

(51) International Patent Classification:
H03M 13/11 (2006.01)(21) International Application Number:
PCT/US2016/023904(22) International Filing Date:
24 March 2016 (24.03.2016)

(25) Filing Language: English

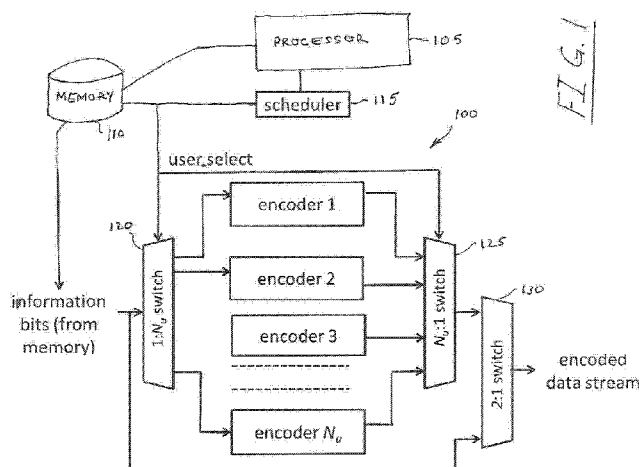
(26) Publication Language: English

(30) Priority Data:
62/137,588 24 March 2015 (24.03.2015) US(71) Applicant: **ALCATEL-LUCENT USA INC.** [US/US];
600-700 Mountain Avenue, Murray Hill, NJ 07974-0636
(US).(72) Inventors: **GOPALAKRISHNA PILLAI, Bipin, Sankar**;
18 Syndal Street, Fawcner, 3060 (AU). **ANTHAPAD-
MANABHAN, Nagaraj, Prasanth**; 600-700 Mountain
Avenue, Murray Hill, NJ 07974-0636 (US). **CHOW, Hun-
gkei**; 600-700 Mountain Avenue, Murray Hill, NJ 07974-
0636 (US). **SUVAKOVIC, Dusan**; 600-700 Mountain Ave-
nue, Murray Hill, NJ 07974-0636 (US).(74) Agent: **WYSE, Stephen, J.**; Alcatel-lucent USA Inc., At-
tention; Docket Administrator-room 3B-212F, 600-700
Mountain Avenue, Murray Hill, NJ 07974-0636 (US).(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,
BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR,
KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG,
MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM,
PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC,
SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ,
TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU,
TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE,
DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,
LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

(54) Title: FLEXIBLE ENCODING FOR FORWARD ERROR CORRECTION IN A COMMUNICATION NETWORK



(57) Abstract: A method and apparatus for efficiently providing FEC (forward error correction) encoding, in particular using quasi-cyclic LDPC codes, for transmissions in a communication network. A flexible encoder is provided for receiving information bits for transmission and for receiving code generation matrices applicable to one or more downstream users. The flexible encoder is initialized using the generator matrix of the associated code and creates parity check bits, which are stored in a parity check bit matrix until selected for inclusion in an encoded bit stream.

FLEXIBLE ENCODING FOR FORWARD ERROR CORRECTION IN A COMMUNICATION NETWORK

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This non-provisional application is related to and claims priority under 35 U.S.C. § 119(e) to U.S. Provisional Patent Application Serial No. 62/137,588 *entitled* Encoding for Forward Error Correction, *filed* on 24 March 2015, the entire contents of which are incorporated herein by reference.

BACKGROUND

Field of the Disclosure

[0002] The present disclosure relates generally to network communication and, more particularly, to a method and apparatus for encoding transmissions for forward error correction.

Description of the Related Art

[0003] The following abbreviations are herewith expanded, at least some of which are referred to within the following description.

[0004] EPON	Ethernet Passive Optical Network
[0005] FEC	Forward Error Correction
[0006] LDPC	Low Density Parity Check
[0007] LLID	Logical Link Identifier
[0008] OLT	Optical Line Terminal
[0009] ONT	Optical Network Terminal
[0010] ONU	Optical Network Unit
[0011] PON	Passive Optical Network
[0012] QC LDPC	Quasi-Cyclic LDPC
[0013] RAM	Random Access Memory
[0014] XGEM	XGPON Encapsulation Method
[0015] XG PON	10-Gigabit PON

[0016] Network transmissions in communication networks involve transmitting data from one network node to another, typically as a digital bit stream made up of the of the familiar 1s and 0s. No transmission channel is perfect, however, and some data bits may become lost or unrecognizable. One way to reduce the effect of these lost bits is to use FEC (forward error correction). In FEC, a bit stream is first encoded using a coding scheme that introduces a certain amount of redundancy into the data. When another network node receives the encoded data, it uses a reverse of the coding scheme to reconstruct the original bit stream even when some of the received data bits are unusable.

[0017] A number of commonly-used coding schemes have been developed, which generally involve processing a bit stream of given length to produce codewords, which are then transmitted to the receiving node for decoding. Whatever coding scheme is used, however, the process of producing and decoding codewords consumes energy. Over the vast amounts of data that are now being communicated, this energy usage becomes significant. There is always a need, therefore, for more efficient ways to transmit information while retaining the benefits of FEC.

SUMMARY OF EMBODIMENTS

[0018] The following presents a summary of the disclosed subject matter in order to provide a basic understanding of some aspects of the disclosed subject matter. This summary is not an exhaustive overview of the disclosed subject matter. It is not intended to identify key or critical elements of the disclosed subject matter or to delineate the scope of the disclosed subject matter. Its sole purpose is to present some concepts in a simplified form as a prelude to the more detailed description that is discussed later.

[0019] In one aspect, an apparatus for transmission includes a processor, a scheduler, and a flexible encoder configured to employ a plurality of FEC codes. A main memory device in communication with the encoder and the processor for receiving information bits and providing them to the flexible encoder as directed by the scheduler. In some implementations, the flexible encoder includes a generator matrix memory for populating with generator matrices associated with the plurality of FEC codes, an accumulator configured for receiving generator matrices from the generator matrix memory and processing information bits from the main memory to produce parity check bits, and a parity check bit memory for storing results from the accumulator.

[0020] In some implementations, the flexible encoder may further include a generator matrix control circuit for controlling operation of the generator matrix memory and a parity check bit memory control circuit for controlling operation of the parity check bit memory. It may also include a 2:1 switch configured to receive information bits from the main memory and parity check bits from the parity check bit memory and to selectively include them in an encoded bit stream in response to a received control signal and a main control circuit configured at least to provide the control signal to the 2:1 switch. The main control circuit may be further configured to generate a control signal to regulate the reading of information bits from the main memory.

[0021] In a preferred embodiment, the apparatus is an OLT in a PON and further includes an interleaver for interleaving the encoded bit stream and a scrambler for scrambling the interleaved bit stream.

[0022] In another aspect, the present invention provides a method for data transmission including receiving information bits for transmission, selecting an FEC code for encoding the information bits, initializing a flexible encoder, and providing to the flexible encoder a generator matrix associated with the selected FEC code. In some implementations, the information bits are processed in an accumulator using the provided generator matrix to produce parity check bits and the parity check bits in a parity check bit memory. The parity check bits are then combined with information bits to produce an encoded bit stream.

[0023] In a preferred embodiment, the process is performed by an OLT in a PON, and further includes interleaving the encoded bit stream with those of other users and scrambling the interleaved bit stream before providing it to a transmitter for transmission to one or more ONUs.

[0024] Additional aspects of the invention will be set forth, in part, in the detailed description, figures and any claims which follow, and in part will be derived from the detailed description, or can be learned by practice of the invention. It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention as disclosed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] The present disclosure may be better understood, and its numerous features and advantages made apparent to those skilled in the art by referencing the accompanying drawings. The use of the same reference symbols in different drawings indicates similar or identical items.

[0026] FIG. 1 is a block diagram illustrating selected components of an FEC encoder system according to one embodiment.

[0027] FIG. 2 is a block diagram illustrating selected components of an FEC encoder system according to one embodiment.

[0028] FIG. 3 is a block diagram illustrating an encoder according to one embodiment.

[0029] FIG. 4 is a block diagram illustrating an accumulator, such as the accumulator shown in FIG. 3, according to one embodiment.

[0030] FIG. 5 is a block diagram illustrating an adder block, such as an adder block shown in FIG. 3, according to one embodiment.

[0031] FIG. 6a is a block diagram illustrating a gen_cc control circuit, such as the gen_cc control circuit shown in FIG.3, according to one embodiment.

[0032] FIG. 6b is a block diagram illustrating a par_cc control circuit, such as the par_cc control circuit shown in FIG.3, according to one embodiment.

[0033] FIG. 7 is a block diagram illustrating an m_cc control circuit, such as the m_cc control circuit shown in FIG. 3, according to one embodiment.

[0034] FIG. 8 is a block diagram illustrating an accumulator according to another embodiment.

[0035] FIG. 9 is a block diagram illustrating an adder block, such as an adder block, shown in FIG. 8, according to one embodiment.

[0036] FIG. 10 is a block diagram illustrating an accumulator according to another embodiment.

[0037] FIG. 11 is a block diagram illustrating an adder block, such as an adder block, shown in FIG. 10, according to one embodiment.

[0038] FIG. 12 is a block diagram illustrating a gen_cc control circuit, such as the gen_cc control circuit shown in FIG.8, according to one embodiment.

[0039] FIG. 13 is a block diagram illustrating selected components of an exemplary PON (passive optical network) in which embodiments of the present invention may be advantageously implemented.

[0040] FIG. 14 is a block diagram illustrating selected components of a PON such as the PON illustrated in FIG. 13.

[0041] FIG. 15 is a flow diagram illustrating a method according to one embodiment.

DETAILED DESCRIPTION

[0042] This disclosure describes a method and apparatus for efficiently providing FEC (forward error correction) encoding for transmissions in a communication network. Although favorable results have been achieved in a test environment, however, no specific level of performance or efficiency is required unless explicitly recited in a particular embodiment.

[0043] One type of communication network is referred to as a PON (passive optical network). FIG. 13 is a simplified block diagram illustrating selected components of an exemplary PON (passive optical network) 800 in which embodiments of the present invention may be advantageously implemented. Note that PON 800 may, and in many implementations will, include additional components, and the configuration shown in FIG. 1 is intended to be exemplary rather than limiting.

[0044] In the example of FIG. 13, the network 800 includes an OLT (optical line terminal) 810, typically located at a CO (central office) 820. The OLT 810 communicates with ONUs (optical network units) 830a, 830b . . . 830n using an optical source via an optical path such as a fiber optical cable 840. Downstream data from the OLT 810 are addressed to one or more of the ONUs 830, each of which is configured to recognize data addressed to it. The optical cable 840 connects to a splitter/combiner 850 (or for convenience, simply “splitter”).

[0045] The splitter 850 divides the downstream optical signal from the OLT 810 between a number of downstream ports, for example ports 855a, 855b . . . 855n. From the splitter ports, the divided signal is carried to ONUs 830a, 830b . . . 830n via access fibers 860a, 860b . . . 860n. Each ONU 830 may also transmit upstream data to the OLT 810 via the access fibers 860a, 860b . . . 860n, typically during an assigned time slot. Optical signals from each ONU 830 are combined by the splitter 850 and propagate from there to the OLT 810 via optical cable 840. Note that as used herein, “ONU” is intended to be broadly construed to include ONTs (optical network terminals) and other CPE (customer premises equipment).

[0046] As should be apparent, in a point-to-multipoint-system such as a PON (in the downstream direction) each receiving node (such as an ONU) receives the same downstream transmission. While the receiving node is configured to extract data addressed to it, it must of necessity process the entire bit stream until this can occur. Naturally, it is most efficient for extraction to occur early in the process so the remainder of the receive train may process the information at a lower rate. This concept is illustrated in FIG. 14.

[0047] FIG 14 is a simplified block diagram illustrating selected components of a PON such as PON 800 illustrated in FIG. 13. In FIG. 14 only OLT 810 and a representative ONU 830 are shown. In this example, OLT 810 includes an FEC encoder 812, and interleaver 814, and a scrambler 816. Multiple user bit streams enter encoder 812 where they are encoded and the encoded bit streams are provided to interleaver 814. Interleaver 814 interleaves the multiple encoded bit streams into a single bit stream (at a higher rate), which is then scrambled by scrambler 816 for downstream transmission.

[0048] When the bit stream reached ONU 830, it is first decimated by decimator 832, that is, only those bits intended for ONU 830 are extracted from the bit stream (the remainder are typically discarded). The decimated bit stream is then provided to descrambler 834 (at a lower rate) for descrambling and then to FEC decoder 836 for decoding.

[0049] Note that the configurations of FIGS. 13 and 14 are illustrative and intended only to depict an environment where the solutions of the present disclosure may be useful. They are not intended to limit the scope of any claims unless and to the extent that they are specifically recited.

[0050] Although not apparent from the illustration of FIG. 14, it would be an advantage for an FEC encoder to be capable of accommodating numerous coding schemes; each downstream user has their own requirements. It may also be advantageous in some implementations to be able to allocate downstream transmission time slots that are less than a codeword in length to facilitate bit-interleaving of the downstream transmission.

