STABILIZATION CIRCUITS FOR MULTIPLE DIGITAL BITS

STABILISIERKREISE FÜR MEHRERE DIGITALE BITS

CIRCUITS DE STABILISATION POUR PLUSIEURS BITS

Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

Priority: 30.04.1996 US 640367

Date of publication of application:
17.02.1999 Bulletin 1999/07

Proprietor: Agate Semiconductor, Inc.
Santa Clara, CA 95054 (US)

Inventors:
- KORSH, George, J.
  Redwood City, CA 94061 (US)
- KHAN, Sakhawat, M.
  Sunnyvale, CA 94086 (US)

Representative: Brookes Batchelor
102-108 Clerkenwell Road
London EC1M 5SA (GB)

References cited:
WO-A-95/20224
US-A- 4 771 404
WO-A-96/04658

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).
FIELD OF THE INVENTION

[0001] This invention relates in general to semiconductor memories and, in particular, to the stabilization of semiconductor memories subject to drift effects. Such memories may store one or more digital bits per memory cell.

BACKGROUND OF THE INVENTION

[0002] Semiconductor memory integrated circuits, such as EEPROM, EPROM, FLASH and DRAM, have traditionally been used to store a single digital bit per memory cell, which is termed hereafter as single bit storage. Memories capable of storing more than one bit of digital data per memory cell and their benefits have been described previously. Such multiple bit per cell memories are called multilevel memories because they require more than the conventional two (conducting vs. non-conducting) cell threshold voltage, $V_T$, levels that are used for single bit storage techniques.

[0003] Each level in a multilevel memory cell represents a specific range of electrical charge stored in each memory cell and, in the case of nonvolatilite memories, i.e., EEPROM, EPROM, and FLASH memories, represents a specific range of cell $V_T$ values. To store N bits per memory cell requires the cell's $V_T$ range, and the amount of stored charge, to be divided into $2^N$ levels. Each level corresponds to a unique binary data pattern for all N bits. The cell is erased or programmed to store charge such that the cell's $V_T$ is set within one of these $2^N$ levels. Sensing circuits determine which level the storage cell's $V_T$ is in and read out the corresponding binary data pattern stored for the N bits. Digital information can thus be stored at a significantly lower cost per bit since N times the number of bits can be stored in the same memory cell array area previously storing just a single bit per cell.

[0004] In the following description for nonvolatile memory, a "level" is referred to as a range of $V_T$ values and not as a single voltage value. Furthermore, the term, cell $V_T$, is not used in the context since defined by solid state device physics, but rather in terms of how the sensing circuits determine the conductivity state of the memory cell. The conductivity is related to the $V_T$ of the cell. Similarly, a DRAM level represents a range of stored charge and not a single charge value.

[0005] The operation of sensing individual levels is performed by comparing the conductivity (or stored charge) of the memory cell in terms of sensing voltage or current with respect to multiple reference voltages or currents. The present invention is described in terms of voltage sensing since currents can easily be converted to voltages via load circuits by those practiced in the art of sensing schemes.

[0006] There are many problems associated with multilevel memories. Typically $2^N$-1 or $2^N$ reference voltage values, $V_{R_I}$, where $I = 1, 2, ..., 2^{N-1},$ or $2^N$, have been proposed to store N bits per memory cell to separate the $2^N$ levels from each other, with $V_{R_1} < V_{R_2} < ... < V_{R_{2^N}}$. It should be noted that a $V_{R_I}$ is sometimes referred to as simply $V_R$ for brevity. The relationship between reference voltage values and cell $V_T$ are illustrated in Figs. 1A-1C, which show the multilevel sensing reference voltages and cell $V_T$ distributions for a whole memory chip for single, two, and four bits stored per cell respectively.

[0007] An undesirable condition occurs when a memory cell $V_T$ is close to one of the $V_{R_I}$ voltages. The determination of the cell $V_T$ becomes ambiguous. Practical sensing circuits that determine cell $V_T$ are limited by circuit stability and speed, and by voltage and current variations caused by digital switching noise and other variations in power supply voltage, temperature, and silicon processing. Unlike analog signal storage, which does not require discrete multiple levels nor multiple reference voltages, digital memory storage techniques require the level to be unambiguously determined and requires such discrete reference voltages. If the level in a cell is incorrectly sensed, the digital memory malfunctions and up to N bits per cell may be lost.

[0008] To avoid the problems of sensing cell $V_T$ values close to, or equal to, one of the $V_{R_I}$ voltages, a margin voltage range, $V_{MMP}$ (see Fig. 2), to separate the cell's $V_T$ in one level from other adjacent levels has been proposed. This separation is performed at the time the cell is erased and programmed. However, both ends of each $V_{MMP}$ range are not defined. Rather, this proposed technique uses statistical control over the silicon process in conjunction with a lengthy programming algorithm to create adequate $V_{MMP}$ for storing a single bit, and possibly two bits, per cell. Additionally, no mechanism exists to determine if a cell $V_T$ overshoots the desired range. However, the technique is reliable only when two conditions are met. First, the separation between $V_{R_I}$ values must be large enough to provide adequate margin for unambiguous sensing. Secondly, the cell's $V_T$ must remain stable within the desired level and for as long as the data is to remain valid. This period may be as long as the lifetime of the memory chip.

[0009] However, a problem all multilevel storage techniques must overcome is the control of the $V_T$ of the memory cell within a very narrow range for each level. This $V_T$ control problem applies to all operational modes of the memory, including programming, erasing and reading the memory cells. The severity of the $V_T$ control problem increases geometrically as the number of bits stored per memory cell increases since $2^N$ levels per cell are required to store N bits per cell. The range of $V_T$ values within a single level, $V_L$, and the margin range separating the cell $V_T$ in different levels, $V_M$ (see Figs. 3A-C), narrows as the number of levels increase within a fixed range of all useful cell $V_T$ values, $V_P$. In this invention we sometimes refer to a $V_{L_I}$ (where I is one of...
the 2^N levels) as simply V_L, for brevity's sake. [0010] V_T is usually fixed since it is limited by the voltage range that can be applied to the terminals of the memory cell during the program, erase, and sensing operations. V_T is constrained by circuit speed, complexity and data storage reliability. For many previously proposed nonvolatile memory technologies, V_F is roughly equal to the power supply voltage for read operation, V_CC.

[0011] For the extremely simple case where the V_L range for all levels are defined to be the same and there is zero margin between levels, V_L = V_F/2^N. For example, the V_L range for a single bit storage technique operating from a 5 volt V_CC is 5/2 = 2.5V, while the V_L range for a 4-bit per cell multilevel memory operating from 3V reduces to 3/16 = 187.5mV. If a margin voltage range, V_M, is added between each level, V_F = V_L1 + V_M1 + V_L2 - V_M2 +...+V_M2^N-1 + V_L 2^N. Continuing with the previous simple case and assuming all V_M ranges are also equal, the level range now reduces to V_L = [V_F - (2^N-1) x V_M] / 2^N. With the introduction of a V_M equal to 0.1V, the V_L in the previous 4-bit, 3-volt V_CC example is now reduced to (3 - (15x0.1))/16 = 93.8mV.

[0012] Adding to the V_T control problem is the procedure for erasing memory cells. In practical embodiments of memory arrays, the erasure of cells is accomplished in blocks containing many cells so the V_T distribution of the fully erased cells is wider than the other levels that are programmed more selectively. The wider V_L range used to define the fully erased cells, V_L_ERASE, reduces the V_L range for the programmed levels even further. Figs. 3A through 3C graphically illustrate this problem for the same technologies shown in Figs. 1A-1C.

[0013] It should be noted that Figs. 1A, 1B, 1C, 2A, 3A, 3B, and 3C are examples of technologies where programming increases the cell V_T. The previous statements apply analogously for technologies that decrease the cell V_T during programming. In this case these drawings would show the wider erased level at the high end of cell V_T, instead of the low end.

[0014] Thus, V_T control is much more important in multilevel than in single bit storage systems since the range of V_T in each level is geometrically reduced as the number of bits per cell increases.

[0015] Furthermore, there are various mechanisms which can cause the cell's V_T to drift away from its initially programmed value. Many of these mechanisms are caused by the application of voltage stresses to the cell subsequent to its original programming. These conditions are called "disturbs" and are unavoidable when the cell is grouped within an array of other similar cells to make a useful memory system. Program and erase disturb stresses are present for the shortest cumulative time but are important causes of V_T drift due to the high voltages used during these operations. For example, disturb data is routinely reported in the technical literature describing new FLASH memory technology.

[0016] Other sources of V_T drift are due to the effects of changes in the numbers or positions of trapped charges around the floating gate or between the floating gate and substrate of the memory cell. Trapped charges may be due to defects or the cumulative effects of high electric fields applied over time, such as during repeated program/erase (P/E) cycles. The more erase or programming currents are driven through a gate dielectric, the more trapped charges accumulate. This leads eventually to the retardation of those operations and a malfunction of the part.

[0017] Trapped charge may also occur in an unrepeatable fashion with P/E cycling. This has been attributed to the so called "rogue bit" effect. A rogue bit exhibits a shift in programming or erase ability on one cycle and reverts back to normal on another cycle. The rate of V_T drift may not be constant over time and may not be the same from cell to cell due to the statistical variations of the defects present. Each cell in a memory cell array experiences its own unique combination of disturb and charge stress conditions since different rows and columns, within which the cells are interconnected, are accessed in various orders. The drift effects caused by random voltage stresses are added to the drift effects caused by the random defects. These effects are cumulative on the cell's V_T drift.

[0018] Nonvolatile cells eventually lose endurance with repeated P/E cycles, that is, they program or erase too slowly for the time allowed for these operations and eventually fail. Some previous programming techniques count the number of times a particular section of memory cell array has been cycled. Such cycling data is then used, in one example, to prevent further P/E cycling of that overused section of memory cell array based on a predetermined conservative maximum number of cycles. This programming technique may thus act to reduce the functionality of the memory before being necessary.

[0019] Finally, all the V_T disturb and drift mechanism problems described above worsen due to the increasing electric fields in the cell as the physical sizes of memory cells are reduced. These problems are known impediments to the scaling of cells and barriers to more economical memory chips. Cell V_T drift problems become even more severe as the V_L range is narrowed when storing multiple bits per cell or lower power supply voltages are used, for example.


[0021] The present invention solves or substantially mitigates these problems. A memory according to the present invention directly measures the stability, programmability, and erasability of each nonvolatile memory cell over the life of the part.

SUMMARY OF THE INVENTION

[0022] In one aspect the present invention provides an integrated memory system as defined in claim 1.
In another aspect, the invention provides a method of operating an integrated circuit memory system as defined in claim 17.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1A is an example of the sensing reference voltages and cell $V_T$ distributions for a whole memory chip for single bit per cell data storage.

Fig. 1B is a prior art example of the sensing reference voltages and cell $V_T$ distributions for a whole memory chip for two bit per cell data storage.

Fig. 1C is an example of the sensing reference voltages and cell $V_T$ distributions for a whole memory chip for four bit cell data storage.

Fig. 2 is a prior art example of the sensing reference voltages for read, program, and erase modes and resulting cell $V_T$ distributions for a whole memory chip for two bit per cell data storage showing the margins separating different $V_T$ levels.

Fig. 3A is a prior art example defining the ranges of $V_F$, $V_{L-ERASE}$, $V_M$, and $V_L$ cell $V_T$ distributions for a whole memory chip for single bit per cell data storage.

Fig. 3B is a prior art example defining the ranges of $V_F$, $V_{L-ERASE}$, $V_M$, and $V_L$ cell $V_T$ distributions for a whole memory chip for two bit per cell data storage.

Fig. 3C is an example defining the ranges of $V_F$, $V_{L-ERASE}$, $V_M$, and $V_L$ cell $V_T$ distributions for a whole memory chip for four bit per cell data storage.

Fig. 4A is an example of cell $V_T$ drift versus time for rising $V_T$.

Fig. 4B is an example of cell $V_T$ drift versus time for falling $V_T$.

Fig. 5A is an example of cell $V_T$ drift versus time for rising $V_T$ showing the two new multilevel digital data from a memory cell using the BSERD technique as implemented in the preferred embodiment.

Fig. 5B is a continuation of Fig. 14A and is an example showing the two new multilevel digital data from a memory cell using the BSERD technique as defined in claim 17.
The memory of the present invention responds to any of the above three possibilities for each memory cell and determines the action necessary to restore the cell VT. This response and determination may occur long after the original data programming. The memory system does not necessarily perform an erase before programming. Instead, the memory performs mini-erase and mini-program operations which adjust the VT in the correct direction. Only a small amount of stored charge required to make the minor VT restoration is moved during these operations. This avoids disturbing the data on other cells. The wearing effects of previous P/E cycling techniques are minimized.

Additionally, the memory system corrects rogue bits that are created during the large charge flux through the gate dielectric during a normal P/E cycle. During a restore operation, only small changes in the cell VT are required to restore the VT.

FLASH nonvolatile memory cell arrays are normally optimized to reduce the chip's silicon area by requiring that cells be electrically erased in large groups called sectors, or erase blocks. Programming is accomplished in a much more selective way such that fewer bits are simultaneously programmed in a smaller group, such as a page, word, or byte. All the cells within the programming page, word, or byte may also be sensed (read or verified) simultaneously. An erase sector or block may contain multiple programming pages.

Programming shifts the VT of a memory cell in the opposite direction than erasure does. It should be noted that the definitions for programming and erasure used herein are not tied to the polarity nor direction of charge movement onto a charge storage region, such as a floating gate, but instead are determined by the memory cell array. Thus, the present invention is applicable to both technologies which program by increasing or decreasing the cell VT.

In programming the integrated circuit cells, the memory system first completely erases all the cells in the erase block prior to programming any new data bits into any page within that erase block. Previous programming algorithms program each cell to the desired VT in a series of programming pulses. Multilevel programming prefers that the VT of each cell be individually set to obtain the tightest possible VL during programming, in anticipation of future drift. Previous multilevel programming algorithms require many program/verify iterations to shift the cell VT in one direction. Statistical process control is heavily relied upon to prevent the fastest programming cells' VT from overshooting the desired range within a level and resting too close to a VR. To avoid overshoot, many small programming iterations are required.
In contrast, the programming operation of the memory system described herein may shift the cell's \( V_T \) in both directions employing mini-erase operations during the programming mode. The mini-erase operations allow large programming iterations to be performed. Both edges of each \( V_L \) range are controlled during programming, rather than relying on statistical control of the programming characteristics of each cell. Another advantage of the programming technique described below is that the programming time for a multilevel memory is reduced since the mini-programming and mini-erase operations reduce the initially programmed wide cell \( V_T \) distribution in each level and correct any \( V_T \) overshoot.

With the present invention, a memory system can determine when a cell is no longer capable of reliably programming, erasing or storing data by direct measurement while embedded in the external system. Appropriate action is taken only after determining that such action is immediately required. This extends the useful life of the memory chip.

The various aspects of a memory system according to the present invention are described in greater detail below. The multilevel data sensing, mini-programming, mini-erasing, programming and erasing operations follow:

I. Multilevel Digital Data Sensing

General Description of Sensing Operations

The present invention provides for a novel multilevel digital data sensing technique. The technique uses additional reference voltages and guard bands during sensing operations to determine whether a memory cell \( V_T \) has drifted undesirably close to either one of the level separation reference voltages \( V_{RI} \) or \( V_{R(I+1)} \), which define one of the \( V_L \) levels.

