US 20240312530A1

(19) **United States**

(12) **Patent Application Publication**

Parthasarathy et al.

(10) **Pub. No.: US 2024/0312530 A1**

(43) **Pub. Date: Sep. 19, 2024**

(54) **CLASSIFICATION OF ERROR RATE OF DATA RETRIEVED FROM MEMORY CELLS**

(71) Applicant: **Micron Technology, Inc.**, Boise, ID (US)

(72) Inventors: **Sivagnanam Parthasarathy**, Carlsbad, CA (US); **James Fitzpatrick**, Laguna Niguel, CA (US); **Patrick Robert Khayat**, San Diego, CA (US); **AbdelHakim S. Alhussien**, San Jose, CA (US)

(21) Appl. No.: **18/678,949**

(22) Filed: **May 30, 2024**

Related U.S. Application Data

(63) Continuation of application No. 16/807,065, filed on Mar. 2, 2020, now Pat. No. 12,009,034.

Publication Classification

(51) **Int. Cl.**

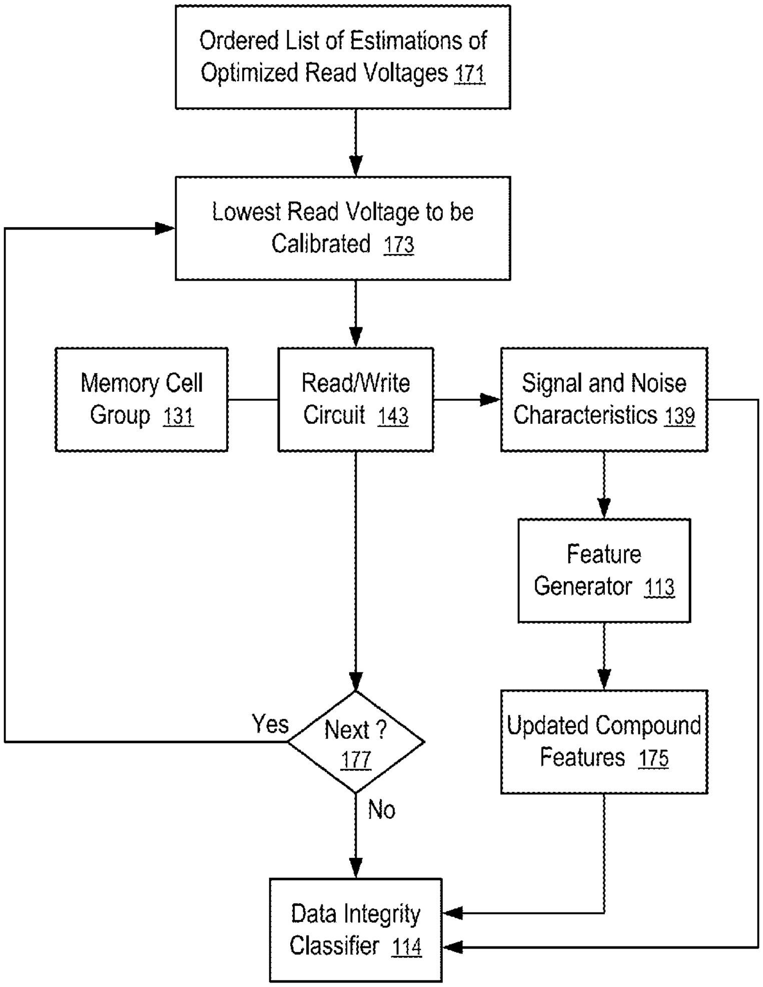
G11C 16/26	(2006.01)
G06F 9/30	(2006.01)
G06F 9/38	(2006.01)
G06F 18/243	(2006.01)
H03M 13/00	(2006.01)
H03M 13/01	(2006.01)

(52) **U.S. Cl.**

CPC ***G11C 16/26*** (2013.01); ***G06F 9/30021*** (2013.01); ***G06F 9/30101*** (2013.01); ***G06F 9/3804*** (2013.01); ***G06F 18/24323*** (2023.01); ***H03M 13/015*** (2013.01); ***H03M 13/612*** (2013.01)

(57) **ABSTRACT**

A memory sub-system configured to: measure a plurality of sets of signal and noise characteristics of a group of memory cells in a memory device; determine a plurality of optimized read voltages of the group of memory cells from the plurality of sets of signal and noise characteristics respectively; generate features from the plurality of sets of signal and noise characteristics, including at least one compound feature generated from the plurality of sets of signal and noise characteristics; generate, using the features, a classification of a bit error rate of data retrievable from the group of memory cells; and control an operation to read the group of memory cells based on the classification.


Ordered List of Estimations of Optimized Read Voltages 171
Lowest Read Voltage to be Calibrated 173
Memory Cell Group 131
Read/Write Circuit 143
Signal and Noise Characteristics 139
Feature Generator 113
Updated Compound Features 175
Yes
Next ? 177
No
Data Integrity Classifier 114

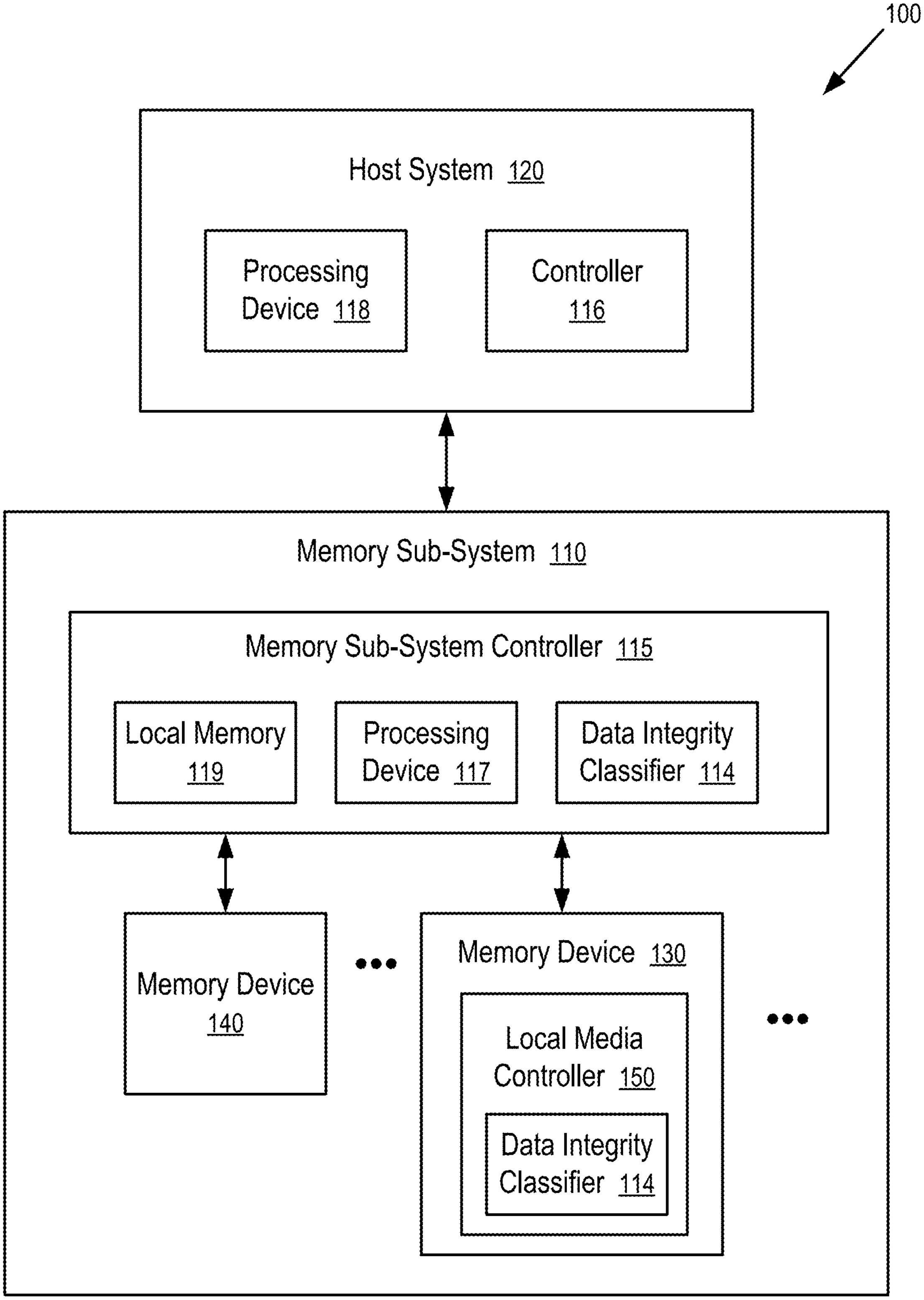

100
Host System 120
Processing Device 118
Controller 116
Memory Sub-System 110
Memory Sub-System Controller 115
Local Memory 119
Processing Device 117
Data Integrity Classifier 114
Memory Device 140
•••
Memory Device 130
Local Media Controller 150
Data Integrity Classifier 114
•••

FIG. 1

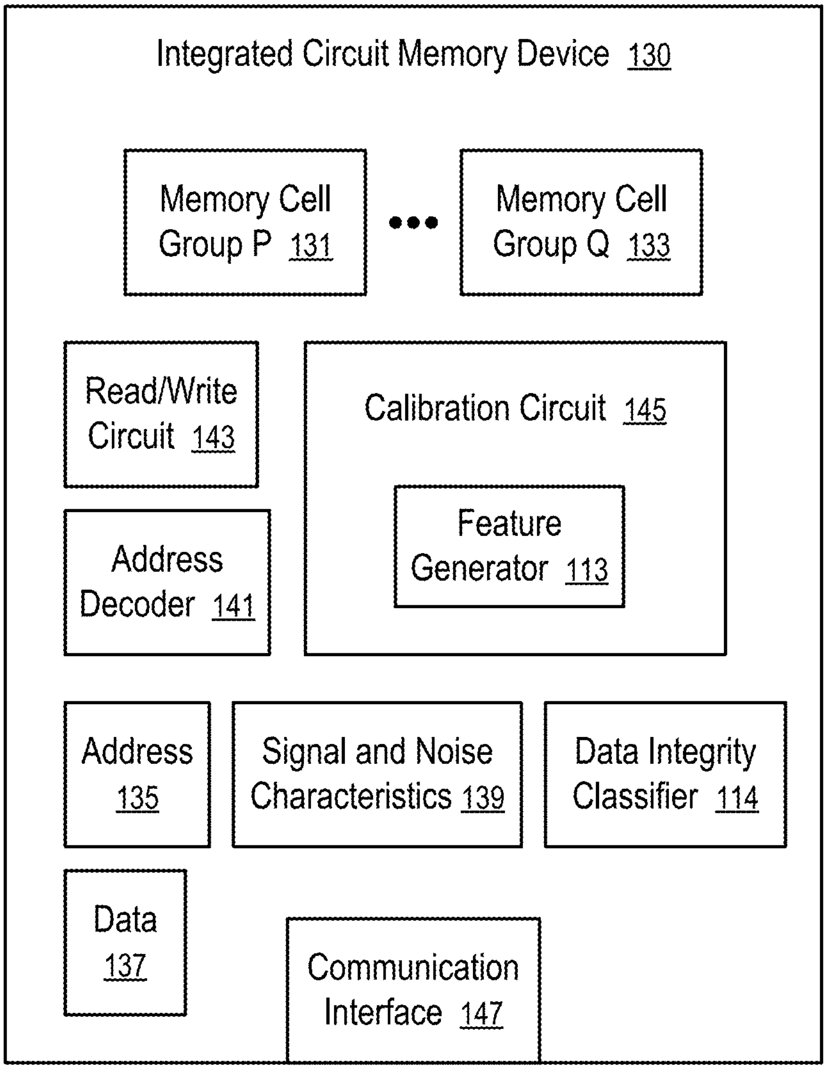

Integrated Circuit Memory Device 130
Memory Cell Group P 131
•••
Memory Cell Group Q 133
Read/Write Circuit 143
Calibration Circuit 145
Feature Generator 113
Address Decoder 141
Address 135
Signal and Noise Characteristics 139
Data Integrity Classifier 114
Data 137
Communication Interface 147

