IEICE TRANS. INF. & SYST., VOL.E105–D, NO.5 MAY 2022

PAPER

A Performance Model for Reconfigurable Block Cipher Array Utilizing Amdahl's Law

Tongzhou QU[†], **Zibin DAI**[†], **Yanjiang LIU**[†a)], **Lin CHEN**[†], ***Nonmembers***, ***and*** **Xianzhao XIA**[††], ***Member***

SUMMARY The existing research on Amdahl's law is limited to multi/many-core processors, and cannot be applied to the important parallel processing architecture of coarse-grained reconfigurable arrays. This paper studies the relation between the multi-level parallelism of block cipher algorithms and the architectural characteristics of coarse-grain reconfigurable arrays. We introduce the key variables that affect the performance of reconfigurable arrays, such as communication overhead and configuration overhead, into Amdahl's law. On this basis, we propose a performance model for coarse-grain reconfigurable block cipher array (CGRBA) based on the extended Amdahl's law. In addition, this paper establishes the optimal integer nonlinear programming model, which can provide a parameter reference for the architecture design of CGRBA. The experimental results show that: (1) reducing the communication workload ratio and increasing the number of configuration pages reasonably can significantly improve the algorithm performance on CGRBA; (2) the communication workload ratio has a linear effect on the execution time.

key words: *coarse-grained reconfigurable block cipher array, block cipher algorithm, Amdahl's law, parallelism, performance*

1. Introduction

Recently, cryptographic processors are widely applied to the communication, finance, and military fields to meet the security requirement of information systems. With the rapid development of information networks, the performance of cipher processors is required to be higher. Of all the existing cryptographic processors, the coarse-grained reconfigurable array (CGRA) is available widely in the embedded systems due to the high-energy efficiency and programmability [1]–[5]. Block cipher algorithm is a typical computation-intensive application, which can give full play to the structural characteristics and high-performance advantages of CGRA. Therefore, the design of high-performance CGRA dedicated to block cipher algorithms has become a research hotspot. Various block cipher algorithms, including the DES [6], [7], AES [8], [9], SM4 [10], IDEA [8], and so on, are performed with CGRA architectures. The CGRA designed for block cipher algorithms is denoted as CGRBA.

In 1967, Dr. Gene M. Amdahl, the father of the IBM mainframe, illustrates the key to parallel computing system design with Amdahl's law [11]. Numerous performance models based on Amdahl's law, including the Amdahl's law for multicore system [12]–[14], Amdahl's law for many-core system [15], [16], Amdahl's law for multicore cryptographic system [17], [18], and Amdahl's law targeted for cost conscious [19], [20] are explored to make Amdahl's law more match with the actual situation. However, those models are only used to evaluate the performance of multi/many-core architectures. Considering the mismatch of several structural parameters, those models cannot be applied to the CGRA. To the best of our knowledge, there is no performance model to evaluate the CGRA architecture.

To addressing this issue, an extended Amdahl's law aiming at CGRBA is proposed in this paper. The multi-level parallelism of block cipher algorithm and its implementation on CGRBA are analyzed. Further, the performance model for the CGRBA is established by introducing several critical performance parameters (communication and configuration overhead). Finally, the hardware simulation is executed, and the optimized parameters are determined using the optimal integer nonlinear programming model and thus the minimum execution time of CGRBA is obtained. The main contributions are listed as follows.

- On the basis of the analysis of multi-level parallelism of block ciphers, we solve the relationship among the processing time, multi-level parallelism of block cipher algorithms, and structural parameters of CGRBA.
- The performance model for CGRBA is established by extending Amdahl's law. And the method to solve the optimal performance is proposed according to our performance model. To the best of our knowledge, this is one of the first attempts to describe the performance of CGRBA based on Amdahl's law.

2. Symbols Definitions and the Structure of CGRBA

Below, we first list the symbols in Table 1 that will be utilized throughout, and then we briefly review the structure of CGRBA.

2.1 Symbols Definitions

The symbols definitions in this paper are shown in Table 1.

2.2 Basic Structure of CGRBA

The structure of CGRBA is shown in Fig. 1, which is mainly

Manuscript received September 22, 2021.
Manuscript revised December 15, 2021.
Manuscript publicized February 17, 2022.

[†]The authors are with the Institute of Information Science and Technology, Zhengzhou, China.
[††]The author is with China Automotive Technology and Research Center, Beijing, China.
a) E-mail: liuyj_1013@126.com

DOI: 10.1587/transinf.2021EDP7195

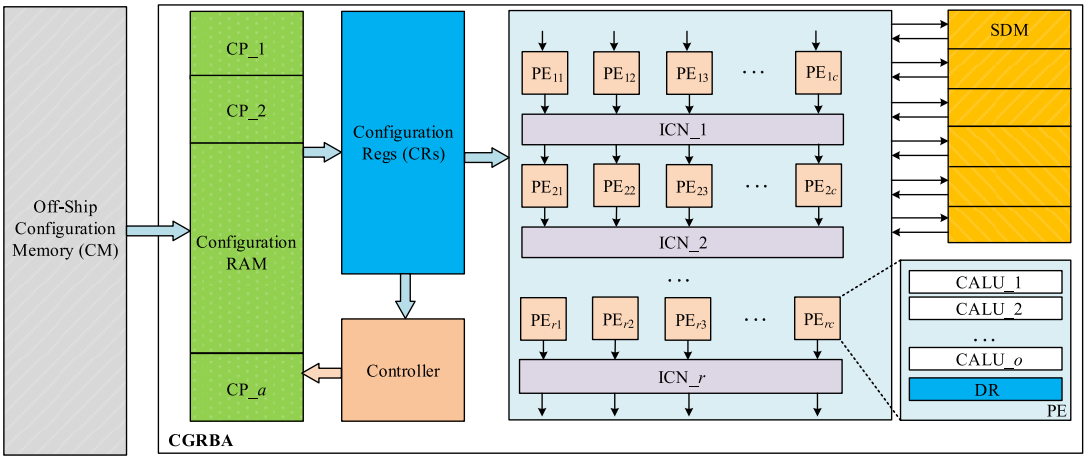

Fig. 1 Basic structure of CGRBA.