As shown in Fig. 5A and Fig. 5B, the reference voltages \( V_{UGI} \) and \( V_{LG1} \) introduce cell \( V_T \) read margin guard bands during the sensing mode. For each level "I", there are two read margin guard bands; 1) the upper read margin guard band, \( V_{MRUI} = V_{R(I+1)} - V_{UGI} \), and 2) the lower read margin guard band, \( V_{MRLI} = V_{LG1} - V_{RI} \). The guard bands \( V_{MRUI} \) and \( V_{MRLI} \) are referenced with respect to the \( V_{RI} \). For example, the \( V_{MRUI} \) is referenced with respect to \( V_{R(I+1)} \) and the \( V_{MRLI} \) is referenced with respect to \( V_{RI} \). The total read margin guard band around a \( V_{RI} \) is \( V_{MRLI} = V_{MRUI} + V_{MRLI} \).

If the new multilevel digital data sensing technique detects that a cell \( V_T \) has drifted into one of the read margin guards bands, then a restore cycle condition is established and a conditional status flag is set. The new multilevel digital data sensing technique also uses \( (2^N + 1) \) level separation reference voltages, \( V_{RI} \), where \( N \) is the number of multilevel digital bits stored per memory cell and \( I = 0 \) to \( 2^N + 1 \). Two additional read margin guard bands are also used to sense whether a cell has drifted into the extreme limits of reliable data sensing.

Binary Search Embedded Restore Detection Methodology

The new multilevel digital data sensing technique, called the Binary Search Embedded Restore Detection (BSERD) technique, senses the multiple bits of data that are stored per memory cell as multilevel digital data, and determines whether a restore cycle condition is established using band sensing detection.

Fig. 6 shows the general operation of a preferred embodiment of the BSERD technique for sensing four bits (\( N = 4 \)) of multilevel digital data per memory cell. The choice of \( N = 4 \) for this embodiment is only for the purpose of teaching this invention and does not limit the application of this invention to other choices for \( N \). This invention could, for example, be used for single bit per cell storage (\( N = 1 \)) or any other multibit storage applications where \( N > 1 \).

During each "Set BitX" cycle (\( X = 0 \) to \( N-1 \)), two subcycle operations are performed. These subcycles are the data detection subcycle followed by the restore detection subcycle. During each of the subcycles, a suitable reference voltage \( V_{CRF} \) is set and compared with the sensed cell \( V_T \). The \( V_{CRF} \) voltage at a certain time point depends on the results of the comparisons between the \( V_{CRF} \) voltage and the sensed cell \( V_T \) at each of the previous time points. After the requisite number of Set BitX cycles are finished, a fringe detection subcycle and a fringe restore subcycle are performed to detect whether the cell \( V_T \) is within the extreme reliable limits of the cell \( V_T \) sensing range and whether the cell \( V_T \) is within the extreme upper or lower restore margin range.

Multilevel Memory Sensing System

Fig. 7 shows the block diagram of a preferred embodiment of the invention. Four bits of multilevel digital data are sensed from each memory cell. Data, address, control, and system clock signals are interfaced between the outside system and the preferred embodiment by a DATA, ADDR & CNTRL SYSTEM INTERFACE block 10 via a DATA bus 11, an ADDR bus 12, a CNTRL bus 13 and a CLKIN line 14 respectively. Alternatively, an internal CLOCK block (not shown in Fig. 7) generates a clock signal on a CLOCK signal line 15, shown in Fig. 7. During sensing operations, the DATA, ADDR & CNTRL SYSTEM INTERFACE block 10 provides the address information over a bi-directional address, ADDRBUS bus 16, to a respective horizontal decoder, X-DEC 17, a vertical decoder, Y-DEC 18, and a block decoder, BLOCK-DEC 19, to select an appropriate memory cell 20, in a memory array block, CELL ARRAY 21.

The decoders provide the appropriate multilevel programming, erase, or sensing voltages at the ter-
mininals of the cell 20, depending on signals on a
PRG-ERS control bus 22, a HVCTRL control bus 23, a
PRGHV high voltage line 24, and a HVOUT high voltage
bus 25. The PRG-ERS and HVCTRL signals are gen-
erated by a PROGRAM ERASE READ RESTORE SE-
QUENCER block 26. The HVOUT and PRGHV signals are
generated by a HVGEN block 27.

[0073] During sensing, a RD line 28 is forced high to
turn on a GATEC switch 29 to connect a MLLINE line
30 to a VMEM line 31. The MLLINE line 30 is connected
to the cell 20 through a Y-MUX block 32 and a BL00 bit
line 71. The particular cell 20 which is connected to the
MLLINE line 30 during sensing depends on the Y-MUX
block 32, responsive to signals on a YDECO bus 34 gen-
erated by the Y-DEC block 18. The VMEM line 31 is con-
ected to the non-inverting input of a comparator CMP
35. When high, the RD line 28 also allows the connec-
tion of a load block circuit, CELL LOAD 36, to the VMEM
line 31.

[0074] When the appropriate voltages are applied at the
terminals of the addressed cell 20 and the CELL
LOAD 36 is connected, a voltage representative of four
bits of digital data is developed at the VMEM line 31.
The PROGRAM ERASE READ RESTORE SEQUENC-
ER block 26 also sends the signals through a PRGRDB
bus 37 and the DATABUS[0-3] bus 44, which in turn are controlled by the sequencers 26 and 43. During sensing, the voltage outputs from the VX VY
GEN block 38 are VX = VRI and VY = VRI, where I is an
integer number between 0 and 16, determined by the
data lines on the DATABUS[0-3] bus 44. During sensing,
as also shown in Fig. 6, the ΔVU ΔVL GEN block 39
generates output voltages ΔVL and ΔVU, where
ΔVL = VMRLI - ΔVU = VR16(I-1), I = 0 to 15, respectively.

[0075] During sensing operations, a BINARY
SEARCH READ SEQUENCER block 43 and the PRO-
GRAM ERASE READ RESTORE SEQUENCER block
26 control and sequence the BSERD technique syn-
chronized by the clock signal on the CLCK line 15 from
the DATA, ADDR & CNTRL SYSTEM INTERFACE
block 10. The BINARY SEARCH READ SEQUENCER
block 43 forces the appropriate data line on a bi-direc-
tional bus, DATABUS[0-3] 44.

[0076] The VCRFGEN block 40 also receives control
signals from a control line, PHSSEL line 46, from the
PROGRAM ERASE READ RESTORE SEQUENCER
block 26 and a control line, STPDIR line 47, from a STP-
DIR latch 52. The STPDIR latch 52 is connected to the
output terminal of the CMP comparator 35 by an output
line VCMP 49 and a transfer gate, GATEA 50. Based
upon the input signals from the DATABUS[0-3] bus 44, the
VX VY GEN block 38 and ΔVU ΔVL GEN block 39,
generate the appropriate voltages to the VCRFGEN
block 40. Table 1 shows the output voltages from the
VCRFGEN block 40 on an output VCRF line 45, during
sensing, based on levels on the PHSSEL and STPDIR
lines 46 and 47. The VCRF line 45 is connected to the
inverting input of the CMP comparator 35.

Table 1.

<table>
<thead>
<tr>
<th>PHSSEL</th>
<th>STPDIR</th>
<th>Voltage at VCRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOW</td>
<td>Don’t Care</td>
<td>VX</td>
</tr>
<tr>
<td>HIGH</td>
<td>LOW</td>
<td>VX + ΔVL</td>
</tr>
<tr>
<td>HIGH</td>
<td>HIGH</td>
<td>VY - ΔVU</td>
</tr>
</tbody>
</table>

[0077] Fig. 6 illustrates the reference voltages VR0
through VR16; any one of which generated as output by
the VX VY GEN block 38 under the control signals of
the PRGRDB bus 37 and the DATABUS[0-3] bus 44,
which in turn are controlled by the sequencers 26 and
43. During sensing, the voltage outputs from the VX VY
GEN block 38 are VX = VRi and VY = VRi, where I is an
integer number between 0 and 16, determined by the
data lines on the DATABUS[0-3] bus 44. During sensing,
as also shown in Fig. 6, the ΔVU ΔVL GEN block 39
generates output voltages ΔVL and ΔVU, where
ΔVL = VMRLI - ΔVU = VR16(I-1), I = 0 to 15, respectively.

[0078] A DATA[0-3] LATCHES block 48 has four latches
that store the four bits from each memory cell 20 as
sensed through the BSERD technique. The CMP compo-
parator 35 compares the voltages at the VMEM input
line 31 and the VCRF input line 45 and generates a logic
high or low signal on the VCMP 49 output line 49. The
VCMP 49 line 49 is driven to a logic high signal when
the voltage on the VMEM line 31 is higher than the
voltage on the VCRF line 45. Conversely, the VCMP 49
line is driven to a logic low signal when the voltage on
the VMEM line 31 is lower than the voltage on the VCRF
line 45.

[0079] Besides the transfer switch, GATEA 50, the
output VCMP 49 line is connected to the input of a
transfer switch, GATEB 51. The output line of GATEA
50 connects to the input line of the STPDIR LATCH 52.
The output line of GATEB 51 is connected to the input
of a latch, RES LATCH 53. The PHSSEL line 46 from
the PROGRAM ERASE READ RESTORE SEQUENC-
ER block 26 controls the operation of the switches,
GATEA 50 and GATEB 51. When the PHSSEL line 46 is
high, the output line VCMP 49 from the CMP compo-
parator 35 is connected to the input of the RES LATCH
53 through the switch GATEB 51. When the PHSSEL
line 46 is low, the VCMP 49 line 49 is connected to the
input of the STPDIR LATCH 52 through the GATEA
switch 50.

[0080] A RST line 54 from the PROGRAM ERASE
READ RESTORE SEQUENCER block 26 is connected to
both the STPDIR LATCH 52 and the RES LATCH 53.
The two latches 52 and 53 are reset when the RST line
54 is logic high. When the RES LATCH 53 is reset, an
output line from the latch 53, RES line 55, is low. The
output STPDIR line 47 from the STPDIR LATCH 52 car-
ries an inverted signal from the latch 52. When the STP-
DIR LATCH 52 is reset, the output STPDIR line 47 is
high. The STPDIR line 47 is connected to a high voltage
switch, HVSW 56, the VCRFGEN block 40, and transfer switches, GATED 57, GATEF 58 and GATEH 59. When the STPDIR line 47 is high, the HVSW switch 56 passes high voltage pulses on the PRGHV line 24, which is connected to the high voltage generator, HVGEN block 27, to the MLLINE line 30. When the STPDIR line 47 is low, the HVSW switch 56 disconnects the PRGHV line 24 from the MLLINE line 30. [0081] The high voltage pulses are generated by the HVGEN block 27 when the HVCTRL bus 23 from the PROGRAM ERASE READ RESTORE SEQUENCER block 26 is forced to set a programming or erase operation. During sensing, the high voltage PRGHV line 24 is allowed to float and the HVGEN block 27 does not generate any high voltage pulses, since the HVCTRL bus is set for sensing. The HVGEN block 27 also provides high voltages through the HVOUT bus 25 to the Y-DEC block 18, the X-DEC block 17 and the BLOCK-DEC block 19 during the programming, the erase, the restore, and potentially also the sensing, operation modes. [0082] The Y-MUX block 32 selects the appropriate bit (sense) line of the addressed cell 20 under the control of the Y-DEC block 18. A REGISTERS & ADDRESS BUFFER block 60 stores the addresses of the memory cells which should be restored along with the restore conditional status flags as determined by the BSERD technique. The REGISTERS & ADDRESS BUFFER block 60, connected to the ADDRBUS bus 16, is controlled by the PROGRAM ERASE READ RESTORE SEQUENCER block 26 and is connected via a BFCTLBUS bus 61. [0083] GATED and GATEE switches 57 and 62 are connected in series to a GROUND line 64, which is set at a logic low signal. These switches 57 and 62 are controlled by signals on the STPDIR line 47 and RES line 55 respectively. Similarly, GATEF and GATEH switches 58 and 63 are connected in series to a POWER line 67, which is set at a logic high signal. These switches 58 and 63 are also controlled by the STPDIR line 47 and RES line 55 respectively. [0084] When both the STPDIR line 47 and the RES line 55 are high together, a logic low signal is passed through the GATED and GATEE switches 57 and 62 to an FLGB line 65. Under any other combinations of signals on the STPDIR line 47 and the RES line 55, the logic low signal is not passed to the FLGB line 65 which allows a PULL UP LOAD block 66 to force a high logic signal on the FLGB line 65. When both the STPDIR line 47 and the RES line 55 are low together, a logic high signal is passed through the GATEF, GATEG switches 58 and 63 to a FUGB line 68. Under any other combinations of signals on the STPDIR line 47 and the RES line 55, the logic high signal is not passed to the FUGB line 68; a PULL DOWN LOAD block 69 then forces a low voltage on the FUGB line 68. The FLGB, FUGB lines 65 and 68 are input lines to the PROGRAM ERASE READ RESTORE SEQUENCER block 26 and the BINARY SEARCH READ SEQUENCER block 43. A data latching pulse, generated by the PROGRAM ERASE READ RESTORE SEQUENCER block 26 and conducted through a LCHDAT line 70, passes through the GATEH switch 59 to the DATA[0-3] LATCHES block 48 when the STPDIR line 47 is low. [0085] Fig. 8 shows the flow chart of the BSERD technique as implemented. The BINARY SEARCH READ SEQUENCER block 43 and the PROGRAM ERASE READ RESTORE SEQUENCER block 26 contain the necessary control circuitry to implement the BSERD technique. The design of the circuitry should be evident to one skilled in the art of integrated circuit design. [0086] Fig. 9 shows an example of sensing multilevel digital data from a memory cell. The arrows in Fig. 9 show the transitions of the reference voltages at the VCRF input line 45 into the CMP comparator 35 at certain time points responsive to the condition of the output of the CMP 35 at each of the previous time points, as shown on the time axis. [0087] Fig. 10 shows the exemplary timing diagram for sensing multilevel digital data from a memory cell 20, as shown in Fig. 9 and described hereafter. The cell V_T of the selected cell 20 which has been appropriately programmed previously is sensed on the VMEM line 31 (see Fig. 7) to be at the dark dotted horizontal line, shown as "Cell V_T" in Fig. 9. At time T_0, the beginning of the BSERD technique, the RST line 54 is pulsed high to reset the STPDIR and the RES latches 52 and 53. The PGRGRD bus 37 is set, the PHSSEL line 46 is low, the HVCTRL bus 23 is set for sensing, the RD line 28 is high, and the ADDRBUS bus 16 contains the address of the selected memory cell being sensed. After a reset pulse on the line 54, the STPDIR line 47 is high and the RES line 55 is low. The FLGB line 65 is high and the FUGB line 68 is low. [0088] Now the "Set BIT3" cycle (see Fig. 6) and the first data detection subcycle begins. The most significant bit line of the DATABUS[0-3] bus 44, Bit3, is forced high by the BINARY SEARCH READ SEQUENCER block 43. With Bit3 forced high, (0001) on the DATABUS[0-3] bus 44, the VX VY GEN block 38 outputs the reference voltage V_RB at time T_1D, and the VCRF generator block 39 outputs the reference voltages V_MRL8 and V_MRL7. Since the PHSSEL line 46 is low at this time, the VCRFGEN block 40 outputs the reference voltage V_RB on the VCRF line 45. Just after time T_1D, the CMP comparator 35 senses that the VMEM line 31 is higher than the VCRF line 45 and generates a logic high on the VCMPO line 49. With the PHSSEL line 46 low, the GATEA switch 50 passes the logic high signal on the VCMPO line 49 to the input line of the STPDIR LATCH 52. The latch 52 is set and generates a logic high signal on the STPDIR line 47. With the STPDIR line 47 low, the latching pulse on the LCHDAT line 70 from the PROGRAM ERASE READ RESTORE SEQUENCER block 26 passes through the GATEH switch 59 to latch a "1" in a DATA3 latch in the DATA[0-3] LATCHES block 48. This marks the end of the data detection subcycle.
Next, the first restore detection subcycle begins. The PHSSEL line 46 is set high. Since the PHSSEL line 46 is high and the STPDIR line 47 is low, the VCRFGEN block 40 outputs the voltage $V_{RR} + V_{MRL8}$ on the VCRF line 45 at time $T_{IR}$. The CMP comparator 35 senses that the signal on the VMEM line 31 is still higher than the signal on the VCRF line 45 and generates a logic high signal on the VCMPO line 49. Since the PHSSEL line 46 is logic high, the signal on the VCMPO line 49 passes to the RES LATCH 53 through the GATEB switch 51 and does not pass through the GATEA switch 50 to the STPDIR LATCH 52. Since the level is high at the input of the RES LATCH 53, the output RES line 55 is set high. This is the end of the first restore detection subcycle and the end of the "Set Bit3" cycle.

