



- (51) **International Patent Classification:**
GI1C 29/52 (2006.01) *GI1C 11/22* (2006.01)
- (21) **International Application Number:**
PCT/US2016/041237
- (22) **International Filing Date:**
7 July 2016 (07.07.2016)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
62/193,468 16 July 2015 (16.07.2015) US
15/174,324 6 June 2016 (06.06.2016) US
- (71) **Applicant:** TEXAS INSTRUMENTS INCORPORATED [US/US]; P. O. Box 655474, Mail Station 3999, Dallas, TX 75265-5474 (US).
- (71) **Applicant (for JP only):** TEXAS INSTRUMENTS JAPAN LIMITED [JP/JP]; 24-1, Nishi-shinjuku 6-chome, Shinjuku-ku, Tokyo, 160-8366 (JP).
- (72) **Inventors:** QIDWAI, Saim, Ahmad; 1489 Abby Way, Allen, TX 75013 (US). CHUNG, Peter, Wongeun; 15080 Camden Lane, Frisco, TX 75035 (US).
- (74) **Agents:** DAVIS, Michael, A., Jr. et al.; Texas Instruments Incorporated, P. O. Box 655474, Mail Station 3999, Dallas, TX 75265-5474 (US).

(81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

[Continued on next page]

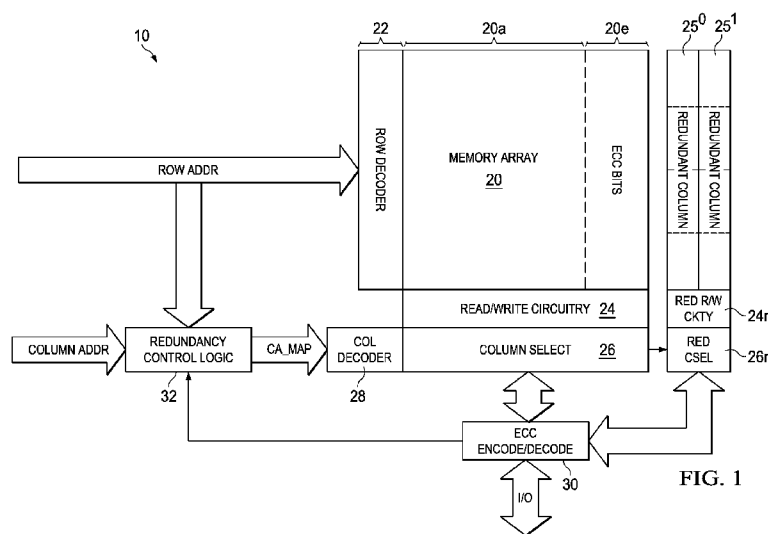
(54) **Title:** DYNAMIC ENABLING OF REDUNDANT MEMORY CELLS DURING OPERATING LIFE

FIG. 1

(57) **Abstract:** In described examples, memory (10), such as a non-volatile ferroelectric memory, has error correction coding (ECC) capability (30) and redundant memory cells (25₀, 25₁). During the system operating life of the memory, upon ECC decoding determining that a symbol read from the memory array (20) at an address cannot be corrected, the failed memory cells are identified, and redundancy enabled to replace those failed cells if available. Redundant columns may be partitioned by row address, to allow the same column of redundant cells to replace bits in different columns for different portions of the memory. The dynamic redundancy extends reliability of the memory during its system operating life.

WO 2017/011255 A1



Published:

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

DYNAMIC ENABLING OF REDUNDANT MEMORY CELLS DURING OPERATING LIFE

[0001] This relates generally to solid-state semiconductor memories, and more particularly to reliability of such memories.

BACKGROUND

[0002] Solid-state semiconductor memory is ubiquitous in electronic systems of many types, from large-scale computing systems and mass data storage (i.e., disk replacement) applications to small-scale controllers and sensors. The low cost-per-bit achievable for solid-state memory has enabled modern electronic systems to implement and execute complex computational and control functions in a wide range of applications.

[0003] Various types of semiconductor memory exist. Dynamic random access memory (DRAM) and static RAM (SRAM) memories provide high density storage and rapid access, but are volatile because their contents are lost at power-down. Accordingly, DRAM and SRAM are usually employed as data memory. Non-volatile memory technologies include electrically erasable programmable read-only memory (EEPROM) (implemented as either “flash” or non-flash erase memory), mask-programmable read-only memory (ROM), one-time-programmable (OTP) ROM, and ferroelectric memory (FRAM). Non-volatile memory is often used as program memory and mass data storage (e.g., solid-state disk) in modern electronic systems.

[0004] The increased memory density for each of these technologies has enabled the implementation of substantial memory capacity within larger scale integrated circuits, such as those including modern complex microprocessors, microcontrollers, digital signal processors and other large-scale logic circuitry. For example, the so-called “system on a chip” (SoC) usually includes embedded memory of various technologies as program and data memory. These memory resources can occupy a relatively large fraction of the chip area of a microcontroller or SoC device. This leverage makes yield loss due to defects in the embedded memories especially costly. The desire to increase memory density by shrinking the semiconductor device features increases yield pressure, particularly for memory arrays.

[0005] Accordingly, many semiconductor memories, whether embedded in a large-scale

integrated circuit (such as an SoC) or in the form of stand-alone memory devices, include redundant memory cells that can be enabled at manufacturing test to logically replace failed memory cells in the main array. These redundant cells are often arranged as one or more additional columns adjacent to the main memory array, along with the necessary read/write circuitry for those columns. Another approach provides one or more redundant rows of memory cells; in this case, additional sense amplifiers and other read/write circuitry may not be required. In either case, the memory is configured to enable the redundant cells in the event one or more failed cells are identified at manufacturing test. The redundant cells are typically enabled by blowing fuses or programming non-volatile configuration registers in the memory.