Table 1 symbols definition

Symbols	Definition
P_1	Parallelism between the sub-blocks
P_2	Parallelism between the blocks
P_3	Parallelism between the packets
P_4	Parallelism between the subprograms
W	The workload of the block cipher
δ	Workload processing capacity of a PE per unit time
i	The degree of parallelism of a program segment
E	A program performed on CGRBA
e_i^n	A program segment with the degree of parallelism i
w_i^n	The workload of e_i^n
c	The number of PE rows in a CGRBA
r	The number of PE columns in a CGRBA
I	The total number of PEs in a CGRBA
l	The average number of PEs occupied by each algorithm.
k	The row spacing between two PEs
β_k	The communication number between all PEs with The row spacing k
α	The ratio of communication traffic to workload
a	The number of configuration pages in a CGRBA
o	the average number of clocks during per PE working under a CP configuration
$\lceil\ \rceil$	Ceiling
$\binom{x}{y}$	Ratio of x to y

composed of the isomorphic processing elements (PEs), interconnection network (ICN), configuration RAM, configuration registers (CRs), and shared data memory (SDM) and controller. The PE consists of various cryptographic arithmetic logic units (CALUs) and data registers (DRs), and each CALU is capable to realize one kind of cryptographic operation, such as the XOR, Sbox, and Shift. The PEs are arranged in a matrix and communicate via ICN. The key, plaintext, temporary data, and output cipher text are stored in SDM. CGRBA is driven by configuration information and reconstructs its hardware function by changing the configuration information. CGRBA includes three-level configuration memory structures: CRs, configuration RAM and CM. The CRs store the configuration information used to drive PEs and ICNs working. The configuration RAM is divided into multiple configuration pages (CPs). CRs receive the configuration from one of them. Switching of CPs is performed by change configuration RAM's address given by the controller. CM is off-chip configuration memory, which configures block ciphers' configuration to the configuration RAM, and transfers plaintext and key to the SDM.

3. Multi-Level Parallelism Analysis of Block Ciphers

In this section, we first review Amdahl's law and summarize the relationship between the degree of parallelism and threads on the CGRBA. Then, we analyze the multi-level parallelism of block ciphers, and finally, we give a multi-level parallelism execution model of CGRBA.

Amdahl's law states that when a component of the system adopts a faster execution method, the improvement of system efficiency depends on the ratio of the execution time of the component to the total execution time of the system [19]. It can be abstracted as Eq. (1), where $S(f, N)$ is the speedup ratio of the parallel system. For a program E, f is the proportion of the parallel parts of E and N is the number of parallel processors.

$$S(f,N) = \frac{1}{f + \dfrac{1-f}{N}} \tag{1}$$

Additionally, the parallel execution time T_2 is expressed as Eq. (2), where T_1 is the serial execution time of E.

$$T_2 = (1-f)\,T_1 + f \cdot T_1/N \tag{2}$$

In the field of computer architecture, the thread is the smallest unit of the program execution. The degree of parallelism refers to the maximum number of threads executed in parallel of a program [17]. For CGRBA, a thread especially refers to a series of cryptographic operations performed sequentially targeted for a certain data collection. As a distributed computing architecture, CGRBA supports multi-threaded concurrent processing due to the multi-PEs

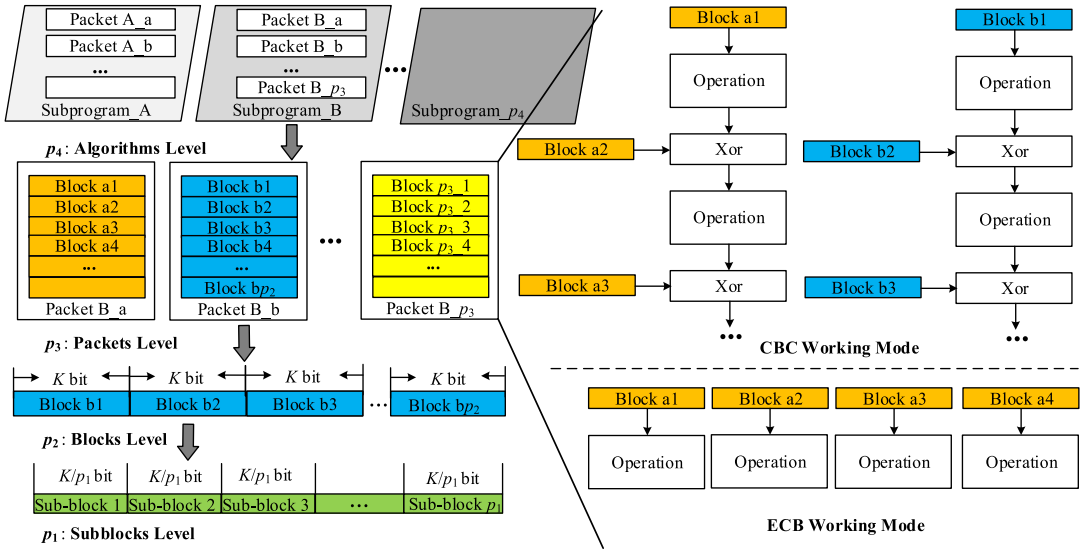

Fig. 2 Four levels parallelisms of block cipher algorithm

hardware structure. Different PEs that perform various cryptographic operations correspond to different threads. Therefore, we can exploit the parallelism of block cipher algorithms on the CGRBA. Amdahl's law has two assumptions. 1) The system has unlimited resources to process, therefore any program can be processed with its maximum degree of parallelism. 2) All parallel parts of E have the same parallelism. However, in fact, the resources of any processor are limited, and a program such as block ciphers may have different levels of parallelism. Therefore, Amdahl's law cannot accurately describe the performance of the block ciphers when executed in parallel on CGRBA.

3.1 Multi-Level Parallelism Analysis of Block Ciphers

To extend Amdahl's law on CGRBA, we study the parallelism of block ciphers and the maximal degree of parallelism that CGRBA can utilize. In this paper, the degree of parallelism specifically refers to the maximum number of threads that can be executed concurrently on CGRBA. And we divide the parallelism of block cipher algorithms into four levels, P_1, P_2, P_3, and P_4, as shown in Fig. 2, and denote the degree of parallelism of $P_j (1 \leq j \leq 4)$ as p_j.

