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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 November 2010 (04.11.2010)

PCT

(10) International Publication Number
WO 2010/126482 A1

(51) International Patent Classification:
G11B 20/10 (2006.01) *G11B 20/14* (2006.01)

(21) International Application Number:
PCT/US2009/041867

(22) International Filing Date:
28 April 2009 (28.04.2009)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **LSI CORPORATION** [US/US]; 1621 Barber Lane, Milpitas, CA 95035 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **TAN, Weijun** [CN/US]; 4123 Ravenna Place, Longmont, CO 80503 (US). **YANG, Shaohua** [CN/US]; 4235 Voltaire Street, Santa Clara, CA 95135 (US). **FITZPATRICK, Kelly** [US/US]; 17 Run Brook Circle, Sudbury, MA 01776 (US). **LI, Zongwang** [CN/US]; 3873 Jasmine Circle, San Jose, CA 95135 (US). **ZHONG, Hao** [CN/US]; 1517 Woodmeadow Court, San Jose, CA 95131 (US).

(74) Agent: **HAMILTON, Douglas, M.**; 8601 W. Cross Dr., F5-301, Littleton, CO 80123 (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, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE,

[Continued on next page]

(54) Title: SYSTEMS AND METHODS FOR DYNAMIC SCALING IN A READ DATA PROCESSING SYSTEM

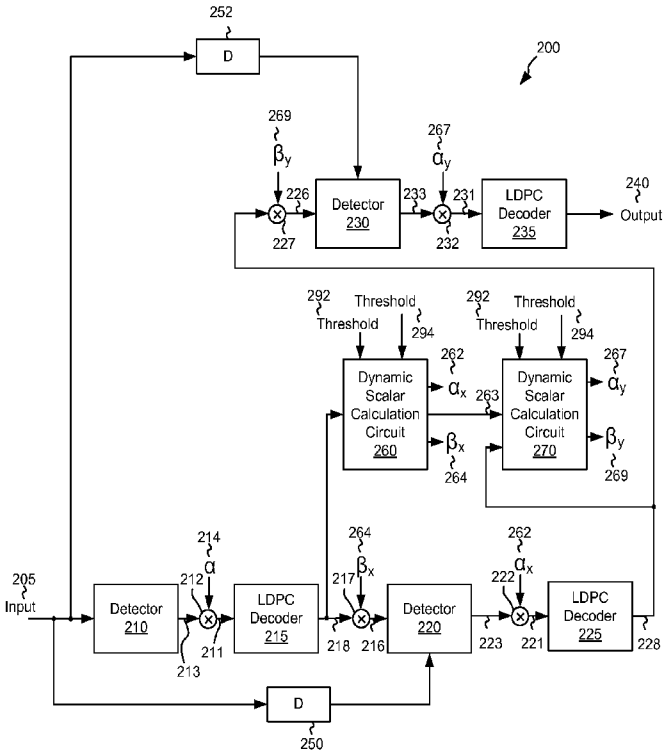


Fig. 2

(57) Abstract: Various embodiments of the present invention provide systems and methods for data processing. For example, a data processing system is disclosed that includes a decoder circuit providing a decoded output, and a dynamic scalar calculation circuit that determines a first dynamic scaling value and a second dynamic scaling value based at least in part on the decoded output. A first multiplier circuit multiplies the decoded output by the first dynamic scaling value and provides a first scaled output. A detector circuit receives the first scaled output and provides a detected output. A second multiplier circuit multiplies the detected output by the second dynamic scaling value and provides a second scaled output.

WO 2010/126482 A1

ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

Systems and Methods for Dynamic Scaling in a Read Data Processing System

BACKGROUND OF THE INVENTION

[0001] The present inventions are related to systems and methods for detecting and/or decoding information, and more particularly to systems and methods for performing multiple data detection and decode processes.

[0002] Various data transfer systems have been developed including storage systems, cellular telephone systems, and radio transmission systems. In each of the systems data is transferred from a sender to a receiver via some medium. For example, in a storage system, data is sent from a sender (i.e., a write function) to a receiver (i.e., a read function) via a storage medium. The effectiveness of any transfer is impacted by any data losses caused by various factors. In some cases, an encoding/decoding process is used to enhance the ability to detect a data error and to correct such data errors. As an example, a simple data detection and decode may be performed, however, such a simple process often lacks the capability to converge on a corrected data stream.

[0003] To heighten the possibility of convergence, various existing processes utilize two or more detection and decode iterations. Turning to Fig. 1, an exemplary prior art two stage data detection and decode circuit 100 is depicted. Two stage data detection and decode circuit 100 receives a data input 105 that is applied to a detector 110. A hard and soft output from detector 110 is provided to an LDPC decoder 115. Input 105 is fed forward via a buffer 130 to another detector 120. Detector 120 uses a soft output of LDPC decider 115 and input 105 to perform an additional data detection process. A hard and soft output from detector 120 is provided to an LDPC decoder 125 that performs a second decoding process and provides an output 135. Where the initial detection and decode provided by detector 110 and LDPC decoder 115 does not converge, the subsequent detection and decode provided by detector 120 and LDPC decoder 125 provide an additional opportunity to converge. In these systems, various scalars are designed into the systems to massage data between various stages of the process. These scalars are either

fixed at design time or are programmable static values that can be adjusted once the system is deployed in a storage medium. In some cases, however, data from one stage to another stage saturates limiting the capability of the system.

[0004] Hence, for at least the aforementioned reasons, there exists a need in the art for advanced systems and methods for data processing.

BRIEF SUMMARY OF THE INVENTION

[0005] The present inventions are related to systems and methods for detecting and/or decoding information, and more particularly to systems and methods for performing iterative data decoding and/or detection.

[0006] Various embodiments of the present invention provide data processing systems. Such data processing systems include a decoder circuit providing a decoded output, and a dynamic scalar calculation circuit that determines a first dynamic scaling value and a second dynamic scaling value based at least in part on the decoded output. A first multiplier circuit multiplies the decoded output by the first dynamic scaling value and provides a first scaled output. A detector circuit receives the first scaled output and provides a detected output. A second multiplier circuit multiplies the detected output by the second dynamic scaling value and provides a second scaled output.

[0007] In some instances of the aforementioned embodiments, the dynamic scalar calculation circuit determines a number of saturated outputs in the decoded output. In some cases, the first dynamic scaling value is set to a first default value when the number of saturated outputs exceeds a threshold value; the first dynamic scaling value is set to a minimum value when the number of saturated outputs is less than the threshold value; the second dynamic scaling value is set to a second default value when the number of saturated outputs exceeds the threshold value, and the second dynamic scaling value is set to a maximum value when the number of saturated outputs is less than the threshold value. In other cases, having a number of saturated outputs exceeding a threshold value is an out of bounds condition; an index corresponding to the number of successive out of bounds conditions is determined; wherein the first dynamic scaling value is set to a first scalar value corresponding to the index; and the second dynamic scaling value is set to a

second scalar value corresponding to the index. In some such cases, the first scalar value and second scalar value are maintained in a lookup table, and the index is a pointer into the lookup table. In other cases, an index corresponding to the number of saturated outputs is determined; the first dynamic scaling value is set to a first scalar value corresponding to the index; and the second dynamic scaling value is set to a second scalar value corresponding to the index. In some such cases, the first scalar value and second scalar value are maintained in a lookup table, and the index is a pointer into the lookup table.

[0008] In other instances of the aforementioned embodiments, the dynamic scalar calculation circuit determines a number of violated checks represented in the decoded output. In some cases, the first dynamic scaling value is set to a first default value when the number of violated checks is less than a first threshold value; the first dynamic scaling value is set to a minimum value when the number of violated checks is greater than the first threshold value; the second dynamic scaling value is set to a second default value when the number of violated checks is less than a second threshold value, and the second dynamic scaling value is set to a maximum value when the number of violated checks is greater than the second threshold value. In other cases, having a number of violated checks less than a threshold value is an out of bounds condition; an index corresponding to the number of successive out of bounds conditions is determined; the first dynamic scaling value is set to a first scalar value corresponding to the index; and the second dynamic scaling value is set to a second scalar value corresponding to the index. In some such cases, the first scalar value and second scalar value are maintained in a lookup table, and the index is a pointer into the lookup table.

[0009] Other embodiments of the present invention provide methods for data processing. The methods include providing a decoder circuit that receives a first input and provides a decoded output, and a detector circuit that receives a second input and provides a detected output. The methods further include dynamically determining a first scaling value and a second scaling value based at least in part on the decoded output; multiplying the decoded output by the first scaling value to yield the second input; and multiplying the detected output by the second scaling value to yield a product output.

[0010] Yet other embodiments of the present invention provide disk drive systems. Such disk drive systems include a storage medium, and a read channel device receiving data derived from

the storage medium. The read channel device includes a decoder circuit providing a decoded output, and a dynamic scalar calculation circuit that determines a first dynamic scaling value and a second dynamic scaling value based at least in part on the decoded output. A first multiplier circuit multiplies the decoded output by the first dynamic scaling value and provides a first scaled output. A detector circuit receives the first scaled output and provides a detected output. A second multiplier circuit multiplies the detected output by the second dynamic scaling value and provides a second scaled output.

[0011] This summary provides only a general outline of some embodiments of the invention. Many other objects, features, advantages and other embodiments of the invention will become more fully apparent from the following detailed description, the appended claims and the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] A further understanding of the various embodiments of the present invention may be realized by reference to the figures which are described in remaining portions of the specification. In the figures, like reference numerals are used throughout several figures to refer to similar components. In some instances, a sub-label consisting of a lower case letter is associated with a reference numeral to denote one of multiple similar components. When reference is made to a reference numeral without specification to an existing sub-label, it is intended to refer to all such multiple similar components.

[0013] Fig. 1 depicts a prior art two stage data detection and decoding system;

[0014] Fig. 2 depicts a data processing circuit including dynamic scalars in accordance with one or more embodiments of the present invention;

[0015] Fig. 3a shows a queuing detection and decoding circuit including dynamic scalars in accordance with various embodiments of the present invention;

[0016] Fig. 3b shows another queuing detection and decoding circuit including dynamic scalars in accordance with various embodiments of the present invention;

[0017] Fig. 4 is a flow diagram showing a method in accordance with some embodiments of the present invention for data processing using dynamic scaling;

[0018] Figs. 5a-5d are four flow diagrams showing different approaches for dynamically calculating scaling factors in accordance with different embodiments of the present invention; and

[0019] Fig. 6 shows a storage system with dynamic scaling in accordance with various embodiments of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0020] The present inventions are related to systems and methods for detecting and/or decoding information, and more particularly to systems and methods for performing iterative data decoding and/or detection.

[0021] Various embodiments of the present invention provide compact hardware solutions for decoding suitable for read channel, wireless transmission and other applications. Such decoding/detection circuitry includes a soft decision decoder providing data to a data detector. The output from the soft decision decoder is scaled by a first scalar before being provided to the data detector, and the output of the data detector is scaled by a second scalar. The first scalar and the second scalar are dynamically calculated based upon the output of the soft decision decoder. As one advantage, such an approach may be used to reduce saturation, and thereby increasing the probability that the decoding/detection circuitry will converge on the desired result. Saturation is the number of soft decisions provided by a decoder circuit that are equal to or exceed a maximum.