FIG. 2

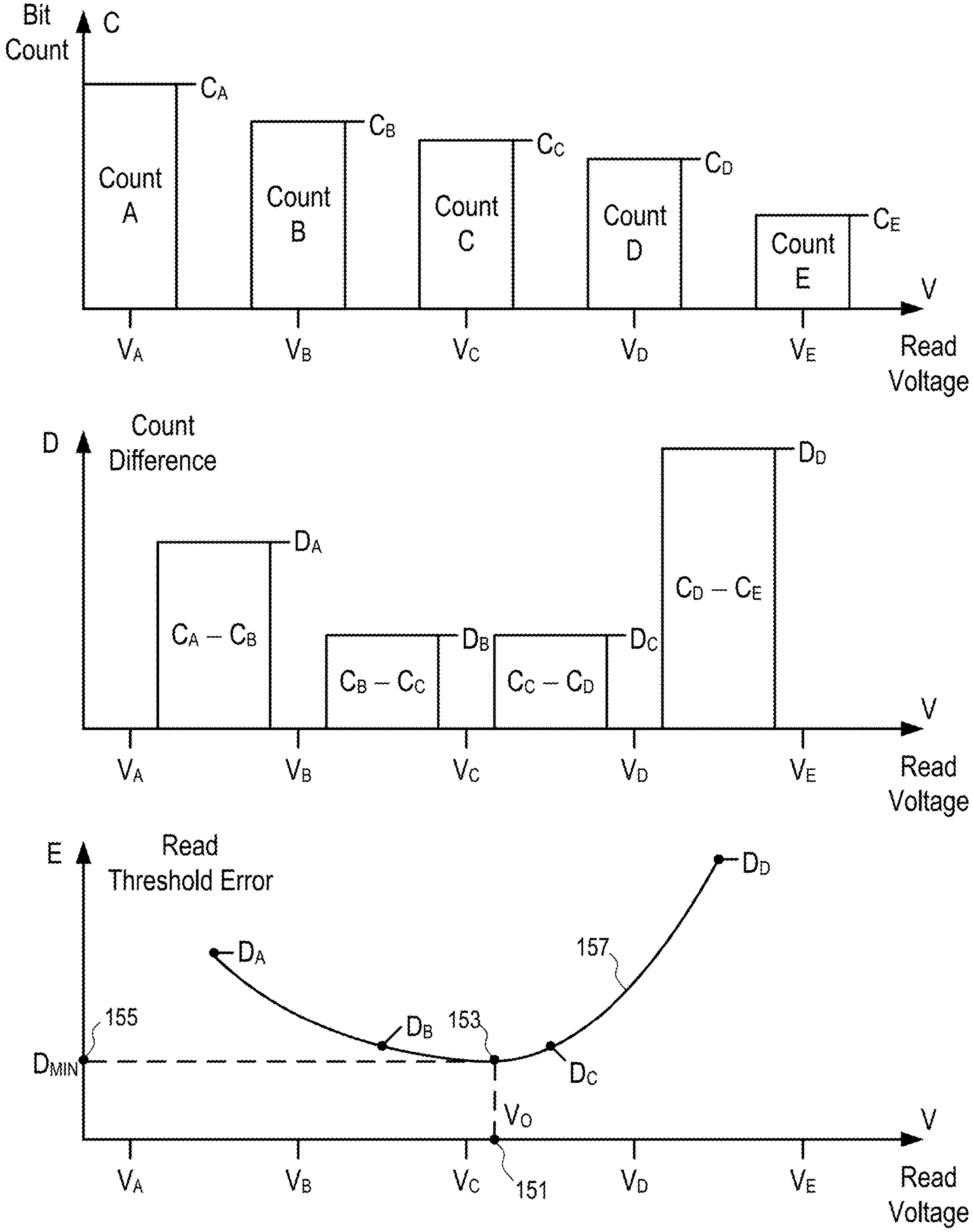

Bit Count
C
C_A
C_B
C_C
C_D
C_E
Count A
Count B
Count C
Count D
Count E
V
V_A
V_B
V_C
V_D
V_E
Read Voltage
D
Count Difference
D_A
D_B
D_C
D_D
$C_A - C_B$
$C_B - C_C$
$C_C - C_D$
$C_D - C_E$
V
V_A
V_B
V_C
V_D
V_E
Read Voltage
E
Read Threshold Error
D_A
D_B
D_C
D_D
157
155
153
D_{MIN}
V_O
151
V
V_A
V_B
V_C
V_D
V_E
Read Voltage

FIG. 3

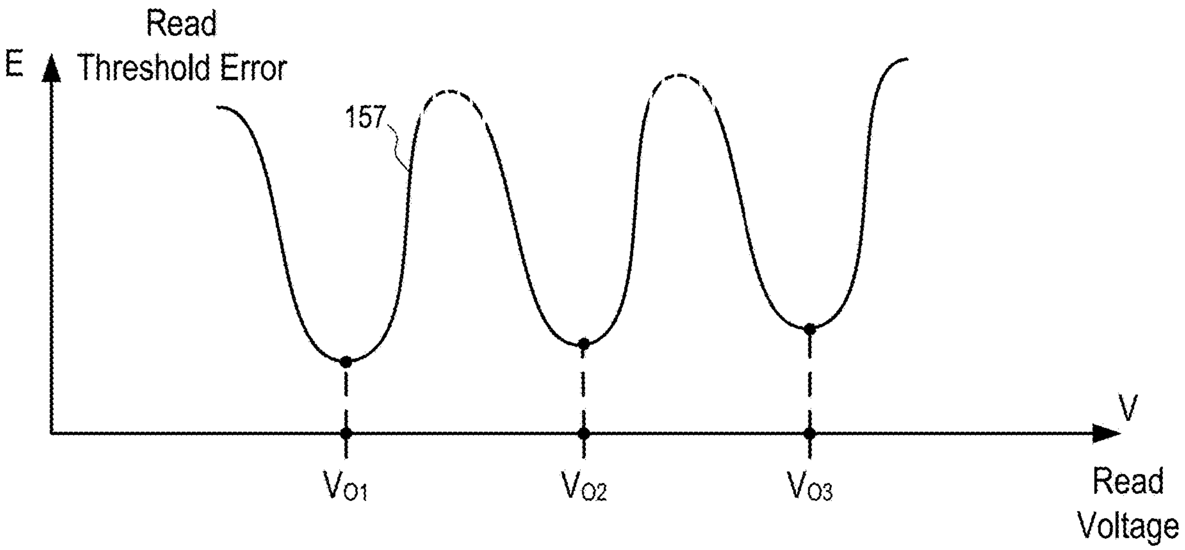

Read
Threshold Error
E
157
V
V_{O1}
V_{O2}
V_{O3}
Read
Voltage

FIG. 4

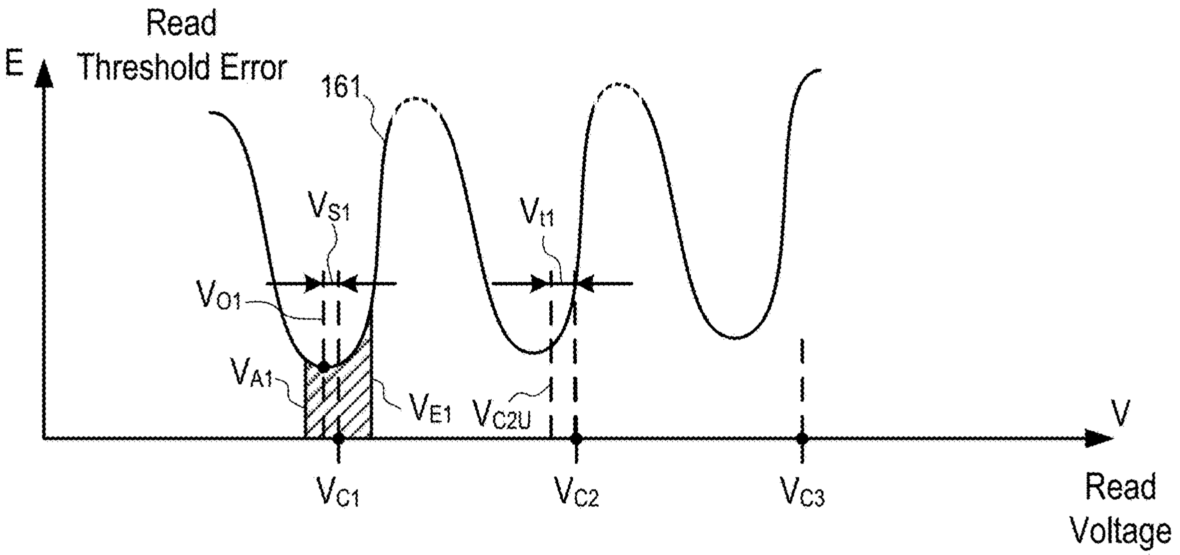

Read
Threshold Error
E
161
V_{S1}
V_{t1}
V_{O1}
V_{A1}
V_{E1}
V_{C2U}
V
V_{C1}
V_{C2}
V_{C3}
Read
Voltage

FIG. 5

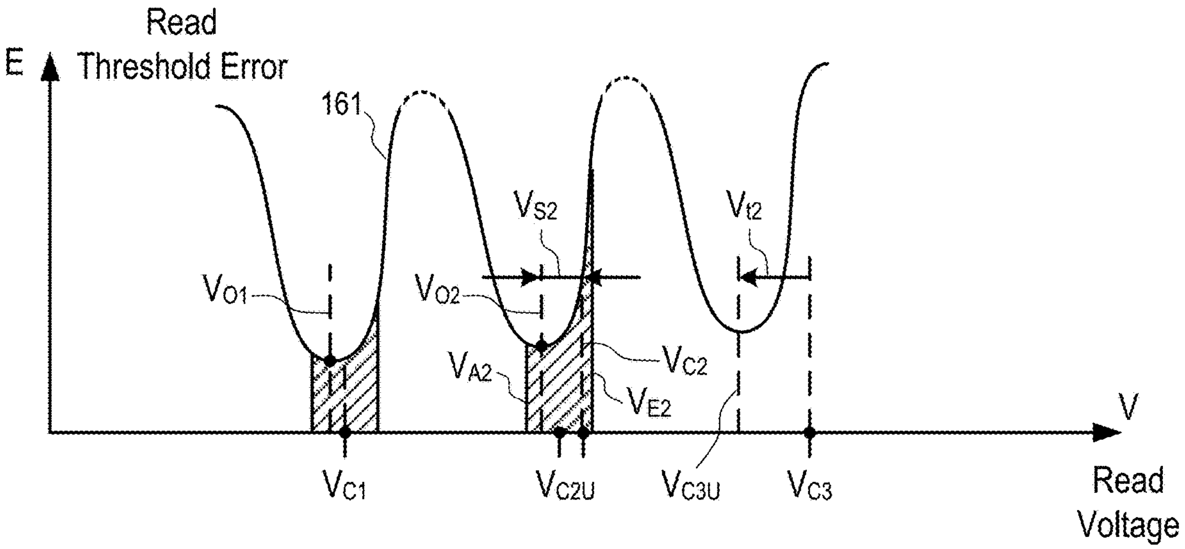

E
Read
Threshold Error
161
V_{O1}
V_{O2}
V_{S2}
V_{t2}
V_{A2}
V_{C2}
V_{E2}
V
V_{C1}
V_{C2U}
V_{C3U}
V_{C3}
Read
Voltage

FIG. 6

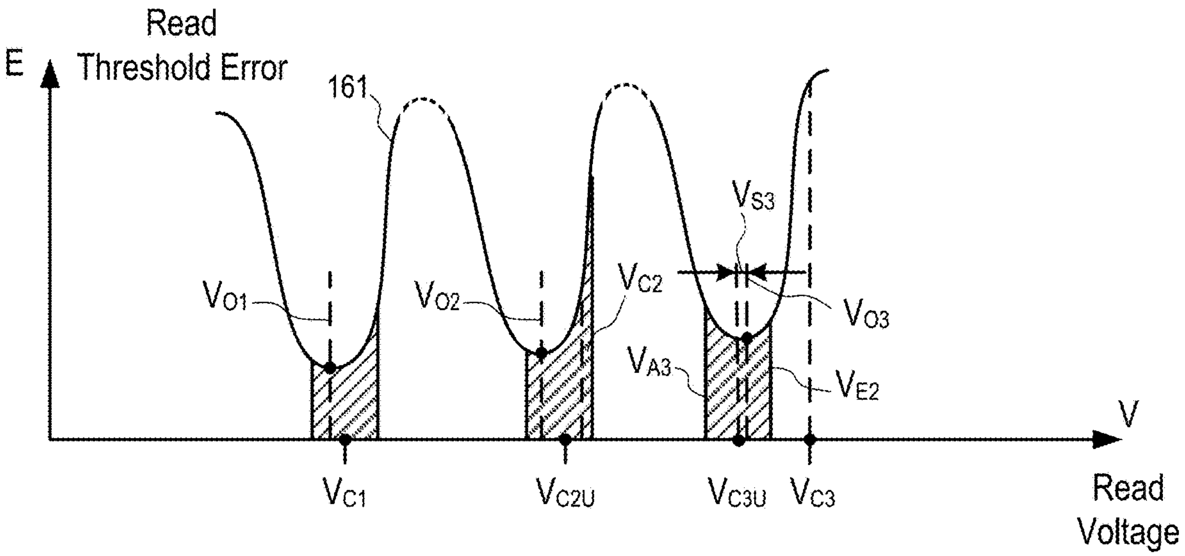

E
Read
Threshold Error
161
V_{O1}
V_{O2}
V_{C2}
V_{S3}
V_{O3}
V_{A3}
V_{E2}
V
V_{C1}
V_{C2U}
V_{C3U}
V_{C3}
Read
Voltage

FIG. 7

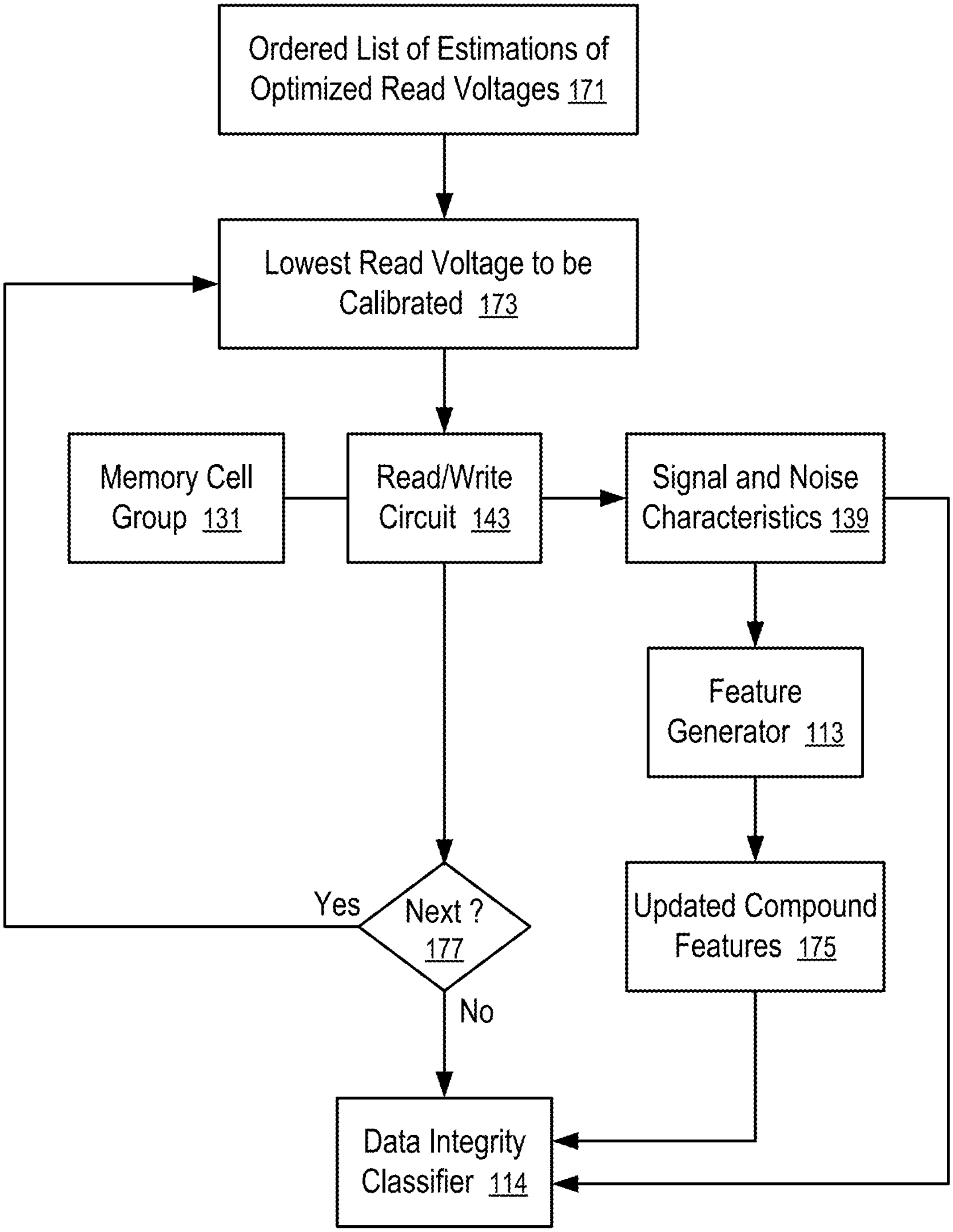

Ordered List of Estimations of Optimized Read Voltages 171
Lowest Read Voltage to be Calibrated 173
Memory Cell Group 131
Read/Write Circuit 143
Signal and Noise Characteristics 139
Feature Generator 113
Updated Compound Features 175
Yes
Next ? 177
No
Data Integrity Classifier 114

