

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
21 August 2008 (21.08.2008)

PCT

(10) International Publication Number
WO 2008/100602 A1

(51) International Patent Classification:

GIIC 29/12 (2006.01) *GIIC 29/00* (2006.01)
GIIC 29/32 (2006.01)

(21) International Application Number:

PCT/US2008/002057

(22) International Filing Date:

15 February 2008 (15.02.2008)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

11/676,049 16 February 2007 (16.02.2007) US

(71) Applicant (for all designated States except US): **ATMEL CORPORATION** [US/US]; 2325 Orchard Parkway, San Jose, California 95131 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **REGGIORI, Riccardo** [IT/IT]; Res. Cantone 432, I-20090 Milano2,

Segrate (MI) (IT). **TASSAN CASER, Fabio** [IT/IT]; Via Monginervo 36, I-20043 Arcore (MI) (IT). **MARSELLA, Mirella** [IT/IT]; Via Tommasone 20 E, I-26025 Pandino (CR) (IT). **MARZIANI, Monica** [IT/IT]; Via Monginervo 36, I-20043 Arcore (MI) (IT).

(74) Agents: **CLISE, Timothy B.** et al.; Schwegman, Lundberg & Woessner, P.A., P.O. Box 2938, Minneapolis, Minnesota 55402 (US).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH,

[Continued on next page]

(54) Title: EMBEDDED ARCHITECTURE WITH SERIAL INTERFACE FOR TESTING FLASH MEMORIES

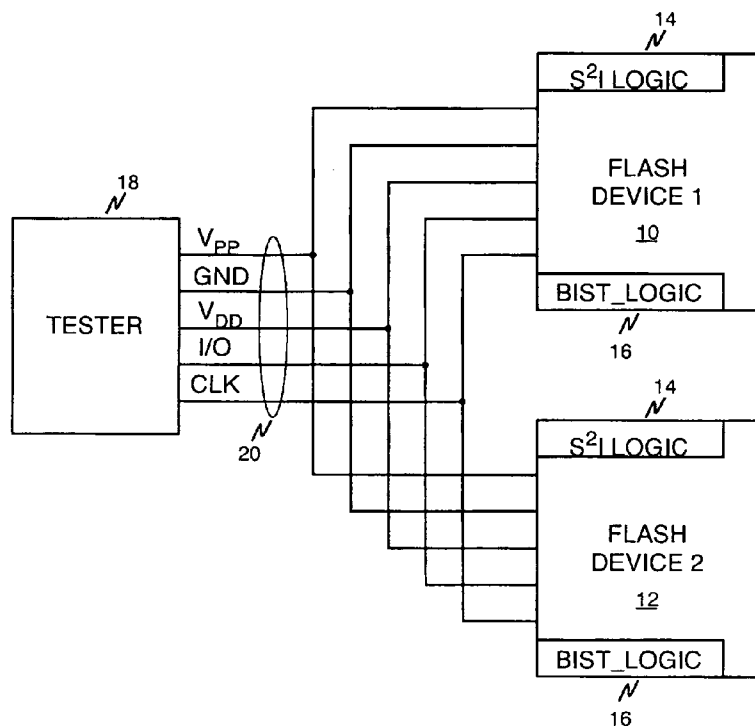


FIGURE 1

(57) Abstract: A flash memory device includes a flash memory array, a set of non-volatile redundancy registers, a serial interface, and testing logic coupled to the serial interface, the testing logic configured to accept a set of serial commands from an external tester; erase the array; program the array with a test pattern; read the array and compare the results with expected results to identify errors; determine whether the errors can be repaired by substituting a redundant row or column of the array, and if so, generate redundancy information; and program the redundancy information into the non-volatile redundancy registers.



GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

- *with international search report*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

EMBEDDED ARCHITECTURE WITH SERIAL
INTERFACE FOR TESTING FLASH MEMORIES

5

CLAIM OF PRIORITY

Benefit of Priority is hereby claimed to U.S. Patent Application No.
11/676,049, filed February 16, 2007 and entitled "EMBEDDED
ARCHITECTURE WITH SERIAL INTERFACE FOR TESTING FLASH
10 MEMORIES", which is incorporated herein by reference in its entirety.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention pertains to testing circuitry for flash memories, and
15 to improvements in the time needed to screen flash memory arrays. More
particularly, the present invention pertains to flash memory integrated circuits
with on-board testing circuitry.

2. The Prior Art

20 It is well known that screening a flash array accounts for the majority of
the time spent during flash memory testing. Flash array screening usually
consists of a sequence of erasing the array, programming the array with a
pattern, and reading and verifying the pattern written into the array. If the
programmed pattern can be verified, then the array is assumed to be defect free,
25 otherwise, if errors occur, some defects have been detected. Repeating the
sequence with different patterns allows for the highlighting of different types of
faults.

Thus, a key point in testing the array is determining whether or not the
defects can be repaired by substituting a redundant row or column of the array,
30 and calculating the redundancy information, which, once permanently stored in
the device, will be used during the entire life cycle of the chip to substitute the
faulty part of the array.

As technology features scale down and flash memory densities grow
larger, the total number of interactions increase between the external tester and

the flash memory during the array screening, becoming a real bottleneck. This is mainly due to the low frequency used to interface the flash memory with the external tester. Such low frequencies are needed to avoid bouncing and noise, because of the long cables connecting the probe card with the testing apparatus.

5 To deal with this issue, one of the testing strategies has been to embed on the chip all the logic and circuitry necessary for testing the array, thus reducing the number of interactions between the tester and the array. This strategy is also known as B.I.S.T. (Built-In Self Test).

10 Another common strategy to reduce test time has usually been to increase the number of dice tested in parallel. In this way, cost is shifted to expensive probe cards with larger buses for address and data, and complexity is transferred to longer software codes.

