

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
17 August 2006 (17.08.2006)

PCT

(10) International Publication Number
WO 2006/086232 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/US2006/003876

(22) International Filing Date: 1 February 2006 (01.02.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/651,128 7 February 2005 (07.02.2005) US
11/317,390 22 December 2005 (22.12.2005) US
11/317,862 22 December 2005 (22.12.2005) US

(71) Applicants (for all designated States except US): **SAN-DISK CORPORATION** [US/US]; 140 Caspian Court, Sunnyvale, California 94089 (US). **DISCRETIX TECHNOLOGIES LTD.** [IL/IL]; 43 HAMELACHA ST., ETGARIM BLDG. POLEG INDUSTRIAL Z, P.o. Box 8718, 42502 Netanya (IL).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **HOLTZMAN, Michael** [IL/US]; 7602 Barnhart Place, Cupertino, California 95014 (US). **COHEN, Baruch Boris** [IL/IL]; Aya 15, 21721 Carmiel (IL). **BARZILAI, Ron** [IL/IL]; 67 Meron Street, 25147 Kfar-vradim (IL). **BAR-EL, Hagai** [IL/IL]; Druyan 4, 76574 Rehovot (IL). **DEITCHER, David** [IL/IL]; Mapu 6, Jerusalem (IL).

(74) Agents: **PARSONS, Gerald, P.** et al.; PARSONS, HSUE & DE RUNTZ LLP, 595 Market Street, Suite 1900, San Francisco, California 94105 (US).

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

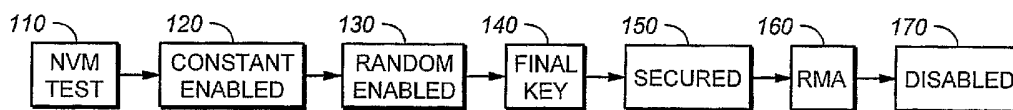
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, 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:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SECURE MEMORY CARD WITH LIFE CYCLE PHASES



(57) Abstract: A secure memory card with encryption capabilities comprises various life cycle states that allow for testing of the hardware and software of the card in certain of the states. The testing mechanisms are disabled in certain other of the states thus closing potential back doors to secure data and cryptographic keys. Controlled availability and generation of the keys required for encryption and decryption of data is such that even if back doors are accessed that previously encrypted data is impossible to decrypt and thus worthless even if a back door is found and maliciously pried open.

SECURE MEMORY CARD WITH LIFE CYCLE PHASES

FIELD OF THE INVENTION

[0001] The invention generally relates to memory cards and encryption, and in particular relates to eliminating the access to secure data and keys through the testing mechanisms in the card.

BACKGROUND OF THE INVENTION

[0002] Quite some time ago an intelligent memory card commonly referred to as the Smart Card was developed and gained acceptance in the marketplace as a form of identification and payment. The Smart Card contains a small amount of memory for storing a user's identification data and for storing transactional related data. The Smart Card is also often referred to as a chip card and is employed in Japan for various things such as the national identity card and in various places as a type of credit or debit card. In order to prevent identity theft and other monetary fraud, various chip designs and encryption schemes have been employed in the cards and the systems that utilize the cards.

[0003] In designing and manufacturing any type of secure memory card, there are two competing interests. One interest is maximizing the security of the card, while the other interest is maximizing the reliability of the card. In order to maximize the reliability of the card, it is important to be able to test the software and the hardware of the card at various manufacturing stages before it ships from the factory, and on some occasions even after it has left the factory in order to perform failure analysis. Testing may involve input and output of signals through test or contact pads on the chip to test both the hardware and software of the card. These test routines and test pads are necessary to ensure quality control but are a potential weak spot or "back door" to the secure data, algorithms, and keys of the card. Thus, there always exists some degree of compromise between (the testing necessary for) maximizing reliability and maximizing security. Different approaches have been put forth to close this "back door" after testing is complete. However, for various reasons prior solutions to date each have commercial and technical shortcomings.

[0004] In one approach, which is believed to be that employed in creation of the aforementioned Smart Card, die of the card is tested before singulation of the memory die from the wafer. The test pads for a particular die are located on an adjacent die of the wafer, and the singulation process severs the test pads from all circuitry of the adjacent die after testing. Therefore, any test pads present on a singulated die are completely isolated and closed as a potential back door to the secure data of the final memory card. However, it is not always practical or desirable to completely remove the test pads. For example, the lack of usable test pads precludes some amount of subsequent hardware based testing of the memory, which, for example, limits the potential methods of failure analysis.

[0005] While this approach may be preferred for a Smart Card, which typically only has a small amount of memory necessary to hold identification and transactional data, it is insufficient to test the comparatively massive amount of memory and complex security routines employed in a mass storage memory card used for storing multiple large files such as photos and music. Some examples of these mass storage memory cards are the Compact Flash card, MMC card, and SD card. The spread of digital content and the associated copyright issues elevate the importance of security, while at the same time the testing and reliability of the card remain paramount. A more comprehensive and flexible system for manufacturing, testing and operating secure mass storage memory cards is needed and is provided by the present invention which will be described below.

[0006] Another important aspect is cost. Several different technologies, such as non-volatile memory, logic, and volatile memory, can be fabricated on a single integrated circuit die (chip). However, mixing different technologies in one die significantly increases the cost of production. In a competitive environment where cost is a major driving force, it is highly desirable to limit the amount of different of technologies provided on one die. However, using multiple die may mean that sensitive information has to pass from one die to another in the final product. This is another potential weakness a hacker can exploit if appropriate precautions are not employed.

