US 20130297948A1

(19) **United States**

(12) **Patent Application Publication**

LEE et al.

(10) **Pub. No.: US 2013/0297948 A1**

(43) **Pub. Date: Nov. 7, 2013**

(54) SYSTEM ON CHIP, METHOD OF OPERATING THE SAME, AND DEVICES INCLUDING THE SYSTEM ON CHIP

(71) Applicant: **SAMSUNG ELECTRONICS CO., LTD.**, Suwon-si (KR)

(72) Inventors: **Heon Soo LEE**, Hwaseong-si (KR); **Hong-Mook CHOI**, Hwaseong-si (KR); **Sang-Hyun PARK**, Hwaseong-si (KR)

(73) Assignee: **SAMSUNG ELECTRONIC CO., LTD.**, Suwon-si (KR)

(21) Appl. No.: **13/718,382**

(22) Filed: **Dec. 18, 2012**

(30) **Foreign Application Priority Data**

May 4, 2012 (KR) 10-2012-0047743

Publication Classification

(51) **Int. Cl.**

G06F 21/60 (2006.01)

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

CPC ***G06F 21/602*** (2013.01)

USPC **713/193**

(57) **ABSTRACT**

A method of operating a system on chip (SoC) includes converting plain data into cipher data by using an encryption key and transmitting the cipher data directly to a memory controller which controls an operation of a non-volatile memory. The encryption key may be output by a one-time programmable (OTP) memory.

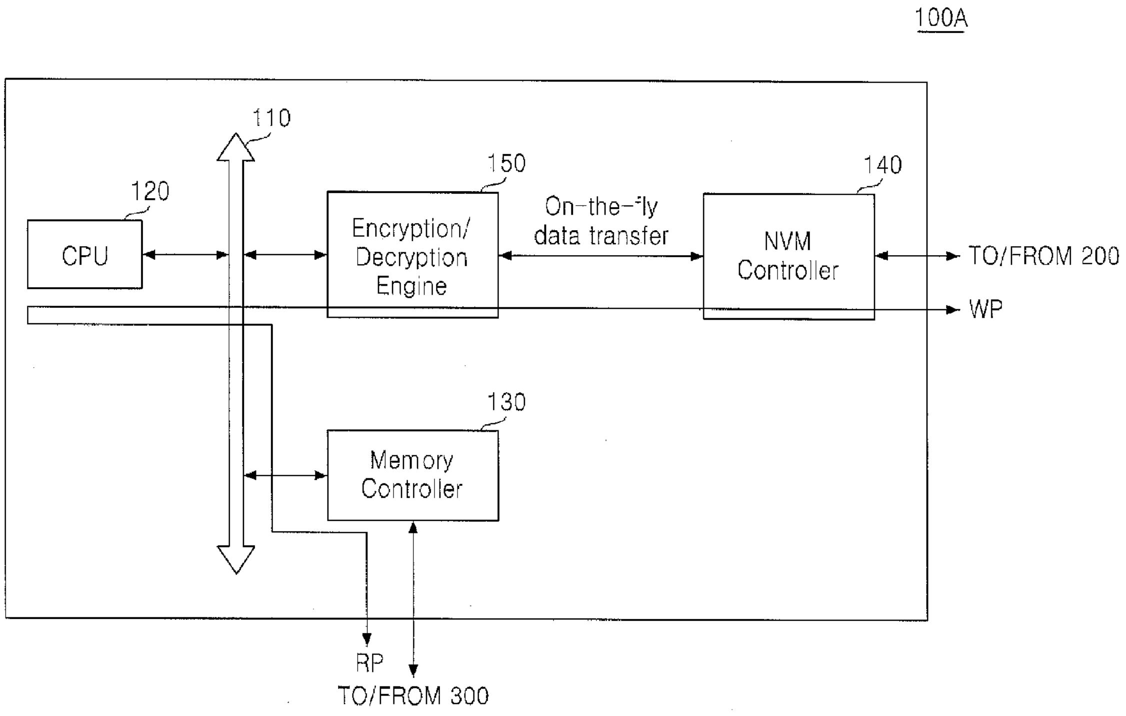


FIG. 1

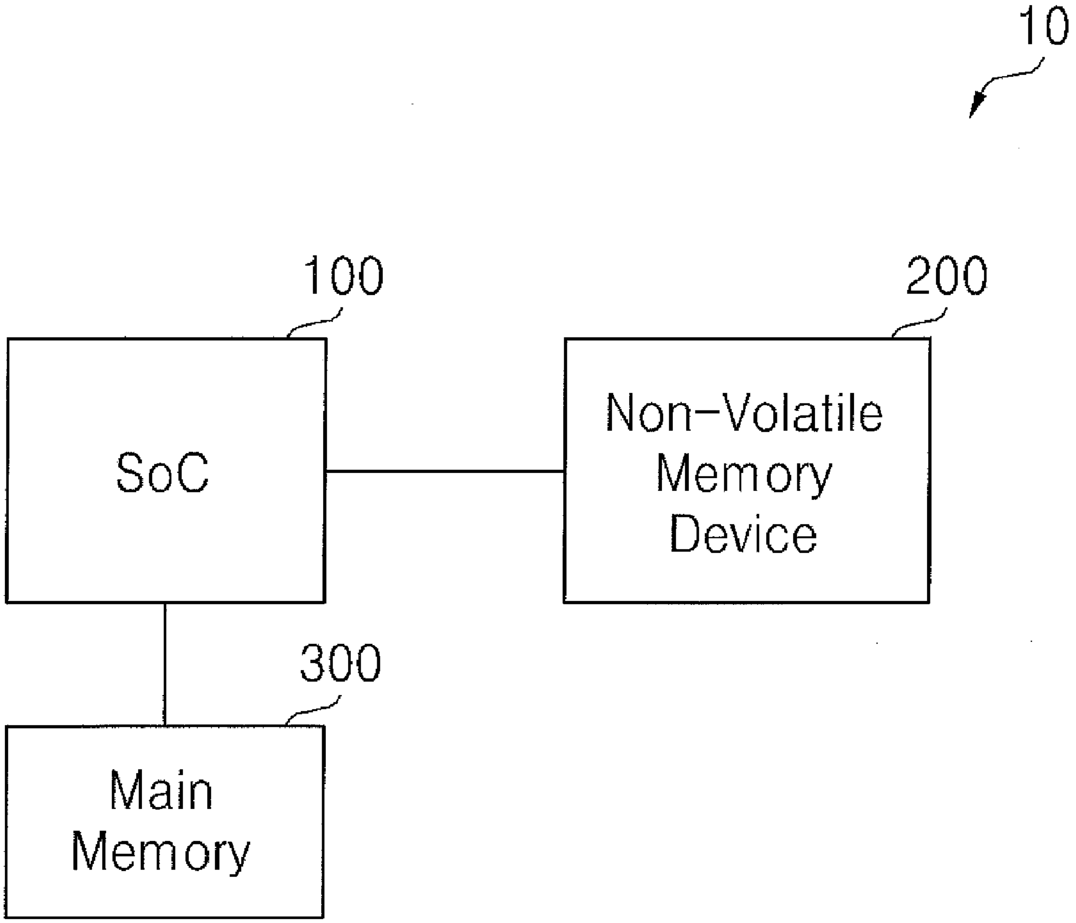


FIG. 2

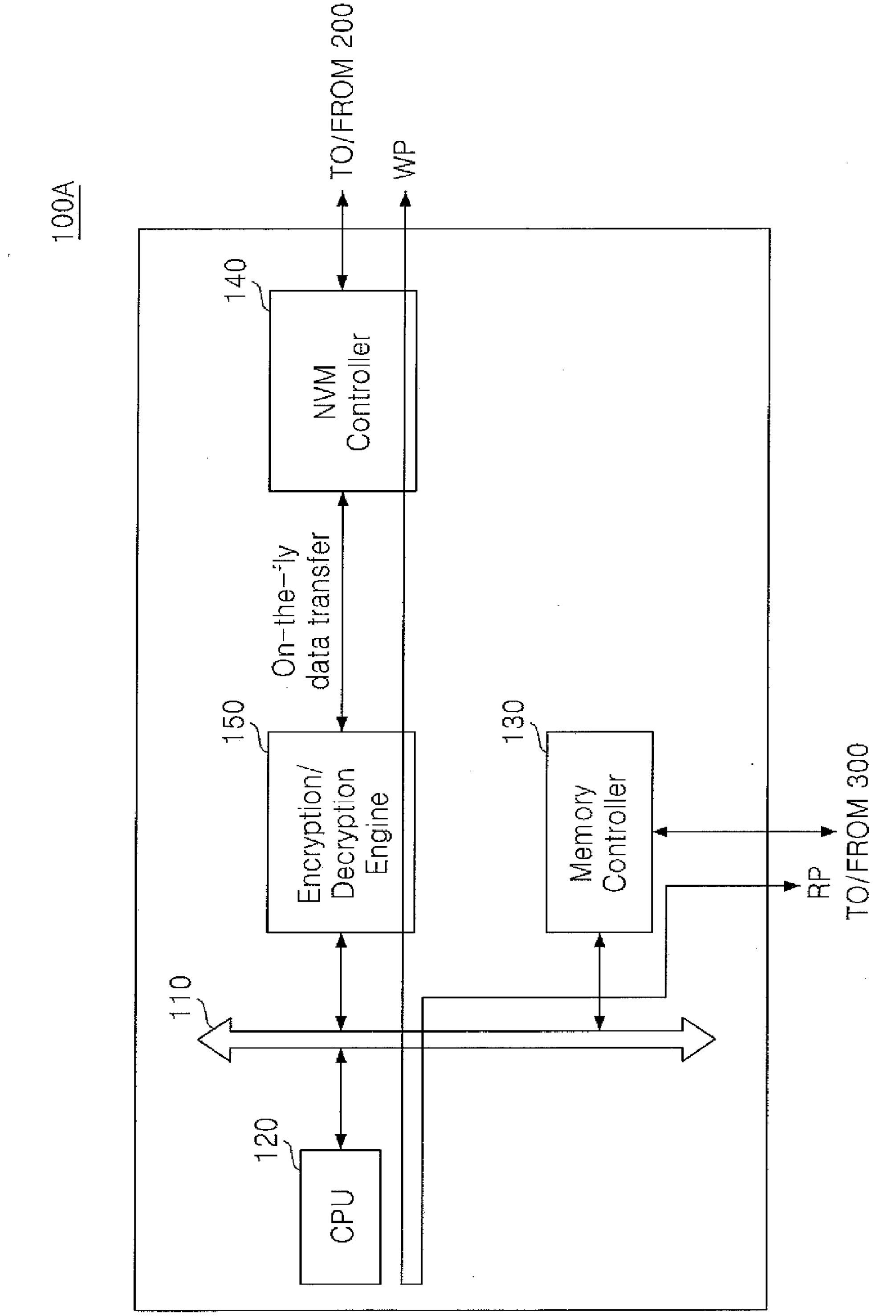

100A
TO/FROM 200
WP
140
NVM Controller
On-the-fly data transfer
150
Encryption/ Decryption Engine
130
Memory Controller
TO/FROM 300
RP
110
120
CPU

FIG. 3

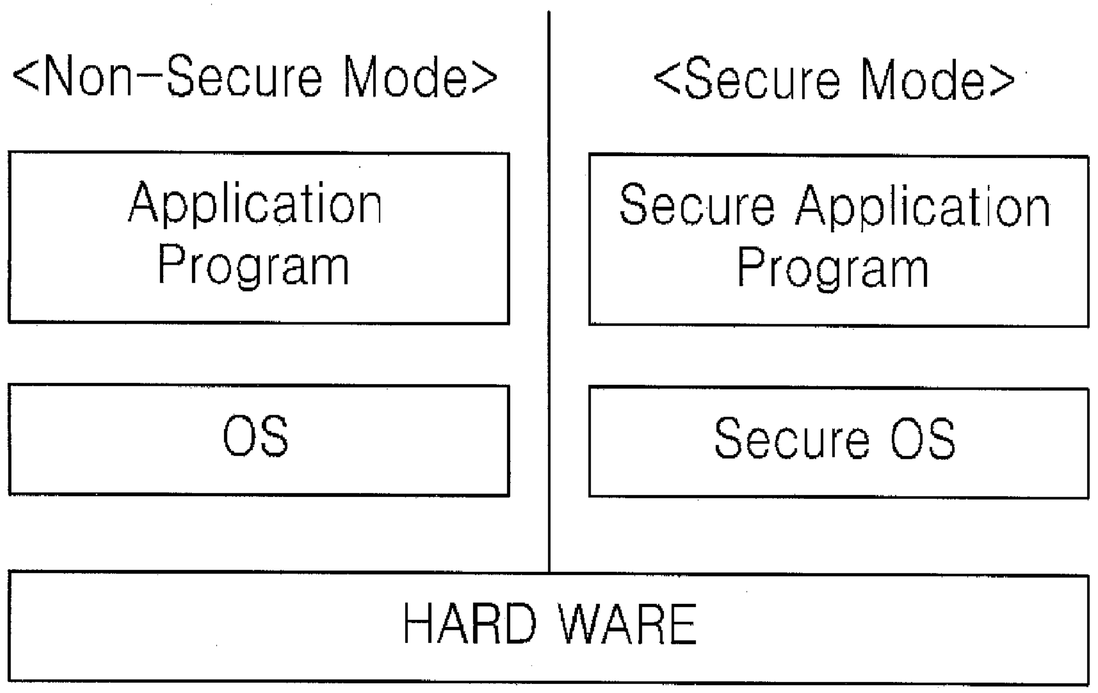

<Non-Secure Mode>
<Secure Mode>
Application Program
Secure Application Program
OS
Secure OS
HARD WARE