[0022] Turning to Fig. 2, a data processing circuit 200 including dynamic scalars is depicted in accordance with one or more embodiments of the present invention. Data processing circuit 200 includes a series of data detector circuits and data decoder circuits that provide for multiple pass processing of a received data input. In particular, data input 201 is received by a data detector

circuit 210. Data detector circuit 210 may be any data detection circuit known in the art. For example, in one particular embodiment of the present invention, data detector circuit 210 is a maximum a posteriori (MAP) detector as are known in the art. Based upon the disclosure provided herein, one of ordinary skill in the art will recognize a variety of other types of detector circuits and/or algorithms that may be used in relation to different embodiments of the present invention. Data detector circuit 210 performs a data detection process on data input 205 and provides a detected output 213.

[0023] Detected output 213 is multiplied by a scaling factor 214 (α) using a multiplier circuit 212. A product output 211 from multiplier circuit 212 is provided to a decoder circuit 215. Decoder circuit 215 may be any circuit capable of applying a decoding algorithm to a received input. In some particular embodiments of the present invention, decoder circuit 215 is a low density parity check (LDPC) decoder circuit as are known in the art. Based upon the disclosure provided herein, one of ordinary skill in the art will recognize a variety of decoder circuits that may be used in relation to different embodiments of the present invention. Scaling factor 214 is a static value selected either at design time or programmable at the time that data processing circuit 200 is deployed. Scaling factor 214 is designed to scale detected output 214 to maintain the operation of decoder circuit 215 in an unsaturated region.

[0024] Decoder circuit 215 provides a decoded output 218. Decoded output 218 is multiplied by a dynamic scaling factor 264 (β_x) using a multiplier circuit 217. A product output 216 from multiplier circuit 217 is provided to a data detector circuit 220 that uses the original input 205 delayed through a delay circuit 250 to align it in time with product output 216. Data detector circuit 220 may be any data detection circuit known in the art. For example, in one particular embodiment of the present invention, data detector circuit 220 is a maximum a posteriori (MAP) detector as are known in the art. Based upon the disclosure provided herein, one of ordinary skill in the art will recognize a variety of other types of detector circuits and/or algorithms that may be used in relation to different embodiments of the present invention. Data detector circuit 220 performs a data detection process on product output 216 and provides a detected output 223.

[0025] Detected output 223 is multiplied by a scaling factor 262 (α_x) using a multiplier circuit 222. A product output 221 from multiplier circuit 222 is provided to a decoder circuit 225. A dynamic scalar computation circuit 260 calculates scaling factor 264 and scaling factor 262 using

decoded output 218. In particular, dynamic scalar computation circuit 260 receives an indication of the number of parity checks for the codeword currently processed by decoder circuit 215 that remain violated. This number is referred to herein as the violation count. Further, dynamic scalar computation circuit 260 counts the number of bit periods in decoded output that are saturated (i.e., have a value equal to the maximum achievable value). This value is referred to as the saturation count. Dynamic scalar computation circuit 260 compares the violation count with a threshold value 292 and compares the saturation count with another threshold value 294. In some embodiments, both threshold value 292 and threshold value 294 are programmable. The values of scaling factor 262 and scaling factor 264 are updated if either the violation count is less than threshold value 292 or if the saturation count is greater than threshold 294. The following pseudocode describes the update condition:

```
If (violation count < threshold 292 || saturation count > threshold 294) {
        Update Scaling Factors
}
Else {
        Maintain Scaling Factors
}
```

In an exemplary case, dynamic modification of scaling factor 262 and scaling factor 264 is only done whenever there is a significant number of saturated soft decisions.

[0026] Whenever an update is called for, updating scaling factor 262 and scaling factor 264 may be done in one a variety of ways depending upon the particular implementation. Initially, scaling factor 262 and scaling factor 264 are set equal to default values. The following pseudocode describes the initial condition:

Scaling Factor 264 = $\beta_{default}$; and
Scaling Factor 262 = $\alpha_{default}$.

The default values, $\beta_{default}$ and $\alpha_{default}$ may be hard coded at design time, or may be programmable allowing for update depending upon a particular deployment of data processing circuit 200. The default values, $\beta_{default}$ and $\alpha_{default}$ may be selected to provide good performance with only minimal saturation. In one embodiment, upon a decision to update the value of scaling factor 262 and scaling factor 264, scaling factor 262 and scaling factor 264 are modified to be minimum value scaling factors in accordance with the following pseudocode:

```
If (violation count < threshold 292 || saturation count > threshold 294) {
```

Scaling Factor 264 = $\beta_{minimum}$; and
Scaling Factor 262 = $\alpha_{minimum}$
}
Else {
Scaling Factor 264 = $\beta_{default}$; and
Scaling Factor 262 = $\alpha_{default}$
}

The values of $\beta_{minimum}$ and $\alpha_{minimum}$ are may be hard coded at design time, or may be programmable allowing for update depending upon a particular deployment of data processing circuit 200. The default values, $\beta_{minimum\ and}$ $\alpha_{minimum}$ may be selected to provide good performance but with values set to avoid some saturation allowed by the default scaling factors. Other examples of dynamic modification of scaling factor 262 and scaling factor 264 are discussed below in relation to dynamic scalar computation circuit 260.

[0027] Decoder circuit 225 may be any circuit capable of applying a decoding algorithm to a received input. In some particular embodiments of the present invention, decoder circuit 225 is a low density parity check (LDPC) decoder circuit as are known in the art. Based upon the disclosure provided herein, one of ordinary skill in the art will recognize a variety of decoder circuits that may be used in relation to different embodiments of the present invention. Decoder circuit 225 provides a decoded output 228. Decoded output 228 is multiplied by a dynamic scaling factor 269 (β_y) using a multiplier circuit 227. A product output 226 from multiplier circuit 227 is provided to a data detector circuit 230 that uses the original input 205 delayed through a delay circuit 252 to align it in time with product output 226. Data detector circuit 230 may be any data detection circuit known in the art. For example, in one particular embodiment of the present invention, data detector circuit 230 is a maximum a posteriori (MAP) detector as are known in the art. Based upon the disclosure provided herein, one of ordinary skill in the art will recognize a variety of other types of detector circuits and/or algorithms that may be used in relation to different embodiments of the present invention. Data detector circuit 230 performs a data detection process using product output 226 and provides a detected output 233. Detected output 233 is multiplied by a scaling factor 267 (α_y) using a multiplier circuit 232. A product output 231 from multiplier circuit 232 is provided to a decoder circuit 235. Decoder circuit 235 may be any circuit capable of applying a decoding algorithm to a received input. In some particular embodiments of the present invention, decoder circuit 235 is a low density parity check (LDPC) decoder circuit as are known in the art. Based upon the disclosure provided

herein, one of ordinary skill in the art will recognize a variety of decoder circuits that may be used in relation to different embodiments of the present invention. Product output 231 is decoded using LDPC decoder 235 to provide a final output 240.

[0028] A dynamic scalar computation circuit 270 calculates scaling factor 269 and scaling factor 267 using decoded output 228. Dynamic scalar computation circuit 270 operates similar to dynamic scalar computation circuit 260, except that it receives a feed forward signal 263 from dynamic scalar calculation circuit 260 allowing for various incremental operational modes. In particular, dynamic scalar calculation circuit 270 receives decoded output 228 and based thereon determines whether an update of scaling factor 267 and scaling factor 269 is to occur. Similar to dynamic scalar calculation circuit 260, a update is done if either the violation count (derived from decoded output 228) is less than threshold 292 or the saturation count (derived from decoded output 228) is greater than threshold 294. The following pseudocode describes the update condition:

```
If (violation count < threshold 292 || saturation count > threshold 294) {
        Update Scaling Factors
}
Else {
        Maintain Scaling Factors
}
```

In an exemplary case, dynamic modification of scaling factor 267 and scaling factor 269 is only done whenever there is a significant number of saturated soft decisions.

[0029] Where an update is to be done, it may be done using the approach described above in relation to dynamic scalar calculation circuit 260. Initially, scaling factor 267 and scaling factor 269 are set equal to default values. The following pseudocode describes the initial condition:

Scaling Factor 269 = $\beta_{default}$; and
Scaling Factor 267 = $\alpha_{default}$.

Again, the default values, $\beta_{default}$ and $\alpha_{default}$ may be hard coded at design time, or may be programmable allowing for update depending upon a particular deployment of data processing circuit 200. The default values, $\beta_{default\ and}$ $\alpha_{default}$ may be selected to provide good performance with only minimal saturation. In one embodiment, upon a decision to update the value of scaling

factor 267 and scaling factor 269, scaling factor 267 and scaling factor 269 are modified to be minimum value scaling factors in accordance with the following pseudocode:

```
If (violation count < threshold 292 || saturation count > threshold 294) {
        Scaling Factor 269 = β_maximum; and
        Scaling Factor 267 = α_minimum
}
Else {
        Scaling Factor 269 = β_default; and
        Scaling Factor 267 = α_default
}
```

Again, the values of $\beta_{maximum}$ and $\alpha_{minimum}$ are may be hard coded at design time, or may be programmable allowing for update depending upon a particular deployment of data processing circuit 200. The default values, $\beta_{maximum\ and}$ $\alpha_{minimum}$ may be selected to provide good performance but with values set to avoid some saturation allowed by the default scaling factors.

[0030] Other examples of dynamic modification of scaling factor 267 and scaling factor 269 involving feed forward signal 263 may also be used. Such approaches allow for modification of scaling factor 267 and scaling factor 269 based upon the results of dynamic scalar factor circuit 260. For example, the values of scaling factor 262 and scaling factor 264, and scaling factor 267 and scaling factor 269 are incrementally modified each successive time a need to update is indicated. Thus, for example, where an update is indicated, dynamic scalar computation circuit 260 selects a value for scaling factor 262 and scaling factor 264 based on an incremental value i=1, and a feed forward signal 263 is asserted to dynamic scalar computation circuit 270. Where an update condition remains based upon decoded output 228, the incremental value i=2 is selected for the next iteration of scaling factor 267 and scaling factor 269. In doing this, a lookup table may be used that includes successive values for scaling factor 262 and scaling factor 264, indexed by i; and scaling factor 267 and scaling factor 269, also indexed by i. The lookup table values may be defined at design time or may be programmable. The following pseudocode describes incremental modification of scaling factor 262 and scaling factor 264, and scaling factor 267 and scaling factor 269:

```
/*at a first instant in time where decoded output 218 is available for a given codeword*/
    If (violation count < threshold 292 || saturation count > threshold 294) {
            i=1;
            Scaling Factor 264 = β(i);
            Scaling Factor 262 = α(i); and
```

```
        Feed Forward Signal 263 = asserted
    }
    Else {
        i=0
        Scaling Factor 264 = β_default;
        Scaling Factor 262 = α_default; and
        Feed Forward Signal 263 = de-asserted
    }
/*at a second instant in time where decoded output 228 is available for the given codeword*/
    If (violation count < threshold 292 || saturation count > threshold 294) {
        If (Feed Forward Signal 263 == asserted){
            i=2;
            Scaling Factor 269 = β(i); and
            Scaling Factor 267 = α(i);
        }
        Else{
            i=1;
            Scaling Factor 269 = β(i); and
            Scaling Factor 267 = α(i);
        }
    Else {
        i=0
        Scaling Factor 269 = β_default; and
        Scaling Factor 267 = α_default
    }
```

In this case, $\beta(i+1)$ is greater than $\beta(i)$ and $\alpha(i+1)$ is less than $\alpha(i)$. In this way, the value of scaling factor 262 and scaling factor 264, and scaling factor 267 and scaling factor 269 can be modified incrementally based upon a prior result.