[0007] In particular, non volatile memory bits are expensive to mix with logic within the same die. The Smart Card employs non-volatile memory for data storage

purposes in the same die as the logic that runs the Smart Card, which is a way of maximizing security. However, nowadays a memory card that benefits from the present invention must store very large music, photo, movie and other user files. Thus, it is cost prohibitive to manufacture a single integrated circuit die memory card that can store massive amounts of information (on the order of several gigabytes in 2005 and increasing), and it is necessary to develop a secure system employing multiple die. In particular, it is highly desirable to create a secure system (employing encryption and decryption) utilizing one or more discrete (cost effective) flash memory die that are separate from the controller die and that can be thoroughly tested before and after assembly, yet is invulnerable to attacks via the test mechanisms.

SUMMARY OF INVENTION

[0008] Because it is overly costly and presents problems in scalability to utilize a single chip that has both the controller functionality and the massive amounts of storage required by today's digital devices, an alternative system has been developed. With a single chip solution security can be achieved with unique chip design that makes it difficult to access testing mechanisms, encryption keys, and encrypted content. However, with a multiple chip design where content passes from a separate memory chip to a controller chip where encryption occurs, special attention must be paid to guarding access to encryption keys and to encrypted content. Furthermore, in a system that (preferably) still has test pads in the final assembly to allow for testing of the assembled system, special attention must be paid to any mechanisms in software and hardware that may serve as a back door for unauthorized access to the encrypted keys and content.

[0009] The present invention has numerous life cycle phases that are entered and passed through during the life of the card. Depending on the phase, logic in the card enables or disables the encryption engine, controls access to hardware (before and after wafer singulation and card assembly) and software testing mechanisms, and controls key generation. These phases not only allow both the hardware and software of the card to be thoroughly tested before and after manufacture (unlike in the Smart Card where the test pads are removed), but also make it virtually impossible to access the encrypted keys and thus the encrypted content when the card is in a secure phase,

the operating phase that the card is in when it is shipped to the user. Therefore, the present invention provides for a memory card that can be well tested but is also resistant to unauthorized access to protected data within the card.

[0010] Furthermore, a more comprehensive and flexible system for manufacturing, testing and operating secure mass storage memory cards is needed and is provided by the present invention which will be described below.

[0011] Additional aspects, advantages and features of the present invention are included in the following description of exemplary examples thereof, which description should be taken in conjunction with the accompanying figures, wherein like numerals are used to describe the same feature throughout the figures. All patents, patent applications, articles and other publications referenced herein are hereby incorporated herein by this reference in their entirety for all purposes.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] FIG. 1A is a schematic diagram of system 10 according to an embodiment of the present invention.

[0013] FIG. 1B is a schematic diagram of another embodiment of system 10.

[0014] FIG. 2A is a flowchart illustrating the various life cycle phases in an embodiment of the present invention.

[0015] FIG. 2B is a chart of the various life cycle phases.

[0016] FIG. 3 is a flow chart illustrating the boot up process and life cycle phases.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

MEMORY SYSTEM ARCHITECTURE

[0017] An example memory system in which the various aspects of the present invention may be implemented is illustrated by the block diagram of Fig. 1A. As shown in Fig. 1A, the memory system 10 includes a central processing unit (CPU) or controller 12, a buffer management unit (BMU) 14, a host interface module (HIM) 16,

flash interface module (FIM) 18, a flash memory 20 and a peripheral access module 22. Memory system 10 communicates with a host device 24 through a host interface bus 26 and port 26a. The flash memory 20, which may be of the NAND type, provides data storage for the host device 24. The software code for CPU 12 may also be stored in flash memory 20. FIM 18 connects to the flash memory 20 through a flash interface bus 28 and in some instances a port, which is not shown, if the flash memory 20 is a removable component. HIM 16 is suitable for connection to a host system like a digital camera, personal computer, personal digital assistant (PDA) and MP-3 players, cellular telephone or other digital devices. The peripheral access module 22 selects the appropriate controller module such as FIM, HIM, and BMU for communication with the CPU 12. In one embodiment, all of the components of system 10 within the dotted line box may be enclosed in a single unit such as in memory card and preferably enclosed in the card.

[0018] The buffer management unit 14 comprises a host direct memory access unit (HDMA) 32, a flash direct memory access unit (FDMA) 34, an arbiter 36, a CPU bus arbiter 35, registers 33, buffer random access memory (BRAM) 38, and a crypto-engine 40 also referred to as encryption engine 40. The arbiter 36 is a shared bus arbiter so that only one master or initiator (which can be HDMA 32, FDMA 34 or CPU 12) can be active at any time and the slave or target is BRAM 38. The arbiter is responsible for channeling the appropriate initiator request to BRAM 38. HDMA 32 and FDMA 34 are responsible for data transported between HIM 16, FIM 18 and BRAM 38 or the RAM 11. The CPU bus arbiter 35 allows for direct data transfer from crypto-engine 40 and flash DMA 34 to RAM 11 via system bus 15, which is used in certain situations such as for example when it is desired to bypass the crypto-engine. The operation of the HDMA 32 and of the FDMA 34 are conventional and need not be described in detail herein. The BRAM 38 is used to store data passed between the host device 24 and flash memory 20. The HDMA 32 and FDMA 34 are responsible for transferring the data between HIM 16/FIM 18 and BRAM 38 or the CPU RAM 12a and for indicating sector completion.