BRIEF DESCRIPTION OF THE INVENTION

15 An architecture for embedded testing of flash memory is presented. The present invention improves total testing time, thus reducing the overall die cost. It combines a serial interface, to increase the number of dice tested in parallel, with on-chip logic able to screen the memory array and reduce the number of interactions with an external tester.

20 In one aspect of the present invention, a flash memory device includes a flash memory array, a set of non-volatile redundancy registers, a serial interface, and testing logic coupled to the serial interface. The testing logic is configured to accept a set of serial commands from an external tester; erase the array; program the array with a test pattern; read the array and compare the results with expected
25 results to identify errors; determine whether the errors can be repaired by substituting a redundant row or column of the array, and if so, generate redundancy information; and program the redundancy information into the non-volatile redundancy registers.

30 In another aspect of the invention, a method for testing a flash memory array on a flash memory device is disclosed. The flash memory device includes a set of non-volatile redundancy registers, a serial interface, and testing logic coupled to the serial interface. The method comprises the steps of the testing logic: accepting a set of serial commands from an external tester; erasing the array; programming the array with a test pattern; reading the array and

comparing the results with expected results to identify errors; determining whether the errors can be repaired by substituting a redundant row or column of the array, and if so, generating redundancy information; and programming the redundancy information into the non-volatile redundancy registers.

5 The present invention aims at reducing the testing time of flash memories by combining a serial interface to increase the number of dice tested in parallel, without over increasing the total number of wires on the probe card (which will be referred to herein as S²I_LOGIC), and embedded logic to speed up the execution of test routines on the array, minimizing the total number of data
10 exchanges between the external tester and the flash device (which will be referred to herein as BIST_LOGIC).

To fulfill this task, the architecture of the present invention is able to accept a set of serial commands from an external tester and start its operations; to erase the array; to program the array using a standard pattern; to read the array
15 and compare the results with an expected pattern; to calculate whether or not the errors can be repaired by substituting a redundant row or column and to calculate the related redundancy information; to store the redundancy information inside its volatile registers; to program into non volatile registers the redundancy information stored in its volatile registers; and to read the information stored in
20 the non volatile registers into its volatile registers.

BRIEF DESCRIPTION OF THE DRAWING FIGURES

FIG. 1 is a block diagram showing flash memory devices configured according to the principles of the present invention coupled to a tester.

25 FIG. 2 is a block diagram of an integrated circuit in accordance with the present invention.

FIG. 3 is a block diagram showing the BIST architecture according to the present invention.

30 FIG. 4 is a diagram showing a S²I_LOGIC block in accordance with the present invention.

FIG. 5 is a block diagram showing in more detail a S²I_LOGIC block in accordance with the present invention.

FIG. 6 is a state diagram in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Persons of ordinary skill in the art will realize that the following description of the present invention is illustrative only and not in any way limiting. Other embodiments of the invention will readily suggest themselves to
5 such skilled persons.

The architecture of the present invention can be extended to chips with N redundancy resources made of K bits, where N is the number of redundancy resources available on the device and K is the number of bits needed to completely describe a single resource.

10 Referring now to FIG. 1, a block diagram shows two flash memory devices 10 and 12 configured according to the principles of the present invention and including a S²I logic block 14 and a BIST logic block 16. Flash memory devices 10 and 12 are coupled to a tester 18 via a 5-wire bus 20. The 5-wire bus 20 includes V_{PP}, a potential that may be used to program the flash memory
15 devices 10 and 12, V_{DD} to supply the standard power for the flash memory devices 10 and 12, GND, to supply the standard ground reference to the flash memory devices 10 and 12, I/O, a bi-directional serial data line used to exchange data between the tester and the flash memory devices 10 and 12, and CLK, a clock signal used to synchronize the data exchange between the tester and the
20 flash memory devices 10 and 12.

Referring now to FIG. 2, a block diagram shows an illustrative integrated circuit 30 having features in accordance with the present invention. As shown in FIG. 2, the integrated circuit 30 includes several circuits. MICRO 32 is a processor that executes instructions. ROM 34 is a read only memory that stores
25 instructions executable by the MICRO 32. S²I_LOGIC 36 is a circuit that decodes the serial commands coming from the external tester and sends back information about the status of the current internal operation (on going or finished). PADS 38 are the five device I/O pads defined by the S²I specification. ADDRESS COUNTER 26 is connected to S²I_LOGIC 36, and DECODER 28 is
30 connected to ADDRESS COUNTER 26 and ARRAY 42.

BIST_LOGIC 40 is a logic circuit that reads the data coming from the ARRAY 42 through the READ/VERIFY SENSE AMPS 44, defines redundancy information to be stored into its volatile registers, loads the PROGRAM LOGIC 46 to program the ARRAY 42 with a standard test pattern, loads the PROGRAM

LOGIC 46 to program the FUSE ARRAY 48, and loads the redundancy information from the FUSE ARRAY 48 into its volatile registers.

Referring now to FIG. 3, a block diagram shows architecture for a BIST logic circuit 50 according to the principles of the present invention. The following discussion assumes a flash memory chip with N redundancy resources of K bits each. Persons of ordinary skill in the art will readily be able to configure such an architecture with other configurations of redundancy resources.

The BIST_LOGIC circuit 50 includes several circuit blocks. PATTERN GENERATOR 52 is a circuit driven by commands from the S²I that generates a reference data pattern given the current row and column addresses (for instance ALL0, ALL1, checkerboard, diagonal, etc.). SCRAMBLER 54 is a circuit that scrambles the data read from the flash array in a standard format (i.e., if the device reads four words of 16 bits each in parallel, the scrambler may, for example, order the words from left to right and from most significant bit (MSB) to least significant bit (LSB) in a sequence of fourth word, third word, second word, first word.)