[0031] In yet another embodiment, the values of scaling factor 262 and scaling factor 264, and scaling factor 267 and scaling factor 269 are incrementally modified each successive time a need to update is indicated, with the magnitude of the increment being controlled not by a predefined increase in the incremental value (e.g., increasing I from one to two), but by an incremental increase corresponding to the value of the saturation count for the particular stage. To do this, lookup table may be used that includes successive values for scaling factor 262 and scaling factor 264 indexed by I; and scaling facto 267 and scaling factor 269 also indexed by i. The lookup table values may be defined at design time or may be programmable. The following pseudocode describes incremental modification of scaling factor 262 and scaling factor 264, and scaling factor 267 and scaling factor 269:

```
/*at a first instant in time where decoded output 218 is available for a given codeword*/
    If (violation count < threshold 292 || saturation count > threshold 294) {
            i = i+saturation count corresponding to decoded output 218;
            Scaling Factor 264 = β(i);
            Scaling Factor 262 = α(i); and
            Feed Forward Signal 263 = asserted
    }
    Else {
            i=0
            Scaling Factor 264 = β_default; and
            Scaling Factor 262 = α_default;
            Feed Forward Signal 263 = de-asserted
    }

 /*at a second instant in time where decoded output 228 is available for the given codeword*/
    If (violation count < threshold 292 || saturation count > threshold 294) {
            If (Feed Forward Signal 263 == asserted){
                    i= i+saturation count corresponding to decoded output 228;
                    Scaling Factor 269 = β(i); and
                    Scaling Factor 267 = α(i);
            }
            Else{
                    i= saturation count corresponding to decoded output 228;
                    Scaling Factor 269 = β(i); and
                    Scaling Factor 267 = α(i);
            }
    Else {
            i=0
            Scaling Factor 269 = β_default; and
            Scaling Factor 267 = α_default
    }
```

Again, $\beta(i+1)$ is greater than $\beta(i)$ and $\alpha(i+1)$ is less than $\alpha(i)$. In this way, the value of scaling factor 262 and scaling factor 264 can be modified incrementally based upon a prior result and based upon a magnitude of the detected saturation numbers.

[0032] Turning to Fig. 3a, a queuing detection and decoding circuit 300 including dynamic scalars is shown in accordance with various embodiments of the present invention. Queuing detection and decoding circuit 300 includes a data input 305 that is fed to a channel detector 309. In some embodiments, data input 305 may be derived from a storage medium. In particular cases, data input 305 is provided as groups of data or data sets that are sometimes referred to as codewords. In the case of a hard disk drive, the received data sets may be sectors of data from the storage medium of the hard disk drive. Based upon the disclosure provided herein, one of

ordinary skill in the art will recognize other sources for data input, and other data sets that may be processed in accordance with different embodiments of the present invention.

[0033] Channel detector 309 may be any type of channel detector known in the art including, but not limited to, a soft output Viterbi algorithm detector (SOVA) or a maximum a posteriori (MAP) detector. Based on the disclosure provided herein, one of ordinary skill in the art will recognize a variety of channel detectors that may be used in accordance with different embodiments of the present invention. In addition, data input 305 is provided to a memory buffer 313 that is designed to hold a number of data sets received from data input 305. The size of memory buffer 313 may be selected to provide sufficient buffering such that a data set provided via data input 305 remains available at least until a first iteration processing of that same data set is complete and the processed data is available in a queue buffer 349 as more fully described below. Memory buffer 313 provides the data sets to a channel detector 317. Similar to channel detector 309, channel detector 317 may be any type of channel detector known in the art including, but not limited to, a SOVA detector or a MAP detector. Again, based on the disclosure provided herein, one of ordinary skill in the art will recognize a variety of channel detectors that may be used in accordance with different embodiments of the present invention.

[0034] An output 381 of channel detector 309 is multiplied by a scaling factor 301 (α) using a multiplier circuit 391. A product output 395 from multiplier circuit 391 is provided to an interleaver circuit 394. An output 383 of channel detector 317 is multiplied by a dynamic scaling factor 303 (α_x) using a multiplier circuit 393. A product output 397 from multiplier circuit 393 is provided to another interleaver circuit 392. Interleaver circuit 394 interleaves the output of channel detector 309 using a ping pong buffer 396, and interleaver circuit 392 interleaves the output of channel detector 317 using a ping pong buffer 398. One of the buffers in ping pong buffer 396 holds the result of a prior interleaving process of the output from channel detector 309 and is unloaded to an LDPC decoder 337 via a multiplexer 321, while the other buffer of ping pong buffer 396 holds a data set from channel detector 309 that is currently being interleaved. Similarly, one of the buffers in ping pong buffer 398 holds the result of a prior interleaving process of the output from channel detector 317 and is unloaded to LDPC decoder 337 via a multiplexer 321, while the other buffer of ping pong buffer 398 holds a data set from channel detector 317 that is currently being interleaved. It should be noted that other

soft decision data decoders may be used in place of LDPC decoder 337 in different embodiments of the present invention.

[0035] LDPC decoder 337 is capable of decoding one or more data sets simultaneously. As an example, LDPC decoder 337 may be designed to decode an interleaved data set from ping pong buffer 396, to decode an interleaved data set from ping pong buffer 398, or to decode interleaved data sets from ping pong buffer 396 and ping pong buffer 398 simultaneously. The decoded data is either provided as a hard decision output 341 or to a de-interleaver circuit 345 that uses queue buffer 349 to de-interleave the decoded data and to store the de-interleaved data until channel detector 317 is available for further processing.

[0036] One of the buffers in queue buffer 349 holds the result of a prior de-interleaving process and is unloaded to channel detector 317, while another buffer of queue buffer 349 holds a decoded data set currently being de-interleaved, and one or more other buffers in queue buffer 349 maintain other non-converged data waiting for processing by channel detector 317. Non-converged data from queue buffer 349 is de-interleaved by de-interleaver 345 and passed to channel detector 317 that has access to the corresponding data set in memory buffer 313. In particular, the de-interleaved data from de-interleaver 345 is multiplied by a dynamic scaling factor 307 (β_x) using a multiplier circuit 327, and a product output of multiplier 329 is provided to channel detector 317. The data detection performed by channel detector 317 is similar to that performed by channel detector 309. Alternatively, where a data set converges in LDPC decoder 337, it is provide as hard decision output 341 to a de-interleaver circuit 357 that de-interleaves the received hard decision output 341 and stores the de-interleaved result in one of a number of memory buffers 361. Ultimately, de-interleaver circuit 357 provides the de-interleaved data stored in memory buffers 361 as an output 371.

[0037] Dynamic scaling factor 303 and dynamic scaling factor 307 are calculated for each codeword based upon a decoded output 385 corresponding to the respective codeword. Each buffer in queue buffer 349 includes an area 341 for storing decoded output 385 for a respective codeword, and another area 343 for storing dynamic scaling factor 303 and dynamic scaling factor 307 corresponding to the respective codeword. When the decoded output is pulled from a respective queue buffer for de-interleaving and processing by channel detector 317, the values

for scaling factor 303 and scaling factor 307 are pulled from area 341 of the corresponding queue buffer and used for multiplication by multiplier 391 and multiplier 393.

[0038] A dynamic scalar calculation circuit 399 calculates the value for scaling factor 303 and scaling factor 307 for each respective codeword based upon decoded output 385 corresponding to the respective codeword. The calculated scaling factors are stored to the buffer in queue buffer 349 that the corresponding to decoded output 385 from which the scaling factors were calculated.

[0039] Using decoded output 385, dynamic scalar calculation circuit 399 first determines whether the values for scaling factor 303 and scaling factor 307 are to be updated. In particular, dynamic scalar computation circuit 399 receives an indication of the number of parity checks for the data set currently processed by LDPC decoder circuit 337 that remain violated. This number is referred to herein as the violation count. Further, dynamic scalar computation circuit 399 counts the number of bit periods in decoded output 385 that are saturated (i.e., have a value equal to the maximum achievable value). This value is referred to as the saturation count. Dynamic scalar computation circuit 399 compares the violation count with a threshold value 323 and compares the saturation count with another threshold value 325. In some embodiments, both threshold value 323 and threshold value 325 are programmable. The values of scaling factor 303 and scaling factor 307 are updated if either the violation count is less than threshold value 323 or if the saturation count is greater than threshold 325. The following pseudocode describes the update condition:

```
If (violation count < threshold 323 || saturation count > threshold 325) {
        Update Scaling Factors
}
Else {
        Maintain Scaling Factors
}
```

In an exemplary case, dynamic modification of scaling factor 303 and scaling factor 307 is only done whenever there is a significant number of saturated soft decisions.

[0040] Whenever an update is called for, updating scaling factor 303 and scaling factor 307 may be done in one a variety of ways depending upon the particular implementation. Initially, scaling

factor 303 and scaling factor 307 are set equal to default values. The following pseudocode describes the initial condition:

Scaling Factor 307 = $\beta_{default}$; and
Scaling Factor 303 = $\alpha_{default}$.

The default values, $\beta_{default\ and}$ $\alpha_{default}$ may be hard coded at design time, or may be programmable allowing for update depending upon a particular deployment of queuing detection and decoding circuit 300. The default values, $\beta_{default\ and}$ $\alpha_{default}$ may be selected to provide good performance with only minimal saturation. In one embodiment, upon a decision to update the value of scaling factor 303 and scaling factor 307, scaling factor 303 and scaling factor 307 are modified to be minimum value scaling factors in accordance with the following pseudocode:

```
If (violation count < threshold 323 || saturation count > threshold 325) {
        Scaling Factor 307 = βmaximum; and
        Scaling Factor 303 = αminimum
}
Else {
        Scaling Factor 307 = βdefault; and
        Scaling Factor 303 = αdefault
}
```

The values of $\beta_{maximum}$ and $\alpha_{minimum}$ are may be hard coded at design time, or may be programmable allowing for update depending upon a particular deployment of data processing circuit 200. The default values, $\beta_{maximum}$ and $\alpha_{minimum}$ may be selected to provide good performance but with values set to avoid some saturation allowed by the default scaling factors.

[0041] In another embodiment, the values of scaling factor 303 and scaling factor 307 are incrementally modified each successive time a need to update is indicated. In doing this, a lookup table may be used that includes successive values for scaling factor 303 and scaling factor 307 indexed by i. The lookup table values may be defined at design time or may be programmable. The following pseudocode describes incremental modification of scaling factor 303 and scaling factor 307:

```
If (violation count < threshold 323 || saturation count > threshold 325) {
        Scaling Factor 307 = β(i+1);
        Scaling Factor 303 = α(i+1); and
        i++
}
Else {
        Scaling Factor 307 = βdefault; and
```

```
        Scaling Factor 303 = α_default
        i=0
}
```

In this case, $\beta(i+1)$ is greater than $\beta(i)$ and $\alpha(i+1)$ is less than $\alpha(i)$. In this way, the value of scaling factor 303 and scaling factor 307 can be modified incrementally until a working limit is achieved.