FIG. 4

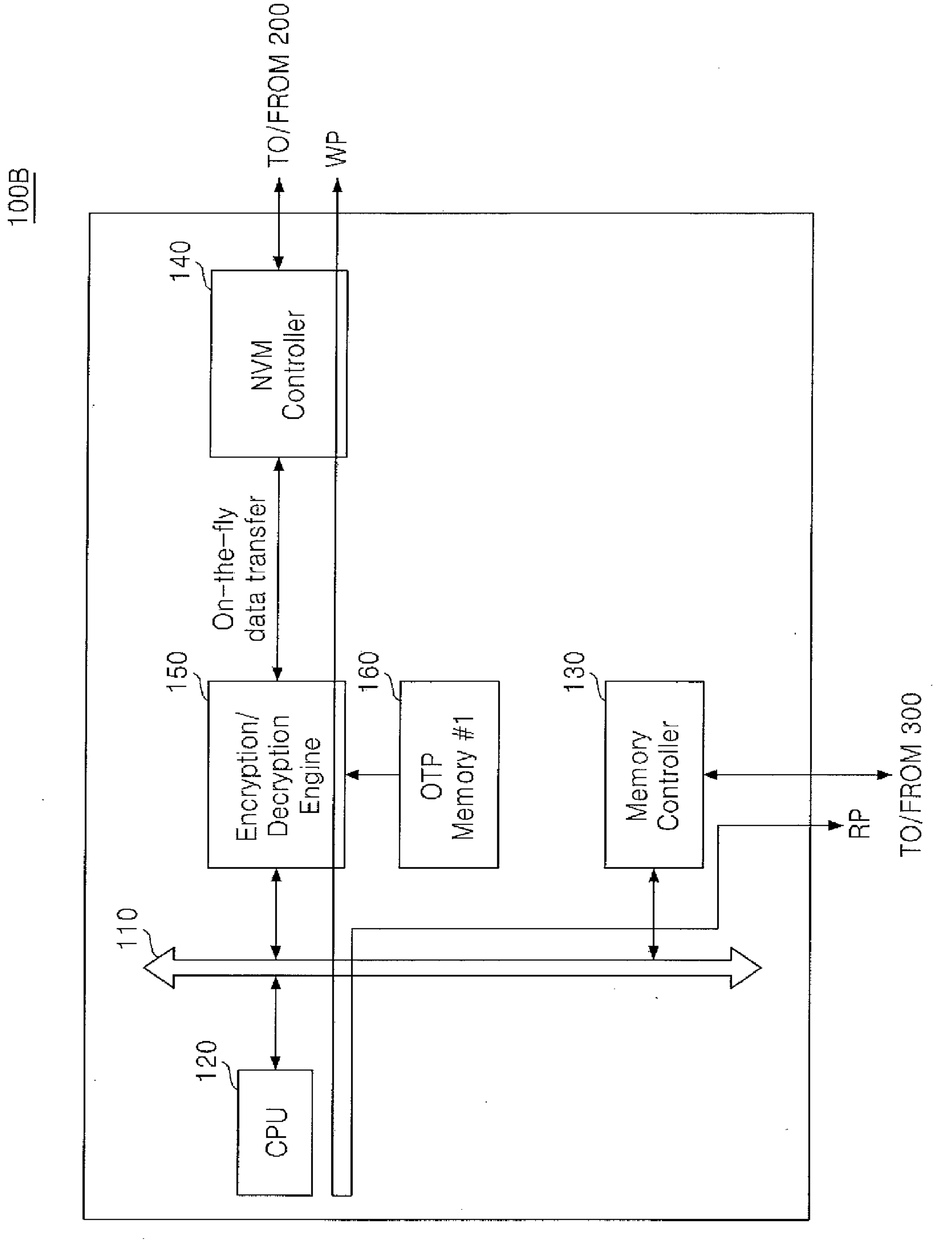

100B
TO/FROM 200
WP
140
NVM Controller
On-the-fly data transfer
150
Encryption/Decryption Engine
160
OTP Memory #1
130
Memory Controller
RP
TO/FROM 300
110
120
CPU

FIG. 5

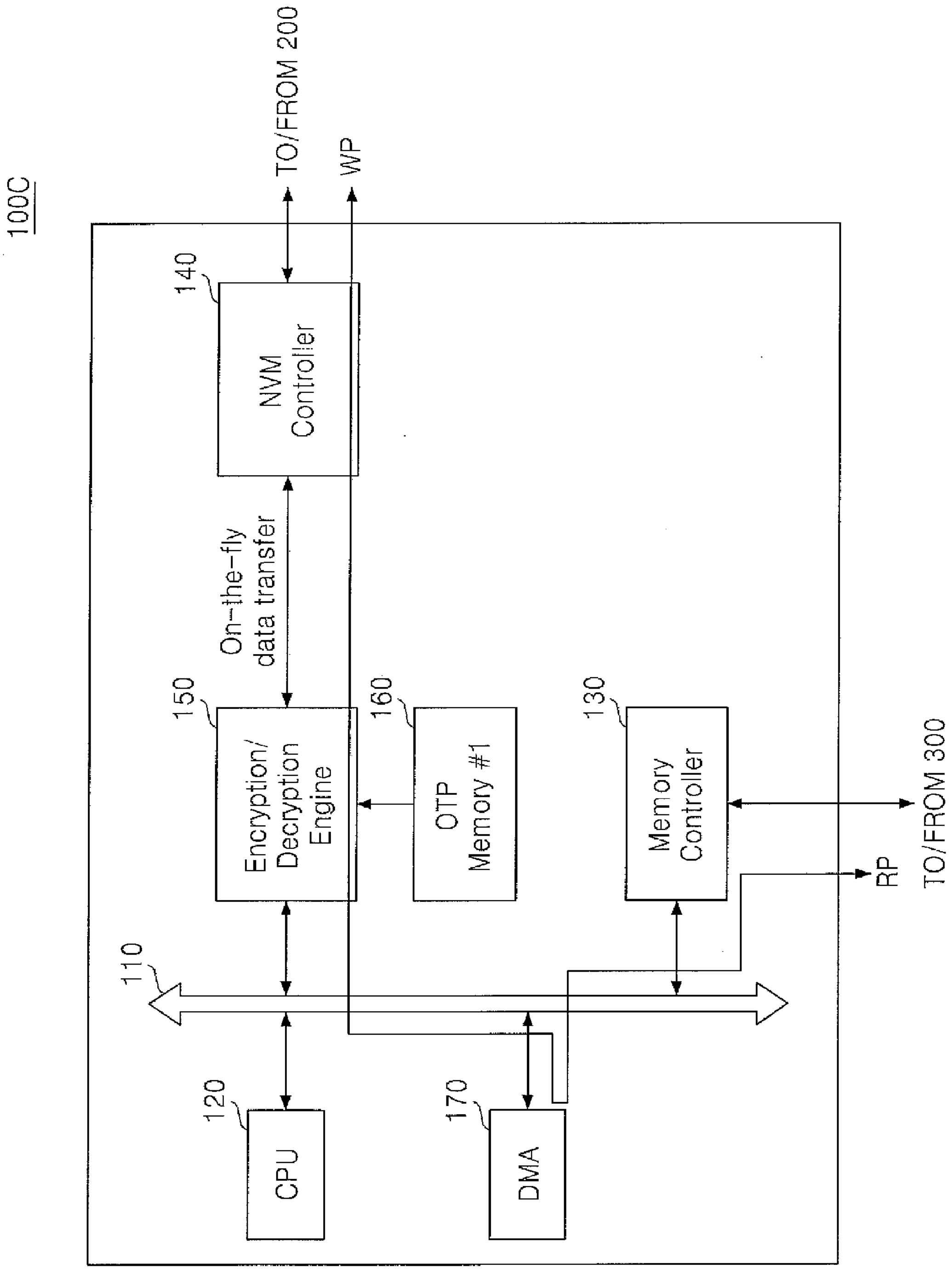

100C
TO/FROM 200
WP
140
NVM Controller
On-the-fly data transfer
150
Encryption/ Decryption Engine
160
OTP Memory #1
130
Memory Controller
TO/FROM 300
RP
110
120
CPU
170
DMA

FIG. 6

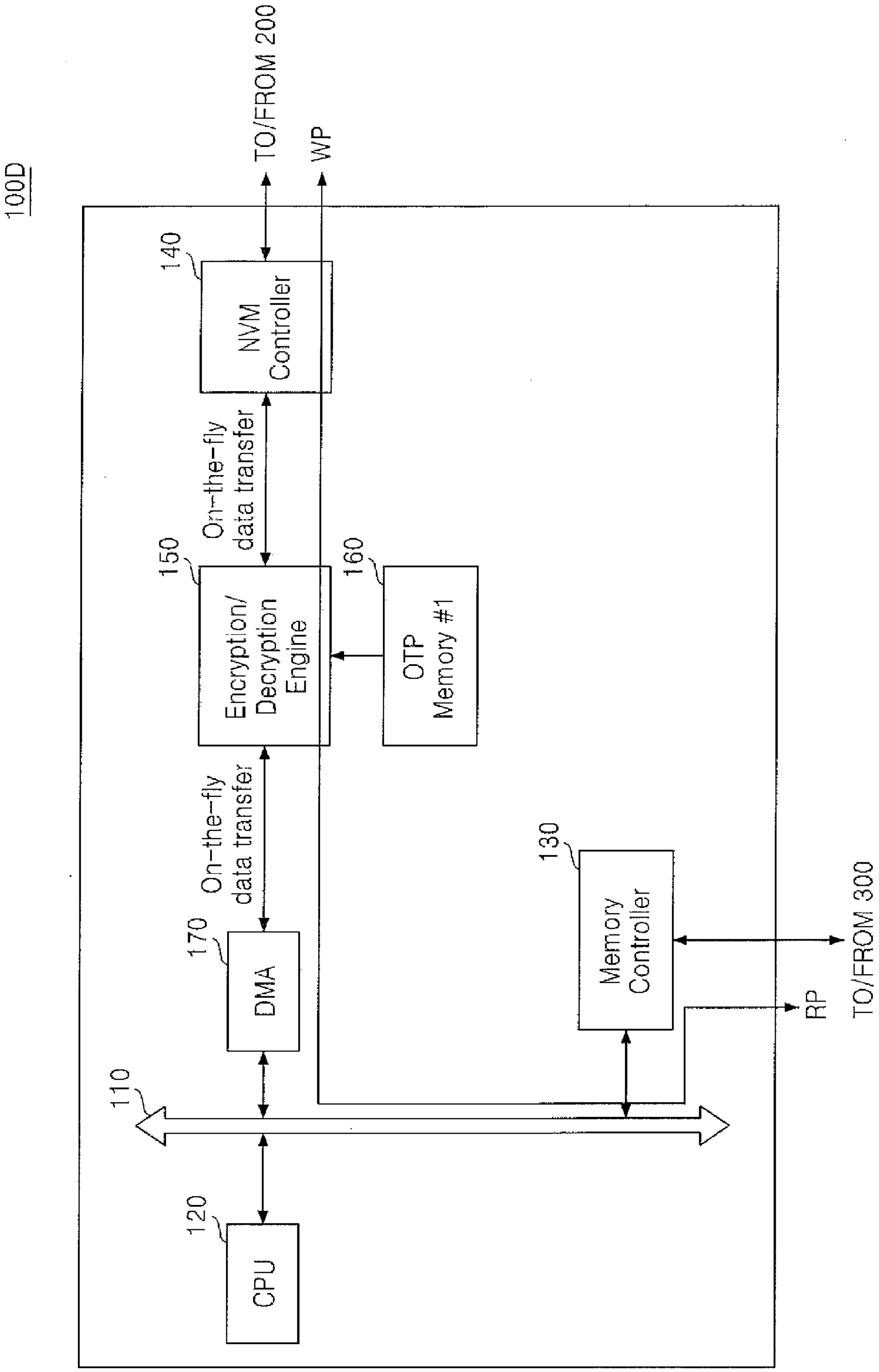

100D
TO/FROM 200
WP
140
NVM Controller
On-the-fly data transfer
150
Encryption/ Decryption Engine
160
OTP Memory #1
On-the-fly data transfer
170
DMA
130
Memory Controller
TO/FROM 300
RP
110
120
CPU