FIG. 8

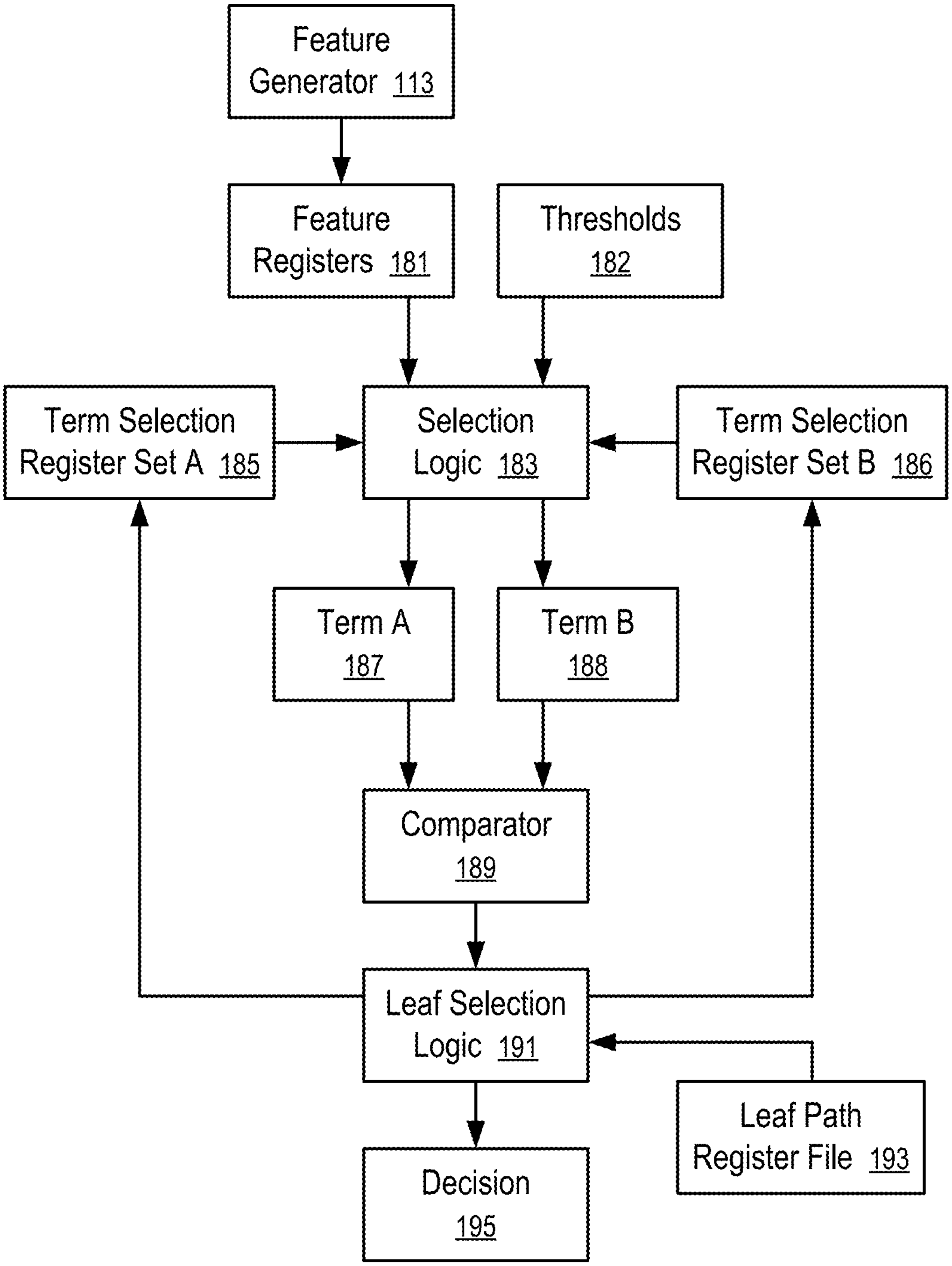

Feature Generator 113
Feature Registers 181
Thresholds 182
Term Selection Register Set A 185
Selection Logic 183
Term Selection Register Set B 186
Term A 187
Term B 188
Comparator 189
Leaf Selection Logic 191
Leaf Path Register File 193
Decision 195

FIG. 9

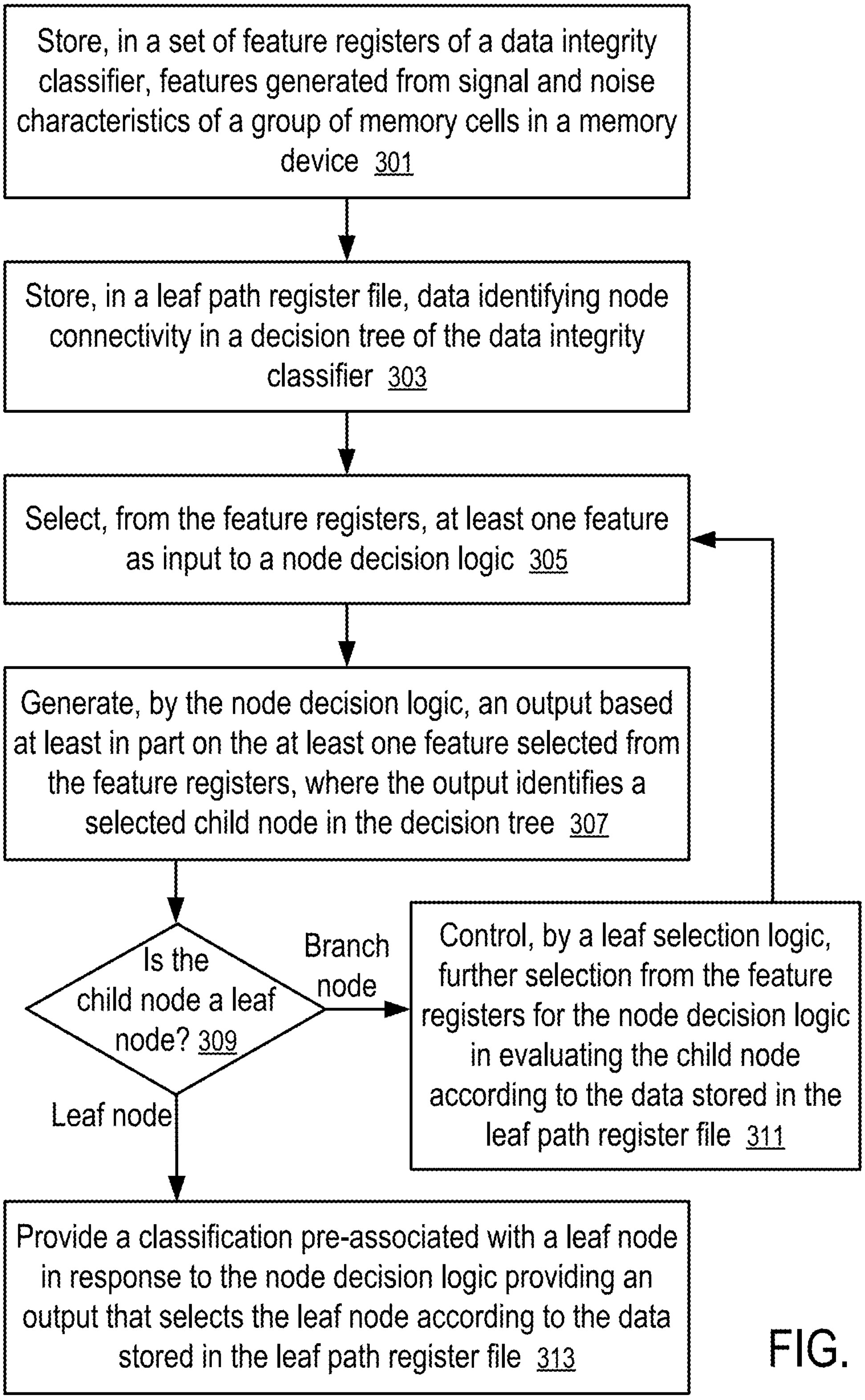

Store, in a set of feature registers of a data integrity classifier, features generated from signal and noise characteristics of a group of memory cells in a memory device 301
Store, in a leaf path register file, data identifying node connectivity in a decision tree of the data integrity classifier 303
Select, from the feature registers, at least one feature as input to a node decision logic 305
Generate, by the node decision logic, an output based at least in part on the at least one feature selected from the feature registers, where the output identifies a selected child node in the decision tree 307
Is the child node a leaf node? 309
Branch node
Control, by a leaf selection logic, further selection from the feature registers for the node decision logic in evaluating the child node according to the data stored in the leaf path register file 311
Leaf node
Provide a classification pre-associated with a leaf node in response to the node decision logic providing an output that selects the leaf node according to the data stored in the leaf path register file 313

FIG. 10

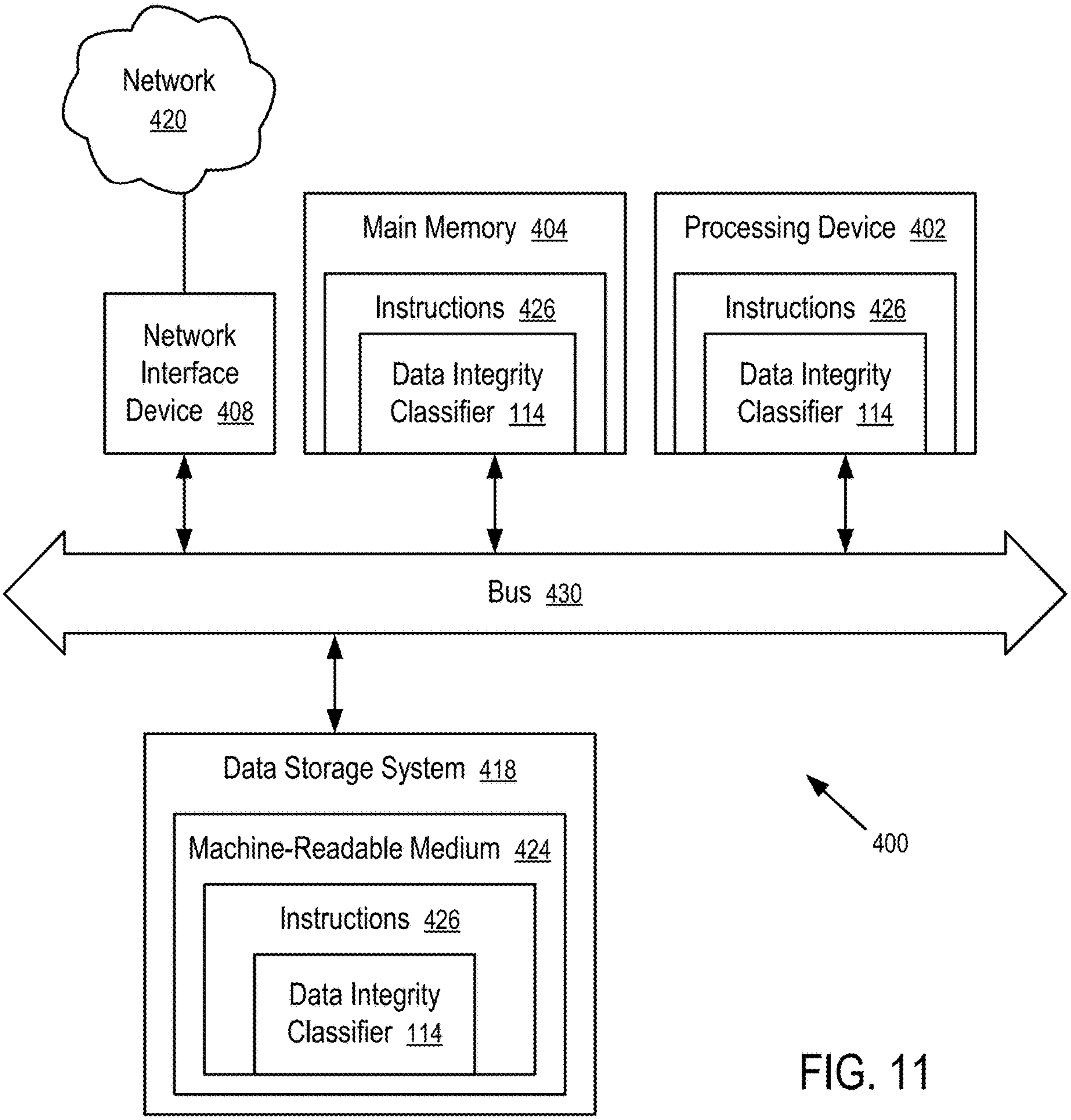

Network 420
Main Memory 404
Processing Device 402
Instructions 426
Data Integrity Classifier 114
Instructions 426
Data Integrity Classifier 114
Network Interface Device 408
Bus 430
Data Storage System 418
Machine-Readable Medium 424
Instructions 426
Data Integrity Classifier 114
400

FIG. 11

CLASSIFICATION OF ERROR RATE OF DATA RETRIEVED FROM MEMORY CELLS

RELATED APPLICATIONS

[0001] The present application is a continuation application of U.S. patent application Ser. No. 16/807,065 filed Mar. 2, 2020 and issued as U.S. Pat. No. 12,009,034 on Jun. 11, 2024, the entire disclosures of which application are hereby incorporated herein by reference.