XOR 56 is a circuit that compares the scrambled data from the array with the reference pattern from the PATTERN GENERATOR 52 and outputs a current error vector. DECODER 58 is a circuit that, given the current redundancy configuration, decodes the already-detected errors (previous error vector). SUB 60 is a subtractor circuit that determines the new error vector, subtracting the previous error vector, which is from the DECODER, from the current error vector, which is from the XOR. ADDER 62 is an adder circuit that counts the number of new errors in the new error vector. CHECK REDUNDANCY RULES 64 is a circuit that determines if the new errors are repairable or not (many different rules may be implemented as defined by a user). If one or more new errors are not repairable, CHECK REDUNDANCY RULES 64 issues a FATAL_ERROR_INT signal. As an example, a simple rule that may be implemented where two redundancy resources are available is to assert the signal FATAL_ERROR_INT if there are more than two new errors in the current page.

CHOOSE REDUNDANCY RESOURCE 66 is a circuit that determines which redundancy resources are to be used. In the architecture shown in FIG. 3,

assume an example with two redundancy resources. If resource[1] is to be used, then the signal EN_REG[1] is driven high from CHOOSE REDUNDANCY RESOURCE 66. If resource[2] is to be used, then signal EN_REG[2] is driven high from CHOOSE REDUNDANCY RESOURCE 66. If both resources are to be used, both signals EN_REG[1] and EN_REG[2] are driven high from CHOOSE REDUNDANCY RESOURCE 66. If no resources are needed then both signals EN_REG[1] and EN_REG[2] are driven low from CHOOSE REDUNDANCY RESOURCE 66. Different filling sequences may be implemented depending on the chip redundancy architecture.

10 REDUNDANCY ENCODER 68 is a circuit that encodes the positions of the new errors, using information used by the SCRAMBLER 54. It can then encode data identifying the word and the particular output within that word in which the errors occurred.

BIST_REG[N:1] 70 are a set of N volatile registers, each having K bits, that can be either loaded with the current status of the redundancy stored in the fuse array or updated with the new redundancy calculated by the embedded redundancy logic depending on the state of the multiplexers 72.

FF[N:1] 74 are N flip-flops used to synchronize N signals EN_REG[N:1] on the rising edge of SAVE_BIST signal. FF[0] 76 is a flip-flop used to synchronize the signal FATAL_ERROR_INT on the rising edge of the SAVE_BIST signal. UPDATE_BIST[N:1] on lines 78 are N signals used to selectively trigger registers BIST_REG[N:1] 70.

The BIST_LOGIC has several inputs. DATA_BUS 80 is driven with a page of data read out of the flash array from the READ/VERIFY SENSE AMPS 44 of FIG. 2. CONF 82 is any number of signals coming from the S²I_LOGIC 36 of FIG. 2 used to select the current desired reference pattern. SAVE_BIST 84 is a pulse issued by the MICRO 32 of FIG. 2 during the execution of a scan routine each time a new page has been read from the array. READ_RED_BIST 86 is a signal issued by the MICRO 32 of FIG. 2 to switch the multiplexers 72 so that the current status of redundancy may be loaded into the BIST_REG 70. FUSE_RESOURCE[N:1] 88 is NxK bits of redundancy information coming from the N non volatile registers of the flash memory.

The BIST_LOGIC has several outputs. BIST_REG[N:1] 70 are N volatile registers made of K bits, each one associated with one of the N

redundancy resources of the chip. FATAL_ERROR 92 is a signal showing if the new errors in the last page read from the array can be repaired by substituting a redundant row or column.

- An example of a simple scan routine useful to verify a pattern and to
- 5 calculate and store the needed redundancy includes the following procedures:
1. load current redundancy status from fuses;
 2. read a page;
 3. trigger a SAVE_BIST pulse;
 4. check FATAL_ERROR signal;

10 5. if FATAL_ERROR=1 then exit and throw away the die;

 6. if FATAL_ERROR=0 then increase the current page address;
 7. jump to step 2 until the scan is completed.

Referring now to FIG. 4, a diagram shows an S²I_LOGIC block that controls the S²I interface in accordance with the present invention. S²I_LOGIC

15 is a logic circuit controlling the interface and has several input and output lines. The FUSE input 100 controls a fuse that if programmed permanently disables the S²I interface. The CLOCK Input 102 carries a clock signal used to synchronize data exchange between the flash memory device and the tester. The BUSY input 104 carries a signal from the MICRO 32 showing if the current

20 routine is being performed or is completed. The FATAL_ERROR input 106 is a signal from the BIST_LOGIC 40 showing whether or not the current portion of the array can be repaired by substituting a redundant row or column. The RESET input 108 is a signal to reset the S²I_LOGIC.

The BUFFER_DISAB output 110 is a signal to shut off the buffer

25 leakage once the pads are left floating. The I/O 112 is a bi-directional line. It accepts input commands from the tester and it outputs either FATAL_ERROR, used to notify the tester if an unrepairable error has occurred, or BUSY, used to notify the external tester when the internal operation has finished. An internal state machine will change the information on the I/O line 112, depending on the

30 current status of the ongoing operation. The I/O_DIRECTION line 114 outputs a signal to show if I/O 112 is in input or output mode. Both the tester and the flash memory device to which it is coupled must share the same protocol to understand when the I/O line is in input or output mode. The

MICRO_CONTROL output 116 is a set of signals used to trigger the MICRO 32 and make it execute the correct test routine.