FIG. 7

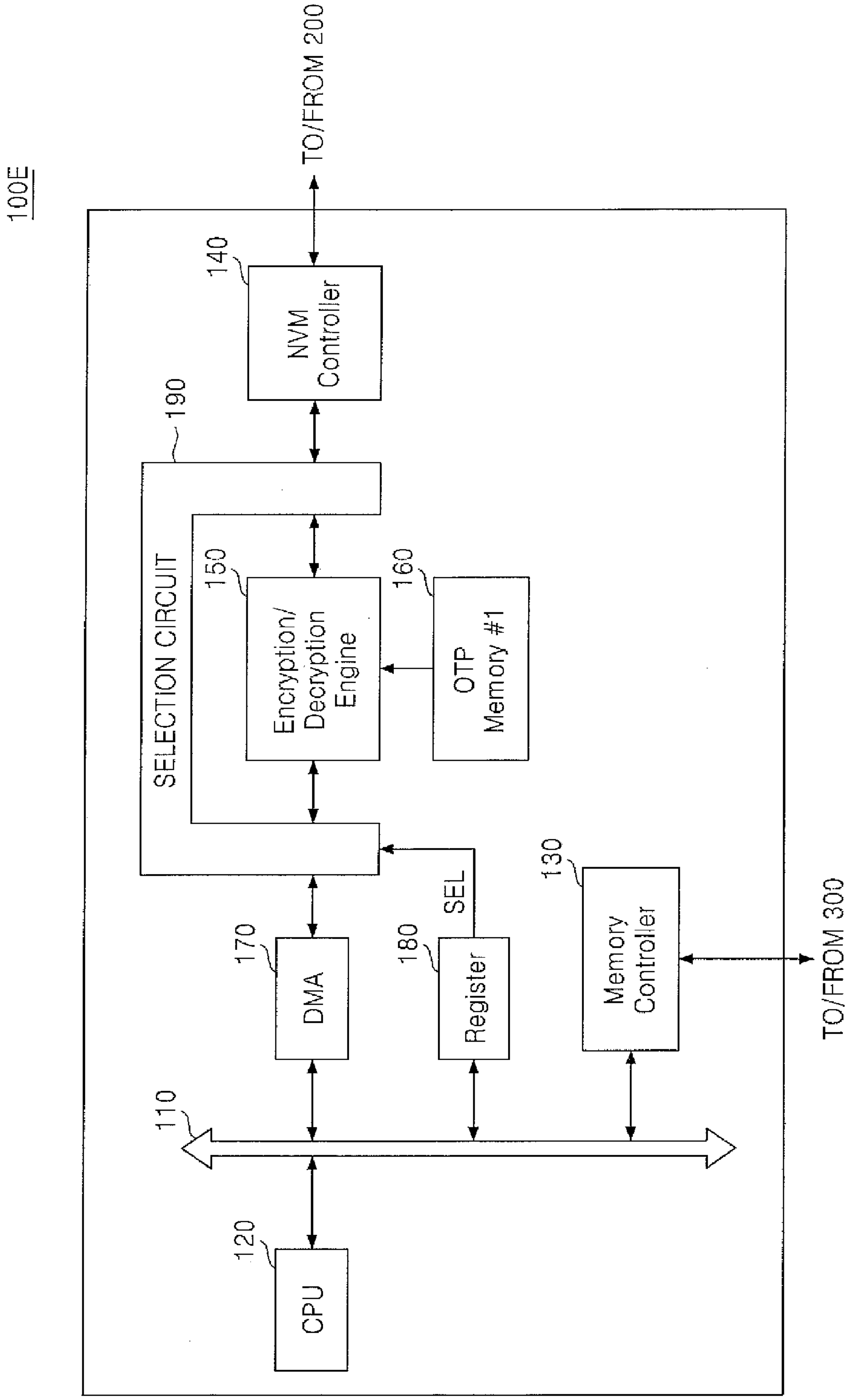

100E
TO/FROM 200
140
NVM Controller
190
SELECTION CIRCUIT
150
Encryption/ Decryption Engine
160
OTP Memory #1
SEL
170
DMA
180
Register
130
Memory Controller
TO/FROM 300
110
120
CPU

FIG. 8

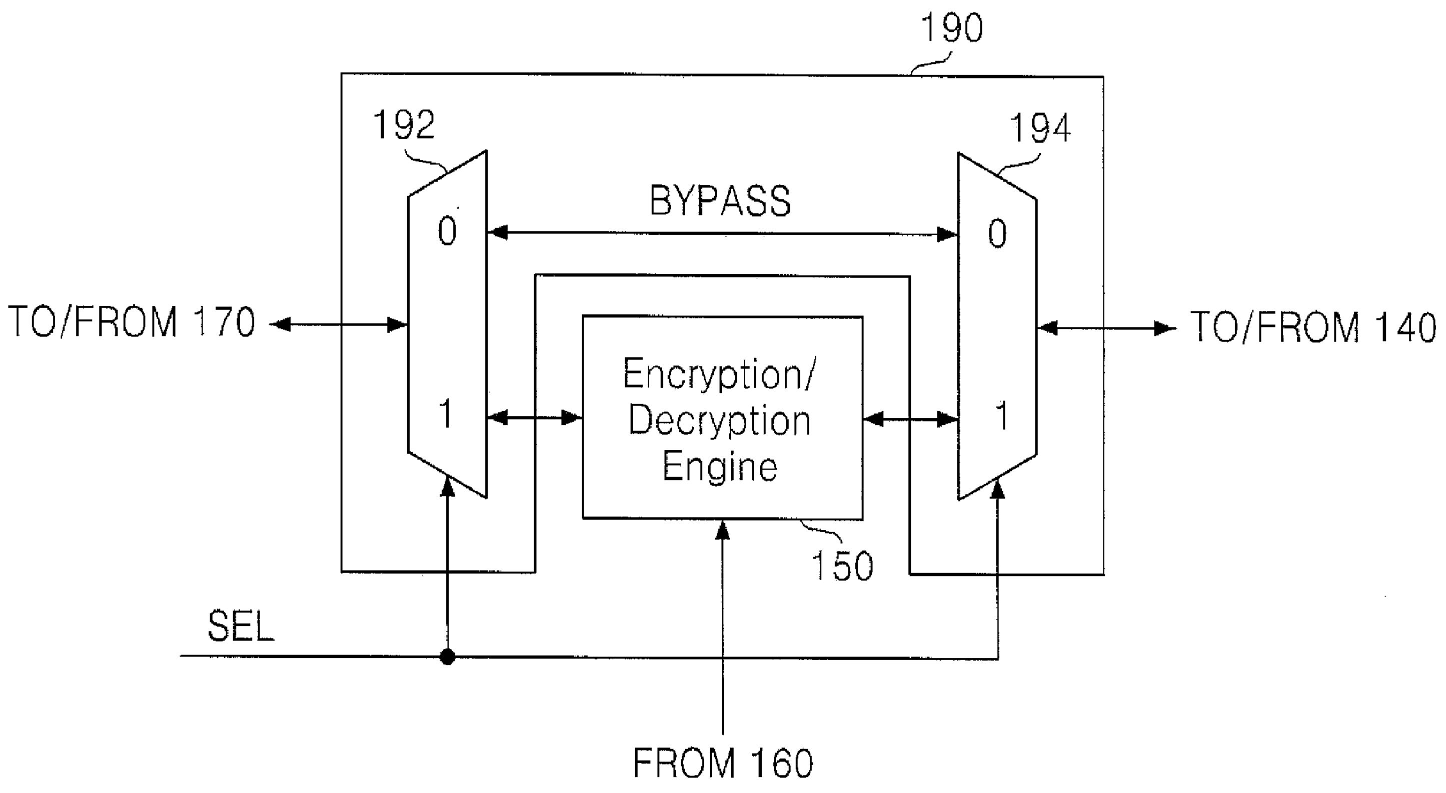

190
192
194
BYPASS
0
0
TO/FROM 170
TO/FROM 140
Encryption/
Decryption
Engine
1
1
150
SEL
FROM 160

FIG. 9

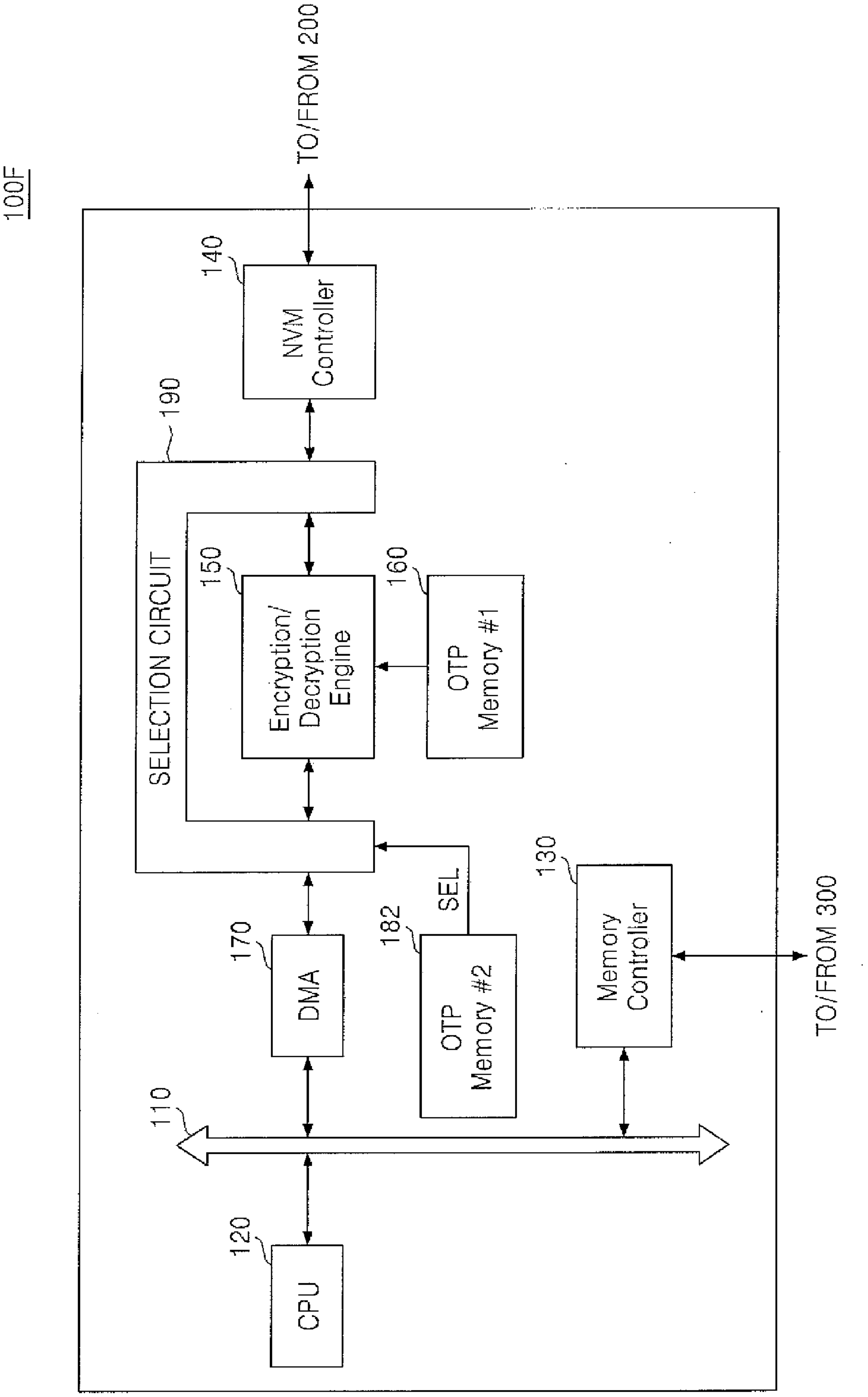

100F
TO/FROM 200
140
NVM Controller
190
SELECTION CIRCUIT
150
Encryption/ Decryption Engine
160
OTP Memory #1
SEL
182
OTP Memory #2
170
DMA
130
Memory Controller
TO/FROM 300
110
120
CPU