[0051] FIG. 1 is a block diagram illustrating selected components of an FEC encoder system 100 according to one embodiment. In the embodiment of FIG. 1, FEC encoder system 100 includes a number of separate encoders referred to as encoder 1 through encoder N_u , accommodating N_u users and allowing each to be associated with a separate coding scheme. In this embodiment, transmission of less than a full codeword is enabled by providing a memory device associated with each encoder (not separately shown). Intermediate encoding results are accumulated in a respective memory until an associated time slot opens. Information bits from memory are selectively provided to each encoder by 1: N_u switch 120. Parity bits from each encoder (or encoder memory) are selectively fed into the bit stream via 1: N_u switch 125 and 2:1 switch 130. The 2:1 switch 130 selects between the encoded parity bit from 1: N_u switch 125 and the actual information bits.

[0052] In this embodiment, a processor 105 is present to control operation of the various components of encoding system 100. Scheduler 115, shown separately, determines the schedule, that is, the time slots, for downstream transmission to each user device (for example, an ONU in a PON). The schedule may be a function of, for example, of the amount of traffic available for sending to each user or the applicable quality of service associated with the use or the type of data being transmitted. Based on this schedule processor 105 or scheduler 115 sends a *user_select* signal to 1: N_u switches 120 and 125 and controls 2:1 switch 130 to accommodate the correct proportion of information and parity bits to produce the encoded data stream.

[0053] Memory device 110 is at least in part a physical memory apparatus such as a RAM (random access memory) and is used to store, among other things, data and program instructions for execution by processor 105. In this embodiment, memory device 110 may store, for example, information bits for downstream transmission generator matrices for codeword generation. Memory device 110 is non-transitory in the sense of not being merely a propagating signal unless explicitly recited in a particular embodiment.

[0054] FIG. 2 is a block diagram illustrating selected components of an FEC encoder system 200 according to some embodiments. In the embodiment of FIG. 2, the processor

205, memory 210, and scheduler 215 function analogously to the processor 105, memory 110, and scheduler 115 of FIG. 1. In the embodiment of FIG 2, however, a single, flexible encoder 240 is used, eliminating the need for the $1:N_u$ switches. The *user_select* function is, in effect, performed by the processor in selecting information bits from memory in accordance with the downstream transmission schedule determined by scheduler 215. A *code_select* signal is sent to the flexible encoder 240 to direct use of the FEC coding indicated for the user whose information is currently selected. Intermediate coding results are stored in a memory device (not separately shown) associated with the flexible encoder 240 to enable transmissions in time slots that are less than a codeword in length. Information bits are, in this embodiment, inserted into the data stream by the encoder 240, which inserts the parity bits as well. A flexible encoder that can be implemented in the embodiment of FIG 2 will now be described in greater detail.

[0055] FIG. 3 is a block diagram illustrating an encoder 300 according to one embodiment. In this embodiment, a main memory 290 stores at least information bits for transmission in a communication network. In a preferred embodiment, the main memory 290 is a RAM (random access memory). The scheduler 295 in the embodiment of FIG. 3 provides at least two signals to encoder 300, *initialize* and *user_id*. Because the flexible encoder 300 performs encoding for a number of users who may each have their coding requirements, the *user_id* signal indicates which user is being served at this time. (alternatively, a *code_select* signal may be employed instead of or in addition to *user_id*. See, for example, FIG. 2.) In the embodiment of FIG. 3, the *initialize* signal resets the encoder 300 as necessary. Encoder 300 receives information bits from main memory 290 and produces an encoded bit stream for further processing and eventual transmission to another network node (see, for example, FIG. 14).

[0056] In the embodiment of FIG. 3, encoder 300 includes at least three control circuits, referred to here as *gen_cc* 320, *par_cc* 325, and *m_cc* 330. An accumulator block 305 is used to generate FEC parity check bits according to the code in use for the current user. These components will be further described below. A generator matrix RAM 310 for storing generator matrices and a parity check bit RAM 315 for storing

generated parity check bits are also present. Note that other types of memory devices may be used, but are not presently preferred. Note also that reading and writing from the RAM may and in a preferred embodiment does occur at different clock edges (for example the rising edge as opposed to the falling edge.) In the illustrated embodiment a 2:1 switch 335 is used to select between information bits from main memory 290 and parity check bits from parity check bit for insertion into the encoded data stream.

[0057] In operation, information bits are received and stored in main memory 290. The *user_id* signal from scheduler 295 identifies the user for which encoder 300 is presently operating. The signal is provided not only to main memory 290 but to the control circuitry and the parity bit RAM 313 as well. The *initialize* signal is provided to the control circuitry and to the generator matrix RAM 310. In this embodiment, when the *initialize* signal is high at the beginning of an encoding cycle, the appropriate generator matrix is loaded into memory, and *initialize* remains low thereafter.

[0058] The control circuitry m_cc 330 is a main control circuit and transmits a signal *trans_par* that indicates when the 2:1 switch 335 should transmit information bits from main memory 290 and when parity bit from RAM 315 should be transmitted, for example when *trans_par* is high or low, respectively. Control circuitry gen_cc 320 controls access to the generator matrix RAM 310 and the control circuitry par_cc 325 controls access to parity check bit RAM 315.

[0059] In the embodiment of FIG. 3, the width of the bus interface with the main memory is given by b . In a preferred embodiment, the bus width b is chosen such that the number of information bits is an integer multiple α of the bus width. Consider U the set of users at a given time; user $u \in U$, k_u is the number of information bits, and p_u the number of parity bits for the user. The number of information bits for user u may then be described as $k_u = \alpha_{k,u} b$.

[0060] Preferably, the generator matrix RAM size is set to allow reprogramming with codes of different code length as desired. In the embodiment of FIG. 3, the size of the generator matrix RAM is given as $N_c \propto_{1,max} \times \alpha_{2,max} b^2$ and the size of the parity check bit RAM is and the size of the parity check bit RAM is $N_u \times p_{max}$, where N_c is the

number of codes for error correction and N_u is the maximum number of users that could be accessing a transmitted frame. The remaining terms of these expressions will be discussed in more detail below.

[0061] If the code rate for a user is $c_{r,u}$ then number of parity check bits may be given by $p_u = k_u(1 - c_{r,u})/c_{r,u}$. Since $k_u = \alpha_{k,u} b$, it can be stated that $p_u = \alpha_{p,u} b$ where $\alpha_{p,u} = \alpha_{k,u} (1 - c_{r,u})/c_{r,u}$. In this embodiment, it should be noted that the code parameters should be chosen such that $\alpha_{p,u}$ is an integer.

[0062] Further define $\alpha_{k,max} = \frac{k_{max}}{b}$, where k_{max} is the maximum number of information bits in a code block that the encoder will encounter and p_{max} the maximum number of parity check bits. If $\alpha_{p,max}$ is considered a constant of proportionality, then $p_{max} = \alpha_{p,max} b$. The generator matrix RAM size may therefore be expressed as $N_c \propto \alpha_{k,max} \times \alpha_{p,max} b^2$.

[0063] It should be noted that this expression is valid for the scenario where the chosen code set requires all columns of the generator matrix to be explicitly specified. Since the generator matrix RAM size has been determined taking reprogrammability into consideration, only $\alpha_{k,c}$ (equal to $\alpha_{k,u}$ for some given user u) generator matrix RAM rows per code will have useful data at any given time and only $\alpha_{p,c} b^2$ ($\alpha_{p,u} b^2$ for a given user u) bits in any row will have useful data.

[0064] Similarly, only N_a parity check RAM rows will have useful data at any given time, where N_a is the number of active users at that time. In other words, user u will have p_u (out of p_{max}) useful bits.

[0065] Finally, note that in some instances the generator matrix RAM size may be reduced by exploiting certain properties of the generator matrix (for example if the generator matrix consists of cyclic sub-matrices). The RAM size expression is given as $N_c \propto \alpha_{1,max} \times \alpha_{2,max} b^2$ to account for this.

[0066] FIG. 4 is a block diagram illustrating an accumulator, such as accumulator 305 shown in FIG. 3, according to one embodiment. In this embodiment, accumulator 305 receives rows of parity bit generator code from generator matrix RAM 312 as indicated by the signals *code_id*, which identifies code in the generator matrix RAM 312 associated with a particular user *u*, and *gen_ct*, which identifies a particular row of the stored generator code. (Note that in this embodiment, RAM 312 has not been reduced in size, as alluded to above, and for this reason is depicted as having the size of $N_c \propto_{k,max} \times p_{max}b$.)

[0067] For illustration generator matrix row 340 is shown in FIG. 4. The generator matrix code in row 340 is provided to adder blocks 345, which are individually referred to as adder block 1, adder block 2 ... adder block p_{max} (where p_{max} is the maximum number of parity bits that may be accommodated). Intermediate adder blocks, if any, are represented by broken lines. In adder blocks 345, the generator matrix code is processed with information bits from the main memory (not shown in FIG. 4) to produce parity bits 350, individually referred to as $p_1, p_2 \dots p_m$. Parity bits 350 are stored in parity bit RAM 315. In this embodiment, a $p_{max}:b$ switch 355 is also provided to select b out of p_{max} bits during parity bit transmission based on a *par_ct* signal.

[0068] FIG. 5 is a block diagram illustrating an adder block, such as adder block 1 shown in FIG. 3, according to one embodiment. In this embodiment, adder block 1 receives generator matrix code (see, for example, generator matrix row 340 shown in FIG. 4) at AND gates 360, which also take as input information bits *a* from main memory. The results of these AND operations are provided to a b bit XOR gate 365, and the results of that operation are provided to XOR gate 370.

[0069] In this embodiment, the output of 2:1 switch 375 is also provided to XOR gate 370 and the results of the XOR operation are provided to a one-bit register 380. The contents of one-bit register 380 are provided to a parity check RAM (see, for example, RAM 315 shown in FIG 4). As noted in FIG. 5, the register 380 is associated with a parity bit p_1 of the RAM. The inputs of the 2:1 switch 375 are a zero value and the value of one-bit register 385, which in turn comes from the parity check RAM. The 2:1 switch

375 is controlled by a signal *ctrad* from compare function 390. In this embodiment, the output of the compare function 395 is 1 when the signal *gen_ct* is 0. When *ctrad* equals 1, the 2:1 switch output is set to 0. By setting the *ctrad* signal, XOR operation 370 is performed with 0 (instead of the contents of the parity check RAM) when the first set of *b* bits from a new data block is received.

[0070] FIG. 6a is a block diagram illustrating a *gen_cc* control circuit, such as *gen_cc* 320 shown in FIG.3, according to one embodiment. As mentioned above, *gen_cc* is the generator RAM control circuitry. The generator matrix RAM control circuitry 320 consists of N_u counters that keep track of the number of blocks of *b* information bits downloaded from the main memory and transmitted per user. The counters store the *gen_ct* signal for this purpose.

[0071] In the embodiment of FIG. 6a, the *gen_cc* control circuitry 320 (and specifically RAM 402) receives *user_id* to identify the user whose data is being read from the main memory. The *initialize* signal allows the setting of the *gen_ct* signal for the user to 0 during the first cycle of operation. The values of $\alpha_{1,u}$ and *code_id* are also stored in memory during this cycle.

[0072] In this embodiment, RAM 402 also receives the output of 2:1 switch 406 (from g-bit register 404), which is either the incremented *gen_ct* count from increment function 408 or 0, as selected by signal *ctr2*. The *cmp_gen* and *ctr2* signals are used to reset *gen_ct* for the user to 0 after all the information bits in a code block have been transmitted. The *ctr2* signal is output from compare function 410, which compares the *cmp_gen* signal from compare function 412 with the inverse of the received *trans_par* signal. (Note that the $\parallel$ symbol, for example in compare function 410, represents an OR operation.) Compare function 412 compares the values *gen_ct* and $\alpha_{1,u} - 1$, which it receives from RAM 402 via register 414. The compare function 412 outputs 1 unless the *gen_ct* signal is equal to $\alpha_{1,u} - 1$, where $\alpha_{1,u}$ is the number of rows of the generator matrix for user *u*.

[0073] Note that in this embodiment, the *gen_ct* signal can have a value between 0 and $\alpha_{1,max} - 1$; hence it requires $g = \text{ceil}(\log_2(\alpha_{1,max}))$ bits for storage. In this embodiment, the counters store $\alpha_{1,u} - 1$ (using g bits), so that the comparison can be performed every clock cycle. Note also that the “inv” function of compare function 410 calculates the compliment of the *trans_par* signal as transmitted by the *m_cc* control circuitry. This sets *ctr* to 1 when parity check bits are being transmitted, holding the *gen_ct* signal at 0. The *gen_cc* control circuitry also stores the *code_id* signal associated with a user and this requires $g = \text{ceil}(\log_2(N_c))$ bits. Finally, note that in the general scenario where no reduction of the generator matrix RAM is performed, $\alpha_{1,u} = \alpha_{k,u}$ and $\alpha_{1,max} = \alpha_{k,max}$.

