH procedure, e.g., based on network configuration. In one example, when CFRA resources are not configured, a reference signal received power (RSRP) threshold is used by the UE 104 to select between 2-step RA type and 4-step RA type. In another example, when CFRA resources for 4-step RA type are configured, the UE 104 performs random access with 4-step RA type. In another example, when CFRA resources for 2-step RA type are configured, the UE 104 performs random access with 2-step RA type.

[0057] The network may not configure CFRA resources for 4-step and 2-step RA types at the same time for a bandwidth part (BWP). Additionally, the CFRA with 2-step RA type is only supported for handover.

[0058] The Msg1 of the 4-step RA type includes a preamble transmitted on a PRACH. After the Msg1 transmission, the UE 104 monitors for a response from the network within a configured window. For CFRA, a dedicated preamble for Msg1 transmission is assigned by the network and upon receiving a random access response (RAR) from the network, the UE 104 ends the random access procedure. For CBRA, upon reception of the RAR, the UE 104 sends a RACH message 3 (Msg3) using a UL grant scheduled in the RAR and monitors for contention resolution. If contention resolution is not successful after Msg3 (re)transmission(s), then the UE 104 goes back to Msg1 transmission.

[0059] The MsgA of the 2-step RA type includes a preamble on the PRACH and a payload on a physical uplink shared channel (PUSCH). After the MsgA transmission, the UE 104 monitors for a response from the network within a configured window. For CFRA, a dedicated preamble and PUSCH resource are configured for MsgA transmission and upon receiving the network response, the UE 104 ends the random access procedure. For CBRA, if contention resolution is successful upon receiving the network response, then the UE 104 ends the random access procedure; however, if a fallback indication is received in a RACH message B (MsgB), the UE 104 performs Msg3 transmission using the UL grant scheduled in the fallback indication and monitors for contention resolution. If contention resolution is not successful after Msg3 (re)transmission(s), the UE 104 goes back to MsgA transmission.

[0060] If the random access procedure with 2-step RA type is not completed after a number of MsgA transmissions, the UE 104 can be configured to switch to CBRA with 4-step RA type.

[0061] In 3GPP NR, the gNB may transmit the maximum 64 SSBs and the maximum 64 corresponding copies of physical downlink control channel (PDCCH) and/or physical downlink shared channel (PDSCH) for delivery of SIB1 in high frequency bands (e.g., 28 GHz). This may cause significant network energy consumption even for a low traffic load condition. According to 3GPP technical report (TR) 38.864 (v18.1.0), for network energy savings, on-demand SSB and/or SIB1 (SSB/SIB1) transmissions and a cell without SSB/SIB1 transmission were considered. When a cell does not transmit SSB/SIB1, for a UE 104 to access the cell, the UE 104 should obtain SI of the cell from other associated carriers/cells and synchronize from other associated carriers/cells. When a cell is in a long period of cell inactivity, a UE 104 served by the cell can trigger SSB/SIB1 transmissions by sending a request to the cell.

[0062] **Figure 2** illustrates an example of a protocol stack 200, in accordance with aspects of the present disclosure. In some embodiments, the protocol stack 200 may be an NR protocol stack used in a 5G NR system. While Figure 2 shows a UE 206, a RAN node 208, and a 5GC 210 (e.g., comprising at least an AMF), these are representative of a set of UEs 104 interacting with an NE 102 (e.g., base station) and a CN 106. As depicted, the protocol stack 200 comprises a user plane protocol stack 202 and a control plane protocol stack 204. The user plane protocol stack 202 includes a physical (PHY) layer 212, a medium access control (MAC) sublayer 214, a radio link control (RLC) sublayer 216, a

packet data convergence protocol (PDCP) sublayer 218, and a service data adaptation protocol (SDAP) sublayer 220. The control plane protocol stack 204 includes a PHY layer 212, a MAC sublayer 214, a RLC sublayer 216, and a PDCP sublayer 218. The control plane protocol stack 204 also includes a radio resource control (RRC) layer 222 and a NAS layer 224.

[0063] The access stratum (AS) layer 226 (also referred to as "AS protocol stack") for the user plane protocol stack 202 includes at least SDAP, PDCP, RLC and MAC sublayers, and the physical layer. The AS layer 228 for the control plane protocol stack 204 includes at least RRC, PDCP, RLC and MAC sublayers, and the physical layer. The layer-1 (L1) includes the PHY layer 212. The layer-2 (L2) is split into the SDAP sublayer 220, PDCP sublayer 218, RLC sublayer 216, and MAC sublayer 214. The layer-3 (L3) includes the RRC layer 222 and the NAS layer 224 for the control plane and includes, e.g., an internet protocol (IP) layer and/or PDU Layer (not depicted) for the user plane. L1 and L2 are referred to as "lower layers," while L3 and above (e.g., transport layer, application layer) are referred to as "higher layers" or "upper layers."

[0064] The PHY layer 212 offers transport channels to the MAC sublayer 214. The PHY layer 212 may perform a beam failure detection procedure using energy detection thresholds, as described herein. In certain embodiments, the PHY layer 212 may send an indication of beam failure to a MAC entity at the MAC sublayer 214. The MAC sublayer 214 offers logical channels to the RLC sublayer 216. The RLC sublayer 216 offers RLC channels to the PDCP sublayer 218. The PDCP sublayer 218 offers radio bearers to the SDAP sublayer 220 and/or RRC layer 222. The SDAP sublayer 220 offers QoS flows to the core network (e.g., 5GC). The RRC layer 222 provides for the addition, modification, and release of carrier aggregation and/or dual connectivity. The RRC layer 222 also manages the establishment, configuration, maintenance, and release of signaling radio bearers (SRBs) and data radio bearers (DRBs).

[0065] The NAS layer 224 is between the UE 206 and an AMF in the 5GC 210. NAS messages are passed transparently through the RAN. The NAS layer 224 is used to manage the establishment of communication sessions and for maintaining continuous communications with the UE 206 as it moves between different cells of the RAN. In contrast, the AS layers 226 and 228 are between the UE 206 and the RAN (i.e., RAN node 208) and carry information over the wireless portion of the network. While not depicted in

Figure 2, the IP layer exists above the NAS layer 224, a transport layer exists above the IP layer, and an application layer exists above the transport layer.

[0066] The MAC sublayer 214 is the lowest sublayer in the L2 architecture of the NR protocol stack. Its connection to the PHY layer 212 below is through transport channels, and the connection to the RLC sublayer 216 above is through logical channels. The MAC sublayer 214 therefore performs multiplexing and demultiplexing between logical channels and transport channels: the MAC sublayer 214 in the transmitting side constructs MAC PDUs (also known as transport blocks (TBs)) from MAC service data units (SDUs) received through logical channels, and the MAC sublayer 214 in the receiving side recovers MAC SDUs from MAC PDUs received through transport channels.

[0067] The MAC sublayer 214 provides a data transfer service for the RLC sublayer 216 through logical channels, which are either control logical channels which carry control data (e.g., RRC signaling) or traffic logical channels which carry user plane data. On the other hand, the data from the MAC sublayer 214 is exchanged with the PHY layer 212 through transport channels, which are classified as UL or DL. Data is multiplexed into transport channels depending on how it is transmitted over the air.

[0068] The PHY layer 212 is responsible for the actual transmission of data and control information via the air interface, i.e., the PHY layer 212 carries all information from the MAC transport channels over the air interface on the transmission side. Some of the important functions performed by the PHY layer 212 include coding and modulation, link adaptation (e.g., adaptive modulation and coding (AMC)), power control, cell search and random access (for initial synchronization and handover purposes) and other measurements (inside the 3GPP system (i.e., NR and/or LTE system) and between systems) for the RRC layer 222. The PHY layer 212 performs transmissions based on transmission parameters, such as the modulation scheme, the coding rate (i.e., the modulation and coding scheme (MCS)), the number of physical resource blocks (PRBs), etc.

[0069] An LTE protocol stack comprises similar structure to the protocol stack 200, with the differences that the LTE protocol stack lacks the SDAP sublayer 220 in the AS layer 226, that an EPC replaces the 5GC 510, and that the NAS layer 224 is between the UE 206 and an MME in the EPC. Also, the present disclosure distinguishes between a protocol layer (such as the aforementioned PHY layer 212, MAC sublayer 214, RLC sublayer 216, PDCP sublayer 218, SDAP sublayer 220, RRC layer 222 and NAS layer 224)

and a transmission layer in multiple-input multiple-output (MIMO) communication (also referred to as a "MIMO layer" or a "data stream").

[0070] eURLLC in 6G is associated with a lower end-to-end latency (e.g., less than 1ms) compared to the 5G NR, and a high level of transmission reliability. Thus, a block error rate (BLER) of less than 10^{-7}eURLLC enables emerging applications, such as future factory applications, tactile internet, distributed utility grid, and metaverse, as well as mission-critical applications, such as telesurgery, autonomous driving and factory automation.

[0071] **Figure 3** illustrates future 5G-advanced and/or 6G use cases and different target requirements. Table 1 describes target requirements of some of these use cases.

Table 1: Examples of ultra-reliable low latency use cases and their target requirements

Scenario	End-to-end latency	Reliability
Discrete automation – motion control	1 ms	99.9999%
Electricity distribution – high voltage	5 ms	99.9999%
Remote control	5 ms	99.999%
Discrete automation	10 ms	99.99%
Intelligent transport systems – infrastructure backhaul	10 ms	99.9999%
Process automation – remote control	50 ms	99.9999%
Process automation – monitoring	50 ms	99.9%
Electricity distribution – medium voltage	25 ms	99.9%

[0072] Polar codes have been the subject of active research in recent times, mainly since they are the first ever provably capacity achieving codes, with explicit construction and low complexity of encoding and decoding. The polar codes relate to a concept called channel polarization. Both the concept of channel polarization as well as polar codes have been extended to several applications and generalizations.