The present invention advantageously utilizes a set of commands that are used to trigger all the operations that will be executed by the embedded testing logic. An exemplary command set is shown in the following table. Persons of ordinary skill in the art will appreciate that the present invention is not limited to the command set shown herein and that other useful commands may be specified for operating the apparatus disclosed herein. Persons of ordinary skill in the art will appreciate that software or firmware for providing this command set will vary depending on the exact circuitry used to implement the circuit blocks disclosed in the drawing figures herein. The provision of such software or hardware is a routine exercise for a person of ordinary skill in the art.

COMMAND	COMMENT
S ² I_INIT	Software reset
S ² I_INIT_BIST	Software reset for BIST registers
S ² I_LATCH_ADD	Latch plane address
S ² I_BIST_ALL0	Scan matrix for ALL0 pattern and check redundancy
S ² I_BIST_ALL1	Scan matrix for ALL1 pattern and check redundancy
S ² I_BIST_CHKB	Scan matrix for CHKB pattern and check redundancy
S ² I_BIST_INVCHKB	Scan matrix for INVCHKB pattern and check redundancy
S ² I_BIST_DIAG0	Scan matrix for DIAG0 pattern and check redundancy
S ² I_BIST_DIAG1	Scan matrix for DIAG1 pattern and check redundancy
S ² I_BIST_PG_ALL0	Program matrix (ALL0 pattern)
S ² I_BIST_PG_CHKB	Program matrix (CHKB pattern)
S ² I_BIST_PG_INVCHKB	Program matrix (INVCHKB pattern)
S ² I_BIST_PG_DIAG0	Program matrix (DIAG0 pattern)

S ² I_BIST_PG_DIAG1	Program matrix (DIAG1 pattern)
S ² I_BIST_ER_MAT	Erase matrix
S ² I_BIST_PG_CAM	Program fuse array (store pattern in BIST registers)
S ² I_BIST_DISAB_CAM	Disable unused fuses

Command Table

Referring now to FIG. 5, a block diagram shows the S²I_LOGIC block 36 in more detail. The clock input 102 and I/O input 112 are used to operate shift register 120. In one embodiment of the present invention, shift register 120 is formed using 25 flip-flops. Shift register 120 is used to acquire from the serial I/O input 112 the instruction code (20 bits), the address (25 bits), and the data (16 bits) to be programmed.

Counter 122 is a 6-bit counter in one embodiment of the present invention, and is also driven from clock input 102. The output of counter 122 is presented to counter mask 124. Three outputs Cnt_25 (reference numeral 126), Cnt_20 (reference numeral 128), and Cnt_16 (reference numeral 130) developed in counter mask 124, indicate the complete acquisition of the instruction code, address, and data, respectively.

Input decoder 132 decodes the 20-bit instruction code provided from shift register 120 and defines which of the operations defined by the commands shown in the command table above is to be performed. A state machine formed from combinatorial logic 144 and status flip-flop 134 determines the future states based on the current state and the status of the inputs of the combinatorial logic 144, which include the decoded command from input decoder 132 on lines 136, the BUSY input 104 of the S²I_LOGIC block, the Cnt_25, Cnt_20, and Cnt_16 outputs of counter mask 124, and the FUSE input 100 of the S²I_LOGIC block. The output of flip-flop 134 includes one bit that drives the enable inputs of shift register 120 and counter 122 on line 138.

Other output bits from status flip-flop 134 on lines 140 drive combinatorial logic 142. Based on the current state and the shift register data, combinatorial logic 142 controls the direction of the data on I/O line 112 of the S²I_LOGIC block by asserting a signal on line 114 of the S²I_LOGIC block, and provides the embedded microcontroller 32 (FIG. 2) with the several inputs discussed with reference to FIG. 4.

Referring now to FIG. 6, a state diagram 150 is shown corresponding to the state machine formed from combinatorial logic 144 and status flip-flop 134. At state 152, the LOAD_COMMAND is issued. If there is a TLATCH SETTING COMMAND, a TLATCH command is issued at state 154, before
-5 returning back to state 152.

If a BIST_CMD is issued at state 152, then a SET BIST REG command is issued at step 156, before proceeding with the BIST at step 158. At state 158, as long as the BUSY signal equals 1 (i.e., the current routine is still being performed), the state machine loops back to state 158. If the BUSY signal equals
10 0 (i.e., the current routine is completed) and there has been a program or erase command, the state machine returns to state 152. If the BUSY signal equals 0 and there has been a verify command, the state machine proceeds to state 160, where the FATAL_ERROR signal is output before returning to state 152.

If a LATCH_ADD_CMD is issued at state 152, then the ENABLE
15 LATCH ADD command is issued at state 162. The state machine then proceeds to state 164, where the INPUT_ADD command is issued until the Cnt_25 value equals 1. If the Cnt_25 value equals 1, the state machine proceeds to state 166, where a LATCH_ADD pulse is implemented before returning to state 152.

If a RESET_FLASH_CMD is issued at step 152, then a RESET pulse is
20 implemented at state 168 before returning to state 152.

While embodiments and applications of this invention have been shown and described, it would be apparent to those skilled in the art that many more modifications than mentioned above are possible without departing from the inventive concepts herein. The invention, therefore, is not to be restricted except
25 in the spirit of the appended claims.

