Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5154769, 11 pages
https://doi.org/10.1155/2021/5154769

Research Article

An Easy-to-Integrate IP Design of AHB Slave Bus Interface for the Security Chip of IoT

Conggui Yuan,[1] Xin Zheng ,[2] Bo Rao,[2] and Shuting Cai[2]

[1]*School of Electronic Information, Dongguan Polytechnic, Dongguan, China*
[2]*School of Automation, Guangdong University of Technology, Guangzhou, China*

Correspondence should be addressed to Xin Zheng; xinzheng9209@gmail.com

Received 16 August 2021; Revised 29 September 2021; Accepted 18 October 2021; Published 1 November 2021

Academic Editor: Chuan Li

Information security is fundamental to the Internet of things (IoT) devices, in which security chip is an important means. This paper proposes an Advanced High-performance Bus Slave Control IP (AHB-SIP), which applies to cryptographic accelerators in IoT security chips. Composed by four types of function registers and AHB Interface Control Logic (AICL), AHB-SIP has a simple and easy-to-use structure. The System on Chip (SoC) design can be realized by quickly converting the nonstandard interface of the security module to the AHB slave interface. AHB-SIP is applied to the security accelerators of SM2, SM3, and SM4 and random number generator (RNG). Combined with a low-power embedded CPU, TIMER, UART, SPI, IIC, and other communication interfaces, a configurable SoC can be integrated. Moreover, SMIC 110 nm technology is employed to tape out the SoC on a silicon chip. The area of AHB-SIP is 0.072 mm^2, only occupying 6‰ of the chip (3.45*3.45 mm^2), and the power consumption of encryption modules combined with AHB-SIP is lower than that combined with AXI interface, which is decreased up to 61.0% and is ideal for the application of IoT.

1. Introduction

The IoT is a network system that is extended and expanded on the Internet and connects people, devices, and servers. With the popularity of intelligent terminals and the rapid development of artificial intelligence, the majority of intelligent nodes will have the access to the Internet in the future. Regardless of its advantages, IoT technology has caused various security threats, such as the leakage of user privacy information and the attack vulnerability of hard-coded security keys [1, 2]. At the end of 2016, a large number of IoT devices were infected with the Mirai malware. The hackers formed a botnet and launched a DDoS attack against Dyn, a globally DNS provider. Consequently, consumers could not pay on PayPal websites, and users could not log in to social networking sites such as Twitter and Tumblr [3]. Frustaci et al. mention that security is the key issue of IoT [3]. Therefore, low-power, secure, and real-time physical layer SoC security chips play a crucial role in the IoT security domain. Besides, it is of great importance to efficiently design this security chip. Shorter life cycles of products can significantly reduce the time-to-market and rapid simulation capabilities are necessary with the increase of the design space at the early stages of design [4]. In this regard, this study focuses on the design of a highly efficient, low-power, and easy-integrated IP interface and integrated crypto modules.

Five SoC bus standards have been widely used in the design of bus interfaces, including the AMBA Bus [5], the Wishbone Bus [6], the CoreConnect Bus [7], the Avalon bus [8], and the OCP bus [9]. The AMBA is a bus standard for high-performance embedded systems. With many third-party supports, the AMBA has become one of the existing widely supported interconnection standards [5]. The CoreConnect bus is a fully constructed general-purpose solution that can connect high-performance systems such as workstations, but it may be too complex for simple embedded applications [7]. The Wishbone bus and the OCP bus are extensively applied in small embedded systems. The Avalon bus only applies to a series of programmable logic devices

(PLD) [6, 9]. The difference between these SoC buses is the features they provide and the integrity of the specification. According to the reference [10], multiple asynchronous AHB bus interface units were present, which allowed the communication of an OpenGL ES 2.0 vertex shader (VS) processor with other hardware units through the AHB bus in the case of different frequencies. The interface of IDE hard disk, reconfigurable arbiter, and DMA controller were designed with the AHB bus interface in [5, 11, 12], respectively. It is possible to interchangeably adopt the majority of AHB slaves in an AHB-Lite or AHB system. The slave designed for the AHB-Lite system will work in the full AHB and AHB-Lite designs.

In this study, the slave modules are security accelerators. To the best of our knowledge, sensitive information can be protected by utilizing cryptographic algorithms in the proposed solutions. Cryptographic algorithms are classified into three categories: symmetric cryptographic algorithms, asymmetric cryptographic algorithms, and hash algorithm. With the characteristics of high efficiency and low overhead, symmetric cryptographic algorithms (such as AES, SM4, and DES) are suitable for big data encryption. Asymmetric cryptographic algorithms, also known as public-key algorithms (such as RSA, SM2, and ECC), show high security. However, due to the large size of the key, it does not apply to big data encryption. Hash algorithms (such as SHA-1, SHA-256, and SM3) are mainly used to generate a message digest with a fixed length. A configurable SoC with built-in FPGA logic gates that can achieve multiple algorithms for AES and DES is proposed in [13]. A SoC is developed in [14], which can be used in the field of mobile security, but it does not apply to the IoT due to its size and power constraints. A SM3 algorithm integrated into financial IC card is designed in [15], which has low power and small area. In [16], a codesign method is employed to propose an AES-ECC hybrid cryptosystem and an interesting trade-off exist between area occupation and speed. Crypto modules are different in terms of functions and interfaces. Therefore, the traditional method changes the nonstandard interface of specific modules into the AHB slave interface. However, due to the different functions of the slave modules, solving this issue will cost mass manpower and resources, leading to a longer product development cycle and higher costs.

For high-performance synthesizable design, the Advanced High-performance Bus Lite (AHB-Lite), as a part of the AMBA, can be employed in IoT chips. It is a transport interface that supports separate transport and provides excellent data transfer capability.