[0074] FIG. 6b is a block diagram illustrating a *par_cc* control circuit, such as *par_cc* 325 shown in FIG.3, according to one embodiment. The *par_cc* control circuit 325 controls the transmission of p_u parity bits, b bits at a time. In this control circuit, counters keep track of the number of transmitted parity bits and store a *par_ct* signal. The parity check RAM address line is generated using the *par_ct* signal (see, for example, parity check RAM 315 shown in FIG. 3).

[0075] In the embodiment of FIG. 6b, the *par_cc* control circuitry 322 (and specifically RAM 422) receives *user_id* to identify the user whose data is being read from the main memory. The *initialize* signal allows the setting of the *par_ct* signal for the user to 0 during the first cycle of operation.

[0076] In this embodiment, RAM 422 also receives the output of 2:1 switch 426 (from g_p -bit register 424), which is either the incremented *par_ct* count from increment function 428 or 0, as selected by signal *ctr4*. The *cmp_par* and *ctr4* signals are used to reset *par_ct* for the user to 0 after all the parity check bits in a code block have been transmitted. The *ctr4* signal is output from compare function 430, which compares the *par_gen* signal from compare function 422 with the received *trans_par* signal. Compare function 432 compares the values *par_ct* and $\alpha_{p,u} - 1$, which it receives from RAM 422 via register 434.

[0077] The compare function 432 outputs 1 when *par_ct* is equal to $\alpha_{p,u} - 1$ and 0 otherwise. The value of $\alpha_{p,u}$ is dependent on the code associated with the current user, so the counters also store $\alpha_{p,u} - 1$ (which uses g_p bits). The *cmp_par* and *ctr4* signals are used to reset *par_ct* to 0 after all the parity bits in a code block (for a given user) have been transmitted. The *trans_par* signal is used to ensure that *par_ct* for the user remains 0 when information bits are being transmitted. Note that the *par_ct* signal has a size of g_p , where $g_p = \text{ceil}(\log_2(\alpha_{p,max}))$. The *par_ct* signal assumes a value between 0 and $\alpha_{p,u} - 1$, where $\alpha_{p,u} = p_u/b$.

[0078] Note also that the configuration of the *par_cc* control circuit of FIG. 6b is generally the same in either the generalized or QC-LDPC scenario, described below.

[0079] FIG. 7 is a block diagram illustrating an *m_cc* control circuit, such as *m_cc* 330 shown in FIG. 3, according to one embodiment. In this embodiment, RAM 450 receives *user_id* to identify the user whose data is being read from the main memory, and also receives the output of 2:1 switch 442 (via one-bit register 452). RAM 450 outputs the *trans_par* signal (via one-bit register 454), which is used by various components in the flexible encoder (see, for example, FIG. 3). The *trans_par* signal 440 is also an input to 2:1 switch 440.

[0080] In the embodiment of FIG. 7, the output for the 2:1 switch 440 is low when the *cmp_gen* signal (see FIG. 6a) is high and the output for the 2:1 switch 442 is high when the *cmpm* signal is high. The 2:1 switch 442 receives the *cmpm* signal from compare function 444, which receives and compares the values of *cmp_par* and *initialize*. The *trans-par* signal generated by the *m_cc* control circuitry 330 is high when information bits are transmitted and low when parity bits are transmitted. This bit can therefore be used to stop the transmission of information bits (from main memory) when parity bits are being transmitted, as shown in FIG. 7. This can be achieved in a preferred embodiment by ensuring that the main memory pointer increment after information bit transmission is a function of the *trans_par* signal. The *trans_par* signal bit is set to one in the first cycle of operation using the *initialize* signal.

[0081] As alluded to above, in some scenarios a reduction in needed RAM (or other memory) size may be achievable, for example in an encoder/generator matrix with cyclic sub-matrices. The parity check matrix of a quasi-cyclic LDPC code consists of cyclic sub-matrices of size $s \times s$. The described architecture may be used for encoding a family of such LDPC codes with a constant s value and parity check matrices of full rank. The generator matrix of such a code for any user u is of size $(t_u - c_u) \times c_u s$ and consists of $(t_u - c_u)c_u$ cyclic sub-matrices of size $s \times s$. For such codes, the number of parity check bits for user u is $p_u = c_u s$ and the number of information bits is $k_u = (t_u - c_u)s$, where $l_u = k_u + p_u = t_u s$ is the code length.

[0082] Since the sub-matrices are cyclic, each column of a sub-matrix can be obtained by a cyclic shift of the previous column. As a result, the generator matrix RAM need store only one out of s columns. In the encoder design described herein, the number of rows s of the sub-matrix meets one of the following criteria: (a) s is an integer multiple of b or (b) s equals b or b is an integer multiple of s .

[0083] FIG. 8 is a block diagram illustrating an accumulator 505 according to one embodiment. Note that accumulator 505 is similar but not identical to the accumulator 305 depicted in FIG. 4. The accumulator 505 of FIG. 8 is applicable, for example, to the QC-LDPC scenario where s is an integer multiple of b . That is, this represents a case where conservation of RAM resources may be achieved.

[0084] In this embodiment, accumulator 505 receives rows of parity bit generator code from generator matrix RAM 314 as indicated by the signals $code_id \propto \alpha_{cy,max}$, which identifies code in the generator matrix RAM 314 associated with a particular user u , and gen_ct , which identifies a particular row of the stored generator code. (Note that in this embodiment, RAM 314 has been reduced in size, as alluded to above, and for this reason is depicted as having the size of $N_c \propto \alpha_{cy,max} \times c_{max} s$.)

[0085] For illustration generator matrix row 540 is shown in FIG. 8. The generator matrix code in row 540 is provided to adder blocks 545, which are individually referred to as adder blocks 1, 2 ... s , adder blocks $s+1$, $s+2$... $2s$, ... adder blocks $(c_{max} - 1)s =$

1, $(c_{max} - 1)s + 2 \dots c_{max}s$. Intermediate adder blocks, if any, are represented by broken lines. In adder blocks 545, the generator matrix code is processed with information bits from the main memory (not shown in FIG. 8) to produce parity bits 550, individually referred to as $p_{11}, p_{12} \dots p_{c_{max}s}$. Parity bits 550 are stored in parity check bit RAM 317.

[0086] In this embodiment, the generator matrix RAM has the size $N_c \alpha_{cy,max} \times c_{max}s$ and $p_{max} = c_{max}s$. For any user u , only $\alpha_{cy,u}$ rows of the generator matrix will have useful information. Each row will only have $g_{11} \dots g_{c_us}$ useful bits. Only p_u out of p_{max} parity check bits will have useful information.

[0087] In this embodiment, $\alpha_{cy,u} = t_u - c_u$ and $\alpha_{cy,max} = \max(t_u - c_u)$ for the family of codes under consideration (“max” representing the maximum value); $c_{max} = \max(c_u)$.

[0088] FIG. 9 is a block diagram illustrating an adder block, corresponding to an adder block shown in FIG. 8, according to one embodiment. It is noted that the adder block of FIG. 9 is similar but not identical to the adder block depicted in FIG. 5. In the embodiment of FIG. 9, $s:b$ switch 555 selects the appropriate b out of s input bits from the generator matrix RAM (*see, for example*, generator matrix row 540 shown in FIG. 4). The switch control signal $ctrl$ is generated using $\log_2(s/b)$ bit counter in the gen_cc control circuitry, described below.

[0089] The AND gates 560 received the generator matrix bits from $s:b$ switch 555, and also take as input information bits a from main memory. The results of these AND operations are provided to a b bit XOR gate 565, and the results of that operation are provided to XOR gate 570.

[0090] In this embodiment, the output of 2:1 switch 575 is also provided to XOR gate 570 and the results of the XOR operation are provided to a one-bit register 580. The contents of one-bit register 580 are provided to a parity check RAM (*see, for example*, RAM 317 shown in FIG 8). As noted in FIG. 9, the register 580 is associated with a parity bit p_{11} of the RAM. The inputs of the 2:1 switch 575 are a 0 value and the value of

one-bit register 585, which in turn comes from the parity check RAM. The 2:1 switch 575 is controlled by a signal *ctrad* from compare function 590. In this embodiment, the output of the compare function 590 is 1 when the signal *gen_ct* is 0. Both the *gen_ct* and *ctrl* signals from the *gen_cc* control circuitry are zero at the start of every new information block, which sets *ctrad* to one. Again, this ensures that XOR operation with zero, instead of the contents of the parity check RAM, is performed when the first set of *b* bits from a new data block is received.

[0091] FIG 10 is a block diagram illustrating an accumulator 605 according to one embodiment. Note that accumulator 605 is identical to the accumulator 505 depicted in FIG. 4. The embodiment of FIG. 10 is applicable, for example, to the QC-LDPC scenario where *b* is an integer multiple of *s*. That is, this represents another case where conservation of RAM resources may be achieved.

[0092] In this embodiment, accumulator 605 receives rows of parity bit generator code from generator matrix RAM 316 as indicated by the signals $code_id \times \alpha_{cy,max}(s/b)$, which identifies code in the generator matrix RAM 314 associated with a particular user *u*, and *gen_ct*, which identifies a particular row of the stored generator code. (Note that in this embodiment, RAM 316 has been reduced in size, as alluded to above, and for this reason is depicted as having the size of $N_c \alpha_{cy,max}(s/b) \times c_{max}b$.)

[0093] For illustration generator matrix row 640 is shown in FIG. 10. The generator matrix code in row 640 is provided to adder blocks 645, which are individually referred to as adder blocks 1, 2 ... *s*, adder blocks *s*+1, *s*+2 ... 2*s*, ... adder blocks $(c_{max} - 1)s = 1, (c_{max} - 1)s + 2 \dots c_{max}s$. Intermediate adder blocks, if any, are represented by broken lines. In adder blocks 645, the generator matrix code is processed with information bits from the main memory (not shown in FIG. 10) to produce parity bits 650, individually referred to as $p_{11}, p_{12} \dots p_{c_{max}s}$. Parity bits 650 are stored in parity bit RAM 318.

[0094] FIG. 11 illustrating an adder block, such as an adder shown in FIG. 10, according to one embodiment. It is noted that the adder of FIG. 11 is similar but not

identical to an adder block depicted in FIG. 9. In the embodiment of FIG. 11, the $s:b$ switch 555 of FIG. 9 is not required. The AND gates 561 receive the generator matrix and also take as input information bits a from main memory. The results of these AND operations are provided to a $b = \mu s$ bit XOR gate 566, and the results of that operation are provided to XOR gate 571; μ is equal to b/s and is therefore equal to one when s is equal to b .

[0095] In this embodiment, the output of 2:1 switch 576 is also provided to XOR gate 571 and the results of the XOR operation are provided to a one-bit register 581. The contents of one-bit register 581 are provided to a parity check RAM (see, for example, RAM 318 shown in FIG 10). As noted in FIG. 9, the register 580 is associated with a parity bit p_{12} of the RAM. The inputs of the 2:1 switch 576 are a 0 value and the value of one-bit register 585, which in turn comes from the parity check RAM. The 2:1 switch 576 is controlled by a signal $ctrad$ from compare function 591. The gen_ct signal is zero at the start of every information block. This sets the $ctrad$ signal to one when the first set of b bits when a new data block is received.

[0096] FIG. 12 is a block diagram illustrating a gen_cc control circuit, such as gen_cc 320 shown in FIG.3, according to another embodiment. In this embodiment, the gen_cc control circuit is applicable to quasi-cyclic scenario a (where s is an integer multiple of b). In the embodiment of FIG. 12, $\alpha_{cy,u} = t_u - c_u$. The gen_ct counters used for the generator matrix RAM address line assume a value 0 and $t_u - c_u - 1$ and are of size $g = \text{ceil}(\log_2(\alpha_{cy,max}))$. Note that the counters of FIG. 12 are similar to those of FIG. 6a except that they will be incremented only once every s/b times (using the $ctrl3a$ signal). This is attained by using another set of counters that also generate the value $ctrl$ (as used in FIG. 9). In the embodiment of FIG. 12, g_{ct} is the size of the $ctrl$ signal in bits and is $\text{ceil}(\log_2(s/b))$. In the case of quasi-cyclic scenario b (where s is equal to b or b is an integer multiple of s), the configuration of FIG. 6a may be used with $\alpha_{1,u}$, in this case given by $(t_u - c_u) s/b$ and $\alpha_{1,max} = \max((t_u - c_u) s/b)$.

[0097] Note that the control circuitry for transmitting the parity bits b bits at a time is represented by par_cc (shown in FIG. 3). The configuration in the general scenario (as

shown in FIG 6b and 7) is applicable to both quasi-cyclic scenarios (a and b). The control circuitry represented by m_cc (see FIG. 3) allows alternating between transmitting information bits and parity check bits for any given user.