What is claimed is:

1. A flash memory device including:
 - a flash memory array;
 - a set of non-volatile redundancy registers;
 - 5 a serial interface;
 - testing logic coupled to the serial interface and configured to:
 - accept a set of serial commands from an external tester;
 - erase the array;
 - program the array with a test pattern;
 - 10 read the array and compare the results with expected results to identify errors;
 - determine whether the errors can be repaired by substituting a redundant row or column of the array, and if so, generate redundancy information; and
 - 15 program the redundancy information into the non-volatile redundancy registers.
2. The flash memory device of Claim 1, wherein the serial interface comprises a state machine formed from a status flip-flop and a first
20 combinatorial logic block having a plurality of inputs, the state machine configured to determine a future state based on a current state and the plurality of inputs.
3. The flash memory device of Claim 2, wherein the plurality of
25 inputs includes:
 - a busy input that carries a signal indicating whether a routine is currently being performed or is completed;
 - a fuse input that controls the permanent disablement of the serial interface;
 - 30 an instruction input that indicates the complete acquisition of an instruction code;
 - an address input that indicates the complete acquisition of an address;
 - and
 - a data input that indicates the complete acquisition of data.

4. The flash memory device of Claim 1, further comprising a sense amplifier, wherein the testing logic is configured to read data from the array through the sense amplifier.

5

5. The flash memory device of Claim 1, wherein the testing logic further comprises a pattern generator configured to generate the expected results based on commands received from the serial interface.

10

6. The flash memory device of Claim 5, wherein the testing logic further comprises a scrambler circuit configured to receive and scramble data read from the array.

15

7. The flash memory array of Claim 6, wherein the testing logic further comprises a circuit configured to compare the expected results received from the pattern generator with the scrambled data received from the scrambler circuit.

20

8. A method for testing a flash memory array on a flash memory device, the flash memory device including a set of non-volatile redundancy registers, a serial interface, and testing logic coupled to the serial interface, the method comprising the testing logic:

25

accepting a set of serial commands from an external tester;
erasing the array;
programming the array with a test pattern;
reading the array and comparing the results with expected results to identify errors;
determining whether the errors can be repaired by substituting a redundant row or column of the array, and if so, generating redundancy information; and
programming the redundancy information into the non-volatile redundancy registers.

9. The method of Claim 8, wherein the serial interface comprises a state machine formed from a status flip-flop and a first combinatorial logic block having a plurality of inputs, the state machine configured to determine a future state based on a current state and the plurality of inputs.

5

10. The method of Claim 9, wherein the plurality of inputs includes:
a busy input that carries a signal indicating whether a routine is currently being performed or is completed;

10 interface;
a fuse input that controls the permanent disablement of the serial

an instruction input that indicates the complete acquisition of an instruction code;

an address input that indicates the complete acquisition of an address; and

15 a data input that indicates the complete acquisition of data.

11. The method of Claim 8, further comprising a sense amplifier, wherein the testing logic is configured to read data from the array through the sense amplifier.

20

12. The method of Claim 8, wherein the testing logic further comprises a pattern generator configured to generate the expected results based on commands received from the serial interface.

25 13. The method of Claim 12, wherein the testing logic further comprises a scrambler circuit configured to receive and scramble data read from the array.

14. The method of Claim 13, wherein the testing logic further
30 comprises a circuit configured to compare the expected results received from the pattern generator with the scrambled data received from the scrambler circuit.