Compared with a complete AHB master, a transport interface can greatly simplify the interface design if masters are designed based on the AHB-Lite interface specification. All masters designed by the full AHB specification apply to an AHB-Lite system with no modification. Although the AHB bus has been widely used in the SoC, the study of AHB slave interface design for security chips is scarce. By analyzing the advantages and disadvantages of different SoC buses and considering the context of practical applications, this study introduces four functional registers and designs a simple and efficient slave bus controller in combination with the AHB-Lite protocol. Moreover, based on the study of symmetric cryptography, public-key cryptography, and hash algorithms, AHB-SIP for cryptographic accelerators in an IoT security chip is proposed. Even without detailed knowledge of the AHB bus protocol, designers can quickly transform a cryptographic accelerator with a nonstandard interface into an accelerator of the AHB slave interface through AHB-SIP. Therefore, AHB-SIP can improve the design efficiency of implementing an SoC system. Based on the above motivation, we make the following contributions:

(i) Based on the AHB-Lite bus, an easy-to-integrate and fast AHB-SIP IP is proposed, which can quickly convert from a nonstandard interface to an AHB-Lite interface. The slave security modules can be easily integrated into an SoC via the AHB-SIP, and all the slave modules can be configured by software.

(ii) The AHB interface control logic that is the key part of AHB-SIP is proposed, which is equipped with strong data transfer capability and low resource consumption.

(iii) This design is taped out on a silicon chip with SMIC 110 nm process. As the experiment results reveal, the area of AHB-SIP only accounts for 6‰ of the chip, and the security accelerators integrated with AHB-SIP can rapidly achieve the encrypted results.

(iv) We integrate three different security accelerators, which can meet the requirement of IoT devices. Specifically, the power consumption of AHB-SIP-based security accelerators is lower than that of AXI-based security accelerators.

The remainder of this paper is structured as follows: Section 2 introduces the background of the AHB. Section 3 describes the design of AHB-SIP. Implementation and integration of cryptographic accelerators are proposed in Section 4, and the results and analysis are shown in Section 5. Section 6 concludes the whole paper.

2. Background of AMBA AHB

The Advanced Microcontroller Bus Architecture (AMBA) is a high-performance embedded microcontroller on-chip communication standard proposed by ARM [17], which has become one of the most popular on-chip bus systems. The AMBA 2.0 bus standard defines three kinds of buses: the Advanced High-performance Bus (AHB), the Advanced System Bus (ASB), and the Advanced Peripheral Bus (APB) [18]. Figure 1 presents a typical AMBA system structure.

In Figure 1, high-performance and high-throughput modules, such as CPU, DMA, and RAM, are connected by the AHB bus. The ASB bus is a high-performance bus that can connect microprocessors and system peripherals. Compared with the AHB bus, the ASB has smaller data width, and a bidirectional data bus is used. Being simple and easy to use, the APB is generally applied in low-speed modules such as UART and SPI. Among the AMBA systems, the most widely used buses are the AHB and the APB. The AHB-Lite bus is simplified based on the AHB, where the

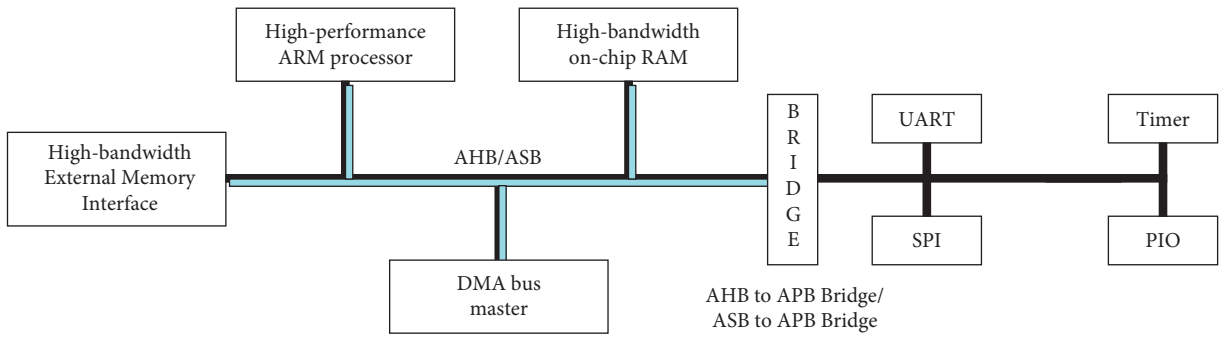

Figure 1: The block diagram of AHB system structure.

AHB supports multiple masters while the AHB-Lite supports only one master. Therefore, it is unnecessary to design an arbiter for the AHB-Lite. Generally, one master is designed in the security chip of IoT, so that the AHB-Lite bus protocol can be considered to use.

The SoC system with the AHB-Lite bus consists of three parts: master, slave, and infrastructure. The master device launches the data transmission, and the slave devices respond after receiving the access request from the master. As shown in Figure 2, an AHB-Lite system is composed of a slave-to-master multiplexor and an address decoder. The address from the master is monitored by the decoder to select the appropriate slave, and the multiplexor routes route the corresponding slave output data back to the master [19]. In our design, the requirements of high-performance synthesizable design are met by using the AHB-Lite.

3. Design of AHB-SIP

AHB-SIP is designed to easily integrate the security units into SoC, which can be configured by software through AHB-SIP, thereby improving the design efficiency of SoC. For the general high-performance computing module, the interface can be classified into four categories: data input, data output, control, and status. Based on different kinds of signals, data interaction is realized by designing four function registers (the status register, the control register, the output register, and the input register), so that the slave modules can be controlled. As is shown in Figure 3, the AHB-SIP consists of four function registers and an AHB Interface Control Logic (AICL) module. In our proposed design, the security units are the slave, and the embedded CPU is the master. The AHB-SIP transfers data between the master and the slave.