FIG. 10

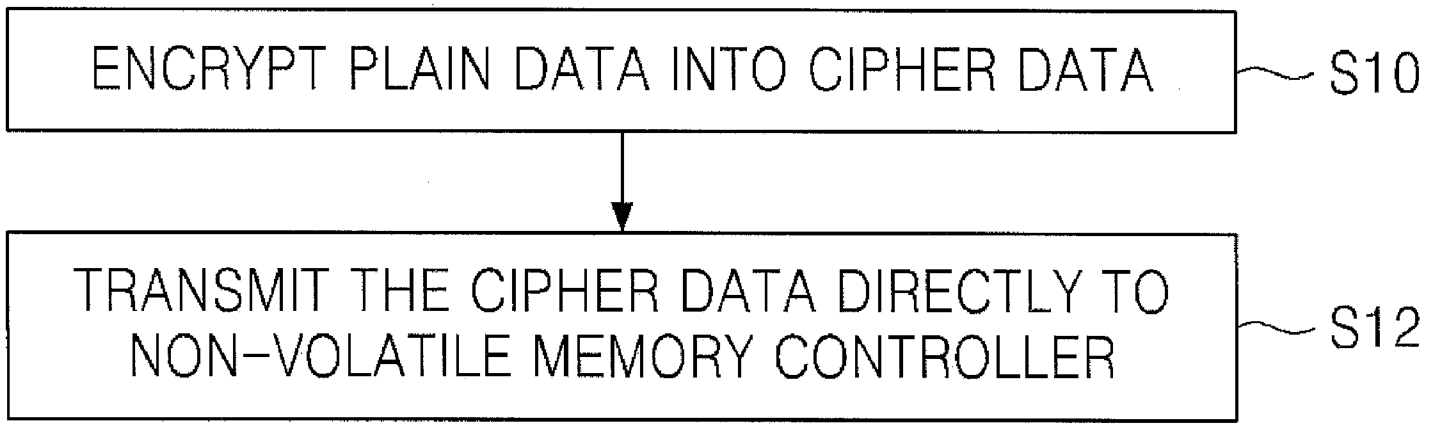

ENCRYPT PLAIN DATA INTO CIPHER DATA
S10
TRANSMIT THE CIPHER DATA DIRECTLY TO NON-VOLATILE MEMORY CONTROLLER
S12

FIG. 11

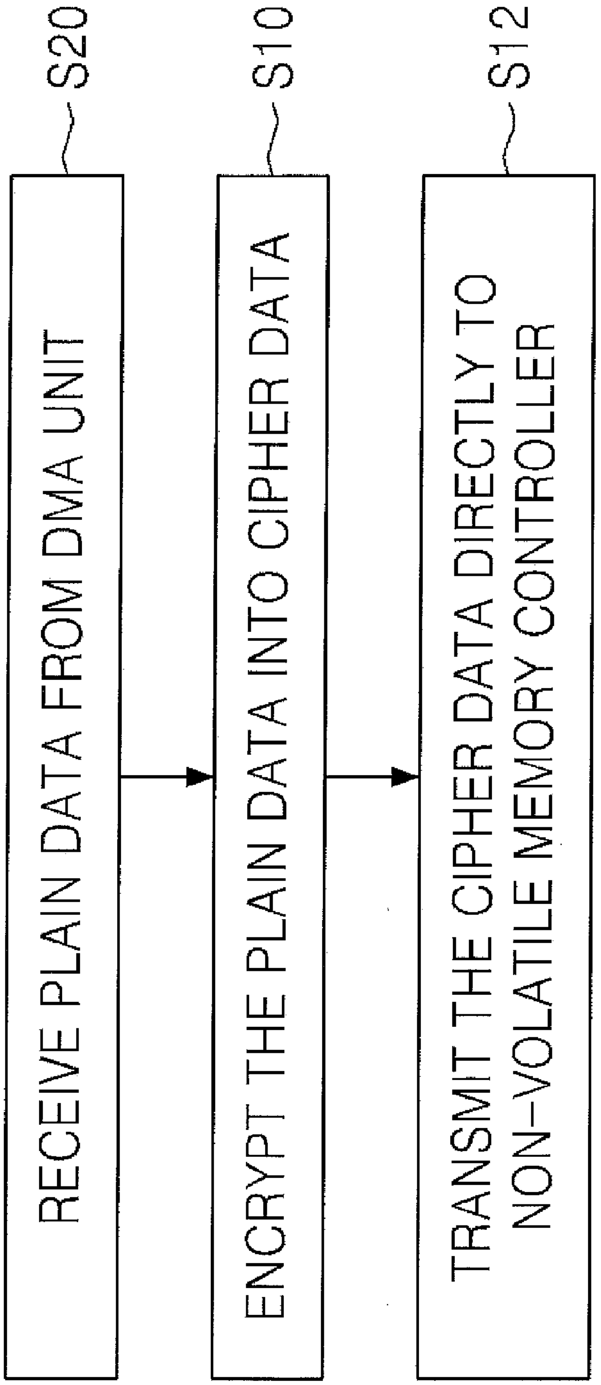

S20
RECEIVE PLAIN DATA FROM DMA UNIT
S10
ENCRYPT THE PLAIN DATA INTO CIPHER DATA
S12
TRANSMIT THE CIPHER DATA DIRECTLY TO NON-VOLATILE MEMORY CONTROLLER

FIG. 12

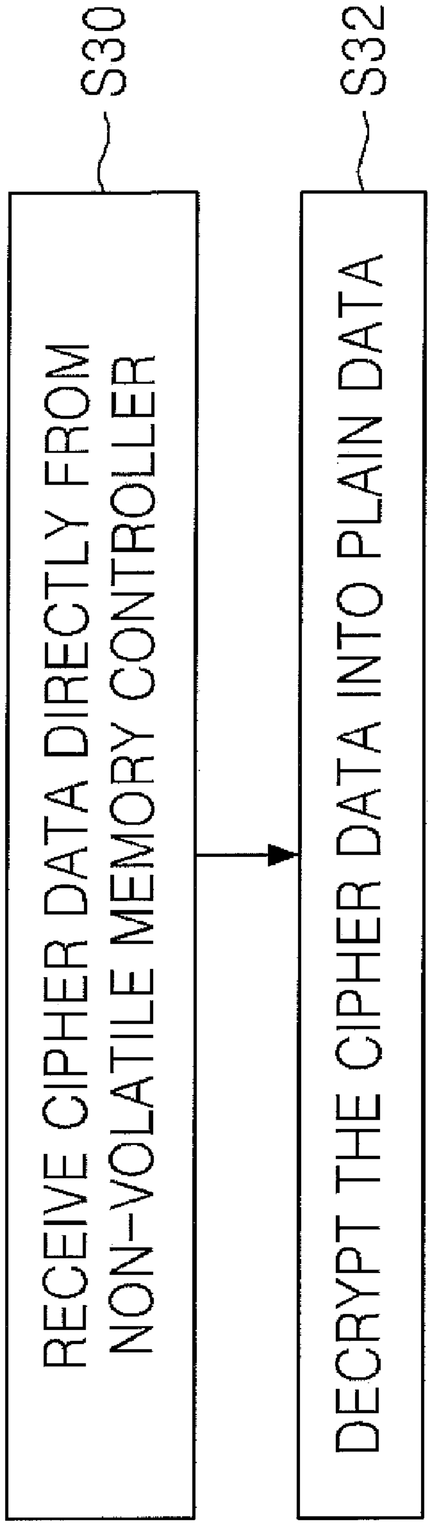

RECEIVE CIPHER DATA DIRECTLY FROM NON-VOLATILE MEMORY CONTROLLER
S30
DECRYPT THE CIPHER DATA INTO PLAIN DATA
S32

FIG. 13

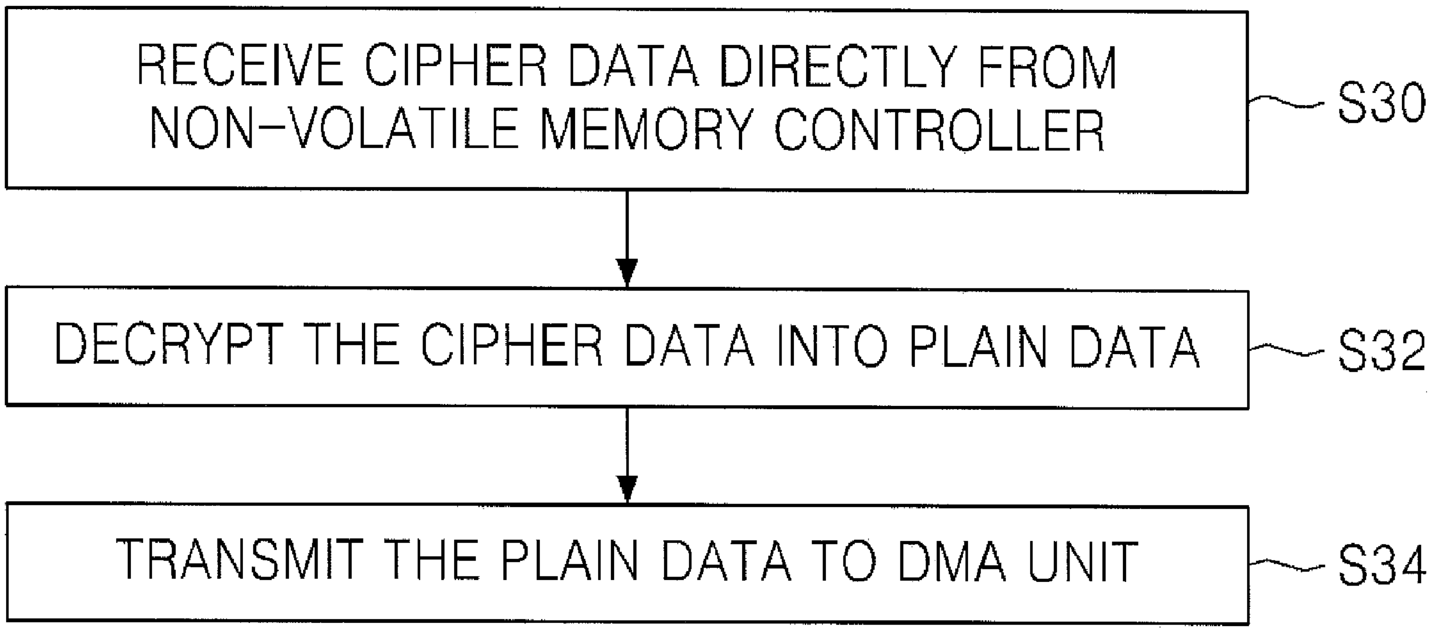

RECEIVE CIPHER DATA DIRECTLY FROM NON-VOLATILE MEMORY CONTROLLER
S30
DECRYPT THE CIPHER DATA INTO PLAIN DATA
S32
TRANSMIT THE PLAIN DATA TO DMA UNIT
S34

FIG. 14

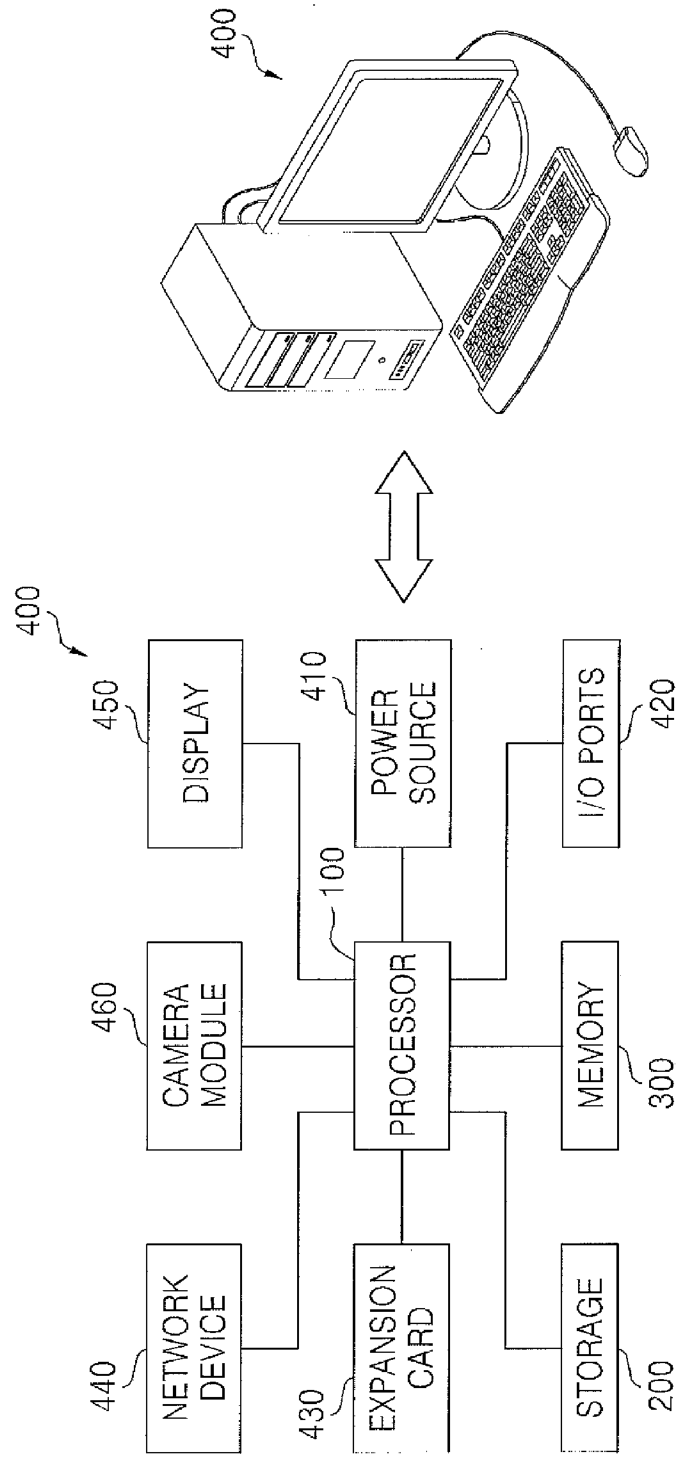

400
400
DISPLAY
450
POWER SOURCE
410
I/O PORTS
420
100
PROCESSOR
CAMERA MODULE
460
MEMORY
300
NETWORK DEVICE
440
EXPANSION CARD
430
STORAGE
200