[0073] Consider W: X → Y to denote a generic binary-input, discrete, memoryless channels (B-DMC) with input alphabet X, output alphabet Y, and transition probabilities W (y|x), x ∈ X , y ∈ Y. The input alphabet X will always be {0,1}, the output alphabet and the transition probabilities may be arbitrary. The vector W_N denotes the channel corresponding to N uses of W; thus,

$$W_N: X_N \rightarrow Y_N \text{ with } W_N(y_1^N|x_1^N) = \prod_{i=1}^{N} W(y_i|x_i).$$

[0074] Given a B-DMC W, there are two channel parameters of primary interest, the symmetric capacity $I(W)$ and the Bhattacharyya parameter $Z(W)$, which are defined as follows:

$$I(W) = \sum_{y\in Y}\sum_{x\in X}\frac{1}{2}W(y|x)\log\left(\frac{W(y|x)}{\frac{1}{2}W(y|0)+\frac{1}{2}W(y|1)}\right)$$

$$Z(W) \triangleq \sum_{y\in Y}\sqrt{W(y|0)W(y|1)}$$

[0075] These parameters are used as measures of rate and reliability, respectively. The symmetric capacity parameter *I(W)* is the highest rate at which reliable communication is possible across W using the inputs of W with equal frequency. The Bhattacharyya parameter *Z(W)* is an upper bound on the probability of maximum-likelihood (ML) decision error when W is used only once to transmit a 0 or 1. It is easy to see that *Z(W)* takes values in [0 ,1], whereby a 0 indicates a null probability of error in ML-sense, and respectively, a 1 indicates a certain probability of error in ML-sense.

[0076] Channel polarization is an operation by which one manufactures out of N independent copies of a given B-DMC W, a second set of N channels $\{W_N^{(i)}: 1 \leq i \leq N\}$ that show a polarization effect in the sense that, as N becomes large, the symmetric capacity terms $\{I(W_N^{(i)})\}$ tend towards 0 or 1 for all but a vanishing fraction of indices i. This operation includes a channel combining phase and a channel splitting phase.

[0077] Regarding channel combining, this phase combines copies of a given B-DMC W in a recursive manner to produce a vector channel $W_N: X_N \rightarrow Y_N$, where N can be any power of two, $N = 2^n$, $n \geq 0$. The recursion begins at the 0-th level (n = 0) with only one copy of W, setting $W_1 \triangleq W$.

[0078] **Figure 4** illustrates a block diagram of channel combining, in accordance with aspects of the present disclosure. The first level (n = 1) of the recursion combines two independent copies of W_1 (as shown in Figure 4) and obtains the channel $W_2: X_2 \rightarrow Y_2$ with the transition probabilities:

$$W_2(y_1, y_2|u_1, u_2) = W(y_1|u_1 \oplus u_2)\, W(y_2|\, u_2)$$

[0079] Regarding Channel Splitting, having synthesized the vector channel W_N out of W^N, the next step of channel polarization is to split W_N back into a set of N binary-input

coordinate channels $W_N^{(i)}: X \rightarrow Y^N \times X^{i-1}$, $1 \leq i \leq N$, defined by the transition probabilities:

$$W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) \triangleq \sum_{u_{i+1}^N \in X^{N-i}} \frac{1}{2^{N-1}} W_N(y_1^N|u_1^N)$$

where (y_1^N, u_1^{i-1}) denotes the output of $W_N^{(i)}$ and u_i its input. To gain an intuitive understanding of the channels $\{W_N^{(i)}\}$, consider a genie-aided successive cancellation decoder in which the ith decision element estimates u_i after observing y_1^N and the past channel inputs u_1^{i-1} (supplied correctly by the genie regardless of any decision errors at earlier stages). If u_1^N is a-priori uniform on X^N, then $W_N^{(i)}$ is the effective channel seen by the ith decision element in this scenario.

[0080] **Figure 5** is a graph 500 illustrating the effects of channel polarization. It is a well-known result that for any B-DMC W, the channels $\{W_N^{(i)}\}$ polarize in the sense that, for any fixed $\delta \in (0, 1)$, as N goes to infinity through powers of two, the fraction of indices $i \in \{1, \ldots, N\}$ for which $I(W_N^{(i)}) \in (1 - \delta, 1]$ goes to *I(W)* and the fraction for which $I(W_N^{(i)}) \in [0, \delta)$ goes to $1-I(W)$.

[0081] Let $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, $F^{\otimes n}$ is a $N \times N$ matrix known as the Kernel or the polar base matrix, where $N = 2^n$, $\otimes n$ denotes $n - th$ Kronecker power, and $F^{\otimes n} = F \otimes F^{\otimes(n-1)}$. Let the n-bit binary representation of integer i be $b_{n-1}, b_{n-2}, \ldots b_0$. The n-bit representation $b_0, b_1, \ldots, b_n$ is a bit-reversal order of i. The generator matrix of polar code is defined as $G_N = B_N F^{\otimes n}$, where B_N is a bit-reversal permutation matrix. The polar code is generated by:

$$x_1^N = u_1^N G_N = u_1^N B_N F^{\otimes n}$$

where $x_1^N = (x_1, x_2, \ldots x_N)$ is the encoded bit sequence, and $u_1^N = (u_1, u_2, \ldots u_N)$ is the encoding bit sequence. The bit indexes of u_1^N are divided into two subsets: the one containing the information bits and the other containing the frozen bits. For simplicity, the frozen bits are set "0".

[0082] The main idea of polar codes encoding is the splitting of data sequence indexes into two different sets before transmission. The first set includes the indexes of the data to be transmitted on the noise-free channels. The other set includes the indexes corresponding to the known frozen bits to be transmitted on the pure-noise channel. Over

the last decades, many techniques were introduced to construct polar codes. The traditional technique was based on Bhattacharyya parameter bounds. This technique has the least complexity relative to all other proposed techniques. In addition, Monte-Carlo estimation approach may be used to construct polar codes; however, this approach has higher complexity than the other techniques.

[0083] A density evolution (DE) technique approximates the exact transition probability of each binary input channel to overcome difficulties in calculating the actual values of the Bhattacharyya parameter. A Gaussian approximation (GA) technique constructs polar codes by Trifonov. This technique estimates a bit channel metric inversely proportional to a defined Q-function, which represents its bit error rate (BER) under GA. Mostly, all these techniques are equally good in improving the signal-to-noise ratio (SNR) for additive white Gaussian noise (AWGN) channel.

[0084] Polar codes construction depends on the Bhattacharyya parameter bounds. In this case, first a generalized upper and lower bound of Bhattacharyya parameter are determined. In fact, the upper bound of this parameter corresponds to the noisiest channel, while its lower bound corresponds to the lowest noisy channel. Thus, for better performance, it is required to increase the gap between the Bhattacharyya parameter extremes. This increases the polarization of the synthetic channels carrying the information bits. Then, the most appropriate kernel matrix associated with Bhattacharyya parameter constraints is selected.

[0085] Many techniques are introduced to decode polar codes. Three main techniques considered for decoding are: SC, SCL and log-likelihood ratio (LLR) based SCL. The SC decoding technique is improved to SCL for a finite small length of polar block codes. Additionally, LLR-based SCL decoding technique may be used to decode polar codes.

[0086] As SC decoding is sub-optimal for finite length polar codes, SCL decoding was introduced achieving the ML bound for a sufficiently large list size L, at the cost of increased complexity due to the list decoding nature. Further enhancement of the code was conducted via concatenating a high-rate outer code such as CRC and parity-check (PC) codes. Under SCL decoding, these CRC-aided polar codes and parity-check concatenated polar codes were shown to outperform the state-of-the-art low-density parity check (LDPC) codes. Further, an extension of polar codes, namely Polar Subcodes, has outperformed the above-mentioned code constructions.

[0087] However, the SCL decoder is characterized by a high complexity and an inherently serial decoding nature, which in turn reduces the decoding throughput and causes high decoding latency. In addition, SCL decoding is not a good match to iterative detection and decoding due to its hard decision output nature (i.e., not a soft-in/soft-out decoder). Iterative decoding of polar codes based on message passing over the encoding graph has been possible through belief propagation (BP) decoders.

[0088] The BP decoder algorithm enjoys some fundamental advantages over SC-based decoding, as it can be easily parallelized, thus high throughput/low latency implementations are possible, and it inherently enables soft-in/soft-out decoding, facilitating joint iterative detection and decoding. Thus, BP decoding is a promising candidate for high data rate and low latency demanding applications. A belief propagation list (BPL) decoder with comparable performance to the SCL decoder of polar codes, which already achieves the ML bound of polar codes for sufficiently large list size L, was also proposed.

[0089] Although the SC decoding algorithm seems unsuitable for high-throughput applications due to its serial nature, state-of-the-art SC decoders managed to significantly simplify and parallelize the decoding process such that the area efficiency of SC decoding has far exceeded that of BP decoding for LDPC. In particular, these works represent SC decoding as a breadth-first binary tree traversal, with each subtree therein representing a shorter polar code.

[0090] **Figure 6A** illustrates an example of binary tree traversal in accordance with aspects of the present disclosure. The binary tree search process starts from the root node to the leaf node and from the left branch to the right. At the p-th ($0 \leq p \leq n$) level of the decoding tree, each parent node referred as $\mathcal{N}_i^p$, has a left child node $\mathcal{N}_{2i-1}^{p-1}$ and a right child node $\mathcal{N}_{2i}^{p-1}$, where $1 \leq i \leq 2n - p$.

[0091] **Figure 6B** illustrates an example of messaging associated with the binary tree traversal, in accordance with aspects of the present disclosure. There are two types of messages, i.e., the soft LLRs $\alpha_i^p[1:2p]$ that are propagated from the parent node to their child nodes, and the hard codeword $\beta_i^p[1:2p]$ that is propagated from the child nodes to their parent node in return.

[0092] The original SC decoding algorithm traverses the tree by visiting all the nodes and edges, leading to high decoding latency. Simplified SC decoders can fast decode certain subtrees (shorter polar codes) and thus "prune" those subtrees. The resulting decoding latency is largely determined by the number of remaining edges and nodes in the pruned binary tree.

[0093] The FSSC decoding algorithm can be significantly simplified for some nodes with special information and frozen bit patterns. In particular, four types of special nodes, i.e., Rate-0, Rate-1, REP and SPC, are considered in the FSSC decoder, and their structures are described as follows: Rate-0 → all bits are frozen bits, c = {0, 0, … , 0}; Rate-1 → all bits are information bits, c = {1, 1, …, 1}; REP → all bits are frozen bits except the rightmost one, c = {0, …., 0, 1}; SPC → all bits are information bits except the leftmost one, c = {0, 1, …., 1}.

[0094] Additional special nodes have been identified, including four new fast decoding modules for special nodes with code rates $\left\{\frac{2}{M}, \frac{3}{M}, \frac{(M-3)}{M}, \frac{(M-2)}{M}\right\}$. Here $M = 2^s$ is the number of leaf nodes in a subtree, where s is the stage number. These additional special nodes are called dual-REP (REP-2), repeated parity check (RPC), parity checked repetition (PCR), and dual-SPC (SPC-2) nodes, respectively. Importantly, these modules reuse existing decoding circuits for REP and SPC nodes.