3.1. AHB Bus Interface Control Logic. In our design, the control logic is implemented based on the AHB timing. Figure 4 demonstrates the diagram of the AHB protocol sequence in the basic transmission mode.

The control logic is designed according to the AHB bus time sequence, which transfers data between the master and the function registers. The specific functions are divided into the following two aspects:

(1) When the master issues the write request of writing the control value, the data will be written from the master to the corresponding control register. The data will be written to the corresponding input register to write the ordinary data.

(2) When the master issues the read request, if the current status is required to be obtained, the value in the status register will be transmitted to the master. To read the ordinary data, the data in the output register will be sent to the master.

The AICL consists of the slave-to-master multiplexer, the data distributor, the address decoder, and the control logic. Different signals on the AHB bus are read by the control logic, and the control signals will be generated to control the data distributor, address decoder, and multiplexer. In this way, data can be read or written from registers. When the master reads data, the data selector outputs the data to the bus from the specified register based on the address signal and the control signal. When the master writes data, the data distributor will write the data to the corresponding register based on the address decoding result. The AICL module is shown in Figure 5.

3.2. Function Register. The four function registers are mainly adopted for control, calculation, data interaction, and reading status, which can not only realize effective control of the cryptographic accelerators but also obtain their current status for software debugging. Finally, with a 32 bit low-power embedded CPU as the master, the AHB-SIP is employed to integrate the three cryptographic modules and random number generation module into an SoC. The four types of registers are described as follows:

(1) Control register: the control register is utilized to control the start, stop, and working modes of the slave module (such as encryption, decryption, and random number generation)

(2) Input register: the data to be processed by the slave module from the master module are stored by the data input register

(3) Output registers: this type of register can store the data that have been processed by the slave module and are required to be transmitted to the master

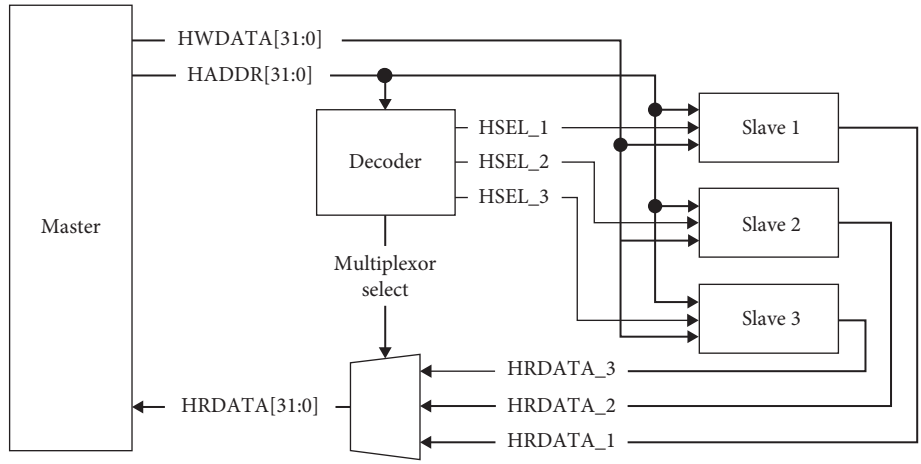

Figure 2: The AHB-Lite block diagram; reprinted from AMBA 5 AHB-Lite protocol [17].

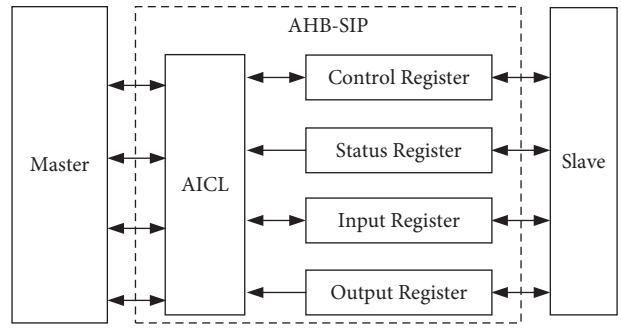

Figure 3: The block diagram of AHB-SIP structure.

(4) Status register: the status of the slave module, such as the mode, the status, and whether the operation is completed, is reflected by the status register

4. Integration of Cryptographic Accelerators

In the SoC, the cryptographic accelerators include the SM2, SM3, SM4, and RNG modules. This section investigates how to integrate these cryptographic accelerators into the SoC quickly by using the AHB-SIP to improve design efficiency. The cryptographic modules are connected with the CPU through AHB. To communicate with disparate IoT devices, the IIC, SPIs, GPIOs, and UARTs are also integrated into the security SoC. Besides, an SRAM is employed to run a real-time operating system (RTOS) and interact with cloud servers. Figure 6 is the architecture of the SoC. This section mainly presents the integration of cryptographic modules.

4.1. Integration of the SM2 Accelerator. SM2 is implemented based on the elliptic curve over GF(p) [20, 21]. The SM2 module is composed of modular operations and scalar multiplication operations. In our design, we utilize the binary extended Euclidean algorithm and the interleaved modular multiplication algorithm to decrease power consumption and chip area [14, 22]. Multiple 256 bit multiplexers, four 256 bit registers, and two 256 bit address are the main hardware overhead of SM2. The structure of the SM2 accelerator can be found in [23]. From the structure, it is observed that the SM2 is a 256 bit ECC. The input data include $(x1, y1)$, $(x2, y2)$, 256 bit key k, and the output data include $(x3, y3)$. Thus, 56 32 bit data registers are needed. The modes of SM2 include point multiplication (PM), multiple point (MP), point addition (PA), modular inverse (MI), modular multiplication (MM), modular subtraction (MS), and modular addition (MA). Therefore, we design a 32 bit status register and a 32 bit control register.

