SCRAMBLING SEQUENCE GENERATION IN A COMMUNICATION SYSTEM

[0001] The present application claims priority to provisional U.S. Application Serial No. 60/977,638, entitled "METHOD AND APPARATUS FOR SCRAMBLING SEQUENCE GENERATION IN A COMMUNICATION SYSTEM," filed October 4, 2007, assigned to the assignee hereof and incorporated herein by reference.

BACKGROUND

I. Field

[0002] The present disclosure relates generally to communication, and more specifically to techniques for generating scrambling sequences in a communication system.

II. Background

[0003] Communication systems are widely deployed to provide various communication content such as voice, video, packet data, messaging, broadcast, etc. These systems may be multiple-access systems capable of supporting multiple users by sharing the available system resources. Examples of such multiple-access systems include Code Division Multiple Access (CDMA) systems, Time Division Multiple Access (TDMA) systems, Frequency Division Multiple Access (FDMA) systems, Orthogonal FDMA (OFDMA) systems, and Single-Carrier FDMA (SC-FDMA) systems.

[0004] A communication system may apply scrambling at various points in a transmit signal generation process. Scrambling is a process in which data is multiplied with a scrambling sequence to randomize the data. The scrambling sequence may be a pseudo-random number (PN) sequence having good spectral and correlation properties. Scrambling may be performed for various purposes such as to randomize persistent interference between signals from different transmitters, to avoid undesired signal spectrum properties, etc. Scrambling may be performed in different manners for

different channels and signals. It is desirable to efficiently perform scrambling for all channels and signals.

SUMMARY

[0005] Techniques for performing scrambling and descrambling in a communication system are described herein. In an aspect, different scrambling sequences for different channels and signals may be generated based on different cyclic shifts of a base scrambling sequence. The base scrambling sequence may be a maximal-length sequence generated with a linear feedback shift register (LFSR). The LFSR may implement a single generator polynomial that is applicable for all supported channels and signals.

[0006] In one design, the LFSR may be initialized to an initial state to generate the base scrambling sequence. A first scrambling sequence may be generated based on modulo-2 addition of a selected combination of LFSR outputs. The selected combination of LFSR outputs may be determined based on a sequence selector value for a channel. The sequence selector value may be determined based on a channel type value and at least one parameter value for the channel. The sequence selector value may comprise (i) a first set of bits that may be assigned different values for different supported channels and signals and (ii) a second set of bits for channel-specific or signal-specific parameters. Scrambling or descrambling for the channel may then be performed with the scrambling sequence.

[0007] In another aspect, two scrambling sequences may be used to generate a reference signal sent on a variable system bandwidth. In one design, the two scrambling sequences may be generated based on different cyclic shifts of a base scrambling sequence. The first scrambling sequence may have its start mapped to the center of the system bandwidth and may traverse outwardly in positive frequency direction. The second scrambling sequence may have its start mapped to the center of the system bandwidth and may traverse outwardly in negative frequency direction. Scrambling or descrambling for positive frequencies for the reference signal may be performed with the first scrambling sequence. Scrambling or descrambling for negative frequencies for the reference signal may be performed with the second scrambling sequence. This design may ensure that the scrambling sequences in the center of the system band are

the same irrespective of the system bandwidth. This design may also simplify generation of the scrambling sequences for the reference signal regardless of the system bandwidth.

[0008] Various aspects and features of the disclosure are described in further detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] FIG. 1 shows a wireless communication system.

[0010] FIG. 2 shows an example frame structure.

[0011] FIG. 3 shows a block diagram of a scrambling sequence generator.

[0012] FIG. 4 shows use of two scrambling sequences for a reference signal.

[0013] FIG. 5 shows a process for performing scrambling or descrambling.

[0014] FIG. 6 shows a process for generating a scrambling sequence.

[0015] FIG. 7 shows an apparatus for performing scrambling or descrambling.

[0016] FIG. 8 shows a process for processing a reference signal.

[0017] FIG. 9 shows an apparatus for processing a reference signal.

[0018] FIG. 10 shows a block diagram of a Node B and a UE.

DETAILED DESCRIPTION

[0019] The techniques described herein may be used for various wireline and wireless communication systems such as CDMA, TDMA, FDMA, OFDMA, SC-FDMA and other systems. The terms "system" and "network" are often used interchangeably. A CDMA system may implement a radio technology such as Universal Terrestrial Radio Access (UTRA), cdma2000, etc. UTRA includes Wideband CDMA (WCDMA) and other variants of CDMA. cdma2000 covers IS-2000, IS-95 and IS-856 standards. A TDMA system may implement a radio technology such as Global System for Mobile Communications (GSM). An OFDMA system may implement a radio technology such as Evolved UTRA (E-UTRA), Ultra Mobile Broadband (UMB), IEEE 802.11 (Wi-Fi), IEEE 802.16 (WiMAX), IEEE 802.20, Flash-OFDM®, etc. UTRA and E-UTRA are part of Universal Mobile Telecommunication System (UMTS). 3GPP Long Term Evolution (LTE) is an upcoming release of UMTS that uses E-UTRA, which employs OFDMA on the downlink and SC-FDMA on the uplink. UTRA, E-

UTRA, UMTS, LTE and GSM are described in documents from an organization named "3rd Generation Partnership Project" (3GPP). cdma2000 and UMB are described in documents from an organization named "3rd Generation Partnership Project 2" (3GPP2). For clarity, certain aspects of the techniques are described below for LTE, and LTE terminology is used in much of the description below.