TECHNICAL FIELD

[0002] At least some embodiments disclosed herein relate to memory systems in general, and more particularly, but not limited to memory systems having a binary classification decision tree for classification of error rate of data retrievable from memory cells in an integrated circuit memory device.

BACKGROUND

[0003] A memory sub-system can include one or more memory devices that store data. The memory devices can be, for example, non-volatile memory devices and volatile memory devices. In general, a host system can utilize a memory sub-system to store data at the memory devices and to retrieve data from the memory devices.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The embodiments are illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicate similar elements.

[0005] FIG. **1** illustrates an example computing system having a memory sub-system in accordance with some embodiments of the present disclosure.

[0006] FIG. **2** illustrates an integrated circuit memory device having a calibration circuit configured to measure signal and noise characteristics according to one embodiment.

[0007] FIG. **3** shows an example of measuring signal and noise characteristics to improve memory operations according to one embodiment.

[0008] FIGS. **4-7** illustrate self adapting iterative read calibration during the execution of a read command according to one embodiment.

[0009] FIG. **8** illustrates the generation of compound features for the classification of the error rate of data retrieved from memory cells according to one embodiment.

[0010] FIG. **9** illustrates an implementation of a data integrity classifier implemented based on binary classification decision tree according to one embodiment.

[0011] FIG. **10** shows a method of classifying the integrity of data retrieved from memory cells using features generated according to one embodiment.

[0012] FIG. **11** is a block diagram of an example computer system in which embodiments of the present disclosure can operate.

DETAILED DESCRIPTION

[0013] At least some aspects of the present disclosure are directed to a memory sub-system having a data integrity classifier implemented efficiently using a binary classification decision tree technique. A memory sub-system can be a storage device, a memory module, or a hybrid of a storage device and memory module. Examples of storage devices and memory modules are described below in conjunction with FIG. **1**. In general, a host system can utilize a memory sub-system that includes one or more components, such as memory devices that store data. The host system can provide data to be stored at the memory sub-system and can request data to be retrieved from the memory sub-system.

[0014] An integrated circuit memory cell (e.g., a flash memory cell) can be programmed to store data by the way of its state at a threshold voltage. For example, if the memory cell is configured/programmed in a state that allows a substantial current to pass the memory cell at the threshold voltage, the memory cell is storing a bit of one; and otherwise, the memory cell is storing a bit of zero. Further, a memory cell can store multiple bits of data by being configured/programmed differently at multiple threshold voltages. For example, the memory cell can store multiple bits of data by having a combination of states at the multiple threshold voltages; and different combinations of the states of the memory cell at the threshold voltages can be interpreted to represent different states of bits of data that is stored in the memory cell.

[0015] However, after the states of integrated circuit memory cells are configured/programmed using write operations to store data in the memory cells, the optimized threshold voltage for reading the memory cells can shift due to a number of factors, such as charge loss, read disturb, cross-temperature effect (e.g., write and read at different operating temperatures), etc., especially when a memory cell is programmed to store multiple bits of data.

[0016] Conventional calibration circuitry has been used to self-calibrate a memory region in applying read level signals to account for shift of threshold voltages of memory cells within the memory region. During the calibration, the calibration circuitry is configured to apply different test signals to the memory region to count the numbers of memory cells that output a specified data state for the test signals. Based on the counts, the calibration circuitry determines a read level offset value as a response to a calibration command.

[0017] At least some aspects of the present disclosure address the above and other deficiencies by classifying the bit error rate of data retrievable from memory cells using signal and noise characteristics measured near estimated locations of optimized read voltages of the memory cells and using at least compound features computed from the signal and noise characteristics measured for the multiple optimized read voltages. For example, a data integrity classifier generates a classification of the bit error rate of data retrievable from memory cells based on features calculated from signal and noise characteristics of memory cells measured for multiple read voltages. The features can include compound features that are calculated iteratively or progressively using signal and noise characteristics of memory cells measured for lower read voltages, while signal and noise characteristics of the memory cells are being measured for a higher read voltage. The classification of the bit error rate of the data retrievable from the memory cells can be used to control the operations to read data from the memory cells. The compound features can be computed efficiently using an iterative or progressive technique where the compound features are calculated initially based on signal and noise characteristics measured for lower optimized read voltages while signal and noise characteristics for higher optimized read voltages are being measured or have not yet been measured. The compound features are further updated based on the signal and noise characteristics measured for each higher optimized read voltage when the signal and noise characteristics for the higher optimized read voltage become available. The classification result of the bit error rate can be used to select a processing path in reading data from the memory cells. For example, based on the bit error rate classification, the memory sub-system can decide whether to further calibrate the read voltages, to skip error detection and data recovery, to skip reading the memory cells for soft bit data by applying read voltages that have a predetermined offset from the optimized read voltages, etc.

[0018] For example, a memory cell programmed to store multiple bits of data is to be read using multiple read voltages to determine the states of the memory cells at the read voltages and thus the multiple bits stored in the memory cell. The optimized read voltages for reading the multiple states can shift due to data retention effects, such as quick charge loss (QCL), storage charge loss (SCL), etc., and/or other effects. A calibration operation can be performed for each of the read voltages to determine the respective optimized read voltages. During the calibration of each read voltage, a set of signal and noise characteristics of the memory cells can be measured. The multiple set of signal and noise characteristics associated with the multiple optimized read voltages can be used to construct features as input for a predictive model for classifying the bit error rate of data that can be retrieved from the memory cells using the multiple optimized read voltages. Such features can include compound features. A compound feature is based on multiple sets of signal and noise characteristics associated with multiple optimized read voltages respectively.

[0019] In some situations, the optimized read voltages can shift over a period of time in a same direction (e.g., towards lower voltages, or towards higher voltages). In general, different optimized read voltages can shift by different amounts, where the higher ones in the optimized read voltages may shift more than the lower ones in the optimized read voltages. A predictive model can be used to predict the shift of a higher optimized read voltage based on the shift(s) of one or more lower optimized read voltages. Thus, once the lower optimized read voltages are determined through calibration, the shift of an optimized read voltage higher than the lower optimized read voltages can be predicted/estimated to correct the initial estimation of the expected location of the higher optimized read voltage. Using the corrected estimation, the calibration for the higher optimized read voltage can be performed to identify an optimized read voltage with improved precision and/or to avoid a failure in calibration.

[0020] FIG. **1** illustrates an example computing system **100** that includes a memory sub-system **110** in accordance with some embodiments of the present disclosure. The memory sub-system **110** can include media, such as one or more volatile memory devices (e.g., memory device **140**), one or more non-volatile memory devices (e.g., memory device **130**), or a combination of such.

[0021] A memory sub-system **110** can be a storage device, a memory module, or a hybrid of a storage device and memory module. Examples of a storage device include a solid-state drive (SSD), a flash drive, a universal serial bus (USB) flash drive, an embedded multi-media controller (eMMC) drive, a universal flash storage (UFS) drive, a secure digital (SD) card, and a hard disk drive (HDD). Examples of memory modules include a dual in-line memory module (DIMM), a small outline DIMM (SO-DIMM), and various types of non-volatile dual in-line memory module (NVDIMM).

[0022] The computing system **100** can be a computing device such as a desktop computer, laptop computer, network server, mobile device, a vehicle (e.g., airplane, drone, train, automobile, or other conveyance), internet of things (IoT) enabled device, embedded computer (e.g., one included in a vehicle, industrial equipment, or a networked commercial device), or such computing device that includes memory and a processing device.

[0023] The computing system **100** can include a host system **120** that is coupled to one or more memory sub-systems **110**. FIG. **1** illustrates one example of a host system **120** coupled to one memory sub-system **110**. As used herein, "coupled to" or "coupled with" generally refers to a connection between components, which can be an indirect communicative connection or direct communicative connection (e.g., without intervening components), whether wired or wireless, including connections such as electrical, optical, magnetic, etc.

[0024] The host system **120** can include a processor chipset (e.g., processing device **118**) and a software stack executed by the processor chipset. The processor chipset can include one or more cores, one or more caches, a memory controller (e.g., controller **116**) (e.g., NVDIMM controller), and a storage protocol controller (e.g., PCIe controller, SATA controller). The host system **120** uses the memory sub-system **110**, for example, to write data to the memory sub-system **110** and read data from the memory sub-system **110**.

[0025] The host system **120** can be coupled to the memory sub-system **110** via a physical host interface. Examples of a physical host interface include, but are not limited to, a serial advanced technology attachment (SATA) interface, a peripheral component interconnect express (PCIe) interface, universal serial bus (USB) interface, fibre channel, serial attached SCSI (SAS), a double data rate (DDR) memory bus, small computer system interface (SCSI), a dual in-line memory module (DIMM) interface (e.g., DIMM socket interface that supports double data rate (DDR)), open NAND flash interface (ONFI), double data rate (DDR), low power double data rate (LPDDR), or any other interface. The physical host interface can be used to transmit data between the host system **120** and the memory sub-system **110**. The host system **120** can further utilize an NVM express (NVMe) interface to access components (e.g., memory devices **130**) when the memory sub-system **110** is coupled with the host system **120** by the PCIe interface. The physical host interface can provide an interface for passing control, address, data, and other signals between the memory sub-system **110** and the host system **120**. FIG. **1** illustrates a memory sub-system **110** as an example. In general, the host system **120** can access multiple memory sub-systems via a same communication connection, multiple separate communication connections, and/or a combination of communication connections.

[0026] The processing device **118** of the host system **120** can be, for example, a microprocessor, a central processing unit (CPU), a processing core of a processor, an execution unit, etc. In some instances, the controller **116** can be referred to as a memory controller, a memory management unit, and/or an initiator. In one example, the controller **116** controls the communications over a bus coupled between the host system **120** and the memory sub-system **110**. In general, the controller **116** can send commands or requests to the memory sub-system **110** for desired access to memory devices **130**,**140**. The controller **116** can further include interface circuitry to communicate with the memory sub-system **110**. The interface circuitry can convert responses received from the memory sub-system **110** into information for the host system **120**.

[0027] The controller **116** of the host system **120** can communicate with the controller **115** of the memory sub-system **110** to perform operations such as reading data, writing data, or erasing data at the memory devices **130**,**140** and other such operations. In some instances, the controller **116** is integrated within the same package of the processing device **118**. In other instances, the controller **116** is separate from the package of the processing device **118**. The controller **116** and/or the processing device **118** can include hardware such as one or more integrated circuits (ICs) and/or discrete components, a buffer memory, a cache memory, or a combination thereof. The controller **116** and/or the processing device **118** can be a microcontroller, special purpose logic circuitry (e.g., a field programmable gate array (FPGA), an application specific integrated circuit (ASIC), etc.), or another suitable processor.

[0028] The memory devices **130**, **140** can include any combination of the different types of non-volatile memory components and/or volatile memory components. The volatile memory devices (e.g., memory device **140**) can be, but are not limited to, random access memory (RAM), such as dynamic random access memory (DRAM) and synchronous dynamic random access memory (SDRAM).

[0029] Some examples of non-volatile memory components include a negative-and (or, NOT AND) 1(NAND) type flash memory and write-in-place memory, such as three-dimensional cross-point ("3D cross-point") memory. A cross-point array of non-volatile memory can perform bit storage based on a change of bulk resistance, in conjunction with a stackable cross-gridded data access array. Additionally, in contrast to many flash-based memories, cross-point non-volatile memory can perform a write in-place operation, where a non-volatile memory cell can be programmed without the non-volatile memory cell being previously erased. NAND type flash memory includes, for example, two-dimensional NAND (2D NAND) and three-dimensional NAND (3D NAND).

[0030] Each of the memory devices **130** can include one or more arrays of memory cells. One type of memory cell, for example, single level cells (SLC) can store one bit per cell. Other types of memory cells, such as multi-level cells (MLCs), triple level cells (TLCs), quad-level cells (QLCs), and penta-level cells (PLC) can store multiple bits per cell. In some embodiments, each of the memory devices **130** can include one or more arrays of memory cells such as SLCs, MLCs, TLCs, QLCs, or any combination of such. In some embodiments, a particular memory device can include an SLC portion, and an MLC portion, a TLC portion, or a QLC portion of memory cells. The memory cells of the memory devices **130** can be grouped as pages that can refer to a logical unit of the memory device used to store data. With some types of memory (e.g., NAND), pages can be grouped to form blocks.

[0031] Although non-volatile memory devices such as 3D cross-point type and NAND type memory (e.g., 2D NAND, 3D NAND) are described, the memory device **130** can be based on any other type of non-volatile memory, such as read-only memory (ROM), phase change memory (PCM), self-selecting memory, other chalcogenide based memories, ferroelectric transistor random-access memory (FeTRAM), ferroelectric random access memory (FeRAM), magneto random access memory (MRAM), spin transfer torque (STT)-MRAM, conductive bridging RAM (CBRAM), resistive random access memory (RRAM), oxide based RRAM (OxRAM), negative-or (NOR) flash memory, and electrically erasable programmable read-only memory (EEPROM).

[0032] A memory sub-system controller **115** (or controller **115** for simplicity) can communicate with the memory devices **130** to perform operations such as reading data, writing data, or erasing data at the memory devices **130** and other such operations (e.g., in response to commands scheduled on a command bus by controller **116**). The controller **115** can include hardware such as one or more integrated circuits (ICs) and/or discrete components, a buffer memory, or a combination thereof. The hardware can include digital circuitry with dedicated (i.e., hard-coded) logic to perform the operations described herein. The controller **115** can be a microcontroller, special purpose logic circuitry (e.g., a field programmable gate array (FPGA), an application specific integrated circuit (ASIC), etc.), or another suitable processor.