[0006] Semiconductor memories can also be vulnerable to cell failures over the operating life of the integrated circuit. In some cases, such as in DRAM memories, the failures may be “soft” failures in which a cell may sporadically return errored data when read, but then function properly for the remainder of its life. “Hard” failures (in which a cell permanently loses the ability to retain data) are typically due to physical mechanisms occurring over time, often at a site of a physical defect or contaminant at the particular cell.

[0007] Accordingly, some memories include error correction capability to correct such errors that occur over the operating life of the device. Error detection and correction is typically provided by storing each data word as a coded “symbol”, where the number of bits in the symbol is greater than the number of bits in the data word itself, such as with the additional bits stored in additional columns in the array. Depending on the particular error detection and correction technique, decoding of the over-specified symbol can provide the correct data word, even if data bits of the symbol are erroneous. The level of error detection and correction can range from a simple parity check, to a check sum provided for each data word, up to more complicated non-systematic codes.

[0008] For the case of FRAM devices, long-term reliability testing has shown that FRAM cells are somewhat vulnerable to depolarization over time. This depolarization is reflected in data retention failures over the device life. In comparison to one-transistor one-capacitor (1T-1C) FRAM cells, two-transistor two-capacitor (2T-2C) FRAM cells are less susceptible to this type of data retention failure, because the 2T-2C cell stores data as complementary polarization states on its two capacitors. The complementary polarization states of the 2T-2C cell are reflected as a differential signal when sensed, providing a larger read margin than the 1T-1C cell, which is read

by comparing the polarization state of the ferroelectric capacitor against a reference level. The 1T-1C FRAM memory has nearly twice the bit density of 2T-2C FRAM, but its reduced read margin is reflected in weaker data retention performance, which limits the use of 1T-1C FRAM in many data-critical applications.

SUMMARY

[0009] In described examples, a semiconductor memory such as a ferroelectric random access memory (FRAM) includes both redundant memory cells and error detection/correction capability. The redundant memory cells are not required to be enabled at the time of manufacture, but are available for enabling during the operating life of the device. Upon the error detection/correction logic detecting a non-correctable error in a retrieved symbol during system operation of the memory, a failed cell corresponding to an errored bit in the symbol is determined, such as by an additional read/write operation, and one or more redundant memory cells are enabled to logically replace the failed cell.

[0010] Example embodiments provide a semiconductor memory and method of operating the same that: provide improved tolerance to the number of bit failures occurring over the operating life of the device; can be implemented with relatively little overhead circuitry; and can enable the use of 1T-1C FRAM memory in data retention-sensitive applications.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 is an electrical diagram, in block form, of a semiconductor memory constructed according to an embodiment.

[0012] FIG. 2 is a block diagram of the memory array of the memory of FIG. 1, illustrating an example of the replacement of cells in the main array with redundant cells, according to an embodiment.

[0013] FIG. 3 is a flow diagram of operation of the memory of FIG. 1 according to an embodiment.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

[0014] This specification describes one or more embodiments as implemented into a semiconductor memory, such as a ferroelectric random access memory (FRAM), in which those embodiments are particularly advantageous. However, concepts of example embodiments are beneficially applicable to other implementations too, such as memories of any of a number of technologies, whether implemented as a stand-alone memory device or as an embedded memory

resource in a larger scale integrated circuit.

[0015] FIG. 1 illustrates the architecture of memory 10 constructed according to an embodiment. As mentioned hereinabove, the particular arrangements and implementations of memory 10 may vary widely, such as one or more memory arrays within a larger scale integrated circuit (e.g., a microprocessor or microcontroller), or realized as a stand-alone memory integrated circuit. Also, memory 10 according to this embodiment may be constructed according to any one of a number of memory technologies, including volatile memory technologies such as dynamic random access memory (DRAM) or static random access memory (SRAM), and non-volatile memory technologies such as electrically erasable programmable read-only memory (EEPROM) (implemented as either “flash” or non-flash erase memory), mask-programmable read-only memory (ROM), one-time-programmable (OTP) ROM, and ferroelectric memory (FRAM). As described hereinabove, this embodiment is especially beneficial when applied to FRAM, particularly by facilitating the use of 1T-1C FRAM cells in reliability-sensitive applications.

[0016] In the architecture of FIG. 1, memory 10 includes memory cells arranged in rows and columns within memory cell array 20. A single instance of memory cell array 20 is shown in FIG. 1, but memory 10 may include multiple memory cell arrays 20, each corresponding to a memory block within the address space of memory 10.

[0017] In the example of FIG. 1, array 20 includes rows and columns of memory cells, with the cells in the same column sharing one or a pair of bit lines, depending on the bit cell construction, and with the cells in the same row sharing one of word lines used to couple the cells in a selected row to those bit lines. Row decoder 22 receives a row address value indicating a row of cells in array 20 to be accessed, and energizes a word line corresponding to that row address value. Read/write circuits 24 are constructed in a conventional manner, such as to include a number of sense amplifiers and write circuits that are each coupled to a bit line for a column. Column decoder 28 controls column select circuit 26 to select a subset of read/write circuits 24, according to a column address value (described in further detail hereinbelow), thus selecting the particular cells within the selected row that contain a symbol stored at a memory address formed by its row address and column address portions.

[0018] Alternatively, memory 10 may be constructed according to an “interleaved” architecture, in which fewer read/write circuits 24 than columns are provided. In this case,

column select circuit 26 selects which bit lines are to be coupled to read/write circuits 24, according to the column address. Various row and column architectures, whether non-interleaved (as described hereinabove) or interleaved, or having some other arrangement, are useful in connection with these embodiments.

[0019] According to this embodiment, array 20 includes a sufficient number of memory cells that each memory address retrieves a number of bits greater than the number of bits in the desired data word, with the additional bits used to perform error detection or error correction. For purposes of this description, the term “error correction coding”, or “ECC”, refers to either or both of error detection and error correction techniques. Error correction techniques necessarily include error detection.