[0019] When data from flash memory 20 is read by the host device 24, encrypted data in memory 20 is fetched through bus 28, FIM 18, FDMA 34, and crypto-engine 40 where the encrypted data is decrypted and stored in BRAM 38. The decrypted data is

then sent from BRAM 38, through HDMA 32, HIM 16, bus 26 to the host device 24. The data fetched from BRAM 38 may again be encrypted by means of crypto-engine 40 before it is passed to HDMA 32 so that the data sent to the host device 24 is again encrypted but by means of a different key and/or algorithm compared to those whereby the data stored in memory 20 is encrypted. Alternatively, rather than storing decrypted data in BRAM 38 in the above-described process, which data may become vulnerable to unauthorized access, the data from memory 20 may be decrypted and encrypted again by crypto-engine 40 before it is sent to BRAM 38. The encrypted data in BRAM 38 is then sent to host device 24 as before. This illustrates the data stream during a reading process.

[0020] When data is written by host device 24 to memory 20, the direction of the data stream is reversed. For example if unencrypted data is sent by host device, through bus 26, HIM 16, HDMA 32 to crypto-engine 40, such data may be encrypted by engine 40 before it is stored in BRAM 38. Alternatively, unencrypted data may be stored in BRAM 38. The data is then encrypted before it is sent to FDMA 34 on its way to memory 20.

LIFE CYCLE PHASES

[0021] A security system or secure operating system that is particularly useful when implemented in a memory card, such as the one described above, for example, has different phases or states. These phases are preferably entered sequentially, such that after progressing from one phase to the next, the previous phase cannot be re-entered. Therefore, they can be thought of as life cycle phases.

[0022] Before describing the phases in detail, another system level diagram will be briefly discussed. FIG. 1B illustrates another embodiment of system 10. Only certain of the components of system 10 are illustrated in this figure for simplicity and clarity. Memory system 10 comprises test pads also referred to as hardware test input/output (I/O) 54. Hardware bus (HW bus) 56 is preferably connected to test pads 54. These test pads and HW bus 56 are connected to various hardware and circuitry (not shown) of system 10 and are used to test the hardware and circuitry of system 10. JTAG bus 62 is connected the system bus 15 (seen in FIG. 1A) and can be used to replace the controller firmware and drive hardware blocks from outside

system 10. It is used for hardware testing that requires register read/write operations. Since JTAG bus 62 can access the RAM and ROM it is also used to test the firmware of system 10. Host bus 26 is utilized to send diagnostic commands to system 10 and is used to test the firmware of the system.

[0023] NVM 50 of encryption engine 40 is also shown. Stored within NVM 50 are (values for) life cycle state 77 and secret key 99. NVM test port 58 is used to test the NVM within encryption engine 40.

[0024] The state indicator fuse 66 is used to indicate that the product is in NVM state 110 (described below) rather than relying on the NVM content. The reason is that the reliability of an initial value stored in NVM during fabrication cannot be guaranteed. Therefore another more reliable indicator such as a fuse is used. The system will determine that it is in state 110 if the fuse is set. If the system 10 is reset it will look at the NVM life cycle state 77 to determine the state.

[0025] FIG. 2A illustrates the various states and the order of transition between the states. Each state defines different behavior and capabilities of the card (or other system in which it is implemented), before and after the card is manufactured, as can be seen in the following table, which is also reproduced as FIG. 2B.

State	Key Generation	NVM Test	HW Test	FW Test	Crypto-engine
110	Regenerated every power up	E	E	E	D
120	Constant and hard wired	D	E	E	E
130	Generated once	D	E	E	E
140	Generated once	D	D	D	E
150	-	D	D	D	E
160	Regenerated every power up	D	E	E	E
170	-	D	E	E	D

[0026] The state is preferably stored as a 32 bit value within the non volatile memory of the encryption engine. There are 6 pre-assigned values out of a huge number of possible ($\approx 10^9$) combinations that are used to represent states 120-170. All other values are indicative of state 110. This is so because it cannot be guaranteed that a

defined value can be reliably stored during fabrication and retrieved thereafter because various processing operations during fabrication, assembly, testing, and shipping may alter any stored value in memory.

[0027] The key value is also preferably stored as a 128 bit field in the non volatile memory of the encryption engine. The key value is normally generated randomly by a seeded algorithm. Regeneration of the key is highly likely to change the value of the key, but this cannot be guaranteed because a (pseudo) random number generator may in fact generate the same value successively. However, the terminology of changing the key is used interchangeably with that of regenerating the key in this application even though it is well understood the value of the key may not change during regeneration. Needless to say, the value of the key used to encrypt information is critical. The same key value must be used for both encryption and decryption. Thus, if a key value is regenerated at every power up of the system, data that was encrypted before that power up is virtually worthless because it cannot be decrypted with the new key. Although the data is still physically present in the memory of the card, the data is useless without the proper key value to unlock it. Thus, if a hacker manages to somehow force the card back into a state, other than secure state 150, he will not be able to get any worthwhile information. In states 110 and 160, a new key will be generated at every power up and the key used to previously store information in state 150 will not be available to decrypt that information. In states 170 and 110 the encryption engine is simply not available, regardless of the key value.