[0095] The structures of these additional special nodes are their structures are described as follows: a node v is defined as a SPC-2 node if V_v includes only two frozen bits, and the frozen bits indices are the two smallest in $i(U_v)$; a node v is defined as a REP-2 node if V_v includes only two information bits, and the information bits indices are the two largest in the $i(U_v)$; a node v is defined as a RPC node if V_v includes only three frozen bits, and the frozen bits indices are the three smallest in the $i(U_v)$; a node v is defined as a PCR node if V_v includes only three information bits, and the information bits indices are the three largest in the $i(U_v)$.

[0096] As used herein, a “frozen bit” refers to a specific bit position in the information block that is predetermined and known to both the transmitter and the receiver. Frozen bits are crucial elements of the polarization process in polar coding, wherein certain bit positions are selectively frozen, meaning their values are fixed and not allowed to change during encoding or decoding. These frozen bits are typically chosen

based on their positions in the binary sequence and their impact on achieving reliable communication over the channel.

[0097] The purpose of freezing certain bits is to ensure that the most reliable channels are utilized for transmitting information (these are referred to as "information bits"), while less reliable channels are effectively "frozen" to minimize errors. By freezing certain bits, polar coding can achieve the capacity of the channel while maintaining low encoding and decoding complexity.

[0098] Moreover, further enhancements to the FSSC decoding speed were achieved by identifying five additional special nodes along with their efficient SC decoders. In addition, some works have also identified a new class of multi-node information and frozen bit patterns, namely SR0/REP node, which includes most of the existing special nodes as special cases.

[0099] **Figure 7** illustrates a generalized structure 700 of a SR0/REP node, in accordance with aspects of the present disclosure. For an SR0/REP node at level p, all its descendants are Rate-0 or REP nodes except the rightmost one at level q, which is a generic source node. A new class of multi-node information and frozen bit patterns is composed of a sequence of Rate-1 or SPC (SR1/SPC) nodes, and thus provides a unified description of a wide variety of existing special nodes. The proposed SR1/SPC node is typically found at higher levels of the decoding tree, thus a higher degree of parallelism can be exploited as compared to the existing special nodes.

[0100] Polar codes can be viewed as monomial codes. In this perspective, each synthetic channel corresponds to a monomial in n binary variables x_i . The set of all monomials in n variables is defined as $\mathcal{M}_n$ and a polar code is a specific subset I, called the information set of the polar code. Every monomial can be written as:

$$f = \prod_{i \in ind(f)} x_i$$

where $ind(f)$is an ordered subset of the variable indices $\Omega = [0, n-1], \{0,1,\ldots,\mathrm{n}-1\}$ and directly corresponds to the ℓ-th row of the generator matrix as:

$$\ell = \sum_{i \in \Omega | ind(f)} 2^i$$

[0101] In other words, the monomial f corresponds to the row whose binary representation has zeros exactly in the bit-positions of the variables contained in f. A message is a polynomial:

$$u(x_0, \ldots, x_{n-1}) = \sum_{f \in I} u_f \, f(x_0, \ldots, x_{n-1})$$

with K coefficients $u_f \in \mathbb{F}_2$. The respective codeword is given by the evaluation of $u(x)$in all N points $x \in \mathbb{F}_2^n$. As a convention, it is assumed that the j-th codeword symbol is obtained from the point x equal to the binary expansion of j.

[0102] A decreasing monomial code is a polar code whose monomial selection obeys the partial order. More precisely, if a synthetic channel is selected as an information channel, all stronger channels with respect to "≤" are also information channels. Mathematically, this can be written as:

$$\forall g \in I;\ \forall f \in \mathcal{M}_n \ with \ f \leq g \ \Rightarrow f \in I$$

[0103] Almost all practical polar code constructions result in decreasing monomial codes. A decreasing monomial code can be fully specified by a minimal information set I_{min} containing only a small number of monomials called generators. All other monomials are implied by the partial order:

$$I = \bigcup_{g \in I_{min}} \{ f \in \mathcal{M}_n; f \leq g \}$$

[0104] The automorphism group $Aut(\mathcal{C})$ of a code $\mathcal{C}$ is the group of codeword symbol permutations, that leave the code unchanged, i.e., map each codeword onto a codeword that is not necessarily different. It was shown that the automorphism group of a decreasing monomial code contains at least $LTA(2, n)$,i.e., affine transformations of the variables x_i in the form $x' = Ax + b$, with $A \in \mathbb{F}_2^{n \times n}$ being a lower triangular matrix with a unit diagonal and arbitrary $b \in \mathbb{F}_2$.

[0105] Stage-shuffling of the polar code factor graph corresponds to a bit-index permutation of both the codeword vector c and the message vector u (including the frozen bits). When viewing such permutations from a monomial code perspective, they exactly correspond to permuting the variables of the monomials x_i from $\mathcal{M}_n$. Depending on the polar code construction (i.e., information/frozen set), there may exist permutations that keep the information set *I* unchanged, i.e., they stabilize it. Such a permutation is related

to that automorphism of the code, where A in $x' = Ax + b$ is the corresponding permutation matrix.

[0106] Sphere decoding is a depth-first tree search, it can find the closest decoded sequence from the received sequence in codeword space under the radius constraint. Similar to ML, sphere decoding (SD) algorithm can solve the problem by enumerating the possible sequence v satisfying the sphere constraint:

$$m(v_1^N) \triangleq \|y - (1 - 2vG)\|^2 \leq r^2$$

[0107] Where r denotes the radius for the SD search and $m(v_1^N)$ is the squared Euclidean distance along with the sequence v_1^N.

[0108] MultiSphere decoding includes a more enhanced SD tree partitioning procedure, which adjusts to the transmission channel while achieving ML performance. The partitioning can take place offline, based on the average channel characteristics, or "on-the-fly," when the transmission channel changes after each QR decomposition (i.e., a decomposition of a matrix A into a product $A = QR$ of an orthonormal matrix Q and an upper triangular matrix R). This adds preprocessing latency to that of the QR decomposition. However, the partitioning latency scales linearly with the number of transmit antennas n_t in contrast to the QR decomposition latency which scales almost cubically with the number of transmit antennae.

[0109] After SD partitioning, MultiSphere applies a new symbol-to-subtree allocation technique which efficiently maps nodes to processing elements (PEs) without introducing dependencies and minimizes the number of redundant calculations across PEs. Each PE performs depth-first subtree traversal with Schnorr-Euchner enumeration, according to which, nodes are visited in ascending order of their partial Euclidean distances (PDs).

[0110] Several approaches have been proposed to avoid exhaustively calculating and sorting the PDs. However, they are not applicable to MultiSphere since their ordering is sequential (to find the kth smallest PD, the $(k-1)\,th$ smallest PDs must be found first, starting from $k = 1$). In addition, a new tree traversal and enumeration procedure is introduced for meeting MultiSphere's needs. MultiSphere runs the parallel SDs in a nearly independent form. They interact only once, after they have all reached the first leaf node. Then, the r^2 of each subtree is replaced by the value of the leaf node with the minimum PD across all parallel SDs. The search is terminated when all parallel trees have

been searched. Then, the detection output is the leaf node with the minimum PD across all subtrees and the overall processing latency is determined by the slowest parallel SD.

[0111] As noted above, one of the bottlenecks for achieving 6G eURLLC and eMBB targeted low latencies and extremely high throughputs (Tbit/s) are polar codes decoding delays and lack of parallelization. Despite the latency and throughput gains of conventional polar code SCL decoders, these conventional techniques are still insufficient to attain 6G KPIs.

[0112] The solutions of the present disclosure describe various techniques, mechanisms and procedures to further enhance and parallelize the FSSC decoder. These techniques, mechanisms and procedures enable the permutation of the received polar codeword and/or the polar factor graph (e.g., an FFG). Permutation of the received codeword enables the reordering of the code into sub-codes with special patterns of information bits and frozen bits known as special nodes (or special kernels). The resulting permuted codeword is then partitioned into independent subtrees, for example according to the MultiSphere techniques. Each of these sub-codes is then fed into of one of the parallel fast component decoders (FSD) that implements modules to fast decode the special nodes without the need to visit all subtree nodes. Accordingly, the techniques, mechanisms and procedures described herein enable high decoding latency and throughput gains without altering the encoding procedure.

[0113] As used herein, a "special node" refers to a specific type of node in the decoding algorithm that facilitates quicker and more efficient computation and reliability of the decoding. The special nodes in polar coding correspond to specific patterns of information and frozen bits that are common or particularly useful. The special nodes exploit the structure of polar codes to optimize the decoding process, thereby achieving low decoding latency while maintaining high decoding performance. A special node may also be referred to as a special kernel.

[0114] As used herein, a "kernel" refers to a subcode that exhibits certain properties. Kernels are typically chosen based on their ability to transform the reliability of the channels, ensuring that the resulting polar code achieves capacity with low encoding and decoding complexity. The kernel represents the building block used in the recursive construction of the polar code.

[0115] The techniques presented in this disclosure may be summarized as follows: a permutation matrix is applied to the received polar code, for example using matrix multiplication or other applicable matrix operation. The permutation matrix allows the re-ordering of the bits of the received polar code into a series of successive special information and frozen bits' patterns. The resulting permuted vector is then partitioned into partitions/ polar sub-codes that are fed into parallel independent fast polar code decoders.

[0116] The permutation matrix may be random or selected from a pre-defined set. The permutation may be applied to the received codeword or the polar code factor graph or a combination thereof. The permutation matrix may also be selected from the automorphism group of polar codes which enables the near-ML error-correction performance to be preserved. This decoder structure allows for high decoding latency and throughput gains.

[0117] In a first solution, the received polar codeword of size M is re-ordered into a series of N_{fast} successive special nodes/kernels that correspond to one the specific information and frozen bit patterns such that rate-0, rate-1, SPC and REP…etc. This re-ordering is based on the design/choice of a permutation matrix. Two different permutation matrices may be respectively applied to the received polar code and to the factor graph (e.g., FFG).

[0118] The decoding enhancements presented herein may be implemented within the UE, or the base station (BS), or any network entity that transmits and receives data over a noisy channel. The channel is assumed to be an AWGN channel.