[0020] **FIG. 1** shows a wireless communication system 100, which may be an LTE system. System 100 may include a number of Node Bs 110 and other network entities. A Node B may be a fixed station that communicates with the UEs and may also be referred to as an evolved Node B (eNB), a base station, an access point, etc. Each Node B 110 provides communication coverage for a particular geographic area. To improve system capacity, the overall coverage area of a Node B may be partitioned into multiple (e.g., three) smaller areas. Each smaller area may be served by a respective Node B subsystem. In 3GPP, the term "cell" can refer to the smallest coverage area of a Node B and/or a Node B subsystem serving this coverage area. .

[0021] UEs 120 may be dispersed throughout the system, and each UE may be stationary or mobile. A UE may also be referred to as a mobile station, a terminal, an access terminal, a subscriber unit, a station, etc. A UE may be a cellular phone, a personal digital assistant (PDA), a wireless modem, a wireless communication device, a handheld device, a laptop computer, a cordless phone, etc. A UE may communicate with a Node B via the downlink and uplink. The downlink (or forward link) refers to the communication link from the Node B to the UE, and the uplink (or reverse link) refers to the communication link from the UE to the Node B.

[0022] LTE utilizes orthogonal frequency division multiplexing (OFDM) on the downlink and single-carrier frequency division multiplexing (SC-FDM) on the uplink. OFDM and SC-FDM partition the system bandwidth into multiple (K) orthogonal subcarriers, which are also commonly referred to as tones, bins, etc. Each subcarrier may be modulated with data. In general, modulation symbols are sent in the frequency domain with OFDM and in the time domain with SC-FDM. The spacing between adjacent subcarriers may be fixed, and the total number of subcarriers (K) may be dependent on the system bandwidth. For example, K may be equal to 128, 256, 512, 1024 or 2048 for system bandwidth of 1.25, 2.5, 5, 10 or 20 MHz, respectively.

[0023] The K total subcarriers may be grouped into resource blocks. Each resource block may include N subcarriers (e.g., $N = 12$ subcarriers) in one slot. The available

resource blocks may be assigned to UEs for transmission of traffic data and control information.

[0024] **FIG. 2** shows an example frame structure 200 used in LTE. The transmission timeline for the downlink may be partitioned into units of radio frames. Each radio frame may have a predetermined duration (e.g., 10 milliseconds (ms)) and may be partitioned into 10 subframes with indices of 0 through 9. Each subframe may include two slots, and each slot may include L symbol periods, e.g., $L = 6$ symbol periods for an extended cyclic prefix (CP) or $L = 7$ symbol periods for a normal cyclic prefix. The 2L symbol periods in each subframe may be assigned indices of 0 through 2L–1.

[0025] The system may support various channels used to send data and control information on the downlink and uplink. The system may also support reference signals and other signals used for various purposes. Table 1 lists some channels and signals that may be supported by the system in accordance with one design. The system may also support other channels and/or signals. A reference signal is a signal generated based on known data and may also be referred to as pilot, preamble, training, sounding, etc. A downlink reference signal (which may also be referred to as a cell-specific reference signal) may be generated with one or more pseudo-random sequences (PRS), as described below.

Table 1

Channel/Signal	Name	Description
Downlink reference signal	PRS	A signal sent by a Node B and used by the UEs for channel estimation and channel quality measurement.
Physical downlink control channel	PDCCH	Carry control information (e.g., scheduling information, ACK, NACK, etc.) on the downlink for different UEs.
Physical control format indicator channel	PCFICH	Carry information indicating number of PDCCH symbols per subframe.
Physical hybrid-ARQ indicator channel	PHICH	Carry feedback information (e.g., ACK, NACK) for HARQ.
Physical broadcast channel	PBCH	Carry cell-specific information broadcast to all UEs.
Physical multicast	PMCH	Carry multicast information sent to multiple

channel		UEs.
Physical downlink shared channel	PDSCH	Carry traffic data on the downlink to different UEs.
Physical uplink shared channel	PUSCH	Carry traffic data sent by different UEs on the uplink.
Physical uplink control channel	PUCCH	Carry control information sent by different UEs on the uplink.

[0026] Scrambling may be performed at a transmitter for various channels and signals, such as those shown in Table 1. For some channels/signals, it may be desirable to have a scrambling sequence be associated with a resource element or block that a channel/signal occupies. For other channels/signals, it may be better to have the scrambling sequence be disassociated from the occupied resource element or block.

[0027] Descrambling may be performed at a receiver to undo the scrambling performed by the transmitter. Descrambling may be performed using the same scrambling sequence used at the transmitter for scrambling. It is desirable to efficiently generate scrambling sequences for different channels/signals.

[0028] In an aspect, a scrambling sequence generation design that may be used for scrambling and descrambling for different channels and signals is described. A base scrambling sequence may be generated such that different shifts of the sequence are sufficiently decorrelated. Different cyclic shifts of the base scrambling sequence may then be used for different channels and signals.

[0029] In one design, a binary maximum length sequence, which is also commonly referred to as an m-sequence, may be used as the base scrambling sequence. An m-sequence has the longest possible length of $L = 2^Q - 1$ for a given degree Q. An m-sequence may be generated with a linear feedback shift register (LFSR) that implements a primitive polynomial. The m-sequence length should be sufficiently long. In one design, a 50-bit LFSR may be used to generate an m-sequence with a length of $2^{50} - 1$. A generator polynomial $G(x)$ for the m-sequence may be expressed as:

$$G(x) = x^{50} + g_{49} \cdot x^{49} + g_{48} \cdot x^{48} + ... + g_2 \cdot x^2 + g_1 \cdot x + 1 , \quad \text{Eq (1)}$$

where g_1 through g_{49} are coefficients of the generator polynomial, and

x^1 through x^{50} are outputs of the first through last delays, respectively, in the LFSR.