[0028] Another security measure comprises limiting the availability of firmware and hardware test mechanisms. The system comprises logic that will either enable or disable the mechanisms. The previously described host bus is one of the mechanisms used to test the firmware of the card. The host can issue diagnostic commands over the host bus to test the firmware. The hardware may also be tested when these commands are executed. The hardware is also directly tested over the hardware bus as well as the JTAG port, which provides direct access to various memories of the system. Note that in states 140 and 150 the NVM test mechanisms, HW test mechanisms, and FW test mechanisms are all disabled.

[0029] The states and the passage between the states as seen in FIG. 2A will now be described in further detail.

[0030] State 110 is referred to as the controller non-volatile memory (NVM) test. This state is the initial state after fabrication of the memory die, and is the state that is used to test non volatile memory of the controller die before the die is packaged and installed into the memory card. The testing that occurs in this state may be performed before singulation while the dice are still integral in wafer format, or may alternatively be performed on the individual die after singulation. Once the NVM is tested, its content (using the NVM tester) is initialized to indicate state 120, and fuse 66 is blown. In this state the encryption engine 40 is disabled. This state is only designed to be entered into once in the life cycle of the card and there is no method within the system for returning to this state. However, as discussed previously, this state is indicated by anything other than the 6 pre-assigned values of the many possible combinations of the 32 bit value used to define the life cycle state. If an illegal value is detected and the fuse is blown (not allowing NVM state 110 to be entered) the crypto-engine will never become ready and the system will not boot, or go beyond step 302 described below with regard to FIG. 3. Therefore, each time the card is powered up and is in this state, a new key will be randomly generated, and the previously encrypted data impossible to decrypt. Even though the encryption engine is not enabled in this mode, because the mode is designed to be used while the wafer is still intact during fabrication, the key is still regenerated at every power up to protect against a hacker who may in some unforeseen way enter this state and try to probe the secure data of the card via the various test ports and mechanisms. Otherwise, by design, after exiting state 110, the NVM testing mechanisms are no longer available.

[0031] State 120 is referred to as the constant enabled state. In this state the encryption engine 40 is enabled. The key that the encryption engine will use is not generated by the random number generator, and is not stored in memory, but is hard wired to some external source and constant during this phase. The hardware and software testing mechanisms are available in this state. This state is entered by a hardware tester.

[0032] State 130 is referred to as the random enabled state. This state is similar to state 120, however, the secret key is randomly generated (once) when state 130 is entered instead of being constant and hard wired. This is the state used for final testing, characterization, and qualification of the memory card. Cryptographic operations including encryption and decryption are possible with the firmware using a secret key or a key derived from the secret key. This state is entered by code that is loaded into system 10 by host device 24 and then executed by system 10.

[0033] State 140 is referred to as the final key state. In this state the card uses the final secret key that will be shipped with the card. The hardware and software test mechanisms are disabled by the card logic and cannot be accessed. This includes the hardware test bus and test pads, seen in FIG. 1B. This state is used to load the card with the final firmware and configuration data that needs to be protected with the key the product is shipped with. The product can be configured in this state, whereas in state 150 it cannot. This state is entered by a host command. The command may be contained in code that is downloaded from the host and executed by the card ("DLE code"). The command may alternatively be issued directly from the host. This is true any time the term DLE code is utilized below.

[0034] State 150 is referred to as the secure state. This is the state in which the card is shipped from the factory. The hardware and software test mechanisms are disabled by the card logic and cannot be accessed. This state is entered at the end of testing and configuration of the product on the manufacturing floor. The key is not regenerated and the value that was stored in memory during state 140 is utilized during state 150. While derivative keys may be utilized for various operations of the card, the key 99 will always be necessary to derive those keys and to encrypt and decrypt data. This key is meant to be utilized for the life of the *secure* card (while in the hands of the consumer as a secure card, not after). The firmware in the card cannot use the secret key for any operation. It is the hardware of the encryption engine that is responsible for performing all encryption and decryption within the card. This state is entered by DLE code.

[0035] State 160 is referred to as the returned merchandise authorization or RMA state. This state is designed to allow testing of a card that has been returned by a

consumer because it is not working properly. This is the state in which failure analysis of the card can be performed. The software and hardware test mechanisms are again available. It is important to note that this state is only accessible by the factory. Furthermore, after the RMA state is entered, the card can never again be used as a secure card. In other words, it can never again enter state 150 or otherwise be used to decrypt information resident on the card or to save encrypted information to the card. The secret key is regenerated when this mode is entered and during every chip reset performed while the card is in this state. Operation using the secret key for decryption is enabled only at boot time and the firmware cannot use the secret key for any operations. This state is entered by a ROM code that is the result of a host command.

[0036] State 170 is referred to as the disabled state. In the disabled state, the crypto-engine 40 is in bypass mode with all of the cryptographic abilities disabled. Only non-secure algorithms are used within the card. Hardware and software test mechanisms are again enabled because without the encryption engine there is nothing worthy of being hacked or otherwise tampered with. Any encrypted information can no longer be decrypted and is rendered worthless. Also, no additional information may be encrypted and subsequently decrypted. This state may be used to produce a non-secure or "regular" card. In this way, the same system can be used to produce both secure and non-secure memory cards. The difference is that in the non-secure card the security system of the card is in the disabled state, or the card can more generally be said to be in state 170. The disabled state can also be used to re-ship a product that has been sent back to the factory for failure analysis, and has therefore been passed into RMA state 160. As mentioned above, after a card enters into RMA state 160, it can never return to any of the previous states, and may never be sold again as a secure card. However, a card that is functional or can be made functional again at the factory can be placed into disabled state 170 and re-sold as a non-secure card. In this way, the card can be salvaged and would for all intensive purposes be the same as a new non-secure or "regular" card. Both the salvaged non-secure card and a new non secure card will be running the same firmware in the same state.