[0098] Note that FIGS. 1-14 illustrate selected components according to their respective embodiments and some variations are described above. Other variations are possible without departing from the claims of the invention as there recited. In some of these embodiments, for example, illustrated components may be integrated with each other or divided into subcomponents. There will often be additional components in the network node and in some cases fewer. The illustrations components may also perform other functions in addition to those described above, and some of the functions may alternately be performed elsewhere than described in these examples.

[0099] FIG. 15 is a flow diagram illustrating a method 700 according to one embodiment. At START it is presumed that the components necessary to performing the method are present and operation at least according to this embodiment. The process then begins when information bits for transmission are received (step 705). A determination is made (step 710) of the intended destination for the information bits, and an FEC code is selected (step 715) accordingly.

[00100] In this embodiment, a flexible encoder is then initialized (step 720) and a generator matrix is provided to a flexible encoder (step 725). Received information bits are then provided to the encoder (step 730). The flexible encoder then processes the information bits with the provided generator matrix and produces parity check bits (step 735). The parity check bits are stored in a parity check bit memory (step 740). The parity check bits are then combined with information bits to form an encoded bit stream (step 745).

[00101] In the preferred embodiment of FIG. 15, the data is being transmitted from an OLT in a PON. The encoded bit stream is then interleaved (step 750) with bit streams for other destinations. The interleaved bit stream is then scrambled (step 755) and provided to a transmitter for transmitting (step 760) toward the intended destination or destinations.

[00102] Note that the sequence of message flow illustrated in FIG. 15 represents an exemplary embodiment; some variation is possible within the spirit of the invention. For example, additional messaging may be added to that shown in FIG. 15, and in some implementations one or more of the illustrated messages may be omitted. In addition, the messages of the method may be transmitted and received in any logically-consistent order unless a definite sequence is recited in a particular embodiment.

[00103] In some embodiments, certain aspects of the techniques described above may be implemented by one or more processors of a processing system executing software. The software comprises one or more sets of executable instructions stored or otherwise tangibly embodied on a non-transitory computer readable storage medium. The software can include the instructions and certain data that, when executed by the one or more processors, manipulate the one or more processors to perform one or more aspects of the techniques described above. The non-transitory computer readable storage medium can include, for example, a magnetic or optical disk storage device, solid state storage devices such as Flash memory, a cache, random access memory (RAM) or other non-volatile memory device or devices, and the like. The executable instructions stored on the non-transitory computer readable storage medium may be in source code, assembly language code, object code, or other instruction format that is interpreted or otherwise executable by one or more processors.

[00104] A computer readable storage medium may include any storage medium, or combination of storage media, accessible by a computer system during use to provide instructions and/or data to the computer system. Such storage media can include, but is not limited to, optical media (e.g., compact disc (CD), digital versatile disc (DVD), Blu-Ray disc), magnetic media (e.g., floppy disc, magnetic tape, or magnetic hard drive), volatile memory (e.g., random access memory (RAM) or cache), non-volatile memory (e.g., read-only memory (ROM) or Flash memory), or microelectromechanical systems (MEMS)-based storage media. The computer readable storage medium may be embedded in the computing system (e.g., system RAM or ROM), fixedly attached to the computing

system (e.g., a magnetic hard drive), removably attached to the computing system (e.g., an optical disc or Universal Serial Bus (USB)-based Flash memory), or coupled to the computer system via a wired or wireless network (e.g., network accessible storage (NAS)). A computer readable storage medium may also be a propagating signal if recited explicitly in a particular embodiment.

[00105] Benefits, other advantages, and solutions to problems have been described above with regard to specific embodiments. However, the benefits, advantages, solutions to problems, and any feature(s) that may cause any benefit, advantage, or solution to occur or become more pronounced are not to be construed as a critical, required, or essential feature of any or all the claims. Moreover, the particular embodiments disclosed above are illustrative only, as the disclosed subject matter may be modified and practiced in different but equivalent manners apparent to those skilled in the art having the benefit of the teachings herein. No limitations are intended to the details of construction or design herein shown, other than as described in the claims below. It is therefore evident that the particular embodiments disclosed above may be altered or modified and all such variations are considered within the scope of the disclosed subject matter. Accordingly, the protection sought herein is as set forth in the claims below.

[00106] Although multiple embodiments of the present invention have been illustrated in the accompanying Drawings and described in the foregoing Detailed Description, it should be understood that the present invention is not limited to the disclosed embodiments, but is capable of numerous rearrangements, modifications and substitutions without departing from the invention as set forth and defined by the following claims.

WHAT IS CLAIMED IS:

1. Apparatus for data transmission, comprising:
a processor comprising a scheduler;
a flexible encoder configured to employ a plurality of FEC codes; and
a main memory device in communication with the encoder and the processor for receiving information bits and providing them to the flexible encoder as directed by the scheduler.
2. The apparatus of claim 1, wherein the flexible encoder comprises:
a generator matrix memory for populating with generator matrices associated with the plurality of FEC codes;
an accumulator configured for receiving generator matrices from the generator matrix memory and processing information bits from the main memory to produce parity check bits; and
a parity check bit memory for storing results from the accumulator.
3. The apparatus of claim 2, wherein the flexible encoder further comprises:
a generator matrix control circuit for controlling operation of the generator matrix memory; and
a parity check bit memory control circuit for controlling operation of the parity check bit memory.
4. The apparatus of claim 3, wherein the flexible encoder further comprises a 2:1 switch configured to receive information bits from the main memory and parity check bits from the parity check bit memory and to selectively include them in an encoded bit stream in response to a received control signal.
5. The apparatus of claim 4, wherein the flexible encoder further comprises a main control circuit configured at least to provide the control signal to the 2:1 switch.

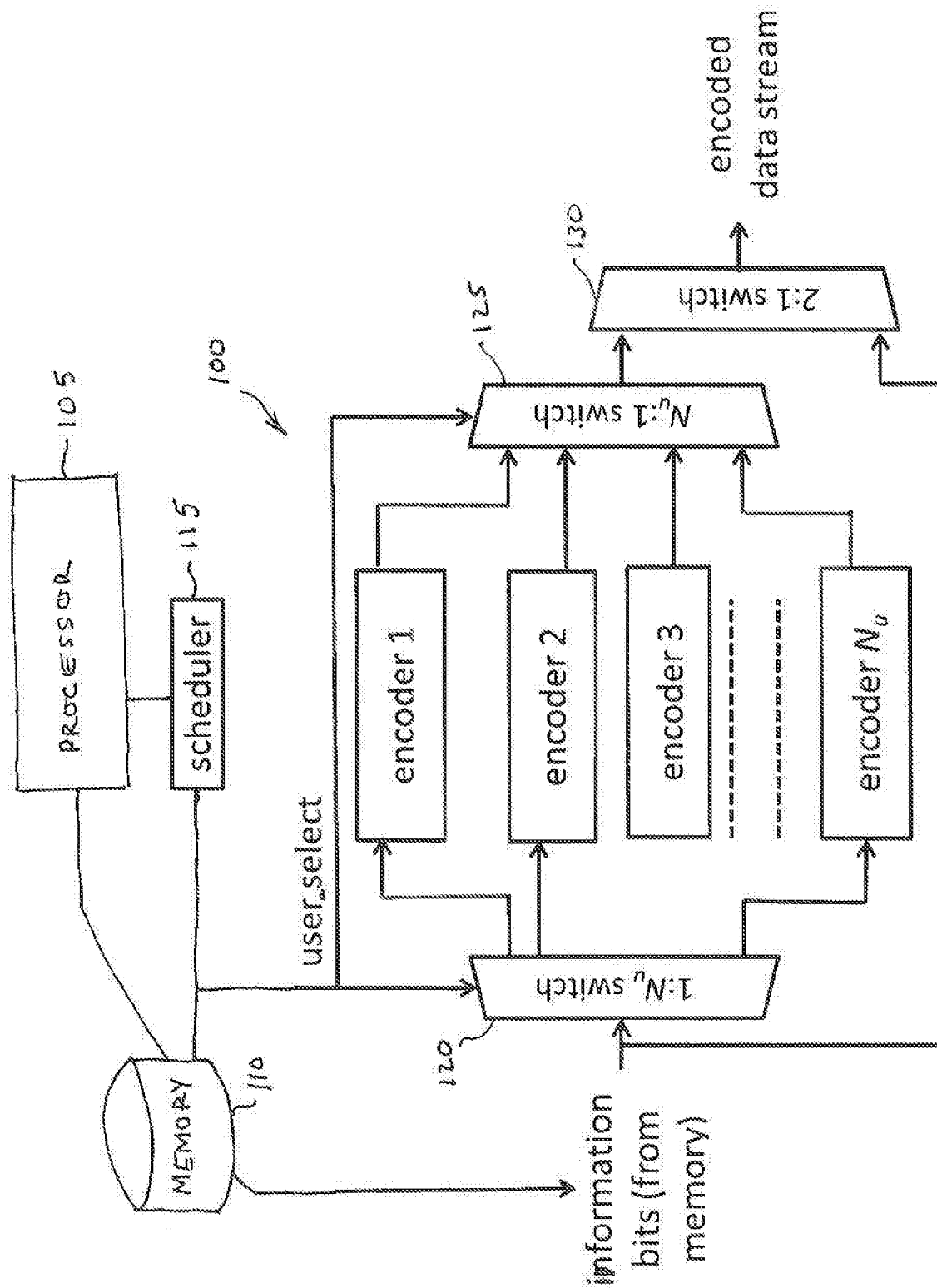
6. The apparatus of claim 5, wherein the main control circuit is further configured to generate a control signal to regulate the reading of information bits from the main memory.

7. The apparatus of claim 1, further comprising an interleaver for interleaving the encoded bit stream and a scrambler for scrambling the interleaved bit stream.

8. The apparatus of claim 1, wherein the apparatus is an OLT (optical line terminal) in a PON (passive optical network).

9. A method for data transmission, comprising:
receiving information bits for transmission;
selecting an FEC code for encoding the information bits;
initializing a flexible encoder;
providing to the flexible encoder a generator matrix associated with the selected FEC code; and
providing the information bits to the flexible encoder for encoding.

10. The method of claim 9, further comprising:
processing the information bits in an accumulator using the provided generator matrix to produce parity check bits;
storing the parity check bits in a parity check bit memory; and
combining parity check bits from the parity check bit memory with information bits to produce an encoded bit stream.