Block is the basic processing unit of block ciphers. Each block can be further divided into several sub-blocks according to the algorithm structure. These sub-blocks may be processed concurrently. So, there is parallelism among the sub-blocks, which kind of parallelism is denoted as P_1. For example, in Fig. 2, block b2 is divided into p_1 sub-blocks that can be executed in parallel. In addition, block ciphers have multiple working modes. In some modes, such as ECB (Electronic Code Book) and CTR (Counter Book) modes, there is no data dependency between different blocks. Under this circumstance, CGRBA is capable to process multiple blocks at the same time, as shown in Fig. 2. In this paper, the parallelism between blocks is denoted as P_2. However, in some working modes such as CBC (Cipher Block Chaining) and CFB (Cipher Feedback), the execution of blocks in the same data packet must be serial, that is, the execution of subsequent blocks must be executed after the previous block. But under this circumstance, the blocks in different data packets are still parallelly executable. We denote the parallelism between packets as P_3. Finally, block ciphers can be divided into multiple subprograms, which implement the same function but have different data input and output. These subprograms can be executed concurrently with enough hardware resources on a CGRBA. We denote the parallelism among the subprograms as P_4.

3.2 Multi-Level Parallelism Realization Model of Block Ciphers on CGRBA

When executed on CGRBA, the block ciphers are divisible into multiple program segments with different degree of parallelism. Let E denote a block cipher program with multiple program segments. And let e_i be a subset of E, in which the degree of parallelism of the program segments are i. Then, E is capable to be represented as set $E = \{e_i | 1 \leq i \leq m\}$, where m is the maximal degree of parallelism of the program segments in E; e_i can be represented as set $e_i = \{e_i^n | 1 \leq n \leq t_i\}$, where e_i^n represents the n-th program segment of e_i, and t_i is the number of program segments in e_i.

As mentioned above, each program segment e_i^n may possess multi-level parallelism, from P_1 to P_4. According to the definition, the degree of parallelism of e_i^n is i. Therefore, E can be expressed as Eq. (3). And Fig. 3 shows two different descriptions of program E, the elements of which are e_i and program segment e_i^n.

$$E = \left\{e_1^1, e_1^2, \ldots, e_1^{t_1}, \ldots, e_m^1, e_m^2, \ldots, e_m^{t_m}\right\} \tag{3}$$

The parallelism realization schematic diagram of block cipher algorithms on CGRBA is illustrated in Fig. 4. In gen-

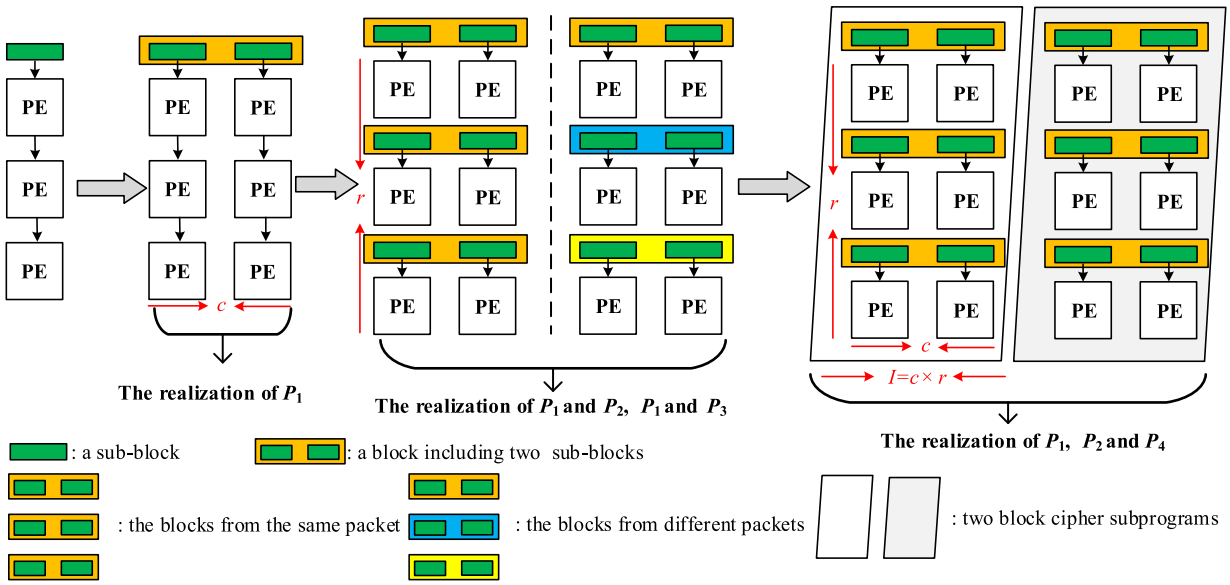

Fig. 4 Four levels parallelism realization process on CGRBA

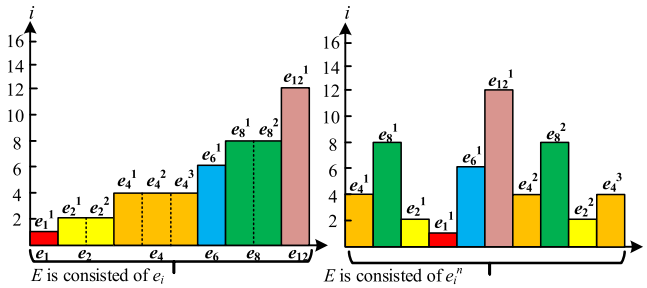

Fig. 3 Two descriptions of a block cipher algorithm E

eral, the processing width of a PE is equal to or greater than the bit width of a sub-block. Therefore, a PE can independently process a certain operation of a sub-block. When mapping block cipher algorithm on CGRBA, multiple operations of a sub-block are usually mapped on multiple rows of PE. And the number of sub-blocks that can be processed in parallel generally depends on the number of PE columns. Consequently, the realization of P_1 depends on p_1 and c. In addition, CGRBA supports pipeline processing by treating each PEs row as a one-level pipelining stack. Hence, the max number of pipeline levels is equal to the number of PE rows, which is also the max number of blocks that can be executed in parallel. It can be concluded that P_2 and P_3 depend on p_2, r, and p_3, r. Therefore, we treat P_2 and P_3 as one parallelism, the degree of parallelism of which is denoted as p_2p_3. For P_4, as shown in Fig. 2, it is determined by p_4, the total number of PEs I, and the number of PEs occupied by a subprogram l.