[0042] In yet another embodiment, the values of scaling factor 303 and scaling factor 307 are incrementally modified each successive time a need to update is indicated, with the magnitude of the increment being controlled not by a counter, but by the value of the saturation count. To do this, lookup table may be used that includes successive values for scaling factor 303 and scaling factor 307 indexed by i. The lookup table values may be defined at design time or may be programmable. The following pseudocode describes incremental modification of scaling factor 303 and scaling factor 307:

```
If (violation count < threshold 323 || saturation count > threshold 325) {
        Scaling Factor 307 = β(i+saturation count);
        Scaling Factor 303 = α(i+saturation count); and
        i = i+saturation count
}
Else {
        Scaling Factor 307 = β_default; and
        Scaling Factor 303 = α_default
        i=0
}
```

Again, $\beta(i+1)$ is greater than $\beta(i)$ and $\alpha(i+1)$ is less than $\alpha(i)$. In this way, the value of scaling factor 303 and scaling factor 307 an be modified incrementally with an increment corresponding to the number of saturated outputs being reported by decoded output 385.

[0043] Queuing detection/decoding circuit 300 allows for performance of a variable number of detection and decoding iterations depending upon the introduced data. Further, in some cases, considerable power savings may be achieved through use of queuing detection/decoding circuit 300. Yet further, in some cases, a faster LDPC decoder may be implemented allowing for an increased throughput where substantial first iteration data convergence exists as multiple iterations are not necessarily required. Yet further, by allowing results of LDPC decoder 337 to be reported out of order, upstream processing does not have to wait for the completion of downstream processing. Re-ordering of the out of order results may be done by queuing

detection/decoding circuit 300 or by a downstream recipient of output 371. In addition, scaling factor 303 and scaling factor 307 can be dynamically updated to enhance the performance of the decoding and detection processes being performed and thereby provide an ability for early convergence. Based upon the disclosure provided herein, one of ordinary skill in the art will recognize a variety of other advantages that may be achieved through implementation of one or more embodiments of the present invention.

[0044] Turning to Fig. 3b, another queuing detection and decoding circuit 801 including dynamic scalars is shown in accordance with other embodiments of the present invention. Queuing detection and decoding circuit 801 is substantially the same as queuing detection and decoding circuit 300. The difference is that each buffer in queue buffer 349 includes area 341 for storing decoded output 385 for a respective codeword, and another area 843 for storing a violation count from decoded output 385. This violation count may be used when the particular queue buffer 349 is selected for additional processing. The respective violation count 849 from area 843 is provided to a dynamic scalar calculation circuit 899. Dynamic scalar calculation circuit 899 receives the output of de-interleaver 345 in comparison with queuing detection and decoding circuit 300 where dynamic scalar calculation circuit 399 that receives the output prior to de-interleaver 345. Prior to applying scaling factor 303 and scaling factor 307 to the data from the selected buffer in queue buffer 349, dynamic scalar calculation circuit 899 calculates the value of scaling factor 303 and scaling factor 307. In some cases, only a violation count is used to determined whether to dynamically modify scaling factor 303 and scaling factor 307. In such a case, the following pseudocode describes the update condition:

```
If (violation count < threshold 323) {
        Update Scaling Factors
}
Else {
        Maintain Scaling Factors
}
```

Updating the scaling factors may be done using any of the approaches discussed herein.

[0045] Turning to Fig. 4, a flow diagram 400 shows a method in accordance with some embodiments of the present invention for data processing using dynamic scaling factors. Following flow diagram 400, a data input is received (block 520). This data input may be, but is not limited to, a series of data bits received from a magnetic recording medium or a series of bits

received from a transmission channel. Based on the disclosure provided herein, one of ordinary skill in the art will recognize a variety of sources and formats for the received data input. A sample of the received data is stored in a buffer and retained for later processing (block 525). Data detection processes are performed on the received data (block 555). The resulting detected data is multiplied by a scaling factor (α)(block 557), and the product of the multiplication is interleaved (block 560), and the interleaved data is decoded (block 565). It is then determined whether the decoding process converged (block 545), and whether there is sufficient buffering available to reprocess the data (block 550).

[0046] Where either the decoding process converged (block 545) or there is insufficient buffering available (block 550), the decoded data is de-interleaved (block 570) and stored in a buffer (block 575). The buffer includes various results that may have become available out of order, and as such the various results are reordered in the buffer to represent the order in which the corresponding data input was originally received (block 580). It is then determined if a complete time set is available in the buffer (block 585). A complete time set includes every result corresponding to received inputs over a given period of time. Thus, for example, where the first result is delayed while two later results are reported, the complete time set exists for the three results once the first result is finally available in the buffer. It should be noted that in some embodiments of the present invention that the results are reported out of order to a recipient. In such cases, there is no need to reorder results or to determine whether complete time sets are available. Where a complete time set is available (block 585) or where the results are to be reported as they are received without regard to order, the result(s) are output to a recipient (block 590).

[0047] Alternatively, where the decoding process failed to converge (block 545) and there is sufficient buffering available (block 550), the process of detection and decoding is repeated. In particular, scaling factors (β_x and α_x) are calculated based upon the decoded data (block 502). The scaling factors may be calculated using one of the approaches described below in relation to Figs. 5a-5c. The decoded data is also de-interleaved (block 505) and the resulting de-interleaved data is stored to a buffer along with the corresponding scaling factors that were recently calculated (block 510). The data is accessed from the buffer and multiplied by the scaling factor (β_x) (block 512). The de-interleaved data multiplied by the scaling factor is aligned with the

corresponding sample of the data input (block 515) once the data detector is available. The de-interleaved data and the corresponding sample data input is provided to the data detector where a subsequent data detection is performed (block 530) on the originally stored sample of data input (block 525) using the soft input developed in the earlier processing of the same data input (blocks 555, 560, 565, 545, 550, 505, 510, 515). The result of the data detection process is interleaved (block 535) and the interleaved data is decoded (block 540). At this point, it is determined whether the data detection and decoding process failed to converge (block 545) and is to be repeated, or whether the result converged (block 545) and is to be reported.

[0048] Turning to Figs. 5a-5c, three different approaches for dynamically calculating scaling factors are shown in accordance with different embodiments of the present invention. Following flow diagram 700 of Fig. 5a, each time a decode process is completed, it is determined whether the violation count is less than a first threshold (i.e., Threshold A) (block 701) or whether the saturation count is greater than a second threshold (i.e., Threshold B) (block 703). Where either are true (blocks 701, 703), the scaling factors are updated (block 707). In particular, the scaling factors are updated in accordance with the following equations:

$$\beta \text{ Scaling Factor} = \beta_{maximum}; \text{ and}$$
$$\alpha \text{ Scaling Factor} = \alpha_{minimum}.$$

Otherwise, where an update is not called for (blocks 701, 703), scaling factors are set to default levels (block 705) in accordance with the following equations:

$$\beta \text{ Scaling Factor} = \beta_{default}; \text{ and}$$
$$\alpha \text{ Scaling Factor} = \alpha_{default}.$$

[0049] Following flow diagram 710 of Fig. 5b, each time a decode process is completed, it is determined whether the violation count is less than a first threshold (i.e., Threshold A) (block 711) or whether the saturation count is greater than a second threshold (i.e., Threshold B) (block 713). Where either are true (blocks 711, 713), an index (i) used to access scaling data from a lookup table is incremented (block 717). The scaling factors are then pulled from the lookup table using the index (block 729). In particular, the scaling factors are updated in accordance with the following equations:

$$\beta \text{ Scaling Factor} = \beta(i); \text{ and}$$
$$\alpha \text{ Scaling Factor} = \alpha(i).$$

$\beta(i+1)$ is greater than $\beta(i)$ and $\alpha(i+1)$ is less than $\alpha(i)$. Otherwise, where an update is not called for (blocks 711, 713), the scaling factors are set to default levels (block 715) in accordance with the following equations:

$$\beta \text{ Scaling Factor} = \beta_{default}; \text{ and}$$
$$\alpha \text{ Scaling Factor} = \alpha_{default}.$$

[0050] Following flow diagram 720 of Fig. 5c, each time a decode process is completed, it is determined whether the violation count is less than a first threshold (i.e., Threshold A) (block 721) or whether the saturation count is greater than a second threshold (i.e., Threshold B) (block 723). Where either are true (blocks 721, 723), an index corresponding to the saturation count is calculated (block 727). In some cases, the index is the saturation count multiplied by a scalar value and raised to the next whole number. The scaling factors are then pulled from the lookup table using the index (block 729). In particular, the scaling factors are updated in accordance with the following equations:

$$\beta \text{ Scaling Factor} = \beta(\text{index}); \text{ and}$$
$$\alpha \text{ Scaling Factor} = \alpha(\text{index}).$$

$\beta(i+1)$ is greater than $\beta(i)$ and $\alpha(i+1)$ is less than $\alpha(i)$. Otherwise, where an update is not called for (blocks 721, 723), the scaling factors are set to default levels (block 725) in accordance with the following equations:

$$\beta \text{ Scaling Factor} = \beta_{default}; \text{ and}$$
$$\alpha \text{ Scaling Factor} = \alpha_{default}.$$

[0051] Following flow diagram 820 of Fig. 5d, each time a decode process is completed, it is determined whether the violation count is less than a threshold (block 821). Where this is true (blocks 821), scaling factors (β_x, α_x) are calculated (block 829). Otherwise, where an update is not called for (block 821), the scaling factors are set to default levels (block 825) in accordance with the following equations:

$$\beta \text{ Scaling Factor} = \beta_{default}; \text{ and}$$
$$\alpha \text{ Scaling Factor} = \alpha_{default}.$$

[0052] Turning to Fig. 6, a storage system 600 including read channel 610 with hard decision assisted decoding in accordance with various embodiments of the present invention. Storage system 600 may be, for example, a hard disk drive. Read channel 610 may include, but is not

limited to, a data processing codec similar to those described above in relation to Fig. 2 or Fig. 3 that include dynamic scalar calculation. In some cases, the data processing codec may operate similar to that described in relation to Fig. 4.

[0053] Storage system 600 also includes a preamplifier 670, an interface controller 620, a hard disk controller 666, a motor controller 668, a spindle motor 672, a disk platter 678, and a read/write head assembly 676. Interface controller 620 controls addressing and timing of data to/from disk platter 678. The data on disk platter 678 consists of groups of magnetic signals that may be detected by read/write head assembly 676 when the assembly is properly positioned over disk platter 678. In one embodiment, disk platter 678 includes magnetic signals recorded in accordance with a perpendicular recording scheme. For example, the magnetic signals may be recorded as either longitudinal or perpendicular recorded signals.