[0030] Equation (1) shows a general form for the generator polynomial $G(x)$. Each coefficient g_i, for $i = 1, ..., 49$, may be equal to '1' or '0'. Different generator polynomials may be defined with different sets of values for the 49 coefficients g_1 through g_{49}.

[0031] In general, a generator polynomial $G(x)$ of any suitable degree may be used for the base scrambling sequence. The same generator polynomial $G(x)$ may be used for all channels and signals, which may then avoid having to reconfigure the LFSR. The LFSR may be set to the same initial state for all channels and signals. Different cyclic shifts of the base scrambling sequence may be obtained by modulo-2 summing different combinations of LFSR outputs. The specific LFSR outputs to include in the modulo-2 sum are determined by the desired cyclic shift. Alternatively, the desired cyclic shift may be achieved by setting different initial states and modulo-2 summing certain combination of LFSR outputs. More than one LFSR may also be used, and the generated outputs may be modulo-2 summed to obtain the desired scrambling sequence.

[0032] **FIG. 3** shows a block diagram of a design of a scrambling sequence generator 300 that implements the generator polynomial shown in equation (1). Generator 300 includes an LFSR 310 (which is also referred to as a PN generator) and a cyclic shift unit 320 (which is also referred to as a masking unit). Within LFSR 310, 50 delay units 312-1 through 312-50 are coupled in series, with each delay unit providing one sample period of delay. The first delay unit 312-1 has its input coupled to an output of a modulo-2 addition unit 316. Each remaining delay unit has its input coupled to an output of a preceding delay unit. The first 49 delay units 312-1 through 312-49 also provide their outputs to 49 multipliers 314-1 through 314-49, respectively. The last delay unit 312-50 provides its output directly to modulo-2 addition unit 316. Each multiplier 314 multiplies its input with a respective coefficient g_i and provides its output to modulo-2 addition unit 316. FIG. 3 shows 49 multipliers 314-1 through 314-49 for 49 coefficients g_1 through g_{49}, respectively. Each multiplier 314 may be present if its coefficient g_i is equal to '1' and may be omitted (for no connection) if its coefficient is equal to '0'. Unit 316 performs modulo-2 addition on all of its inputs and provides the base scrambling sequence, which is fed back to the first delay unit 312-1.

[0033] Within cyclic shift unit 320, 50 AND gates 322-1 through 322-50 receive the outputs of 50 delay units 312-1 through 312-50, respectively. The 50 AND gates 322-1

through 322-50 also receive 50 bits m_0 through m_{49}, respectively, of a sequence selector. The sequence selector determines the amount of cyclic shift, and different values of the 50 selector bits m_0 through m_{49} correspond to different cyclic shifts of the base scrambling sequence. Each AND gate 322 performs logical AND on its two inputs and provides its output to a modulo-2 addition unit 324. Unit 324 performs modulo-2 addition on all of its 50 inputs and provides a scrambling sequence with the selected cyclic shift.

[0034] Scrambling sequence generator 300 may be reset at the beginning of every application of scrambling. For example, scrambling sequence generator 300 may be reset once at each symbol period for a downlink reference signal, once for every code block sent on the PDSCH, etc. For scrambling of binary encoded bits, one scrambling bit may be generated for each encoded bit, and each encoded bit may be multiplied with its scrambling bit to generate a scrambled bit. For descrambling, each scrambled bit may be multiplied with its scrambling bit to obtain the corresponding encoded bit.

[0035] The scrambling sequences for different channels and signals may be individualized by selecting different LFSR outputs for summing. In the design shown in equation (1) and FIG. 3, the 50 delay units 312-1 through 312-50 support 50 selector bits, which may be used to obtain 2^{50} different cyclic shifts. The 50 selector bits may be allocated for different channels and signals in various manners. Table 2 shows one design of allocating the 50 selector bits.

Table 2

Field	Sequence selector assignment	Number of bits
Reserved	m_{48} and m_{49}	2
Channel/signal type	m_{44} … m_{47}	4
Channel/signal-specific fields	m_0 … m_{43}	44

[0036] In the design shown in Table 2, the 50 selector bits are divided such that the first two bits are reserved, the next four bits are for channel/signal type, and the remaining 44 bits are for channel/signal-specific parameters.

[0037] Table 3 gives the values of the four selector bits m_{44} through m_{47} for different channel/signal types, in accordance with one design.

Table 3 – Channel/Signal Type

Channel/Signal	Channel type value	Channel/Signal	Channel type value
PRS (normal CP)	'0000'	PBCH	'0101'
PRS (extended CP)	'0001'	PMCH	'0110'
PDCCH	'0010'	PDSCH	'0111'
PCFICH	'0011'	PUSCH	'1000'
PHICH	'00100'	Other	Reserved

[0038] A periodicity of 10 ms may be assumed for the PRS with normal cyclic prefix and extended cyclic prefix, PDCCH, PDSCH and PUSCH. A periodicity of 40 ms may be assumed for the PBCH. A periodicity of 10, 20 or 40 ms may be assumed for the PCFICH, PHICH and PMCH.