According to the analysis above, the realization of four levels of parallelism does not conflict with each other, henceforth the degree of parallelism of e_i^n can be expressed as $i = {}_i^n p_1 \cdot {}_i^n p_2 \cdot {}_i^n p_3 \cdot {}_i^n p_4$. When $c \geq_i^n p_1$, $r \geq_i^n p_2\, p_3$, and $I/l \geq_i^n p_4$, all parallelism is fully realized. Let T_i^n and w_i^n represent the execution time and the workload of e_i^n. When the program E is executed serially, T_i^n is w_i^n/δ. And when E is executed in parallel, T_i^n is equal to $w_i^n/\delta \cdot i$. If the above conditions ($c \geq {}_i^n p_1$, $r \geq {}_i^n p_2.\, p_3$, and $I/l \geq {}_i^n p_4$) are not met, i will be greater than the maximal achievable degree of parallelism. In this case, the resources of CGRBA are not enough to realize each level of parallelism, therefore it is necessary to execute multithreading by reusing hardware resources in a time-multiplexed manner. For a program segment with $p_1 > c$ and $p_2 = p_3 = p_4 = 1$, the number of reused hardware resources is $\lceil p_1/c \rceil$. And the execution time is $(w_i^n/\delta \cdot p_1)\lceil p_1/c \rceil$. Similarly, T_i^n is described in Eq. (4).

$$T_i^n = \frac{w_i^n}{\delta} \cdot \left(\left\lceil \frac{{}_i^n p_1}{c} \right\rceil \Big/ {}_i^n p_1 \right) \cdot \left(\left\lceil \frac{{}_i^n p_2 {}_i^n p_3}{r} \right\rceil \Big/ {}_i^n p_2 {}_i^n p_3 \right) \cdot \left(\left\lceil \frac{{}_i^n p_4}{I/l} \right\rceil \Big/ {}_i^n p_4 \right) \tag{4}$$

Let T be the total execution time of E. T includes two parts: 1) The part that can be accelerated by the realization of the algorithm's parallelism, which is denoted as T_p. 2) The parts that cannot be accelerated in parallel, which is denoted as T_s. According to the above analysis, T_p is the sum of all T_i^n, as shown in Eq. (5).

$$T_p = \sum_{i=1}^{m} \sum_{n=1}^{t_i} \frac{w_i^n}{\delta \cdot i} \left(\left\lceil \frac{{}_i^n p_1}{c} \right\rceil \left\lceil \frac{{}_i^n p_2 p_3}{r} \right\rceil \left\lceil \frac{{}_i^n p_4 l}{I} \right\rceil \right) \tag{5}$$

4. A performance Model for CGRBA Based on Amdahl's Law

Now we establish a performance model for CGRBA based on Amdahl's law, which considers key factors affecting the performance of CGRBA, such as communication and configuration overhead, and improves the accuracy of Eq. (5).

Although there exists communication overhead under the multi-PEs architecture of CGRBA, it ensures that multiple threads execute concurrently. For CGRBA, communication only occurs when operations on different PEs have data dependencies. The number of bytes transferred between PEs is related to the algorithm structure, the size of the data to be processed, and the mapping method of the algorithm. However, the width of data transmitted between PEs is determined, and it is equal to the granularity of each PE, that is, the data processing width of a thread. For CGRBA, each sub-block corresponds to a thread, so we set it to the sub-block width of the typical algorithm, which is 32 bits/4 bytes.

For CGRBA, the communication overhead only involves data transmission among different PEs. If using a larger PE, multiple distributed PEs become a centralized single processor. Although communication overhead can be avoided, CGRBA cannot support the concurrent execution of multiple threads. However, a PE with too smaller granularity will lead to a sharp increase in communication overhead, which may even lead to the unsuccessful implementation of an algorithm due to insufficient interconnection resources. There is no communication delay between PEs in the same row, and one clock is consumed between PEs of adjacent rows.

Let $\beta = \sum_{k=1}^{r-1} \beta_k = \frac{\alpha W}{\delta}$ be the total communication traffic of E, k be the row spacing between PEs, β_k be the communication traffic between PEs with the row spacing k, W be the workload of E, α be the ratio of communication traffic to workload, and t_b be the time of communication time among PEs located in different rows. And t_b can be modeled as Eq. (6).

$$t_b = \sum_{k=1}^{r-1} \begin{pmatrix} \beta_k \\ \beta \end{pmatrix} \frac{\alpha W}{\delta} k \tag{6}$$

Compared to the other processors (e.g. multi-core processors and stream processors), the influence of configuration overhead t_c is particularly important for CGRBA. Without optimization schemes, t_c includes two parts: page switching (CRs' configuration changed from a CP to another) and configuration loading (load the configuration to configuration RAM from CM). When the amount of configuration information of E is greater than the capacity of a single CP and less than the sum of the capacity of all CPs, CP switching is required to execute a complete program. Let o be the average execution time (in clock cycles) of a single PE under a certain CP configuration. For program E, the number of CPs required can be expressed as $\lceil (W/\delta \cdot I \cdot o) \rceil$, and the switching times of CPs are $\lceil (W/\delta \cdot I \cdot o) - 1 \rceil$. If the amount of E's configuration information exceeds the sum of the capacities of all CPs, the excessive configuration should be reconfigured from CM. The number of reconfigurations is $\lceil (W/\delta \cdot I \cdot o \cdot a) - 1 \rceil$, where a represents the number of CPs. In summary, t_c is feasible to be modeled as Eq. (7), where $t_{\mathrm{c}1}$ is the switching overhead of CPs and $t_{\mathrm{c}2}$ is the overhead of loading the configuration of a single PE from CM to the configuration RAM.