[0037] Currently, the vast majority of cards are non-secure cards. While the drive to bring to market secure cards is high due mainly to the demands of content providers,

it is unclear what percentage of future memory card sales will be for secure cards vs. non-secure cards. What is clear is that there should likely always be an abundance of non secure content and therefore a demand for non-secure cards. The present invention not only enables testing of all of the hardware and software of a secure card (by *only* authorized personnel), but also provides the ability to salvage returned secure cards for various non-secure uses. Moreover, the system of the present invention allows for a card that has robust security, but that need not be discarded or have its security system compromised (with accessible “back doors”) in order to perform failure analysis. Given the widespread and increasing proliferation of devices that use memory cards, the ability to salvage what would otherwise be a defective secure card is a great benefit to the consumer and manufacturer alike.

[0038] Figure 3 illustrates the booting process for a memory card implementing the system described above. For more information on the boot up process please refer to a co-pending application “Method of Hardware Driver Integrity Check Of Memory Card Controller Firmware” to Micky Holtzman et al., Application No. 11/284,623, Attorney Docket No. SNDK.408US1, hereby incorporated by reference in its entirety.

[0039] In step 302, the system checks if the cryptographic hardware, including crypto-engine 40 and other components, is ready. The system will wait to proceed until the hardware is ready. When the hardware is ready the system advances to step 304. In step 304 the system checks to see if the card is in state 170, the disabled state. If the card is in state 170, in step 306 the system will upload the boot loader (“BLR”) which is a minimal amount of startup code, from flash memory 20 to RAM 11. Next, in step 308 the system checks to see if the BLR was properly uploaded. If so, in step 310 the system will upload the firmware necessary to run in non-secure mode (the standard firmware minus the cryptographic functionality). If the BLR was not properly uploaded in as determined in step 308, the system will advance to step 324 described below.

[0040] If in step 304 the system determined that the card was not in state 170, the system will clear the RAM contents in step 312. After that the system will again check to see what state the card is in step 314. If the card is in state 120, 130, or 140 the BLR will be uploaded in step 316. In step 318 the system will check to see if the

BLR was properly uploaded. Next, in step 320 an integrity check of the BLR code will be performed. This integrity check is a hardware based check performed by calculating message authentication code (MAC) values and comparing them with reference values. The result of the integrity check is a simple flag stored in memory. In step 322 the firmware checks the flag to see if the integrity was verified or not. If the integrity is OK, the system will then in step 342 upload the firmware necessary to run in secure mode, which also of course allows for non secure data to be stored and retrieved. If the integrity is not OK as determined in step 322, the system will wait for a diagnostic command from the host to download and execute certain instructions from the host (DLE command), as is represented by step 324. If a DLE command is received, as seen in step 326, the system will proceed to load the DLE code into RAM in step 328. In step 330 the DLE code will be executed by the controller.

[0041] If in step 314 it was determined that the card was not in state 120, 130, or 140 the system will check in step 332 to see if the card is in state 150. If so, the system will then upload the BLR in step 334. This is done by the ROM code. If the BLR upload was OK, as determined in step 336, a hardware based integrity check, as described above in step 320, will be performed in step 338. After this hardware based integrity check, another integrity check, this time a software based integrity check will be performed in step 340. If the integrity is OK, the system will then in step 342 upload the firmware necessary to run in secure mode, which also of course allows for non secure data to be stored and retrieved.

[0042] If in step 332 it was determined that the card was not in state 150, the system will then check the state of card and if the card is in state 160 and if so it will wait for a diagnostic command as represented by step 348. If, however, in step 344 it is determined that the card was not in state 160, the system will wait for a command to go into RMA state 160, as seen in step 346.

IT IS CLAIMED:

1. A method of making a memory card, comprising:
providing a memory die with test ports;
providing a security system comprising:
a cryptographic engine;
a cryptographic key;
a testing state wherein the testing ports are available for hardware testing and software test routines are available that enable operation of the cryptographic engine and testing of the software of the memory card;
an operating state wherein the testing ports are unavailable and wherein the software test routines are unavailable;
testing the memory card software by utilizing the testing state; and
providing a command from a host to the card that irreversibly switches the card from the testing state to the operating state.
2. A method of testing the functionality of a memory card comprising a non volatile memory die having test pads, the method comprising:
testing the memory die before any firmware is loaded into it and before it is part of the memory card;
loading the firmware into the memory of the memory die;
changing the phase of the card to a first phase that allows testing of the hardware and software of the memory card; and thereafter
testing the hardware and software of the memory card; and thereafter

changing the phase of the card to a second phase that does not allow testing of the hardware and software of the memory card.

3. The method of claim 2, wherein the test pads are still present on the memory die when the die is installed in the card.

4. The method of claim 2, further comprising, after a failure of the card, changing the phase of the card to a phase that allows testing of the software and hardware of the memory card for failure analysis purposes, wherein, after entering this phase the memory card is no longer able to decrypt data that was encrypted prior to entering this phase.

5. The method of claim 2, wherein an indicator of the phase is stored in a non volatile memory of the controller chip.

6. The method of claim 2, wherein in the second phase signals are not transferred to and from the test pads.

7. A method for providing secure storage of data in a memory card, the method comprising:

providing flash memory within the memory card;

providing a controller chip with an encryption engine that encrypts and decrypts data that is stored in the flash memory;

storing a key within non volatile memory of the controller chip;

encrypting a user of the card's data file with the key before the file is stored in the flash memory chip;

storing the encrypted data file in the flash memory; and

if usage of card testing mechanisms is desirable, regenerating the value of the key such that the encrypted file cannot be decrypted.