FIG. 15

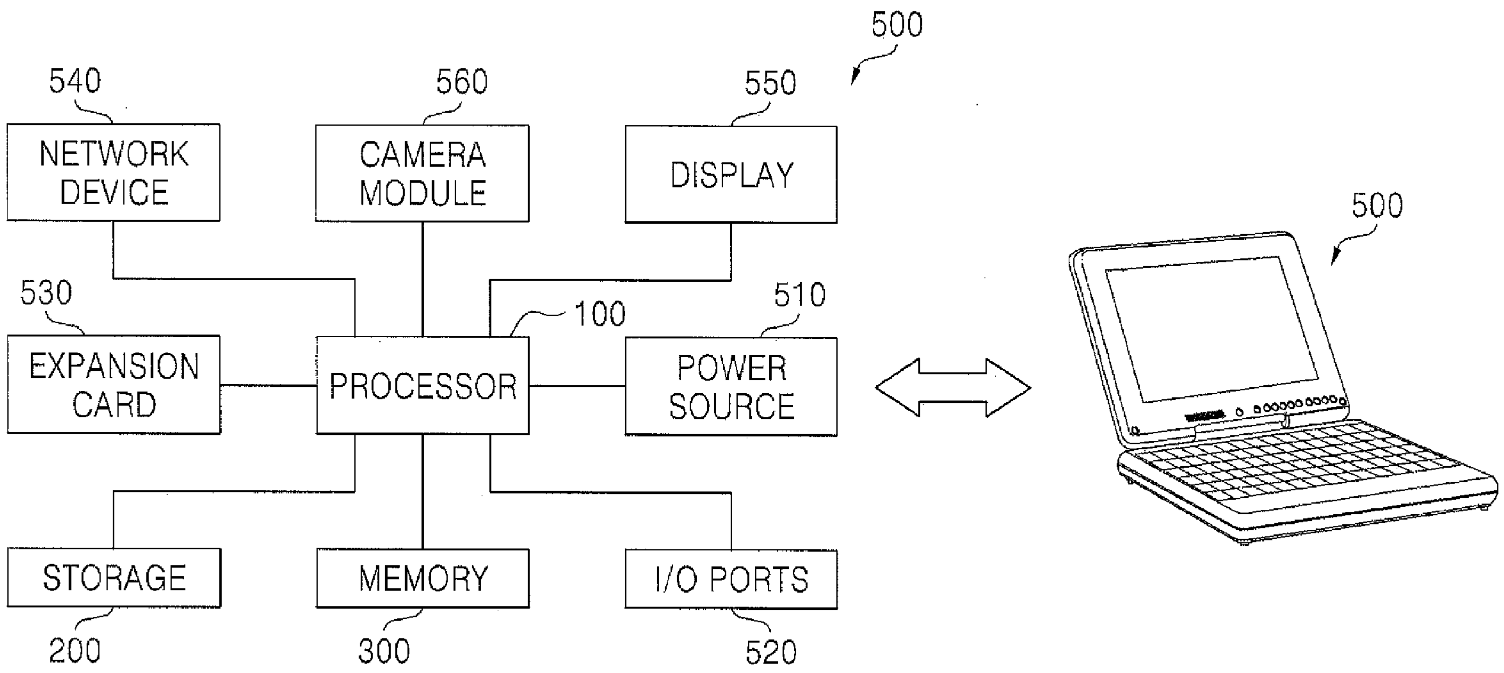

500
540
NETWORK DEVICE
560
CAMERA MODULE
550
DISPLAY
530
EXPANSION CARD
100
PROCESSOR
510
POWER SOURCE
STORAGE
200
MEMORY
300
I/O PORTS
520
500

FIG. 16

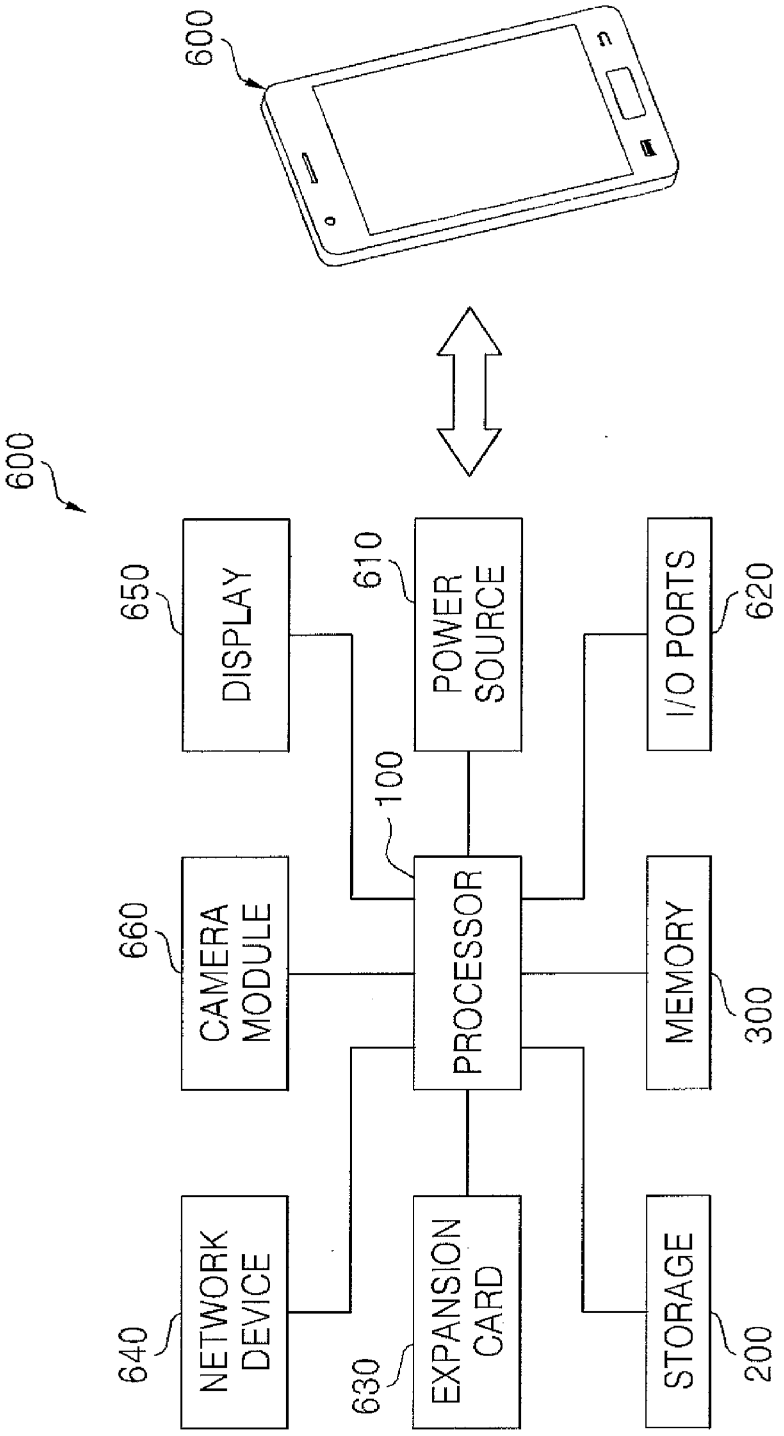

600
600
DISPLAY
650
POWER SOURCE
610
I/O PORTS
620
100
CAMERA MODULE
660
PROCESSOR
MEMORY
300
NETWORK DEVICE
640
EXPANSION CARD
630
STORAGE
200

FIG. 17

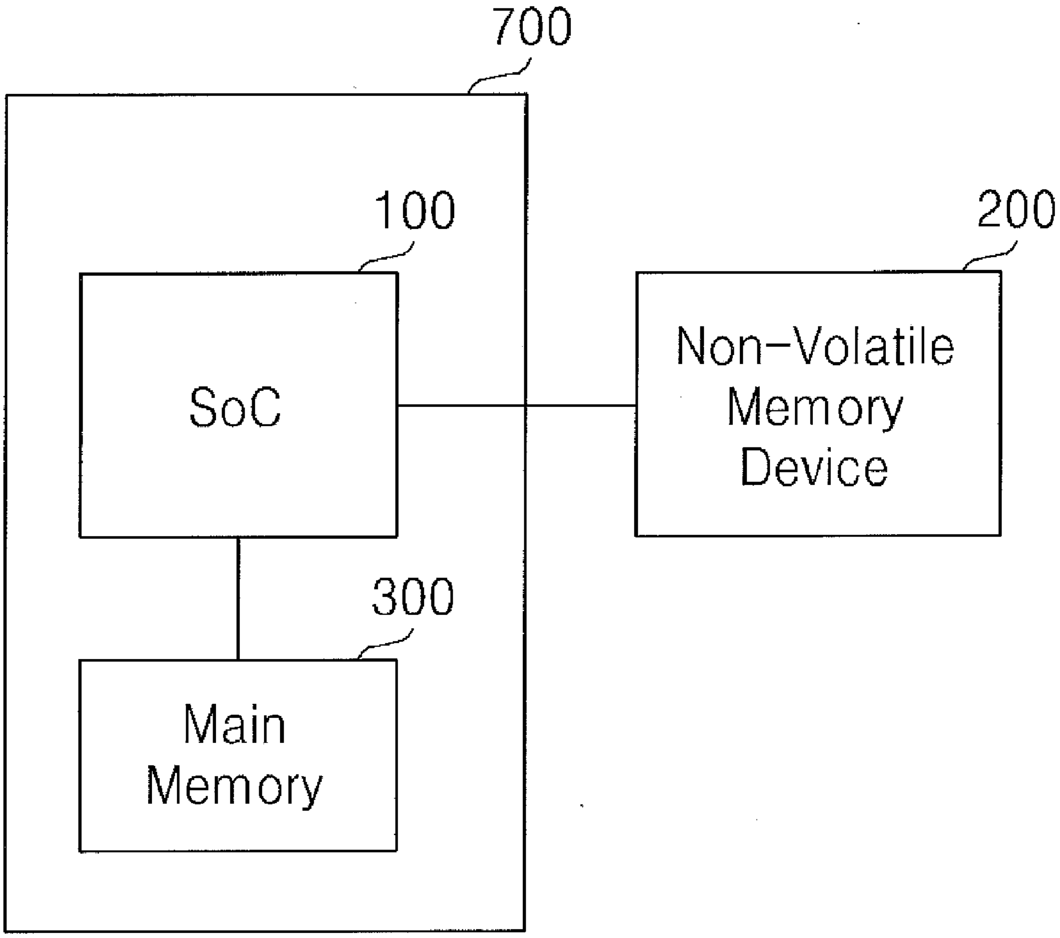
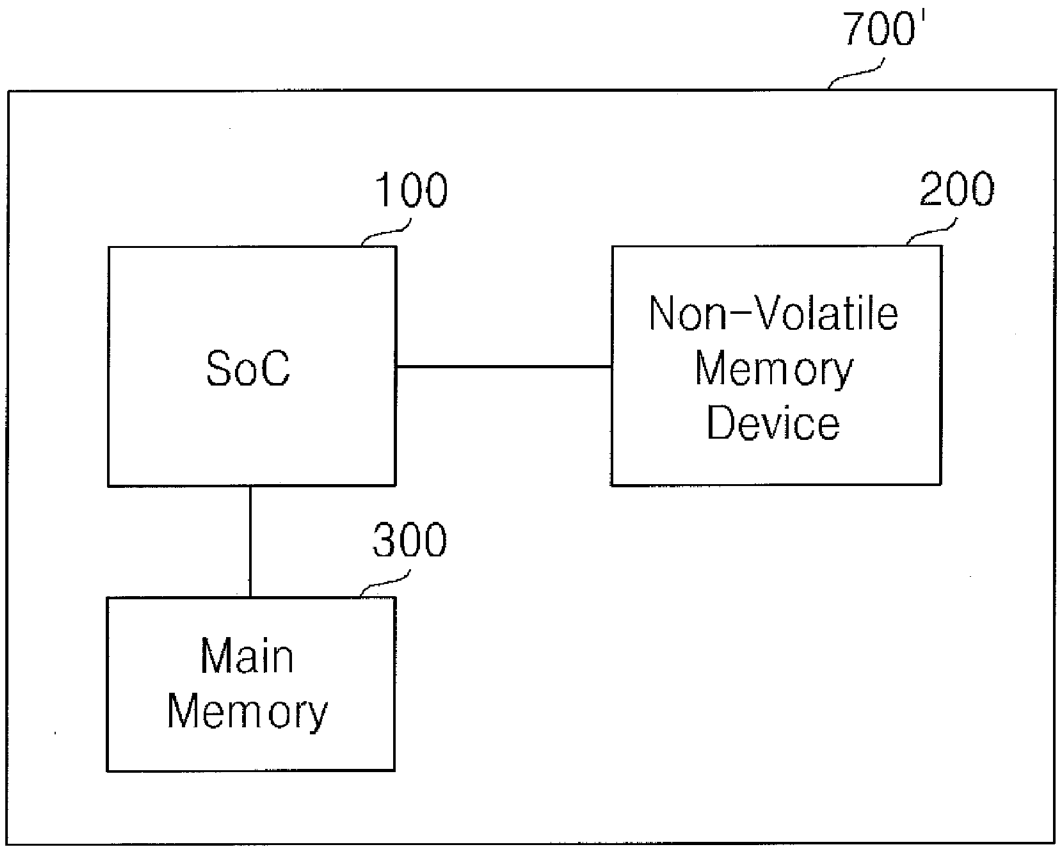


FIG. 18



SYSTEM ON CHIP, METHOD OF OPERATING THE SAME, AND DEVICES INCLUDING THE SYSTEM ON CHIP

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to Korean Patent Application No. 10-2012-0047743, filed on May 4, 2012, in the Korean Intellectual Property Office, the disclosure of which is incorporated by reference in its entirety herein.