$$t_c = \left(\left\lceil \frac{W}{\delta \cdot I \cdot o} \right\rceil - 1 \right) t_{c1} + \left\lceil \frac{W}{\delta \cdot I \cdot o \cdot a} - 1 \right\rceil I.\ t_{c2} \tag{7}$$

Based on the above analysis, this paper establishes the CGRBA performance model by extending Amdahl's law, as shown in Eq. (8). Where W, l, α, β, p are the parameters corresponding to the block cipher algorithm. $\delta, c,\ r,\ o,\ a$ are the structural parameters of CGRBA. Because the hardware structure of CGRBA is considered in the modeling, the performance model proposed is more accurate than Amdahl's law.

$$\begin{aligned} T &= T_p + t_b + t_c \\ &= \sum_{i=1}^{m} \sum_{n=1}^{t_m} \frac{w_i^n}{\delta \cdot i} \left(\left\lceil \frac{{}_i^n p_4}{c} \right\rceil \left\lceil \frac{{}_i^n p_2 p_3}{r} \right\rceil \left\lceil \frac{{}_i^n p_1 l}{I} \right\rceil \right) \\ &+ \sum_{k=1}^{r-1} \begin{pmatrix} \beta_k \\ \beta \end{pmatrix} \frac{\alpha W}{\delta} k + \left(\left\lceil \frac{W}{\delta \cdot I \cdot o} \right\rceil - 1 \right) t_{c1} \\ &+ \left\lceil \frac{W}{\delta \cdot I \cdot o \cdot a} - 1 \right\rceil I . t_{c2} \end{aligned} \tag{8}$$

5. Experimental Results and Analysis

In this section, we first simulate the proposed performance model and analyze the simulation results. Then an integer nonlinear programming model towards CGRBA is proposed to solve the optimal parameters of the targeted platform. In addition, we build a hardware platform to verify the reliability of the performance model. And we finally analyze the portability of the proposed methods.

5.1 Model Simulation and Analysis

Through the structural analysis and mapping of a large number of block cipher algorithms, we obtain the appropriate value range of each variable in Eq. (8). Generally, a block cipher algorithm can execute more than one block in parallel, and one block has an even number of sub-blocks. So, this paper set p_1 to 2, 4, or 8, p_2. p_3 is greater than 1, and p_4 is 1 to 3. The maximum value of k is set to 10, which can meet the mapping requirements of the existing block cipher algorithms. And to simplify the simulation process, the value of β_k under different k is the same. In addition, according to the analysis for existing literature [3]–[10], the structure parameters of CGRBA in this paper are set as follows: $\{c = 2z | 1 <= z <= 8, z \in Z_N*\}$, $r \in [4, 20]$, $a \in [1, 5]$, $o = 5$, $t_{c1} = 1$, $t_{c2} = 50$.

1) The influence of communication workload ratio (α)

This paper assuming that the time for program E to be performed serially on a single PE is 1280 ($W/\delta = 1280$) clock cycles, and the workload of different program segments is the same. The degrees of parallelism of 10 program segments are set as follows: $1(p_1 = 1,\ p_2 p_3 = 1,\ p_4 = 1)$, $4(p_1 = 2,\ p_2 p_3 = 1,\ p_4 = 2)$, $4(p_1 = 4,\ p_2 p_3 = 1,$

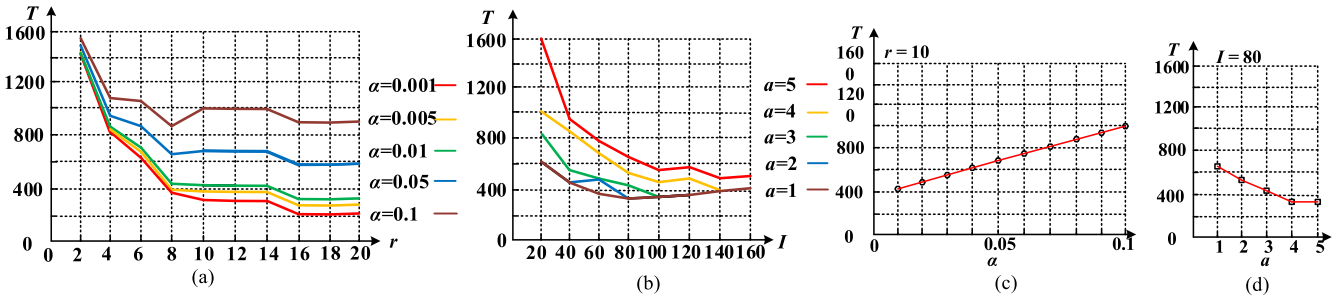

Fig. 5 The trend of execution time of CGRBA with different communication workload ratio (a), CPs' numbers (b), communication workload ratio and the number of PEs rows is 10 (c), CPs' numbers and the number of PEs is 80(d).

$p_4 = 1$), 8($p_1 = 4$, $p_2p_3 = 2$, $p_4 = 1$), 8($p_1 = 1$, $p_2p_3 = 4$, $p_4 = 2$), 16($p_1 = 2$, $p_2p_3 = 8$, $p_4 = 1$), 16($p_1 = 4$, $p_2p_3 = 4$, $p_4 = 1$), 32($p_1 = 4$, $p_2p_3 = 4$, $p_4 = 2$), 32($p_1 = 4$, $p_2p_3 = 8$, $p_4 = 2$), 64($p_1 = 4$, $p_2p_3 = 8$, $p_4 = 2$). During the simulation, the other structural parameters of CGRBA are fixed, where $c = 4$, $l = 32$, $o = 5$, $a = 4$. Figure 5 (a) shows the effect of r on the execution time T under different α.

According to Fig. 5 (a), T decreases with the increase of α, which indicates that the performance of CGRBA is negatively correlated with α. In addition, the difference in the values of T between different curves increases as r rises, which is mainly because k is increased with the increase of r. But when r is greater than the maximal value of k, it will not change and T will decrease as r increases. Therefore, when the number of PEs in CGRBA is small, the impact of communication overhead on algorithm performance should not be ignored. In addition, when r is equal to 10, the function of T concerning α is shown in Fig. 5 (c). It can be concluded that the influence of the communication workload ratio on the acceleration ratio is linear.