FIG. 1

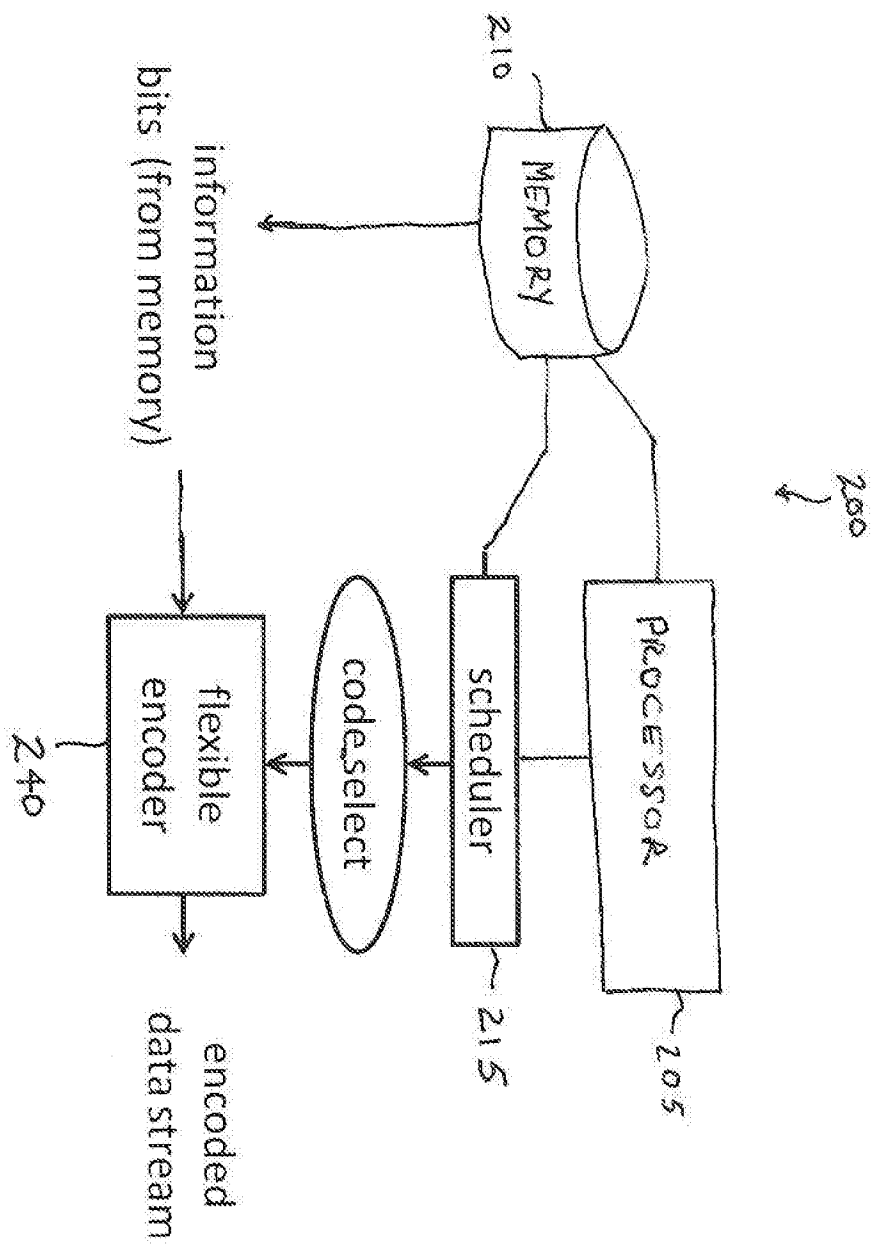
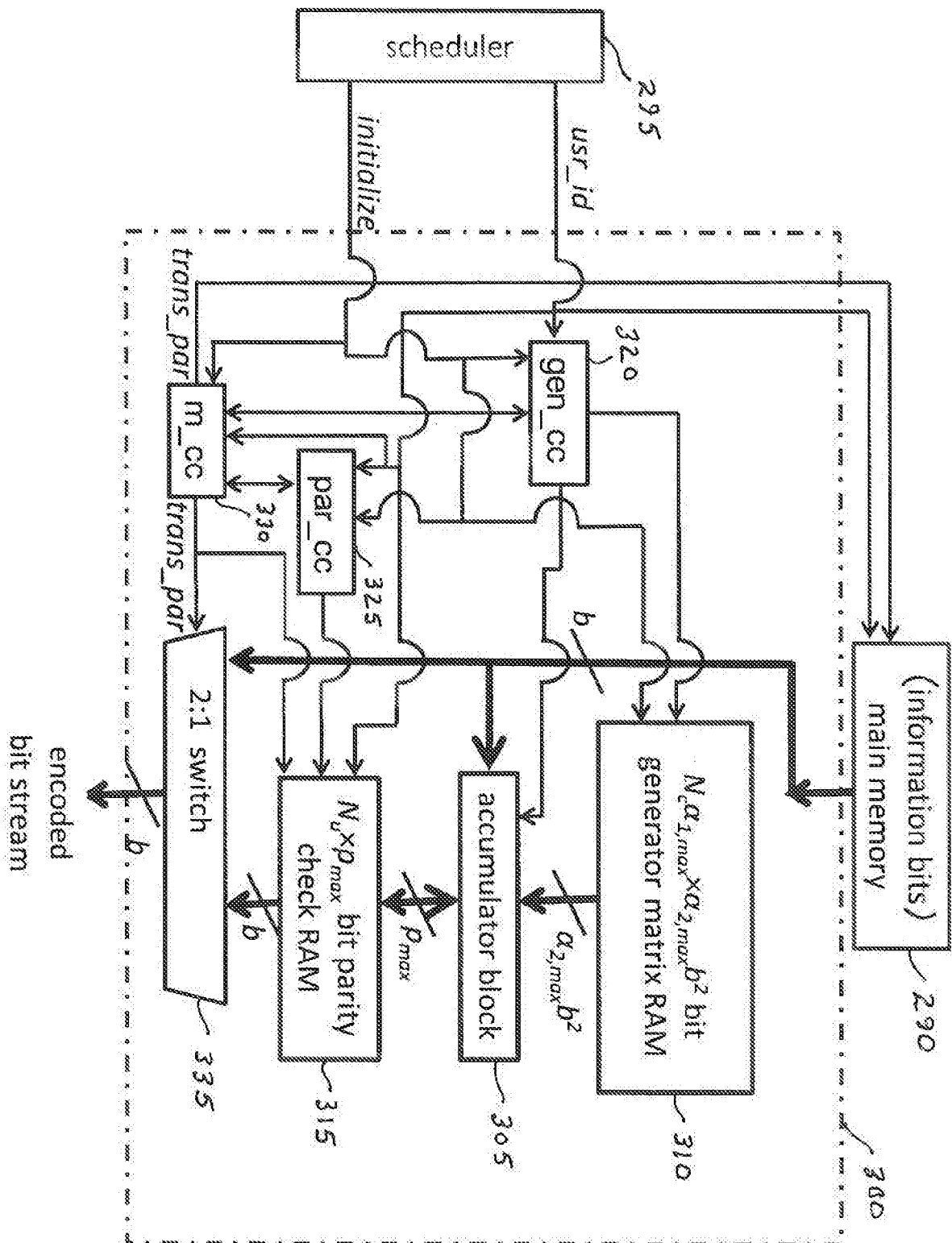


FIG. 2

FIG. 3



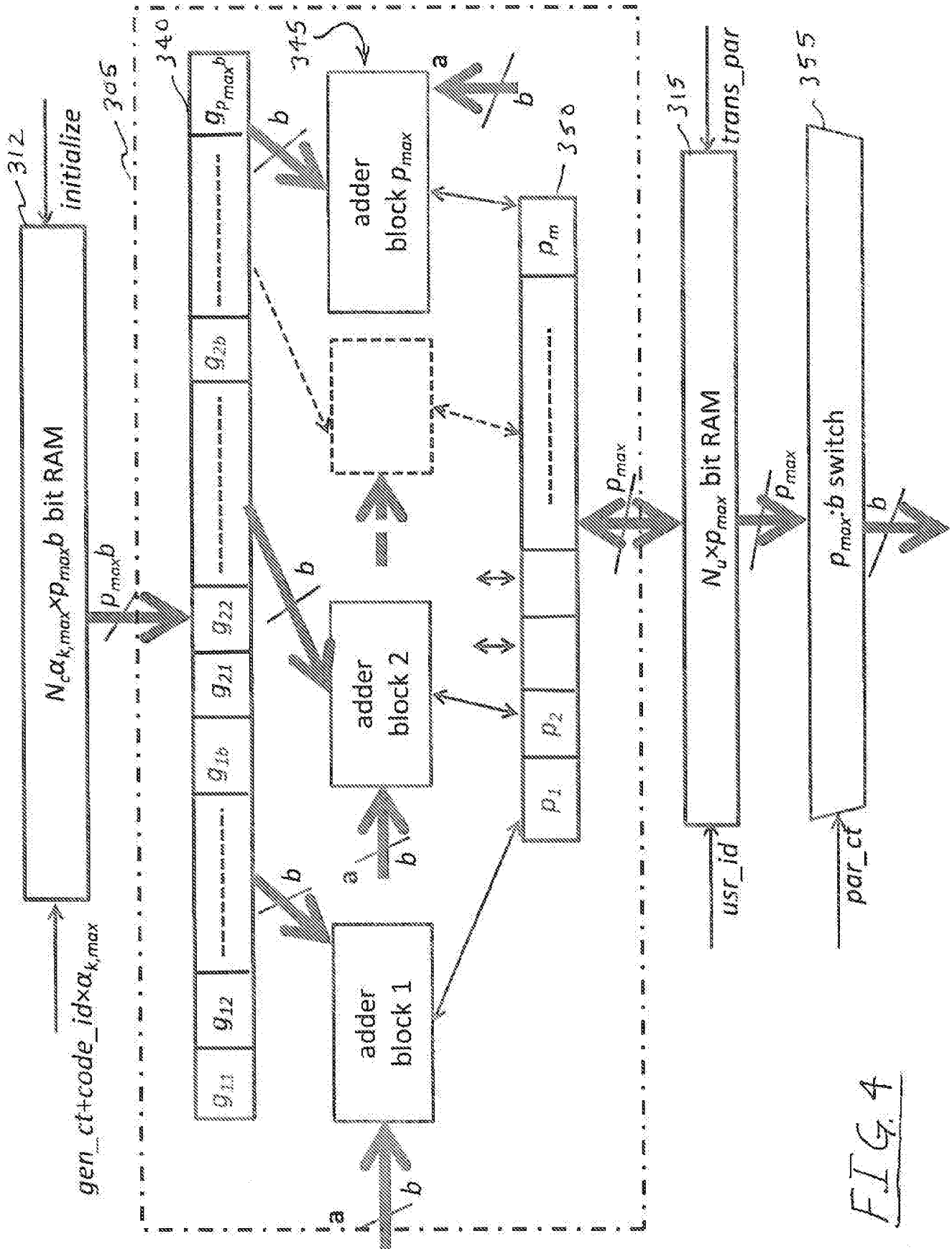


FIG. 4

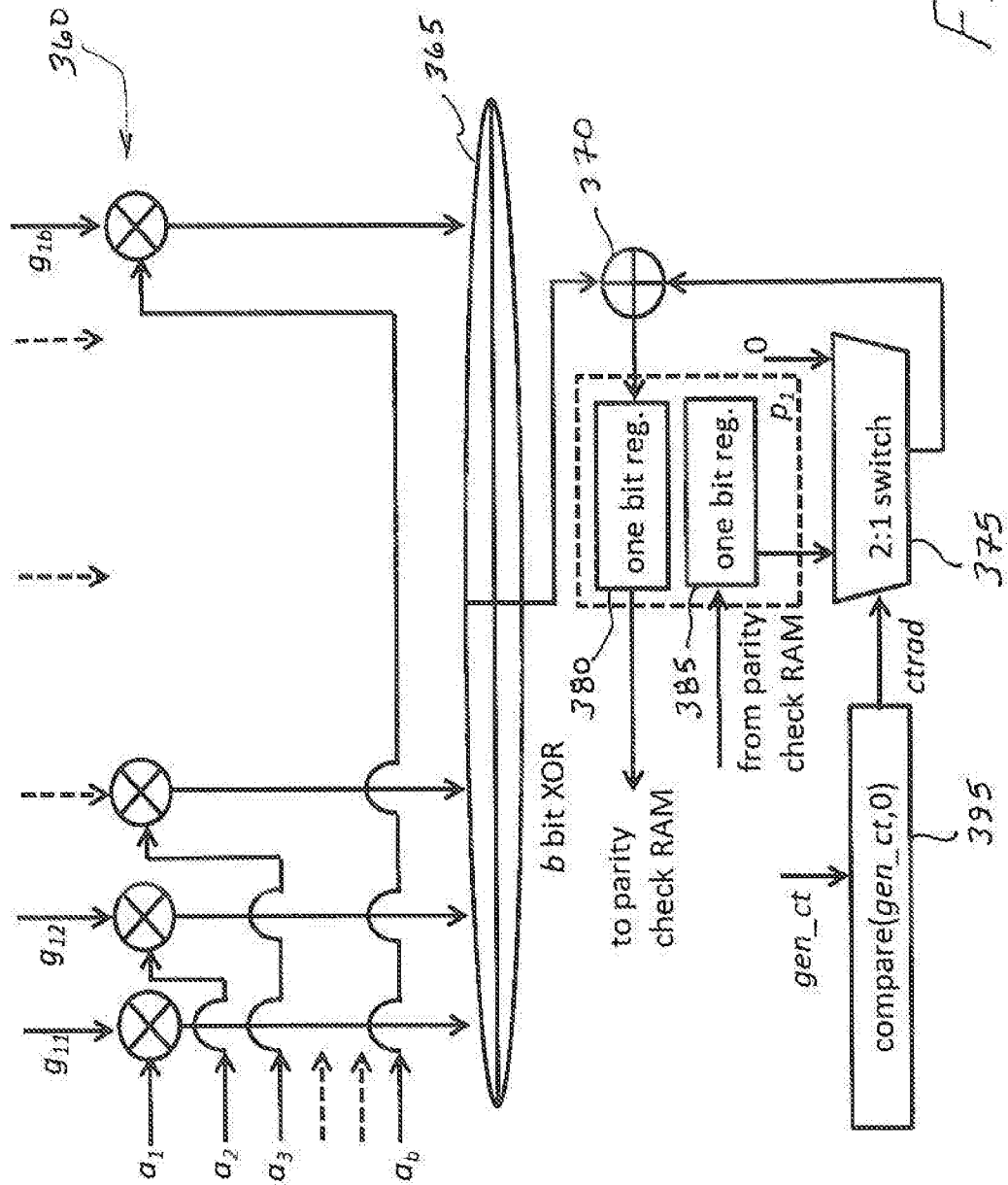
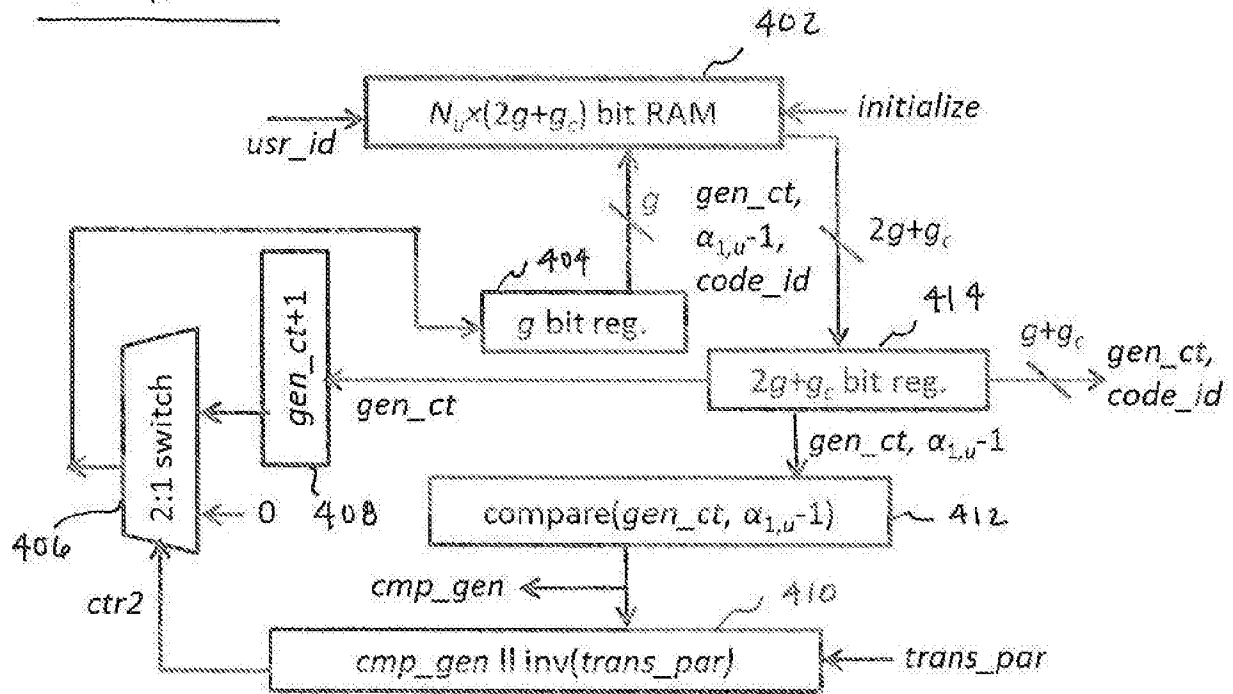
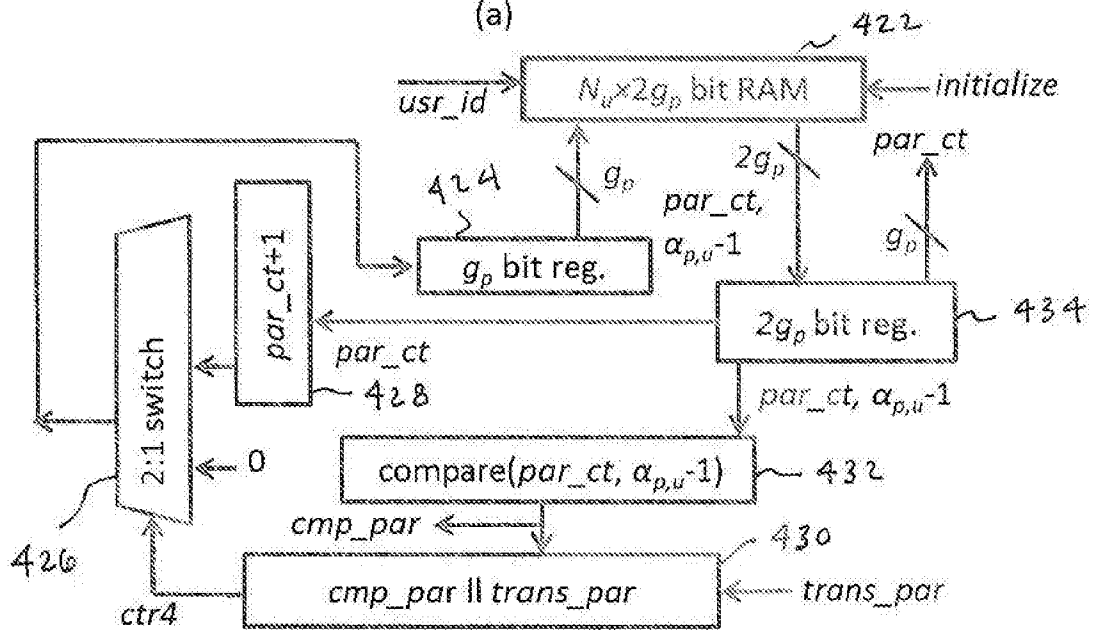