[0039] Table 4 gives signal-specific fields for the PRS with normal cyclic prefix and extended cyclic prefix, in accordance with one design. The Cell_ID parameter indicates a cell identifier (ID) for a cell transmitting a downlink reference signal. The SSC_ID parameter indicates a specific secondary synchronization code (SSC) sequence among a set of available SSC sequences. The Antenna_ID parameter indicates a specific antenna at a Node B. The Subframe_ID parameter indicates a specific subframe out of 10 subframes in a radio frame. The Symbol_ID parameter indicates a specific OFDMA symbol out of either 12 or 14 OFDMA symbols in a subframe. The Frequency_+/- parameter indicates whether a scrambling sequence is for positive or negative frequencies, as described below.

Table 4 – Signal-Specific Fields for PRS

PRS (normal CP)	
Parameter	**Number of Bits**
SSC_ID	8
Antenna_ID	2
Subframe_ID	4
Symbol_ID	4
Frequency_+/-	1

PRS (extended CP)	
Parameter	**Number of Bits**
Cell_ID	9
Antenna_ID	2
Subframe_ID	4
Symbol_ID	4
Frequency_+/-	1

Reserved	25	Reserved	24

[0040] Table 5 gives channel-specific fields for the PDCCH and PDSCH, in accordance with one design. The UE MAC_ID parameter indicates a Medium Access Control (MAC) ID assigned to a UE. The Stream_ID parameter indicates a data stream being sent on the PDSCH. The Code_Block_ID parameter indicates a specific code block being sent on the PDSCH. The design in Table 5 support scrambling as a function of the Cell_ID and the UE MAC_ID for the PDSCH.

Table 5 – Channel-Specific Fields for PDCCH and PDSCH

PDCCH	
Parameter	**Number of Bits**
Cell_ID	9
Subframe_ID	4
Symbol_ID	4
Reserved	27

PDSCH	
Parameter	**Number of Bits**
Cell_ID	9
UE MAC_ID	16
Stream_ID	1
Code_Block_ID	6
Reserved	12

[0041] Table 6 gives channel-specific fields for the PBCH and PCFICH, in accordance with one design. The Frame_ID parameter indicates a specific radio frame.

Table 6 – Channel-Specific Fields for PBCH and PCFICH

PBCH	
Parameter	**Number of Bits**
Cell_ID	9
Frame_ID	2
Subframe_ID	4
Symbol_ID	5
Reserved	24

PCFICH	
Parameter	**Number of Bits**
Cell_ID	9
Subframe_ID	4
Reserved	31

[0042] Table 7 gives channel-specific fields for the PHICH and PMCH, in accordance with one design.

Table 7 – Channel-Specific Fields for PHICH and PMCH

PHICH	
Parameter	**Number of Bits**
Cell_ID	9
Reserved	35

PMCH	
Parameter	**Number of Bits**
Cell_ID	9
Reserved	35

[0043] Table 8 gives channel-specific fields for the PUSCH in accordance with one design.

Table 8 – Channel-Specific Fields for PUSCH

PUSCH	
Parameter	**Number of Bits**
UE MAC_ID	16
Code_Block_ID	6
Reserved	22

[0044] In the design described above, a scrambling sequence for a channel or a signal may be generated based on a sequence selector value, which may comprise a set of parameter values for parameters/attributes of the channel or signal. The time periodicity of the scrambling sequence may be flexibly controlled by including the symbol ID, slot ID, subframe ID, frame ID, etc., in the set of parameters for the sequence selector. The scrambling sequence may be associated or disassociated with resource elements or blocks by including or excluding the resource element/block index in the set of parameters for the sequence selector.

[0045] The scrambling sequence generator described herein may simplify generation of scrambling sequences for all channels and signals of interest. The scrambling sequence generator can generate scrambling sequences for current channels and signals as well as new channels and signals that may be added in the future. New channels, signals and/or parameters may be supported by using some of the reserved values to convey the new channels, signals and/or parameters.

[0046] In one design, a single scrambling sequence generator may be used to generate scrambling sequences for all channels and signals. The scrambling sequence

generator may be clocked as many times as needed for the longest scrambling sequence. For shorter scrambling sequences, only the initial part for the desired length may be generated. In another design, multiple instances of the scrambling sequence generator may be implemented, and each scrambling sequence generator may support one or more channels and/or signals. This design may be more suitable for certain hardware architectures.

[0047] In another aspect, two scrambling sequences may be used to generate a downlink reference signal that may be sent on a variable system bandwidth. For example, the system bandwidth may be 1.25, 2.5, 5, 10 or 20 MHz in LTE. It may be desirable to have a nested structure for the scrambling sequences for the downlink reference signal. For example, the scrambling sequences for system bandwidth of 10 MHz should match the center part of the scrambling sequences for system bandwidth of 20 MHz.

[0048] In one design, two scrambling sequences may be generated for the downlink reference signal and may be distinguished by the Frequency_+/- bit in the sequence selector. The scrambling sequences may be used as PRS sequences to generate the downlink reference signal. The first scrambling sequence may be used to scramble 'positive frequencies', and the second scrambling sequence may be used to scramble 'negative frequencies'.

[0049] **FIG. 4** shows a design of generating two scrambling sequences for the downlink reference signal. The first scrambling sequence includes N scrambling bits u_0 through u_{N-1}, where N is the maximum number of scrambling bits needed for the largest system bandwidth of 20 MHz. The second scrambling sequence includes N scrambling bits v_0 through v_{N-1}. The first scrambling sequence may be generated with the parameters shown in Table 4 and with the Frequency_+/- parameter set to '1'. The second scrambling sequence may be generated with the same parameters but with the Frequency_+/- parameter set to '0'.