[0033] The controller **115** can include a processing device **117** (processor) configured to execute instructions stored in a local memory **119**. In the illustrated example, the local memory **119** of the controller **115** includes an embedded memory configured to store instructions for performing various processes, operations, logic flows, and routines that control operation of the memory sub-system **110**, including handling communications between the memory sub-system **110** and the host system **120**.

[0034] In some embodiments, the local memory **119** can include memory registers storing memory pointers, fetched data, etc. The local memory **119** can also include read-only memory (ROM) for storing micro-code. While the example memory sub-system **110** in FIG. **1** has been illustrated as including the controller **115**, in another embodiment of the present disclosure, a memory sub-system **110** does not include a controller **115**, and can instead rely upon external control (e.g., provided by an external host, or by a processor or controller separate from the memory sub-system).

[0035] In general, the controller **115** can receive commands or operations from the host system **120** and can convert the commands or operations into instructions or appropriate commands to achieve the desired access to the memory devices **130**. The controller **115** can be responsible for other operations such as wear leveling operations, garbage collection operations, error detection and error-correcting code (ECC) operations, encryption operations, caching operations, and address translations between a logical address (e.g., logical block address (LBA), namespace) and a physical address (e.g., physical block address) that are associated with the memory devices **130**. The controller **115** can further include host interface circuitry to communicate with the host system **120** via the physical host interface. The host interface circuitry can convert the commands received from the host system into command instructions to access the memory devices **130** as well as convert responses associated with the memory devices **130** into information for the host system **120**.

[0036] The memory sub-system **110** can also include additional circuitry or components that are not illustrated. In some embodiments, the memory sub-system **110** can include a cache or buffer (e.g., DRAM) and address circuitry (e.g., a row decoder and a column decoder) that can receive an address from the controller **115** and decode the address to access the memory devices **130**.

[0037] In some embodiments, the memory devices **130** include local media controllers **150** that operate in conjunction with the memory sub-system controller **115** to execute operations on one or more memory cells of the memory devices **130**. An external controller (e.g., memory sub-system controller **115**) can externally manage the memory device **130** (e.g., perform media management operations on the memory device **130**). In some embodiments, a memory device **130** is a managed memory device, which is a raw memory device combined with a local controller (e.g., local controller **150**) for media management within the same memory device package. An example of a managed memory device is a managed NAND (MNAND) device.

[0038] The controller **115** and/or a memory device **130** can include a data integrity classifier **114** that has a feature generator configured to generate compound features as input for the classification of the bit error rate of data retrievable from the memory cells using multiple optimized read voltages. The compound features are generated based on multiple sets of signal and noise characteristics measured during the calibration of the multiple optimized read voltages respectively. In some embodiments, the controller **115** in the memory sub-system **110** includes at least a portion of the data integrity classifier **114**. In other embodiments, or in combination, the controller **116** and/or the processing device **118** in the host system **120** includes at least a portion of the data integrity classifier **114**. For example, the controller **115**, the controller **116**, and/or the processing device **118** can include logic circuitry implementing the data integrity classifier **114**. For example, the controller **115**, or the processing device **118** (processor) of the host system **120**, can be configured to execute instructions stored in memory for performing the operations of the data integrity classifier **114** described herein. In some embodiments, the data integrity classifier **114** is implemented in an integrated circuit chip disposed in the memory sub-system **110**. In other embodiments, the data integrity classifier **114** can be part of firmware of the memory sub-system **110**, an operating system of the host system **120**, a device driver, or an application, or any combination therein.

[0039] The feature generator of the data integrity classifier **114** can receive multiple sets of signal and noise characteristics measured for multiple optimized read voltages of the memory cells in the memory device **130** and process the signal and noise characteristics to generate compound features for the data integrity classifier **114** of the bit error rate of the data retrievable using the multiple optimized read voltages.

[0040] For example, the data integrity classifier **114** can be implemented using a binary classification decision tree (BCDT) technique, or another decision tree based classification technique. For example, the data integrity classifier **114** can be implemented using an artificial neural network (ANN). The data integrity classifier **114** can be trained using a machine learning technique (e.g., a supervised machine learning technique) to compute a classification of the bit error rate in data retrievable from memory cells using a set of optimized read voltages, based on features constructed using signal and noise characteristics of the memory cells measured during the calibration/determination of the optimized read voltages.

[0041] For example, data can be encoded to contain redundant information for storing in memory cells. For example, error correction code (ECC) or low-density parity-check (LDPC) code can be used to encode data for storing in memory cells. The data retrieved from the memory cells can be decoded in error detection and recovery operations. When the decoding is successful, the bit error rate in the retrieved data can be calculated and/or classified. When the decoding is not successful, the bit error rate is in a category of too high for decoding. A training data set can be generated by computing features from signal and noise characteristics of the memory cells measured during the calibration/determination of optimized read voltages and the bit error rate/classification of the data retrieved using the optimized read voltages, where the bit error rate/classification is calculated from the result of decoding the retrieved data. The training data set can be used to train the data integrity classifier **114** to minimize the differences between the bit error rate/classification predicted by the data integrity classifier **114** using the features and the corresponding the bit error rate/classification calculated from the result of decoding the retrieved data.

[0042] For example, the data integrity classifier **114** can be trained to predict whether the retrieved data can decode successfully, and if so the estimated bit error rate of the retrieved data. For example, the memory sub-system **110** can have multiple decoders that have different requirements/inputs and different performance levels in power consumption, error recovery capability, latency, etc. The data integrity classifier **114** can be trained to predict which of the decoders, if any, can successfully decode the retrieved data. After the data integrity classifier **114** is trained, the prediction of the data integrity classifier **114** can be used to select a decoder before attempting to decode.

[0043] A compound feature can be constructed as a function of multiple sets of signal and noise characteristics measured for multiple optimized read voltages respectively. An example of the compound feature is a minimum (or a maximum) of a quantity across the multiple sets of signal and noise characteristics. Another example of the compound feature is a minimum (or a maximum) of the sum (or difference) of a quantity in two sets of signal and noise characteristics associated with two adjacent optimized read voltages when the optimized read voltages are sorted in an increasing order.

[0044] Optionally, the feature generator **113** can start the computation of the compound feature after receiving the multiple sets of signal and noise characteristics associated with the multiple optimized read voltages respectively.

[0045] Preferably, the feature generator **113** can start the computation of the compound feature before the multiple sets of signal and noise characteristics are all available. The feature generator **113** can iteratively or progressively compute the compound feature based on available sets of signal and noise characteristics. When the signal and noise characteristics associated with an optimized read voltage becomes available, the compound feature computed based on signal and noise characteristics of optimized read voltages lower than the optimized read voltage can be updated, while another optimized read voltage is being calibrated to measure its signal and noise characteristics. Thus, the compound feature can be built on the fly as more optimized read voltages are calibrated and their signal and noise characteristics measured. Such an iterative or progressive approach in calculating the compound feature can reduce the resource requirements of the feature generator **113** and/or its latency in providing the computation results relative to the availability of the last set of signal and noise characteristics of the highest optimized read voltage.

[0046] FIG. **2** illustrates an integrated circuit memory device **130** having a calibration circuit **145** configured to measure signal and noise characteristics according to one embodiment. For example, the memory devices **130** in the memory sub-system **110** of FIG. **1** can be implemented using the integrated circuit memory device **130** of FIG. **2**.

[0047] The integrated circuit memory device **130** can be enclosed in a single integrated circuit package. The integrated circuit memory device **130** includes multiple groups **131**, . . . , **133** of memory cells that can be formed in one or more integrated circuit dies. A typical memory cell in a group **131**, . . . , **133** can be programmed to store one or more bits of data.

[0048] Some of the memory cells in the integrated circuit memory device **130** can be configured to be operated together for a particular type of operations. For example, memory cells on an integrated circuit die can be organized in planes, blocks, and pages. A plane contains multiple blocks; a block contains multiple pages; and a page can have multiple strings of memory cells. For example, an integrated circuit die can be the smallest unit that can independently execute commands or report status; identical, concurrent operations can be executed in parallel on multiple planes in an integrated circuit die; a block can be the smallest unit to perform an erase operation; and a page can be the smallest unit to perform a data program operation (to write data into memory cells). Each string has its memory cells connected to a common bitline; and the control gates of the memory cells at the same positions in the strings in a block or page are connected to a common wordline. Control signals can be applied to wordlines and bitlines to address the individual memory cells.

[0049] The integrated circuit memory device **130** has a communication interface **147** to receive an address **135** from the controller **115** of a memory sub-system **110** and to provide the data **137** retrieved from the memory address **135**. An address decoder **141** of the integrated circuit memory device **130** converts the address **135** into control signals to select the memory cells in the integrated circuit memory device **130**; and a read/write circuit **143** of the integrated circuit memory device **130** performs operations to determine data stored in the addressed memory cells or to program the memory cells to have states corresponding to storing the data **137**.

[0050] The integrated circuit memory device **130** has a calibration circuit **145** configured to determine measurements of signal and noise characteristics **139** of memory cells in a group (e.g., **131**, . . . , or **133**) and provide the signal and noise characteristics **139** to the controller **115** of a memory sub-system **110** via the communication interface **147**.

[0051] In at least some embodiments, the calibration circuit **145** also provides, to the controller **115** via the communication interface **147**, the signal and noise characteristics **139** measured to determine the read level offset value. In some embodiments, the read level offset value can be used to understand, quantify, or estimate the signal and noise characteristics **139**. In other embodiments, the statistics of memory cells in a group or region that has a particular state at one or more test voltages can be provided as the signal and noise characteristics **139**.

[0052] For example, the calibration circuit **145** can measure the signal and noise characteristics **139** by reading different responses from the memory cells in a group (e.g., **131**, . . . , **133**) by varying operating parameters used to read the memory cells, such as the voltage(s) applied during an operation to read data from memory cells.

[0053] For example, the calibration circuit **145** can measure the signal and noise characteristics **139** on the fly when executing a command to read the data **137** from the address **135**. Since the signal and noise characteristics **139** is measured as part of the operation to read the data **137** from the address **135**, the signal and noise characteristics **139** can be used in the feature generator **113** with reduced or no penalty on the latency in the execution of the command to read the data **137** from the address **135**.

[0054] The calibration circuit **145** is configured to calibrate the read voltages of a group of memory cells (e.g., **131** or **133**) one after another in the order of ascending read voltages, starting from the lowest optimized read voltage to the highest optimized read voltage. During a calibration/determination of a particular optimized read voltage, the signal and noise characteristics **139** measured for the optimized read voltages lower than the particular optimized read voltage are available to the feature generator **113** to generate/calculate compound features from the available signal and noise characteristics **139**. When the calibration circuit **145** completes the calibration of the particular optimized read voltage, its signal and noise characteristics **139** becomes available for the feature generator **113** to update the compound features to include the consideration of the signal and noise characteristics **139** of the particular optimized read voltage. The feature generator **113** can perform the updating of the compound features during the time period in which the calibration circuit **145** calibrates the next optimized read voltage that is higher than the particular optimized read voltage. The updating can be repeated for successive higher optimized read voltages until all optimized read voltages are calibrated, their signal and noise characteristics **139** measured and used to update the compound features.

[0055] FIG. **3** shows an example of measuring signal and noise characteristics **139** to improve memory operations according to one embodiment.

[0056] In FIG. **3**, the calibration circuit **145** applies different read voltages V_A, V_B, V_C, V_D, and V_E to read the states of memory cells in a group (e.g., **131**, . . . , or **133**). In general, more or less read voltages can be used to generate the signal and noise characteristics **139**.

[0057] As a result of the different voltages applied during the read operation, a same memory cell in the group (e.g., **131**, . . . , or **133**) may show different states. Thus, the counts C_A, C_B, C_C, C_D, and C_E of memory cells having a predetermined state at different read voltages V_A, V_B, V_C, V_D, and V_E can be different in general. The predetermined state can be a state of having substantial current passing through the memory cells, or a state of having no substantial current passing through the memory cells. The counts C_A, C_B, C_C, C_D, and C_E can be referred to as bit counts.

[0058] The calibration circuit **145** can measure the bit counts by applying the read voltages V_A, V_B, V_C, V_D, and V_E one at a time on the group (e.g., **131**, . . . , or **133**) of memory cells.

[0059] Alternatively, the group (e.g., **131**, . . . , or **133**) of memory cells can be configured as multiple subgroups; and the calibration circuit **145** can measure the bit counts of the subgroups in parallel by applying the read voltages V_A, V_B, V_C, V_D, and V_E. The bit counts of the subgroups are considered as representative of the bit counts in the entire group (e.g., **131**, . . . , or **133**). Thus, the time duration of obtaining the counts C_A, C_B, C_C, C_D, and C_E can be reduced.

[0060] In some embodiments, the bit counts C_A, C_B, C_C, C_D, and C_E are measured during the execution of a command to read the data **137** from the address **135** that is mapped to one or more memory cells in the group (e.g., **131**, . . . , or **133**). Thus, the controller **115** does not need to send a separate command to request for the signal and noise characteristics **139** that is based on the bit counts C_A, C_B, C_C, C_D, and C_E.

[0061] The differences between the bit counts of the adjacent voltages are indicative of the errors in reading the states of the memory cells in the group (e.g., **133**, . . . , or **133**).

[0062] For example, the count difference D_A is calculated from C_A-C_B, which is an indication of read threshold error introduced by changing the read voltage from V_A to V_B.

[0063] Similarly, $D_B=C_B-C_C$; $D_C=C_C-C_D$; and $D_D=C_D-C_E$.

[0064] The curve **157**, obtained based on the count differences D_A, D_B, D_C, and D_D, represents the prediction of read threshold error E as a function of the read voltage. From the curve **157** (and/or the count differences), the optimized read voltage V_O can be calculated as the point **153** that provides the lowest read threshold error D_{MIN} on the curve **157**.