FIG. 5

FIG. 6a

(a)



(b)

FIG. 6b

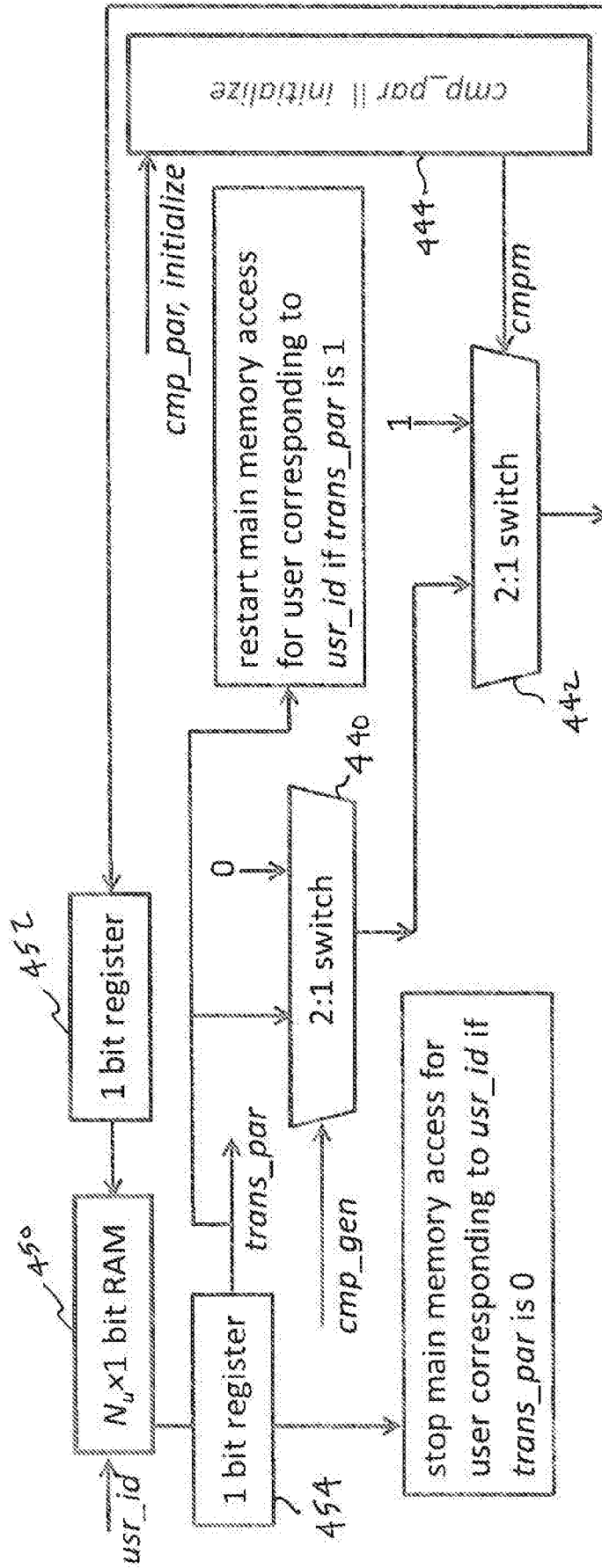


FIG. 7

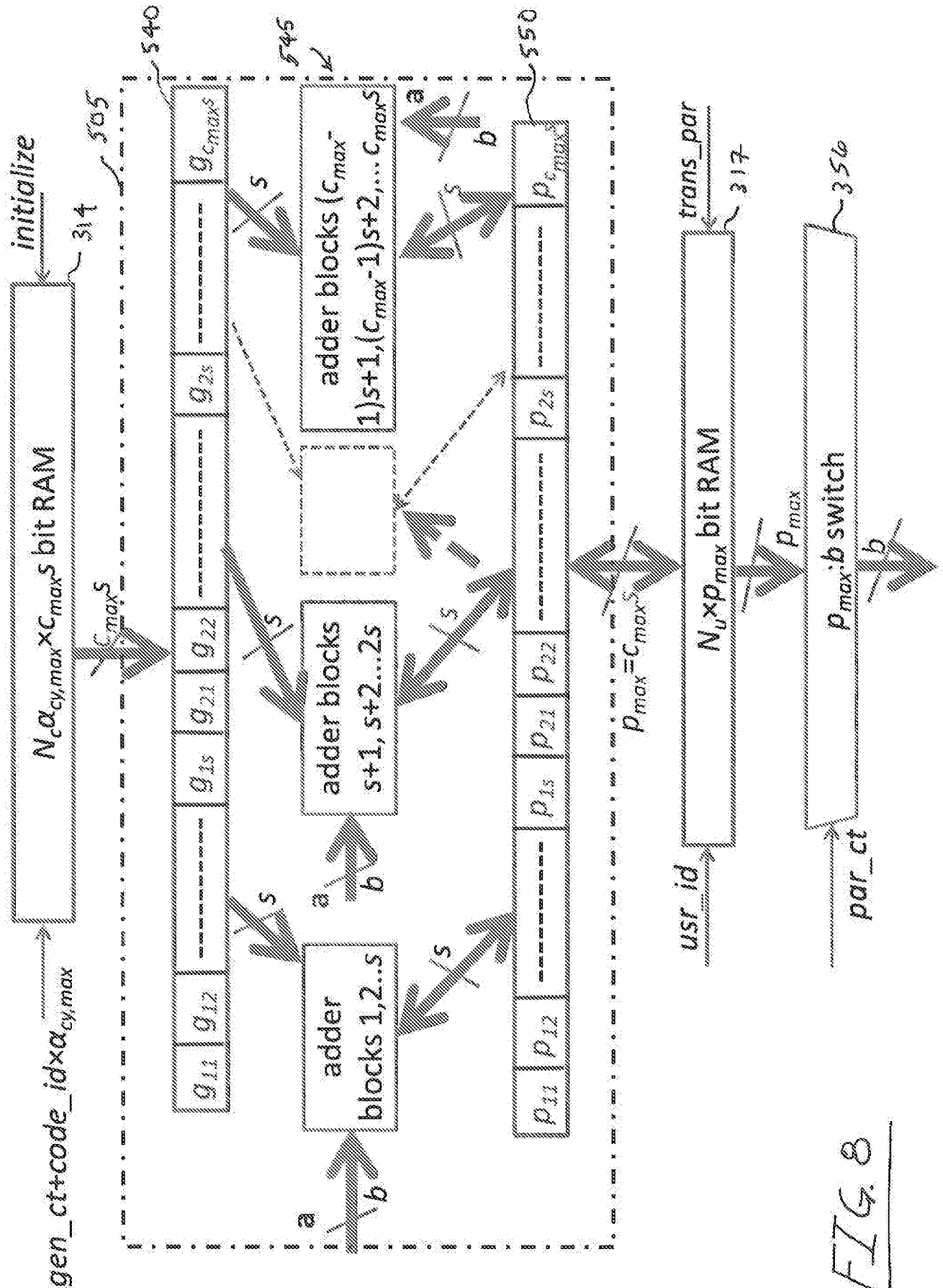
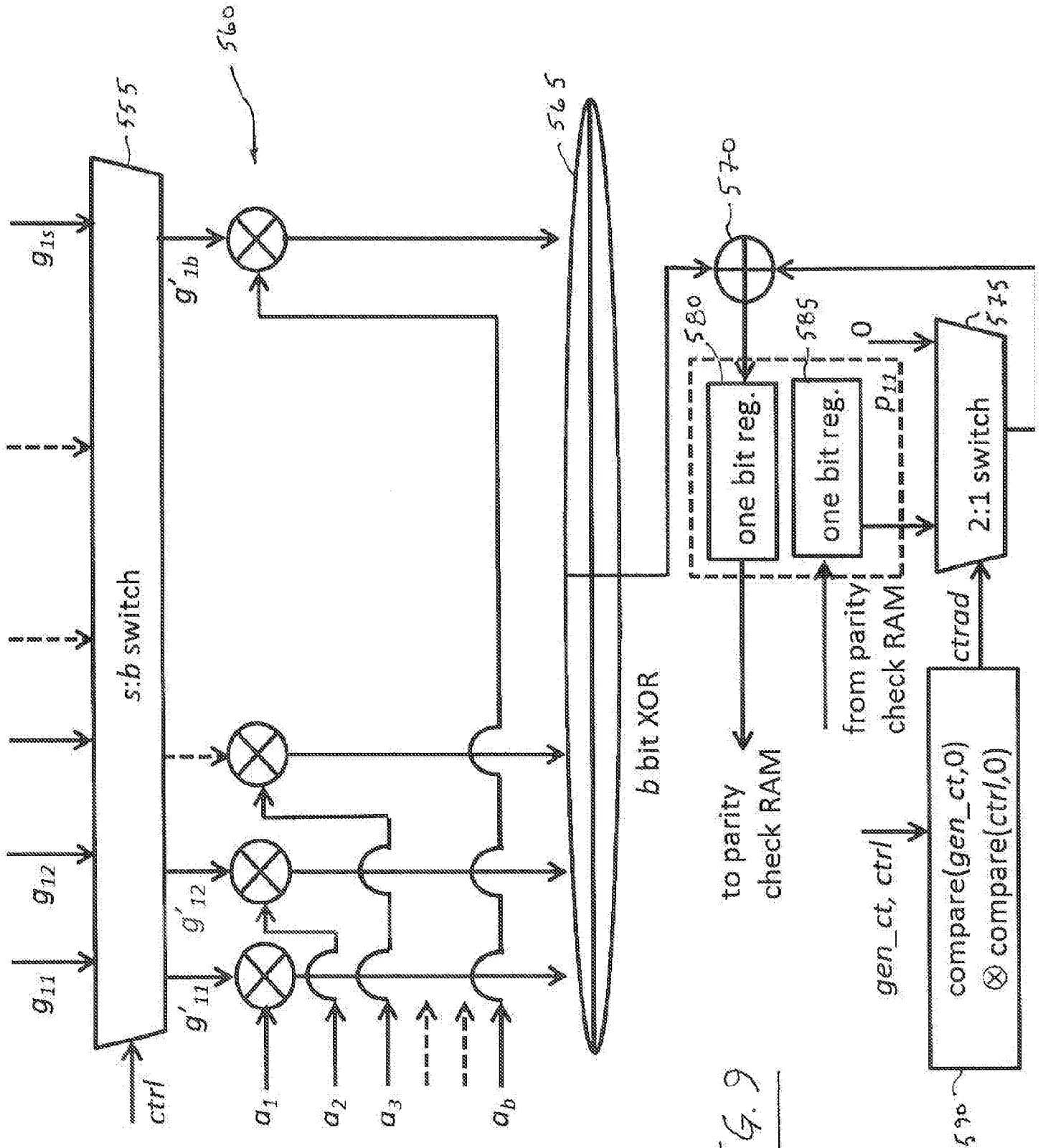