[0119] According to aspects of the first solution, a permutation matrix π is performed over the received polar codeword y_1^M of length M. This permutation matrix is designed to re-order the received vector into sub-codes, each sub-code presents certain frozen and information bits' patterns and may be fast decoded without searching the binary tree and visiting all its nodes and leaves. At the best case when the received codeword can be fully decomposed into a series of successive special nodes, high decoding latency and throughput gains may be reached and could surpass the performance in terms of latency and throughput of iterative decoding techniques. The permutation matrix is selected from a set of permutation matrices or designed such that the near-ML error-correction performance of CRC-aided SCL decoding is preserved.

[0120] Consider a polar code $\mathcal{P}(M, K)$ of length M with K information bits that is constructed by applying a linear transformation to the message word $u = \{u_0, u_1, \ldots u_{M-1}\}$ as $x = uG^{\otimes n}$ where $x = \{x_0, x_1, \ldots x_{M-1}\}$ is the codeword, $G^{\otimes n}$ is the n-th Kronecker power of the polarizing matrix $G = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$, and $n = log_2(M)$. The vector u contains a set $\mathcal{A}$ of K information bits and a set $\mathcal{F}$ of (M – K) frozen bits.

[0121] The received vector y_1^M of length M after going through a noisy channel may be written as:

$$y_1^M = hx + z$$

where z is an additive white Gaussian noise and h are channel coefficients. At the receiver, the received vector undergoes a permutation procedure and may be written as follows:

$$y' = \pi\, y_1^M$$

[0122] In this case, the received vector y_1^M can then be decomposed into a series of successive polar sub-codes as follows:

$$y_1^M = (y_1^{M/N_{PE}}, y_{\frac{M}{N_{PE}}+1}^{2M/N_{PE}}, \ldots., y_{\frac{(N_{PE}-1)M}{N_{PE}}+1}^{M})$$

where N_{Fast} sub-code components of $y_1^{M/N_{PE}}, y_{\frac{M}{N_{PE}}+1}^{2M/N_{PE}}, \ldots., y_{\frac{(N_{PE}-1)M}{N_{PE}}+1}^{M}$ correspond to one of the special nodes/kernels, e.g., as described above. In the best case, all sub-codes correspond to special nodes/kernels, but this is not always possible. Accordingly, the permutation is designed to maximize the number of special nodes.

[0123] The special nodes/kernels include the rate-0, rate-1, SPC, REP or a combination thereof. The special nodes might also include the REP-2, the SPC-2, the RPC, or the PCR, or a combination thereof. In this case, each of these subcodes may be fast decoded using parallel modules without the need for binary tree traversal techniques. This allows higher decoding latency and high throughput without altering the polar encoding procedure.

[0124] The parameter N_{Fast} may be any integer such as $N_{Fast} \ll M$ and at the optimal case, the received codeword may be fully decomposed into special kernels/nodes each of length $\frac{M}{N_{fast}}$. In such case, high decoding latency and throughput gains of polar

codes may be achieved, which might surpass those of iterative decoding, such as BP or sum-product (SP) decoding for LDPC codes.

[0125] **Figure 8** illustrates an exemplary structure of a decoder 800 for permutation-based fast polar code decoding, in accordance with aspects of the present disclosure. The decoder 800 includes a permutation block 802 that applies at least one permutation matrix π_1 to a received polar codeword, i.e., represented by the vector y_1^M. Additionally, the decoder 800 includes a tree partitioning block 804 that partitions (i.e., decomposes) the permuted polar codeword into a plurality of polar subcodes. The tree partitioning of the received codeword may be performed based on the special characteristics of the polar code Kernel matrix.

[0126] Each polar subcode is fed to one or a plurality of parallel component decoders, here depicted as fast decoder blocks 806, which generate log-likelihood ratios (LLRs) of the respective inputs. In certain embodiments, each parallel component decoder works (in parallel) to independently decode a respective polar subcode. The number of parallel component decoders is represented as N_{PE}. In certain embodiments, the decoding procedure at each of the parallel component decoders is independent of others. The decoder 800 also includes a detection block and bit-reorder block 808, which is common to all component decoders.

[0127] In one embodiment, the permutation matrix may be performed (i.e., applied) over the received vector y_1^M or may be performed for stage-permutation of the FFG of the polar code, or a combination thereof.

[0128] **Figure 9** illustrates an exemplary scenario 900 of stage-shuffling permutation of the FFG of the polar code, in accordance with aspects of the present disclosure. The unshuffled FFG 902 of the polar code comprises multiple stages, denoted as Stage-3, Stage-2, Stage-1, and Stage-0, respectively. In contrast, the shuffled FFG 904 of the polar code comprises the same stages as the unshuffled FFG 902, but in a different order. Thus, in the shuffled FFG 904 the input signals first pass through Stage-0, then Stage-3, then Stage-2 and finally Stage-1.

[0129] At each stage of the FFG of the polar code, pairs of input signals (denoted as d_0 through d_{15}) are combined. As depicted, each stage combines the input signals by connecting a lower branch to an upper branch. In certain embodiments, the signal on the lower branch is multiplexed with the signal on the upper branch. Each successive stage

operates on signals mixed from earlier stages, finally resulting in a set of output signals (denoted as x_0 through x_{15}).

[0130] According to first implementation, two different and uncorrelated permutations π_1 and π_2 may be applied respectively to the received codeword and to the factor graph. In another implementation, a first permutation matrix π_1 is selected and applied to the received codeword, and another permutation matrix π_2 is designed as a function of π_1, such that $\pi_2 = f(\pi_1)$ and is applied to the polar code factor graph at the decoder.

[0131] Regarding stage-permutation of the FFG of the polar code, for a polar code of length M, there are $log_2(M)!$ redundant representations of the polar code factor graph, among which $log_2(M)$ are cyclic permutations. These representations may be constructed by different permutations of the layers (stages) of the FFG.

[0132] In another embodiment, one permutation matrix π_1 may be performed over the received polar codeword and another permutation matrix π_2 may be performed over the factor graph (e.g., FFG). In this case, the permutation π_2 may be a function of π_1 and should belong to the automorphism group $Aut(\mathcal{C})$ of polar codes. The automorphism group of polar codes includes the lower triangular affine group (LTA) and block LTA (BLTA).

[0133] The choice of the permutation matrix π_2 enables the near ML error-correction performance of polar codes to be maintained. Consider the following definition of the permutation matrix π_2:

$$\pi_2 \colon G_1 \to G_2$$
$$: x_1 = G_1 u_1 \to \pi_1(x_1) = \pi_2(G_2) u_1$$
$$\pi_2 = f(\pi_1),\ \pi_2 \epsilon\ Aut(\mathcal{C})$$

[0134] In this case, the layers of the original factor graph are denoted as $\{l_{n-1}, l_{n-2}, \ldots . l_1, l_0\}$. Accordingly, the Hadamard matrix G_1 at the encoder may be written as follows:

$$G_1 = F^{\otimes n} = \prod_{i=0}^{(n-1)} \left(I_{2^{n-1-i}} \otimes F \otimes I_{2^i}\right) = \left(I_{2^{n-1-i}} \otimes F \otimes I_{2^i}\right)\left(I_{2^{n-1-j}} \otimes F \otimes I_{2^j}\right)$$
$$= \left(I_{2^{n-1-j}} \otimes F \otimes I_{2^j}\right)\left(I_{2^{n-1-i}} \otimes F \otimes I_{2^i}\right)$$

which can be easily verified at the $n = log_2(M)$ stages of the encoder.

[0135] Furthermore, define the automorphism group of the polar code as $Aut(\mathcal{C})$, which is the set of permutations of codewords that map the whole code onto itself: $Aut(\mathcal{C}) = \{\pi_1, \pi_2, \ldots \pi_e\}$ such that $\forall\, x \in \mathcal{C} \rightarrow \pi_i(x) \in \mathcal{C}$, where $\pi_i \in Aut(\mathcal{C})$. The stage-shuffling permutation should be selected from the automorphism group $Aut(\mathcal{C})$ in order to allow good error-correction performance of permuted polar codeword. In this case, the stage-shuffling permutation might impact the order of the codeword and information vector bits.

[0136] **Figure 10** illustrates an exemplary decoder 1000 for permutation-based fast polar code decoding, in accordance with aspects of the present disclosure. The decoder 1000 receives a vector y_1^M includes a first permutation block 1002 that applies a first permutation matrix π_1 to the vector y_1^M. Additionally, the decoder 1000 includes a second permutation block 1004 that applies a second permutation matrix π_2 to an output of the first permutation block (i.e., to the permutation vector $\pi_1 y_1^M$), as described in greater detail below. The output of the second permutation block 1004 is fed back to the first permutation block 1002. Additionally, an output of the first permutation block 1002 is output to a tree partitioning block 1006 to complete the decoding, as described herein. In certain embodiments, the tree partitioning block 1006 outputs a plurality of polar subcodes may then be fed to one or more parallel modules or fast component decoders (FCDs).

[0137] In a second embodiment, the polar code may be constructed such that any random permutation matrices applied to the received codeword and/or the factor graph does not impact the error-correction performance of polar code. Since the polar code is constructed to optimize its performance assuming that the decoder uses standard successive cancellation decoder (SCD) structure, it may not be fair to expect the same performance on a new sub-optimal decoder, for which the code is not matched. Thus, the design of a matching code construction might allow the near ML error-correction performance of the permuted SCD.

[0138] In a first implementation of the second embodiment, the matched code construction may perform the same estimations of the Bhattacharyya bounds of the bit channels, but in a different order. By similarly freezing the least reliable bits in such a new order, a similar BLER performance may be obtained, whenever these bounds are

tight. More importantly, based on this invariance of bounds to permutation, the permuted received codeword (and/or polar factor graph) with the matched code construction continues to be capacity achieving.

[0139] In a third embodiment, the code construction may optimize the lower bound for the error correction performance. In this case, the set, S_π, of permutation matrices for the fixed frozen bits set $\mathcal{F}$ may be optimized as follows. Firstly, the bit error probability is calculated for each synthetic subchannel. Then, the block error probability is calculated for each layer's permutation matrix. Finally, the layers' permutation matrices are sorted by the corresponding block error probability in ascending order, and $\mathcal{L}$ permutation matrices with the lowest block error probability are selected and the set of permutation matrices is defined accordingly as $S_\pi = \{\pi_1, \pi_2, \ldots \pi_{\mathcal{L}}\}$. In this case, the design of the set of permutation matrices is matched to the polar code construction. This set S_π may be signaled to the decoder and the decoder selects the permutation that enables the best error-correction performance.