[0065] In one embodiment, the calibration circuit **145** computes the optimized read voltage V_O and causes the read/write circuit **143** to read the data **137** from the address **135** using the optimized read voltage V_O.

[0066] Alternatively, the calibration circuit **145** can provide, via the communication interface **147** to the controller **115** of the memory sub-system **110**, the count differences D_A, D_B, D_C, and D_D and/or the optimized read voltage V_O calculated by the calibration circuit **145**.

[0067] FIG. **3** illustrates an example of generating a set of statistical data (e.g., bit counts and/or count differences) for reading at an optimized read voltage V_O. In general, a group of memory cells can be configured to store more than one bit in a memory cell; and multiple read voltages are used to read the data stored in the memory cells. A set of statistical data can be similarly measured for each of the read voltages to identify the corresponding optimize read voltage, where the test voltages in each set of statistical data are configured in the vicinity of the expected location of the corresponding optimized read voltage. Thus, the signal and noise characteristics **139** measured for a memory cell group (e.g., **131** or **133**) can include multiple sets of statistical data measured for the multiple threshold voltages respectively.

[0068] For example, the controller **115** can instruct the memory device **130** to perform a read operation by providing an address **135** and at least one read control parameter. For example, the read control parameter can be a read voltage that is suggested, estimated, or predicted by the controller **115**.

[0069] The memory device **130** can perform the read operation by determining the states of memory cells at the address **135** at a read voltage and provide the data **137** according to the determined states.

[0070] During the read operation, the calibration circuit **145** of the memory device **130** generates the signal and noise characteristics **139**. The data **137** and the signal and noise characteristics **139** are provided from the memory device **130** to the controller **115** as a response. Alternatively, the processing of the signal and noise characteristics **139** can be performed at least in part using logic circuitry configured in the memory device **130**. For example, the processing of the signal and noise characteristics **139** can be implemented partially or entirely using the processing logic configured in the memory device **130**. For example, the processing logic can be implemented using complementary metal-oxide-semiconductor (CMOS) circuitry formed under the array of memory cells on an integrated circuit die of the memory device **130**. For example, the processing logic can be formed, within the integrated circuit package of the memory device **130**, on a separate integrated circuit die that is connected to the integrated circuit die having the memory cells using through-silicon vias (TSVS) and/or other connection techniques.

[0071] The signal and noise characteristics **139** can be determined based at least in part on the read control parameter. For example, when the read control parameter is a suggested read voltage for reading the memory cells at the address **135**, the calibration circuit **145** can compute the read voltages V_A, V_B, V_C, V_D, and V_E that are in the vicinity of the suggested read voltage.

[0072] The signal and noise characteristics **139** can include the bit counts C_A, C_B, C_C, C_D, and C_E. Alternatively, or in combination, the signal and noise characteristics **139** can include the count differences D_A, D_B, D_C, and D_D.

[0073] Optionally, the calibration circuit **145** uses one method to compute an optimized read voltage V_O from the count differences D_A, D_B, D_C, and D_D; and the controller **115** uses another different method to compute the optimized read voltage V_O from the signal and noise characteristics **139** and optionally other data that is not available to the calibration circuit **145**.

[0074] When the calibration circuit **145** can compute the optimized read voltage V_O from the count differences D_A, D_B, D_C, and D_D generated during the read operation, the signal and noise characteristics can optionally include the optimized read voltage V_O. Further, the memory device **130** can use the optimized read voltage V_O in determining the hard bit data in the data **137** from the memory cells at the address **135**. The soft bit data in the data **137** can be obtained by reading the memory cells with read voltages that are a predetermined offset away from the optimized read voltage V_O. Alternatively, the memory device **130** uses the controller-specified read voltage provided in the read control parameter in reading the data **137**.

[0075] The controller **115** can be configured with more processing power than the calibration circuit **145** of the integrated circuit memory device **130**. Further, the controller **115** can have other signal and noise characteristics applicable to the memory cells in the group (e.g., **133**, . . . , or

133). Thus, in general, the controller **115** can compute a more accurate estimation of the optimized read voltage V_O (e.g., for a subsequent read operation, or for a retry of the read operation).

[0076] In general, it is not necessary for the calibration circuit **145** to provide the signal and noise characteristics **139** in the form of a distribution of bit counts over a set of read voltages, or in the form of a distribution of count differences over a set of read voltages. For example, the calibration circuit **145** can provide the optimized read voltage V_O calculated by the calibration circuit **145**, as signal and noise characteristics **139**.

[0077] The calibration circuit **145** can be configured to generate the signal and noise characteristics **139** (e.g., the bit counts, or bit count differences) as a byproduct of a read operation. The generation of the signal and noise characteristics **139** can be implemented in the integrated circuit memory device **130** with little or no impact on the latency of the read operation in comparison with a typical read without the generation of the signal and noise characteristics **139**. Thus, the calibration circuit **145** can determine signal and noise characteristics **139** efficiently as a byproduct of performing a read operation according to a command from the controller **115** of the memory sub-system **110**.

[0078] In general, the calculation of the optimized read voltage V_O can be performed within the memory device **130**, or by a controller **115** of the memory sub-system **110** that receives the signal and noise characteristics **139** as part of enriched status response from the memory device **130**.

[0079] In some instances, the calibration circuit **145** of the memory sub-system **110** is configured to use the signal and noise characteristics **139**, measured during calibration of one or more lower optimized read voltages of a group of memory cells (e.g., **131** or **133**), to identify an estimated location of a higher optimized read voltage and thus improve the calibration operation performed for the higher optimized read voltage.

[0080] For example, the calibration circuit **145** can use a predictive model, trained via machine learning or established via an empirical formula, to predict the location of the higher optimized read voltage. The predication can be based on an initial estimation of the location of the higher optimized read voltage, the initial estimation(s) of the location of the one or more lower optimized read voltages, and the calibrated locations of the one or more lower optimized read voltages, where the calibrated locations of the one or more lower optimized read voltages are determined from the signal and noise characteristics **139** measured during the calibration of the one or more lower optimized read voltages. The prediction can be used in the calibration of the higher optimized read voltage, during which further signal and noise characteristics **139** can be measured in the vicinity of the predicted location to identify a calibrated location of the higher optimized read voltage. The result of the calibration of the higher optimized read voltage can be further used in the calibration of even further higher optimized read voltage iteratively.

[0081] For example, a controller **115** of the memory sub-system **110** can initially identify the expected/estimated/predicted locations of the multiple optimized read voltages for reading the states of each memory cell in a group for executing a read command. In response to the read command, the memory device **130** starts to calibrate the lowest one of the multiple optimized read voltages first, using the expected/estimated/predicted location of the lowest optimized read voltage initially identified by the controller **115**. The calibration results in the identification of an optimized location of the lowest optimized read voltage, which can have an offset or shift from the expected/estimated/predicted location of the lowest optimized read voltage. The offset or shift of the lowest optimized read voltage can be used to predict/estimate the offset or shift of the second lowest optimized read voltage, and thus improve or correct the expected/estimated/predicted location of the second lowest optimized read voltage. The improved or corrected location for the estimation of the second lowest optimized read voltage can be used in its calibration, which results in the identification of an optimized location of the second lowest optimized read voltage. Subsequently, a further higher optimized read voltage of the memory cells can be calibrated using an improved or corrected location determined from its initial estimated identified by the controller **115** and one or more offsets/shifts of one or more optimized read voltages as calibrated from their initial estimations. Thus, the higher optimized read voltages of a memory cell can be iteratively and adaptively calibrated based on the results of the lower optimized read voltages of the memory cell.

[0082] FIGS. **4**-**7** illustrate self adapting iterative read calibration during the execution of a read command according to one embodiment. For example, the self adapting iterative read calibration can be controlled by the controller **115** of the memory sub-system **110** of FIG. **1**, and/or by the calibration circuit **145** of an integrated circuit memory device **130** of FIG. **2**, using the signal and noise characteristics **139** measured according to FIG. **3**.

[0083] FIG. **4** illustrates a read threshold error distribution **157** for reading a group of memory cells (e.g., **131** or **133**) at various read voltages. The optimized read voltages V_{O1}, V_{O2}, and V_{O3} have locations corresponding to local minimum points of the read threshold error distribution **157**. When the group of memory cells (e.g., **131** or **133**) is read at the optimized read voltages V_{O1}, V_{O2}, and V_{O3} respectively, the errors in the states determined from the read operations are minimized.

[0084] FIG. **4** illustrates an example with multiple optimized read voltages V_{O1}, V_{O2}, and V_{O3} for reading a group of memory cells (e.g., **131** or **133**). In general, a group of memory cells (e.g., **131** or **133**) can be programmed to be read via more or less optimized read voltages as illustrated in FIG. **4**.

[0085] The read threshold error distribution **157** can be measured using the technique illustrated in FIG. **3** (e.g., by determining bit count differences of neighboring read voltages).

[0086] When the group of memory cells (e.g., **131** or **133**) is initially programmed, or recently calibrated, the locations of the optimized read voltages V_{O1}, V_{O2}, and V_{O3} are known. However, after a period of time, the locations of the optimized read voltages V_{O1}, V_{O2}, and V_{O3} can shift, e.g., due to quick charge loss (QCL), storage charge loss (SCL), etc.

[0087] FIGS. **5**-**7** illustrate a read threshold error distribution **161** where the locations of the optimized read voltages have shifted on the axis of read voltage. For example, the locations of the optimized read voltages V_{O1}, V_{O2}, and V_{O3} can shift downwards such that the new location has a voltage smaller than the corresponding prior location. In other examples, the locations of the optimized read voltages V_{O1},

V_{O2}, and V_{O3} can shift upwards such that the new location has a voltage larger than the corresponding prior location.

[0088] The calibration technique of FIG. **3** determines the location of an optimized read voltage (e.g., V_O) on the axis of the read voltage by sampling a portion of the read threshold error distribution **157** in the vicinity of an estimated location (e.g., V_C) and determine the location of the local minimum point of the sampled read threshold error distribution **157**.

[0089] To determine locations of the optimized read voltages that have shifted, the previously known locations of the optimized read voltages V_{O1}, V_{O2}, and V_{O3} can be used as estimated locations (e.g., V_C) for the application of the calibration technique of FIG. **3**.

[0090] FIGS. **5**-**7** illustrate the estimated locations V_{C1}, V_{C2}, and V_{C3} of the optimized read voltages V_{O1}, V_{O2}, and V_{O3} relative to the new read threshold error distribution **161**. In some instances, the controller **115** can compute the estimated locations V_{C1}, V_{C2}, and V_{C3}, based on a formula and/or a predictive model, using parameters available to the controller **115**.

[0091] FIG. **5** illustrates the application of the technique of FIG. **3** to determine the location of the lowest optimized read voltage V_{O1}. Test voltages in the range of V_{A1} to V_{E1} are configured in the vicinity of the estimated location V_{C1}. The test voltages V_{A1} to V_{E1} can be applied to read the group of memory cells (e.g., **131** or **133**) to determine bit counts at the test voltages, and the count differences that are indicative of the magnitude of read threshold errors. The optimized read voltage V_{O1} can be determined at the local minimum of the portion of the read threshold error distribution **161** sampled via the measured bit differences; and the offset or shift V_{S1} from the estimated location V_{C1} to the calibrated location V_{O1} can be used to determine the estimated shift Vu from the estimated location V_{C2} for the next, higher optimized read voltage V_{O2}.

[0092] For example, the estimated shift V_{t1} can be determined as the same as the measured shift V_{S1} in the lower optimized read voltage V_{O1} from its initial estimation V_{C1}. An alternative empirical formula or predictive model can be used to calculate the estimated shift V_{t1} of the higher optimized read voltage V_{O2} from at least the measured shift V_{S1} of the lower optimized read voltage V_{O2}.

[0093] The estimated shift V_{t1} determines the improved estimation V_{C2U} of the location of the optimized read voltage V_{O2}.

[0094] FIG. **6** illustrates the application of the technique of FIG. **3** to determine the location of the optimized read voltage V_{O2}. After adjusting the estimation from V_{C2} to V_{C2U}, test voltages in the range of V_{A2} to V_{E2} are configured in the vicinity of the improved estimation V_{C2U} (instead of relative to V_{C2}). As a result of the improved estimation V_{C2U}, the test voltage range from V_{A2} to V_{E2} is better positioned to capture the optimized read voltage V_{O2}. The test voltages V_{A2} to V_{E2} can be applied to read the group of memory cells (e.g., **131** or **133**) to determine bit counts at the test voltages, and the count differences that are indicative of the magnitude of read threshold errors. The optimized read voltage V_{O2} can be determined at the local minimum of the portion of the read threshold error distribution **161** sampled via the measuring of the bit differences; and the offset or shift V_{S2} from the initial estimated location V_{C2} to the calibrated location V_{O2} can be used in determining the estimated shift V_{t2} from the estimated location V_{C3} for the next, higher optimized read voltage V_{O3}.

[0095] For example, the estimated shift V_{t2} can be determined as the same as the measured shift V_{S2} in the lower optimized read voltage V_{O2} from its initial estimation V_{C2}. Alternatively, the estimated shift V_{t2} can be determined as a function of both the measured shift V_{S2} in the lower optimized read voltage V_{O2} from its initial estimation V_{C2} and the measured shift V_{S1} in the further lower optimized read voltage V_{O1} from its initial estimation V_{C1}. An alternative empirical formula or predictive model can be used to calculate the estimated shift V_{t2} of the higher optimized read voltage V_{O3} from at least the measured shift(s) (e.g., V_{S2} and/or V_{S1}) of one or more lower optimized read voltages (e.g., V_{O2} and/or V_{S1}).

[0096] The estimated shift V_{t2} provides the improved estimation V_{C3U} of the location of the optimized read voltage V_{O2}.

[0097] FIG. **7** illustrates the application of the technique of FIG. **3** to determine the location of the optimized read voltage V_{O3}. Test voltages in the range of V_{A3} to V_{E3} are configured in the vicinity of the improved estimation V_{C3U}. The test voltages V_{A3} to V_{E3} can be applied to read the group of memory cells (e.g., **131** or **133**) to determine bit counts at the test voltages, and the count differences that are indicative of the magnitude of read threshold errors. The optimized read voltage V_{O3} can be determined at the local minimum of the portion of the read threshold error distribution **161** sampled via the bit differences.