[0054] In a typical read operation, read/write head assembly 676 is accurately positioned by motor controller 668 over a desired data track on disk platter 678. The appropriate data track is defined by an address received via interface controller 620. Motor controller 668 both positions read/write head assembly 676 in relation to disk platter 678 and drives spindle motor 672 by moving read/write head assembly to the proper data track on disk platter 678 under the direction of hard disk controller 666. Spindle motor 672 spins disk platter 678 at a determined spin rate (RPMs). Once read/write head assembly 678 is positioned adjacent the proper data track, magnetic signals representing data on disk platter 678 are sensed by read/write head assembly 676 as disk platter 678 is rotated by spindle motor 672. The sensed magnetic signals are provided as a continuous, minute analog signal representative of the magnetic data on disk platter 678. This minute analog signal is transferred from read/write head assembly 676 to read channel 610 via preamplifier 670. Preamplifier 670 is operable to amplify the minute analog signals accessed from disk platter 678. In turn, read channel module 610 decodes and digitizes the received analog signal to recreate the information originally written to disk platter 678. The decoding process includes hard decision assist which, in some cases, may limit the amount of processing required for convergence and in other cases may allow convergence that otherwise would not be possible. The read data is provided as read data 603. A write operation is substantially the opposite of the preceding read operation with write data 601 being provided to read channel module 610. This data is then encoded and written to disk platter 678.

[0055] In conclusion, the invention provides novel systems, devices, methods and arrangements for performing data decoding and/or detection. While detailed descriptions of one or more embodiments of the invention have been given above, various alternatives, modifications, and equivalents will be apparent to those skilled in the art without varying from the spirit of the invention. For example, one or more embodiments of the present invention may be applied to various data storage systems and digital communication systems, such as, for example, tape recording systems, optical disk drives, wireless systems, and digital subscribe line systems. Therefore, the above description should not be taken as limiting the scope of the invention, which is defined by the appended claims.

WHAT IS CLAIMED IS:

1. A data processing system, the system comprising:

a decoder circuit, wherein the decoder circuit provides a decoded output;

a dynamic scalar calculation circuit, wherein the dynamic scalar calculation circuit determines a first dynamic scaling value and a second dynamic scaling value based at least in part on the decoded output;

a first multiplier circuit, wherein the first multiplier circuit multiplies the decoded output by the first dynamic scaling value and provides a first scaled output;

a detector circuit, wherein the detector circuit receives the first scaled output and provides a detected output; and

a second multiplier circuit, wherein the second multiplier circuit multiplies the detected output by the second dynamic scaling value and provides a second scaled output.

2. The data processing system of claim 1, wherein the dynamic scalar calculation circuit determines a number of saturated outputs in the decoded output.

3. The data processing system of claim 2, wherein the first dynamic scaling value is set to a first default value when the number of saturated outputs exceeds a threshold value; wherein the first dynamic scaling value is set to a minimum value when the number of saturated outputs is less than the threshold value; wherein the second dynamic scaling value is set to a second default value when the number of saturated outputs exceeds the threshold value, and wherein the second dynamic scaling value is set to a maximum value when the number of saturated outputs is less than the threshold value.

4. The data processing system of claim 2, wherein having a number of saturated outputs exceeding a threshold value is an out of bounds condition; wherein an index corresponding to the number of successive out of bounds conditions is determined; wherein the first dynamic scaling value is set to a first scalar value corresponding to the index; and wherein the second dynamic scaling value is set to a second scalar value corresponding to the index.

5. The data processing system of claim 4, wherein the first scalar value and second scalar value are maintained in a lookup table, and wherein the index is a pointer into the lookup table.

6. The data processing system of claim 2, wherein an index corresponding to the number of saturated outputs is determined; wherein the first dynamic scaling value is set to a first scalar value corresponding to the index; and wherein the second dynamic scaling value is set to a second scalar value corresponding to the index.

7. The data processing system of claim 6, wherein the first scalar value and second scalar value are maintained in a lookup table, and wherein the index is a pointer into the lookup table.

8. The data processing system of claim 1, wherein the dynamic scalar calculation circuit determines a number of violated checks represented in the decoded output.

9. The data processing system of claim 8, wherein the first dynamic scaling value is set to a first default value when the number of violated checks is less than a first threshold value; wherein the first dynamic scaling value is set to a minimum value when the number of violated checks is greater than the first threshold value; wherein the second dynamic scaling value is set to a second default value when the number of violated checks is less than a second threshold value, and wherein the second dynamic scaling value is set to a maximum value when the number of violated checks is greater than the second threshold value.

10. The data processing system of claim 9, wherein having a number of violated checks less than a threshold value is an out of bounds condition; wherein an index corresponding to the number of successive out of bounds conditions is determined; wherein the first dynamic scaling value is set to a first scalar value corresponding to the index; and wherein the second dynamic scaling value is set to a second scalar value corresponding to the index.

11. The data processing system of claim 10, wherein the first scalar value and second scalar value are maintained in a lookup table, and wherein the index is a pointer into the lookup table.

12. A method for data processing, the method comprising:

providing a decoder circuit, wherein the decoder circuit receives a first input and provides a decoded output;

providing a detector circuit, wherein the detector circuit receives a second input and provides a detected output;

dynamically determining a first scaling value and a second scaling value based at least in part on the decoded output;

multiplying the decoded output by the first scaling value to yield the second input; and

multiplying the detected output by the second scaling value to yield a product output.

13. The method of claim 12, wherein dynamically determining the first scaling value and the second scaling value includes determining a number of saturated outputs in the decoded output.

14. The method of claim 13, wherein the first dynamic scaling value is set to a first default value when the number of saturated outputs exceeds a threshold value; wherein the first dynamic scaling value is set to a minimum value when the number of saturated outputs is less than the threshold value; wherein the second dynamic scaling value is set to a second default value when the number of saturated outputs exceeds the threshold value, and wherein the second dynamic scaling value is set to a maximum value when the number of saturated outputs is less than the threshold value.

15. The method of claim 13, wherein having a number of saturated outputs exceeding a threshold value is an out of bounds condition; wherein an index corresponding to the number of successive out of bounds conditions is determined; wherein the first dynamic scaling value is set to a first scalar value corresponding to the index; and wherein the second dynamic scaling value is set to a second scalar value corresponding to the index.

16. The method of claim 12, wherein dynamically determining the first scaling value and the second scaling value includes determining a number of violated checks represented in the decoded output.

17. The method of claim 16, wherein the first dynamic scaling value is set to a first default value when the number of violated checks is less than a first threshold value; wherein the first dynamic scaling value is set to a minimum value when the number of violated checks is greater than the first threshold value; wherein the second dynamic scaling value is set to a second default value when the number of violated checks is less than a second threshold value, and wherein the second dynamic scaling value is set to a maximum value when the number of violated checks is greater than the second threshold value.

18. The method of claim 16, wherein having a number of violated checks less than a threshold value is an out of bounds condition; wherein an index corresponding to the number of successive out of bounds conditions is determined; wherein the first dynamic scaling value is set to a first scalar value corresponding to the index; and wherein the second dynamic scaling value is set to a second scalar value corresponding to the index.

19. A disk drive system, the disk drive system comprising:

a storage medium;

a read channel device receiving data derived from the storage medium, wherein the read channel device includes:

a decoder circuit, wherein the decoder circuit provides a decoded output;

a dynamic scalar calculation circuit, wherein the dynamic scalar calculation circuit determines a first dynamic scaling value and a second dynamic scaling value based at least in part on the decoded output;

a first multiplier circuit, wherein the first multiplier circuit multiplies the decoded output by the first dynamic scaling value and provides a first scaled output;

a detector circuit, wherein the detector circuit receives the first scaled output and provides a detected output; and

a second multiplier circuit, wherein the second multiplier circuit multiplies the detected output by the second dynamic scaling value and provides a second scaled output.

20. The data processing system of claim 1, wherein the dynamic scalar calculation circuit determines a number of saturated outputs in the decoded output; and wherein the dynamic scalar calculation circuit determines a number of violated checks represented in the decoded output.

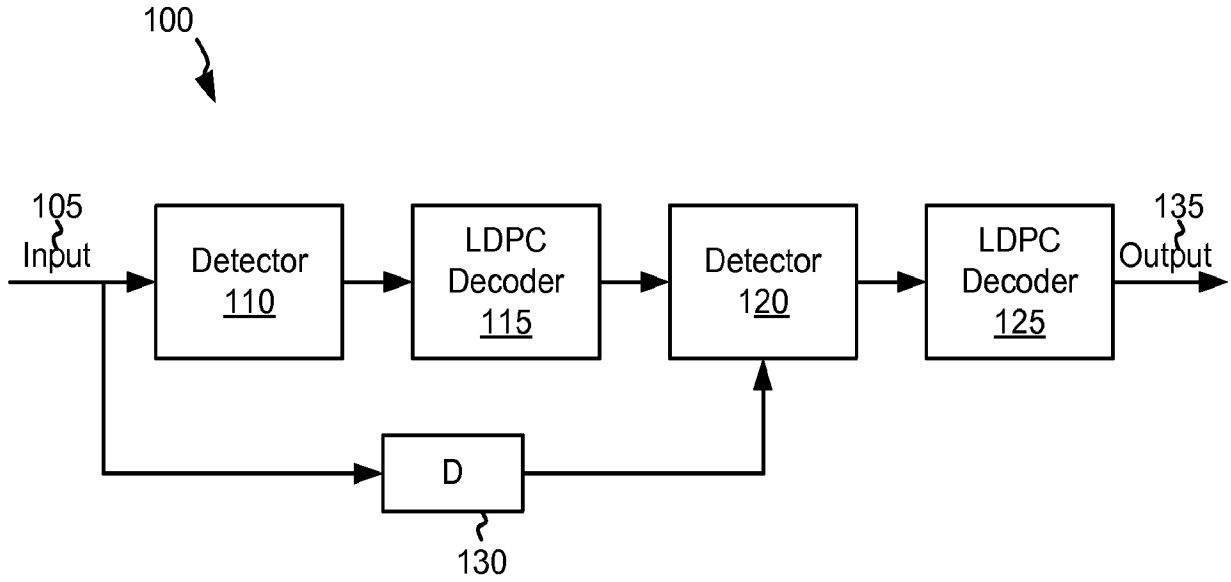


Fig. 1(Prior Art)