The BSERD technique performs the sequence of events and subcycles as described in the above paragraph, performing the Set Bit2, the Set Bit1 and the Set Bit0 cycles in sequence. Fig. 9 shows the reference voltages set at the VCRF line 45 at various time points. Fig. 10 shows the output logic voltages on the VCMPO line 49, STPDIR line 47, RES line 55, FLGB line 65, FUGB line 68 and the DATABUS[0-3] bus 44 at various discrete times. If at the end of one of the data detection subcycles, the output line VCMPO 49 from the CMP comparator 35 remains low, which signifies that the voltage on the VMEM line 31 is lower than the voltage on the VCRF line 45, then the STPDIR line 47 is high. When that occurs, no data latching pulse from the LCHDAT line 70 is passed through the GATEH switch 59. Then the appropriate data latch in the DATA[0-3] LATCHES block 48 remains reset at "0". This condition does not occur in this example. Similarly, if at the end of any of the restore detection subcycles, both the STPDIR line 47 and the RES line 55 become high, then a restore low cycle condition is established, with the FLGB line 65 being forced low. If at the end of any of the restore detection subcycles, both the STPDIR line 47 and the RES line 55 become low, then a restore high cycle condition is established, with the FUGB line 68 being forced high.

At the completion of the data detection and restore detection subcycles, the fringe detection and the fringe restore subcycles are performed by the BSERD technique. At the end of the final "Set Bit0" cycle in the prior example, the data on the DATABUS[0-3] bus 44 is (1111). At the beginning of the fringe detection subcycle, the PROGRAM ERASE READ RESTORE SEQUENCER block 26 detects the condition and the status of the FLGB line 65 and FUGB line 68 are internally saved on restore cycle status latches. The restore cycle status latches are read out via the DATABUS[0-3] bus 44, the DATA, ADDR & CNTRL SYSTEM INTERFACE block 10, and the DATA bus 11.

In the preferred embodiment, there are two conditions, restore low and restore high, that can cause a restore cycle condition at the end of the restore detection or fringe restore subcycles. STPDIR line 47 = high and RES line 55 = high is the restore low condition which forces the FLGB line 65 low. STPDIR line 47 = low and RES line 55 = low is the restore high condition which forces the FUGB line 68 high (see Fig. 7 and Fig. 8). If any time during the operation of the BSERD technique, a restore cycle condition is established, then the PROGRAM ERASE READ RESTORE SEQUENCER block 26 detects the condition and the status of the FLGB line 65 and FUGB line 68 are internally saved on restore cycle status latches. The restore cycle status latches reside inside the PROGRAM ERASE READ RESTORE SEQUENCER block 26. If a restore cycle condition has been set during the operation of the BSERD technique, then upon its completion, the PROGRAM ERASE READ RESTORE SEQUENCER block 26 initiates an operation to save the address of the memory location requiring the restore cycle and also the type of restore cycle required, in the REGISTERS & ADDRESS BUFFER block 60.

Storage of the address requiring a restore cycle and also the type of restore required, allows the flex-
ibility to perform the restore cycle at a later time. If the restore cycle operation is immediately performed on the current memory location, then the address and type of restore cycle required is not stored in the REGISTERS & ADDRESS BUFFER block 60. The REGISTERS & ADDRESS BUFFER block 60 can store the restore cycle information for a plurality of addresses which require a restore cycle operation. The REGISTERS & ADDRESS BUFFER block 60 can either reside in the same integrated circuit or in a separate system and may consist of either volatile or nonvolatile memory.

[0096] Fig. 6 and Fig. 7 show data sensing from one memory cell. The preferred embodiment also works for sensing data from a page of cells. A Y-DEC block 33 in Fig. 7 is repeated as many times as the number of cells, J, that exist in a page. The lines going across the boundary of the Y-DRIVER block 33 in Fig. 7 are common to all the Y-DEC blocks 33, except the lines to the CELL ARRAY block 21. The FLGB line 65 and FUGB line 68 are connected together in a wired-OR manner between all the Y-DEC blocks 33.

[0097] Fig. 11 shows a more detailed organization of the memory array and the Y-DRIVER blocks 33 for page mode operation, in one embodiment. The same organization is used during programming and restoring cycles operating on a page of cells and during erase cycles on an erase block of cells. There are "J" Y-DRIVER blocks 33 shown. The YDECO bus 34 from the Y-DEC block 18 in Fig. 7 controls the selection of the active page, through the Y-MUX 32 within each Y-DRIVER block 33.

[0098] The circuits and memory cell array shown in Fig. 11 within the outlined block labeled CELL ARRAY 21 are not part of this invention, but are described here to explain how the embodiment of this invention interfaces to an arbitrary memory cell array.

[0099] There are "S" pages shown in Fig. 11. There are also "K" blocks 83-84 shown. Each block may consist of a plurality of rows of cells. The block selection and the row selection are together performed by the BLOCK-DEC and the X-DEC blocks 19 and 17 respectively, as shown in Fig. 7. BSU sub-blocks 77 and BSL sub-blocks 78 within each block 83-84 perform selection or deselection of bit (sense) lines (BL00 line 71 through BLSJ line 72) or common lines (CLO line 75 through CLK line 76) for each block 83-84 in the array. The BSU blocks 77 and BSL blocks 78 are controlled by SLUK lines 79 and SLLK lines 80. The SLLO line 81, SLU0 line 82 and CL0 line 75 for the BLOCKO block 83 and the SLLK line 80, SLUK line 79 and CLK line 76 for the BLOCKKK block 84, as shown in Fig. 11 are generated by the BLOCK-DEC block 19 in Fig. 7.

[0100] Word lines (WL00 line 85 through WLKT line 86) select the appropriate row within all blocks 83-84. There are "T" wordlines per block shown in Fig. 11. The word lines are controlled by the X-DEC block 17 shown in Fig. 7. A plurality of X-DEC blocks 17, BLOCK-DEC blocks 19 and Y-DEC blocks 18 are used. For brevity, only one of each is shown in Fig. 7. In page operation, a block 83-84 is selected, a row is selected within that block, and a page is selected within that row. The heavily outlined cells 20 shown in Fig. 11, form page 1 in row 1 in block 0. This page 1 is selected through the bit (sense) BL10 line 73 through BL1J line 74, the word WL01 line 87 and the block select SLU0 line 82, SLL0 line 81, and CL0 line 75. An erase operation directly effects all the cells within a selected block 83. A program operation directly affects only the cells in a selected page. This embodiment of the invention can be interfaced to any nonvolatile memory cell array.

II. Mini-Programming

General Description of Mini-Programming Operations

[0101] A new multilevel programming technique called mini-programming is described in this section as part of the present invention. Unlike prior art programming techniques which start programming only into cells which are in a fully erased condition, the mini-programming technique operates without the need for a full erase and on cells which still contain valid data. In the prior art, the erase operation shifts the VT of all the cells in the selected erase block or sector all the way over to one end of the useable cell VT range, V_F, as shown in Figs. 3A-3C, and labeled V_LERASE. After an erase operation, the prior art programming technique shifts the selected cell’s VT out of the V_LERASE range and into one of the other levels by moving a great deal of charge through the oxides surrounding the floating gate of the memory cell.

[0102] In contrast to the prior art, the mini-programming technique of the present invention moves only a small amount of charge without the need for a full erase. Each selected cell’s V_F is shifted only a small amount and only within the limits of that cell’s original V_L level. The deleterious effects on cell P/E cycling endurance caused by the prior art programming technique are avoided, since only a small amount of charge is moved by the new mini-programming technique. Also, only that amount of charge is moved as is necessary to correct the previously determined V_T drift. The mini-programming technique takes less time than programming from the erase condition as is done by the prior art. Additionally, the mini-program affects the selected cell’s V_F only slightly, so the resulting program disturb effect on other unselected cells is negligible. The mini-programming operation can shift the V_F a small amount within a V_L level in the opposite direction of erase and thus performs one of the functions necessary for the desired V_F restoration during a restore operation to enable reliable multilevel data storage.

[0103] The mini-programming technique utilizes the band sensing capabilities of the present invention, similar to those previously described in the multilevel digital sensing section, to read and verify that each cell’s V_F is stored in the desired band of the V_L range. This ensures
adequate margin guard bands to prevent ambiguous sensing at both ends of that V\textsubscript{L} range. This is accomplished by new multiple reference voltages during the mini-programming verify operations.

The mini-programming technique of this invention has two additional new reference voltages at V\textsubscript{RI} + V\textsubscript{MPLI} and V\textsubscript{RI+1} - V\textsubscript{MPUI}. These reference voltages, in conjunction with V\textsubscript{RI} and V\textsubscript{RI+1}, define two new lower and upper mini-programming guard bands, V\textsubscript{MPLI} and V\textsubscript{MPUI}, within each of the V\textsubscript{L} levels, as shown in Fig. 12. V\textsubscript{MPLI} and V\textsubscript{MPUI} may, or may not, be equal to, and may, or may not, be greater than the previously described V\textsubscript{MRLI} and V\textsubscript{MRUI} read margin guard bands. The V\textsubscript{MPLI} and V\textsubscript{MPUI} bands within each of the V\textsubscript{L} levels are chosen such that the resulting V\textsubscript{T} distribution after mini-programming is optimized within a desired band which is narrower than the V\textsubscript{L} range. The optimization may require the V\textsubscript{MPLI} and V\textsubscript{MPUI} bands to be unequal to each other and may require different band values for each of the V\textsubscript{L} levels and even arbitrary band values dictated by the memory cell array process technology. Alternatively, the V\textsubscript{MPLI} and V\textsubscript{MPUI} in each of the V\textsubscript{L} levels may be of equal value. This invention can be adapted to accommodate the optimization for a wide variety of memory processing technologies.

The verify technique of this invention contrasts with previous techniques which verifies the cell V\textsubscript{T} only relative to a single value and relies on statistical process control to prevent the cell V\textsubscript{T} from overshooting the desired V\textsubscript{T} band. V\textsubscript{T} overshoot is a possibility during any programming operation and may occur due to the previously mentioned rogue cell effects. These effects are caused by charges trapped in the oxides surrounding the floating gate of the memory cell or other process-induced defects.

The mini-programming technique minimizes charge trapping and the possibility of creating a rogue cell by moving only small amounts of charge through the gate oxides. A mini-programming verify operation detects if the V\textsubscript{T} has been shifted too much into one of the guard bands so that a mini-erase operation (described below) is used to reestablish the desired V\textsubscript{T} within that level. This recovery capability for V\textsubscript{T} overshoot is not present in the prior art without first completing a full erase and reprogram operation. Using the restore methodology of this invention, full erase and reprogramming is avoided during a V\textsubscript{T} overshoot recovery and the P/E endurance lifetime of the chip is preserved.

Mini-Programming Methodology

The mini-programming technique is best described by considering a plurality of cells which are connected in the same programming page. A practical multilevel memory system utilizes a page mode architecture to reduce programming time such that multiple cells are programmed simultaneously. It is necessary to comprehend the behavior of all the cells at least within such a page, if not the whole chip, to observe all the possible ramifications of any form of programming.

Fig. 12 shows the V\textsubscript{T} distribution of a plurality of cells in the memory which are within the same level, V\textsubscript{L}, at two different times, T\textsubscript{RESP} and T\textsubscript{RESP}. In this example, one V\textsubscript{T} distribution shows the condition after the cells have been read using the previously described BSERD technique of this invention at time T\textsubscript{RESP}. The BSERD technique has detected two cells with V\textsubscript{T} values at A and B within the V\textsubscript{MRLI} guard band. The restore conditional flags have been set to indicate that a restore operation is necessary. The BSERD technique has also determined that the restore operation may be accomplished without the need for a mini-erase since no cells were detected within the V\textsubscript{MRUI} guard band and a restore high condition flag remains set for this page address.

In one embodiment of the present invention, the chip conveniently executes a mini-programming operation at a later time to restore the V\textsubscript{T} of the page of cells which have previously been flagged for a restore operation without mini-erase. First, the cells are read by a simplified BSERD technique using only the data detection subcycle without the restore detection subcycle to latch the N binary bits corresponding to the V\textsubscript{T} which was stored in each read cell.

In an alternative embodiment of the present invention, if a mini-program operation is performed immediately after a regular full BSERD read, the first simplified BSERD step is omitted since the N binary bits have been already latched by the full BSERD technique. Next, the cells’ V\textsubscript{T} are compared (verified) against a new lower mini-programming margin reference voltage, V\textsubscript{RI+1} - V\textsubscript{MPUI}, where I corresponds uniquely to the level represented by the latched N binary bits. This verify sensing differs from a read sensing during BSERD since the verify does not change the status of the previously latched N binary bits corresponding to the V\textsubscript{T} stored in the cell. Therefore, any cells now verified with V\textsubscript{T} below V\textsubscript{RI+1} - V\textsubscript{MPUI} are within the lower mini-programming margin guard band, V\textsubscript{MPLI}, defined by the values V\textsubscript{RI} and V\textsubscript{RI+1} - V\textsubscript{MPUI}.

Next, a mini-programming pulse of appropriate voltage and duration is selectively applied only to cells which were previously determined to have V\textsubscript{T} within the lower V\textsubscript{MPLI} guard band by the previous verify step, within the selected page. The cells with V\textsubscript{T} above the V\textsubscript{MPLI} guard band are excluded from the mini-programming pulse. After the first mini-programming pulse, another verify operation is executed. If any cells are verified within the lower V\textsubscript{MPLI} guard band, another mini-programming pulse is selectively applied, again, only to those cells within the V\textsubscript{MPLI} band. The verify/mini-programming pulse sequence is repeated until time T\textsubscript{RESP}. In this embodiment, when all the cells within the page are verified to have their V\textsubscript{T} above the V\textsubscript{MPLI} guard band, or until a predetermined maximum number of mini-programming pulses have been applied and a system error
flag is set. The resulting cell $V_T$ distribution may now appear as shown in the second distribution shown in Fig. 12 at $T_{RES}$. 