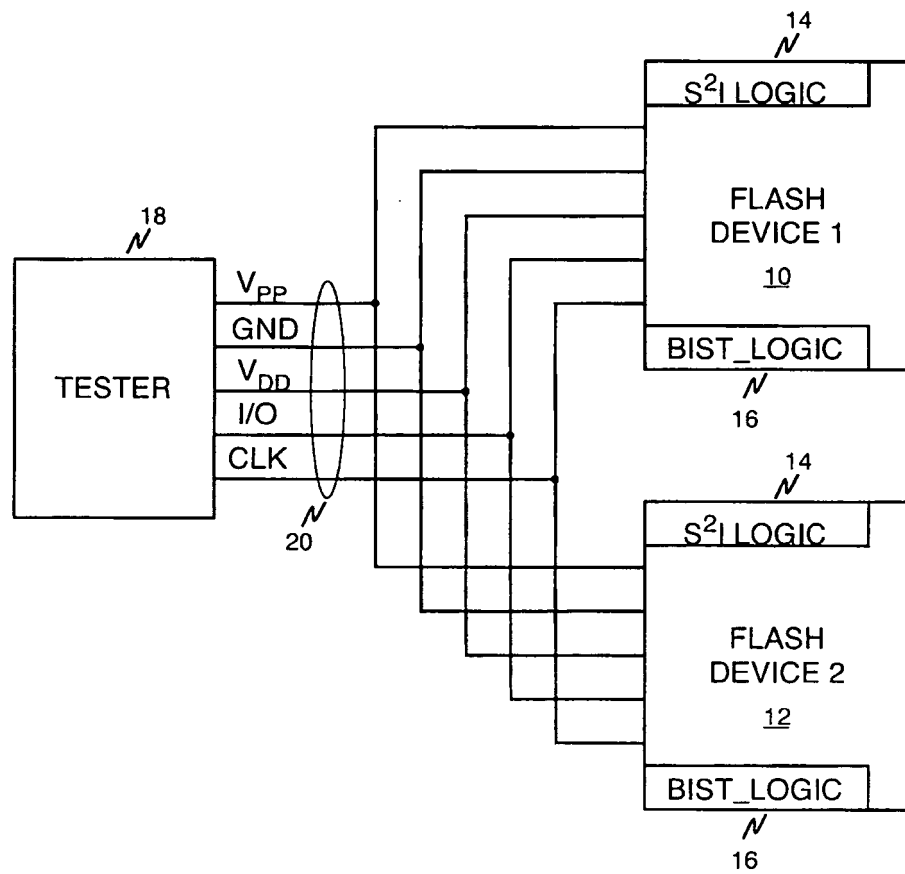


FIGURE 1

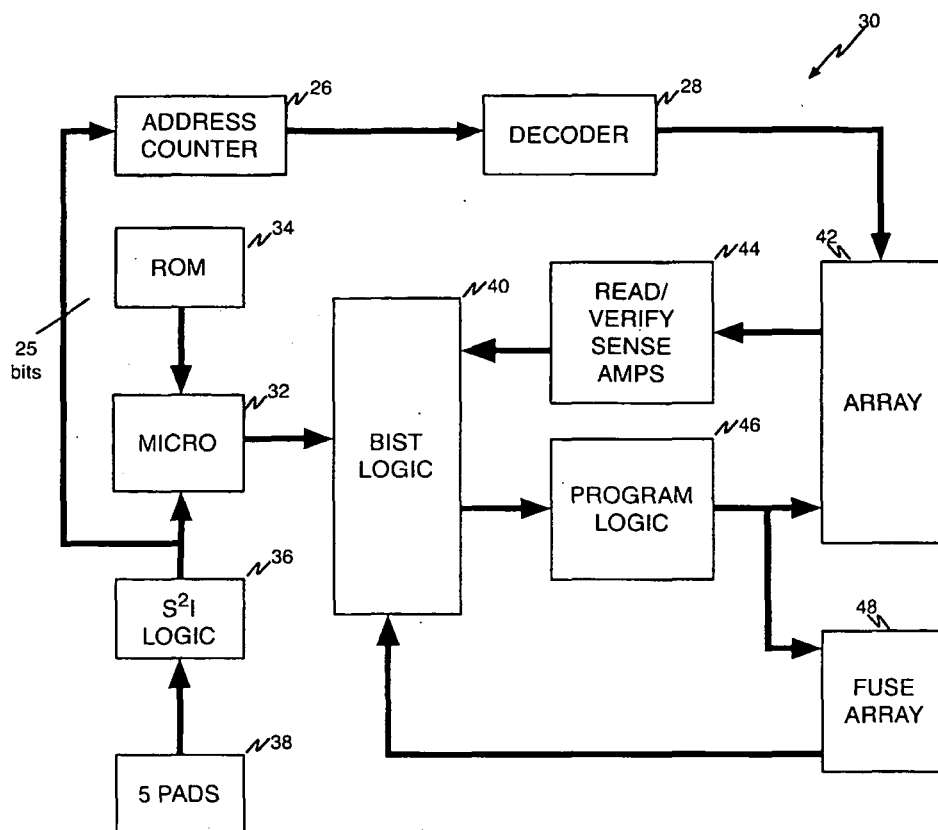


FIGURE 2

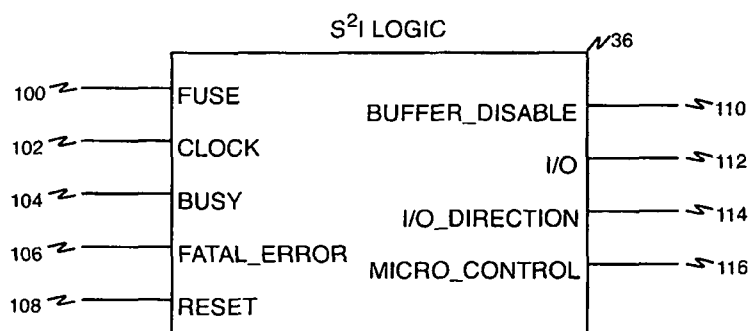
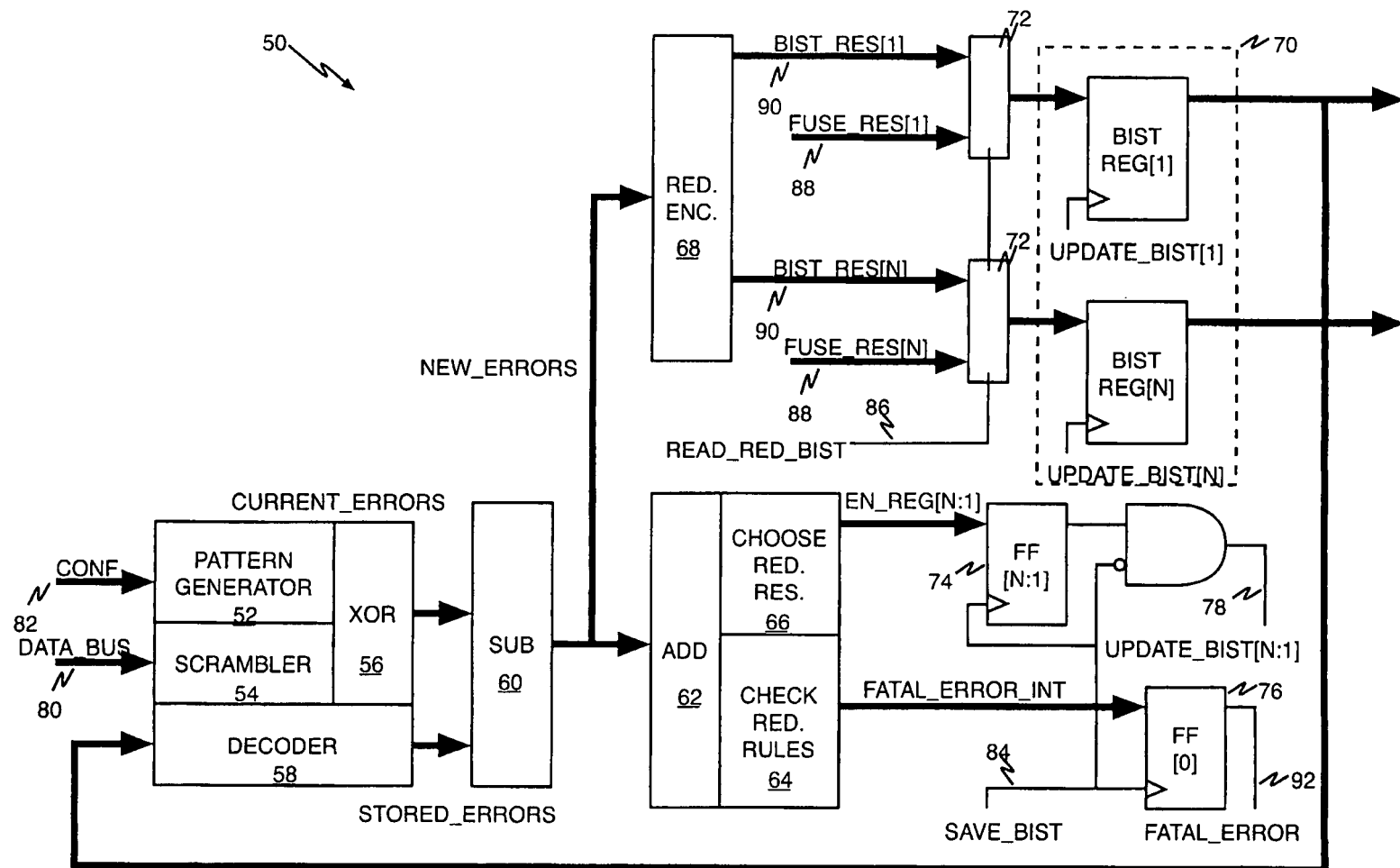