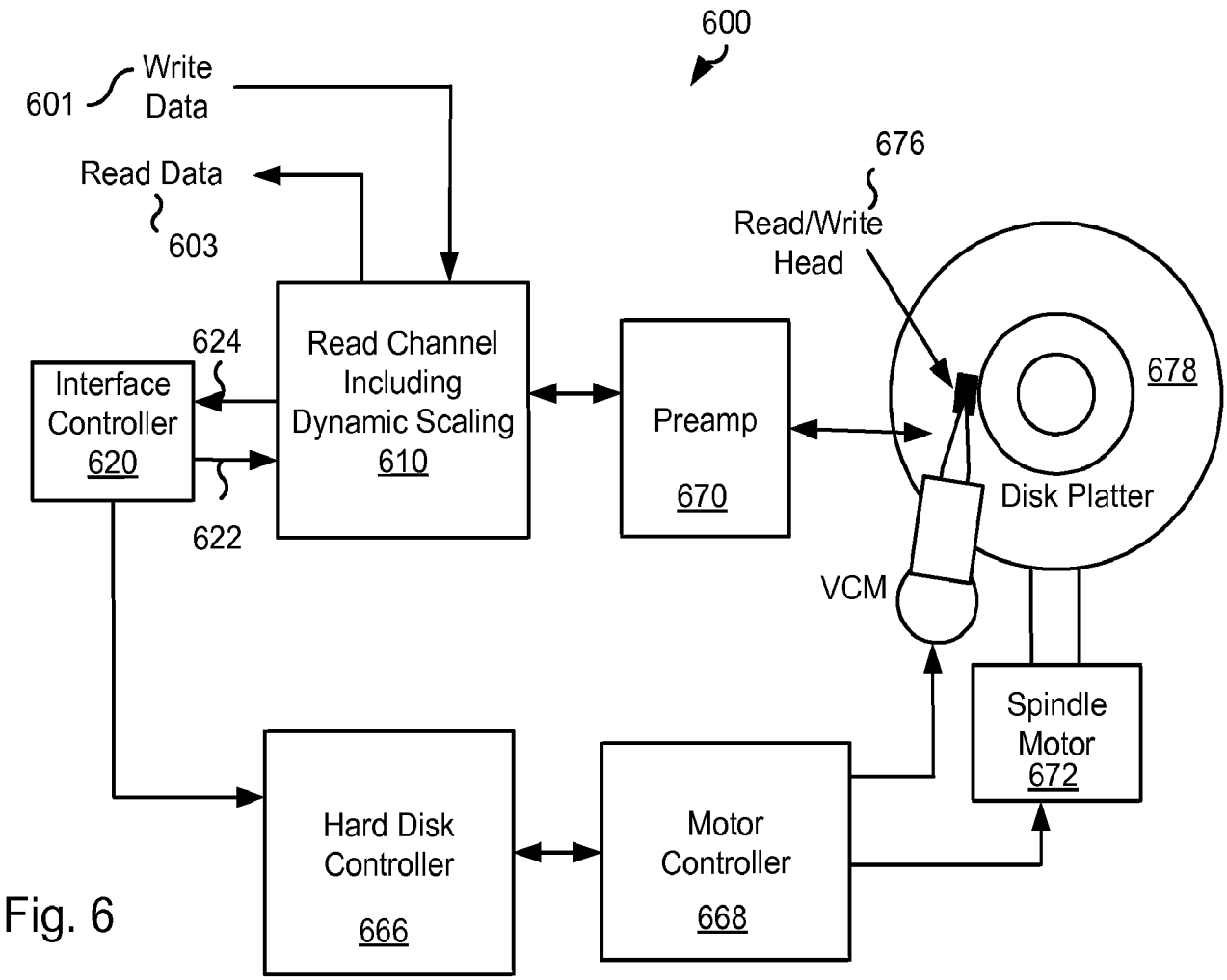


Fig. 6

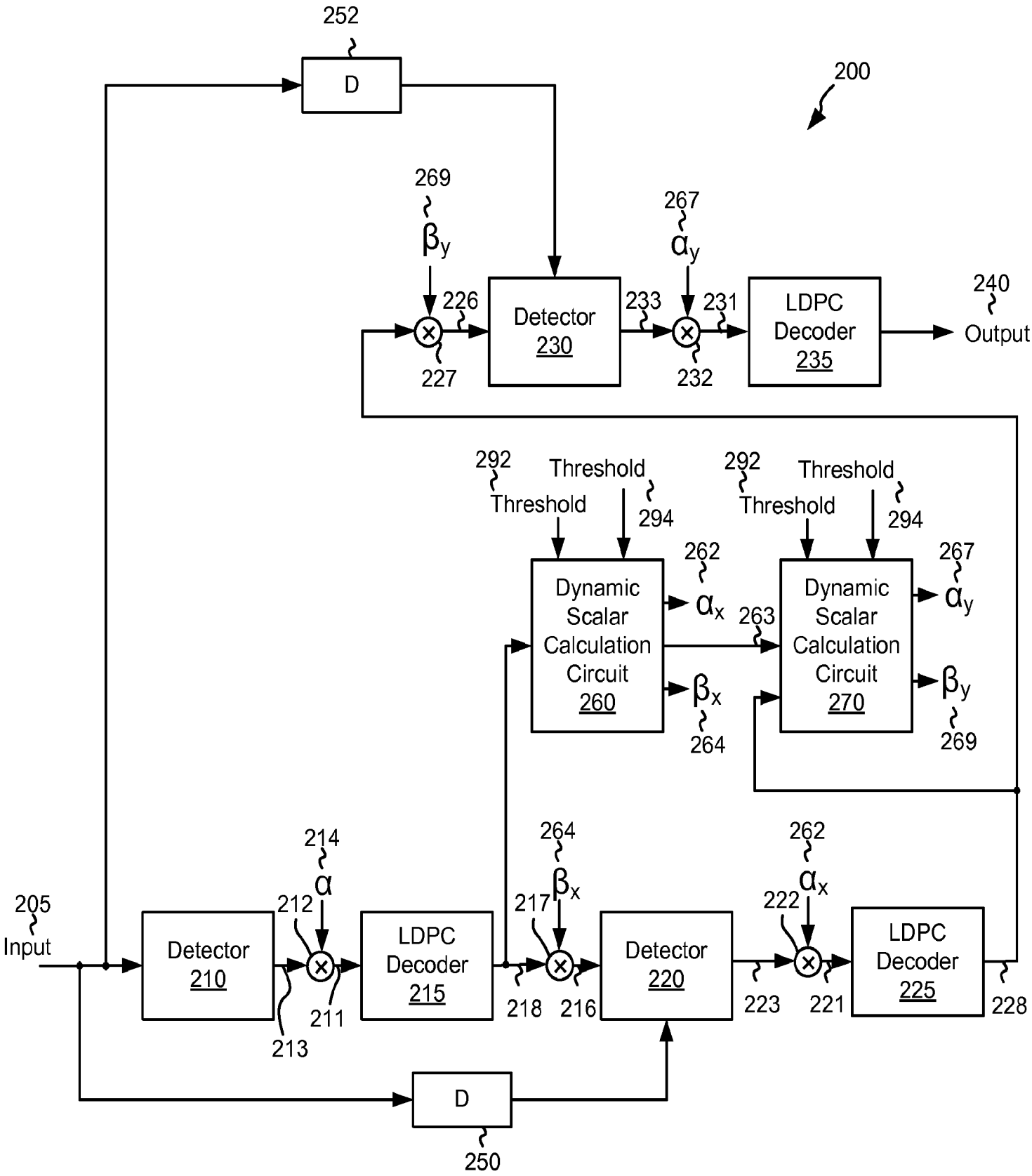

252
D
200
269
β_y
267
α_y
226
Detector
230
233
231
LDPC
Decoder
235
240
Output
227
232
292
Threshold
294
292
Threshold
294
262
Dynamic
Scalar
Calculation
Circuit
260
α_x
263
Dynamic
Scalar
Calculation
Circuit
270
267
α_y
β_x
264
β_y
269
214
α
264
β_x
262
α_x
205
Input
212
217
222
Detector
210
LDPC
Decoder
215
Detector
220
LDPC
Decoder
225
211
213
218
216
223
221
228
D
250

Fig. 2

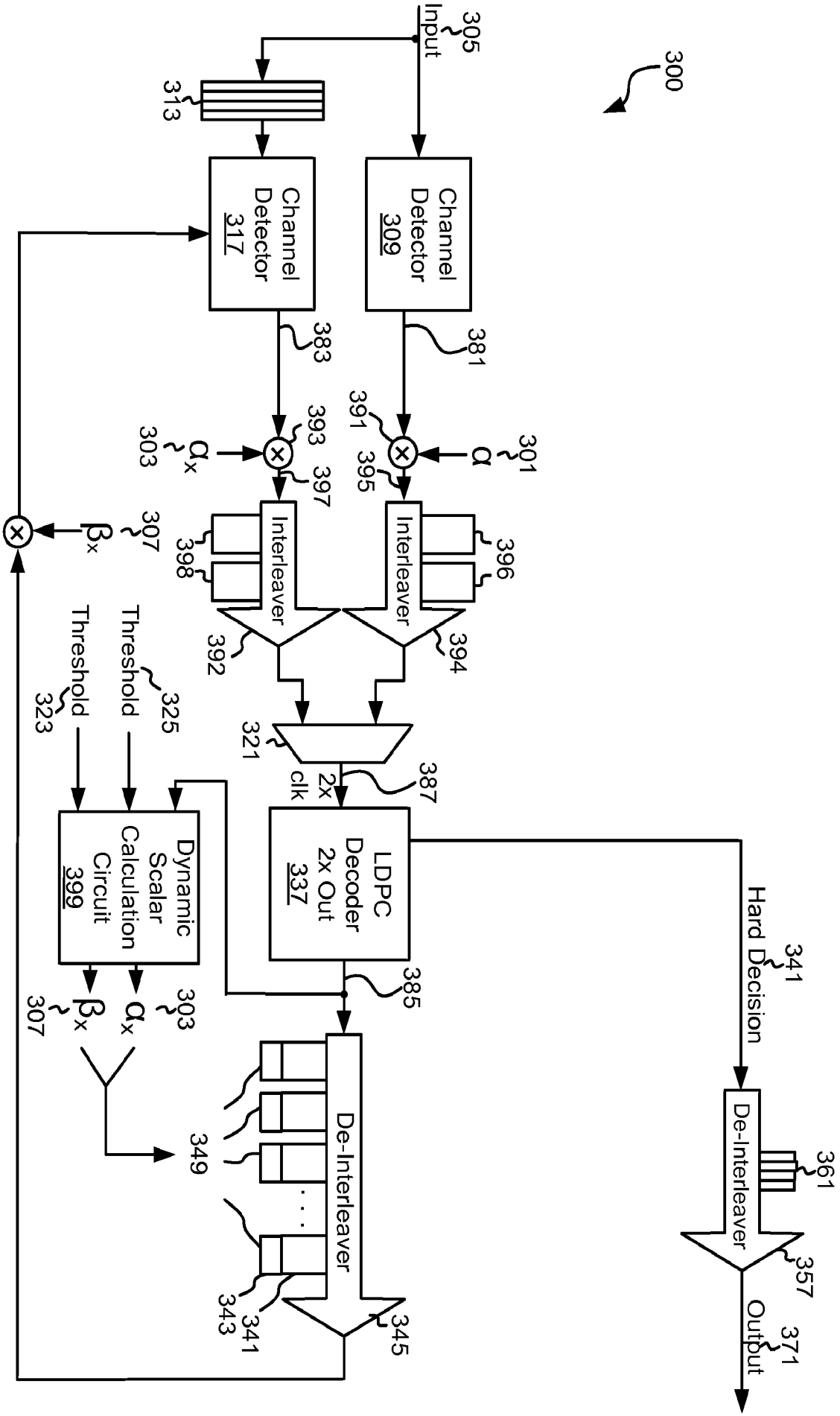

300
305
Input
313
Channel Detector 317
Channel Detector 309
383
381
391
393
395
397
303
α_x
301
α
307
β_x
Interleaver
398
396
392
394
Threshold 323
Threshold 325
321
clk
2x
387
Dynamic Scalar Calculation Circuit 399
LDPC Decoder 2x Out 337
385
Hard Decision
341
349
De-Interleaver
343
345
361
357
371
Output