**[0112]** In this example, the restore operation was originally triggered by the cells which were previously found to be within the lower read margin guard band, $V_{MRLI}$, such as cells A and B in Fig. 12. Cells with $V_T$ within $V_{MPUI}$ have also been mini-programmed, resulting in a lighter $V_T$ distribution for the entire page in addition to correcting the originally flagged cells within $V_{MRLI}$. The resulting tighter page-wide $V_T$ distribution improves system performance by eliminating the need for future restore operations on this page.

**[0113]** To detect a $V_T$ overshoot condition, an additional verify step is performed after the verify/mini-programming pulse sequence described above. Alternatively, the additional verify step can be done after each mini-programming pulse but would take longer to accomplish. This additional step verifies the $V_T$ of all the cells in the selected mini-programmed page against a second new mini-programming upper margin reference voltage, $V_{R(I+1)} - V_{MPUI}$, where I, again, corresponds uniquely to the same level represented by the latched N binary bits. Any cells now verified with their $V_T$ above $V_{R(I+1)} - V_{MPUI}$ are within the upper mini-programming margin guard band, $V_{MPUI}$, defined by the values $V_{R(I+1)}$ and $V_{R(I+1)} - V_{MPUI}$. A $V_T$ overshoot system conditional flag is set if any cell $V_T$ is detected within the $V_{MPUI}$ guard band. Later, a restore operation checks if the $V_T$ overshoot system conditional flag is set and determines if a $V_T$ overshoot recovery operation is necessary. The $V_T$ overshoot recovery is a sequence of operations including the mini-erase operation described in more detail below in the mini-erase section of this invention.

**Multilevel Memory Mini-Programming System**

**[0114]** Fig. 7 shows the block diagram of a preferred embodiment of the present invention and has already been described along with the BSERD sensing technique. The mini-programming verify sensing operations differ from the read sensing operation in that the PROGRAM ERASE READ RESTOR SEQUENCER block 26 instructs the $\Delta VU \DeltaVL$ GEN block 39 via the PRGRDB bus 37 to generate the $V_{MPUI}$ and $V_{MPLI}$ voltages, instead of the $V_{MRLI}$ and $V_{MRUI}$ voltages, from the reference voltage bus RVBUS 42. Also, the VX VY GEN block 38 is instructed to generate the $V_{RI}$ and $V_{R(I+1)}$ Voltages, where I corresponds to the level represented by the N binary bits previously latched in the DATAT[0-3] LATCHES block 48 within each Y-DRIVER 33 during the previous BSERD step.

**[0115]** After the verify operation sequence, the STPDIR LATCH 52 and RES LATCH 53 within each Y-DRIVER 33 are set such that the STPDIR line 47 is high if the cell $V_T$ verified within the $V_{MPLI}$ guard band. A high signal on the STPDIR line 47 turns on the high voltage HVS W switch 56 which passes the high voltage programming pulse from the PRGHV line 24 to the MLLINE line 30. The Y-MUX multiplexer 32 within each Y-DRIVER 33 passes the high voltage pulse to the appropriate cell 20 of the selected page in the CELL ARRAY 21 to be programmed as selected by the Y-DEC decoder 18. The STPDIR line 47 is low if the cell $V_T$ is verified outside the $V_{MPLI}$ guard band. A low signal on the STPDIR line 47 turns off the high voltage HVS W switch 56. This isolates the high voltage programming pulse from the ML-LINE line 30.

**[0116]** If any cell $V_T$ is verified within the upper $V_{MPUI}$ guard band, the $V_T$ overshoot conditional flag is set within the PROGRAM ERASE READ RESTORE SEQUENCER block 26. The address location and corresponding $V_T$ overshoot conditional flag is stored in the REGISTERS & ADDRESS BUFFER block 60. Each time the verify/mini-programming pulse sequence is done, a counter is incremented in the PROGRAM ERASE READ RESTORE SEQUENCER block 26. If that counter exceeds a predetermined maximum number, the mini-programming for that page is stopped and a system error flag in the PROGRAM ERASE READ RESTORE SEQUENCER block 26 is set, along with the address, in the REGISTERS & ADDRESS BUFFER block 60.

**III. Mini-Erasing**

**General Description of Mini-Erasing Operations**

**[0117]** A multilevel erasing technique called mini-erasing is described in this section as part of the present invention. Unlike previous erasing techniques which destroy the data stored in the cells, the mini-erasing technique leaves intact the data originally stored in the cell. In the prior art, the erase operation shifts the $V_T$ of all the cells in the selected erase block all the way over to one end of the useable cell $V_T$ range, $V_F$, as shown in Figs. 3A-3C, and labeled $V_{LERASE}$. This erasing shifts the selected cell's $V_T$ out of one of the other $V_L$ levels and into the $V_{LERASE}$ range by moving a great deal of charge through the oxides surrounding the floating gate of the memory cell.

**[0118]** In contrast, mini-erasing moves only a small amount of charge. As in the case of mini-programming, the cell's $V_T$ is shifted only a small amount and only within the limits of that cell's original $V_L$ level. The deleterious effects on P/E cycling endurance caused by the prior art erasing are avoided by mini-erasing. Only a small amount of charge is moved. Also, only that amount of charge necessary to correct the previously determined $V_T$ drift is moved. The mini-erasing operations require less time than erasing to the fully erased condition as is done by the prior art. Additionally, mini-erasing affects the selected cells' $V_T$ only slightly, and hence the resulting erase disturb effect on other unselected cells is negligible. The mini-erasing operation can shift the $V_T$ a
small amount within a \( V_{L} \) level in the opposite direction of programming and thus perform one of the functions necessary to do the desired \( V_{T} \) restoration to enable reliable multilevel data storage.

[0119] The mini-erasing technique works together with the mini-programming technique previously described, to overcome a problem associated with large erase block size. A large number of cells which may be connected within the same erase block results in a wider distribution of cell \( V_{T} \) after an erase pulse than the distribution obtained after a programming pulse is applied to a smaller number of cells within a programming page. The mini-erase technique uses a combination of mini-program and mini-erase operations to reduce the width of the cell \( V_{T} \) distribution within each \( V_{L} \) level for all the cells within an erase block.

[0120] The mini-erase operation utilizes the band sensing capabilities of the present invention, similar to those described in the multilevel digital sensing section, to verify that each cell’s \( V_{T} \) is shifted into the desired band of the \( V_{L} \) range ensuring adequate margin guard bands to prevent ambiguous sensing at both ends of that \( V_{L} \) range. This is accomplished by multiple reference voltages during the mini-erase verify. Three additional reference voltages at \( V_{R_{L}}+V_{MPELI}, V_{R_{L}}+V_{MELI}, \) and \( V_{R_{L}(I-1)}-V_{MEUI} \), which in conjunction with \( V_{R_{L}} \) and \( V_{R_{L}(I+1)} \) define three mini-erasing margin guard bands, \( V_{MPELI}, V_{MELI}, \) and \( V_{MEUI} \), within each of the \( V_{L} \) levels as shown in Fig. 13A and 13B. \( V_{MPELI}, V_{MELI}, \) and \( V_{MEUI} \) may, or may not, be equal to and may, or may not, be greater than the previously described \( V_{MRUI} \) and \( V_{MRUI} \) margin guard bands or \( V_{MPUI} \) margin guard bands for \( V_{MPUI} \) and \( V_{MPUI} \), mini-programming margin guard bands. The \( V_{MPELI}, V_{MELI}, \) and \( V_{MEUI} \) margin guard bands within each of the \( V_{L} \) levels are chosen such that the resulting \( V_{T} \) distribution after mini-erasing is optimized within a desired band which is narrower than the \( V_{R} \) range.

[0121] The optimization may require the \( V_{MPELI}, V_{MELI}, \) and \( V_{MEUI} \) bands to be unequal to each other and may require different band values for each of the \( V_{L} \) levels or even arbitrary band values dictated by the memory cell array process technology. Alternatively, the margin guard bands in each level may be of equal value. Mini-erasing is flexible enough to accommodate the optimization for a wide variety of memory processing technologies.

[0122] The mini-erase verify technique of this invention contrasts to the prior art which verifies the cell \( V_{T} \) only relative to a single voltage reference value and which relies on statistical process control to prevent the cell \( V_{T} \) from undershooting the desired \( V_{T} \) band. \( V_{T} \) undershoot (the erase analogue of \( V_{T} \) overshoot during programming) is a possibility during any erasing operation and may occur due to the previously mentioned rogue cell effects from trapped charges in the oxides surrounding the floating gate of the memory cell or other process induced defects. In addition, mini-erasing minimizes charge trapping and the possibility of creating a rogue cell by moving only small amounts of charge through the gate oxides.

[0123] The mini-erase verify operation detects if the \( V_{T} \) of any of the cells in the erase block has been shifted into one of the \( V_{MELI} \) guard bands and stops further mini-erasing so that the mini-programming technique can later reestablish the desired \( V_{T} \) within that level. This recovery capability for \( V_{T} \) undershoot is not present in the prior art without first completing a full erase and reprogramming operation. Using the restore operation of this invention, full erase and reprogramming is avoided during a \( V_{T} \) undershoot recovery and the P/E endurance lifetime of the chip is enhanced.

[0124] The direction of cell \( V_{T} \) shift during erase is specific to a particular silicon technology. Mini-erasing accommodates either direction of erase \( V_{T} \) shift and is applicable to a wide range of memory technologies. For simplicity, the mini-erasing operation is described with accompanying examples and figures to illustrate the case where the erase operation causes the cell \( V_{T} \) to decrease. The opposite case of increasing \( V_{T} \) with erase can be analogously handled by those skilled in the art. Thus the particular examples and methods should not be limited to a particular memory cell technology with respect to erase polarity.

**Mini-Erasing Methodology**

[0125] The mini-erasing technique is best described by considering a plurality of cells which are connected in the same erase block. A practical multilevel memory system should use an erase block memory cell architecture which reduces cell area so that many cells are erased simultaneously. It is necessary to comprehend the behavior of all the cells at least within such an erase block, if not the whole chip, to observe all the possible ramifications of any form of erasing.

[0126] Figs. 13A and 13B show the \( V_{T} \) distribution of a plurality of cells which are within the same level, \( V_{L} \), at six different times; \( T_{RESR}, T_{RESP1}, T_{RESP2}, T_{RESE1}, T_{RESE2} \), and \( T_{RESP} \). In this example, the \( V_{T} \) distribution at time \( T_{RESP} \) shows the condition of the cells after the cells have been read using the previously described BSERD technique. The BSERD technique detects two cells with \( V_{T} \) values at A and B within the \( V_{MRUI} \) guard band and two more cells with \( V_{T} \) values at C and D within the \( V_{MRUI} \) guard band. The restore conditional flags are set to indicate a restore operation is necessary. The BSERD technique has also determined that the restore operation may be accomplished with a mini-erase, since cells were detected within the \( V_{MRUI} \) guard band and a restore low condition remains set for this block address.

[0127] In one embodiment of the mini-erase technique, the integrated circuit executes a series of operations including mini-program and mini-erase. The mini-program operations have been previously described. The mini-erase technique uses two versions of the mini-programming operation. One version is identical to the
mini-programming described before. The previously described \( V_{\text{MPLI}} \) and \( V_{\text{MPUI}} \) margin guard bands help perform the final \( V_T \) adjustment at the end of the mini-erase sequence.

[0128] The second version of mini-programming operation differs from the first version by the use of a different (pre-erase) programming guard band margin, \( V_{\text{MPELI}} \). The mini-programming operation with \( V_{\text{MPELI}} \) is executed before each mini-erase operation. Several mini-erase operations may be performed up to a predetermined maximum number of mini-erase operations. The sequence of operations in one embodiment of the mini-erase technique is described below.

[0129] In one embodiment, the integrated circuit conveniently executes a restore operation with mini-erase at some later time to restore the \( V_T \) of the erase block of cells which have previously been flagged for restore with a mini-erase. All the cells within the erase block to be mini-erased are first mini-programmed such that the \( V_T \) of all the cells within the erase block are above the \( V_{\text{MPELI}} \) margin guard band for each of the \( I \) levels. This pre-erase mini-program is performed, in turn, for each page within the block using the mini-programming technique with the lower guard band verify value set to \( V_{R(I+1)}+V_{\text{MPERI}} \) instead of \( V_{R(I+1)}+V_{\text{MPLI}} \). The pre-erase mini-program is first completed for all the pages in the block at time \( T_{\text{RESPE1}} \). An example of the resulting cell \( V_T \) distribution within each \( V_L \) level and for all cells within the erase block appears at time \( T_{\text{RESPE1}} \) in Fig. 13A. With the mini-program operation first, cells within the \( V_{\text{MPELI}} \) band too close to a \( V_R \) value, such as cells A and B at time \( T_{\text{RESL}} \), might be shifted into the adjacent \( V_L \) range during a mini-erase operation. This would result in data loss. The pre-erase mini-programming prevents data loss that might occur if a cell \( V_T \) in the erase block is within the \( V_{\text{MPELI}} \) band.

[0130] Next, a mini-erase pulse is applied to all the cells in the erase block. The mini-erase pulse shifts the \( V_T \) of all the cells in the erase block towards the erase direction (the direction opposite to that of a programming operation). A mini-erase verify operation is next done for every cell in the erase block (consisting of multiple pages) so the first page in the erase block is selected for the mini-erase block verify operation. Tighter process control of the erase characteristics of all the cells in the erase block is relied upon to prevent data loss. Notice that Fig. 13A shows a long tail at the end of the cell \( V_T \) distribution, on the erase direction side. This is due to some cells mini-erasing faster than others in the large population of cells in the erase block. The erase block's \( V_T \) distribution has now been shifted in the erase direction as far as it can without \( V_T \) undershoot and without any data loss. The cells (with \( V_T \) at C and D in Fig. 13A) which originally triggered the mini-erase have now been shifted in the desired direction by an amount shown as \( V_{\text{DELE1}} \) in Fig. 13A.

[0133] An alternative embodiment which is faster uses a single mini-erase pulse without the \( V_{\text{MPELI}} \) guard band verify operation. Tighter process control of the erase characteristics of all the cells in the erase block is relied upon to prevent data loss. Notice that Fig. 13A shows a long tail at the end of the cell \( V_T \) distribution, on the erase direction side. This is due to some cells mini-erasing faster than others in the large population of cells in the erase block. The erase block's \( V_T \) distribution has now been shifted in the erase direction as far as it can without \( V_T \) undershoot and without any data loss. The cells (with \( V_T \) at C and D in Fig. 13A) which originally triggered the mini-erase have now been shifted in the desired direction by an amount shown as \( V_{\text{DELE1}} \) in Fig. 13A.