2) The influence of the number of CPs (a)

Figure 5 (b) shows the function image of T with respect to I under different a. The value range of a is 1 to 5, and the other parameters remain unchanged. When I is constant, T decreases as a increases. The results show that the performance of CGRBA is positively correlated with the number of CPs. Moreover, T and I are basically negatively correlated. But when I is equal to 140, the values of T are the same in the curves corresponding to $a = 2, 3, 4$, and 5. Besides, when $I = 80$, $T_{(a=4)}$ is equal to $T_{(a=5)}$. And when $I = 100$, $T_{(a=3)} = T_{(a=4)} = T_{(a=5)}$. This is because when I increases to a certain value, the execution time of CGRBA cannot continue to decrease. Therefore, the number of CPs should be set reasonably to improve the resource utilization of CGRBA. In addition, when I is fixed at 80, the function of T about a is shown in Fig 5 (d). It shows that under the condition of limited resources, the influence of the number of CPs on the execution time from linear to no impact.

3) The impact of i and the PEs number

This section studies the relationship among the execution time T, the degree of parallelism i, and the number of PEs I. Figure 6 shows the relationship among the parallelism i, r, and T. The parameters are set as follows. α, a, c, p_1 and p_4 are set to 0.05, 4, 4, 1, and, 1 respectively, $p_2p_3 \in [8, 24]$, $r \in [4, 20]$, $I = 4r \in [16, 80]$. According to the color change, as I and i increase, T gradually decreases to the same value. Further, as shown in the rectangle area 1 in Fig. 6, T is a constant value when $i = 10$ and $I > 40$. It can be seen that when I is larger than i, T remains unchanged as I increases. And the rectangle area 2 in Fig. 6 shows that T remains unchanged when $i > 15$ and $I = 60$. It can be concluded that when i is greater than $I/4$, T remains the same even if i rises. Therefore, if i exceeds the maximal degree of parallelism that can be achieved on CGRBA, the execution time will not decrease as the degree of parallelism increases.

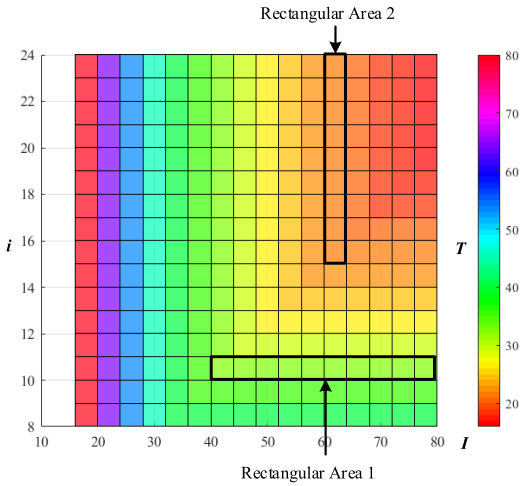

Fig. 6 The influence of algorithm parallelism i and PEs size I on total program execution time T.

5.2 Optimal Parameters Solving

The optimal integer nonlinear programming model is proposed to compute the optimal structural parameters of CGRBA under multiple factors. The objective function is to solve the minimum execution time T as shown in Eq. (9). The constraints of the model include the number of PEs, the degree of parallelism of each level, and the workload of the algorithms. The solution process consists of 4 steps: 1) Determine the values of α, o, and l by analyzing the charac-

Table 2 Experimental results.

$r\times c$	a	Conf-cycles	Total cycles	Model cycles	Accuracy	Speedup
1×4	1	57,344	60,160	59,904	1.004	1.000
2×4	1	57,472	58,880	58,752	1.002	1.022
4×4	1	19,644	20,476	20,284	1.009	2.938
8×4	1	10,636	11,180	10,956	1.020	5.381
10×4	1	1,480	1,745	1,736	1.005	34.476
1×4	2	562	3,122	3,084	1.012	19.270
2×4	2	572	1,980	1,852	1.069	30.384
4×4	2	804	1,636	1,444	1.133	36.773
8×4	2	1,364	1,908	1,684	1.133	31.530
10×4	2	1,480	1,745	1,736	1.005	34.476

Conf-cycles: the number of configuration clock cycles

teristics of block cipher algorithms. 2) Obtain the degree of parallelism of sub-blocks p_1 and let $c = p_1$. 3) Set $r = p_2 p_3$ according to the results in Sect. 4.1. 4) Solve the value of a corresponding to the optimal value of T. Taking $W/\delta = 500$ as an example, set α, o, l, p_4 and p_1 to 0.05, 5, 20, 4, and 2 respectively. The optimal solution of r and a are 24 and 2 respectively, and T reaches the minimum value of 142.1.

$$\begin{aligned}\min(T) = & \sum_{i=1}^{m}\sum_{n=1}^{t_m}\frac{w_i^n}{\delta\cdot i}\left(\left\lceil\frac{{}_i^n p_4}{c}\right\rceil\left\lceil\frac{{}_i^n p_2 p_3}{r}\right\rceil\left\lceil\frac{{}_i^n p_1 l}{I}\right\rceil\right)\\ & + \sum_{k=1}^{r-1}\binom{\beta_k}{\beta}\frac{\alpha W}{\delta}k\\ & + \left(\left\lceil\frac{W}{\delta\cdot I\cdot o}\right\rceil - 1\right)t_{c1}\\ & + \left\lceil\frac{W}{\delta\cdot I\cdot o\cdot a} - 1\right\rceil I.t_{c2}; \quad c,r,o,a\in N^* \end{aligned} \tag{9}$$

Moreover, t_b and t_c can be further reduced by optimizing the algorithm mapping. More specifically, the number of communications between PEs in different rows should be reduced as much as possible, and the number of CPs should be large enough to avoid reconfiguration.

5.3 Hardware Platform Verification

This paper utilizes Verilog HDL language to build a verification platform under 55-nm CMOS technology based on literature [9], [10]. Functional simulation and synthesis are carried out through Synopsys series EDA software under Linux operating system. The maximal working frequency is 110 MHz and the configuration interface width is 32 bit. The platform consists of PEs, ICN, SDM, controller, and CPs. Each PE contains 5 CALUs and two DRs with 32-bit processing granularity. DRs store intermediate data to support pipeline processing. Each PEs row has a CR with a bit width of 148×32 bit, and 4 CPs of the same size form a RAM. We map the AES-128 algorithm (without the key extension algorithm) to the targeted CGRBA. Additionally, the configuration information is obtained through manual mapping, and the number of execution clock cycles is calculated by the EDA simulation software. The experimental results are shown in Table 2.