[0050] To generate the downlink reference signal, the first scrambling sequence may be used to scramble positive frequencies starting from the smallest positive frequency. The center or DC subcarrier is typically not used. The first scrambling sequence may be mapped to subcarriers used for the downlink reference signal in the order of increasing frequency, as shown in FIG. 4. The second scrambling sequence may be used to scramble 'negative frequencies' starting from the highest negative

frequency, i.e., the subcarrier closest to the DC subcarrier. The second scrambling sequence may be mapped to the subcarriers used for the downlink reference signal in the opposite direction.

[0051] As shown in FIG. 4, for system bandwidth of 20 MHz, the entire first scrambling sequence may be used, and all N scrambling bits u_0 through u_{N-1} may be mapped to N positive subcarriers used for the downlink reference signal. The entire second scrambling sequence may also be used, and all N scrambling bits v_0 through v_{N-1} may be mapped to N negative subcarriers used for the downlink reference signal. For system bandwidth of less than 20 MHz, the initial portion of the first scrambling sequence may be used, and M scrambling bits u_0 through u_{M-1} may be mapped to the M positive subcarriers used for the downlink reference signal. The initial portion of the second scrambling sequence may also be used, and M scrambling bits v_0 through v_{M-1} may be mapped to the M negative subcarriers used for the downlink reference signal. M may be dependent on the system bandwidth and may be less than N.

[0052] The scrambling sequence generation design shown in FIG. 4 ensures that the scrambling sequences in the center of the system band are the same irrespective of the system bandwidth. Furthermore, this design does not require the entire first and second scrambling sequences to be generated in each OFDMA symbol carrying the downlink reference signal. By mapping the start of the first and second scrambling sequences to the center of the system bandwidth and by traversing outwardly in both positive and negative frequency directions, as shown in FIG. 4, only scrambling bits that are actually used are generated. This design may simplify scrambling sequence generation and may avoid redundant operations. This design may also efficiently support different system bandwidths, different number of subcarriers used for the downlink reference signal, and different scrambling sequence lengths.

[0053] The design in FIG. 4 may also allow for simultaneously generation of the first and second scrambling sequences for the positive and negative frequency parts of the downlink reference signal. If the scrambling sequences are very long, then the scrambling sequences may be partitioned into segments, and different segments of the scrambling sequences may be processed in parallel with multiple instances of the scrambling sequence generator. A memory may store the initial LFSR states for the scrambling sequence generator for each segment.

[0054] In an alternative design, a single scrambling sequence of length 2N may be used to generate the downlink reference signal. The start of this scrambling sequence may be mapped to the leftmost (or most negative) frequency, and the scrambling sequence may traverse to the rightmost (or most positive) frequency. In this design, an initial portion of the scrambling sequence may be discarded, with the amount to discard being dependent on the system bandwidth. This design essentially generates the entire scrambling sequence for 20MHz and uses only the variable length center portion as needed for the particular system bandwidth. This design may require more overhead because a portion of the generated scrambling bits would be thrown away.

[0055] **FIG. 5** shows a design of a process 500 for performing scrambling at a transmitter or descrambling at a receiver in a communication system. Process 500 may be performed by a Node B, a UE, or some other entity.

[0056] An LFSR may be initialized to an initial state to generate a base scrambling sequence (block 512). A first scrambling sequence may be generated based on modulo-2 addition of a selected combination of LFSR outputs, with the first scrambling sequence being a first cyclic shift of the base scrambling sequence (block 514). Scrambling or descrambling for a first channel may be performed with the first scrambling sequence, with the first channel being one of a set of channels and possibly signals supported by the base scrambling sequence (block 516). A second scrambling sequence may also be generated based on modulo-2 addition of a second selected combination of LFSR outputs, with the second scrambling sequence being a second cyclic shift of the base scrambling sequence. The first and second scrambling sequences may be generated in parallel based on different combinations of LFSR outputs. Scrambling or descrambling for a second channel or a reference signal may be performed with the second scrambling sequence.

[0057] In one design of block 512, the LFSR may implement a single generator polynomial that is applicable for all supported channels and signals. The LFSR may be reset for each application of scrambling or descrambling for the first channel. The LFSR may be initialized to the same initial state for all supported channels and signals.

[0058] **FIG. 6** shows a design of a process 600 for generating the first scrambling sequence. Process 600 is one design of block 514 in FIG. 5. A channel type value for the first channel may be determined (block 612). At least one parameter value for at least one parameter of the first channel may also be determined (block 614). A

sequence selector value for the first channel may be determined based on the channel type value and the at least one parameter value (block 616). The combination of LFSR outputs to perform modulo-2 addition may be determined based on the sequence selector value (block 618). The first scrambling sequence may be obtained by (i) selecting or deselecting each output bit of the LFSR based on the sequence selector value and (ii) summing the selected output bits of the LFSR with modulo-2 addition to obtain the first scrambling sequence.

[0059] The sequence selector value may comprise a first set of bits that may be assigned different values for different channels and signals supported by the base scrambling sequence. The sequence selector value may also comprise a second set of bits for channel-specific or signal-specific parameters. The supported channels and signals may include a broadcast channel, a multicast channel, a downlink shared channel, a downlink control channel, an uplink shared channel, an uplink control channel, a reference signal, some other channel or signal, or any combination thereof. The channel-specific or signal-specific parameters may include a cell ID, an antenna ID, a frame ID, a subframe ID, a symbol ID, a UE ID, a stream ID, a code block ID, some other parameter, or any combination thereof.