[0134] The next verify operation checks for any cells that exhibited \( V_T \) overshoot behavior during the previous pre-erase mini-programming operation, such as cell E in Fig. 14A at time \( T_{\text{RESPE1}} \), and checks if more mini-erasing is required. This mini-erase verify operation is again performed for every cell in the erase block so the first page in the erase block is selected for the mini-erase verify operation. The cells are next read by a simplified BSERD technique with only the data bit detection subcycle, i.e., without the restore detection subcycle, to latch the N binary bits corresponding to the \( V_T \) stored in the cell. The cells' \( V_T \) in the selected page of the erase block are then compared (verified) against a new lower mini-erasing margin reference voltage, \( V_{R(I+1)}+V_{\text{MELI}} \) where \( I \) corresponds uniquely to the level represented by the latched N binary bits.

[0135] This verify sensing operation again differs from a read sensing operation described above since the verify operation does not change the status of the previously latched N binary bits corresponding to the \( V_T \) stored in the cell. Therefore, any cells now verified with \( V_T \) above \( V_{R(I+1)}+V_{\text{MELI}} \) are within the upper mini-erasing guard band, \( V_{\text{MELI}} \), defined by the values \( V_{R(I+1)} \) and
If no cells are verified within the $V_{MEUI}$ guard band of the first page, the next page is selected and similarly read and verified. If any cells are verified within the $V_{MEUI}$ guard band in the whole erase block, the above sequence of pre-erase mini-programming/mini-erasing/verify operations is applied again to the erase block unless a predetermined maximum number of such sequences has been done and resulting error flag is set.

If any cells during the verify are detected within the $V_{MEUI}$ guard band, the address location of those cells are compared to the previously stored address locations of the cells which originally triggered the mini-erase, such as C and D in Fig. 13A. If a new address is detected, such as E in Fig. 14A, the $V_T$ overshoot conditional flag is set and the page address is stored. The example shown in Figs. 13A and 13B requires two of the above sequences since the two cells at C and D were still within the $V_{MEUI}$ guard band at time $T_{RESE1}$. In this example, a second pre-erase mini-programming is completed at time $T_{RESP2}$ followed by a second series of mini-erase pulses resulting in shifting the $V_T$ of cell C by the amount $V_{DELE2}$ at time $T_{RESE2}$. The resulting cell $V_T$ distribution is shown in Fig. 13B.

The final mini-programming operation begins after no cells are found within the $V_{MEUI}$ guard band during the above erase block verification at time $T_{RESE}$ as shown in the example in Fig. 13B. The final mini-programming operation is performed in the same manner as described in the mini-programming section, except that each page within the erase block is mini-programmed in turn. During the final mini-programming operation, if any cell $V_T$ is detected within the upper $V_{MPUI}$ guard band, a $V_T$ overshoot system conditional flag is set and the page address is stored. The final mini-programming operation restores the entire erase block’s cell $V_T$ distribution for each $V_L$ level between the $V_{MPUL}$ and $V_{MPUI}$ guard bands, as shown in the example in Fig. 13B at time $T_{RESP}$. The $V_T$ overshoot system conditional flag is checked before any mini-programming operation is begun on any page. Whenever a page to be mini-programmed is encountered with this flag set, the mini-programming pulse sequence is altered to produce a smaller than normal incremental cell $V_T$ shift with each pulse, called micro-programming. The cell $V_T$ distribution within each level in the selected page after the verify/micro-programming pulse sequence is completed is tighter than after a verify/micro-programming pulse sequence. Microprogramming a flagged page takes a longer time than micro-programming but overall system performance is optimized, since only the requisite flagged pages are micro-programmed.

In an alternate micro-programming operation, the $V_{MPUL}$, $V_{MPUI}$, and $V_{MPUI}$ guard bands used during mini-programming are replaced with $V_{CPELI}$, $V_{CPUL}$, and $V_{CPUI}$ guard bands such that the resulting cell $V_T$ distribution within each $V_L$ range is optimized (see Fig. 14A and 14B). As in the mini-programming technique, if the number of verify/micro-programming pulse sequences exceeds a predetermined number, the micro-programming is stopped and a system error flag is set. The $V_T$ overshoot system conditional flag is reset after the flagged page is successfully micro-programmed. Setting the $V_T$ overshoot system conditional flag during a restore without mini-erase simply starts the mini-erase operations for that erase block immediately or, alternatively, at a later convenient time. The mini-erase technique used in conjunction with micro-programming serves as the $V_T$ overshoot recovery method.

Multilevel Memory Mini-Erasing System

The PROGRAM ERASE READ RESTORE SEQUENCER block 26 together with the BINARY SEARCH READ SEQUENCER 43 of Fig. 7 perform all the necessary control and sequencing of events to perform the mini-erase operations. Fig. 11 shows a particular interface to the cell array 21.

For example, the BLOCK0 block 83 is selected for a mini-erase operation because a restore low condition was previously set for an address in this erase block. In Fig. 13A at time $T_{RESP}$, the PROGRAM ERASE READ RESTORE SEQUENCER 26 begins a mini-erase operation based on the restore conditional flags. As shown in Fig. 11, there are J cells per page, there are S pages per row, and there are T rows per block. Therefore, there are $(S \times T)$ pages per block and $(S \times T \times J)$ cells per block. Using the Fig. 13A example, a pre-erase mini-program operation is first performed on all the pages of the BLOCK0 block 83. For each page of the BLOCK0 block 83, a simplified BSERD technique without the restore detection sub-cycle is first performed to simultaneously read the 4 bits of data from each cell 20 for all J cells in the selected page. After the read operation is complete, the DATA[0-3] LATCHES block 48 within each Y-DRIVER 33 contains the individual 4 bits per cell data for the respective J cells of the selected page.

Next, the PRGRDB bus 37 is set by the PROGRAM ERASE READ RESTORE SEQUENCER block 26. Based on the output from the DATA[0-3] LATCHES block 48 for each Y-DRIVER 33, the appropriate $V_{RI}$ voltage is generated as the VX output and the $V_{RI(i+1)}$ voltage is generated as the VY output from the VX VX GEN block 38. The $\Delta VU$, $\Delta VL$ GEN block 39 generates $V_{MPUL}$ as $\Delta VL$ output. The VCRFGEN block 40 generates the voltage $(V_{RI} + V_{MPUL})$ on the VCRF line 45 during all of the pre-erase mini-programming operations. A pre-erase mini-programming operation is now performed with a repeated sequence of mini-program pulses and verify operations until all the J cells within the selected page have $V_T$ above the $(V_{RI} + V_{MPUL})$ voltage. The PROGRAM ERASE READ RESTORE SEQUENCER block 26 and the BINARY SEARCH READ SEQUENCER block 43 then continue pre-erase mini-pro-
gramming of all the S pages in the first row WL₀₀ line 85. This continues until all T rows with S pages have been pre-erase mini-programmed with all the cell VT's programmed above the (VRᵢ+VMPELI) voltage. This completes the pre-erase mini-programming of the selected BLOCK0 block 83. A typical distribution of the cell VT for the BLOCK0 is shown in Fig. 13A at time TRESP₁.

[0143] The PRG-ERS bus 22 is forced to set the BLOCK-DEC block 19, the X-DEC block 17 and the Y-DEC block 18 to allow an erase operation on the selected memory block or sector. Next, a mini-erase pulse is applied to all the cells 20 in the selected BLOCK0 block 83. The mini-erase pulse is generated by the HVGEN block 27 and passed via the HVOUT bus 25.

[0144] After the application of the mini-erase pulse, a verify operation is performed on all the cells 20 in the selected BLOCK0 block 83. The BSERD technique without the restore detection sub-cycle latches the 4-bits of data from each cell 20 in a page in the DATA[0-3] LATCHES block 48 within each Y-DRIVER 33. Based on the output from the DATA[0-3] LATCHES block 48 within each Y-DRIVER 33, the VX VY GEN block 38 generates the VRᵢ voltage as VX output and the VRᵢ₊₁ voltage as VY output. The ΔVU  ΔVL GEN block 39 generates the VMELI voltage as ΔVL output.

[0145] During all of the mini-erase verify operations, the VCRFGEN block 40 outputs the voltage (VRᵢ+VMELI) on the VCRF line 45, which is uniquely defined by the output from the DATA[0-3] LATCHES block 48 for each Y-DRIVER 33. The PROGRAM ERASE READ RESTORE SEQUENCER block 26 and the BINARY SEARCH READ SEQUENCER block 43 then perform the mini-erase verification operations on all S pages of all T rows of the selected BLOCK0 block 83. The mini-erase verification compares the voltage on the VMEM line 31 with the voltage on the VCRF line, (VRᵢ+VMELI), for each Y-DRIVER 33. If during the mini-erase verification operation the VMEM line 31 is lower than the VCRF line 45, a conditional flag is set in the PROGRAM ERASE READ RESTORE SEQUENCER block 26. If the conditional flag is not set, then another mini-erase pulse and mini-erase verification operation is performed on the selected BLOCK0 block 83. Whenever a conditional flag is set or the maximum count of mini-erase pulse operations is reached, the PROGRAM ERASE READ RESTORE SEQUENCER block 26 stops the mini-erase/verification operations. Fig. 13A shows at time TRESP₁ the distribution of the cell VT after a conditional flag has been set.

[0146] The overshoot cell verification operation is next performed on all the cells of the selected BLOCK0 block 83. The PROGRAM ERASE READ RESTORE SEQUENCER block 26 and the BINARY SEARCH READ SEQUENCER block 43 perform the over-shoot cell verification on all S pages of all T rows of the selected BLOCK0. During the entire over-shoot verification operation, the voltage on the VCRF line 45 is set to VRᵢ₊₁ VMELI, which uniquely depends on the output of the DATA[0-3] LATCHES block 48 within each Y-DRIVER 33. For each page, and for each Y-DRIVER 33, the over-shoot verification continues to the next page if the VMEM line 31 is lower than the VCRF line 45. If any time for a selected page, the VMEM line 31 is higher than the VCRF line 45 for any Y-DRIVER 33, then the address of those cells 20 are stored in the REGISTERS & ADDRESS BUFFER block 60.

[0147] The addresses of the overshoot conditionally flagged cells are matched with the addresses of the cells which originally triggered the mini-erase conditional flag. If an address mismatch is found, then the overshoot conditional flag is set for that page along with the page address in the REGISTERS & ADDRESS BUFFER block 60. If the addresses match, another pre-erase mini-programming operation is performed. Fig. 13A and 13B show the distribution of the cell VT at time TRESP₂ after the second pre-erase mini-programming operation.

[0148] Then another mini-erase pulse is applied and mini-erase verification performed. Fig. 13B at time TRESP₂ shows the distribution of the cell VT after the second mini-erase/verification operation. If the overshoot conditional flag is not set after the VMELI guard band verify, the final mini-programming operation of all the pages in the selected BLOCK0 block 83 is performed. At the beginning of the final mini-programming operation for each page, the PROGRAM ERASE READ RESTORE SEQUENCER block 26 performs a VT overshoot conditional flag check for that page in the REGISTERS & ADDRESS BUFFER block 60. If no flag is found, a regular mini-programming operation is performed. Fig. 13B at time TRESP₂ shows the distribution of the cell VT after the final regular mini-programming operation is performed. Whenever a conditional flag is found for a page, a micro-programming operation is performed on the page. It is preferred that the micro-programming operation be identical to the mini-programming operation with the following exceptions; 1) the PROGRAM ERASE READ RESTORE SEQUENCER block 26 instructs the ΔVU ΔVL GEN block 39 via the PRGRDB bus 37 to generate the VPELI, VPELI, and VPELI voltages, instead of the VMPELI, VMPUI and VMPUI voltages from the analog voltage bus RVBUS 42, and 2) the PROGRAM ERASE READ RESTORE SEQUENCER block 26 instructs the HVGEN block 27 via the HVCTRL 23 bus to alter the mini-programming pulse to a micro-programming pulse so that the cells' VT shifts less than during the mini-programming operation.

IV. Programming

General Description of Programming Operations

[0149] The described integrated circuit also operates with a novel multilevel programming technique. This programming technique differs from the previously de-
scribed mini-programming technique. This programming stores new data into a previously erased page while mini-programming restores the existing data without altering the previously stored data. Compared to mini-programming, the programming technique moves greater amounts of charge through the oxides surrounding the floating gate and may shift the cell V_T by a larger amount. Unlike previous programming techniques which only shifts the cell V_T in one (programming) direction, the programming technique of the present invention shifts the cell V_T in both programming and erase directions.

Such previous multilevel memory programming requires a very tight cell V_T distribution. Very wide margins within each level after programming allow for future V_T drift which will occur over the entire lifetime of the memory chip. Fig. 15A shows an example of the initially tight distribution of cell V_T within a V_L level immediately after prior art programming using a single margin value at PV at time T_F. Also shown in the same figure is the distribution after a long period during which V_T drift effects have substantially widened the initially tight distribution to the point of ambiguous sensing failure at time T_F.

In contrast, the present invention allows the use of a very wide cell V_T distribution within each level resulting from the initial programming step to store the data reliably for a short time (see Fig. 15B at time T_P) until the memory can execute a restore operation at a near future time (as shown at T_RESP in Fig. 15B). Additionally, the margins maintained and described by the cell V_T restore techniques of this invention (see Fig. 15B at T_F) are not as wide as the margins required during programming without the use of this invention. The tight programming V_T distribution required in the prior techniques greatly lengthens the programming time since many verify/program pulse sequences are needed to shift the V_T in small increments before verifying successfully to the desired reference voltage. In most system applications, longer programming time is highly undesirable from the external system’s point of view. On the other hand, the present invention reduces programming time for multilevel storage by enabling the external system’s perceived programming algorithm to employ a wide cell V_T distribution within each level.

One embodiment of this invention differs from the previous techniques by requiring the final operation that sets the cell V_T for all V_LI levels be a programming operation and not an erase operation as is used in the prior art for one of the levels at either end of the V_F range.

This invention divides the programming cycle into two parts. The first part initially stores the data quickly into the cells with only small margins about the V_R reference values for each level. The small margins are adequate to reliably store the data temporarily before the long term V_T drift effects cause data loss. This first part optimizes the programming time to be as short as possible from the external system’s point of view. The total programming time for the external system is only the time required to do the data storage during the first part. The first part performs most of the charge movement through the gate oxides and most of the cells’ V_T shift.

The second part is identical to the previously described restore with mini-erase operations. Since data has already been stored in the cells in the first part, the second part can be accomplished autonomously of the external system. The second part can be performed after the programming period perceived by the external system is over and during periods when the memory chip would otherwise be idle. The second part of the programming cycle establishes the wider margins about the V_R values which are required for optimized system operation and long term reliability. This second part of the programming cycle takes longer than the first part.

In one embodiment of the present invention, the second part occurs immediately after the first. In another embodiment, the slower second part is postponed to a later time soon after the first part but in such a way that the programming time seen by the external system is greatly reduced.