[0020] Error correction coding (ECC) encodes a given data word of m bits into a symbol of n bits ($n > m$) representing that data word. The additional $n - m$ bits provide the ability to detect whether an error exists in any bit of the symbol, and for many codes, provide the ability to correct one or more errored bits in the symbol. Various ECC codes exist, such as parity bits, checksums, cyclic redundancy checks, convolutional codes, and block codes (e.g., Hamming codes, Reed-Solomon codes). For semiconductor memories, ECC codes are usually systematic codes, so the data word (i.e., payload) is embedded in the symbol in combination with separate check bits (ECC bits). Non-systematic codes, in which the data word is not embedded within the symbol but can only be discovered after decoding the entire symbol, may alternatively be used. Some ECC codes are only error-detecting, in that they are only able to detect errors but are unable to correct the errors. Other ECC codes are error-correcting, permitting the correction of one or more errored bits in a retrieved symbol. The number of errored bits that are correctable by an ECC code is determined by the particular error-correcting code along with the number $n - m$ of code bits in each symbol beyond the m bits of the encoded data word.

[0021] The architecture shown in FIG. 1 corresponds to a systematic code, in which main array portion 20a stores the data words, and ECC portion 20e corresponds to a number of columns of memory cells that store the ECC bits for those data words. Accordingly, retrieval of a row of cells in memory array 20 retrieves one or more data words in that row of main array portion 20a, along with the corresponding ECC bits from portion 20e in that same row for those data words. These ECC bits from portion 20e are sensed by read/write circuitry 24 and, if selected by column select 26, are forwarded along the corresponding data word as a symbol to ECC encoder/decoder

30.

[0022] ECC encoder/decoder 30 may be constructed as conventional logic (e.g., programmable logic, semi-custom or custom logic circuitry) capable of encoding data words into symbols and of decoding symbols into data words, according to the applicable ECC code. ECC encoder/decoder 30 may be integrally implemented within the architecture of memory 10, as suggested by FIG. 1, or alternatively may be realized and carried out by logic circuitry elsewhere in the integrated circuit, including by a processor or logic circuitry in a larger scale integrated circuit.

[0023] In the read context, ECC encoder/decoder 30 decodes the symbol to determine whether any errored bits exist. If not, the data word corresponding to the retrieved symbol is forwarded on input/output lines I/O to the desired destination. If an error-correcting code is implemented, and if the number of errored bits does not exceed the correctable number for the code, ECC encoder/decoder 30 corrects the errored bits in the symbol and forwards the data word on input/output lines I/O. If an error is detected by ECC encoder/decoder 30 applying an error-detecting code, or if the number of errored bits exceeds the correctable number for an error-correcting code, ECC encoder/decoder 30 issues a fault, sets a flag, or issues another appropriate notification.

[0024] Conversely, ECC encoder/decoder 30 receives input data in the form of data words from input/output lines I/O and encodes those input data words into symbols that are written to main array portion 20a and ECC portion 20e.

[0025] According to this embodiment, memory 10 also includes redundant memory cells that may be selectively enabled to replace memory cells in array 20 in the address space of memory 10. In the architecture of FIG. 1, these redundant memory cells are arranged in two redundant columns 25₀, 25₁. Alternatively, only a single redundant column 25₀ or more than two such columns may be deployed. Redundant columns 25₀, 25₁ are associated with redundant read/write circuits 24r and redundant column select 26r, and as such are coupled to ECC encoder/decoder 30 in similar manner as read/write circuitry 24 and column select 26. Redundant column select 26r receives signals from column decoder 28 indicating selection of either, both, or none of redundant columns 25₀, 25₁ for forwarding to ECC encoder/decoder 30.

[0026] Alternatively, redundant memory cells may be provided in memory 10 in the form of one or more redundant rows of memory cells. In this alternative arrangement, row decoder 22

would be extended to drive word lines for the redundant rows, when replacement of a row of memory cells is enabled. Row redundancy in this manner eliminates the need for additional read/write circuitry and column select circuitry, as in the case of the column redundancy shown in FIG. 1. Some column redundancy and row redundancy techniques are known.

[0027] In conventional memories, whether embedded memories in a large-scale integrated circuit (such as an SoC) or stand-alone memory devices, usually only either error correction coding or redundancy is implemented. More specifically, redundancy techniques are conventionally enabled at manufacturing test to replace time-zero defective bits, rows, or columns, as the case may be, before assembly of the integrated circuit into its end system application.

[0028] However, according to these embodiments, redundancy in memory 10 is implemented in combination with ECC circuitry (and corresponding additional memory cells), with ECC and redundancy operating cooperative during the life of memory 10 in its system application. The dynamic redundancy provided by the combination of ECC and redundant cells extends the error correction capability of memory 10. This extended reliability is especially useful in connection with memory technologies, such as FRAM cells and other non-volatile memories, that are vulnerable to data retention degradation over operating life.

[0029] As shown in FIG. 1, ECC encoder/decoder 30 is coupled to redundancy control logic 32 in this embodiment. In its general operation, ECC encoder/decoder 30 is capable of indicating to redundancy control logic 32 the results of its error correction decoding, specifically indicating whether one or more errored bits exist in a symbol read from array 20 for a particular memory address. As described in further detail hereinbelow, this signal may indicate the occurrence of an errored bit even if it is correctable, or may be arranged to only indicate the occurrence of an uncorrectable error in that symbol.