[0140] In another embodiment, the permutation matrices might be pre-defined or selected from a pre-defined permutation set S_π. In this case, the polar code encoding procedure may be performed according to legacy 5G NR polar encoding, e.g., as described above. In various embodiments, the error correction performance of polar codes is preserved by choosing a permutation matrix that belongs to the automorphism group LTA (GLTA).

[0141] In a first implementation, two permutation matrices may be selected (i.e., sampled) from the pre-defined set. In such embodiments, the first permutation matrix allows the decomposition of received codeword into a number N_{fast} sub-codes that correspond to special nodes/kernels and remaining $(M - (Total_{bits} * N_{fast}))$ bits, and the other permutation matrix is performed over the FFG of polar code and allows the preservation of error-correction performance by adapting the decoding procedure to the encoding. In another implementation, two different sets of permutation matrices may be defined, i.e., S_{π_1} and S_{π_2}, wherein the first permutation matrix π_1 performed over the received codeword is selected from set S_{π_1} and the second permutation matrix π_2 performed over the polar code factor graph is selected from set S_{π_2}.

[0142] In an alternate implementation, the permutation matrix π_1 applied to the received codeword may be chosen randomly to allow the construction of maximum

number of special kernels/nodes or to transform the received codeword into a series of successive special kernels/nodes. In this case, the permutation matrix may be dynamically determined (e.g., determined 'on the fly') given the received codeword y_1^M. The dynamic design/determination of the permutation matrix may be done in an exhaustive greedy manner, referring to an optimization strategy that considers all possible choices (i.e., "exhaustive") while making locally optimal decisions at each step (i.e., "greedy") to design or construct the permutation matrix. In such case, the algorithm applies several random permutations to the received codeword and the best permutation, in terms of maximizing the number of special kernels/nodes within the received polar code, may be maintained. This allows a faster SCL decoding of the polar code at the expense of more computational complexity.

[0143] The permutation matrix design may be, in another implementation, in a best effort manner. In this case, the algorithm starts from rightmost to leftmost (or leftmost to rightmost) bits and constructs the special kernels/nodes one-by-one by re-ordering the codeword bits until no further special kernels/nodes may be constructed. The codeword bits that could not fit into any of the special kernels/nodes, may be decoded using legacy binary tree traversal techniques (known as SCL/SC decoding) or any other decoder, e.g., BP decoder or list-BP decoder. In this case, the overall decoding latency would depend on the latency of this SCL decoder, or SC decoder, or BP decoder that should search the remaining subtree.

[0144] **Figure 11** illustrates an exemplary structure of a decoder 1100 for polar code decoding using multiple permutations from the automorphism group, in accordance with aspects of the present disclosure. The decoder 1100 selects multiple permutation matrices $\{\pi_1, \pi_2, \ldots . \pi_P\}$ from the pre-defined set S_π, corresponding to the automorphism group. In a first stage 1102, the selected permutation matrices are separately applied to received codeword y_1^M.

[0145] In a second stage 1104, each of the outputs of the permutation matrices are fed into a series of parallel fast decoders, such as FSSC decoders, or fast simplified SCL, or fast simplified BP decoders. In a third stage 1106, an inverse permutation operation is performed on the outputs of the parallel fast decoders.

[0146] During the first stage 1102 (e.g., a permutation stage, in which the permutation matrices are applied to received codeword), bit interleaving may occur (i.e., from the

permutation matrices). Accordingly, the third stage 1106 may perform bit de-interleaving on the decoded bits, e.g., based on an inverse of the respective permutation matrices.

[0147] In a fourth stage 1108, the chosen decoded bits should have the highest likelihood with the received codeword. In one embodiment, the decoded bits are chosen based on metrics. In another embodiment, the decoded bits are chosen based on a maximizing the likelihood and/or validates the CRC checksum. This may impact the decoding latency of the decoder; however, it allows enhanced error-correction performance.

[0148] After undergoing the permutation stage, each of these sub-polar codes (i.e., subtrees) is fed into one of the parallel modules or FCDs which in the latter case perform a tree traversal search procedure over the polar sub-code and in the former case fast decode the polar sub-code without searching the binary tree. This enables higher gains in terms of latency and throughput without altering the error correction performance of polar codes and the hardware implementation complexity of the encoder and decoder.

[0149] According to a further embodiment of the first solution, the tree partitioning may be performed according to the tree partitioning procedure used for MultiSphere-based SD, as described above. In this case, the tree partitioning scheme adjusts to the transmission channel or its statistics instead of leveraging the Kernel matrix properties. MultiSphere runs the parallel component decoders (CDs) in a nearly independent form. They interact only once, after they all have reached the first leaf node.

[0150] The subtree construction for MultiSphere-based SD may be summarized as including a seed identification phase and a MultiSphere subtree construction phase.

[0151] During the seed identification phase, the decoder determines the N_{PE} most promising paths given the received polar codeword. This determination is based on a metric $\mathcal{M}_{PA}$ which is function of the distances between the transmitted symbols. These metrics $\mathcal{M}_{PA}$ characterize each of the paths, and the paths that have the smallest metrics are designated. The determination of the metrics $\mathcal{M}_{PA}$ may be performed offline based on channel statistics or 'on the fly' anytime the channel changes.

[0152] During the MultiSphere subtree construction phase, after calculating the N_{PE} seeds with metrics $\mathcal{M}_{PA}$ (i = 1, ..., N_{PE}), each seed is used to construct a corresponding subtree T_i so that the union of all subtrees forms the original polar code

tree. The process may be designed so that PEs can independently construct their subtrees in parallel in order to minimize the latency of the procedure.

[0153] In one embodiment, the path metric $\mathcal{M}_{PA}$ used to for MultiSphere subtree construction may be determined recursively thanks to the Plotkin construction of polar codes. In this case:

$$\mathcal{M}_{PA}^{l} = \mathcal{M}_{PA}^{l+1} + \mathcal{O}$$

where $\mathcal{O}$ is the path cost between the level l and the level $(l+1)$.

[0154] The optimum path metric $\mathcal{M}_{PA}$ should maximize the likelihood probability:

$$\mathcal{M}_{PA} = \sum_{all\ d_1^{(i-1)}} P\left(y\middle|x_i^M x_1^{(i-1)}\right) P\left(x_1^{(i-1)} = d_1^{(i-1)}\right)$$

where $d_1^{(i-1)} = (d_1, d_2, \ldots, d_{(i-1)})$ is a random vector from where $x_1^{(i-1)}$ may take possible value. $d_j, j = 1,2 \ldots, (i-1)$ is either $+1$ or -1 with equal probability of $1/2$.

[0155] The path cost $\mathcal{O}$ may be expressed as: $\mathcal{O} = (\frac{E}{N_0})^2 |x_l^{(c)} - x_l|^2$, where $x_l^{(c)}$ being the symbol at level l which is closest to the received point. This metric is not a function of the received symbols but a function of the ordered distance between transmitted symbols which may be known in advance and can be pre-calculated.

[0156] These distances may be, according to the first implementation, Hamming distances. In this case, the minimum weight distribution (MWD) of the polar code may be determined and enumerated using different techniques, for example the SD-like procedure or by transmitting an all zero vector.

[0157] The search is terminated when all parallel trees have been searched. Then, the detection output is the leaf node with the minimum metric $\mathcal{M}_{PA}$ across all subtrees.

[0158] Regarding the decoding latency gains, the decoding latency may be N_{PE} times faster than legacy LLR-based SCL decoders without added complexity and hardware implementation. The SCL decoder was proved to complete decoding in $(2M + K - 2)$ timesteps, where K is the number of information bits. The tree partitioning algorithm presented above reduces the decoding latency by an N_{PE} factor, which makes the decoding process completes in $\frac{(2M+K-2)}{N_{PE}}$ timesteps. In this case, the larger the number of

parallel CDs and partitioned subtrees, the higher decoding latency gains can be achieved. This technique allows high throughput gains as well.

[0159] **Figure 12** illustrates an example of a UE 1200 in accordance with aspects of the present disclosure. The UE 1200 may include a processor 1202, a memory 1204, a controller 1206, and a transceiver 1208. The processor 1202, the memory 1204, the controller 1206, or the transceiver 1208, or various combinations thereof or various components thereof may be examples of means for performing various aspects of the present disclosure as described herein. These components may be coupled (e.g., operatively, communicatively, functionally, electronically, electrically) via one or more interfaces.

[0160] The processor 1202, the memory 1204, the controller 1206, or the transceiver 1208, or various combinations or components thereof may be implemented in hardware (e.g., circuitry). The hardware may include a processor, a digital signal processor (DSP), an application-specific integrated circuit (ASIC), or other programmable logic device, or any combination thereof configured as or otherwise supporting a means for performing the functions described in the present disclosure.

[0161] The processor 1202 may include an intelligent hardware device (e.g., a general-purpose processor, a DSP, a central processing unit (CPU), an ASIC, a field programmable gate array (FPGA), or any combination thereof). In some implementations, the processor 1202 may be configured to operate the memory 1204. In some other implementations, the memory 1204 may be integrated into the processor 1202. The processor 1202 may be configured to execute computer-readable instructions stored in the memory 1204 to cause the UE 1200 to perform various functions of the present disclosure.

[0162] The memory 1204 may include volatile or non-volatile memory. The memory 1204 may store computer-readable, computer-executable code including instructions that, when executed by the processor 1202, cause the UE 1200 to perform various functions described herein. The code may be stored in a non-transitory computer-readable medium such the memory 1204 or another type of memory. Computer-readable media includes both non-transitory computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A non-

transitory storage medium may be any available medium that may be accessed by a general-purpose or special-purpose computer.

[0163] In some implementations, the processor 1202 and the memory 1204 coupled with the processor 1202 may be configured to cause the UE 1200 to perform various functions (e.g., operations, signaling) described herein (e.g., executing, by the processor 1202, instructions stored in the memory 1204). In some implementations, the processor 1202 may include multiple processors and the memory 1204 may include multiple memories. One or more of the multiple processors may be coupled with one or more of the multiple memories, which may be individually or collectively, configured to perform various functions (e.g., operations, signaling) of the UE 1200 as disclosed herein.

[0164] For example, the processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to receive a polar codeword comprising a set of bits. The processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to generate a permuted vector based at least in part on the polar codeword and a permutation matrix.