The control register of SM2 is responsible for controlling the computing pattern (enable, disable, or reset). The function of the control register is described in Table 1. For the enable control bit, it will be set to 1 automatically after completing the calculation. The reset control bit must be cleared before writing data. Otherwise, this module is always in reset.

The status register is designed to record the current working status of the SM2 accelerator so that the CPU can achieve the status of this module in real-time. SM2 has four states, which are idle, calculating, finish, and error, respectively.

4.2. Integration of the SM3 Accelerator. To meet the requirement of low power consumption, the proposed SM3 cryptographic accelerator mainly expands and compresses messages that are the most time-consuming parts. The padding and parsing processes are developed by software.

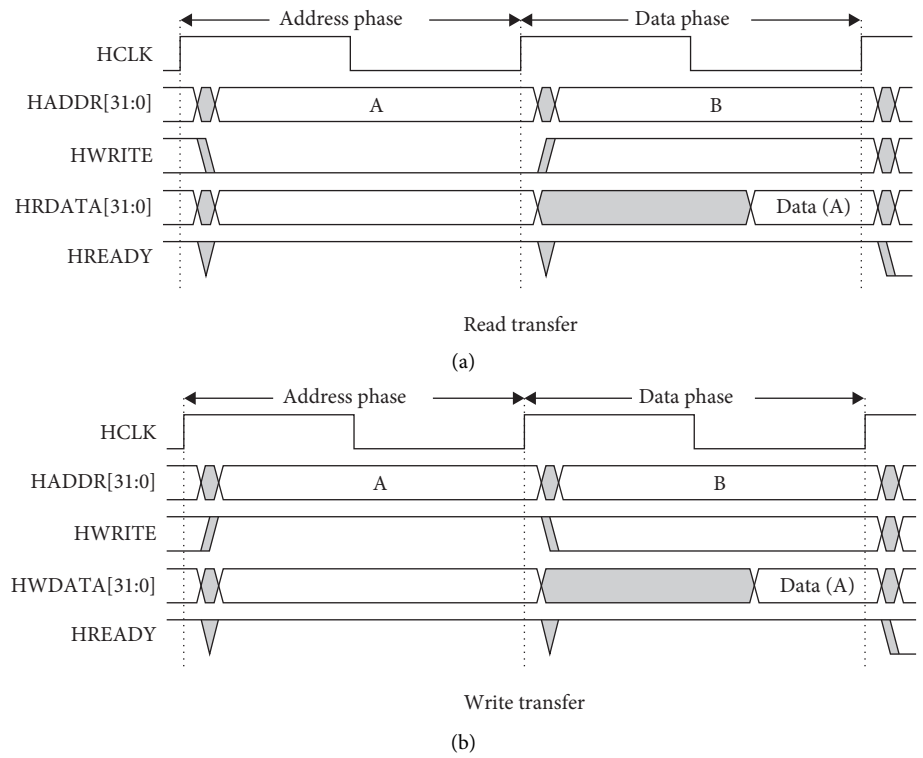

Figure 4: AHB protocol sequence in the basic transmission mode; reprinted from AMBA 5 AHB-lite protocol [17]. (a) Read transfer. (b) Write transfer.

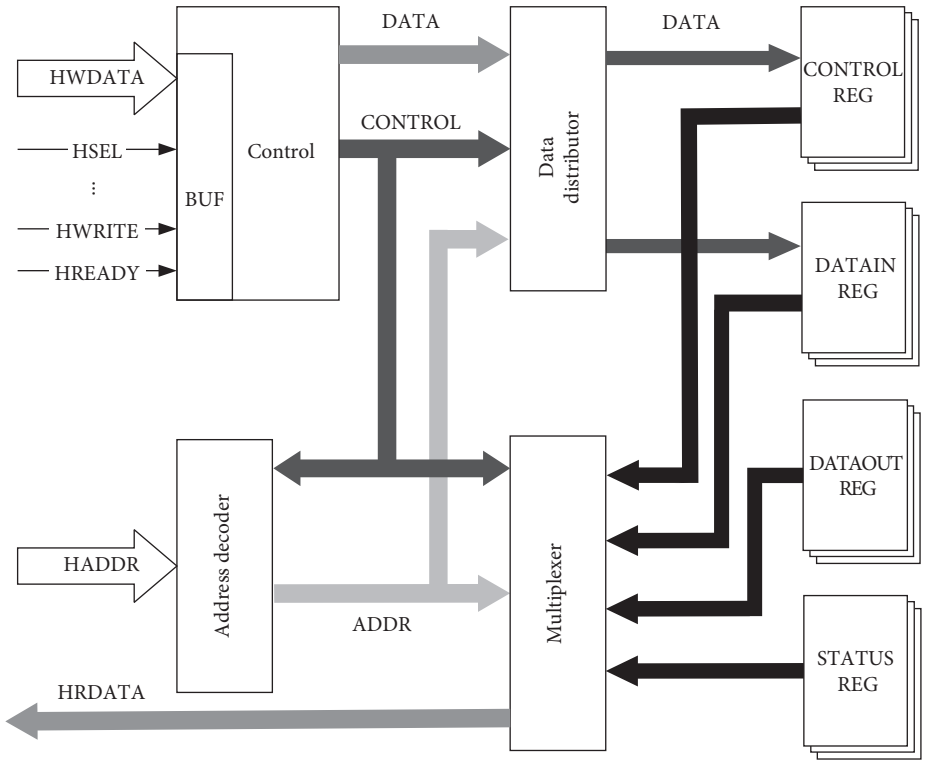

Figure 5: The block diagram of AICL structure.