Fig. 3a

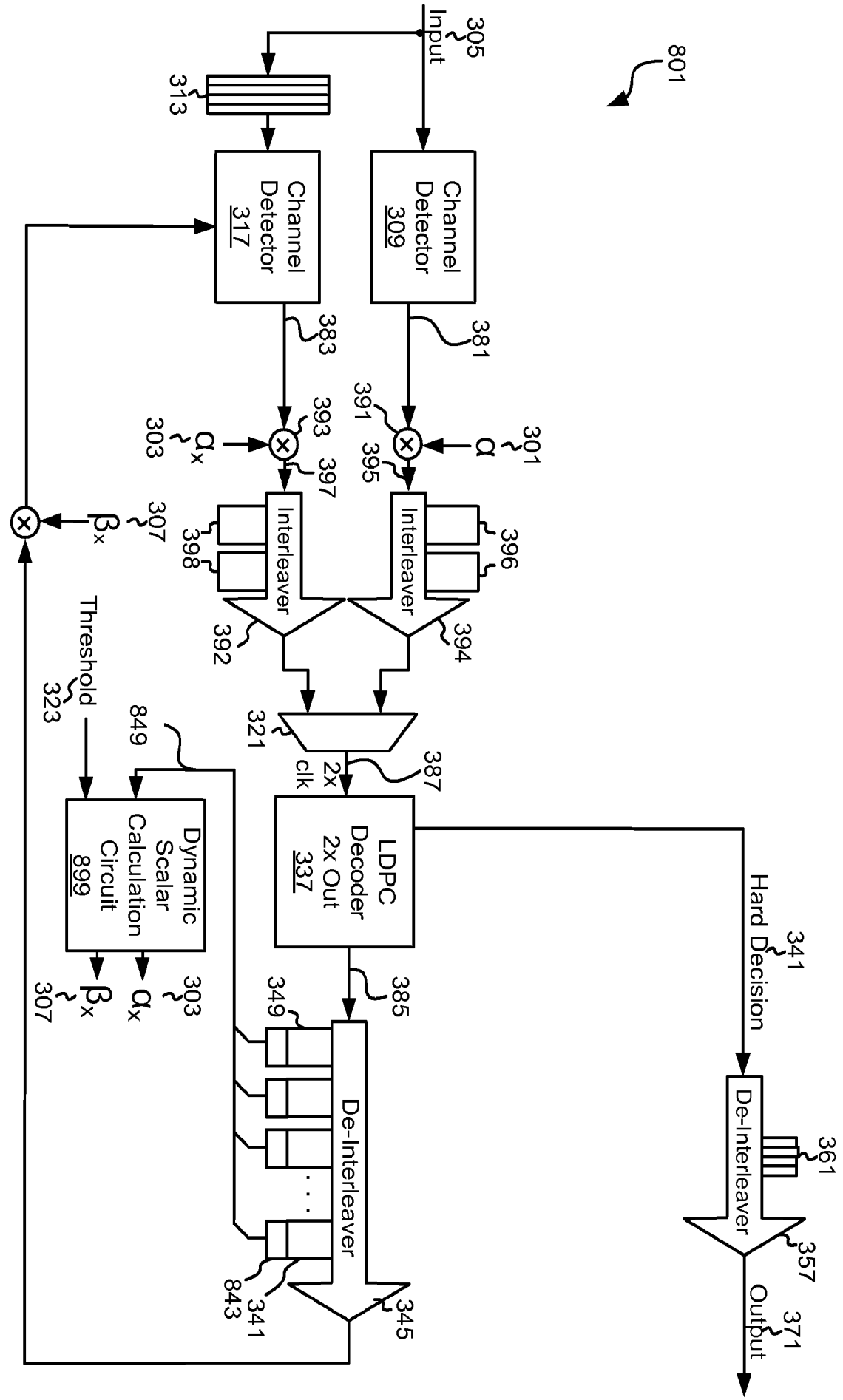

801
305
Input
313
Channel Detector 317
Channel Detector 309
383
381
391
393
303
α_x
395
α
301
397
307
β_x
398
396
Interleaver
Interleaver
392
394
Threshold
323
849
321
2x clk
387
Dynamic Scalar Calculation Circuit 899
LDPC Decoder 2x Out 337
Hard Decision
341
349
385
De-Interleaver
361
357
371
Output
843
345

Fig. 3b

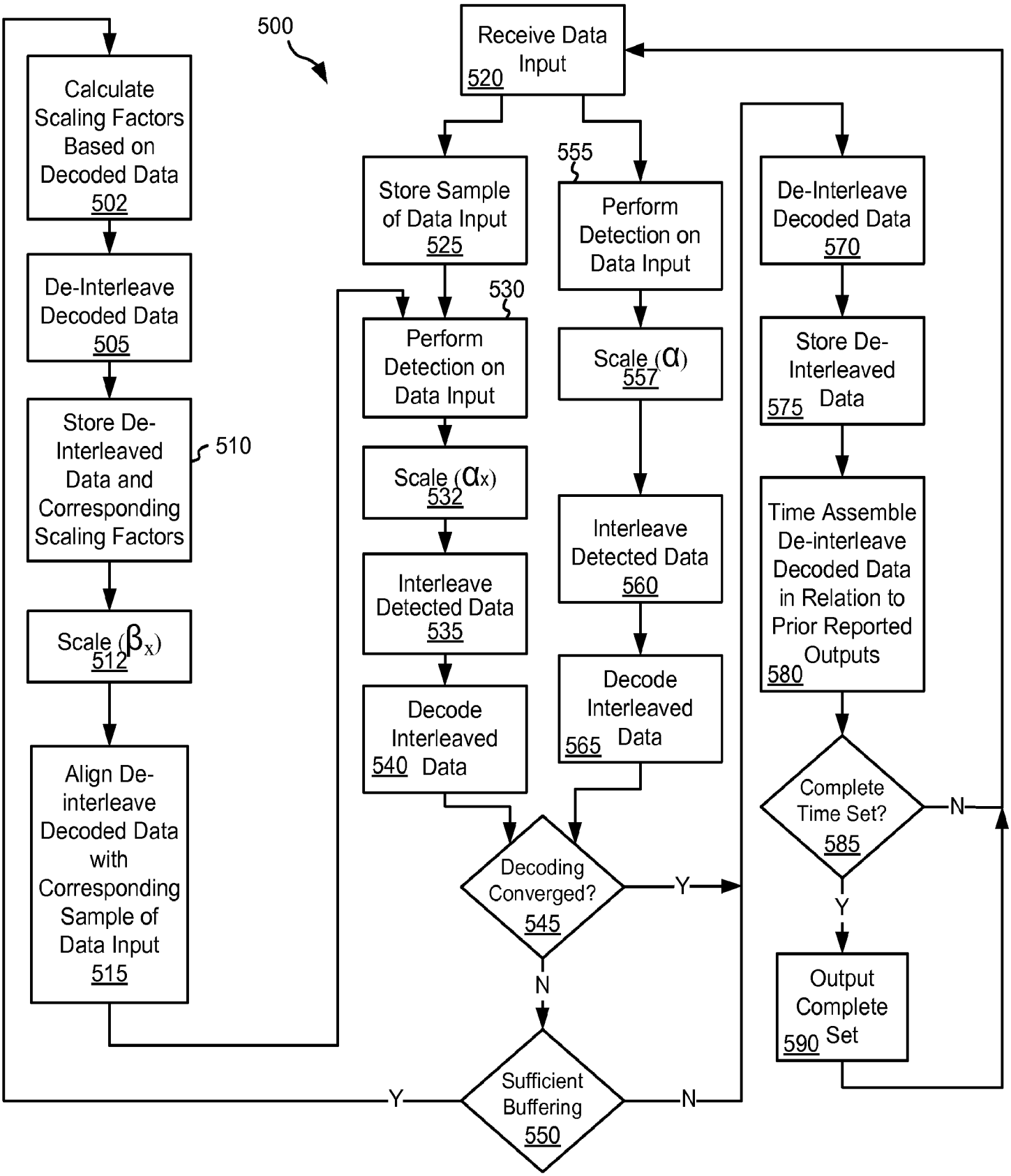

500
Receive Data Input 520
Calculate Scaling Factors Based on Decoded Data 502
De-Interleave Decoded Data 505
Store De-Interleaved Data and Corresponding Scaling Factors
510
Scale (β_x) 512
Align De-interleave Decoded Data with Corresponding Sample of Data Input 515
Store Sample of Data Input 525
530
Perform Detection on Data Input
Scale (α_x) 532
Interleave Detected Data 535
Decode Interleaved 540 Data
555
Perform Detection on Data Input
Scale (α) 557
Interleave Detected Data 560
Decode Interleaved 565 Data
Decoding Converged? 545
Y
N
Sufficient Buffering 550
Y
N
De-Interleave Decoded Data 570
Store De-Interleaved 575 Data
Time Assemble De-interleave Decoded Data in Relation to Prior Reported Outputs 580
Complete Time Set? 585
N
Y
Output Complete 590 Set

Fig. 4

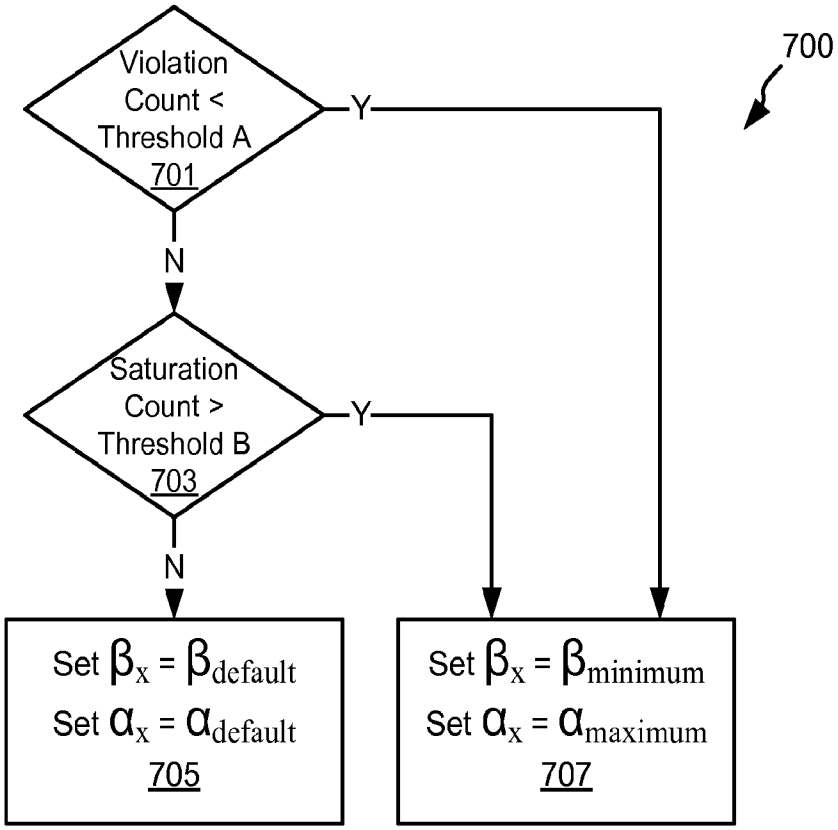

700
Violation Count < Threshold A 701
Y
N
Saturation Count > Threshold B 703
Y
N
Set $\beta_x = \beta_{default}$
Set $\alpha_x = \alpha_{default}$
705
Set $\beta_x = \beta_{minimum}$
Set $\alpha_x = \alpha_{maximum}$
707