[0165] The processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to determine a plurality of polar subcodes based on the permuted vector. The processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to identify, for each polar subcode of the plurality of polar subcodes, a set of kernels (e.g., special kernels), where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns.

[0166] The processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to decode the plurality of polar subcodes using a plurality of parallel polar code component decoders. In such embodiments, each parallel polar code component decoder includes one or more fast decode modules to decode a corresponding kernel of the set of kernels.

[0167] In some embodiments, to generate the permutation vector, the processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to apply the permutation matrix to the received polar codeword or to a FFG of the polar codeword, or a combination thereof.

[0168] In some embodiments, the permutation matrix is selected from a pre-defined permutation set comprising an automorphism group of the polar codeword. In some embodiments, the permutation matrix may be selected from a set of permutation matrices that match an encoding procedure associated with the polar codeword.

[0169] In some embodiments, the processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to receive a set of permutation matrices. In such embodiments, the permutation matrix is selected from the set of permutation matrices based at least in part on an error-correction performance.

[0170] In some embodiments, the processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to: A) select a plurality of permutation matrices; B) generate a plurality of permuted vectors based on the plurality of permutation matrices; and C) input the plurality of permuted vectors to a plurality of parallel sets of the polar code component decoders.

[0171] In certain embodiments, the processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 may be further configured to: 1) select the plurality of permutation matrices from different pre-defined permutation sets (e.g., sets of permutation matrices); 2) apply a first permutation matrix to the polar codeword; and 3) apply a second permutation matrix to a factor graph associated with the polar codeword, where the second permutation matrix is a function of the first permutation matrix.

[0172] In some embodiments, a first error-correction performance associated with the permuted vector is within a predetermined variation of a second error-correction performance associated with the polar codeword. In some embodiments, the permutation matrix is designed to re-order the set of bits into a maximum number of successive special kernels.

[0173] In some embodiments, the set of kernels corresponds to a set of special nodes associated with a decoding tree, and the set of one or more bit patterns corresponds to special bit patterns of information bits or frozen bits, or both.

[0174] In certain embodiments, the set of one or more bit patterns comprises one or more of: A) a Rate-0 node having only frozen bits, a Rate-1 node having only information bits; B) a REP node, where a rightmost bit is an information bit and a remainder of the

bits are frozen bits; C) a SPC node, where a leftmost bit is a frozen bit and the remainder of the bits are information bits; D) a REP-2 node, where the node includes only two information bits, and bit indices of the two information bits correspond to two largest indices; E) a RPC node, where the node includes only three frozen bits, and bit indices of the three frozen bits correspond to three smallest indices; F) a PCR node, where the node includes only three information bits, and the bit indices of the information bits correspond to three largest indices; G) a SPC-2 node, where the node includes only two frozen bits, and bit indices of the two frozen bits correspond to two smallest indices, or a combination thereof.

[0175] In some embodiments, the processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to: A) determine a code tree based on the permuted vector; and B) partition the code tree into the plurality of polar subcodes, where each polar subcode of the plurality of polar subcodes corresponds to one of a plurality of subtrees. In certain embodiments, the plurality of parallel polar code component decoders implements the one or more fast decode modules at intermediate levels of the plurality of subtrees to fast decode the set of kernels.

[0176] In certain embodiments, the processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 may be further configured to: 1) prune a respective subtree based on a sphere decoding procedure; and 2) perform a simplified subtree traversal of the pruned respective subtree based on a successive cancellation decoder or a successive cancellation list decoder.

[0177] In certain embodiments, the processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to: 1) determine a path metric based on a maximized likelihood probability between a received vector and a random vector; and 2) partition the code tree based on the path metric and a MultiSphere subtree construction procedure.

[0178] In further embodiments, the MultiSphere subtree construction procedure is based on a minimum weight distributions of candidate codewords. In such embodiments, the processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to partition the code tree based at least in part on a candidate codewords associated with a lowest path metric.

[0179] In some embodiments, the processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to decode the polar codeword based on a combined output of the plurality of parallel polar code component decoders. In certain embodiments, the processor 1202 coupled with the memory 1204 may be configured to, capable of, or operable to cause the UE 1200 to decode the polar codeword based on a combined log-likelihood ratio associated with the plurality of polar subcodes.

[0180] In some embodiments, the plurality of parallel polar code component decoders comprises one or more of: 1) SC decoders, 2) SCL decoders, 3) list sphere decoders (List-SDs), 4) ML decoders, or a combination thereof.

[0181] The controller 1206 may manage input and output signals for the UE 1200. The controller 1206 may also manage peripherals not integrated into the UE 1200. In some implementations, the controller 1206 may utilize an operating system (OS) such as iOS®, ANDROID®, WINDOWS®, or other operating systems (OSes). In some implementations, the controller 1206 may be implemented as part of the processor 1202.

[0182] In some implementations, the UE 1200 may include at least one transceiver 1208. In some other implementations, the UE 1200 may have more than one transceiver 1208. The transceiver 1208 may represent a wireless transceiver. The transceiver 1208 may include one or more receiver chains 1210, one or more transmitter chains 1212, or a combination thereof.

[0183] A receiver chain 1210 may be configured to receive signals (e.g., control information, data, packets) over a wireless medium. For example, the receiver chain 1210 may include one or more antennas for receiving the signal over the air or wireless medium. The receiver chain 1210 may include at least one amplifier (e.g., a low-noise amplifier (LNA)) configured to amplify the received signal. The receiver chain 1210 may include at least one demodulator configured to demodulate the received signal and obtain the transmitted data by reversing the modulation technique applied during transmission of the signal. The receiver chain 1210 may include at least one decoder for decoding/ processing the demodulated signal to receive the transmitted data.

[0184] A transmitter chain 1212 may be configured to generate and transmit signals (e.g., control information, data, packets). The transmitter chain 1212 may include at least one modulator for modulating data onto a carrier signal, preparing the signal for

transmission over a wireless medium. The at least one modulator may be configured to support one or more techniques such as amplitude modulation (AM), frequency modulation (FM), or digital modulation schemes like phase-shift keying (PSK) or quadrature amplitude modulation (QAM). The transmitter chain 1212 may also include at least one power amplifier configured to amplify the modulated signal to an appropriate power level suitable for transmission over the wireless medium. The transmitter chain 1212 may also include one or more antennas for transmitting the amplified signal into the air or wireless medium.

[0185] **Figure 13** illustrates an example of a processor 1300 in accordance with aspects of the present disclosure. The processor 1300 may be an example of a processor configured to perform various operations in accordance with examples as described herein. The processor 1300 may include a controller 1302 configured to perform various operations in accordance with examples as described herein. The processor 1300 may optionally include at least one memory 1304, which may be, for example, an L1/L2/L3 cache. Additionally, or alternatively, the processor 1300 may optionally include one or more arithmetic-logic units (ALUs) 1306. One or more of these components may be in electronic communication or otherwise coupled (e.g., operatively, communicatively, functionally, electronically, electrically) via one or more interfaces (e.g., buses).

[0186] The processor 1300 may be a processor chipset and include a protocol stack (e.g., a software stack) executed by the processor chipset to perform various operations (e.g., receiving, obtaining, retrieving, transmitting, outputting, forwarding, storing, determining, identifying, accessing, writing, reading) in accordance with examples as described herein. The processor chipset may include one or more cores, one or more caches (e.g., memory local to or included in the processor chipset (e.g., the processor 1300) or other memory (e.g., random access memory (RAM), read-only memory (ROM), dynamic RAM (DRAM), synchronous dynamic RAM (SDRAM), static RAM (SRAM), ferroelectric RAM (FeRAM), magnetic RAM (MRAM), resistive RAM (RRAM), flash memory, phase change memory (PCM), and others).

[0187] The controller 1302 may be configured to manage and coordinate various operations (e.g., signaling, receiving, obtaining, retrieving, transmitting, outputting, forwarding, storing, determining, identifying, accessing, writing, reading) of the processor 1300 to cause the processor 1300 to support various operations in accordance with examples as described herein. For example, the controller 1302 may operate as a

control unit of the processor 1300, generating control signals that manage the operation of various components of the processor 1300. These control signals include enabling or disabling functional units, selecting data paths, initiating memory access, and coordinating timing of operations.

[0188] The controller 1302 may be configured to fetch (e.g., obtain, retrieve, receive) instructions from the memory 1304 and determine subsequent instruction(s) to be executed to cause the processor 1300 to support various operations in accordance with examples as described herein. The controller 1302 may be configured to track memory address of instructions associated with the memory 1304. The controller 1302 may be configured to decode instructions to determine the operation to be performed and the operands involved. For example, the controller 1302 may be configured to interpret the instruction and determine control signals to be output to other components of the processor 1300 to cause the processor 1300 to support various operations in accordance with examples as described herein. Additionally, or alternatively, the controller 1302 may be configured to manage flow of data within the processor 1300. The controller 1302 may be configured to control transfer of data between registers, arithmetic logic units (ALUs), and other functional units of the processor 1300.

[0189] The memory 1304 may include one or more caches (e.g., memory local to or included in the processor 1300 or other memory, such RAM, ROM, DRAM, SDRAM, SRAM, MRAM, flash memory, etc. In some implementations, the memory 1304 may reside within or on a processor chipset (e.g., local to the processor 1300). In some other implementations, the memory 1304 may reside external to the processor chipset (e.g., remote to the processor 1300).

[0190] The memory 1304 may store computer-readable, computer-executable code including instructions that, when executed by the processor 1300, cause the processor 1300 to perform various functions described herein. The code may be stored in a non-transitory computer-readable medium such as system memory or another type of memory. The controller 1302 and/or the processor 1300 may be configured to execute computer-readable instructions stored in the memory 1304 to cause the processor 1300 to perform various functions. For example, the processor 1300 and/or the controller 1302 may be coupled with or to the memory 1304, the processor 1300, the controller 1302, and the memory 1304 may be configured to perform various functions described herein. In some examples, the processor 1300 may include multiple processors and the memory 1304 may

include multiple memories. One or more of the multiple processors may be coupled with one or more of the multiple memories, which may, individually or collectively, be configured to perform various functions herein.