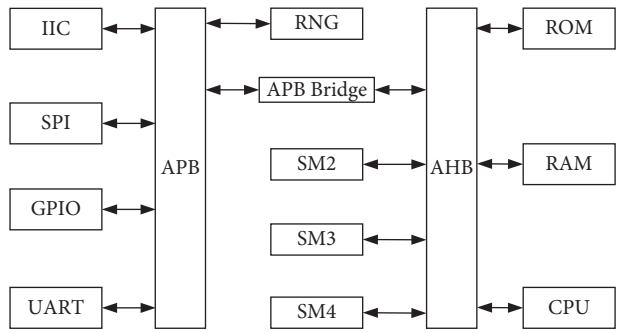

Figure 6: The architecture of the security SoC.

Finally, the 256 bit hash result can be obtained. The detailed codesign procedure can be found in [24].

The input signals of the SM3 accelerator include 512 bit input x, read control signal r, and write control signal w. The output signals include the 256 bit hash value y, the finish signal f, and the state signals. Therefore, this study designs a 32 bit status register, a 32 bit control register, eight 32 bit output registers, and 16 32 bit input registers. The control register of SM3 is responsible for the write/read control, enable/disable, and reset. Before writing to the module, bit5 is set to 0, bit4 and bit3 are set to 1, and then data are written to the input register. After writing the data, bit4 is set to 0, and the module starts the calculation. Once the operation is completed, the result can be read by setting bit4 to 1 and bit3 to 0. If there are several data blocks to be encrypted, bit4 and bit3 are set to 1, and the data are written into the input register until all operations are completed. The function of this control register is described in Table 2.

The status register mainly presents four working states and the exception of SM3. SM3 has four basic states, which are idle, writing, encrypting, and finish, respectively.

4.3. Integration of the SM4 Accelerator. SM4 accelerator contains the round key generation circuit part and the encryption/decryption circuit part. The 128 bits message could be encrypted with 32 clocks. The architecture of SM4 is depicted in [23]. For each group of plaintext M, the ciphertext will be generated after 32 round encryptions. The input signal of the SM4 module consists of 128 bit data input, 128 bit data output, status, and control signal. Therefore, it is necessary to set one 32 bit status register, one 32 bit control register, four 32 bit output registers, and four 32 bit input registers.

The function of the SM4 control register is described in Table 3. First, bit2 and bit3 are set to 1 before data encryption/decryption. Second, the 128 bit key or message is written to the input register. Finally, the corresponding data flag is set so that the module can identify the type of input data. It is worth noting that, since the round key is used in descending order for the encryption process, the message can be directly written to the input register after the key is loaded. For the decryption process, the data to be decrypted cannot be input until the round key has been generated.

The status register of SM4 is designed to present the current work mode, including the encryption mode, the decryption mode, whether the round key is generated, and whether the encryption/decryption process is completed.

In addition, the RNG module in the proposed SoC is an intellectual property depicted in [23]. The ring oscillators are employed to generate pseudo-random numbers or high-speed true random numbers. It consists of an online test module, a postprocessing module, and a high entropy true random source. The standard NIST SP800-22 test is carried out to verify the validity and stability of RNG.

4.4. Overall Steps of the Proposed Method. In this paper, a new method of easy-to-integrate IP design of the AHB slave bus interface for the security chip is proposed, which consists of two steps:

(i) First, the master and slave modules of the system should be determined before designing the interface IP, and the corresponding address space is allocated to these modules through the address decoder.

(ii) Second, the AHB interface control module is designed according to the modules in this security chip and AHB-Lite bus protocol.

(iii) Third, the required function registers for each slave module are designed and integrated with the AHB interface control module.

(iv) Fourth, CPU, memory (RAM and ROM), and security modules (SM2, SM3, and SM4) are integrated into SoC through AHB-SIP, and the functional registers of each security module are designed and configured. CPU is the master module of AHB-Lite bus, while other modules are the slave modules.

(v) Fifth, the RNG module and other low-speed modules are mounted on APB bus through APB bridge that is also the slave module of AHB-Lite bus.

(vi) Finally, the software calls the underlying operation of the hardware security module via CPU. The CPU reads and writes registers using the mode of bus addressing. The encryption/decryption operations are implemented by configuring the function registers of each security module through the CPU, thus realizing the data interaction between software and hardware.

Table 1: The function description of SM2 control register.

Bit	Operation	Type	Description
[3 : 0]	Mode control	R/W	0001 : MM mod N 1001 : MM mod P 0001 : MM mod N 1001 : MM mod P 0010 : MA mod N 1010 : MA mod P 0011 : MS mod N 1011 : MS mod P 0000 : MI mod N 1000 : MI 0101 : PA 0110 : MP 0111 : PM 1111: Idle
4	Enable control		0: Enable 1: Disable
5	Reset control		1: Reset
[31 : 6]	Reserve		

Table 2: The function description of the SM3 control register.

Bit	Operation	Type	Description
[2 : 0]	Reserve	R/W	
3	Data read/write		0: data read 1: data write
4	Enable control		0: enable 1: disable
5	Reset control		1: reset
[31 : 6]	Reserve		

Table 3: The function description of the SM4 control register.

Bit	Operation	Type	Description
[1 : 0]	Data flag	R/W	01: key 10: data to be encrypted 11: data to be decrypted 00: invalid input data
2	Data update		0: no update 1: update
3	Reset control		0: rset
[31 : 4]	Reserve		

5. Experiment Results and Analysis

We first analyze the reasons why we choose the AHB-Lite bus as the SoC bus is that it will be more efficient and save resources. With the rapid development of SoC systems, there are increasing demands for SoC buses. For the widely used SoC bus standards, the AMBA is a bus with complete functions and advanced protocols. In the AMBA, the AHB is an advanced high-performance bus, and AXI focuses on the advanced extensible interface. The bus latency in AHB is lower than that of AXI, and the AHB bus is used more frequently than AXI. As a subset of AHB, the AHB-Lite protocol supports only one master device, and there is no need for the arbiter and the request/authorization protocol. The goal of our design is to develop an efficient and low-power consumption information security chip that can be used for various intelligent hardware platforms and smart home devices. The structure of this chip requires only one master device, which mainly focuses on high-performance and low-power SoC design. To this end, we finally choose the AHB-Lite bus as the SoC system bus.