The first part of the program operation utilizes the band sensing capabilities of the present invention, similar to those described in the multilevel digital sensing section. Each cell’s V_T is verified to have shifted into the desired band of the V_L range to ensure adequate margin guard bands to prevent ambiguous sensing at both ends of that V_L range. Multiple reference voltages are used during the programming verify operations. The first part of the program technique uses reference voltages at V_Ri+V_PLI and V_R(i+1)-V_PUI which, in conjunction with V_Ri and V_R(i+1), define two new lower and upper programming margin guard bands, V_PLI and V_PUI within each of the V_LI levels as shown in Fig. 15B. V_PLI and V_PUI may, or may not, be equal to and may, or may not, be greater than the previously described V_MRLI and V_MRULI, or V_MPLI and V_MPUI, mini-programming, or V_IPELI, V_MELI, and V_MUI, mini-erase, or V_CPELI, V_CLPLI, and V_CPUI, micro-programming guard bands. The V_PLI and V_PUI guard bands within each of the V_LI levels are selected such that the result V_T distribution after the first part of programming is optimized within a desired band which is narrower than the V_L range. The optimization may require the V_PLI and V_PUI bands to be unequal to each other and may require different band values for each of the I levels, or even arbitrary band values, dictated by the memory cell array process technology. Alternatively, the bands in each level may be of equal value. The programming technique is flexible enough to accommodate the optimization for a wide variety of memory processing technologies.

The verify technique of this invention contrasts with previous programming techniques which verifies the cell V_T only relative to a single reference voltage per level. In the prior art, a cell may be programmed such
that the $V_T$ is so close to a $V_{RI(I+1)}$ reference value that the previously described ambiguous sensing problem is created. By verifying the programmed data is within an optimized band using two additional verify reference voltages per level, the present invention prevents the ambiguous data sensing problem which may occur with previous techniques.

**Programming Methodology**

[0158] First, the new data is latched into the page to be programmed and converted into one of the $2^N V_L$ levels for each cell. In one embodiment of the present invention, the integrated circuit executes the first part of the programming sequence. The cells’ $V_T$ are compared (verified) against a new lower programming margin reference voltage, $V_{RI} V_{PLI}$, where $I$ corresponds uniquely to the level represented by the latched $N$ binary bits. This verify sensing technique differs from the read and verify sensing previously described and differs from the prior art in that the value of the $V_{PLI}$ guard band, defined by the values $V_R$ and $V_{RI}+V_{PLI}$, is much smaller than any of the former could allow. This invention allows such a small $V_{PLI}$ programming margin guard band since: 1) the second part of the programming sequence increases the margin for long term $V_T$ stabilization and 2) the second part is performed soon after the first part is completed, before appreciable $V_T$ drift can occur. This verify sensing differs from the read sensing operations (see BSERD technique previously described) since the verify does not change the status of the previously latched $N$ binary bits corresponding to the $V_T$ to be stored in the cell.

[0159] Next, a program pulse of appropriate voltage and duration is applied to only selected cells in the selected page. The programming pulses are selectively applied only to cells which were previously determined to have $V_T$ below the $V_{RI}+V_{PLI}$ voltage by the previous verify step. The cells with $V_T$ above the $V_{RI} V_{PLI}$ voltage are excluded from programming. After the first programming pulse, another verify operation is executed. If any cells are verified below the $V_{RI} V_{PLI}$ voltage, another programming pulse is selectively applied, again, only to those cells with $V_T$ below the $V_{RI}+V_{PLI}$ voltage.

[0160] The programming technique is best described by considering a plurality of cells which are connected in the same page. Fig. 16A and 16B show the $V_T$ distribution of a plurality of cells in the memory which are within the same level, $V_L$, at six different times; $T_P$, $T_{RESP1}$, $T_{RESPE1}$, $T_{RESE1}$, $T_{RESPE2}$, $T_{RESPE2}$, and $T_{RESP}$. The verify/programming pulse sequence is repeated until time $T_P$, show in Fig. 16A, when all the cells within the page are verified to have their $V_T$ above the $V_{RI} V_{PLI}$ voltage or until a predetermined maximum number of pulses have been applied and a system error flag is set.

[0161] Once the first page is successfully programmed, the next erased page to be programmed is selected and the procedure repeated through the last page. Pages to be programmed must reside in previously erased blocks or sectors. The address of each erase block upon which the first part programming operation was performed is stored so that the second part of the programming operation can be performed later. The conclusion of this first part programming procedure determines the programming time seen by the external system. Significant multilevel programming time reduction is achieved by this invention since the programming pulses used in this part of programming do not require a cell $V_T$ distribution within each $V_L$ level as tight as required in the prior art.

[0162] This invention adds an additional verify step after the verify/programming pulse sequence described above. Alternatively, the new verify step can be done after each programming pulse. Of course, this variation would take more time. The additional step verifies the $V_T$ of all the cells in the selected page just programmed against a second new programming margin reference voltage, $V_{RI(I+1)} V_{PUI}$, where $I$, again, corresponds uniquely to the same level represented by the latched $N$ binary bits (see Fig. 16A). Any cells now verified with their $V_T$ above the voltage, $V_{RI(I+1)} V_{PUI}$, are over-programmed and a system error flag is set. The verify operation ensures the $V_T$ of the programmed cells that pass verification are set within a band which is narrower than the $V_L$ range and establishes both upper and lower programming margins, $V_{PUL}$ and $V_{PUI}$. The possibility of ambiguous sensing when a bit programs too close to the $V_{RI(I+1)} V_{PUI}$ voltage reference value is avoided. In contrast, prior art techniques established only a lower margin.

[0163] At a convenient time after the first part programming sequence described above, the integrated circuit performs restore with mini-erase operations sequentially on all the erase blocks which had been just programmed. The cell $V_T$ distributions shown in Fig. 13A at time $T_{RESP}$ before a restore with mini-erase and Fig. 16A at time $T_P$ are illustrative of the result of these operations.

[0164] Thus, the second part of the programming sequence is accomplished entirely by the previously described restore with mini-erase technique. Fig. 16A and 16B show these steps and correspond to the same steps shown in Fig. 13A and 13B for the restore with mini-erase operations after times $T_{RESP}$, $T_{RESPE1}$, $T_{RESE1}$, $T_{RESPE2}$, $T_{RESPE2}$, and $T_{RESP}$. The full guard bands, $V_{MPUI}$ and $V_{MPUL}$, have been restored before the cell $V_T$ stored with minimized guard bands in the first part of programming has had time to drift. The resulting cell $V_T$ distribution may now appear as shown in the last distribution shown in Fig. 16B at time $T_{RESP}$.

[0165] Alternatively, the integrated circuit conveniently executes the second part of the programming operation (a restore with mini-erase) at a later time to restore the $V_T$ of the erase block of cells previously flagged for the second part programming operation.

[0166] In another variation, the second part program-
ming on the previously first part programmed erase blocks is performed immediately following that first part programming step. In either case, the second part programming operation is performed autonomously since all the data is already stored in the memory cells. Thus, the autonomously performed second part programming sequence of this invention does not add to the programming time perceived by the external system.

Multilevel Memory Programming System

The preferred embodiment for programming has already been described along with the BSERD and the mini-programming technique and the mini-erasing technique. First, data from an external source is loaded into the DATA[0-3] LATCHES block 48 within each Y-DRIVER 33 through the DATA, ADDR & CNTRL SYSTEM INTERFACE block 10 in Fig. 7. The address location for storing the data is also provided. Data is programmed into an already erased block. The first part of programming is very similar to the mini-programming technique previously described. This time the PROGRAM ERASE READ RESTORE SEQUENCER block 26 instructs the ΔVU ΔVL GEN block 39 via the PRGRDB bus 37 to generate the VPUI and VPPLI reference voltages, instead of the VMPUI and VMPPLI voltages from the reference voltage bus RVBUS 42. The VX VY GEN block 38 is instructed to generate the VRi and VRi+1 reference voltages, where i corresponds to the level represented by the N binary bits previously loaded and latched in the DATA[0-3] LATCHES block 48 within each Y-DRIVER 33. The program/verify and overshoot verify operations are similar to those used in the previously described mini-programming technique. The second part of the programming technique is exactly the same as the mini-erasing technique described previously.

V. Erasing

General Description During Erasing Operations

The integrated circuit also operates with a multilevel erasing technique which is different from the mini-erasing technique described earlier. Erasing destroys the previously stored data in the erased block or sector, while mini-erasing restores the existing data without altering the previously stored data. Also, erasing moves greater amounts of charge through the oxides surrounding the floating gate and may shift the cell VT by a larger amount than that which occurs during mini-erasing.

Unlike previous erasure techniques which only shift the cell VT in one (erasing) direction, the erasing technique of the present invention shifts the cell VT in both programming and erase directions. Prior art erasing for multilevel memory storage creates a very wide cell VT distribution for all the cells in the erase block. The wide erased cell VT distribution, VLE, has been used in previous techniques for one of the 2N VT ranges for multilevel storage. However, the present erasing technique does not use the wide erased cell VT distribution as one of the 2N VT ranges. All the VT ranges are optimized such that the VLE range is similar in width to any of the other VT ranges even while simultaneously overlapping the VLE range.

Figs. 17A-17C show examples of the long term data storage distributions of cell VT for all the cells within an erase block for four bit per cell multilevel storage. Also, Figs. 17A-17C all share the same memory array processing technology and thus have exactly the same:

1) range of useful cell VT for programming and erasing, indicated as VF and VLE range.

However, these drawings differ from each other in the type of erasing technique and resulting VR value, VL range, and VM range. Fig. 17A shows an example where the VLE range is used for the VL0 level (between the VR0 and VR1 reference values) similar to the previously used techniques. Sixteen levels are shown in Fig. 17A. Each level is separated by VM; fifteen levels are VL wide; one level is VLE wide: %VL >>

Another variation is shown in Fig. 17B. Unlike the VL levels in Fig. 17B, which are spread over the whole VF range, the VL levels in Fig. 17C do not overlap into the VLE range. This does not give the benefits of a wide VL + VM voltage, but may be necessary for more reliable data sensing in some technologies. Additionally, as previously described in the mini-erasing section, the erase technique of the integrated circuit accommodates either direction of erase VT shift. This invention allows for a wide variety of memory array processes and sensing circuits for multilevel storage.

The operation utilizes band sensing capabilities, similar to those described in the multilevel digital sensing section, to verify that each cell’s VT is shifted into the desired band of the VL range. This ensures adequate margin guard bands to prevent ambiguous sensing at both ends of the VL range.

Multiple reference voltages are used during the erasing verify. Reference voltages at VR0+VMFL and VR1+VMFL which, in conjunction with VR0 and VR1, define two new lower and upper erasing margin guard bands, VMEFL and VMFL, within the VL0 level, as shown in Figs. 18A and 18B. Erase reference voltages at VER, VEB and programming reference voltages at VPEB, VPEF.
are used. Other voltages might also be used, as explained below. \( V_{\text{MEFL}} \) and \( V_{\text{MEFU}} \) may, or may not, be equal to and may, or may not, be greater than the previously described \( V_{\text{MRUI}} \) and \( V_{\text{MRUL}} \) read, or \( V_{\text{PLI}} \) and \( V_{\text{PUI}} \) micro-programming, or \( V_{\text{PEL}} \), \( V_{\text{PELU}} \) and \( V_{\text{MIUI}} \) mini-erase, or \( V_{\text{PPEL}} \), \( V_{\text{CPUI}} \) micro-programming, or \( V_{\text{PLI}} \), \( V_{\text{PUI}} \) programming guard bands.

[0175] The \( V_{\text{MEFL}} \) and \( V_{\text{MEFU}} \) margin guard bands within the \( V_{\text{LO}} \) level are chosen so that the resulting \( V_{T} \) distribution after the erasing technique is optimized within a desired band which is narrower than the \( V_{\text{LO}} \) range. The optimization may require the \( V_{\text{MEFL}} \) and \( V_{\text{MEFU}} \) bands to be unequal to each other. Alternatively, the bands may be of equal value.

[0176] This technique is optimized to accommodate a wide variety of memory processing technologies. The verify technique contrasts with the prior art which verifies the cell \( V_{T} \) only relative to a single erase reference voltage. In the previous techniques, a cell may be erased such that the \( V_{T} \) is so close to a \( V_{R0} \) or \( V_{R1} \) reference value that the previously described ambiguous sensing problem is created. By verifying the erased data within an optimized band using additional verify reference voltages, the present erase technique prevents the ambiguous data sensing problem.

**Erasing Methodology**

[0177] The erasing technique is best described by considering a plurality of cells which are connected in the same erase block. As stated previously, practical multilevel memory systems utilize an erase block memory cell architecture for erasing many cells simultaneously to reduce cell area. It is necessary to comprehend the behavior of all the cells at least within such an erase block, if not the whole chip, to observe all the ramifications of any form of erasing.

[0178] Figs. 18A and 18B show the \( V_{T} \) distribution of a plurality of cells in the memory which are within four levels, \( V_{\text{LO}} \) through \( V_{\text{L3}} \), at seven different times, \( T_{\text{PE}} \), \( T_{\text{E1}} \), \( T_{\text{EPE2}} \), \( T_{\text{E2}} \), \( T_{\text{EPEF}} \), \( T_{\text{EF}} \), and \( T_{\text{PLI}} \). Figs. 18A and 18B follow the example in Fig. 17B. In this example, the \( V_{\text{ERASE}} \) range overlaps four \( V_{\text{L}} \) ranges. However, this technique should not be considered limited to this particular example.

[0179] The \( V_{T} \) distribution at time \( T_{\text{PE}} \) shows the condition of the cells just prior to erasure. The erasure methodology is similar in many respects to the mini-erasing methodology and is described in less detail since many of the same principles are used. First, the chosen sector is given an erase pulse. All the cells in the sector are then verified by either one of two ways depending on the characteristics of the memory array processing. The first way verifies that all cells are below the \( V_{\text{EA}} \) reference voltage, while the second way verifies that all the cells are above the \( V_{\text{R0}}+V_{\text{MEFL}} \) margin reference voltage. Subsequent erase pulse/verify sequences are applied until either all cells verify below the \( V_{\text{EA}} \) reference voltage, until the first cell is verified within the \( V_{\text{MEFL}} \) margin guard band, or until a maximum number of sequences (error flag) is reached. The resulting cell \( V_{T} \) distribution is shown at time \( T_{\text{E1}} \) in Fig. 18A.

[0180] Next, a series of pre-erase mini-programming/mini-erase steps are performed which alternatively tighten up and shift the erased cell distribution towards the \( V_{\text{R0}}+V_{\text{MEFL}} \) margin reference voltage. The number of such pre-erase mini-programming/mini-erase sequences is again dependent upon the process to construct the memory and is minimized by the appropriate choice of reference voltage values.

[0181] The example in Figs. 18A and 18B shows two such pre-erase mini-programming/mini-erase sequences at times \( T_{\text{EPE2}} \), \( T_{\text{E2}} \), and \( T_{\text{EPEF}} \), \( T_{\text{EF}} \) respectively. In the example shown in Fig. 18A, the first pre-erase mini-programming reference voltage is set at \( V_{\text{PEB}} \). Mini-programming pulse/verify at \( V_{\text{PEB}} \) operations are applied until all cells verify above the \( V_{\text{PEB}} \) reference voltage. The cell \( V_{T} \) distribution resulting from this first pre-erase mini-programming step is shown at time \( T_{\text{EPE2}} \). Next, mini-erase pulse/verify operations are applied until either all cells verify below the \( V_{\text{EB}} \) reference voltage, until the first cell is verified within the \( V_{\text{MEFL}} \) margin guard band, or until a maximum number of sequences (error flag) is reached. The \( V_{\text{EB}} \) reference voltage is chosen optimally between the previous \( V_{\text{EA}} \) and \( V_{\text{R0}}+V_{\text{MEFL}} \) reference voltages. The resulting cell \( V_{T} \) distribution is shown at time \( T_{\text{E2}} \) in Figs. 18A-18B.