[0030] Redundancy control logic 32 may be realized as logic circuitry, such as programmable logic, or custom or semi-custom logic, for mapping memory addresses to redundant memory cells, such as performed in conventional memories with redundancy enabled at manufacturing test (i.e., before assembly of the integrated circuit into its system application). However, according to this embodiment, redundancy control logic 32 includes some of the decision-making logic for determining whether to enable redundant memory cells, specifically in response to signals from ECC encoder/decoder 30 as described hereinabove; in conventional

memories, this decision-making logic is typically embodied in the automated test equipment carrying out the memory test. The operation of this logic is described in further detail hereinbelow. These functions are performed in the appropriate manner for particular implementations by appropriate program instructions for, or logic circuitry in, redundancy control logic 32.

[0031] Also, in this embodiment, as in conventional redundancy-capable memories, redundancy control logic 32 includes some non-volatile memory capability for storing address information pertaining to which memory addresses that originally refer to cells in main array 20a are to instead refer to redundant cells in redundant columns 25₀, 25₁. Where memory array 20 is an arrangement of FRAM cells, the non-volatile storage in redundancy control logic 32 can also be realized by FRAM. For example, the FRAM cells in memory array 20 may be of the 1T-1C type, while those in redundancy control logic 32 are 2T-2C FRAM cells. Alternatively, other types of non-volatile storage may be used.

[0032] As described hereinabove, the redundant cells in this architecture are arranged as columns 25. Redundancy control logic 32 thus receives the column address portion of the memory address and, if a redundant cell is to replace a cell in memory array 20, maps that column address to the physical address of the appropriate redundant column 25₀, 25₁, and applies that mapped physical address to column decoder 28 on lines CA_MAP. Column decoder 28 issues the appropriate control signals so that column select 26 and redundant column select 26r select and forward the columns of memory array 20 and redundant columns 25 to which the mapped address refers. Conversely, if the incoming column address is not mapped to a redundant column 25₀, 25₁, redundancy control logic 32 forwards the received column address to column decoder 28 on lines CA_MAP, in response to which column decoder 28 issues the appropriate control signals to column select 26 to select the columns of memory array 20 to which the received column address refers.

[0033] In the architecture of FIG. 1, redundant columns 25₀, 25₁ are each partitioned into four partitions to allow for additional flexibility in the replacement of cells in memory array 20. This partitioning, while optional, allows a single redundant column 25 to replace cells in four different columns of array 20, so long as the failed cells are in different row address segments (quarter-array segments, in this example). Accordingly, a portion of the row address is applied to redundant control logic 32, as shown in FIG. 1, with that row address portion indicating the

row address segment to which the current memory address is directed. For this four-partition example, this row address portion may be the two most significant bits of the row address. This portion of the row address is included in the mapping performed by redundant control logic 32 to determine whether one of the redundant columns 25₀, 25₁ is to be selected for the memory address being accessed.

[0034] FIG. 2 illustrates an example of this partitioned redundancy, for the example of redundant column 25₁ replacing cells in four different columns of memory array 20. In this example, the top partition 25₁[0] of redundant column 25₁ is mapped, in redundant control logic 32, to replace a portion 20[a] of a column a in main array portion 20a. Specifically, this portion 20[a] resides in the uppermost row address segment of array 20, corresponding to row addresses 00xx...xx (assuming correspondence between the logical address and the physical rows of array 20). Partition 25₁[1] of redundant column 25₁ replaces a portion 20[c] of a different column c in main array portion 20a, for the row address segment corresponding to row addresses 01xx...xx. Partition 25₁[2] of redundant column 25₁ similarly replaces a portion 20[b] of a column b in main array portion 20a for the row address segment corresponding to row addresses 10xx...xx. According to this embodiment, the column partitions replaced by partitions of redundant columns 25₀, 25₁ may be in either main array 20a or in ECC portion 20e; this is evident in the example of FIG. 2 by redundant partition 25₁[2] replacing a portion 20[d] of a column d in ECC portion 20e for the row address segment corresponding to row addresses 11xx...xx.

[0035] As described hereinabove, the redundancy techniques of these embodiments are directed to later life failures. Because these failures tend to be single bit failures in modern non-volatile memories (e.g., data retention failures), instead of full row or full column failures, the partitioning of redundant columns 25₀, 25₁ is especially beneficial, allowing redundant replacement of a larger number of cell failures and thus significantly improved reliability for the device.

[0036] This partitioning of redundant columns 25 (and, alternatively, similar partitioning of redundant rows if implemented) is an option in the memory architecture according to these embodiments. Also, if such partitioning is implemented, the number of partitions in each redundant column or row may vary from the four partitions shown in FIG. 2, with either fewer or more partitions implemented. In one example, the redundant columns for a memory array of 512 rows may be partitioned into sixteen partitions of thirty-two rows each. The partitioning

technique in any particular implementation can be selected according to the reliability performance of the particular memory technology involved, considered in combination with the overhead circuitry required to implement the desired partitioning. The redundant memory cells are arranged to optimize for their particular implementations.

[0037] FIG. 3 is a flow diagram of operation of memory 10 according to an embodiment. As mentioned hereinabove, redundancy control logic 32 is constructed or programmed to carry out some or all of the functions described in connection with this embodiment, in combination with other peripheral circuitry in and external to memory 10 that are involved with its access. According to these embodiments, this operation of memory 10 occurs during its normal system use, instead of as part of a manufacturing or other test sequence. Accordingly, before the operations of FIG. 3, memory 10 will have been fully tested in its manufacture, and the integrated circuit in which memory 10 is realized will have been installed or otherwise assembled into its system application and will have been in operation for some time before performance of these operations.

[0038] In process 40, memory 10 is accessed by an external function, such as a microprocessor or processor core, to write data into memory array 20 at certain memory addresses. As part of this process 40, ECC encoder/decoder 30 encodes this input data according to the ECC code being used, and the encoded symbol is written into array 20 at that address. For the example described hereinabove in which a systematic ECC code is used, the data word of the symbol will be written into main array portion 20a, and the corresponding ECC bits will be written into ECC portion 20e. The writing of input data into array 20 in this process 40 will usually write data for many, if not all, addresses in the address space of memory 10, either as a burst or block or in a distributed manner interleaved with other memory operations. The particular manner in which data are written into memory array 20 is not of particular importance according to this embodiment, so long as encoded data are stored for some of the memory addresses of array 20.