[0060] **FIG. 7** shows a design of an apparatus 700 for performing scrambling or descrambling in a communication system. Apparatus 700 includes a module 712 to initialize a LFSR to an initial state to generate a base scrambling sequence, a module 714 to generate a first scrambling sequence based on modulo-2 addition of a selected combination of LFSR outputs, with the first scrambling sequence being a first cyclic shift of the base scrambling sequence, and a module 716 to perform scrambling or descrambling for a first channel with the first scrambling sequence, the first channel being one of a set of channels supported by the base scrambling sequence.

[0061] **FIG. 8** shows a design of a process 800 for processing a reference signal. Process 800 may be performed by a Node B, a UE, or some other entity. A first scrambling sequence may be generated based on a first cyclic shift of a base scrambling sequence (block 812). A second scrambling sequence may be generated based on a second cyclic shift of the base scrambling sequence (block 814). The first scrambling sequence may have its start mapped to the center of the system bandwidth and may traverse outwardly in positive frequency direction, e.g., as shown in FIG. 4 (block 816). The second scrambling sequence may have its start mapped to the center of the system

bandwidth and may traverse outwardly in negative frequency direction, e.g., as also shown in FIG. 4 (block 818). Scrambling or descrambling for positive frequencies for a reference signal may be performed with the first scrambling sequence (block 820). Scrambling or descrambling for negative frequencies for the reference signal may be performed with the second scrambling sequence (block 822).

[0062] In one design of blocks 812 and 814, a first sequence selector value comprising a first value for a frequency polarity parameter (e.g., the Frequency_+/- parameter) may be determined. A second sequence selector value comprising a second value for the frequency polarity parameter may also be determined. The base scrambling sequence may be cyclically shifted based on the first sequence selector value to obtain the first scrambling sequence. The base scrambling sequence may also be cyclically shifted based on the second sequence selector value to obtain the second scrambling sequence.

[0063] In one design of blocks 812 and 814, at least one scrambling sequence generator may be reset to a starting point. The first M scrambling bits of the first scrambling sequence may be generated with the scrambling sequence generator(s), where M is determined based on the system bandwidth. The first M scrambling bits of the second scrambling sequence may also be generated with the scrambling sequence generator(s).

[0064] **FIG. 9** shows a design of an apparatus 900 for processing a reference signal. Apparatus 900 includes a module 912 to generate a first scrambling sequence based on a first cyclic shift of a base scrambling sequence, a module 914 to generate a second scrambling sequence based on a second cyclic shift of the base scrambling sequence, a module 916 to map the start of the first scrambling sequence to the center of the system bandwidth and traverse outwardly in positive frequency direction, a module 918 to map the start of the second scrambling sequence to the center of the system bandwidth and traverse outwardly in negative frequency direction, a module 920 to perform scrambling or descrambling for positive frequencies for a reference signal with the first scrambling sequence, and a module 922 to perform scrambling or descrambling for negative frequencies for the reference signal with the second scrambling sequence.

[0065] The modules in FIGS. 7 and 9 may comprise processors, electronics devices, hardware devices, electronics components, logical circuits, memories, etc., or any combination thereof.

[0066] **FIG. 10** shows a block diagram of a design of a Node B 110 and a UE 120, which may be one of the Node Bs and one of the UEs in FIG. 1. In this design, Node B 110 is equipped with T antennas 1034a through 1034t, and UE 120 is equipped with R antennas 1052a through 1052r, where in general $T \geq 1$ and $R \geq 1$.

[0067] At Node B 110, a transmit processor 1020 may receive traffic data for one or more UEs from a data source 1012, process the traffic data for each UE based on one or more modulation and coding schemes selected for that UE, and provide symbols for shared/data channels. Transmit processor 1020 may receive and process broadcast and multicast data and provide symbols for broadcast and multicast channels. Transmit processor 1020 may also receive and process control information from a controller/processor 1040 and provide symbols for control channels. Transmit processor 1020 may also generate symbols for reference signals, synchronization signals, etc. Transmit processor 1020 may perform scrambling for different channels and signals, as described above.

[0068] A transmit (TX) multiple-input multiple-output (MIMO) processor 1030 may multiplex the symbols for different channels and signals. TX MIMO processor 1030 may perform spatial processing (e.g., precoding) on the multiplexed symbols, if applicable, and provide T output symbol streams to T modulators (MODs) 1032a through 1032t. Each modulator 1032 may process a respective output symbol stream (e.g., for OFDM) to obtain an output sample stream. Each modulator 1032 may further process (e.g., convert to analog, amplify, filter, and upconvert) the output sample stream to obtain a downlink signal. T downlink signals from modulators 1032a through 1032t may be transmitted via T antennas 1034a through 1034t, respectively.

[0069] At UE 120, antennas 1052a through 1052r may receive the downlink signals from Node B 110 and provide received signals to demodulators (DEMODs) 1054a through 1054r, respectively. Each demodulator 1054 may condition (e.g., filter, amplify, downconvert, and digitize) a respective received signal to obtain input samples and may further process the input samples (e.g., for OFDM) to obtain received symbols. A MIMO detector 1056 may obtain received symbols from all R demodulators 1054a through 1054r, perform MIMO detection on the received symbols if applicable, and provide detected symbols. A receive processor 1058 may process (e.g., demodulate, deinterleave, and decode) the detected symbols, provide decoded data for UE 120 to a data sink 1060, and provide decoded control information to a controller/processor 1080.