BACKGROUND

[0002] 1. Technical Field

[0003] Exemplary embodiments of the inventive concept relate to a system on chip (SoC), a method of operating the SoC, and devices including the SoC.

[0004] 2. Discussion of Related Art

[0005] A main memory may receive programs that are to be executed by a central processing unit (CPU) and data required by the CPU, from a separate storage medium, for example, a non-volatile memory device. The main memory may transmit data to the separate storage medium, for example, the non-volatile memory device, to store the data.

[0006] Data exchanged between the main memory and the non-volatile memory device may be encrypted to prevent unauthorized users from accessing the data. Performance of a system including the main memory and the non-volatile memory device may vary based on the path used to exchange data between the main memory and the non-volatile memory device.

SUMMARY

[0007] According to an exemplary embodiment of the present inventive concept, a method of operating a system on chip (SoC) includes converting plain data into cipher data by an engine within the SoC using an encryption key; and transmitting, by the engine, the cipher data directly to a memory controller within the SoC that controls an operation of a non-volatile memory. According to an exemplary embodiment, the plain data may be data read from a main memory via a bus under the control of a central processing unit (CPU). For example, prior to the converting, the method may include the SoC reading the plain data from the main memory. According to an exemplary embodiment, the plain data may be data output from a direct memory access (DMA) unit. For example, prior to the converting, the method may include a DMA unit within the SoC outputting the plain data to the engine.

[0008] The encryption key may be output from a one-time programmable (OTP) memory. For example, prior to the converting, the method may include an OTP within the SoC outputting the key to the engine. The encryption key may be input only according to a secure program. For example, prior to the converting, the method may include outputting the key to the engine only while a secure program is being executed (e.g., by a CPU within the SoC).

[0009] In the converting, the plain data may be converted into the cipher data in units of blocks.

[0010] According to an exemplary embodiment of the present inventive concept, a method of operating a system on chip (SoC) includes receiving, by an engine within the SoC, cipher data directly from a memory controller within the SoC which controls an operation of a non-volatile memory; and converting, by the engine, the cipher data into plain data by using an encryption key.

[0011] The method may further include transmitting the plain data to a direct memory access (DMA) unit. For example, prior to the converting, the method may further include a DMA unit within the SoC transmitting the plain data to the engine. In the converting, the cipher data may be converted into the plain data in units of blocks.

[0012] According to an exemplary embodiment of the present inventive concept, a system on chip (SoC) includes an encryption/decryption engine which encrypts first plain data into first cipher data or decrypts second cipher data into second plain data, by using an encryption key; and a memory controller which is directly connected to the encryption/decryption engine and transmits the first cipher data to a non-volatile memory or receives the second cipher data from the non-volatile memory.

[0013] The SoC may further include a one-time programmable (OTP) memory which stores the encryption key. The SoC may further include a direct memory access (DMA) unit which transmits the first plain data, which is received from a data source, to the encryption/decryption engine or transmits the second plain data, which is received from the encryption/decryption engine, to the data source. For example, the DMA unit may receive the first plain data from a device outside the SOC and transmit the first plain data to the engine or transmit the second plain data received from the engine to the device.

[0014] The DMA unit may be directly connected to the encryption/decryption engine. The SoC may further include a CPU which controls transmission of the first plain data or the second plain data between a data source and the encryption/decryption engine. For example, the data source may be a device located outside the SoC.

[0015] According to an exemplary embodiment of the present inventive concept, a system-in package includes the SoC and a data source which communicates data with a non-volatile memory under the control of the SoC. The data source may be a device outside the SoC. According to an exemplary embodiment of the present inventive concept, a system-in package includes the SoC, a non-volatile memory, and a data source which communicates data with the non-volatile memory under the control of the SoC.

[0016] According to an exemplary embodiment of the present inventive concept, a system on chip (SoC) includes a memory controller configured to control a non-volatile memory, and an encryption/decryption engine directly connected to the memory controller and configured to encrypt or decrypt data. The SoC controls transmission of data between a data source (e.g., a device outside the SoC) and the non-volatile memory. The memory controller and the engine correspond to a first data path for transmitting data.

[0017] According to an exemplary embodiment of the present inventive concept, an electronic device includes a data source; a non-volatile memory; and a system on chip (SoC) which controls transmission of data between the data source and the non-volatile memory. The SoC may include a memory controller which controls the non-volatile memory; and an encryption/decryption engine which is directly connected to the memory controller and encrypts or decrypts the data. The encryption/decryption engine may encrypt or decrypt the data by using an encryption key that is stored in a one-time programmable (OTP) memory.

[0018] According to an exemplary embodiment of the inventive concept, a system on chip (SoC) includes a data bus, a main memory controller configured to output plain data to the bus, an engine configured to encrypt the plain data from the bus into cipher data using a key, a NVM controller, a first electrical path connecting the bus to the NVM controller to bypass the encryption engine, and a second electrical path connecting the bus to the NVM controller through the encryption engine. The SoC activates only the first electrical path (e.g., deactivates the second electrical path) in a non-secure mode for sending the plain data from the bus to the NVM controller. The SoC activates only the second electrical path (e.g., deactivates the first electrical path) in a secure mode for sending the plain data to the engine and the cipher data from the engine to the NVM controller.

[0019] The first electrical path may include a path of the plain text through a multiplexer and a demultiplexer to the NVM controller. The second electrical path may include a path of the plain data through the multiplexer to the engine and a path of the cipher data through the demultiplexer to the NVM controller. The SoC may further include an OTP configured to provide a same selection signal to the multiplexer and the demultiplexer for activating one of the first and second electrical paths. The engine may be configured to decrypt cipher data received from the NVM controller across the second electrical path.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] FIG. **1** is a block diagram of a system including a system on chip (SoC) according to an exemplary embodiment of the inventive concept;

[0021] FIG. **2** is a block diagram of the SoC illustrated in FIG. **1** according to an exemplary embodiment of the inventive concept;

[0022] FIG. **3** is a conceptual diagram for describing a secure mode in which a encryption key can be input to an encryption/decryption engine illustrated in FIG. **2**;

[0023] FIG. **4** is a block diagram of the SoC illustrated in FIG. **1** according to an exemplary embodiment of the inventive concept;

[0024] FIG. **5** is a block diagram of the SoC illustrated in FIG. **1** according to an exemplary embodiment of the inventive concept;

[0025] FIG. **6** is a block diagram of the SoC illustrated in FIG. **1** according to an exemplary embodiment of the inventive concept;

[0026] FIG. **7** is a block diagram of the SoC illustrated in FIG. **1** according to an exemplary embodiment of the inventive concept;

[0027] FIG. **8** is a block diagram of a selection circuit and an encryption/decryption engine illustrated in FIG. **7** according to an exemplary embodiment of the inventive concept;

[0028] FIG. **9** is a block diagram of the SoC illustrated in FIG. **1** according to an exemplary embodiment of the inventive concept;

[0029] FIG. **10** is a flowchart of a method of operating an SoC, according to an exemplary embodiment of the inventive concept;

[0030] FIG. **11** is a flowchart of a method of operating an SoC, according to an exemplary embodiment of the inventive concept;

[0031] FIG. **12** is a flowchart of a method of operating an SoC, according to an exemplary embodiment of the inventive concept;

[0032] FIG. **13** is a flowchart of a method of operating an SoC, according to an exemplary embodiment of the inventive concept;

[0033] FIG. **14** is a block diagram of a data processing device including the system of FIG. **1**, according to an exemplary embodiment of the inventive concept;

[0034] FIG. **15** is a block diagram of a data processing device including the system of FIG. **1**, according to an exemplary embodiment of the inventive concept;

[0035] FIG. **16** is a block diagram of a data processing device including the system of FIG. **1**, according to an exemplary embodiment of the inventive concept;

[0036] FIG. **17** is a block diagram of a system-in package including the SoC illustrated in FIG. **1**, according to an exemplary embodiment of the inventive concept, and a non-volatile memory device illustrated in FIG. **1**; and

[0037] FIG. **18** is a block diagram of a system-in package including the SoC illustrated in FIG. **1**, according to an exemplary embodiment of the inventive concept.

DETAILED DESCRIPTION

[0038] Herein, when one device is described as transmitting data to another device, the data is either transmitted from the one device directly (e.g., "directly transmitted") or indirectly (e.g., "indirectly transmitted") to the other device. In an exemplary embodiment where the one device directly transmits the data, the data is transmitted from the one device to the other device without passing through another device except a wire. In an exemplary embodiment where one device directly transmits the data, the data is transmitted from the one device to the other device through a multiplexer or a demultiplexer without using additional devices.

[0039] Herein, when one device is described as being connected to another device, these devices may directly connected to one another (e.g., "connected directly") or indirectly connected (e.g., "connected indirectly") to one another. In an embodiment where the two devices are directly connected to one another, no devices are present between the two devices except a wire. In an embodiment where two devices are directly connected to one another, no devices are present between the two devices other than a multiplexer or a demultiplexer.

[0040] It is to be understood that the apparatuses and methods described herein may be implemented in various forms of hardware, software, firmware, special purpose processors, or a combination thereof. In particular, a portion of the present inventive concept may be implemented as an application comprising program instructions that are tangibly embodied on one or more program storage devices or computer readable media (e.g., hard disk, magnetic floppy disk, RAM, ROM, CD ROM, etc.) and executable by any device or machine comprising suitable architecture, such as a general purpose digital computer having a processor, memory, and input/output interfaces. It is to be further understood that, because some of the constituent apparatus components and process steps depicted in the accompanying figures may be implemented in software, the connections between apparatus modules (or the logic flow of method steps) may differ depending upon the manner in which the present inventive concept is programmed.

[0041] FIG. **1** is a block diagram of a system **10** including a system on chip (SoC) **100** according to an exemplary embodiment of the present inventive concept. Referring to FIG. **1**, the system **10** includes the SoC **100**, a non-volatile memory device **200**, and a main memory **300**.

[0042] The system **10** may be implemented by using a personal computer (PC), a data server, or a portable device. For example, the portable device may be implemented by using a laptop computer, a mobile phone, a smart phone, a tablet PC, a personal digital assistant (PDA), an enterprise digital assistant (EDA), a digital still camera, a digital video camera, a portable multimedia player (PMP), a personal (or portable) navigation device (PND), a handheld game console, or an e-book.

[0043] The SoC **100** may control data transmission and reception between the non-volatile memory device **200** and the main memory **300**. A structure and an operation of the SoC **100** will be described in detail later with reference to FIGS. **2** and **4** to **9**.

[0044] The non-volatile memory device **200** may store a variety of programs and data. The non-volatile memory device **200** may be implemented by using an electrically erasable programmable read-only Memory (EEPROM), a flash memory, a magnetic random access memory (MRAM), a spin-transfer torque MRAM, a conductive bridging RAM (CBRAM), a ferroelectric RAM (FeRAM), a phase change RAM (PRAM), a resistive RAM (RRAM), a nanotube RRAM, a polymer RAM (PoRAM), a nano floating gate memory (NFGM), a holographic memory, a molecular electronics memory device, an insulator resistance change memory, or the like.

[0045] The main memory **300** may receive programs which are to be executed in the SoC **100** and data required by the SoC **100**, from the non-volatile memory device **200** via the SoC **100**. The main memory **300** may transmit data that is to be stored, to the non-volatile memory device **200** via the SoC **100**. The main memory **300** may be implemented by using a RAM, for example, a dynamic RAM (DRAM) or a static RAM (SRAM), which is a volatile memory. However, the main memory **300** is not limited to a RAM, DRAM, or SRAM, as different types of memories may be used.

[0046] FIG. **2** is a block diagram of an SoC **100**A, which is an embodiment of the SoC **100** of FIG. **1**. Referring to FIG. **2**, the SoC **100**A includes a bus **110**, a central processing unit (CPU) **120**, a memory controller **130**, a non-volatile memory controller **140**, and an encryption/decryption engine **150**.

[0047] The CPU **120** may be connected to the bus **110** and may control the entire operation of the SoC **100**A. The memory controller **130** may control an operation of the main memory **300**, for example, a read or write operation. The memory controller **130** may be connected to the bus **110**.

[0048] The non-volatile memory controller **140** may control a data access operation of the non-volatile memory device **200**, for example, a write operation, a read operation, a program operation, or an erase operation.

[0049] The encryption/decryption engine **150** may convert, namely, encrypt, plain data received from the main memory **300** via the memory controller **130** and the bus **110**, into cipher data (e.g., encrypted data). The encryption/decryption engine **150** may transmit the cipher data directly, for example, on-the-fly, to the non-volatile memory controller **140** without passing through the bus **110**.