On the other hand, it is complicated to design a highly dedicated SoC, especially if the structure of the on-chip bus is based on unfamiliar or new protocols. It is difficult to accurately predict the architectural performance via an unfamiliar bus protocol, resulting in the risk of tape-out. Furthermore, the design period is prone to delay because of using a new protocol. The lack of easy-to-use bus interface IP makes the verification environment setup and test vector design more complex. Before communication, it is necessary to ensure that all slave modules have a unified AHB slave interface, or the communication cannot proceed. According to the practical requirements of modules, four functional registers are introduced, and a simple and efficient slave bus controller is designed in combination with the AHB-Lite protocol. Compared with the existing technology, our proposed interface IP and method are featured with the following advantages:

(1) Four functional registers for the communication between the slave and the master modules are introduced to realize the data transfer. Thus, it is unnecessary to know exactly about the AHB-Lite bus protocol.

(2) The AHB-Lite bus can realize the data transmission between the master and the slave modules via simply modifying the four types of function registers.

(3) By converting the nonstandard interface into the AHB slave interface via the AHB-SIP, the SoC system design can be achieved efficiently. The risk of chip tape-out can be reduced, the design period can be shortened, and the performance of SoC can be enhanced.

By utilizing the AHB-SIP and the integration method described in Section 4, we successfully integrate SM2, SM3, SM4 cryptographic accelerators, IIC, SPI, GPIO, UART interfaces, and RNG module into an SoC, accomplishing a low-power IoT security chip. The security chip is taped out with SMIC 110 nm technology process and QFN56 package technology. The system clock frequency is 36 MHz, and the voltage of core and IO are 1.2 V and 3.3 V, respectively. The area of this chip is $3.45 \times 3.45\,\text{mm}^2$. The gates and area of each module are listed in Table 4.

According to Table 4, SM2, SM3, and SM4 cryptographic accelerators have a total area of about $1.0\,\text{mm}^2$. It is noteworthy that the area of AHB-SIP is $0.072\,\text{mm}^2$, only occupying 6‰ of the chip. Since the 128K RAM is applied to the SoC, it occupies about 1/3 of the chip area.

The ASIC layout is shown in Figure 7. The two RAMs are distributed on the right of the layout. 8 KB ROM is in the upper left of the layout, and the CPU is in the lower left of the layout. The rest are SM2, SM3, SM4 cryptographic accelerators, RNG, and other modules. Since the AHB-SIP is

Table 4: The gates and area of each module.

Module	Gates (K)	Area (mm^2)
AHB-SPI	14.199	0.072
SM2/3/4	199.075	1.013
RNG	57.003	0.29
CPU	88.371	0.45
ROM	70.343	0.358
RAM	763.055	3.886
Others	1202.355	5.834
Total chip	2337.398	11.903

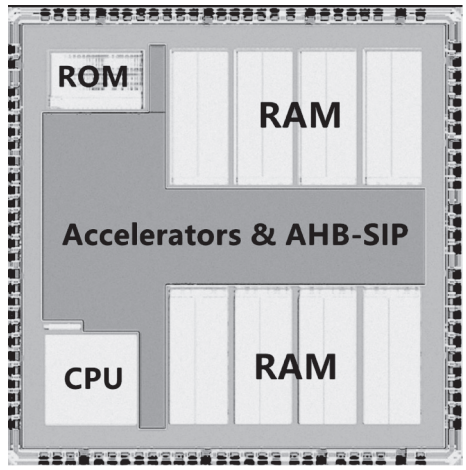

Figure 7: The architecture of the security SoC.

scattered in the layout, the size of AHB-SIP cannot be observed directly from the layout.

Also, to compare with using AXI bus, we experimented to evaluate the total power of crypto accelerator with different SoC bus interfaces. The experiment was implemented on the Xilinx FPGA of Virtex 6 architecture under the frequency of 100 MHz. The ISE Design Suite of Xilinx provides a power simulator XPower Analyzer, which can analyze the power of programable logic devices. By taking the cases of SM2, SM3, and SM4 modules, we found that the power consumption of these encryption modules combined with AHB-SIP was lower than that combined with the AXI interface, which decreased by 61.0%, 49.7%, 48.0%, respectively, as shown in Figure 8. This demonstrates that the fewer hardware resources we used, the lower power is consumed.

For ASIC design, the proposed method is compared with other state-of-the-art schemes to test the performance of the cryptographic accelerators and AHB-SIP. Table 5 lists different implementation methods of the cryptographic accelerators. The results indicate that the proposed method combining the cryptographic accelerators and AHB-SIP provides low power consumption and good performance for the three sorts of cryptographic algorithms.

It can be observed that it is infeasible to compare the results, as technology library, methodologies, and application areas are different. According to Table 5, for the SM2 accelerator, the throughput of PM operation is higher than that in [25], indicating the times of point multiplication per unit time are more than that of [25]. Except for [25], the power consumption of this design is the lowest. Since 40 nm process technology is adopted in [25], no equivalent comparison can be made. In other architectures, the performance of [26] is better than ours, but its area and power consumption are greater. Although the speed of PM operation in [27] is the highest, the area is also the largest. Besides, the number of logic gates is 11.76 times that of the design architecture in our work and approximately triple that of other designs. Considering the cost of developing IoT chips, high power consumption and a large area are inappropriate for IoT chips. For the SM3 accelerator, several implementation methods of the hash algorithm are listed in Table 5. As Table 5 reveals, the implementation method proposed in [24] has high throughput, small area, and high power consumption. Although 886 gates are required in the SHA-3 design in [25], the power consumption and throughput are inferior to our design. At the normalized frequency, the throughput of our design is 13.8 times higher compared with the design in [28]. Compared with the AES architecture implemented in [25], the power consumption of the SM4 accelerator is close to ours at the same frequency, while the efficiency is much higher than [25]. Compared with the architecture implementation in [14], the saved gates with the proposed architecture are approximately 197.5 K. It is evident that, although the throughput in [29] is the highest, it has higher power consumption and a larger area than other architectures. Therefore, it does not apply to IoT security chips.