8. A security system implemented in a memory card operable either as a secure card or as a regular card, comprising:

a cryptographic engine that encrypts and decrypts information in order to safeguard information stored on the card;

a secret key in the card used for the encryption and decryption;

a testing state wherein:

the encryption engine is enabled;

the secret key is generated upon entering the testing state;

firmware in the card may be tested;

hardware of the card may be tested;

a secure operating state wherein:

firmware in the card cannot be tested;

hardware of the card cannot be tested;

the encryption engine is enabled;

the secret key is used by the encryption engine to encrypt and decrypt information; and

a bypass operating state wherein:

the encryption engine is disabled;

hardware of the card can be tested;

firmware of the card can be tested;

the bypass operating state is used if operating as a regular card.

9. The system of claim 8, wherein the card is sold initially as a non-secure card configured in the bypass operating state.

10. The system of claim 8, wherein the card is sold initially as a secure card configured in the secure operating state.

11. The system of claim 10, wherein the card initially sold as a secure card, upon return to the manufacturer, is put into the bypass operating state.

12. The system of claim 11, wherein the card after being put into the bypass state is sold as a non-secure card.

13. The system of claim 8, wherein the encryption engine is hardware based.

14. The system of claim 8, wherein the secret key is stored in a non volatile memory of the encryption engine.

15. The system of claim 8, further comprising:

a failure analysis operating state wherein:

hardware of the card can be tested;

firmware of the card can be tested; and

the secret key is regenerated every time the system is powered up in the failure analysis state.

16. The system of claim 15, wherein after entering the failure analysis operating state the card can never again be used in the secure operating state.

17. The system of claim 16, wherein the failure analysis operating state is activated before entering the bypass state.

18. A system for testing a memory card having firmware, comprising:

test ports on an integrated circuit die;

testing mechanisms in firmware; and

a security system comprising:

a cryptographic engine;

one or more testing states of operation available during testing of the card, the testing states allowing usage of the test ports and of the testing mechanisms in the firmware;

a secure state of operation available during normal usage of the card, the secure state prohibiting usage of the test ports and of the testing mechanisms in the firmware;

logic for switching between a testing state of the one or more testing states and the secure state, wherein once the card is switched into the secure state the one or more testing states can never again be accessed, thereby prohibiting access to secure data on the card by preventing access to the testing mechanisms in the firmware and to the test ports.

19. The system of claim 18, wherein the security system further comprises a bypass operating state, wherein the cryptographic engine is disabled.

20. The system of claim 19, wherein the bypass state is used to produce a non-secure memory card.

21. The system of claim 19, wherein the bypass state is activated when a secure card is returned for service, thereby transforming the secure card into a non-secure card and salvaging the card for further usage.

22. The system of claim 21, wherein the non-secure card can never again be used in a secure state after transformation.

23. The system of claim 19, further comprising a failure analysis state wherein a new key used to encrypt and decrypt data is generated at every power-up, and wherein data encrypted with one key cannot be decrypted with another key.

24. The system of claim 23, wherein the failure analysis state is entered before the bypass state can be entered.

25. The system of claim 18, further comprising a final key state used to load the card with the final firmware to be used in the secure state, the testing mechanisms in the firmware disabled in the final key state.

26. The system of claim 25, wherein in the final key state communication with the test ports is unavailable.

27. A memory card comprising:

a security system implemented in the memory card that can be used to produce either a secure card or a non-secure card;

a cryptographic engine designed to encrypt and decrypt information in order to safeguard information stored in the card;

a testing state wherein:

the encryption engine is enabled;

firmware in the card may be tested;

hardware of the card may be tested;

a secure operating state wherein:

firmware in the card cannot be tested;

hardware of the card cannot be tested;

the encryption engine is enabled;

a bypass operating state wherein:

the encryption engine is disabled.

28. The memory card of claim 27, wherein in the bypass operating state the:

hardware of the card can be tested; and

firmware of the card can be tested.

29. The memory card of claim 27, further comprising a final key state wherein:
- a final key to be used in the secure operating state is utilized in the card;
 - the card can be configured;
 - hardware test mechanisms of the card cannot be tested; and
 - software test mechanisms of the card cannot be tested.
30. A flash memory card comprising:
- a security system comprising:
 - two or more operating states, the card operating in one operating state at a time;
 - means for encrypting and decrypting data within the card;
 - means for testing the card while the card is in a first of the operating states;
 - means for advancing the state of the card, such that a previous state cannot be entered again after the state is advanced;
 - in one of the two or more operating states, means for controlling access to the data of the card, such that the means for encrypting and decrypting data are available but the means for testing the card are unavailable in the operating state, thereby preventing un-authorized access to the data within the card via the means for testing.
31. A flash memory card comprising:
- a security system comprising:

two or more operating states, the card operating in one operating state at a time;

an encryption engine that encrypts and decrypts data within the card;

testing mechanisms available while the card is in a first of the operating states;

logic that advances the state of the card, such that a previous state cannot be entered again after the state is advanced; and

wherein data encrypted in at least one of the states cannot be decrypted in a subsequently entered state.