[0191] The one or more ALUs 1306 may be configured to support various operations in accordance with examples as described herein. In some implementations, the one or more ALUs 1306 may reside within or on a processor chipset (e.g., the processor 1300). In some other implementations, the one or more ALUs 1306 may reside external to the processor chipset (e.g., the processor 1300). One or more ALUs 1306 may perform one or more computations such as addition, subtraction, multiplication, and division on data. For example, one or more ALUs 1306 may receive input operands and an operation code, which determines an operation to be executed. One or more ALUs 1306 be configured with a variety of logical and arithmetic circuits, including adders, subtractors, shifters, and logic gates, to process and manipulate the data according to the operation. Additionally, or alternatively, the one or more ALUs 1306 may support logical operations such as AND, OR, exclusive-OR (XOR), not-OR (NOR), and not-AND (NAND), enabling the one or more ALUs 1306 to handle conditional operations, comparisons, and bitwise operations.

[0192] In various implementations, the processor 1300 may support various functions (e.g., operations, signaling) of a UE, in accordance with examples as disclosed herein. For example, the controller 1302 coupled with the memory 1304 may be configured to, capable of, or operable to cause the processor 1300 to receive, at a UE, a polar codeword comprising a set of bits; generate a permuted vector based at least in part on the polar codeword and a permutation matrix; determine a plurality of polar subcodes based on the permuted vector; identify, for each polar subcode of the plurality of polar subcodes, a set of kernels (e.g., special kernels), where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and decode the plurality of polar subcodes using a plurality of parallel polar code component decoders. In such embodiments, each parallel polar code component decoder may include one or more fast decode modules to decode a corresponding kernel of the set of kernels. Additionally, the controller 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the processor 1400 to perform one or more functions (e.g., operations, signaling) of the UE as described herein.

[0193] In various implementations, the processor 1300 may support various functions (e.g., operations, signaling) of a base station (e.g., gNB), in accordance with examples as disclosed herein. For example, the controller 1302 coupled with the memory 1304 may be configured to, capable of, or operable to cause the processor 1300 to receive, at a base station, a polar codeword comprising a set of bits; generate a permuted vector based at least in part on the polar codeword and a permutation matrix; determine a plurality of polar subcodes based on the permuted vector; identify, for each polar subcode of the plurality of polar subcodes, a set of kernels (e.g., special kernels), where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and decode the plurality of polar subcodes using a plurality of parallel polar code component decoders. In such embodiments, each parallel polar code component decoder may include one or more fast decode modules to decode a corresponding kernel of the set of kernels. Additionally, the controller 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the processor 1400 to perform one or more functions (e.g., operations, signaling) of the base station as described herein.

[0194] **Figure 14** illustrates an example of a NE 1400 in accordance with aspects of the present disclosure. The NE 1400 may include a processor 1402, a memory 1404, a controller 1406, and a transceiver 1408. The processor 1402, the memory 1404, the controller 1406, or the transceiver 1408, or various combinations thereof or various components thereof may be examples of means for performing various aspects of the present disclosure as described herein. These components may be coupled (e.g., operatively, communicatively, functionally, electronically, electrically) via one or more interfaces.

[0195] The processor 1402, the memory 1404, the controller 1406, or the transceiver 1408, or various combinations or components thereof may be implemented in hardware (e.g., circuitry). The hardware may include a processor, a DSP, an ASIC, or other programmable logic device, or any combination thereof configured as or otherwise supporting a means for performing the functions described in the present disclosure.

[0196] The processor 1402 may include an intelligent hardware device (e.g., a general-purpose processor, a DSP, a CPU, an ASIC, an FPGA, or any combination thereof). In some implementations, the processor 1402 may be configured to operate the memory 1404. In some other implementations, the memory 1404 may be integrated into the processor 1402. The processor 1402 may be configured to execute computer-readable

instructions stored in the memory 1404 to cause the NE 1400 to perform various functions of the present disclosure.

[0197] The memory 1404 may include volatile or non-volatile memory. The memory 1404 may store computer-readable, computer-executable code including instructions when executed by the processor 1402 cause the NE 1400 to perform various functions described herein. The code may be stored in a non-transitory computer-readable medium such the memory 1404 or another type of memory. Computer-readable media includes both non-transitory computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A non-transitory storage medium may be any available medium that may be accessed by a general-purpose or special-purpose computer.

[0198] In some implementations, the processor 1402 and the memory 1404 coupled with the processor 1402 may be configured to cause the NE 1400 to perform various functions (e.g., operations, signaling) described herein (e.g., executing, by the processor 1402, instructions stored in the memory 1404). In some implementations, the processor 1402 may include multiple processors and the memory 1404 may include multiple memories. One or more of the multiple processors may be coupled with one or more of the multiple memories, which may be individually or collectively, configured to perform various functions (e.g., operations, signaling) of the NE 1400 as disclosed herein.

[0199] For example, the processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to receive a polar codeword comprising a set of bits. The processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to generate a permuted vector based at least in part on the polar codeword and a permutation matrix.

[0200] The processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to determine a plurality of polar subcodes based on the permuted vector. The processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to identify, for each polar subcode of the plurality of polar subcodes, a set of kernels (e.g., special kernels), where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns.

[0201] The processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to decode the plurality of polar subcodes using a plurality of parallel polar code component decoders. In such embodiments, each parallel polar code component decoder includes one or more fast decode modules to decode a corresponding kernel of the set of kernels.

[0202] In some embodiments, to generate the permutation vector, the processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to apply the permutation matrix to the received polar codeword or to a FFG of the polar codeword, or a combination thereof.

[0203] In some embodiments, the permutation matrix is selected from a pre-defined permutation set comprising an automorphism group of the polar codeword. In some embodiments, the permutation matrix may be selected from a set of permutation matrices that match an encoding procedure associated with the polar codeword.

[0204] In some embodiments, the processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to receive a set of permutation matrices. In such embodiments, the permutation matrix is selected from the set of permutation matrices based at least in part on an error-correction performance.

[0205] In some embodiments, the processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to: A) select a plurality of permutation matrices; B) generate a plurality of permuted vectors based on the plurality of permutation matrices; and C) input the plurality of permuted vectors to a plurality of parallel sets of the polar code component decoders.

[0206] In certain embodiments, the processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 may be further configured to: 1) select the plurality of permutation matrices from different pre-defined sets of permutation matrices; 2) apply a first permutation matrix to the polar codeword; and 3) apply a second permutation matrix to a factor graph associated with the polar codeword, where the second permutation matrix is a function of the first permutation matrix.

[0207] In some embodiments, a first error-correction performance associated with the permuted vector is within a predetermined variation of a second error-correction

performance associated with the polar codeword. In some embodiments, the permutation matrix is designed to re-order the set of bits into a maximum number of successive special kernels.

[0208] In some embodiments, the set of kernels corresponds to a set of special nodes associated with a decoding tree, and the set of one or more bit patterns corresponds to special bit patterns of information bits or frozen bits, or both.

[0209] In certain embodiments, the set of one or more bit patterns comprises one or more of: A) a Rate-0 node having only frozen bits, a Rate-1 node having only information bits; B) a REP node, where a rightmost bit is an information bit and a remainder of the bits are frozen bits; C) a SPC node, where a leftmost bit is a frozen bit and the remainder of the bits are information bits; D) a REP-2 node, where the node includes only two information bits, and bit indices of the two information bits correspond to two largest indices; E) a RPC node, where the node includes only three frozen bits, and bit indices of the three frozen bits correspond to three smallest indices; F) a PCR node, where the node includes only three information bits, and the bit indices of the information bits correspond to three largest indices; G) a SPC-2 node, where the node includes only two frozen bits, and bit indices of the two frozen bits correspond to two smallest indices, or a combination thereof.

[0210] In some embodiments, the processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to: A) determine a code tree based on the permuted vector; and B) partition the code tree into the plurality of polar subcodes, where each polar subcode of the plurality of polar subcodes corresponds to one of a plurality of subtrees. In certain embodiments, the plurality of parallel polar code component decoders each implement the one or more fast decode modules at intermediate levels of the plurality of subtrees to fast decode the set of kernels.

[0211] In certain embodiments, the processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 may be further configured to: 1) prune a respective subtree based on a sphere decoding procedure; and 2) perform a simplified subtree traversal of the pruned respective subtree based on a successive cancellation decoder or a successive cancellation list decoder.

[0212] In certain embodiments, the processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to: 1) determine a

path metric based on a maximized likelihood probability between a received vector and a random vector; and 2) partition the code tree based on the path metric and a MultiSphere subtree construction procedure.

[0213] In further embodiments, the MultiSphere subtree construction procedure is based on a minimum weight distributions of candidate codewords. In such embodiments, the processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to partition the code tree based at least in part on a candidate codewords associated with a lowest path metric.

[0214] In some embodiments, the processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to decode the polar codeword based on a combined output of the plurality of parallel polar code component decoders. In certain embodiments, the processor 1402 coupled with the memory 1404 may be configured to, capable of, or operable to cause the NE 1400 to decode the polar codeword based on a combined log-likelihood ratio associated with the plurality of polar subcodes.

[0215] In some embodiments, the plurality of parallel polar code component decoders comprises one or more of: 1) SC decoders, 2) SCL decoders, 3) List-SDs, 4) ML decoders, or a combination thereof.

[0216] The controller 1406 may manage input and output signals for the NE 1400. The controller 1406 may also manage peripherals not integrated into the NE 1400. In some implementations, the controller 1406 may utilize an OS such as iOS®, ANDROID®, WINDOWS®, or other OSes. In some implementations, the controller 1406 may be implemented as part of the processor 1402.

[0217] In some implementations, the NE 1400 may include at least one transceiver 1408. In some other implementations, the NE 1400 may have more than one transceiver 1408. The transceiver 1408 may represent a wireless transceiver. The transceiver 1408 may include one or more receiver chains 1410, one or more transmitter chains 1412, or a combination thereof.

[0218] A receiver chain 1410 may be configured to receive signals (e.g., control information, data, packets) over a wireless medium. For example, the receiver chain 1410 may include one or more antennas for receiving the signal over the air or wireless

medium. The receiver chain 1410 may include at least one amplifier (e.g., a low-noise amplifier (LNA)) configured to amplify the received signal. The receiver chain 1410 may include at least one demodulator configured to demodulate the received signal and obtain the transmitted data by reversing the modulation technique applied during transmission of the signal. The receiver chain 1410 may include at least one decoder for decoding/ processing the demodulated signal to receive the transmitted data.