[0070] On the uplink, at UE 120, traffic data from a data source 1062 and control information from controller/processor 1080 may be processed by a transmit processor 1064, further processed by a TX MIMO processor 1066 if applicable, conditioned by modulators 1054a through 1054r, and transmitted to Node B 110. At Node B 110, the uplink signals from UE 120 may be received by antennas 1034, conditioned by demodulators 1032, processed by a MIMO detector 1036 if applicable, and further processed by a receive processor 1038 to obtain the traffic data and control information transmitted by UE 120.

[0071] Controllers/processors 1040 and 1080 may direct the operation at Node B 110 and UE 120, respectively. Memories 1042 and 1082 may store data and program codes for Node B 110 and UE 120, respectively. A scheduler 1044 may schedule UEs for downlink and/or uplink transmission and may provide assignments of resources for the scheduled UEs.

[0072] Process 500 in FIG. 5, process 600 in FIG. 6, process 800 in FIG. 8 and/or other processes for the techniques described herein may be performed by one or more processors at Node B 110 and also by one or more processors at UE 120. For example, transmit processors 1020 and 1064 may perform process 500 and/or 600 for scrambling, and receive processors 1038 and 1058 may perform process 500 and/or 600 for descrambling. Transmit processor 1020 may also perform process 800 for scrambling of a downlink reference signal, and receive processor 1058 may perform process 800 for descrambling of the downlink reference signal.

[0073] Those of skill in the art would understand that information and signals may be represented using any of a variety of different technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description may be represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof.

[0074] Those of skill would further appreciate that the various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the disclosure herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is

implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present disclosure.

[0075] The various illustrative logical blocks, modules, and circuits described in connection with the disclosure herein may be implemented or performed with a general-purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general-purpose processor may be a microprocessor, but in the alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

[0076] The steps of a method or algorithm described in connection with the disclosure herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary storage medium is coupled to the processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an ASIC. The ASIC may reside in a user terminal. In the alternative, the processor and the storage medium may reside as discrete components in a user terminal.

[0077] In one or more exemplary designs, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over as one or more instructions or code on a computer-readable medium. Computer-readable media includes both computer storage media and communication media including any medium that facilitates transfer of a computer program from one place to another. A storage

media may be any available media that can be accessed by a general purpose or special purpose computer. By way of example, and not limitation, such computer-readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to carry or store desired program code means in the form of instructions or data structures and that can be accessed by a general-purpose or special-purpose computer, or a general-purpose or special-purpose processor. Also, any connection is properly termed a computer-readable medium. For example, if the software is transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and blu-ray disc where *disks* usually reproduce data magnetically, while *discs* reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[0078] The previous description of the disclosure is provided to enable any person skilled in the art to make or use the disclosure. Various modifications to the disclosure will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other variations without departing from the spirit or scope of the disclosure. Thus, the disclosure is not intended to be limited to the examples and designs described herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

[0079] WHAT IS CLAIMED IS:

CLAIMS

1. A method for communication, comprising:

initializing a linear feedback shift register (LFSR) to an initial state to generate a base scrambling sequence;

generating a first scrambling sequence based on modulo-2 addition of a selected combination of LFSR outputs, the first scrambling sequence being a first cyclic shift of the base scrambling sequence; and

performing scrambling or descrambling for a first channel with the first scrambling sequence, the first channel being one of a set of channels supported by the base scrambling sequence.

2. The method of claim 1, wherein the generating the first scrambling sequence comprises

determining a sequence selector value for the first channel, and

determining the selected combination of LFSR outputs based on the sequence selector value.

3. The method of claim 2, wherein the determining the sequence selector value comprises

determining a channel type value for the first channel, and

determining the sequence selector value based on the channel type value.

4. The method of claim 3, wherein the determining the sequence selector value further comprises

determining at least one parameter value for at least one parameter of the first channel, and

determining the sequence selector value based further on the at least one parameter value.

5. The method of claim 2, wherein the sequence selector value comprises a first set of bits assigned different values for different channels and signals supported by the base scrambling sequence.

6. The method of claim 5, wherein the channels and signals supported by the base scrambling sequence comprise at least one of a broadcast channel, a multicast channel, a downlink shared channel, a downlink control channel, an uplink shared channel, an uplink control channel, and a reference signal.

7. The method of claim 5, wherein the sequence selector value further comprises a second set of bits for channel-specific or signal-specific parameters.

8. The method of claim 7, wherein channel-specific parameters for the first channel comprise at least one of a cell identifier (ID), an antenna ID, a frame ID, a subframe ID, a symbol ID, a user equipment (UE) ID, a stream ID, and a code block ID.

9. The method of claim 1, wherein the generating the first scrambling sequence comprises

selecting or deselecting each output bit of the LFSR based on a sequence selector value for the first channel, and

summing selected output bits of the LFSR with modulo-2 addition to obtain the first scrambling sequence.

10. The method of claim 1, wherein the LFSR implements a single generator polynomial applicable for all channels and signals supported by the base scrambling sequence and is reset for each application of scrambling or descrambling for the first channel.

11. The method of claim 1, wherein the LFSR is initialized to the same initial state for all channels and signals supported by the base scrambling sequence.

12. The method of claim 1, further comprising:

generating a second scrambling sequence based on modulo-2 addition of a second selected combination of LFSR outputs, the second scrambling sequence being a second cyclic shift of the base scrambling sequence; and

performing scrambling or descrambling for a second channel or a reference signal with the second scrambling sequence.

13. The method of claim 12, wherein the first and second scrambling sequences are generated in parallel based on different combinations of LFSR outputs.