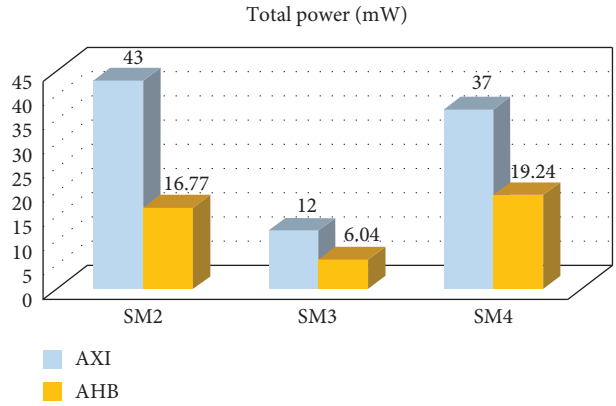

Figure 8: The power comparison among different accelerators combined with different SoC buses of security chip.

Table 5: Performance comparison with other architecture.

Design	library	Frequency (MHz)	Area (gate)	Power (mW)	Throughput
SM2 (our)	0.11-μm	36	56K	3.24	27.64 Kbps
ECC233 [24]	0.04-μm	28.8	-	1.13	8.59 Kbps
SM2 [25]	0.13-μm	214	208K	40.28	1.20 Mbps
SM2 [26]	0.13-μm	163.7	659K	—	12.57 Mbps
SM3(our)	0.11-μm	36	18K	0.18	245.7 Mbps
SM3 [23]	0.13-μm	36	6036	1.24	263 Mbps
SHA-3 [24]	0.04-μm	28.8	886	4.87	14 Mbps
SHA-256 [27]	0.13-μm	102	9036	3.06	47 Mbps
SM4(our)	0.11-μm	36	124K	3.39	115.2 Mbps
AES [24]	0.04-μm	28.8	—	2.8	5.08 Mbps
AES [14]	0.13-μm	200	321.5 K	325	564 Mbps
AES [28]	0.04-μm	1000	9028K	6.17 K	128 Gbps

Table 6: The test result of the RNG.

Statistical Test	Full entropy	High speed
Frequency	990/1000	992/1000
Block frequency	992/1000	995/1000
Runs	984/1000	989/1000
Longest run	988/1000	985/1000
Rank	989/1000	990/1000
FFT	984/1000	989/1000
Cumulative sums	Pass	Pass
Nonoverlapping template	Pass	Pass
Overlapping template	987/1000	987/1000
Universal	992/1000	990/1000
Approximate entropy	990/1000	991/1000
Random excursions	Pass	Pass
Random excursions variant	Pass	Pass
Serial	Pass	Pass
Linear complexity	Pass	Pass

Our designed RNG module in the SoC is tested based on the standard NIST SP800-22. The random numbers to be tested for each set are divided into 1000 groups, with each group containing 1M bit random numbers. According to the NIST standard, if at least 980 of the 1000 random numbers pass a statistical test, it can be considered to pass. We tested a total of five sets. Since the standard of NIST's nonoverlap template matching test is quite strict, if the pass rate is not very poor, it is usually negligible. Thus, the five sets of random numbers are verified to be of quite high quality. Table 6 presents the results of our test.

In conclusion, compared with the above baseline designs, we obtain the following results:

(i) Since the bus latency of AHB is lower than that of AXI and the different structures of the security module design, our proposed method is more efficient than others when using the AHB-SIP.

(ii) We use fewer hardware resources for designing the AHB-SIP, and the area of the chip is smaller. The total power consumption is only 8.4 mW @36 MHz, which is very suitable for IoT devices.

(iii) The results indicate that the balance between the throughput, area, and power consumption of our proposed SoC with AHB-SIP at the normalized frequency is excellent.

6. Conclusion and Future Work

This study proposed a design of AHB-SIP in the field of IoT security, which can easily integrate the security units into an SoC and transform a cryptographic accelerator with a nonstandard interface into an accelerator with the AHB slave interface. Besides, the SM2, SM3, SM4, and RNG security modules are configured by software through AHB-SIP to improve the design efficiency of SoC. Finally, a low-power IoT security chip is realized by using 110 nm process technology. The implementation and test results indicate that the area of AHB-SIP is quite small, the power consumption is lower than AXI-based architecture, and the performance of accelerators is ideal for IoT applications. In the future, it is necessary to study the construction and optimization of AHB-SIP to enhance performance and flexibility.

Data Availability

The Verilog data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Key-Area Research and Development Program of Guangdong Province under Grant 2019B010145001 and in part by the Science and Technology Planning Project of Guangdong Province of China under Grant 2019B010140002.

References

[1] Y. H. Hwang, "IoT security & privacy: threats and challenges," in *Proceedings of the 1st ACM Workshop on IoT Privacy, Trust, and Security-IoTPTS '15*, vol. 1, ACM Press, Singapore, April 2015.

[2] W. Zhou, "The effect of iot new features on security and privacy: new threats, existing solutions, and challenges yet to be solved," *IEEE Internet of Things Journal*, vol. 6, pp. 1606–1616, 2018.

[3] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: present and future challenges," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2483–2495, 2018.

[4] J. E. Siegel, K. Sumeet, and E. S. Sanjay, "The future internet of things: secure, efficient, and model-based," *IEEE Internet of Things Journal*, vol. 54, pp. 2386–2398, 2017.