[0219] A transmitter chain 1412 may be configured to generate and transmit signals (e.g., control information, data, packets). The transmitter chain 1412 may include at least one modulator for modulating data onto a carrier signal, preparing the signal for transmission over a wireless medium. The at least one modulator may be configured to support one or more techniques such as AM, FM, or digital modulation schemes like PSK or QAM. The transmitter chain 1412 may also include at least one power amplifier configured to amplify the modulated signal to an appropriate power level suitable for transmission over the wireless medium. The transmitter chain 1412 may also include one or more antennas for transmitting the amplified signal into the air or wireless medium.

[0220] **Figure 15** illustrates one embodiment of a method 1500 in accordance with aspects of the present disclosure. In various embodiments, the operations of the method 1500 may be implemented by a UE as described herein. In some implementations, the UE may execute a set of instructions to control the function elements of the UE to perform the described functions.

[0221] At step 1502, the method 1500 may include receiving a polar codeword comprising a set of bits. The operations of step 1502 may be performed in accordance with examples as described herein. In some implementations, aspects of the operation of step 1502 may be performed by a UE, as described with reference to Figure 12.

[0222] At step 1504, the method 1500 may include generating a permuted vector based at least in part on the polar codeword and a permutation matrix. The operations of step 1504 may be performed in accordance with examples as described herein. In some implementations, aspects of the operation of step 1504 may be performed by a UE, as described with reference to Figure 12.

[0223] At step 1506, the method 1500 may include determining a plurality of polar subcodes based on the permuted vector. The operations of step 1506 may be performed in accordance with examples as described herein. In some implementations, aspects of the

operation of step 1506 may be performed by a UE, as described with reference to Figure 12.

[0224] At step 1508, the method 1500 may include identifying, for each polar subcode of the plurality of polar subcodes, a set of kernels, where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns. The operations of step 1508 may be performed in accordance with examples as described herein. In some implementations, aspects of the operation of step 1508 may be performed by a UE, as described with reference to Figure 12.

[0225] At step 1510, the method 1500 may include decoding the plurality of polar subcodes using a plurality of parallel polar code component decoders, where each parallel polar code component decoder comprises one or more fast decode modules to decode a corresponding kernel of the set of kernels. The operations of step 1510 may be performed in accordance with examples as described herein. In some implementations, aspects of the operation of step 1510 may be performed by a UE, as described with reference to Figure 12.

[0226] It should be noted that the method 1500 described herein describes one possible implementation, and that the operations and the steps may be rearranged or otherwise modified and that other implementations are possible.

[0227] **Figure 16** illustrates one embodiment of a method 1600 in accordance with aspects of the present disclosure. The operations of the method 1600 may be implemented by a NE, e.g., in a RAN, as described herein. In some implementations, the NE may execute a set of instructions to control the function elements of the NE to perform the described functions.

[0228] At step 1602, the method 1600 may include receiving a polar codeword comprising a set of bits. The operations of step 1602 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of step 1602 may be performed by a NE, as described with reference to Figure 14.

[0229] At step 1604, the method 1600 may include generating a permuted vector based at least in part on the polar codeword and a permutation matrix. The operations of step 1604 may be performed in accordance with examples as described herein. In some

implementations, aspects of the operations of step 1604 may be performed by a NE, as described with reference to Figure 14.

[0230] At step 1606, the method 1600 may include determining a plurality of polar subcodes based on the permuted vector. The operations of step 1606 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of step 1606 may be performed by a NE, as described with reference to Figure 14.

[0231] At step 1608, the method 1600 may include identifying, for each polar subcode of the plurality of polar subcodes, a set of kernels, where each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns. The operations of step 1608 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of step 1608 may be performed by a NE, as described with reference to Figure 14.

[0232] At step 1610, the method 1600 may include decoding the plurality of polar subcodes using a plurality of parallel polar code component decoders, where each parallel polar code component decoder comprises one or more fast decode modules to decode a corresponding kernel of the set of kernels. The operations of step 1610 may be performed in accordance with examples as described herein. In some implementations, aspects of the operations of step 1610 may be performed by a NE, as described with reference to Figure 14.

[0233] It should be noted that the method 1600 described herein describes one possible implementation, and that the operations and the steps may be rearranged or otherwise modified and that other implementations are possible.

[0234] The description herein is provided to enable a person having ordinary skill in the art to make or use the disclosure. Various modifications to the disclosure will be apparent to a person having ordinary skill in the art, and the generic principles defined herein may be applied to other variations without departing from the scope of the disclosure. Thus, the disclosure is not limited to the examples and designs described herein but is to be accorded the broadest scope consistent with the principles and novel features disclosed herein.

CLAIMS

What is claimed is:

1. A method performed by a decoder, the method comprising:
receiving a polar codeword comprising a set of bits;
generating a permuted vector based at least in part on the polar codeword and a permutation matrix;
determining a plurality of polar subcodes based on the permuted vector;
identifying, for each polar subcode of the plurality of polar subcodes, a set of kernels, wherein each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and
decoding the plurality of polar subcodes using a plurality of parallel polar code component decoders, wherein each parallel polar code component decoder comprises one or more fast decode modules to decode a corresponding kernel of the set of kernels.

2. The method of claim 1, wherein generating the permutation vector comprises applying the permutation matrix to the received polar codeword or to a Forney-style factor graph (FFG) of the polar codeword, or a combination thereof.

3. The method of claim 1, wherein the permutation matrix is selected from a pre-defined permutation set comprising an automorphism group of the polar codeword.

4. The method of claim 1, wherein the permutation matrix is selected from a set of permutation matrices that match an encoding procedure associated with the polar codeword.

5. The method of claim 1, further comprising receiving a set of permutation matrices, and wherein the permutation matrix is selected from the set of permutation matrices based at least in part on an error-correction performance.

6. The method of claim 1, further comprising:
selecting a plurality of permutation matrices;
generating a plurality of permuted vectors based on the plurality of permutation matrices; and

inputting the plurality of permuted vectors to a plurality of parallel sets of the plurality of parallel polar code component decoders.

7. The method of claim 6, further comprising:

selecting the plurality of permutation matrices from different pre-defined permutation sets;

applying a first permutation matrix to the polar codeword; and

applying a second permutation matrix to a factor graph associated with the polar codeword, wherein the second permutation matrix is a function of the first permutation matrix.

8. The method of claim 1, wherein a first error-correction performance associated with the permuted vector is within a predetermined variation of a second error-correction performance associated with the polar codeword.

9. The method of claim 1, wherein the permutation matrix is designed to re-order the set of bits into a maximum number of successive kernels.

10. The method of claim 1, wherein the set of kernels corresponds to a set of special nodes associated with a decoding tree, and wherein the set of one or more bit patterns corresponds to bit patterns of information bits or frozen bits, or both.

11. The method of claim 10, wherein the set of one or more bit patterns comprises one or more of:

a Rate-0 node having only frozen bits, a Rate-1 node having only information bits,

a repetition (REP) node wherein a rightmost bit is an information bit, and a remainder of the bits are frozen bits,

a single parity check (SPC) node wherein a leftmost bit is a frozen bit, and the remainder of the bits are information bits,

a dual REP (REP-2) node wherein the node includes only two information bits, and bit indices of the two information bits correspond to two largest indices,

a repeated parity check (RPC) node wherein the node includes only three frozen bits, and bit indices of the three frozen bits correspond to three smallest indices,

a parity-checked repetition (PCR) node wherein the node includes only three information bits, and the bit indices of the information bits correspond to three largest indices,

a dual SPC (SPC-2) node wherein the node includes only two frozen bits, and bit indices of the two frozen bits correspond to two smallest indices,

or a combination thereof.

12. The method of claim 1, further comprising:

determining a code tree based on the permuted vector; and

partitioning the code tree into the plurality of polar subcodes, wherein each polar subcode of the plurality of polar subcodes corresponds to one of a plurality of subtrees.

13. The method of claim 12, wherein the plurality of parallel polar code component decoders implements the one or more fast decode modules at intermediate levels of the plurality of subtrees to fast decode the set of kernels.

14. The method of claim 12, further comprising:

pruning a respective subtree based on a sphere decoding procedure; and

performing a simplified subtree traversal of the pruned respective subtree based on a successive cancellation decoder or a successive cancellation list decoder.

15. The method of claim 12, further comprising:

determining a path metric based on a maximized likelihood probability between a received vector and a random vector; and

partitioning the code tree based on the path metric and a multisphere subtree construction procedure.

16. The method of claim 1, further comprising decoding the polar codeword based on a combined output of the plurality of parallel polar code component decoders.

17. The method of claim 1, wherein the plurality of parallel polar code component decoders comprises one or more of: successive cancellation (SC) decoders,

successive cancellation list (SCL) decoders, list sphere decoders (List-SDs), maximum-likelihood (ML) decoders, or a combination thereof.

18. A user equipment (UE) for wireless communication, comprising:
at least one memory; and
at least one processor coupled with the at least one memory and configured to cause the UE to:
receive a polar codeword comprising a set of bits;
generate a permuted vector based at least in part on the polar codeword and a permutation matrix;
determine a plurality of polar subcodes based on the permuted vector;
identify, for each polar subcode of the plurality of polar subcodes, a set of kernels, wherein each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and
decode the plurality of polar subcodes using a plurality of parallel polar code component decoders, wherein each parallel polar code component decoder comprises one or more fast decode modules to decode a corresponding kernel of the set of kernels.

19. A base station for wireless communication, comprising:
at least one memory; and
at least one processor coupled with the at least one memory and configured to cause the base station to:
receive a polar codeword comprising a set of bits;
generate a permuted vector based at least in part on the polar codeword and a permutation matrix;
determine a plurality of polar subcodes based on the permuted vector;
identify, for each polar subcode of the plurality of polar subcodes, a set of kernels, wherein each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and
decode the plurality of polar subcodes using a plurality of parallel polar code component decoders, wherein each parallel polar code component decoder comprises one or more fast decode modules to decode a corresponding kernel of the set of kernels.

20. A processor for wireless communication, comprising:

at least one controller coupled with at least one memory and configured to cause the processor to:

receive a polar codeword comprising a set of bits;

generate a permuted vector based at least in part on the polar codeword and a permutation matrix;

determine a plurality of polar subcodes based on the permuted vector;

identify, for each polar subcode of the plurality of polar subcodes, a set of kernels, wherein each kernel of the set of kernels is associated with a bit pattern of a set of one or more bit patterns; and

decode the plurality of polar subcodes using a plurality of parallel polar code component decoders, wherein each parallel polar code component decoder comprises one or more fast decode modules to decode a corresponding kernel of the set of kernels.