For the AES-128 algorithm working in the ECB mode, $p_1 = 4$, $p_2 = 4$, $p_3 = 1$, $p_4 = 1$, and W/$\delta = 160$. $r \times c$ represents the number of rows and columns of PE, respectively. The meanings of other variables are the same as in Table 1. If $c < 4$, the platform cannot implement the AES algorithm. This is because the bit width of the shift operation is 128 bits and requires 4 PEs. So we choose c equal to 4. The cycles represent the number of clock cycles required to encrypt 1 KB of data. The verification platform costs 1 and 37 clocks to switch configuration pages and configure a single PE respectively. In Table 2, total cycles are the actual number of execution clock cycles, and the model cycles are the corresponding results derived from the proposed performance model. In the experiment, the control group is the CGRBA structure with the parameters that $c = 4$, $r = 1$, and $a = 1$. The experimental groups are the CGRBA structures with different parameters of c, r, and a shown in Table 2. The speedup for different structures refers to the ratio of the number of clock cycles spent in the experimental group to the number in the control group. Table 2 compares the accuracy of the performance model and speedup under different parameter value settings. And whether this change is consistent with the conclusion in Sect. 5.

According to Table 2, most values of accuracy change from 1.004 to 1.069 except where the values of accuracy are 1.133 in the experimental groups($r = 4$, $c = 4$, $a = 2$, and $r = 8$, $c = 4$, $a = 2$). It means that the difference between the model cycles and the total cycles is mostly within 7%, and the maximum is 13.3%. Because the main reason for the difference is that the proposed model does not consider the data transfer time from off-chip configuration memory to the SDM. However, the modeling of the overhead is relatively simple as it can be directly determined by the amount of data transmitted and the transmission bandwidth. Moreover, the model ignores the filling time before the pipeline reaches the full load. This value is equal to the number of levels in the pipeline minus one. But this part of time accounts for a small proportion of the total execution time and will not increase with the increase of the amount of data, hence, it has little impact on model analysis.

When a is 1, the increase of r will increase the speedup, which proves that the expansion of PEs can increase the degree of parallelism (e.g. p_2). But when a is 2, increasing r from 4 to 8 does not reduce the total number of cycles, because the increase in configuration time exceeds the decrease in computation time, which proves the necessity of considering configuration time when modeling. Keeping r unchanged, when a changes from 1 to 2, the configuration overhead is significantly reduced except $r = 10$. The reason is that when $r = 10$, one CP is sufficient to store the configuration of the AES algorithm. From the Table 2, it can be seen that increasing the number of CPs can significantly reduce the number of configuration clock cycles, although this means that the configuration memory overhead will increase. In addition, the expansion of CPs brings several times of improvement of performance, while the circuit area of CPs is only a part of the circuit area of the entire CGRBA, consequently, the area efficiency will be greatly improved. These conclusions are consistent with the results

shown in the performance model.

5.4 Portability Analysis

The theoretical derivation of this paper is based on CGRBA architecture. However, for CGRA in other fields, such as CGRA for multimedia, various streaming applications, etc., the analysis and modeling methods of multi-level parallelism, communication overhead, and configuration overhead in this paper are also applicable. Moreover, for CGRA towards other types of cryptographic algorithms, only the parallelism part of the performance model needs to be adjusted. For example, for stream cipher algorithms and hash functions, only the parameters p_1 and p_2 need to be modeled again. And for other parallel processing platforms of block cipher algorithms, the multi-level parallel analysis and modeling method in this paper are still applicable. But it is necessary to study the parallel mapping of the algorithm on the target platform. And following the technical route of this paper, the corresponding theoretical derivations and laws are still capable to be derived.

6. Conclusion

This paper first analyzes the particularity of CGRBA, including working method, interconnection structure, and multi-page structure; then this paper studies the multi-level parallelism of block ciphers and its parallel implementation on CGRBA architecture; and we abstract the key parameters that affect the performance of CGRBA; on this basis, this paper proposes a CGRBA performance model based on Amdahl's law, which improves the accuracy of the CGRBA performance model. Moreover, to solve the optimal parameters of CGRBA under multiple factors, this paper constructs an optimal integer nonlinear programming model. This model provides a parameter setting reference for the architecture design of CGRBA. In the experiment, data simulation is firstly carried out. And the related parameters and performance are analyzed qualitatively and quantitatively. The conclusions are as follows. (1) Reducing the communication overhead has a significant impact on improving the algorithm implementation performance on CGRBA. (2) The number of CPs should be set reasonably to increase the utilization rate of CGRBA resources. We also verify the correctness of the data simulation results through hardware platform simulation.

References

[1] L. Bossuet, M. Grand, L. Gaspar, V. Fischer, and G. Gogniat, "Architectures of flexible symmetric key crypto engines–a survey: From hardware coprocessor to multi-crypto-processor system on chip" ACM Comput. Surv., vol.45, no.4, pp.1–32, April 2013. DOI: 10.1145/2501654.2501655.

[2] Z. Zhao, W. Sheng, Q. Wang, W. Yin, P. Ye, J. Li, and Z. Mao, "Towards Higher Performance and Robust Compilation for CGRA Modulo Scheduling" IEEE Trans. Parallel Distrib. Syst., vol.31, no.9, pp.2201–2219, Sept. 2020. DOI: 10.1109/TPDS.2020.2989149.

[3] O. Akbari, M. Kamal, A. Afzali-Kusha, M. Pedram, and M. Shafique, "X-CGRA: An Energy-Efficient Approximate Coarse–Grained Reconfigurable Architecture," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol.39, no.10, pp.2558–2571, Oct. 2020. DOI: 10.1109/TCAD.2019.2937738.