FIG. 8



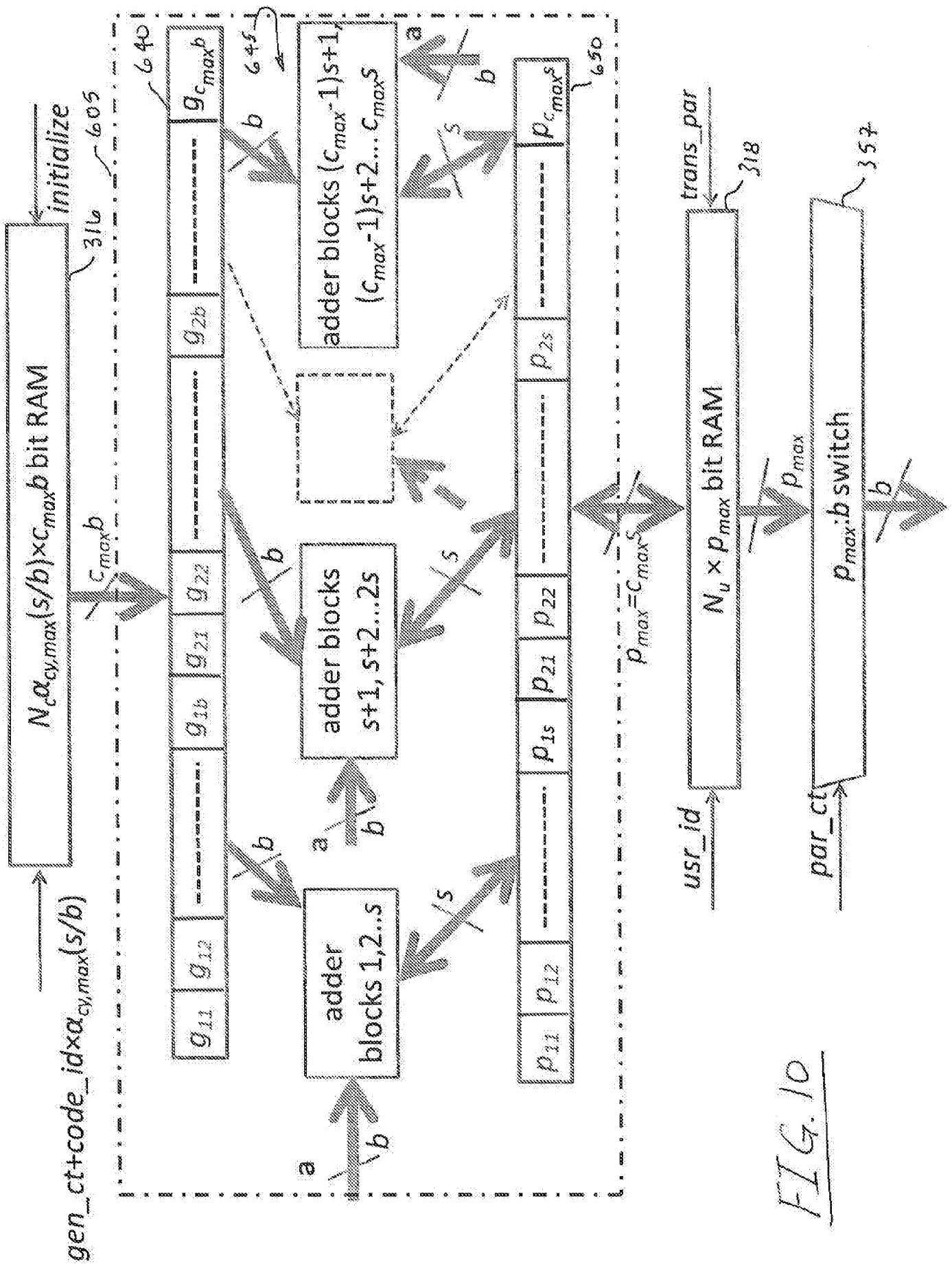


FIG. 10

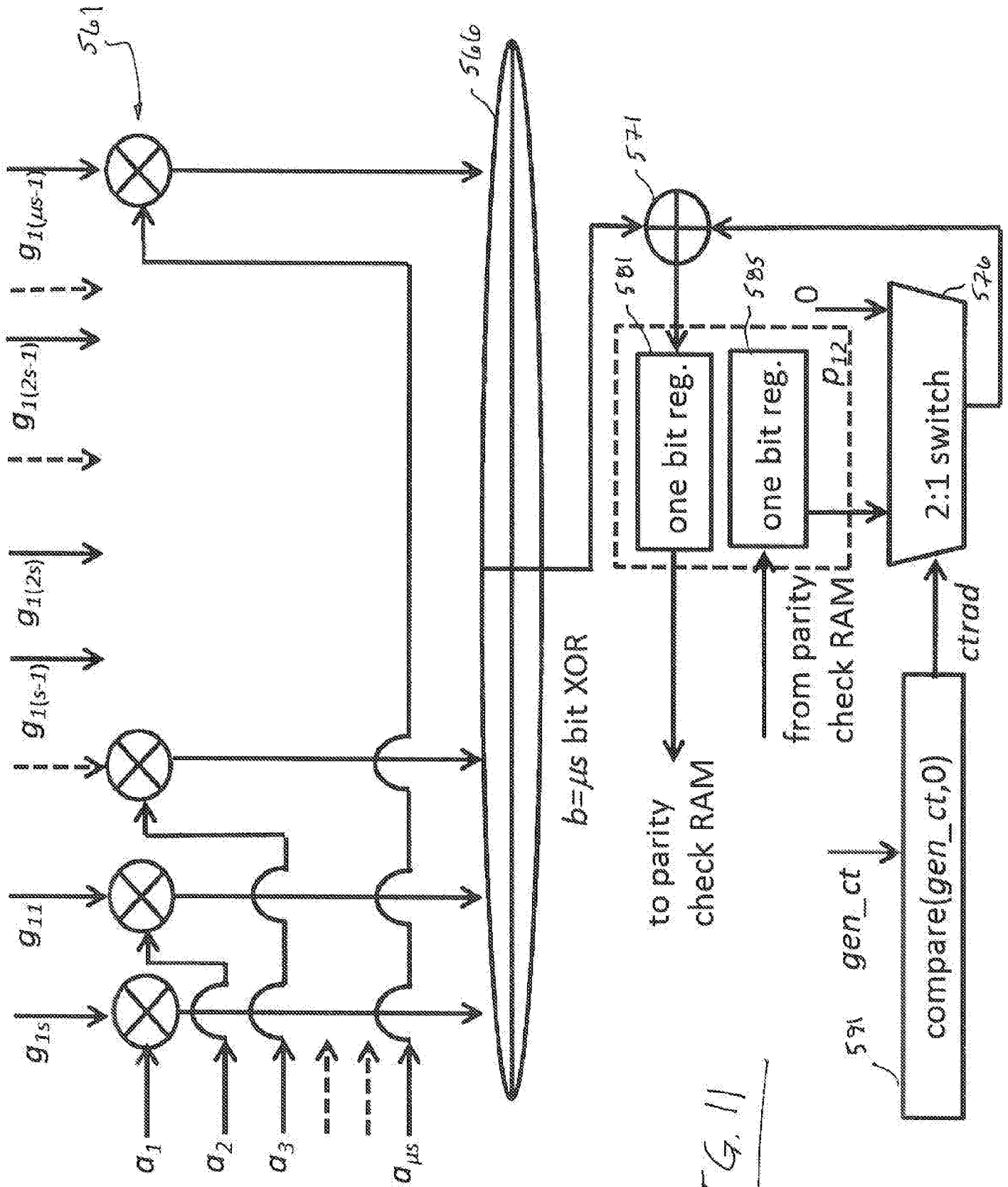
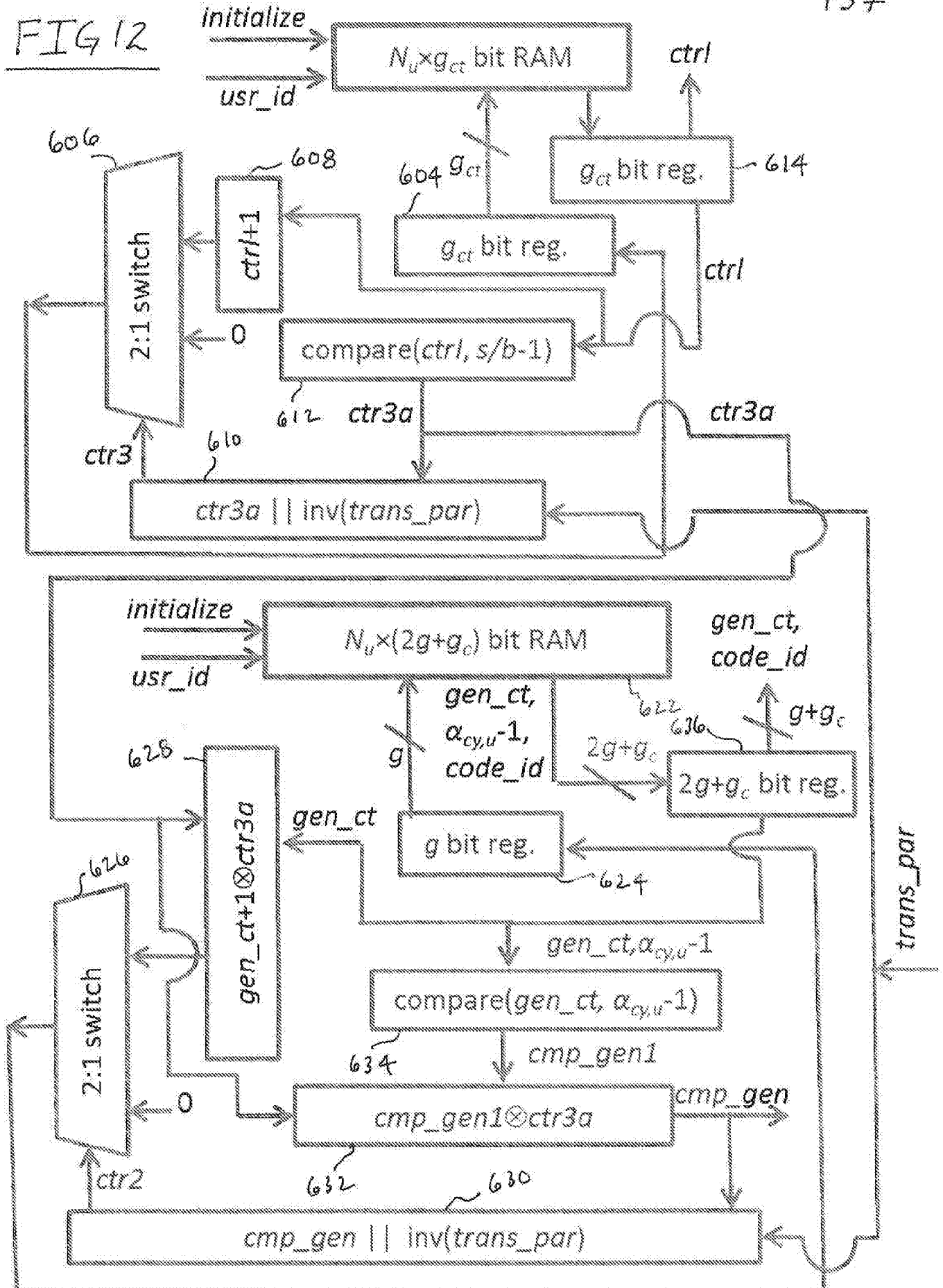