[5] G. Ma and He Hu, "Design and implementation of an advanced DMA controller on AMBA-based SoC," in *Proceedings of the 2009 IEEE 8th International Conference on ASIC, IEEE*, pp. 419–422, Changsha, China, October 2009.

[6] A. K. Swain and K. Mahapatra, "Design and verification of WISHBONE bus interface for system-on-chip integration," in *Proceedings of the 2010 Annual IEEE India Conference (INDICON), IEEE*, pp. 1–4, Kolkata, India, December 2010.

[7] R. Hofmann and B. Drerup, "Next generation CoreConnect/spl trade/processor local bus architecture," in *Proceedings of the 15th Annual IEEE International ASIC/SOC Conference*, pp. 221–225, IEEE, Rochester, NY, USA, December 2002.

[8] Q. Zhou, S. Yu-Kun, D.-L. Zhang, and G.-M. Du, A design of multi-core system based on A valon bus," in *Proceedings of the 2011 International Conference on Computer Science and Network Technology*, pp. 1456–1459, IEEE, Harbin, China, December 2011.

[9] C.-Y. Chang, Y.-J. Chang, J.-C. Yeh, S. Y. Lin, and J.-L. Ma, "Design of on-chip bus with OCP interface," in *Proceedings of the 2010 International Symposium on VLSI Design, Automation and Test*, pp. 211–214, IEEE, Hsin Chu, Taiwan, April 2010.

[10] S.-F. Hsiao, C.-G. Lin, P.-H. Wu, and C.-S. Wen, "Asynchronous AHB bus interface designs in a multiple-clock-domain graphics system," in *Proceedings of the 2012 IEEE Asia Pacific Conference on Circuits and Systems*, pp. 408–411, IEEE, Kaohsiung, Taiwan, December 2012.

[11] Z.-Y. Li, S.-B. Liu, and Y. Feng, "Design of an interface between the IDE controller and the AHB bus based on FPGA," *Computer Engineering & Science*, vol. 2, 2017.

[12] A. K. Singh, A. Shrivastava, and G. Tomar, "Design and implementation of high performance AHB reconfigurable arbiter for onchip bus architecture," in *Proceedings of the 2011 International Conference on Communication Systems and Network Technologies*, pp. 455–459, IEEE, Katra, India, June 2011.

[13] S. Xuan, J. Han, Z. Yu, Y. Ren, and X. Zeng, "A configurable SoC design for information security," in *Proceedings of the 2015 IEEE 11th International Conference on ASIC (ASICON)*, pp. 1–4, IEEE, Chengdu, China, November 2015.

[14] W. Huang, J. Han, S. Wang, and X. Zeng, "The design and implement of a mobile security SoC," in *Proceedings of the 2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology*, pp. 96–98, IEEE, Shanghai, China, November 2010.

[15] Y. Hu, L. Wu, A. Wang, and B. Wang, "Hardware design and implementation of SM3 hash algorithm for financial IC card," in *Proceedings of the 2014 Tenth International Conference on Computational Intelligence and Security*, pp. 514–518, IEEE, Kunming, China, November 2014.

[16] A. Hafsa, N. Alimi, A. Sghaier, M. Zeghid, and M. Machhout, "A hardware/software Co-designed AES-ECC cryptosystem," in *Proceedings of the 2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, pp. 50–54, Hammamet, Tunisia, January 2017.

[17] "ARM AMBA 5 AHB protocol specification AHB5," *AHB-Lite*, vol. 86, 2001.
[18] *AMBA Specification (Rev 2.0)*, AMBA, ARM Limited, UK, 1999.
[19] S. Kante, H. K. Kakarla, and A. Yadlapati, "Design and verification of AMBA AHB-lite protocol using verilog HDL," *International Journal of Engineering and Technology*, vol. 8, pp. 734–741, 2016.
[20] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology*, H. C. Williams, Ed., vol. 218pp. 417–426, 1986.
[21] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, p. 203, 1987.
[22] C. H. Wang, C. Y. Lo, and M. S. Lee, "A network security processor design based on an integrated SOC design and test platform," in *Proceedings of the 43rd Annual Design Automation Conference*, pp. 490–495, New York, NY, USA, July 2006.
[23] X. Zheng, C. Xu, X. Hu, Y. Zhang, and X. Xiong, "The software/hardware Co-design and implementation of SM2/3/4 encryption/decryption and digital signature system," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 10, pp. 2055–2066, 2020.
[24] X. Zheng, X. Hu, J. Zhang, J. Yang, S. Cai, and X. Xiong, "An efficient and low-power design of the SM3 hash algorithm for IoT," *Electronics*, vol. 8, no. 9, p. 1033, 2019.
[25] Y. Zhang, L. Xu, Q. Dong, J. Wang, D. Blaauw, and D. Sylvester, "Recryptor: a reconfigurable cryptographic cortex-M0 processor with in-memory and near-memory computing for IoT security," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 4, pp. 995–1005, 2018.
[26] D. Zhang and G. Bai, "Ultra high-performance ASIC implementation of SM2 with SPA resistance," in *Information and Communications Security*, S. Qing, E. Okamoto, K. Kim, and D. Liu, Eds., vol. 9543, pp. 212–219, 2016.
[27] Z. Zhao and G. Bai, "Ultra high-speed SM2 ASIC implementation," in *Proceedings of the 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 182–188, IEEE, Beijing, China, September 2014.
[28] X. Cao, L. Lu, and M. O'Neill, "A compact SHA-256 architecture for RFID tags," in *Proceedings of the 22nd IET Irish Signals and Systems Conference*, ISSC, Ireland City, Dublin, June 2011.
[29] G. Sayilar and D. Chiou, "Cryptoraptor: high throughput reconfigurable cryptographic processor," in *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, vol. 155, November 2014.