[0098] As illustrated in FIGS. **6** and **7**, the improved estimates V_{C2U} and V_{C3U}, calculated adaptively and iteratively, allow the calibrations of higher optimized read voltages V_{O2} and V_{O3} to be performed in improved test voltage ranges that are close to the optimized read voltages V_{O2} and V_{O3}. If the test voltages were to be constructed using the initial estimations V_{C2} and V_{C3}, the test ranges might not capture the optimized read voltages V_{O2} and V_{O3}; and calibrations might fail to identify the optimized read voltages V_{O2} and V_{O3}, or fail to identify the optimized read voltages V_{O2} and V_{O3} with sufficient accuracy.

[0099] FIG. **8** illustrates the generation of compound features for the classification of the error rate of data retrieved from memory cells according to one embodiment. For example, the compound features can be calculated according to FIG. **8** using the signal and noise characteristics of successively higher optimized read voltages, such as the calibrated/calculated read voltages optimized in a way as illustrated in FIGS. **4**-**7**.

[0100] In FIG. **8**, an ordered list **171** of estimations of optimized read voltages is identified to, or in, an integrated circuit memory device **130**. The calibration circuit **145** starts to perform calibration for the lowest read voltage that is to be calibrated using a technique of FIG. **3**.

[0101] For example, the corresponding estimations in the list **171** can be used to identify a set of test voltages. The bit counts and/or count differences of a group of memory cells (e.g., **131** or **133**) can be measured for the set of test voltages, as illustrated in FIG. **3**.

[0102] Optionally, when one or more lower optimized read voltages have been computed through calibration, the set of test voltages can be identified based on the corresponding estimations in the list **171** and the offsets of the lower optimized read voltages from their calibrated read voltages, in a way as illustrated in FIGS. **4**-**7**.

[0103] During the calibration of the lowest read voltage **173**, the read/write circuit **143** applies the test voltages to read the group of memory cell **131**. A set of signal and noise characteristics **139** is generated from the statistics (e.g., bit counts and count differences) of the states of the memory cells in the group **131** as read using the test voltages.

[0104] After the calibration of the lowest read voltage **173** to be calibrated, the calibration circuit **145** can proceed to calibrate the next **177** lowest read voltage **173** to be calibrated.

[0105] During the time period of the calibration circuit **145** calibrating the next **177** lowest read voltage **173** to be calibrated, the feature generator **113** uses the signal and noise characteristics **139** measured for the just calibrated read voltage to generate the updated compound features **175** to include the considerations of the signal and noise characteristics **139** that have been obtained so far.

[0106] For example, each set of signal and noise characteristics can include the lowest error indicator D_{MIN} **155** of the calculated read voltage V_O optimized for reading the group **133** of memory cells. D_{MIN} can be used as a feature associated with the optimized read voltage V_O. A compound feature can be the minimum (or the maximum) of D_{MIN} of the multiple calibrated/optimized read voltages V_O that correspond to the ordered list **171**.

[0107] When D_{MIN} **155** is calculated for the lowest one in the ordered list **171**, the compound feature can take the value of the D_{MIN} **155** of the lowest one in the ordered list **171**. When D_{MIN} **155** is calculated for the next lowest one in the ordered list **171**, the feature generator **113** can update the compound feature by comparing the existing value of the compound feature and the D_{MIN} **155** calculated for the next lowest one in the ordered list **171**. If the existing value of the compound feature is higher than the D_{MIN} **155** calculated for the next lowest one in the ordered list **171**, the compound feature is updated to be equal to the D_{MIN} **155** calculated for the next lowest one in the ordered list **171**; otherwise, the existing value of the compound feature is not changed in view of the D_{MIN} **155** calculated for the next lowest one in the ordered list **171**. After the updating is performed iteratively/progressively for the entire list **171**, the compound feature has the value corresponding to the minimum/smallest of D_{MIN} of the corresponding optimized read voltages V_O.

[0108] The maximum/largest of D_{MAX} of the corresponding optimized read voltages V_O **151** can be calculated in a similar way as a compound feature.

[0109] In some implementations, D_{MIN} can be estimated as the smallest one of the bit differences D_A to D_D.

[0110] In another example, the count differences D_A to D_D measured to calculate the optimized voltages V_O **151** can be evaluated to identify an indication of the maximum D_{MAX} of the sampled read threshold error (e.g., the maximum of D_A to D_D). D_{MAX} can be used as a feature associated with the optimized voltages V_O. The smallest of D_{MAX} of the optimized voltages can be used as a compound feature; and the largest of D_{MAX} of the optimized voltages can be used as another compound feature.

[0111] In a further example, the range of read threshold error sampled for the optimized voltage V_O can be determined as $R=D_{MAX}-D_{MIN}$. Such a range R can be used as a feature associated with the optimized voltage V_O. The largest of such ranges R of the optimized voltages can be used as a compound feature; and the smallest of such ranges R of the optimized voltages can be used as another compound feature.

[0112] The compound features updated for all of the optimized read voltages and other features corresponding to the signal and noise characteristics **139** can be used in the data integrity classifier **114** to generate a classification of the bit error rate of data retrievable from the group **131** of memory cells using the multiple calibrated/optimized read voltages V_O **151**.

[0113] In general, the data integrity classifier **114** and/or the feature generator **113** can be implemented in the memory device **130** and/or in the controller **115** of the memory sub-system **110**. For example, a feature generator **113** can be implemented in the memory device **130** and configured to iteratively or progressively update **175** the compound features using the most recently obtained signal and noise characteristics **139** of an optimized read voltage, before the signal and noise characteristics **139** of the next optimized read voltage become available.

[0114] Alternatively, a data integrity classifier **114** and/or a feature generator **113** can be implemented in the controller **150**. After the memory device **130** reports the calibration result of lower read voltages (e.g., V_{O1} and V_{O2}) to the controller **150**, the feature generator **113** updates **175** the compound features **175** using the signal and noise characteristics **139** included in the calibration result, while the calibration circuit **145** measures the signal and noise characteristics **139** of higher read voltages (e.g., V_{O3}).

[0115] A data integrity classifier **114** and/or a feature generator **113** implemented in the controller **115** can use not only the signal and noise characteristics **139** received from the memory device **130** for the data **137** but also other information that may not be available in the memory device **130**, such as charge loss, read disturb, cross-temperature effect, program/erase, data retention, etc. The data integrity classifier **114**/feature generator **113** implemented in the controller **115** and the data integrity classifier **114**/feature generator **113** implemented in the memory device **130** can have different complexity, and/or different levels of accuracy in their predictions. The data integrity classifier **114**/feature generator **113** implemented in the controller **115** and the data integrity classifier **114**/feature generator **113** implemented in the memory device **130** can communicate with each other to collaboratively control the calibration operations performed by the calibration circuit **145**.

[0116] The processing logic of the data integrity classifier **114**/feature generator **113** can be implemented using complementary metal-oxide-semiconductor (CMOS) circuitry formed under the array of memory cells on an integrated circuit die of the memory device **130**. For example, the processing logic can be formed, within the integrated circuit package of the memory device **130**, on a separate integrated circuit die that is connected to the integrated circuit die having the memory cells using through-silicon vias (TSVS) and/or other connection techniques.

[0117] FIG. **9** illustrates an implementation of a data integrity classifier implemented based on binary classification decision tree according to one embodiment. For example, the technique of FIG. **9** can be used to implement the data integrity classifier **114** of FIGS. **1**, **2**, and/or **8**.

[0118] For example, the data integrity classifier **114** of FIG. **9** has feature registers **181** that are configured to store the value of the features generated by the feature generator **113**. The features can include compound features generated iteratively or progressively in a way as illustrated in FIG. **8**, while the calibration circuit **145** progresses from calibrating low read voltages to high read voltages.

[0119] The data integrity classifier **114** of FIG. **9** can store a set of thresholds **182**. For example, the thresholds **182** can be stored in programmable memory and/or registers.

[0120] The data integrity classifier **114** of FIG. **9** further includes two sets **185** and **186** of term selection registers. Each of the term selection register sets (e.g., **185** or **186**) has multiple registers, identifying features or thresholds to be selected from the feature registers **181** and the stored thresholds **182** for a comparator **189**.

[0121] The selection logic **183** is controlled by the term selection register sets **185** and **186** to output two terms **187** and **188** as the input for the comparator **189**. In general, the term selection register sets **185** and **186** can be programmed to select two features as terms **187** and **188**, a feature as the term **187** and a threshold as the term **188**, or a threshold as the term **187** and a feature as the term **188**.

[0122] The comparator **189** is configured to compare the terms **187** and **188** to determine whether a pre-defined relation is satisfied between the terms **187** and **188**. For example, the comparator **189** can be configured to determine whether the term A **187** is greater than or equal to the term B **188** (or whether the term A **187** is less than or equal to the term B **188**).

[0123] The data integrity classifier **114** of FIG. **9** has a leaf path register file **193** configured to store data identifying the connectivity of nodes in the binary classification decision tree of the data integrity classifier **114**.

[0124] In general, a node in the binary classification decision tree is either a branch node or a leaf node. A leaf node identifies a classification result/decision. A branch node has an associated comparison and two child nodes. The result of the associated comparison determines which of the child nodes is to be selected in the search for a leaf node that provides a classification result/decision.

[0125] The data integrity classifier **114** of FIG. **9** has a leaf selection logic **191**. After the comparator **189** generates the comparison result of a branch node, the leaf selection logic **191** identifies a child node based on the leaf path register file **193**.

[0126] If the leaf path register file **193** indicates that the child node is a further branch node, the leaf selection logic **191** determines the registers in the term selection register sets **185** and **186** associated with the child node (i.e., the further branch node), causing the selection logic **183** to select the corresponding terms **187** and **188** of the comparison of the child node.

[0127] However, if the leaf path register file **193** indicates that the child node is a leaf node, the leaf selection logic **191** provides the classification result of the leaf node as the decision **195**.

[0128] For example, the leaf selection logic **191** can be used to initially identify the registers that stores, in the term selection register sets **185** and **186**, the identifications of inputs for a top node in the binary classification decision tree. The identifications provided by the term selection register sets **185** and **186** from the registers identified for the top node causes the selection logic **183** to output terms **187** and **188** by selecting from the feature registers **181** and the thresholds **182** according to the identifications. The comparator **189** generates a comparison result of the top node from the terms **187** and **188** received from the selection logic **183**. Based on the result of the comparator **189** and the leaf path register file **193**, the leaf selection logic **191** identifies the next node and its associated registers in the term selection register sets **185** and **185**. The operation of selecting the next node can be repeated until a leaf node is reached. In response to reaching the leaf node, the leaf selection logic **191** provides the classification result pre-associated with the leaf node as the classification decision **195**.

[0129] The structure of the data integrity classifier **114** of FIG. **9** can minimize/reduce the number of states for tree-based classification, minimize/reduce the logic circuit required to implement a tree-based classifier, and provide flexibility to configure and re-configure the decision tree.

[0130] The data integrity classifier **114** of FIG. **9** evaluates the decision tree one node at a time. Thus, the data integrity classifier **114** of FIG. **9** uses one comparator at a time.

[0131] In general, the data integrity classifier **114** of FIG. **9** can use a plurality of different types of comparisons in evaluating the decisions of the branch nodes. Since the data integrity classifier **114** uses one comparator at a time, one comparator for each type of comparisons is sufficient for the data integrity classifier **114** of FIG. **9**.

[0132] FIG. **9** illustrates an implementation of a data integrity classifier **114** using a binary classification decision tree. The technique of FIG. **9** can be extended to other types of decision trees. In general, a branch node in the decision tree can have more than two child nodes; and the selection of the child nodes can be based on more than two terms. Thus, multiple sets of term selection registers can be used to select the respective terms for the decision of a branch node; and a node decision generator can be used to generate the child selection result from the terms selected from the feature registers **181** and the thresholds **182**. The child selection result can be used in the leaf selection logic **191** to select the registers of the next branch node based on the leaf path register file **193** and the child selection result, until a leaf node is reached for the decision **195**.

[0133] In some implementations, the leaf path register file **193** further stores the term identifications of features and/or thresholds to be selected by the selection logic **183** from the feature registers **181** and the pre-defined thresholds **182**. To process a branch node, the leaf selection logic **191** provides the term identifications of the branch node and update the term selection register sets **185** and **186** to cause the selection logic **183** to output the terms **187** and **188** for the branch node.

[0134] FIG. **10** shows a method of classifying the integrity of data retrieved from memory cells using features generated according to one embodiment. The method of FIG. **10** can be performed by processing logic that can include hardware (e.g., processing device, circuitry, dedicated logic, programmable logic, microcode, hardware of a device, integrated circuit, etc.), software/firmware (e.g., instructions run or executed on a processing device), or a combination thereof. In some embodiments, the method of FIG. **10** is performed at least in part by the controller **115** of FIG. **1**, or processing logic in the memory device **130** of FIG. **2**. Although shown in a particular sequence or order, unless otherwise specified, the order of the processes can be modified. Thus, the illustrated embodiments should be understood only as examples, and the illustrated processes can be performed in a different order, and some processes can be performed in parallel. Additionally, one or more processes can be omitted in various embodiments. Thus, not all processes are required in every embodiment. Other process flows are possible.

[0135] For example, the method of FIG. **10** can be implemented in a computing system of FIG. **1** with a memory device of FIG. **2** and signal noise characteristics illustrated in FIG. **3** and generated in a way as illustrated in FIGS. **4-7**. The method of FIG. **10** can optionally use compound features calculated using the technique of FIG. **8** and implemented at least in part using the structure of a data integrity classifier **114** illustrated in FIG. **9**.

[0136] At block **301**, a set of feature registers **181** of a data integrity classifier **114** stores features generated from signal and noise characteristics **139** of a group of memory cells (e.g., **131** or **133**) in a memory device **130**.

[0137] At block **303**, a leaf path register file **193** stores data identifying node connectivity in a decision tree of the data integrity classifier **114**.