FIG. 11

137

FIG 12



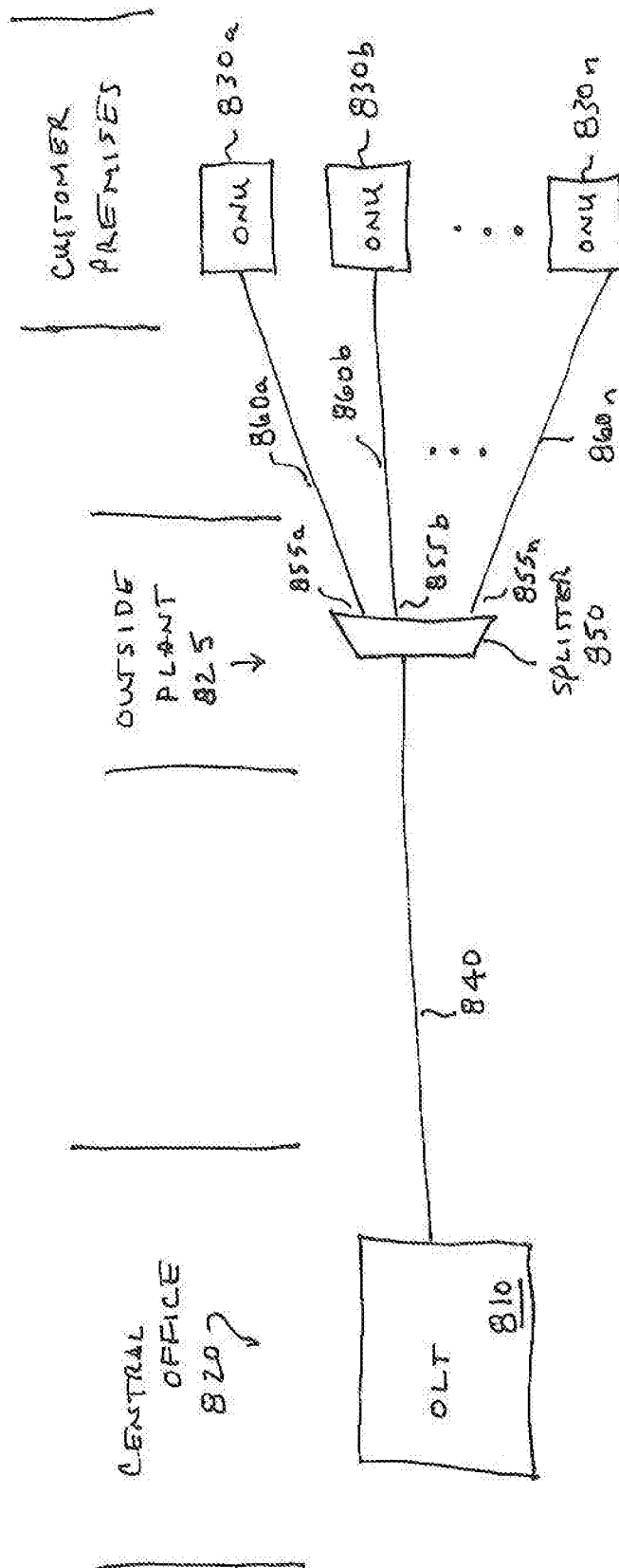


FIG. 13

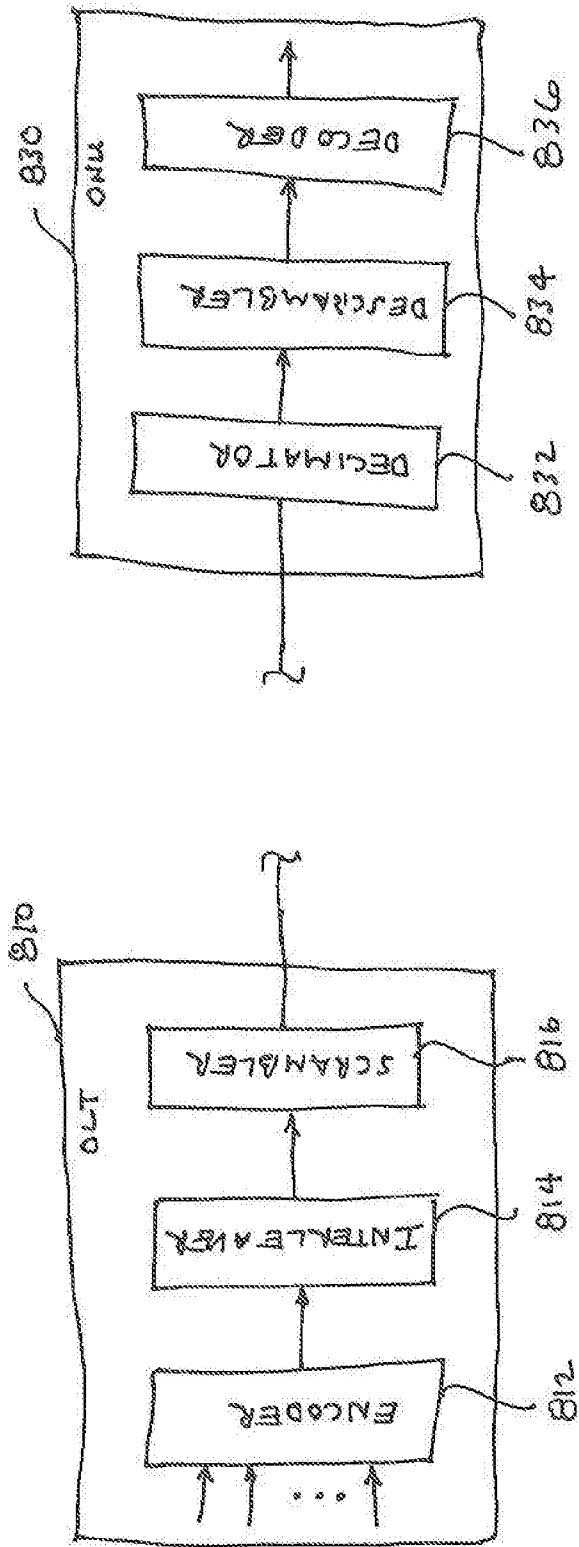
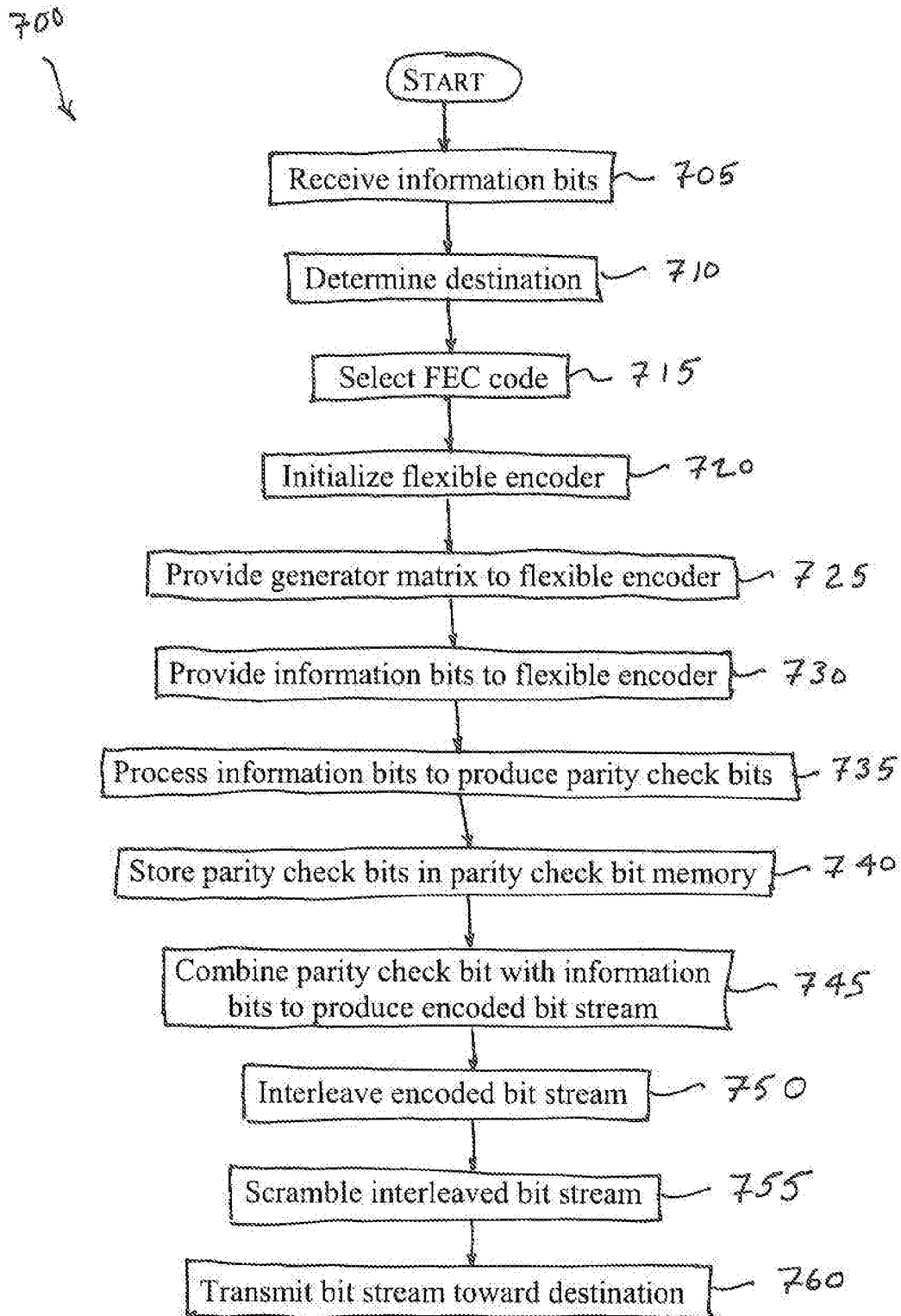


FIG. 14

FIG. 15

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2016/023904

A. CLASSIFICATION OF SUBJECT MATTER
 INV. H03M13/11
 ADD.

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

B. FIELDS SEARCHED

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

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ZHIXING YANG ET AL: "A fast and efficient encoding structure for QC-LDPC codes", PROC., IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, CIRCUITS AND SYSTEMS, ICCAS 2008, IEEE, PISCATAWAY, NJ, USA, 25 May 2008 (2008-05-25), pages 16-20, XP031352867, DOI: 10.1109/ICCCAS.2008.4657717 ISBN: 978-1-4244-2063-6 page 16 - page 19 ----- -/--	1-10



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

21 June 2016

Date of mailing of the international search report

05/07/2016

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040,
 Fax: (+31-70) 340-3016

Authorized officer

Offer, Elke

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2016/023904

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	QI CHEN ET AL: "A general folded encoding structure for quasi-cyclic LDPC codes", PROC., IEEE 4TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND SERVICE SCIENCE, IEEE, 27 June 2014 (2014-06-27), pages 1011-1014, XP032667607, ISSN: 2327-0586, DOI: 10.1109/ICSESS.2014.6933736 ISBN: 978-1-4673-4997-0 [retrieved on 2014-10-21] page 1011 - page 1012 -----	1-10
X	US 2008/028274 A1 (LIN SHU [US]) 31 January 2008 (2008-01-31) abstract paragraph [0060] - paragraph [0064] paragraph [0068] - paragraph [0074]; figures 3,4-7,9 -----	1-10
X	US 8 504 894 B1 (ZENG LINGQI [US] ET AL) 6 August 2013 (2013-08-06) column 6 - column 8; figures 3,5,6,7 -----	1-10
A	Mohammed El-Hajjar ET AL: "A Survey of Digital Television Broadcast Transmission Techniques", IEEE Communications Surveys & Tutorials, 1 January 2013 (2013-01-01), pages 1924-1949, XP055282247, New York DOI: 10.1109/SURV.2013.030713.00220 Retrieved from the Internet: URL:http://ieeexplore.ieee.org/ielx7/9739/6661627/06492307.pdf?tp=&arnumber=6492307&isnumber=6661627 [retrieved on 2016-06-21] page 1939, right-hand column - page 1940, left-hand column -----	7

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2016/023904

Patent document cited in search report	Publication date	Patent family member(s)	Publication date	
US 2008028274	A1	31-01-2008	US 2008028274 A1	31-01-2008
			US 2008126908 A1	29-05-2008

US 8504894	B1	06-08-2013	NONE	