[0039] In process 42, a read of memory array 20 at a particular memory address is performed. Because the data are stored in the form of ECC symbols, process 42 involves the decoding of the symbol retrieved from the cells of array 20 corresponding to the memory address, to read the data word portion of the symbol from main array portion 20a and the ECC bits from ECC portion 20e, in this systematic code example. Upon the decoding operation performed by ECC encoder/decoder 30 determining either that no error exists in the retrieved symbol, or that the

number of errored bits are no greater than the correctable number for the code being used (decision 43 returns a “no” result), the read of process 42 is successful, and the data word decoded from the retrieved symbol (with any errored bits corrected in the ECC decoding) is forwarded on input/output lines I/O to the desired destination. The operation of memory 10 then continues in the normal manner, with writes (process 40) and reads (process 42) performed as desired under the control of the processor core or other accessing function in the system.

[0040] However, according to this embodiment, if the ECC decoding performed in process 42 in a read of a memory address determines that an uncorrectable error exists in the symbol as retrieved from array 20 for that address, more specifically because the decoding detects a number of errored bits greater than the correctable number for that code (decision 43 returns a “yes” result), memory 10 issues an interrupt to the memory access in process 44. In this example, the usual operation for the system involved in handling such an operation will be performed, such as by the system storing its current context. According to this embodiment, an interrupt handler is then executed to enable redundant memory cells to replace the array cells corresponding to those errored bits.

[0041] The interrupt handler begins with process 46 in which the errored bits at the failing memory address are identified. If the decoding operation is able to determine which bit or bits in the symbol is errored, process 46 simply identifies those bits in the symbol. However, when the number of errored bits exceeds the correctable number of bits, the decoding operation itself usually cannot identify which particular bits of the symbol are errored, or else it could readily correct those bits. Rather, the decoding operation only knows that a data word cannot be resolved from the retrieved symbol. In this situation, additional operations are performed in process 46 to identify the failed cells in array 20 corresponding to the errored bits in the symbol.

[0042] According to one embodiment, process 46 is performed by writing and reading both data states to and from all of the cells at the failing memory address. A simple approach for this is to write all of the cells at this address to a “0” data state, and then read those cells to identify any not returning a “0” data state; similarly, all cells at this address are written to a “1” data state and then read back to identify any not returning the “1”. Alternatively, other patterns (such as alternating “0” and “1” states) are useful, so long as each cell is tested for each data state. The ECC encoding applied by ECC encoder/decoder 30 may be temporarily disabled during this process 46, or the data pattern written and read may be adjusted to ensure that the main array bits

and also the ECC bits are tested at both data states. Those cells corresponding to the failing memory address that fail to return both data states are thus identified.

[0043] ECC encoder/decoder 30 issues the appropriate signals resulting from process 46 to redundancy control logic 32, such as with some identification of the identified failed cells for the failing memory address. Redundancy control logic 32 determines, in decision 47, whether redundant memory cells are available for one or more these failed cells. In the architecture of FIG. 1, this determination depends on the number of failed cells identified, and whether the appropriate partitions of redundant columns 25₀, 25₁ are already being used for a different memory address. Depending on the strength of the ECC code, not all failed cells are required to be replaced for the number of errored symbol bits to be correctable by ECC decoding. For example, if the number of errored bits exceeds the correctable number by one, then only one cell needs to be replaced for the data word to be correctly recovered. If the availability of redundant memory cells depends on the bit position within array 20, redundancy control logic 32 may include sufficient logic functionality to select how many and which cells to replace.

[0044] If redundant memory cells are available to replace one or more failed cells at the failing memory address (decision 47 is “yes”), redundancy control logic 32 enables those redundant cells accordingly in process 48. This process 48 will usually involve the storing of the failing memory address in non-volatile memory within or accessible to redundancy control logic 32, so that the received logical memory address will be correctly mapped to refer to a redundant cell in one of redundant columns 25₀, 25₁ in place of the failed cell, in this embodiment. The appropriate return from the interrupt handler is executed in process 50, such as to restore context (including restoring of the contents of memory 20 at the failing address, if possible), and memory 10 returns to its normal system operation.

[0045] Or, if redundancy is not available for the failed cells (decision 47 is “no”), such as when the redundant columns 25 have previously been assigned to other memory addresses, then the read of memory array 20 at the failing memory address is considered to have failed. The system in which memory 10 is implemented then issues and processes the appropriate fault for a memory failure.

[0046] Various alternatives to this operation of memory 10 according to this embodiment are possible. For example, decision 43 may invoke interrupt 44 to enable redundancy on the first errored bit detected in the decoding of the symbol read in process 42, even if that error can be

corrected by ECC. Other alternative decision criteria may alternatively be used, such as replacement upon reading a symbol having exactly the correctable number of errored bits, to avoid the event of a failed data read.

[0047] These embodiments provide important benefits for solid-state memory devices and embedded memories, particularly for non-volatile memories such as FRAM. The combination of ECC and redundancy in the same memory device according to these embodiments provides extended reliability to the memory during its system life, by enabling replacement of failed memory cells if error correction is no longer adequate to correct for errored bits in the data word. This ability is especially beneficial when applied to non-volatile memories (such as FRAM), considering the tendency of ferroelectric material to depolarize over system life. For example, the extended reliability provided by these embodiments may enable the higher bit density of 1T-1C FRAM cells despite the inherent lesser read margin as compared with that of 2T-2C cells.

[0048] Modifications are possible in the described embodiments, and other embodiments are possible, within the scope of the claims.