14. An apparatus for communication, comprising:

at least one processor configured to initialize a linear feedback shift register (LFSR) to an initial state to generate a base scrambling sequence, to generate a first scrambling sequence based on modulo-2 addition of a selected combination of LFSR outputs, the first scrambling sequence being a first cyclic shift of the base scrambling sequence, and to perform scrambling or descrambling for a first channel with the first scrambling sequence, the first channel being one of a set of channels supported by the base scrambling sequence.

15. The apparatus of claim 14, wherein the at least one processor is configured to determine a channel type value for the first channel, to determine a sequence selector value for the first channel based on the channel type value, and to determine the selected combination of LFSR outputs based on the sequence selector value.

16. The apparatus of claim 15, wherein the at least one processor is configured to determine at least one parameter value for at least one parameter of the first channel, and to determine the sequence selector value based further on the at least one parameter value.

17. The apparatus of claim 14, wherein the at least one processor is configured to determine a sequence selector value for the first channel, the sequence selector value comprising a first set of bits for a channel or signal type and a second set

of bits for channel-specific or signal-specific parameters, and to determine the selected combination of LFSR outputs based on the sequence selector value.

18. An apparatus for communication, comprising:

means for initializing a linear feedback shift register (LFSR) to an initial state to generate a base scrambling sequence;

means for generating a first scrambling sequence based on modulo-2 addition of a selected combination of LFSR outputs, the first scrambling sequence being a first cyclic shift of the base scrambling sequence; and

means for performing scrambling or descrambling for a first channel with the first scrambling sequence, the first channel being one of a set of channels supported by the base scrambling sequence.

19. The apparatus of claim 18, wherein the means for generating the first scrambling sequence comprises

means for determining a channel type value for the first channel,

means for determining a sequence selector value for the first channel based on the channel type value, and

means for determining the selected combination of LFSR outputs based on the sequence selector value.

20. The apparatus of claim 19, wherein the means for determining the sequence selector value further comprises

means for determining at least one parameter value for at least one parameter of the first channel, and

means for determining the sequence selector value based further on the at least one parameter value.

21. The apparatus of claim 18, wherein the means for generating the first scrambling sequence comprises

means for determining a sequence selector value for the first channel, the sequence selector value comprising a first set of bits for a channel or signal type and a second set of bits for channel-specific or signal-specific parameters, and

means for determining the selected combination of LFSR outputs based on the sequence selector value.

22. A computer program product, comprising:

a computer-readable medium comprising:

code for causing at least one computer to initialize a linear feedback shift register (LFSR) to an initial state to generate a base scrambling sequence,

code for causing the at least one computer to generate a first scrambling sequence based on modulo-2 addition of a selected combination of LFSR outputs, the first scrambling sequence being a first cyclic shift of the base scrambling sequence, and

code for causing the at least one computer to perform scrambling or descrambling for a first channel with the first scrambling sequence, the first channel being one of a set of channels supported by the base scrambling sequence.

23. A method for communication, comprising:

performing scrambling or descrambling for positive frequencies for a reference signal with a first scrambling sequence; and

performing scrambling or descrambling for negative frequencies for the reference signal with a second scrambling sequence.

24. The method of claim 23, further comprising:

mapping start of the first scrambling sequence to center of system bandwidth and traversing outwardly in positive frequency direction; and

mapping start of the second scrambling sequence to the center of the system bandwidth and traversing outwardly in negative frequency direction.

25. The method of claim 23, further comprising:

generating the first scrambling sequence based on a first cyclic shift of a base scrambling sequence; and

generating the second scrambling sequence based on a second cyclic shift of the base scrambling sequence.

26. The method of claim 23, further comprising:

resetting at least one scrambling sequence generator to a starting point;

generating first M scrambling bits of the first scrambling sequence with the at least one scrambling sequence generator, where M is determined based on system bandwidth; and

generating first M scrambling bits of the second scrambling sequence with the at least one scrambling sequence generator.

27. The method of claim 23, further comprising:

determining a first sequence selector value comprising a first value for a frequency polarity parameter;

determining a second sequence selector value comprising a second value for the frequency polarity parameter;

cyclically shifting a base scrambling sequence based on the first sequence selector value to obtain the first scrambling sequence; and

cyclically shifting the base scrambling sequence based on the second sequence selector value to obtain the second scrambling sequence.

28. An apparatus for communication, comprising:

at least one processor configured to perform scrambling or descrambling for positive frequencies for a reference signal with a first scrambling sequence, and to perform scrambling or descrambling for negative frequencies for the reference signal with a second scrambling sequence.

29. The apparatus of claim 28, wherein the at least one processor is configured to map start of the first scrambling sequence to center of system bandwidth and traverse outwardly in positive frequency direction, and to map start of the second scrambling sequence to the center of the system bandwidth and traverse outwardly in negative frequency direction.

30. The apparatus of claim 28, wherein the at least one processor is configured to generate the first scrambling sequence based on a first cyclic shift of a base scrambling sequence, and to generate the second scrambling sequence based on a second cyclic shift of the base scrambling sequence.

31. The apparatus of claim 28, wherein the at least one processor is configured to determine a first sequence selector value comprising a first value for a frequency polarity parameter, to determine a second sequence selector value comprising a second value for the frequency polarity parameter, to cyclically shift a base scrambling sequence based on the first sequence selector value to obtain the first scrambling sequence, and to cyclically shift the base scrambling sequence based on the second sequence selector value to obtain the second scrambling sequence.