[0050] In exemplary embodiment, the encryption/decryption engine **150** receives cipher data directly, for example, on-the-fly, from the non-volatile memory controller **140** without passing through the bus **110**. The encryption/decryption engine **150** may convert, for example, decrypt, the cipher data into plain data (e.g., un-encrypted data).

[0051] An encryption key may be used in encryption or decryption of the encryption/decryption engine **150**. In an exemplary embodiment, the encryption/decryption engine **150** includes a storage medium (not shown) which stores the encryption key. For example, the storage medium could be a register, a latch, flash memory, etc. According to an exemplary embodiment, the encryption key is input to the storage medium only in a secure mode. The secure mode will be described later with reference to FIG. **3**.

[0052] The encryption/decryption engine **150** may encrypt or decrypt data in units of blocks of a predetermined size, for example, 64 bits, 128 bits, or 256 bits.

[0053] When the encryption/decryption engine **150** encrypts or decrypts in units of blocks, an encryption key and an algorithm which are used in encryption or decryption may be applied to each block. As an example, the algorithm may be a data encryption standard (DES) algorithm or an advanced encryption standard (AES) algorithm. However, the algorithm is not limited thereto, as other encryption or decryption algorithms may be used.

[0054] The secure mode may determine the method used by the encryption engine **150** to convert data, for example, encrypt or decrypt data in units of blocks. Examples of the secure mode include an electronic code book (ECB) mode, a cipher block chaining (CBC) mode, a propagating cipher block chaining (PCBC) mode, or a cipher feed back (CFB) mode. In ECB, each block is encrypted independently. In CBC, each block of plaintext (e.g., un-encrypted data) is XORed with the previous ciphertext block (e.g., encrypted data) before being encrypted. PCBC is a variation on CBC and is designed to extend or propagate a single bit error in the ciphertext to allow errors in transmission to be captured and the resultant plaintext to be rejected. In CFB, data may be encrypted in units smaller than the block size. However, the secure mode is not limited to being set to one of the modes described above.

[0055] In a write data path WP when data is written to the non-volatile memory device **200**, plain data (e.g., un-encrypted data) output from the main memory **300** is transmitted to the CPU **120** via the memory controller **130** and the bus **110**, and is then transmitted from the CPU **120** to the encryption/decryption engine **150** via the bus **110**. In other words, the plain data is transmitted to the encryption/decryption engine **150** under the control of the CPU **120**.

[0056] The encryption/decryption engine **150** may convert the plain data into cipher data by using an encryption key. The cipher data may be transmitted to the non-volatile memory device **200** via the non-volatile memory controller **140**. In a read data path RP when data is read from the non-volatile memory device **200**, cipher data output from the non-volatile memory device **200** is transmitted to the encryption/decryption engine **150** via the non-volatile memory controller **140**.

[0057] The encryption/decryption engine **150** may convert the cipher data into plain data by using a decryption key. The decryption key may or may not be the same as a encryption key. The plain data may be transmitted to the CPU **120** via the bus **110** and then transmitted from the CPU **120** to the main memory **300** via the bus **110** and the memory controller **130**. In other words, the plain data may be transmitted to the main memory **300** via the bus **110** and the memory controller **130** under the control of the CPU **120**.

[0058] FIG. **3** is a conceptual diagram for describing a secure mode in which an encryption key or a decryption key can be input to the encryption/decryption engine **150** illustrated in FIG. **2**. Referring to FIGS. **2** and **3**, a general operating system (OS) may manage hardware and may be installed in the hardware to execute an application program.

[0059] A secure OS may also be installed in the hardware to execute a secure application program that requires security, independently from a general operating system (OS). The secure OS may be implemented by using a real time operating system (RTOS). For example, the RTOS may be used to execute an application program, which needs to be completed within a predetermined period of time, for example, a secure application program. The secure OS may be a trusted operating system that provides sufficient support for multilevel security and evidence of correctness to meet a particular set of government requirements.

[0060] A non-secure mode may denote an example where the application program is executed by the general OS, and a secure mode may denote an example where the secure application program is executed by the secure OS.

[0061] In an exemplary embodiment, while the secure application program is executed in the secure mode, the CPU **120** inputs an encryption key or a decryption key to the encryption/decryption engine **150**. In an exemplary embodiment, the encryption/decryption engine **150** has access to an encryption key or a decryption key stored within itself or outside the encryption/decryption engine **150**. In an exemplary embodiment, while the secure application program is executed in the secure mode, an encryption key or a decryption key may be changed or re-set. For example, when the key is changed, the next encryption/decryption that occurs uses the updated key.

[0062] FIG. **4** is a block diagram of an SoC **100**B which is an exemplary embodiment of the SoC **100** of FIG. **1**. Referring to FIGS. **1** to **4**, the SoC **100**B includes a bus **110**, a CPU **120**, a memory controller **130**, a non-volatile memory controller **140**, an encryption/decryption engine **150**, and a one-time programmable (OTP) memory **160**. In an exemplary embodiment, the OTP memory **160** stores an encryption key or decryption key that is used in encryption or decryption by the encryption/decryption engine **150**.

[0063] In an exemplary embodiment, the OTP memory **160** is implemented by using a fuse, an anti-fuse, or an e-fuse. In an exemplary embodiment, an anti-fuse is an electrical device that performs the opposite function to a fuse. For example, whereas a fuse starts with a low resistance and is designed to permanently break an electrically conductive path (e.g., when the current through the path exceeds a specified limit), an anti-fuse starts with a high resistance and is designed to permanently create an electrically conductive path (e.g., when the voltage across the anti-fuse exceeds a certain level). In an exemplary embodiment, an e-fuse allows for dynamic real-time reprogramming of computer chips.

[0064] In contrast with the SoC **100**A of FIG. **2**, in the SoC **100**B of FIG. **4**, the CPU **120** may be implemented so that it is prevented from accessing an encryption key or a decryption key stored in the OTP memory **160**, even when a secure application program is executed in a secure mode. For example, the CPU **120** may be implemented so that it is prevented from reading, writing, or erasing the stored key. A write data path WP and a read data path RP of the SoC **100**B of FIG. **4** are substantially the same to those of the SoC **100**A of FIG. **2**, respectively, except that an encryption key or a decryption key that may be used in encryption or decryption of data is provided by the OTP memory **160** to the encryption/decryption engine **150**.

[0065] FIG. **5** is a block diagram of an SoC **100**C which is an exemplary embodiment of the SoC **100** illustrated in FIG. **1**. Referring to FIGS. **1** and **5**, the SoC **100**C includes a bus **110**, a CPU **120**, a memory controller **130**, a non-volatile memory controller **140**, an encryption/decryption engine **150**, an OTP memory **160**, and a direct memory access (DMA) unit **170**.

[0066] The DMA unit **170** may access the main memory **300** or the non-volatile memory device **200** via a component (e.g., the memory controller **130**, the non-volatile memory controller **140**, or the encryption/decryption engine **150**) without passing data through the CPU **120**. In this example, the DMA unit **170** may be connected to the bus **110**.

[0067] In a write data path WP when data is written to the non-volatile memory device **200**, plain data output from the main memory **300** is transmitted to the DMA unit **170** via the memory controller **130** and the bus **110**. The plain data is transmitted from the DMA unit **170** to the encryption/decryption engine **150** via the bus **110**.

[0068] The encryption/decryption engine **150** converts the plain data into cipher data. The cipher data output from the encryption/decryption engine **150** may be transmitted directly to the non-volatile memory controller **140** and then to the non-volatile memory device **200**. In other words, the encryption/decryption engine **150** may transmit the cipher data directly, for example, on-the-fly, to the non-volatile memory controller **140**.

[0069] In a read data path RP when data is read from the non-volatile memory device **200**, cipher data output from the non-volatile memory device **200** is transmitted to the encryption/decryption engine **150** via the non-volatile memory controller **140**. In other words, the encryption/decryption engine **150** may receive cipher data directly, for example, on-the-fly, from the non-volatile memory controller **140**.

[0070] The encryption/decryption engine **150** converts the cipher data into plain data. The plain data is transmitted to the DMA unit **170** via the bus **110**. The plain data may be transmitted from the DMA unit **170** to the main memory **300** via the bus **110** and the memory controller **130**.

[0071] FIG. **6** is a block diagram of an SoC **100**D which is an exemplary embodiment of the SoC **100** illustrated in FIG. **1**. Referring to FIGS. **1** and **6**, the SoC **100**D includes a bus **110**, a CPU **120**, a memory controller **130**, a non-volatile memory controller **140**, an encryption/decryption engine **150**, an OTP memory **160**, and a DMA unit **170**.

[0072] The DMA unit **170** may be connected between the bus **110** and the encryption/decryption engine **150**. Data may be transmitted on-the-fly between the DMA unit **170** and the encryption/decryption engine **150**.

[0073] In a write data path WP when data is written to the non-volatile memory device **200**, plain data output from the main memory **300** is transmitted to the encryption/decryption engine **150** via the memory controller **130**, the bus **110**, and the DMA unit **170**.

[0074] The encryption/decryption engine **150** may convert, for example, encrypt, the plain data into cipher data. The cipher data may be transmitted to the non-volatile memory device **200** via the non-volatile memory controller **140**. In this example, the encryption/decryption engine **150** may transmit the cipher data directly, for example, on-the-fly, to the non-volatile memory controller **140**.

[0075] In a read data path RP when data is read from the non-volatile memory device **200**, cipher data output from the non-volatile memory device **200** is transmitted to the encryption/decryption engine **150** via the non-volatile memory controller **140**. In this example, the encryption/decryption engine **150** may receive the cipher data directly, for example, on-the-fly, from the non-volatile memory controller **140**.

[0076] The encryption/decryption engine **150** may convert, for example, decrypt, the cipher data into plain data. The plain data may be transmitted to the main memory **300** via the DMA unit **170**, the bus **110**, and the memory controller **130**.

[0077] FIG. **7** is a block diagram of an SoC **100**E which is an exemplary embodiment of the SoC **100** illustrated in FIG. **1**. Referring to FIGS. **1**, **3**, and **7**, the SoC **100**E includes a bus **110**, a CPU **120**, a memory controller **130**, a non-volatile memory controller **140**, an encryption/decryption engine **150**, an OTP memory **160**, a DMA unit **170**, a register **180**, and a selection circuit **190**.

[0078] The register **180** may be connected to the bus **110**. The register **180** may operate as a selection signal generator that generates a selection signal SEL. The register **180** may change the selection signal SEL based on whether the CPU **120** executes a secure application program, that is, based on an indication signal that indicates a secure mode. The indication signal may be output by the CPU **120**. For example, the indication signal may be a logic high in a secure mode, and the indication signal may be a logic low in a non-secure mode.

[0079] The selection circuit **190** may select a data path according to the selection signal SEL output by the register **180**. An exemplary structure and an operation of the selection circuit **190** will now be described with reference to FIG. **8**.

[0080] FIG. **8** is a block diagram of the selection circuit **190** and the encryption/decryption engine **150** illustrated in FIG. **7** according to an exemplary embodiment of the inventive concept. Referring to FIGS. **3**, **7**, and **8**, the selection circuit **190** include a first selector **192** and a second selector **194**. The first selector **192** may be implemented by using a demultiplexer, and the second selector **194** may be implemented by using a multiplexer.

[0081] For example, when the selection signal SEL is a logic high, the selection circuit **190** selects a data path including the encryption/decryption engine **150**. When the CPU **120** executes a secure application program, that is, in a secure mode, the selection circuit **190** selects the data path including the encryption/decryption engine **150**. In an exemplary embodiment, when the selection signal SEL is a logic low, the selection circuit **190** selects a data path excluding the encryption/decryption engine **150**, that is, a bypass path. For example, when the CPU **120** performs a general application program, that is, in a non-secure mode, the selection circuit **190** may select the data path excluding the encryption/decryption engine **150**, that is, the bypass path.

[0082] FIG. **9** is a block diagram of an SoC **100**F which is an exemplary embodiment of the SoC **100** illustrated in FIG. **1**. Referring to FIGS. **1**, **8**, and **9**, the SoC **100**F includes a bus **110**, a CPU **120**, a memory controller **130**, a non-volatile memory controller **140**, an encryption/decryption engine **150**, an OTP memory **160**, a DMA unit **170**, a second OTP memory **182**, and a selection circuit **190**.

[0083] In an exemplary embodiment, the second OTP memory **182** operates as a selection signal generator that generates a selection signal SEL. In an exemplary embodiment, the OTP memory **182** is programmed to generate a selection signal SEL having one logic level, for example, a logic high level. In this embodiment, the selection circuit **190** selects only the data path including the encryption/decryption engine **150**.

[0084] FIG. **10** is a flowchart of a method of operating an SoC, according to an exemplary embodiment of the present inventive concept. Referring to FIGS. **2**, **4** to **7**, **9**, and **10**, the encryption/decryption engine **150** converts, for example, encrypts, plain data into cipher data by using an encryption key (S **10**). The encryption/decryption engine **150** may transmit the cipher data directly, for example, on-the-fly, to the non-volatile memory controller **140** (S**12**). Since the encryption key is present within the SoC **100** and is not output outside the Soc **100**, a probe of communications between the Soc and the other memory devices (e.g., **200** and **300**) will not discover the key.