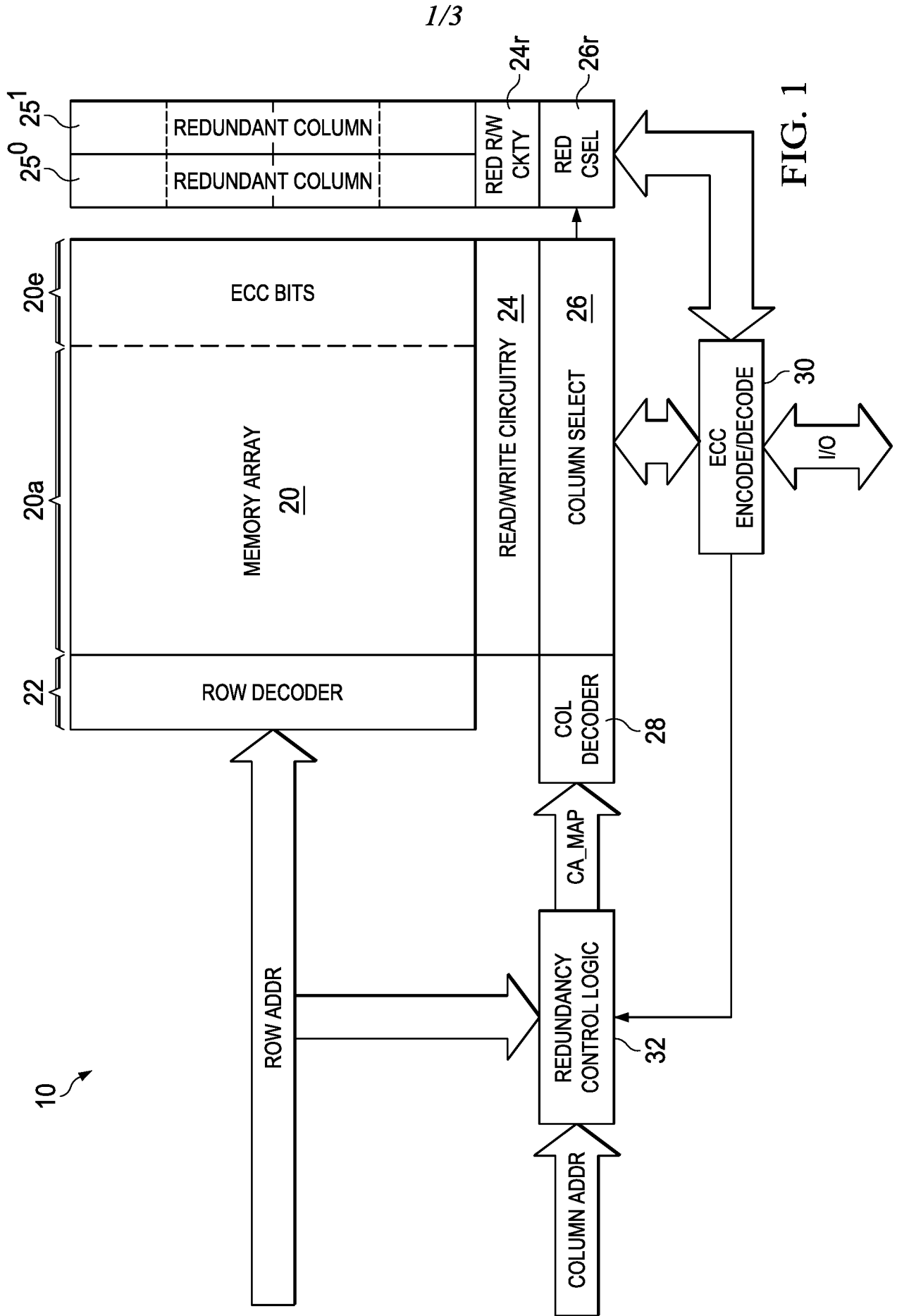
CLAIMS

What is claimed is:

1. A method of operating a system including a memory, the method comprising:
reading a symbol representing a data word from the memory at a memory address;
decoding the symbol to detect whether an error exists in the data word;
responsive to detecting an error in the data word: identifying a failed cell in the memory at the memory address; and mapping the memory address to refer to a redundant memory cell in place of the failed cell; and
during the operating of the system, reading a symbol from the memory at the memory address by retrieving at least one bit of the symbol from the redundant memory cell.
2. The method of claim 1, wherein the identifying includes:
writing a first data state to each cell in the memory at the memory address;
then reading each cell at the memory address;
writing a second data state to each cell in the memory at the memory address; and
then reading each cell at the memory address;
wherein the failed cell is identified by one of the reading steps detecting an incorrect data state from one of the cells at the memory address.
3. The method of claim 1, further comprising:
responsive to detecting a number of errors in the symbol not exceeding a correctable number, correcting the errors in the symbol and forwarding the data word represented by the symbol.
4. The method of claim 3, wherein the mapping is performed responsive to the number of errors in the symbol exceeding the correctable number.
5. The method of claim 3, wherein the mapping is performed responsive to the number of errors in the symbol being one.
6. The method of claim 1, wherein the symbol includes a plurality of data bits and one or more error correction code bits.
7. The method of claim 6, wherein, responsive to the failed cell being one of the error correction code bits, the mapping maps the redundant memory cell to the failed error correction code bit.