32. A flash memory card comprising:

a first integrated circuit die comprising non volatile flash memory for mass storage of data files; and

a second integrated circuit die comprising:

a microprocessor;

an encryption engine that encrypts and decrypts data;

non volatile memory;

a secret key that enables operation of the encryption engine, said secret key stored in the non volatile memory of the second integrated circuit die,

wherein in a normal operating mode the memory card utilizes a first value of the secret key for encryption and decryption, and

wherein in a testing mode the memory card utilizes second value of the secret key for encryption and decryption, and

wherein the second value cannot be used to decrypt data encrypted with the first value, and

wherein the first value cannot be used to decrypt data encrypted with the second value.

33. A flash memory card comprising:

an encryption engine that encrypts and decrypts data stored in the card;

a first state of operation for testing the memory card, wherein the memory card generates a first key that enables encryption and decryption using the encryption engine;

a second state of operation for consumer usage of the memory card, wherein the memory card utilizes a second key to encrypt and decrypt data stored in the card, and

wherein data encrypted with the first key cannot be decrypted with the second key and data encrypted with the second key cannot be decrypted with the first key, and

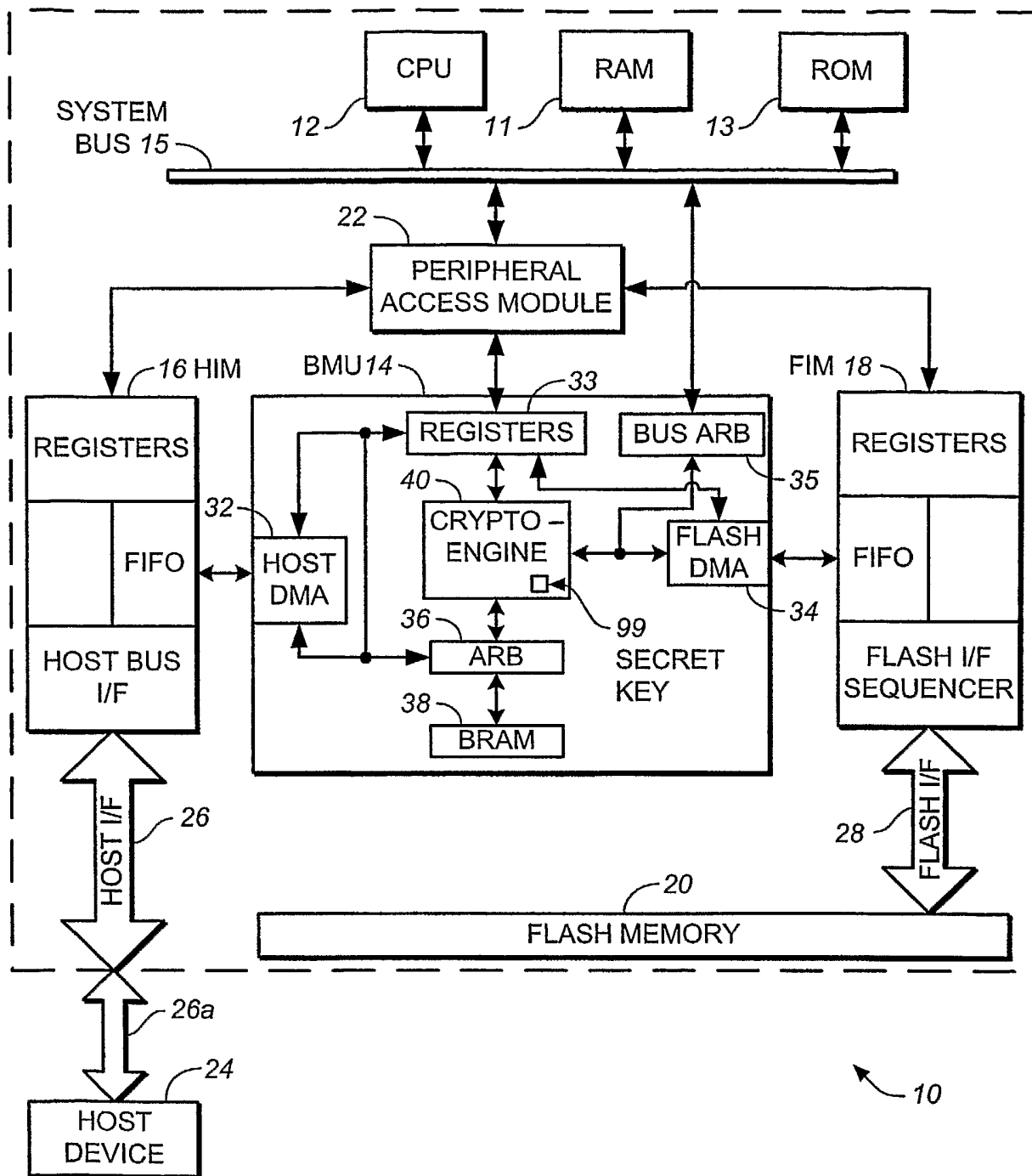
wherein the encryption engine can be tested in the first state but data encrypted while the card is in the first state cannot be read while the card is in the second state.