Fig. 5a

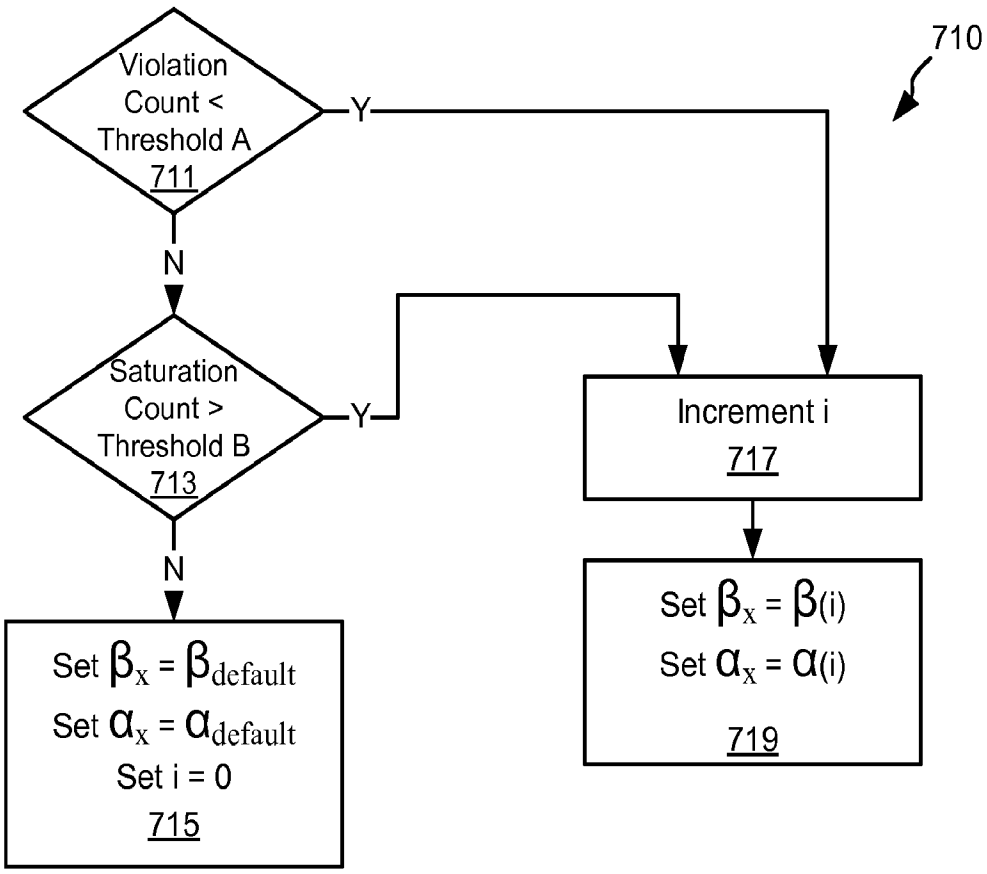

710
Violation Count < Threshold A 711
Y
N
Saturation Count > Threshold B 713
Y
N
Increment i
717
Set $\beta_x = \beta_{(i)}$
Set $\alpha_x = \alpha_{(i)}$
719
Set $\beta_x = \beta_{default}$
Set $\alpha_x = \alpha_{default}$
Set i = 0
715

Fig. 5b

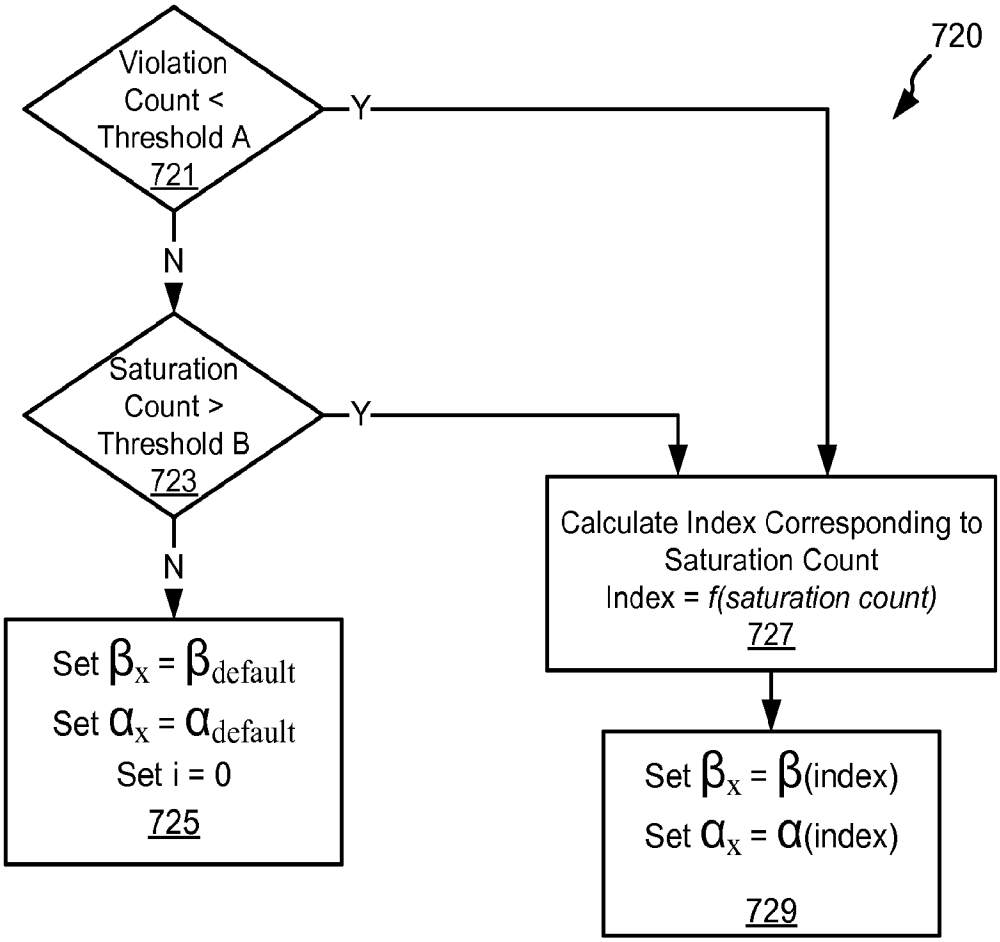

720
Violation Count < Threshold A 721
Y
N
Saturation Count > Threshold B 723
Y
N
Calculate Index Corresponding to Saturation Count
Index = f(saturation count)
727
Set $\beta_x = \beta_{default}$
Set $\alpha_x = \alpha_{default}$
Set i = 0
725
Set $\beta_x = \beta$(index)
Set $\alpha_x = \alpha$(index)
729

Fig. 5c

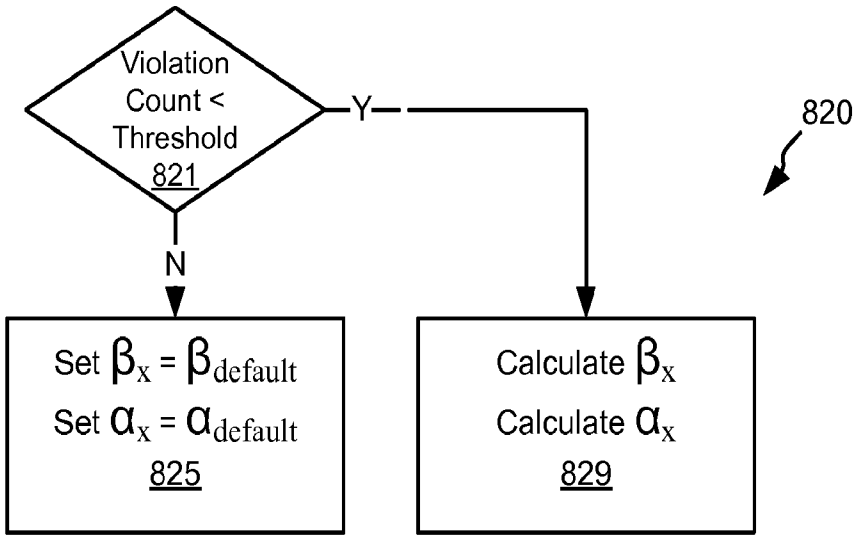

820
Violation Count < Threshold 821
Y
N
Set $\beta_x = \beta_{default}$
Set $\alpha_x = \alpha_{default}$
825
Calculate β_x
Calculate α_x
829

Fig. 5d

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2009/041867

A. CLASSIFICATION OF SUBJECT MATTER

G11B 20/10(2006.01)i, G11B 20/14(2006.01)i

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)

IPC 8: G11B 20/10; G11B 20/00; G11B 5/09; G11C 29/00; H04B 7/212

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

Korean utility models and applications for utility models
Japanese utility models and applications for utility models
(Chinese Patents and application for patent)

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

eKOMPASS(KIPO internal) & Keywords: decoder, scalar, detector, data, processing, threshold, multiplier, and dynamic

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 01-39188 A2 (SEAGATE TECHNOLOGY LLC) 31 May 2001 See abstract; page 7, line 23 - page 12, line 20; and figures 7-8.	1-20
A	US 2007-0230407 A1 (MICHAEL C. PETRIE et al.) 04 October 2007 See abstract; paragraph [0021] - paragraph [0035]; and figures 1-3.	1-20
A	US 7113356 B1 (ZINING WU) 26 September 2006 See abstract; column 5, line 59 - column 7, line 7; and figures 8A-10.	1-20
A	US 2006-0123285 A1 (DANIEL F. DE ARAUJO et al.) 08 June 2006 See abstract; paragraph [0015] - paragraph [0021]; and figures 3-4.	1-20

☐ 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 citation or other special reason (as specified)

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 JANUARY 2010 (25.01.2010)	**26 JANUARY 2010 (26.01.2010)**

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 139 Seonsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

BYUN, SUNG CHEAL

Telephone No. 82-42-481-8262

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2009/041867

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 01-39188 A2	31.05.2001	CN 1241198 C	08.02.2006
		CN 1241198 C0	08.02.2006
		CN 1391693 A	15.01.2003
		CN 1391693 A0	15.01.2003
		DE 10085214 T0	31.10.2002
		GB 0210724 D0	19.06.2002
		GB 2372142 A	14.08.2002
		GB 2372142 B	12.11.2003
		JP 2003-515861 A	07.05.2003
		JP 2003-515861 T	07.05.2003
		KR 10-0561798 B1	21.03.2006
		WO 01-39188 A3	22.11.2001
		WO 01-39188 A3	31.05.2001
US 2007-0230407 A1	04.10.2007	WO 2007-117859 A2	18.10.2007
		WO 2007-117859 A3	18.10.2007
		WO 2007-117859 A3	17.01.2008
US 7113356 B1	26.09.2006	US 07315427 B1	01.01.2008
US 2006-0123285 A1	08.06.2006	CN 100388217 C	14.05.2008
		CN 1776633 A	24.05.2006
		CN 1776633 C0	24.05.2006