8. The method of claim 1, wherein the memory includes a plurality of memory cells arranged in rows and columns, and wherein the mapping includes: mapping the memory address to refer to at least a portion of a row of redundant memory cells in place of the failed cell.
9. The method of claim 1, wherein the memory includes a plurality of memory cells arranged in rows and columns, and wherein the mapping includes: mapping the memory address to refer to at least a portion of a column of redundant memory cells in place of the failed cell.
10. The method of claim 9, wherein the mapping includes: mapping part of a row address portion of the memory address and the column address portion of the memory address to refer to a portion of the column of redundant memory cells.
11. A memory, comprising:
 - a memory array having a plurality of memory cells arranged in rows and columns;
 - address circuitry for selecting a plurality of memory cells responsive to a memory address, the selected memory cells storing a symbol corresponding to a data word coded according to an error detection code;
 - read/write circuitry, for reading data from and writing data to the selected memory cells;
 - error detection logic, for determining whether an error exists in a symbol read from the selected memory cells;
 - a plurality of redundant memory cells; and
 - redundancy logic for mapping the memory address to refer to a redundant memory cell in place of a failed cell corresponding to the memory address, responsive to the error detection logic detecting an error in the symbol retrieved from the selected memory cells.
12. The memory of claim 11, wherein the error detection logic includes:
 - error correction logic, for correcting one or more errored bits in the symbol retrieved from the selected memory cells.
13. The memory of claim 12, wherein the error correction logic is capable of corrector up to a correctable number of errored bits in a symbol, and wherein the redundancy logic maps the memory address to refer to the redundant memory cell responsive to the error detection logic detecting a number of errored bits in the symbol exceeding the correctable number.
14. The memory of claim 12, wherein the redundancy logic maps the memory address to refer to the redundant memory cell responsive to the error detection logic detecting a single errored bit in the symbol.

15. The memory of claim 11, wherein the plurality of redundant memory cells are arranged in one or more redundant rows;
and wherein the redundancy logic is for mapping a row address portion of the memory address to refer to at least a portion of a redundant row.
16. The memory of claim 11, wherein the plurality of redundant memory cells are arranged in one or more redundant columns;
and wherein the redundancy logic is for mapping a column address portion of the memory address to refer to at least a portion of a redundant column.
17. The memory of claim 16, wherein the one or more redundant columns are each subdivided into redundant column portions, and wherein the redundancy logic is for mapping a portion of a row address portion of the memory address and a column address portion of the memory address to refer to a redundant column portion.
18. The memory of claim 11, wherein the plurality of memory cells are non-volatile memory cells.
19. The memory of claim 18, wherein the non-volatile memory cells are ferroelectric memory cells.
20. The memory of claim 19, wherein the ferroelectric memory cells are of the one transistor - one capacitor type.