[0085] FIG. **11** is a flowchart of a method of operating an SoC, according to an exemplary embodiment of the present inventive concept. Referring to FIGS. **5** to **7**, **9**, and **10**, the encryption/decryption engine **150** receives plain data from the DMA unit **170** (S**20**). In an exemplary embodiment, the encryption/decryption engine **150** receives the plain data directly, for example, on-the-fly, from the DMA unit **170**. Then similar to FIG. **10**, the engine **150** encrypts the plain data into cipher data (S **10**) and transmits the cipher data directly to the non-volatile memory controller (S**12**).

[0086] FIG. **12** is a flowchart of a method of operating an SoC, according to an exemplary embodiment of the present inventive concept. Referring to FIGS. **2**, **4** to **7**, **9**, and **10**, the encryption/decryption engine **150** receive cipher data directly, for example, on-the-fly, from the non-volatile memory controller **140** (S**30**). The encryption/decryption engine **150** decrypts the cipher data into plain data (S**32**).

[0087] FIG. **13** is a flowchart of a method of operating an SoC, according to an exemplary embodiment of the present inventive concept. Similar to FIG. **12**, the engine **150** receives cipher data directly from the non-voltage memory controller (S**30**) and decrypts the cipher data into plain data (S**32**). Referring to FIGS. **5** to **7**, **9**, and **13**, the encryption/decryption engine **150** transmits the plain data to the DMA unit **170** (S**34**). In an exemplary embodiment, the encryption/decryption engine **150** transmits plain data directly, for example, on-the-fly, to the DMA unit **170**.

[0088] FIG. **14** is a block diagram of a data processing device **400** including the system **10** of FIG. **1**, according to an exemplary embodiment of the present inventive concept. Referring to FIGS. **1** and **14**, the data processing device **400** may be implemented by using a personal computer (PC) or a data server.

[0089] The data processing device **400** includes a processor **100**, a storage device **200**, a memory **300**, a power source **410**, input/output (I/O) ports **420**, an expansion card **430**, a network device **440**, and a display **450**. The data processing device **400** may further include a camera module **460**. In an exemplary embodiment, one or more of the elements of the processing device **400** may be omitted.

[0090] The processor **100** may correspond to the SoC **100** of FIG. **1**. The processor **100** may be a multi-core processor. In an exemplary embodiment, the processor **100** includes the SoC **100** of FIG. **1**. The processor **100** may control the operation of at least one of the elements **200**, **300**, and **410-460**.

[0091] The storage device **200** may correspond to the non-volatile memory device **200** of FIG. **1**. The storage device **200** may be implemented by using a hard disk drive or a solid state drive (SSD).

[0092] The memory **300** may correspond to the main memory **300** of FIG. **1**. The memory **300** may be implemented by using a volatile memory or a non-volatile memory. In an exemplary embodiment, the memory controller **140** of FIG. **2** is capable of controlling a data access operation, for example, a read operation, a write operation (or a program operation), or an erase operation, with respect to the memory **300**. The memory **300** may be may be integrated into or embedded in the processor **100**.

[0093] The power source **410** may supply an operational voltage to at least one of the elements **100**, **200**, **300**, and **420-460**. The I/O ports **420** may be capable of transmitting data to the storage device **200** or transmitting data output from the storage device **200** to an external device. For example, the I/O ports **420** may be a port for connecting a pointing device, such as a computer mouse, to the data processing device **400**, a port for connecting a printer to the data processing device **400**, or a port for connecting a universal serial bus (USB) drive to the data processing device **400**.

[0094] The expansion card **430** may be implemented by using a secure digital (SD) card or a multimedia card (MMC). In an exemplary embodiment, the expansion card **430** is a Subscriber Identification Module (SIM) card or a Universal Subscriber Identity Module (USIM) card.

[0095] The network device **440** may correspond to a device capable of connecting the storage device **200** to a wired or wireless network. The display **450** may display data output from the storage device **200**, the memory **300**, the I/O ports **420**, the expansion card **430**, or the network device **440**.

[0096] The camera module **460** may be capable of converting an optical image into an electrical image. Accordingly, an electrical image output from the camera module **460** may be stored in the storage device **200**, the memory **300**, or the expansion card **430**. The electrical image output from the camera module **460** may be displayed on the display **450**.

[0097] FIG. **15** is a block diagram of a data processing device **500** including the system **10** of FIG. **1**, according to an exemplary embodiment of the present inventive concept. As an example, the data processing device **500** may be implemented by using a laptop computer.

[0098] Similar to the data processing device **400** of FIG. **14**, the data processing device **500** of FIG. **15** includes a processor **100**, a storage device **200**, a memory **300**, a power source **510**, input/output (I/O) ports **520**, an expansion card **530**, a network device **540**, and a display **550**. The data processing device **500** may further include a camera module **560**. In an exemplary embodiment, one or more of the elements of the processing device **500** may be omitted.

[0099] FIG. **16** is a block diagram of a data processing device **600** including the system **10** of FIG. **1**, according to an exemplary embodiment of the present inventive concept. Referring to FIGS. **1** and **16**, the data processing device **600** may be implemented by using a portable device.

[0100] The portable device may be implemented by using a mobile phone, a smart phone, a tablet PC, a personal digital assistant (PDA), an enterprise digital assistant (EDA), a digital still camera, a digital video camera, a portable multimedia player (PMP), a personal (or portable) navigation device (PND), a handheld game console, or an e-book.

[0101] Similar to the data processing device **400** of FIG. **14**, the data processing device **600** of FIG. **16** includes a processor **100**, a storage device **200**, a memory **300**, a power source **610**, input/output (I/O) ports **620**, an expansion card **630**, a network device **640**, and a display **650**. The data processing device **600** may further include a camera module **660**. In an exemplary embodiment, one or more of the elements of the processing device **600** may be omitted.

[0102] FIG. **17** is a block diagram of a system-in package (SiP) **700** including the SoC **100** of FIG. **1**, according to an exemplary embodiment of the present inventive concept, and the non-volatile memory device **200** of FIG. **1**. FIG. **18** is a block diagram of a SiP **700'** including the SoC **100** of FIG. **1**, according to an embodiment of the present inventive concept. In an exemplary embodiment, a SIP may be referred to as a Chip Stack MCM (multi chip module). A SiP may include a number of integrated circuits enclosed in a single module or package.

[0103] Referring to FIGS. **1** and **17**, the SoC **100** and the main memory **300** are packaged into the SiP **700**. Thus, the non-volatile memory **200** is located outside the SiP **700** and may be connected to a pin of the SiP **700**. Referring to FIGS. **1** and **18**, the SoC **100**, the non-volatile memory device **200**, and the main memory **300** are all packaged into the SiP **700'**.

[0104] An SoC according to an exemplary embodiment of the present inventive concept encrypts data within the SoC, and thus a probe of communications between the SoC and other devices may be prevented from accessing un-encrypted data. Further, the SoC according to an exemplary embodiment stores an encryption key for encryption within itself without outputting the key outside itself, and thus the encryption key is prevented from being exposed.

[0105] In an exemplary embodiment of the inventive concept, software of the SoC is incapable of accessing an encryption key used for encryption, which may prevent the encryption key from being leaked due to hacking. Since the software of the SoC does not participate in encryption, a burden on the software is reduced. In an exemplary embodiment, the SoC includes a directly connected encryption/decryption engine and a memory controller, thereby yielding a shortened data transmission path within the SoC. Therefore, the SoC may have improved performance.

[0106] Although exemplary embodiments of the present inventive concept have been shown and described, it will be appreciated by those skilled in the art that various changes may be made in these embodiments without departing from the spirit and scope of the inventive concept.

What is claimed is:

1. A method of operating a system on chip (SoC), the method comprising:

converting plain data into cipher data by an engine within the SoC using a encryption key; and

transmitting, by the engine, the cipher data directly to a memory controller within the SoC,

wherein the memory controller controls an operation of a non-volatile memory.

2. The method of claim **1**, wherein prior to the converting, the method comprises the SoC reading the plain data from a main memory via a bus under the control of a central processing unit (CPU).

3. The method of claim **1**, wherein prior to the converting, the method comprises a direct memory access (DMA) unit within the SoC outputting the plain data to the engine.

4. The method of claim **1**, wherein prior to the converting, the method comprises a one-time programmable (OTP) memory within the SoC outputting the encryption key to the engine.

5. The method of claim **1**, wherein prior to the converting, the method comprising outputting the encryption key to the engine only while a secure program is being executed.

6. The method of claim **1**, wherein the converting comprises converting the plain data into the cipher data in units of blocks.

7. A method of operating a system on chip (SoC), the method comprising:

receiving, by an engine within the SoC, cipher data directly from a memory controller within the SoC, wherein the memory controller controls an operation of a non-volatile memory; and

converting, by the engine, the cipher data into plain data using a encryption key.

8. The method of claim **7**, wherein prior to the converting, the method comprises transmitting the plain data from a direct memory access (DMA) unit within the SoC to the engine.

9. The method of claim **7**, wherein prior to the converting, the method comprises a one-time programmable (OTP) memory within the SoC outputting the encryption key to the engine.

10. The method of claim **7**, wherein prior to the converting, the method comprises outputting the encryption key to the engine only while a secure program is being executed.

11. The method of claim **7**, wherein the converting comprises converting the cipher data into the plain data in units of blocks.

12. A system on chip (SoC) comprising:

an encryption/decryption engine which encrypts first plain data into first cipher data or decrypts second cipher data into second plain data, using a encryption key; and

a memory controller directly connected to the encryption/decryption engine, wherein the memory controller transmits the first cipher data to a non-volatile memory or receives the second cipher data from the non-volatile memory.

13. The SoC of claim **12**, further comprising a one-time programmable (OTP) memory which stores the encryption key.

14. The SoC of claim **12**, further comprising a direct memory access (DMA) unit that receives the first plain data from a device outside the SOC and transmits the first plain data to the encryption/decryption engine or transmits the second plain data received from the encryption/decryption engine to the device.

15. The SoC of claim **14**, wherein the DMA unit is directly connected to the encryption/decryption engine.

16. The SoC of claim **12**, further comprising a central processing unit CPU which controls transmission of the first plain data or the second plain data between a device outside the SoC and the encryption/decryption engine.

17. A system-in package comprising:

the SoC of claim **12**; and

a device which communicates data with the non-volatile memory under the control of the SoC.

18. A system-in package comprising:

the SoC of claim **12**;

the non-volatile memory; and

a device which communicates data with the non-volatile memory under the control of the SoC.

19. A system on chip (SoC) comprises:

a memory controller configured to control a non-volatile memory; and

an encryption/decryption engine directly connected to the memory controller and configured to encrypt or decrypt data,

wherein the SoC controls transmission of data between a device outside the SoC and the non-volatile memory, and

wherein the memory controller and the engine correspond to a first data path for transmitting data.

20. The SoC of claim **19**, wherein the SoC comprises a one-time programmable (OTP) memory storing a key, and the encryption/decryption engine encrypts or decrypts the data by using the key stored in the (OTP) memory.

21. The SoC of claim **19**, wherein the first data path further comprises a direct memory access (DMA) unit which receives data from the device and transmits the data to the encryption/decryption engine or receives data from the encryption/decryption engine and transmits the data to the device.

22. The SoC of claim **21**, wherein the DMA unit is directly connected to the encryption/decryption engine.

23. The SoC of claim **22**, further comprising a second data path which transmits the data which has not been encrypted.

24. The SoC of claim **23**, further comprising a selection circuit which selects either the first data path or the second data path based on a selection signal.

25. The SoC of claim **24**, further comprising a selection signal generator which generates the selection signal,

wherein the selection signal generator is an one-time programmable (OTP) memory or a register.

26. A system on chip (SoC) comprising:

a data bus;

a main memory controller configured to output plain data to the bus;

an engine configured to encrypt the plain data from the bus into cipher data using a key;

a non-volatile memory (NVM) controller;

a first electrical path connecting the bus to the NVM controller to bypass the encryption engine; and

a second electrical path connecting the bus to the NVM controller through the encryption engine,

wherein the SoC activates only the first electrical path in a non-secure mode for sending the plain data from the bus to the NVM controller, and

wherein the SOC activates only the second electrical path in a secure mode for sending the plain data to the engine and the cipher data from the engine to the NVM controller.

27. The SoC of claim **26**, wherein the first electrical path comprises a path of the plain data through a multiplexer and a demultiplexer to the NVM controller.

28. The SoC of claim **27**, wherein the second electrical path comprises a path of the plain data through the multiplexer to the engine and a path of the cipher data through the demultiplexer to the NVM controller.

29. The SoC of claim **28**, further comprising a one-time programmable (OTP) configured to provide a same selection signal to the multiplexer and the demultiplexer for activating one of the first and the second electrical paths.

30. The SoC of claim **26**, wherein the engine is configured to decrypt cipher data received from the NVM controller across the second electrical path.

* * * * *