[0182] The pre-erase mini-programming/mini-erase steps are repeated again. Next, the second pre-erase mini-programming reference voltage is set at \( V_{\text{PEF}} \). The \( V_{\text{PEF}} \) reference voltage is chosen optimally between the previous \( V_{\text{PEB}} \) and \( V_{\text{R0}}+V_{\text{MEFL}} \) reference voltages. Mini-programming pulse/verify at \( V_{\text{PEF}} \) operations are applied until all cells verify above the \( V_{\text{PEF}} \) reference voltage. The cell \( V_{T} \) distribution resulting from this second pre-erase mini-programming step is shown at time \( T_{\text{EPEF}} \). Next, mini-erase pulse/verify operations are applied until either all cells verify below the \( V_{\text{R1}}+V_{\text{MEFL}} \) guard band reference voltage, until the first cell is verified within the \( V_{\text{MEFL}} \) margin guard band, or until a maximum (error flag) is reached.

[0183] This time, instead of applying another mini-programming step, it has been predetermined to be the time to verify using the \( V_{\text{R1}}+V_{\text{MEFL}} \) guard band reference voltage. If no cells are verified within the \( V_{\text{MEFL}} \) guard band or if a maximum number of erase steps (error flag) have been performed, the erase operation is completed. Alternatively, additional pre-erase mini-programming/mini-erase sequences are performed up to a maximum (error flag) using the next \( V_{\text{PE}} \) reference voltage only if cells were verified to be above the \( V_{\text{R1}}+V_{\text{MEFL}} \) guard band reference voltage at the end of the previous mini-erase verify step. The next \( V_{\text{PE}} \) reference voltage is chosen optimally between the previous \( V_{\text{PE}} \) and \( V_{\text{R0}}+V_{\text{MEFL}} \) reference voltages. The resulting cell \( V_{T} \) distribution after the second mini-erase is shown at time...
VPEB, VEB, and VPEF reference voltages are generated in a similar fashion and referenced to the closest VR1 not to remain in the VLO level. The cell VT distribution block is now ready for programming the cells which are the VR1-VMEFU guard band reference voltage. The cell VT distribution after the first part programming technique previously described, is shown at time Tp in Fig. 18B.

Multilevel Memory Erasing System

The erasing operations are nearly identical to those of the mini-erasing technique described previously. The exceptions are that the PROGRAM ERASE READ RESTORE SEQUENCER block 26 instructs the ∆VUVXVY GEN block 39 via the PRGRDB bus 37 to generate the VMEFU and VMEFL reference voltages, instead of the VMEFU andVMELI voltages from the reference voltage bus RVBUS 42. The VX VY GEN 38 is instructed to generate the VR0 and VR1 reference voltages. The VEA, VPEB, VEB, and VPEF reference voltages are generated in a similar fashion and referenced to the closest VR1 depending on the optimization previously described. Also, as previously described in the mini-erasing section, the PRG-ERS bus 22 is forced to setup the BLOCK-DEC blocks 19, X-DEC blocks 17 and the Y-DEC blocks 18 to allow an erase operation to occur on the selected memory block or sector.

While the above is a complete description of the preferred embodiments of the present invention, various alternatives, modifications and equivalents may be used. It should be evident that the present invention is equally applicable by making appropriate modifications to the embodiment described above. The memory technologies that apply to this invention are many and are not limited to just the storage of electric charge. Other fundamental properties of nature are capable of being stored in an integrated circuit memory cell. For instance, ferroelectric polarization in a film over a semiconductor channel modulates the conductivity of the channel and forms the basis for ferroelectric memories. The amount of polarization can be controlled by the voltage applied to a conductor, the control gate, over the ferroelectric film such that different degrees of conductivity can be obtained. Reference voltages may be used to set the control gate voltage. The present invention can thus be applied to ferroelectric memories analogously to charge storage memories.

Therefore, the above description should not be taken as limiting the scope of invention which is defined by the metes and bounds of the appended claims.

Claims

1. An integrated circuit memory system comprising

a plurality of memory cells, each memory cell storing one of a plurality of discrete states corresponding to bits of information;

characterised in that each discrete state has a preselected range; by

means for sensing a drift by said memory cell inside the preselected range of said one discrete state, and by

means for restoring said one discrete state upon drift by said memory cell outside of said preselected range of said one discrete state to maintain said bits of information corresponding to said one discrete state.

2. The memory system of claim 1 wherein said sensing means operates substantially simultaneously and independently upon a selected plurality of memory cells.

3. The memory system of claim 1 wherein each memory cell comprises a nonvolatile-memory cell storing an amount of electric charge on a floating gate corresponding to said bits of information; and further comprising

circuit means for maintaining said amount of charge in each floating gate of said plurality of memory cells responsive to said amount of charge; whereby loss of said bits of information in each memory cell is avoided.

4. The memory system of claim 3 wherein said maintaining circuit means incrementally adds charge to or subtracts charge from each nonvolatile memory cell to adjust said amount of charge without first erasing substantially all charge from said nonvolatile memory cell.

5. The memory system of claim 3 wherein said bits of information in each memory cell comprise at least two bits.

6. The memory system of claim 1 further comprising

means for generating more than $2^N$ reference values; and wherein

each memory cell is capable of storing one of $2^N$ discrete states corresponding to N bits of information; and said sensing means senses said one discrete state in said memory cell by comparison to said reference values and determines a drift outside a preselected range of said one discrete state with reference to said reference values.

7. The memory system of claim 6 wherein said sensing means and said reference value generating means cooperate to sequentially compare said one discrete state with said reference values in an ordered sequence to determine said one discrete state, and to determine said drift in said one discrete
state with reference to said reference values.

8. The memory system of claim 6 wherein said reference values comprise first and second sets, said sensing means responsive to said first set of reference values for sensing said one discrete state and said sensing means responsive to said second set of reference values for determining drift in said one discrete state in said memory cell, said second set of reference values having a predetermined relationship with said first set of reference values.

9. The memory system of claim 8 further comprising means responsive to said sensing means for restoring said memory cell in said one discrete state so that information corresponding to said one discrete state is maintained in each one of said plurality of memory cells.

10. The memory system of claim 6 wherein each memory cell stores an amount of electrical charge indicative of said one discrete state; and wherein said sensing means determines whether electrical charge should be added to or removed from said memory cell to maintain charge indicative of said one discrete state; and further comprising means for setting at least one status flag responsive to said sensing means determination.

11. The memory system of claim 10 further comprising means for storing said at least one status flag with a memory location related to said memory cell.

12. The memory system of claim 10 wherein said sensing means compares said one discrete state with respect to at least two of said reference values to determine whether surplus or insufficient charge has been added to said memory cell to restore said one discrete state.

13. The memory system of claim 1, wherein said plurality of memory cells are arranged in an array, each memory cell storing an amount of electric charge corresponding to said information bits; and further comprising a programming circuit coupled to said memory cell array, said circuit supplying signals to said memory cell array to move electric charge to and from said memory cells so that charge stored in said memory cells corresponds to said information bits; a reference voltage circuit capable of generating a plurality of reference voltages; comparators coupled to said memory cell array and said reference voltage circuit; and control circuitry coupled to said memory cell array, said reference voltage circuit and said comparators, said control circuitry connecting a selected plurality of memory cells in said memory cell array and said reference voltage circuit to said comparators to sense an amount of charge, and to determine drift in said amount of charge, corresponding to information bits in each of said selected plurality of memory cells.

14. The memory system of claim 13, wherein said control circuitry responsive to drift determination engages said programming circuit to adjust electric charge in said selected plurality of memory cells so that loss of information bits in each memory cell is avoided.

15. The memory system of claim 14 wherein said programming circuitry adjusts said charge without first erasing substantially all charge from said plurality of memory cells.

16. The memory system of claim 13 wherein said comparators compare a voltage responsive to said electric charge amount in each of said selected plurality of memory cells with respect to a first set of reference voltages in said plurality of reference voltages to determine said digital information in each of said selected plurality of memory cells; and wherein said comparators compare said voltage responsive to said electric charge amount in each of said selected plurality of memory cells with respect to a second set of reference voltages to determine said drift in said charge amount in said charge corresponding to said digital information; and wherein said control circuitry sets status flags indicative of said determination of said drift in said charge amount in said charge corresponding to said information bits.

17. A method of operating an integrated circuit memory system having a plurality of memory cells, comprising storing charge in each memory cell in one of a plurality of discrete amounts, each discrete amount corresponding to information bits, characterised by each amount having a preselected range; passively sensing a drift by said memory cell outside the preselected range of said one discrete amount; and restoring charge in said memory cell responsive to said drift from said preselected range of said one discrete amount.

18. The method of claim 17 wherein said sensing step comprises iteratively comparing a value corresponding to said amount of charge against said first sequence of reference values, a reference value in one comparing step dependent upon results of a previous comparing step; and further comprising
determining a drift in said amount of charge in each memory cell against a second sequence of reference values; and

setting at least one status flag indicative of a direction of drift if such drift determination step is affirmative.

19. The method of claim 18 further comprising storing said at least one status flag with a memory location related to said memory cell.

20. The method of claim 19 wherein said drift determining step comprises

iteratively comparing said value corresponding to said amount of charge against said second sequence of reference values, a reference value in one comparing step dependent upon results of a previous comparing step.

21. The method of claim 20 wherein each of said iteratively comparing steps against said second sequence of reference values follows one of said iteratively comparing steps against said first sequence of reference values.

22. The method of claim 18 further comprising comparing said value corresponding to said amount of charge against a reference value greater or smaller than any of said sequence of reference values, dependent upon whether said sensing step determines that said corresponding information bits are either all logic "1"s or all logic "0"s.

23. The method of claim 18 wherein said plurality of discrete amounts for storing charge in each memory cell comprising $2^N$, and said comparing steps are repeated at least $N$ times.

Patentansprüche


2. Speichersystem gemäß Anspruch 1, wobei die Abtasteinrichtung im Wesentlichen gleichzeitig und unabhängig an einer Vielzahl ausgesuchter Speicherzellen arbeitet.

3. Speichersystem gemäß Anspruch 1, wobei jede Speicherzelle einen nicht-flüchtigen Speicher umfasst, der eine der bitweisen Information entsprechende Ladungsmenge auf einem schwebenden Gate speichert; und ferner umfassend eine Schaltungeneinrichtung zur Aufrechterhaltung der Ladungsmenge in jedem der schwebenden Gates der Vielzahl von Speicherzellen, die diese Ladungsmenge aufnehmen; wodurch der Verlust der bitweisen Information in jeder der Speicherzellen verhindert wird.

4. Speichersystem gemäß Anspruch 3, wobei die Aufrechterhaltungseinrichtung jeder nicht-flüchtigen Speicherzelle Ladungen hinzufügt oder von ihr abzieht, um die Ladungsmenge anzupassen, in dem Wesentlichen ohne zuerst die gesamte Ladung aus einer nicht-flüchtigen Speicherzelle zu entnehmen.

5. Speichersystem gemäß Anspruch 3, wobei die bitweise Information in jeder Speicherzelle wenigstens zwei Bits umfasst.


7. Speichersystem gemäß Anspruch 6, wobei die Abtasteinrichtung und die Einrichtung zur Erzeugung von Referenzwerten zusammenwirken zum sequentiellen Vergleichen des jeweiligen diskreten Zustandes mit den Referenzwerten in einer geordneten Reihenfolge, um den jeweiligen diskreten Zustand zu bestimmen und die Drift des jeweiligen diskreten Zustandes im Verhältnis zu den Referenzwerten zu bestimmen.

8. Speichersystem gemäß Anspruch 6, wobei die Referenzwerte einen ersten und einen zweiten Satz umfassen, die Abtasteinrichtung verwendet den ersten Satz der Referenzwerte zur Bestimmung des jeweiligen diskreten Zustandes, während der zweite Satz Referenzwerte von der Abtasteinrichtung zur...
Bestimmung der Drift des jeweiligen Zustands in der Speicherzelle verwendet wird, der zweite Satz Referenzwerte steht in einem vorbestimmten Verhältnis zum ersten Satz Referenzwerte.

9. Speichersystem gemäß Anspruch 8, ferner umfassend eine Vergleichsvorrichtung, die auf die Abtasteinrichtung reagiert und den jeweiligen diskreten Zustand in der Speicherzelle so wieder herstellt, dass die zu dem jeweiligen diskreten Zustand entsprechende Information in jeder der vielfältigen Speicherzellen erhalten bleibt.

10. Speichersystem gemäß Anspruch 6, wobei jede Speicherzelle eine in Bezug zum jeweiligen diskreten Zustand stehende elektrische Ladungsmenge speichert; und die Abtasteinrichtung bestimmt, ob elektrische Ladungen der Speicherzelle hinzugefügt oder aus ihr entfernt werden sollen, um eine mit dem jeweiligen diskreten Zustand in Bezug stehende Ladung aufrechtzuhalten; und ferner umfassend eine Vorrichtung zum Setzen von wenigstens einem Statusanzeiger für die von der Abtasteinrichtung bestimmten Festlegungen.

11. Speichersystem gemäß Anspruch 10, ferner umfassend eine Vorrichtung zur Speicherung wenigstens eines Statusanzeigers an einer Speicherstelle, die mit der Speicherzelle in Verbindung steht.

12. Speichersystem gemäß Anspruch 10, wobei die Abtasteinrichtung den jeweiligen diskreten Zustand mit wenigstens zwei Referenzwerten vergleicht, um zu bestimmen, ob zuviel oder zu wenig Ladung der Speicherzelle zur Herstellung des jeweiligen diskreten Zustandes hinzugefügt wurde.

13. Speichersystem gemäß Anspruch 1, wobei die Vielzahl von Speicherzellen in einer Matrix angeordnet ist und jede Speicherzelle eine elektrische Ladungsmenge speichert, die der bitweisen Information entspricht; und ferner umfassend eine programmierbare Einheit, die mit der Matrix der Speicherzellen verbunden ist; die programmierbare Einheit versorgt die Matrix der Speicherzellen derart mit Steuersignalen zur Verschiebung elektrischer Ladungen zu und von den Speicherzellen, dass die Ladung in den Speicherzellen der jeweiligen bitweisen Information entspricht; eine Referenzspannungseinheit, die in der Lage ist, eine Vielzahl von Referenzspannungen einzustellen; den Komparatoren, die mit der Matrix der Speicherzellen und der Referenzspannungseinheit in Verbindung stehen und ein Regelungsschaltkreis, verbunden mit der Matrix der Speicherzellen, der Referenzspannungseinheit und den Komparatoren; der Regelungsschaltkreis verbindet eine Vielzahl ausgesuchter Speicherzellen in der Matrix der Speicherzellen und die Referenzspannungseinheit mit den Komparatoren zur Messung der Ladungsmenge, welche der bitweisen Information in jeder der vielfältigen ausgesuchten Speicherzellen entspricht sowie der entsprechenden Drift.

14. Speichersystem gemäß Anspruch 13, wobei der Regelungsschaltkreis zur Bestimmung der Drift die programmierbare Einheit dazu in Anspruch nimmt, um die elektrische Ladung in den vielfältigen ausgesuchten Speicherzellen so anzupassen, dass ein Verlust der bitweisen Information in jeder Speicherzelle vermieden wird.