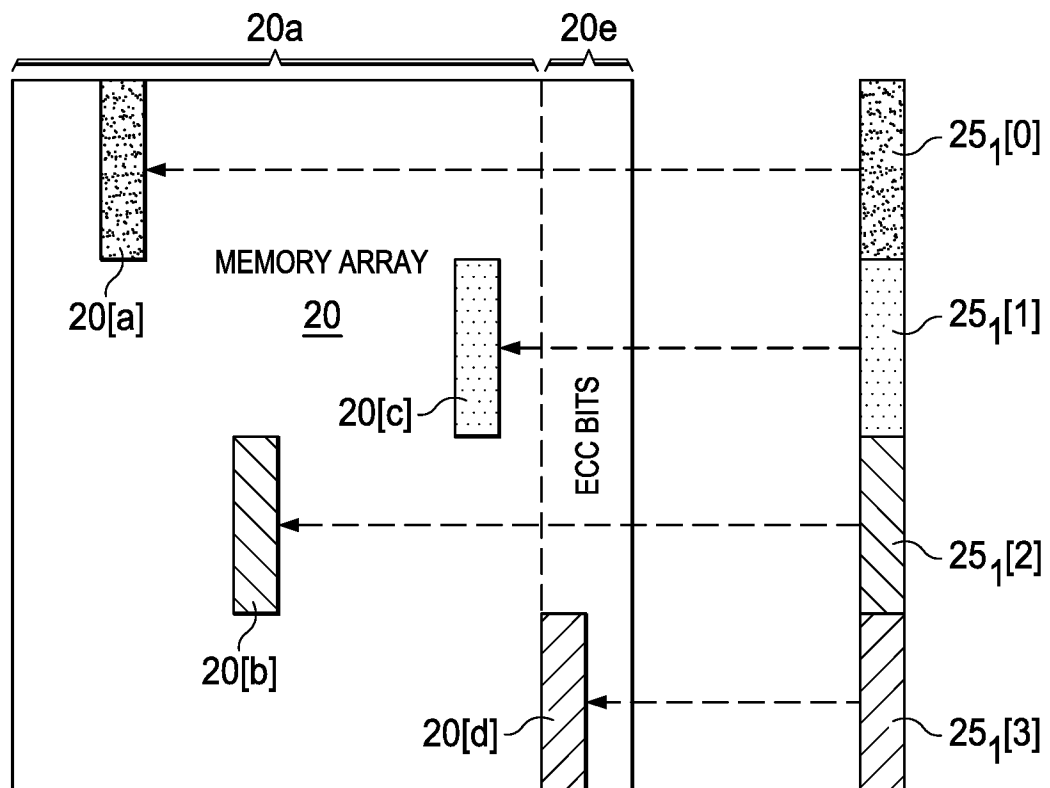


FIG. 2

3/3

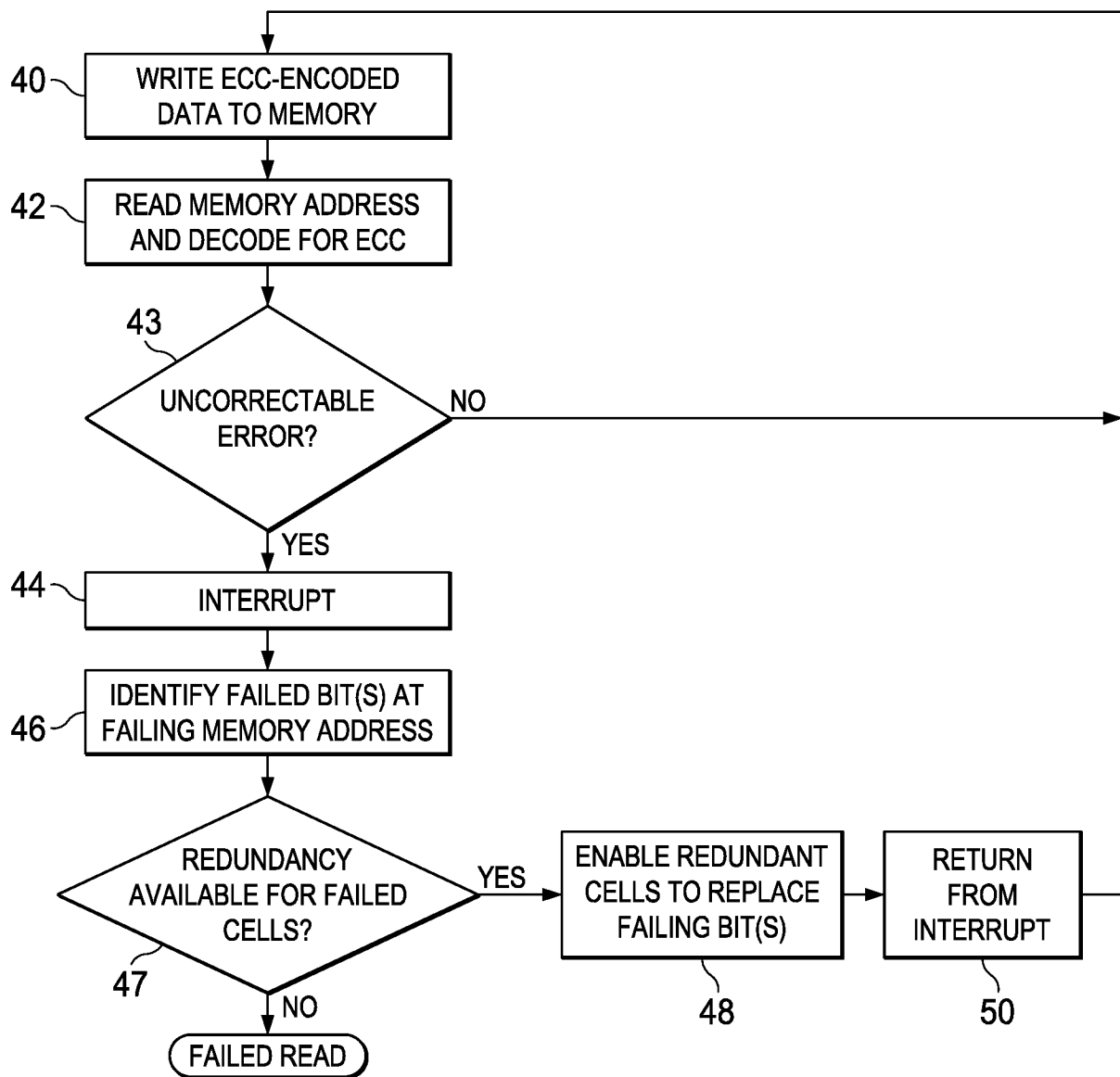


FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2016/041237

A. CLASSIFICATION OF SUBJECT MATTER <i>G11C 29/52 (2006.01)</i> <i>G11C 11/22 (2006.01)</i> According to International Patent Classification (IPC) or to both national classification and IPC				
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) G11C 29/00, 29/04, 29/52, 11/00, 11/21, 11/22, 11/34, 11/40, 11/401, 11/4063, 11/407, 11/409, 11/4099, 16/00, 16/02, 16/06, 16/10, H03K 19/00, 19/02, 19/185, H01G 7/00, 7/06, G11B 9/00, 9/02 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) PatSearch (RUPTO internal), USPTO, PAJ, Esp@cenet, DWPI, EAPATIS, PATENTSCOPE				
C. DOCUMENTS CONSIDERED TO BE RELEVANT				
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.		
X	US 2013/0212449 A1 (TAIWAN SEMICONDUCTOR MANUFACTURING COMPANY, LTD.) 15.08.2013, par. [0016], [0017], [0019], [0020], [0025], [0027]-[0030], [0033], [0036]-[0040], [0044], [0056], [0058]-[0060], [0062], [0064]	1-17		
Y		18-20		
Y	US 2015/0162063 A1 (FRAUNHOFER-GESELLSCHAFT ZUR FOERDERUNGDER ANGEWANDTEN FORSCHUNG E.V.) 11.06.2015, par. [0037]	18-20		
A	US 6766429 B1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 20.07.2004	1-20		
☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.				
* Special categories of cited documents: <table border="0"> <tr> <td style="vertical-align: top;"> “A” document defining the general state of the art which is not considered to be of particular relevance “E” earlier document but published on or after the international filing date “L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) “O” document referring to an oral disclosure, use, exhibition or other means “P” document published prior to the international filing date but later than the priority date claimed </td> <td style="vertical-align: top;"> “T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention “X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone “Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art “&” document member of the same patent family </td> </tr> </table>			“A” document defining the general state of the art which is not considered to be of particular relevance “E” earlier document but published on or after the international filing date “L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) “O” document referring to an oral disclosure, use, exhibition or other means “P” document published prior to the international filing date but later than the priority date claimed	“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention “X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone “Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art “&” document member of the same patent family
“A” document defining the general state of the art which is not considered to be of particular relevance “E” earlier document but published on or after the international filing date “L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) “O” document referring to an oral disclosure, use, exhibition or other means “P” document published prior to the international filing date but later than the priority date claimed	“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention “X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone “Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art “&” document member of the same patent family			
Date of the actual completion of the international search 15 September 2016 (15.09.2016)		Date of mailing of the international search report 13 October 2016 (13.10.2016)		
Name and mailing address of the ISA/RU: Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37		Authorized officer I. Kryazhev Telephone No. 499-240-25-91		