[4] L. Liu, D. Wang, M. Zhu, Y. Wang, S. Yin, P. Cao, J. Yang, and S. Wei, "An energy-efficient coarse-grained reconfigurable processing unit for multiple-standard video decoding," IEEE Trans. Multimedia, vol.17, no.10, pp.1706–1720, Oct. 2020. DOI: 10.1109/TMM.2015.2463735.

[5] S. Gokhan and C. Drek, "Cryptoraptor: High throughput reconfigurable cryptographic processor" 2014 International Conference on Computer Aided Design, San Jose, USA, pp.155–161, Nov. 2014. DOI: 10.1109/ICCAD.2014.7001346.

[6] A.J. Elbirt and C. Paar, "An instruction-level distributed processor for symmetric-key cryptography" IEEE Trans. Parallel Distrib. Syst., vol.16, no.5, pp.468–480, May 2005. DOI: 10.1109/TPDS.2005.51.

[7] B. Wang and L. Liu, "A flexible and energy-efficient reconfigurable architecture for symmetric cipher processing," 2015 IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, Portugal, pp.1182–1185, May 2015. DOI: 10.1109/ISCAS.2015.7168850.

[8] L. Liu, B. Wang, C. Deng, M. Zhu, S. Yin, and S. Wei, "Anole: A Highly Efficient Dynamically Reconfigurable Crypto-Processor for Symmetric-Key Algorithms," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol.37, no.12, pp.3081–3094, Dec. 2018. DOI:10.1109/TCAD.2018.2801229.

[9] Y. Du, W. Li, Z. Dai, and L. Nan, "PVHArray: An Energy-Efficient Reconfigurable Cryptographic Logic Array With Intelligent Mapping," IEEE Trans. Very Lagre Scale Integr. (VLSI) Syst., vol.28, no.5, pp.1302–1315, May 2020. DOI: 10.1109/TVLSI.2020.2972392.

[10] Y. Du, W. Li, and Z. Dai, "PVHArray:A Pipeline Variable Hierarchical Reconfigurable Cryptographic Logic Array Structure," Acta Electronica Sinica, vol.48, no.4, pp.781–789, April 2020.

[11] G.M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," AFIPS Spring Joint Computer Conference, Atlantic City, USA, pp.483–485, April 1967. DOI: 10.1109/N-SSC.2007.4785615.

[12] M.D. Hill and M.R. Marty, "Amdahl's Law in the Multicore Era," Computer, vol.41, no.7, pp.33–38, July. 2008. DOI: 10.1109/MC.2008.209.

[13] H. Che and N. Minh, "Amdahl's law for multithreaded multicore processors," J. Parallel. Distr. Com., vol.74, no.10 pp.3056–3069. DOI: 10.1016/j.jpdc.2014.06.012.

[14] S. Pei, J. Zhang, N. Xiong, et al, "Energy efficiency of heterogeneous multicore system based on the enhanced Amdahl's law," International Journal of High Performance Computing and Networking, vol.12, no.3, pp.236–255, 2016. DOI: 10.1504/IJHPCN.2018.094944.

[15] S.M. Zahedi, Q. Llull, and B.C. Lee, "Amdahl's Law in the Datacenter Era: A Market for Fair Processor Allocation," 2018 IEEE International Symposium on High Performance Computer Architecture, Vienna, Austria, pp.1–14, Feb. 2018. DOI: 10.1109/HPCA.2018.00011.

[16] M.A.N. Al-hayanni, F. Xia, A. Rafiev, A. Romanovsky, R. Shafik, and A. Yakovlev, "Amdahl's law in the context of heterogeneous many-core systems," IET Computers & Digital Techniques, vol.14, no.4, pp.133–148, July 2020. DOI: 10.1049/iet-cdt.2018.5220.

[17] F. Xiao, D. Zibin, L. Wei, C. Luting, "Performance model of multicore crypto processor based on Amdahl's law," Journal of Electronics & Information Technology, vol.38, no.4, pp.827–833, April 2016. DOI: 10.11999/JEIT150474.

[18] S. Wang, G. Li, Y. Yan, et al, "Four dimensions parallel processing architecture for block cipher," Acta Electronic Sinica, vol.45, no.10, pp.2457–2463, Oct. 2017. DOI:

10.3969/j.issn.0372-2112.2017.10.021.

[19] L. Yavits, A. Morad, and R. Ginosart, "The Effect of Communication and Synchronization on Amdahl Law in Multicore Systems" Parallel. Comput., vol.40, no.1, pp.1–14, 2013. DOI: 10.1016/j.parco.2013.11.001.

[20] S. Pei, M.-S. Kim, and J.-L. Gaudiot, "Extending Amdahl's Law for Heterogeneous Multicore Processor with Consideration of the Overhead of Data Preparation," IEEE Embedded Syst. Letters, vol.8, no.1, pp.26–29, 2016. DOI: 10.1109/LES.2016.2519521.

Tongzhou Qu received the B.S. and M.S. degrees in Electrical Engineering from Information Engineering University, Zhengzhou, China, in 2016 and 2019 respectively. He is currently pursuing the Ph.D. degree in the Information Engineering University, Zhengzhou, China. His current research interests include VLSI design of crypto-ICs and cryptographic arithmetic for security applications.

Zibin Dai received the Ph.D. degree from the Information Engineering University, Zhengzhou, China, in 2008. His current research interests include VLSI design of crypto-ICs, energy-efficient SoC platform and cryptographic arithmetic for security applications.

Yanjiang Liu received the B.S. degree from the Zhoukou Normal University, Zhoukou in 2013, and the M.S. degree from the Guangdong University of Technology in 2016, and the Ph.D. degree from the Tianjin University in 2020. He is currently a postdoctoral research fellow with the Information Engineering University. His current research interests include hardware Trojan detection, secure digital circuit design, and EDA for security.

Lin Chen received the Ph.D. degree from the Information Engineering University, Zhengzhou, China, in 2013. Her current research interests include VLSI design of crypto-ICs and cryptographic arithmetic for security applications.

Xianzhao Xia received the Ph.D. degree from the Tianjin University in 2019. He is currently a postdoctoral research jointly with the China Automotive Technology Research Center and Tianjin University. His current research interests include automotive chip design and functional safety detection technology. He is mainly responsible for the design of software/hardware testing equipment for the functional safety and cyber security of automotive chip.