15. Speichersystem gemäß Anspruch 14, wobei die programmierbare Einheit die Ladungen im Wesentlichen ohne ein völliges Entnehmen der Ladungen aus der Vielzahl der Speicherzellen anpasst.

16. Speichersystem gemäß Anspruch 13, wobei die Komparatoren eine Spannung, die aus der elektrischen Ladungsmenge in jeder der vielfältigen ausgesuchten Speicherzellen mit einem ersten Satz von Referenzspannungen aus der Vielzahl von Referenzspannungen vergleichen, um die digitale Information in jeder der vielfältigen ausgesuchten Speicherzellen zu bestimmen; und wobei die Komparatoren die Spannungen, die sich aus der Ladungsmenge in jeder der vielfältigen ausgesuchten Speicherzellen ergeben, mit dem zweiten Satz von Referenzspannungen vergleichen, um die Drift der Ladungen, die der digitalen Information entsprechen, zu bestimmen; und wobei die Regelungseinheit Statusanzeigen zu den, der bitweisen Information entsprechenden, Ladungen abhängig von der Festlegung der Drift in der Ladungsmenge setzt.

18. Verfahren gemäß Anspruch 17, wobei der Abtastschritt einen iterativen Vergleich eines aus der Ladungsmenge resultierenden Wertes mit einer ersten Folge von Referenzwerten umfasst, der Referenzwert in einem Vergleichsschritt hängt von den Resultaten des vorherigen Vergleichsschritts ab; und ferner umfassend die Bestimmung einer Drift in der Ladungsmenge in jeder Speicherzelle im Vergleich zu einer zweiten Folge von Referenzwerten; und Setzen, für den Fall, dass ein Driftbestimmungs schritt bejaht wird, eines Statusanzeigers, der die Richtung der Drift anzeigt.


22. Verfahren gemäß Anspruch 18, ferner umfassend den Vergleich des der Ladungsmenge entsprechenden Wertes mit einem Referenzwerte, der größer oder kleiner ist als alle Werte aus der Folge der Referenzwerte in Abhängigkeit davon, ob im Abtastschritt festgestellt wird, dass die korrespondierende bitweise Information entweder aus nur logischen "1" oder nur aus logischen "0" besteht.

23. Verfahren gemäß Anspruch 18, wobei die Vielzahl diskreter Werte für die Speicherung von Ladung in einer Speicherzelle 2N umfasst, und die Vergleichsschritte wenigstens N Mal durchgeführt werden.

Revendications

1. Un système à mémoire à circuit intégré comprenant une pluralité de cellules de mémoire, chaque cellule de mémoire stockant un d'une pluralité d'états discrets correspondant à des bits d'information ; caractérisé en ce que chaque état discret a une plage présélectionnée ; par des moyens pour détecter une dérive de ladite cellule de mémoire hors de la plage présélectionnée dudit état discret, et par des moyens pour restaurer ledit état discret en cas de dérive de ladite cellule de mémoire hors de ladite plage présélectionnée dudit état discret pour maintenir lesdits bits d'information correspondant audit état discret.

2. Le système à mémoire selon la revendication 1 dans lequel lesdits moyens de détection opèrent pour l'essentiel simultanément et de façon indépendante sur une pluralité sélectionnée de cellules de mémoire.

3. Le système à mémoire selon la revendication 1 dans lequel chaque cellule de mémoire comprend une cellule de mémoire non volatile stockant une quantité de charge électrique sur une porte flottante correspondant auxdits bits d'information ; et comprenant en outre des moyens de circuit pour maintenir ladite quantité de charge dans chaque porte flottante de ladite pluralité de cellules de mémoire réagissant à ladite quantité de charge ; dans lequel la perte desdits bits d'information dans chaque cellule de mémoire est évitée.

4. Le système à mémoire selon la revendication 3 dans lequel lesdits moyens de circuit de maintien ajoutent ou soustraient de façon incrémentale des charges de chaque cellule de mémoire non volatile pour ajuster ladite quantité de charge sans effacer auparavant pour l'essentiel toutes les charges de ladite cellule de mémoire non volatile.

5. Le système à mémoire selon la revendication 3 dans lequel lesdits bits d'information dans chaque cellule de mémoire comprennent au moins deux bits.

6. Le système à mémoire selon la revendication 1 comprenant en outre des moyens pour générer plus de 2N valeurs de référence ; et dans lequel chaque cellule de mémoire est capable de stocker un parmi 2N états discrets correspondant à N bits d'information ; et lesdits moyens de détection détectent ledit état discret dans ladite cellule de mémoire par comparaison auxdites valeurs de référence et déterminent une dérive hors d'une plage présélectionnée dudit état discret par rapport auxdites valeurs de référence.

7. Le système à mémoire selon la revendication 6 dans lequel lesdits moyens de détection et lesdits moyens de génération de valeur de référence coopèrent pour comparer séquentiellement ledit état
discret avec lesdites valeurs de référence dans une séquence ordonnée pour déterminer ledit état discret, et pour déterminer ladite dérive dans ledit état discret par référence auxdites valeurs de référence.

8. Le système à mémoire selon la revendication 6 dans lequel lesdites valeurs de référence comprennent un premier et un second ensembles, lesdits moyens de détection réagissant audit premier ensemble de valeurs de référence pour déterminer et lesdits moyens de détection réagissant audit second ensemble de valeurs de référence pour déterminer la dérive dans ledit état discret dans ladite cellule de mémoire, ledit second ensemble de valeurs de référence ayant une relation prédéterminée avec ledit premier ensemble de valeurs de référence.

9. Le système selon la revendication 8 comprenant en outre des moyens réagissant auxdits moyens de détection pour restaurer ladite cellule de mémoire dans un état discret de façon que les informations correspondant audit état discret soient maintenues dans chacune d’une pluralité de cellules de mémoire.

10. Le système à mémoire selon la revendication 6 dans lequel chaque cellule de mémoire stocke une quantité de charge électrique indicative d’un tel état discret ; et dans lequel lesdits moyens de détection déterminent si la charge électrique doit être ajoutée ou retirée de ladite cellule de mémoire pour maintenir la charge indicative d’un état discret ; et comprenant en outre des moyens pour définir au moins un indicateur d’état réagissant à la détermination desdits moyens de détection.

11. Le système à mémoire selon la revendication 10 comprenant en outre des moyens pour stocker au moins un indicateur d’état avec un emplacement mémoire lié à ladite cellule de mémoire.

12. Le système à mémoire selon la revendication 10 dans lequel lesdits moyens de détection comparent ledit état discret par rapport à au moins deux valeurs de référence pour déterminer si une charge en excès ou insuffisante a été ajoutée à ladite cellule de mémoire pour restaurer ledit état discret.

13. Le système à mémoire selon la revendication 1, dans lequel ladite pluralité de cellules de mémoire sont disposées en matrice, chaque cellule de mémoire stockant une quantité de charge électrique correspondant auxdits bits d’information ; et comprenant en outre un circuit de programmation couplé à ladite matrice de cellules de mémoire, ledit circuit fournissant des signaux à ladite matrice de cellules de mémoire pour déplacer les charges électriques vers et depuis lesdites cellules de mémoire de façon que la charge stockée dans lesdites cellules de mémoire corresponde auxdits bits d’information :
   5 un circuit de tension de référence capable de générer une pluralité de tensions de référence ;
   10 des comparateurs couplés à ladite matrice de cellules de mémoire et audit circuit de tension de référence ; et
   15 un circuit de contrôle couplé à ladite matrice de cellules de mémoire, audit circuit de tension de référence et auxdits comparateurs, ledit circuit de contrôle connectant une pluralité sélectionnée de cellules de mémoire dans ladite matrice de cellules de mémoire et ledit circuit de tension de référence auxdits comparateurs pour détecter une quantité de charge, et pour déterminer la dérive dans ladite quantité de charge, correspondant aux bits d’information dans chacune de la pluralité sélectionnée de cellules de mémoire.

14. Le système à mémoire selon la revendication 13, dans lequel ledit circuit de contrôle réagissant à la détermination de la dérive engage ledit circuit de programmation pour ajuster la charge électrique dans ladite pluralité de cellules de mémoire de façon que la perte de bits d’information dans chaque cellule de mémoire soit évitée.

15. Le système à mémoire selon la revendication 14 dans lequel la circuiterie de programmation ajuste ladite charge sans d’abord effacer pour l’essentiel toutes les charges de ladite pluralité de cellules de mémoire.

16. Le système à mémoire selon la revendication 13 dans lequel lesdits comparateurs comparent une tension réagissant à ladite quantité de charge dans chacune de ladite pluralité sélectionnée de cellules de mémoire par rapport à un premier ensemble de tensions de référence dans ladite pluralité de tensions de référence pour déterminer ladite information numérique dans chacune de ladite pluralité sélectionnée de cellules de mémoire ; et dans lequel
   40 lesdits comparateurs comparent ladite tension réagissant à ladite quantité de charge électrique dans chacune de ladite pluralité sélectionnée de cellules de mémoire par rapport à un second ensemble de tensions de référence pour déterminer ladite dérive dans ladite quantité de charge dans ladite charge correspondant à ladite information numérique ; et dans lequel
   45 ledit circuit de contrôle configure des indicateurs d’état indicatifs de ladite détermination de ladite dérive dans ladite quantité de charge dans la-
dite charge correspondant auxdits bits d'information.

17. Une méthode pour faire fonctionner un système de mémoire à circuit intégré ayant une pluralité de cellules de mémoire, comprenant

le stockage de charge dans chaque cellule de mémoire en l'une d'une pluralité de quantités discrètes, chaque quantité discrète correspondant à des bits d'information.

**caractérisée en ce que** chaque quantité a une plage présélectionnée ;

la détection passive d'une dérive par ladite cellule de mémoire hors de la plage présélectionnée de ladite quantité discrète ; et

la restauration de la charge dans ladite cellule de mémoire en réponse à ladite dérive par rapport à ladite plage présélectionnée de ladite quantité discrète.

18. La méthode selon la revendication 17 dans laquelle l'étape de détection comprend la comparaison itérative d'une valeur correspondant à ladite quantité de charge par rapport à ladite première séquence de valeurs de référence, une valeur de référence dans une étape de comparaison dépendant des résultats d'une précédente étape de comparaison ; et comprenant en outre

la détermination d'une dérive dans ladite quantité de charge dans chaque cellule de mémoire par rapport à une seconde séquence de valeurs de référence ; et

la définition d'au moins un indicateur d'état indicatif d'une direction de dérive si cette étape de détermination de dérive est affirmative.

19. La méthode selon la revendication 18 comprenant en outre le stockage dudit au moins un indicateur d'état avec un emplacement mémoire lié à ladite cellule de mémoire.

20. La méthode selon la revendication 19 dans laquelle ladite étape de détermination de la dérive comprend

la comparaison itérative de ladite valeur correspondant à ladite quantité de charge par rapport à ladite seconde séquence de valeurs de référence, une valeur de référence dans une étape de comparaison dépendant des résultats d'une précédente étape de comparaison.

21. La méthode selon la revendication 20 dans laquelle chacune des étapes de comparaison itérative par rapport à la seconde séquence de valeurs de référence suit une desdites étapes de comparaison itérative par rapport à ladite première séquence de valeurs de référence.

22. La méthode selon la revendication 18 comprenant en outre la comparaison de ladite valeur correspondant à ladite quantité de charge par rapport à une valeur de référence supérieure ou inférieure à l'une des valeurs de ladite séquence de valeurs de référence, selon que ladite étape de détection détermine que lesdits bits d'information correspondants sont soit tous des "1" logiques soit tous des "0" logiques.

23. La méthode selon la revendication 18 dans laquelle ladite pluralité de quantités discrètes pour stocker la charge dans chaque cellule de mémoire comprenant $2^N$ et lesdites étapes de comparaison sont répétées au moins $N$ fois.
**Prior Art**

**FIG. 1A.**

Number of Cells

<table>
<thead>
<tr>
<th>V_R1</th>
<th>V_R2</th>
<th>V_R3</th>
</tr>
</thead>
</table>

**Prior Art**

**FIG. 1B.**

Number of Cells

<table>
<thead>
<tr>
<th>V_R1</th>
<th>V_R2</th>
<th>V_R3</th>
<th>V_R4</th>
<th>V_R5</th>
<th>V_R6</th>
<th>V_R7</th>
<th>V_R8</th>
<th>V_R9</th>
<th>V_R10</th>
<th>V_R11</th>
<th>V_R12</th>
<th>V_R13</th>
<th>V_R14</th>
<th>V_R15</th>
</tr>
</thead>
</table>

**FIG. 1C.**

Number of Cells

<table>
<thead>
<tr>
<th>V_R1</th>
<th>V_R2</th>
<th>V_R3</th>
<th>V_R4</th>
<th>V_R5</th>
<th>V_R6</th>
<th>V_R7</th>
<th>V_R8</th>
<th>V_R9</th>
<th>V_R10</th>
<th>V_R11</th>
<th>V_R12</th>
<th>V_R13</th>
<th>V_R14</th>
<th>V_R15</th>
</tr>
</thead>
</table>
Prior Art
FIG. 2.
**Prior Art**

**FIG. 3A.**

**Prior Art**

**FIG. 3B.**

**FIG. 3C.**
**FIG. 4A.**

**FIG. 4B.**
FIG. 5A.

FIG. 5B.
Comparator Input

\[ V_{CRF} \]

\[ V_{R16} \]
\[ V_{R14} \]
\[ V_{R12} \]
\[ V_{RR} \]
\[ V_{LG(1+1)} \]
\[ V_{RI} \]
\[ V_{UG(1-2)} \]
\[ V_{R1} \]
\[ V_{R0} \]

\( V_{L0} \quad V_{L1} \quad V_{L2} \quad \ldots \quad V_{MRU} \quad V_{MRL} \)

Time: \( T_0 \quad T_{1D} \quad T_{1R} \quad T_{2D} \quad T_{2R} \quad T_{3D} \quad T_{3R} \quad T_{4D} \quad T_{4R} \quad T_{5L} \quad T_{5R} \)

**FIG. 6.**
N=3
Apply RESET Pulse
STPDIR = 1, RES = 0
FLGB = 1, FUGB = 0

Set BitN = 1

STPDIR = 1

VMEM > VX

RES = 1 Restore low Condition
FLGB = 0

STPDIR = 0

VMEM > VX

RES = 0
RES = 1

RES = 0 Restore High Condition

RES = 0
RES = 1

Reset BitN = 0

N = 0

N = N - 1

DATA
1111 or 0000

VX = V_{R16}

VX = V_{R0}

END

VMEM > VX

STPDIR = 1

RES = 1 Restore low Condition
FLGB = 0

STPDIR = 0

VMEM > VX

RES = 0
RES = 1

RES = 0 Restore High Condition
FUGB = 1

END

FIG. 8
Comparator Inputs; VMEM, VCRF

$V_{T}$

$V_{R15}$

$V_{R14}$

$V_{R12}$

$V_{R8}$

$V_{R4}$

$V_{R1}$

$V_{R0}$

$V_{U}$

$V_{LG(I-1)}$

$V_{U_{G(I-2)}}$
FIG. 12.
FIG. 13A.
FIG. 13B.
FIG. 14B.
Prior Art

FIG. 15A.
FIG. 15B.
FIG. 16A.
FIG. 16B.
FIG. 18B.