[0138] At block **305**, a selection logic **183** selects, from the feature registers, at least one feature as input (e.g., **187** and/or **188**) to a node decision logic (e.g., **189**).

[0139] At block **307**, the node decision logic (e.g., **189**) generates an output based at least in part on the at least one feature selected from the feature registers **181**. The output identifies or indicates a selected child node in the decision tree of the data integrity classifier **114**.

[0140] At block **309**, the data integrity classifier **114** determines whether the child node is a leaf node or a branch node.

[0141] If the child node is a branch node, at block **311**, a leaf selection logic **191** controls further selection from the feature registers for the node decision logic (e.g., **189**) in evaluating the child node according to the data stored in the leaf path register file **193**. Operations in blocks **305** to **311** can be repeated until reaching a child node that is a leaf node.

[0142] If the child node is a leaf node, at block **313**, the leaf selection logic **191** provide a classification pre-associated with the leaf node, in response to the node decision logic (e.g., **189**) providing an output that selects the leaf node according to the data stored in the leaf path register file **193**.

[0143] For example, the classification characterizes a bit error rate of data retrievable from the group of memory cells (e.g., **131** or **133**) using the read voltages optimized according to the signal and noise characteristics **139** of the group of memory cells (e.g., **131** or **133**). The classification can be used to control an operation to read the group of memory cells (e.g., **131** or **133**). For example, based on the classification decision **195**, the memory device **130** and/or the memory sub-system **110** can decide to further calibrate the read voltages, to skip error detection and recovery operation, to select a decoder from a plurality of decoders available in the memory device **130** and/or the memory sub-system **110** in decoding the read retrieved from the group of memory cells (e.g., **131** or **133**) using the optimized read voltages, etc.

[0144] Using the data in the leaf path register file **193**, the leaf selection logic **191** can control the selection of the features from the feature registers **181** as inputs (e.g., **187** and/or **188**) to the node decision logic (e.g., **189**).

[0145] Optionally, a set of threshold registers is provided in the data integrity classifier **114** to store pre-defined thresholds. Using the data in the leaf path register file **193**, the leaf selection logic **191** can also control the selection of the thresholds from the threshold registers (e.g., **182**) as input (e.g., **187** or **188**) to the node decision logic (e.g., **189**).

[0146] For example, a plurality of term selection register sets (e.g., **185** and **186**) can be used to store locations in the feature registers (e.g., **181**) and the threshold registers (e.g., **182**). The outputs of the plurality of term selection register sets (e.g., **185** and **186**) instructs the selection logic **183** to select, from the feature registers (e.g., **181**) and the threshold registers (e.g., **182**) respective terms (e.g., **187**, **188**) as inputs to the node decision logic (e.g., **189**). The node decision log (e.g., **189**) computes an output based a predefined function of the input terms (e.g., **187**, **188**). The output identifies or indicates a selected child node according to the connectivity specified in the leaf path register file **193**.

[0147] For example, the connectivity specified in the leaf path register file **193** can correspond to a binary classification decision tree; and the node decision logic includes a comparator **189**.

[0148] Operations in blocks **305** to **311** can be performed for branch nodes in the decision one at a time until the leaf node is reached in the decision tree.

[0149] Optionally, the data integrity classifier **113** can include a feature generator **113** that computes compound features based on measured sets signal and noise characteristics for some optimized read voltages, while the calibration circuit **145** is measuring further sets of measured sets signal and noise characteristics other optimized read voltages, as illustrated in FIG. **8**.

[0150] For example, the calibration circuit **145** can measure multiple sets of signal and noise characteristics to calculate multiple optimized read voltages respectively. The multiple sets of signal and noise characteristics can include first sets of signal and noise characteristics, and a second set of signal and noise characteristics measured after measuring the first sets of signal and noise characteristics. The feature generator **113** calculates a first compound feature from the first sets of signal and noise characteristics, at least in part in parallel with the calibration circuit **145** measuring the second set. The feature generator **113** updates the first compound feature according to the second set of signal and noise characteristics after the second set becomes available. After the multiple sets of signal and noise characteristics are all measured, the first compound feature can be updated and stored into one of the feature registers **181**.

[0151] A non-transitory computer storage medium can be used to store instructions of the firmware of a memory sub-system (e.g., **113** and/or **114**). When the instructions are executed by the controller **115** and/or the processing device **117**, the instructions cause the controller **115** and/or the processing device **117** to perform the methods discussed above.

[0152] FIG. **11** illustrates an example machine of a computer system **400** within which a set of instructions, for causing the machine to perform any one or more of the methodologies discussed herein, can be executed. In some embodiments, the computer system **400** can correspond to a host system (e.g., the host system **120** of FIG. **1**) that includes, is coupled to, or utilizes a memory sub-system (e.g., the memory sub-system **110** of FIG. **1**) or can be used to perform the operations of a data integrity classifier **114** (e.g., to execute instructions to perform operations corresponding to the data integrity classifier **114** described with reference to FIGS. **1-10**). In alternative embodiments, the machine can be connected (e.g., networked) to other machines in a LAN, an intranet, an extranet, and/or the Internet. The machine can operate in the capacity of a server or a client machine in client-server network environment, as a peer machine in a peer-to-peer (or distributed) network environment, or as a server or a client machine in a cloud computing infrastructure or environment.

[0153] The machine can be a personal computer (PC), a tablet PC, a set-top box (STB), a personal digital assistant (PDA), a cellular telephone, a web appliance, a server, a network router, a switch or bridge, or any machine capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken by that machine. Further, while a single machine is illustrated, the term "machine" shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein.

[0154] The example computer system **400** includes a processing device **402**, a main memory **404** (e.g., read-only memory (ROM), flash memory, dynamic random access memory (DRAM) such as synchronous DRAM (SDRAM) or Rambus DRAM (RDRAM), static random access memory (SRAM), etc.), and a data storage system **418**, which communicate with each other via a bus **430** (which can include multiple buses).

[0155] Processing device **402** represents one or more general-purpose processing devices such as a microprocessor, a central processing unit, or the like. More particularly, the processing device can be a complex instruction set computing (CISC) microprocessor, reduced instruction set computing (RISC) microprocessor, very long instruction word (VLIW) microprocessor, or a processor implementing other instruction sets, or processors implementing a combination of instruction sets. Processing device **402** can also be one or more special-purpose processing devices such as an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a digital signal processor (DSP), network processor, or the like. The processing device **402** is configured to execute instructions **426** for performing the operations and steps discussed herein. The computer system **400** can further include a network interface device **408** to communicate over the network **420**.

[0156] The data storage system **418** can include a machine-readable storage medium **424** (also known as a computer-readable medium) on which is stored one or more sets of instructions **426** or software embodying any one or more of the methodologies or functions described herein. The instructions **426** can also reside, completely or at least partially, within the main memory **404** and/or within the processing device **402** during execution thereof by the computer system **400**, the main memory **404** and the processing device **402** also constituting machine-readable storage media. The machine-readable storage medium **424**, data storage system **418**, and/or main memory **404** can correspond to the memory sub-system **110** of FIG. **1**.

[0157] In one embodiment, the instructions **426** include instructions to implement functionality corresponding to a data integrity classifier **114** (e.g., the data integrity classifier **114** described with reference to FIGS. **1-10**). While the machine-readable storage medium **424** is shown in an example embodiment to be a single medium, the term "machine-readable storage medium" should be taken to include a single medium or multiple media that store the one or more sets of instructions. The term "machine-readable storage medium" shall also be taken to include any medium that is capable of storing or encoding a set of instructions for execution by the machine and that cause the machine to perform any one or more of the methodologies of the present disclosure. The term "machine-readable storage medium" shall accordingly be taken to include, but not be limited to, solid-state memories, optical media, and magnetic media.

[0158] Some portions of the preceding detailed descriptions have been presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the ways used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of operations leading to a desired result. The operations are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0159] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. The present disclosure can refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage systems.

[0160] The present disclosure also relates to an apparatus for performing the operations herein. This apparatus can be specially constructed for the intended purposes, or it can include a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program can be stored in a computer readable storage medium, such as, but not limited to, any type of disk including floppy disks, optical disks, C_D-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, each coupled to a computer system bus.

[0161] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose systems can be used with programs in accordance with the teachings herein, or it can prove convenient to construct a more specialized apparatus to perform the method. The structure for a variety of these systems will appear as set forth in the description below. In addition, the present disclosure is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages can be used to implement the teachings of the disclosure as described herein.

[0162] The present disclosure can be provided as a computer program product, or software, that can include a machine-readable medium having stored thereon instructions, which can be used to program a computer system (or other electronic devices) to perform a process according to the present disclosure. A machine-readable medium includes any mechanism for storing information in a form readable by a machine (e.g., a computer). In some embodiments, a machine-readable (e.g., computer-readable) medium includes a machine (e.g., a computer) readable storage medium such as a read only memory ("ROM"), random access memory ("RAM"), magnetic disk storage media, optical storage media, flash memory components, etc.

[0163] In this description, various functions and operations are described as being performed by or caused by computer instructions to simplify description. However, those skilled in the art will recognize what is meant by such expressions is that the functions result from execution of the computer instructions by one or more controllers or processors, such as a microprocessor. Alternatively, or in combination, the functions and operations can be implemented using special purpose circuitry, with or without software instructions, such as using application-specific integrated circuit (ASIC) or field-programmable gate array (FPGA). Embodiments can be implemented using hardwired circuitry without software instructions, or in combination with software instructions. Thus, the techniques are limited neither to any specific combination of hardware circuitry and software, nor to any particular source for the instructions executed by the data processing system.

[0164] In the foregoing specification, embodiments of the disclosure have been described with reference to specific example embodiments thereof. It will be evident that various modifications can be made thereto without departing from the broader spirit and scope of embodiments of the disclosure as set forth in the following claims. The specification and drawings are, accordingly, to be regarded in an illustrative sense rather than a restrictive sense.

What is claimed is:

1. A device, comprising:

memory cells having a plurality of read voltages configured at a plurality of levels respectively; and

a circuit configured to:

apply to the memory cells, for each respective read voltage, a plurality of test voltages centered at the respective read voltage;

count, when each respective test voltage in the plurality of test voltages is applied for the respective read voltage, a number of a subset of the memory cells, wherein each memory cell in the subset has a predetermined state in response to the respective test voltage; and

determine, based on the number counted for each respective test voltage in the plurality of test voltages being applied for the respective read voltage, a classification of a bit error rate of data retrievable from the memory cells.

2. The device of claim **1**, wherein the circuit is configured to determine the classification using a decision tree.

3. The device of claim **2**, wherein the circuit is configured to count the number of the subset of the memory cells for a first read voltage among the plurality of read voltages in parallel with calculation of a portion of features applied in the decision tree.

4. The device of claim **3**, wherein the portion of features are compound features determined based on counting the number of the subset of the memory cells for more than two of the test voltages being applied for the first read voltage.

5. The device of claim **3**, wherein the portion of features are compound features determined based on counting the number of the subset of the memory cells for more than two of the plurality of read voltages.

6. The device of claim **1**, wherein the memory cells are configured as a plurality of subgroups; and the circuit is configured to apply different test voltages, among the plurality of test voltages, concurrently to the subgroups respectively in counting memory cells having the predetermined state.

7. The device of claim **1**, wherein the circuit is configured to apply to the memory cells the plurality of test voltages and count the number of the subset of the memory cells during execution of a read command configured with an address identifying a portion of the memory cells.

8. A method, comprising:

programming a plurality of memory cells of a device to have a plurality of read voltages configured at a plurality of levels respectively;

applying to the memory cells, for each respective read voltage, a plurality of test voltages centered at the respective read voltage;

determining, when each respective test voltage in the plurality of test voltages is applied for the respective read voltage, a count of first memory cells, among the plurality of memory cells, having a predetermined state in response to the respective test voltage; and

determining, based on the count determined for each respective test voltage in the plurality of test voltages being applied for the respective read voltage, a classification of a bit error rate of data retrievable from the plurality of memory cells.

9. The method of claim **8**, wherein the determining of the classification is based on a decision tree.

10. The method of claim **9**, wherein the determining of the count for a first read voltage among the plurality of read voltages is performed in parallel with calculation of a portion of features applied in the decision tree.

11. The method of claim **10**, wherein the portion of features are compound features determined based on counting the first memory cells for more than two of the test voltages being applied for the first read voltage.

12. The method of claim **10**, wherein the portion of features are compound features determined based on counting the first memory cells for more than two of the plurality of read voltages.

13. The method of claim **8**, wherein the memory cells are configured as a plurality of subgroups; and the method includes:

applying different test voltages, among the plurality of test voltages, concurrently to the subgroups respectively in counting the first memory cells.

14. The method of claim **8**, wherein the applying the plurality of test voltages and the determining of the count of the first memory cells are performed during execution of a read command configured with an address identifying a portion of the memory cells.

15. A memory sub-system, comprising:

a processing device; and

at least one memory device, the memory device having:

a plurality of memory cells configured as a plurality of subgroups; and

a circuit configured to, responsive to a command from the processing device:

- apply to the plurality of subgroups concurrently, for each respective read voltage among a plurality of read voltages configured at a plurality of levels, a plurality of test voltages centered at the respective read voltage;
- determine, when each respective test voltage in the plurality of test voltages is applied to a respective subgroup, a count of first memory cells, within the respective subgroup, having a predetermined state in response to the respective test voltage; and
- determine, based on the count of the first memory cells counted within the respective subgroup, a classification of a bit error rate of data retrievable from the plurality of memory cells.

16. The memory sub-system of claim **15**, wherein the command is a read command configured with an address identifying a portion of the memory cells.

17. The memory sub-system of claim **16**, wherein the circuit is configured to determine the classification using a decision tree.

18. The memory sub-system of claim **17**, wherein the circuit is configured to determine the count of the first memory cells for a first read voltage among the plurality of read voltages in parallel with calculation of a portion of features applied in the decision tree.

19. The memory sub-system of claim **18**, wherein the portion of features are compound features determined based on determining the count of the first memory cells for more than two of the test voltages being applied for the first read voltage.

20. The memory sub-system of claim **18**, wherein the portion of features are compound features determined based on determining the count of the first memory cells for more than two of the plurality of read voltages.

* * * * *