FIGURE 4



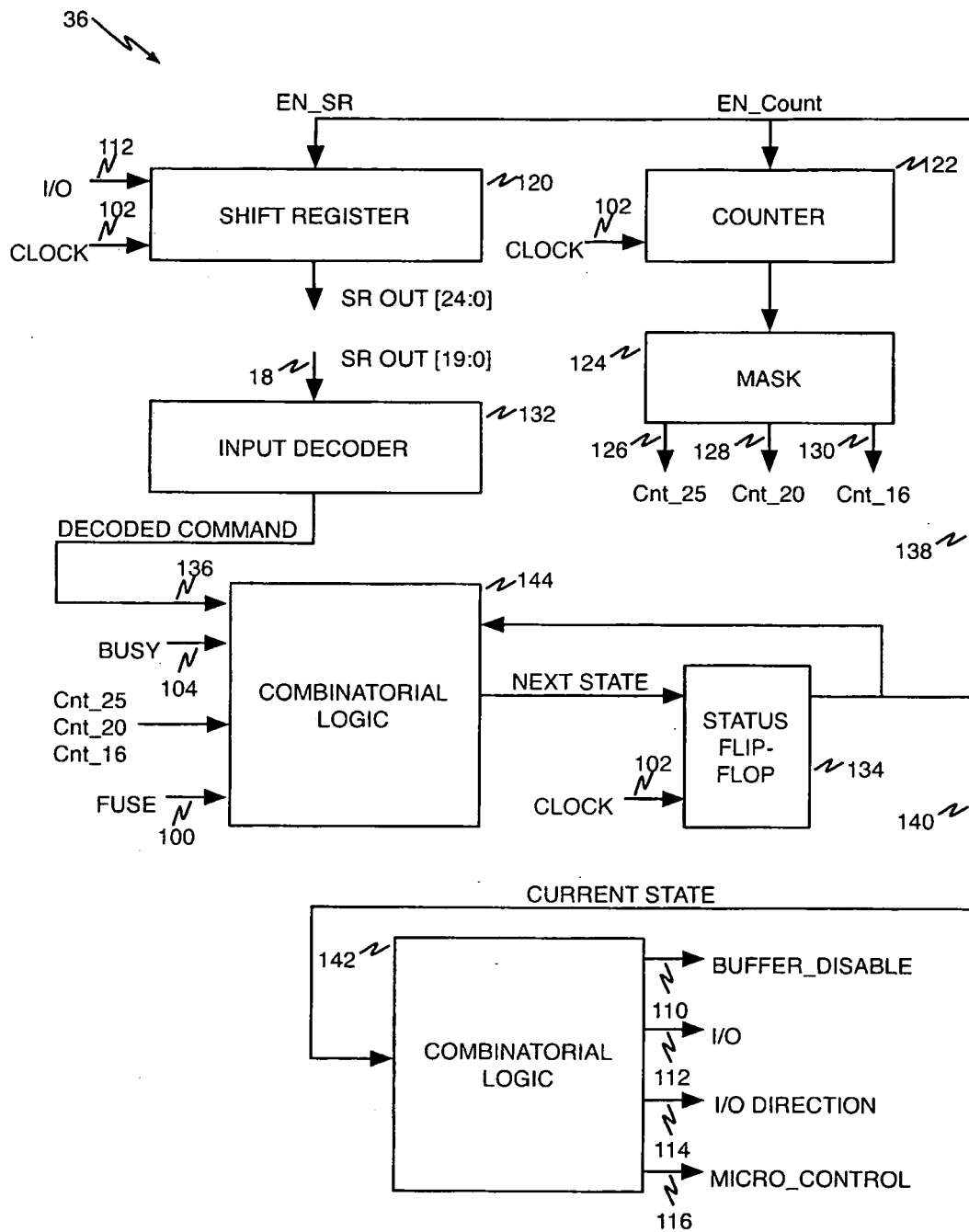


FIGURE 5

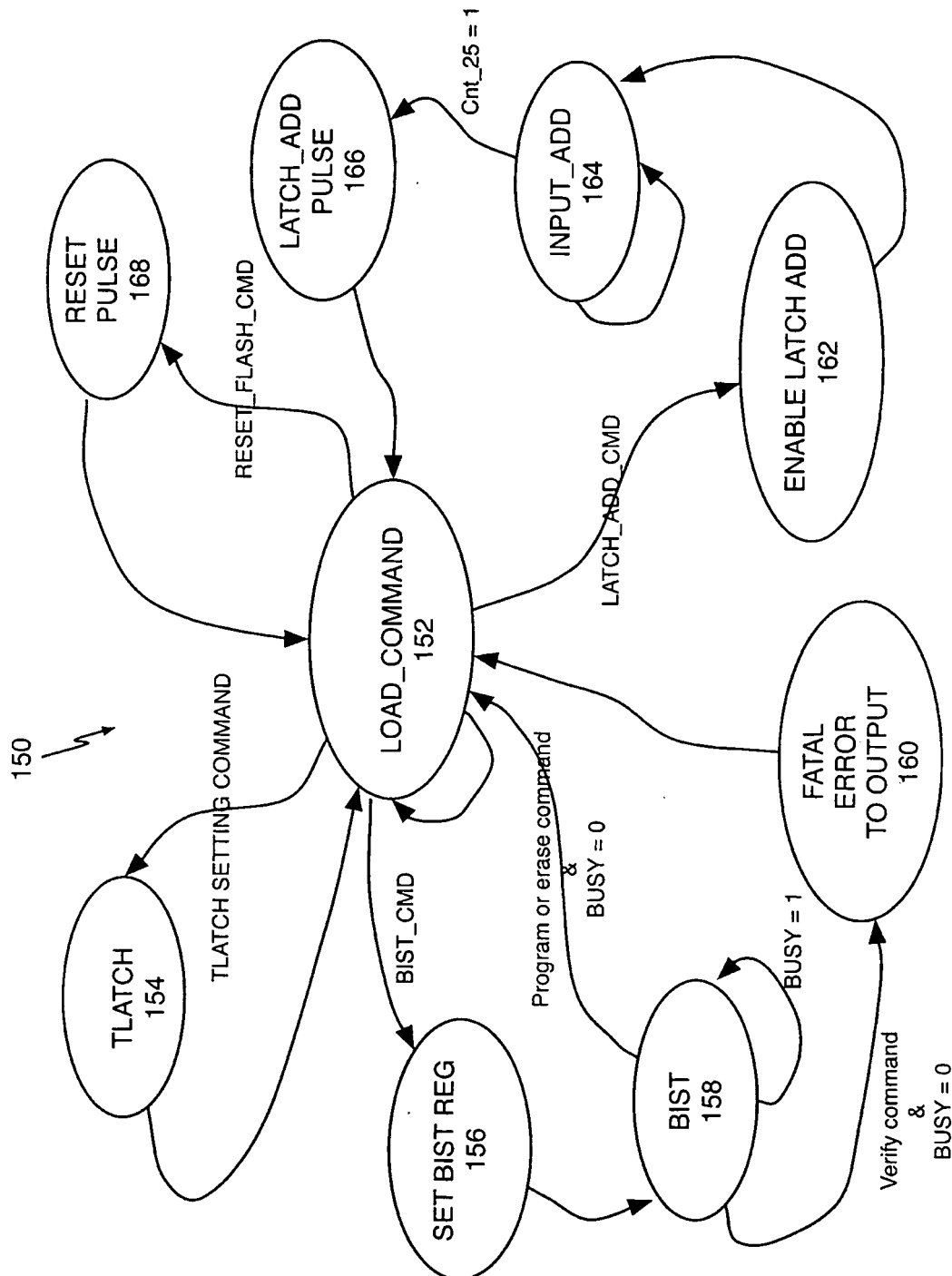


FIGURE 6

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2008/002057

A. CLASSIFICATION OF SUBJECT MATTER

INV. G11C29/12 G11C29/32 G11C29/00

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G11C

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

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

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7 028 240 B1 (BAUTISTA JR EDWARD V [US] ET AL) 11 April 2006 (2006-04-11) figures 6-9, 13, 22, 25, 26, 59 column 10, line 32 - column 14, line 18 column 21, line 51 - column 23, line 33 column 31, line 60 - column 35, line 30 column 73, line 48 - column 74, line 20	1, 2, 4-9, 11-14
Y	-----	3, 10
Y	US 2004/060013 A1 (ISOM MELVIN T [US] ET AL ISOM III MELVIN T [US] ET AL) 25 March 2004 (2004-03-25) paragraphs [0004] - [0007]; figure 1	3, 10
X	US 2002/199139 A1 (DORTU JEAN-MARC [DE] ET AL) 26 December 2002 (2002-12-26) paragraphs [0005] - [0016], [0023], [0024], [0039] - [0041]; figure 1	1, 8
	----- -/--	

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:

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

E earlier document but published on or after the international filing date

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

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

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

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

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

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

Z document member of the same patent family

Date of the actual completion of the international search

13 May 2008

Date of mailing of the international search report

20/06/2008

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Stocken, Christian

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2008/002057

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 7 053 470 B1 (SELLERS SCOTT D [US] ET AL) 30 May 2006 (2006-05-30) column 3, line 29 - column 6, line 59; figures 2-4,7,9	1,8
A	US 2006/168491 A1 (LEE MIMI [US] ET AL LEE MIMI [US] ET AL) 27 July 2006 (2006-07-27) paragraphs [0061] - [0069], [0089], [0090]; figures 3,4,9	1,8

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2008/002057

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 7028240	B1	11-04-2006	NONE	
US 2004060013	A1	25-03-2004	DE 10333527 A1 US 2005034085 A1	01-04-2004 10-02-2005
US 2002199139	A1	26-12-2002	DE 10129771 A1	23-01-2003
US 7053470	B1	30-05-2006	NONE	
US 2006168491	A1	27-07-2006	WO 2006081168 A1	03-08-2006