34. The memory card of claim 33, wherein the value of first key changes each time the memory card is powered up.

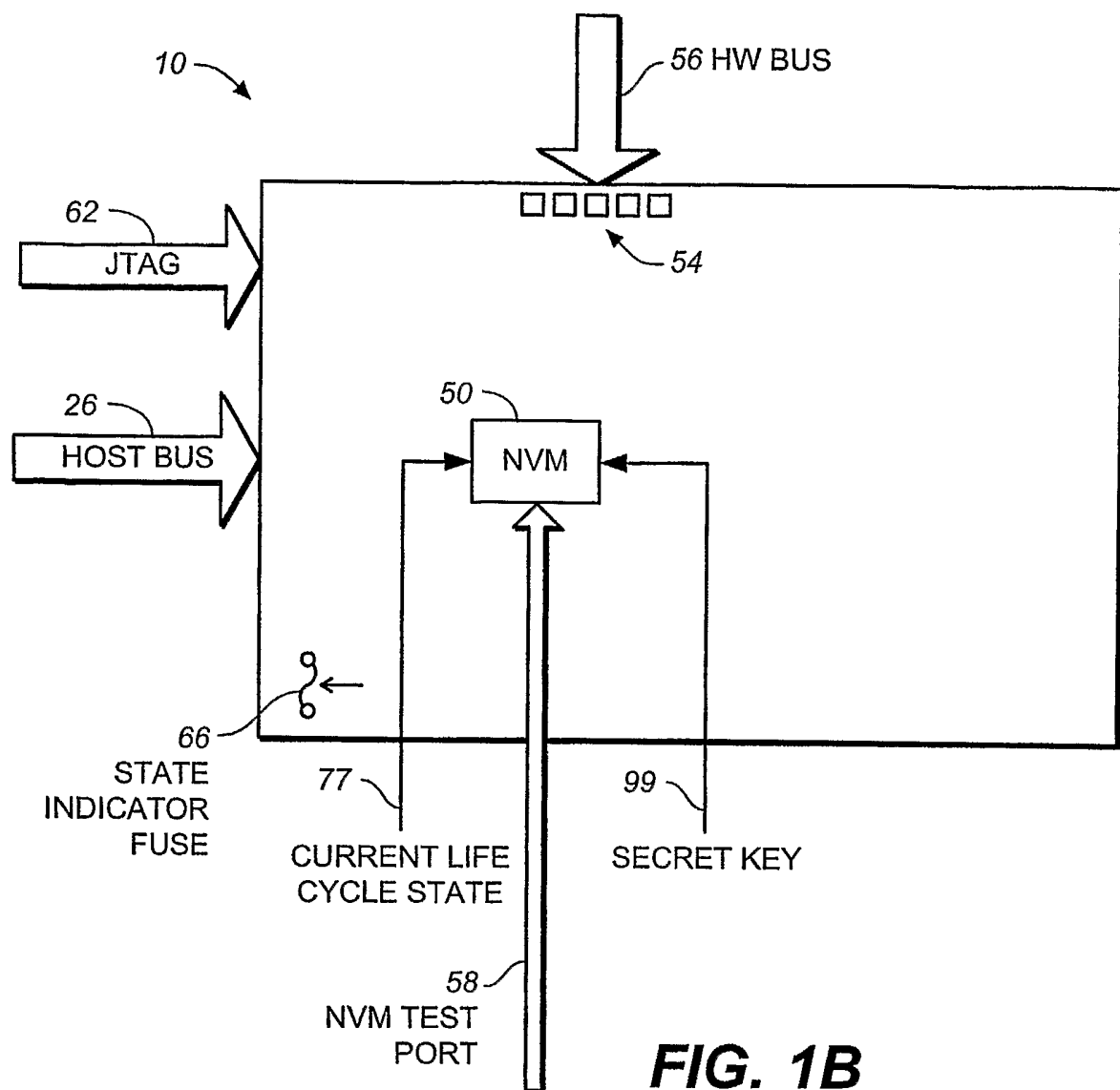
35. The memory card of claim 33, wherein the value of the first key is constant.

36. The memory card of claim 33, wherein the value of the first key is regenerated once each time the first state of operation is entered.
37. The memory card of claim 33, wherein the value of the second key is constant and stored in a non volatile memory of the card.
38. The memory card of claim 37, wherein the value of the second key is the equal to the value of the first key last stored in memory during the first state.
39. The memory card of claim 33, wherein hardware and firmware test mechanisms are available for testing in the first state.
40. The memory card of claim 33, wherein hardware and firmware test mechanisms are not available for testing in the second state.

1 / 4

**FIG. 1A**

2 / 4

**FIG. 1B**

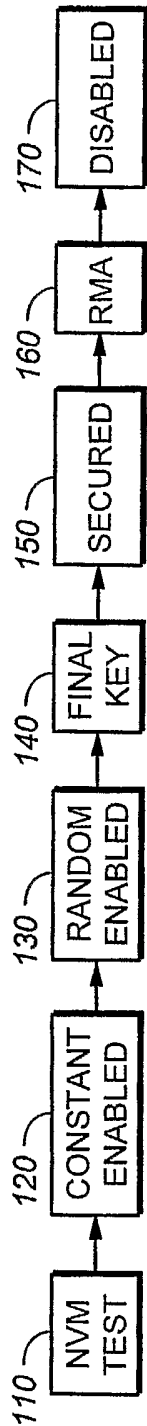


FIG. 2A

State	Key Generation	NVM Test	HW Test	FW Test	Crypto-Engine
110	Regenerated every power up	E	E	E	D
120	Constant and hard wired	D	E	E	E
130	Generated once	D	E	E	E
140	Generated once	D	D	D	E
150	-	D	D	D	E
160	Regenerated every power up	D	E	E	E
170	-	D	E	E	D

FIG. 2B

FIG. 3

