An Efficient Organization of Configuration for CGRAs Applied to Block Ciphers

Liu Xinliang1, Yang Jinjiang2*
1China Center for Information Industry Development, Beijing, China
2Wuxi Research Institute of Applied Technologies Tsinghua University, Wuxi, China
*yangjinjiang@tsinghua-wx.org

Abstract. An efficient organization of configuration is proposed to decrease the size of configuration memory and the time of configuration transmission. Different block ciphers are analysed. And, computing features of algorithms are summarized. Then, the configuration is organized according to the heterogeneous Processing Elements (PEs). It is applied to a coarse-grained reconfigurable architecture (CGRA) which is implemented under TSMC 40nm CMOS technology to compare with similar work. The comparison results show that the configuration capacity decreases with the proposed organization structure. Configuration capacity of the algorithm is reduced by 90.93% to 96.91%.

1. Introduction
The Coarse-Grained Reconfigurable Architecture adopts lots of computing resources and simplifies the interconnection among computing resources, thus CGRA has outstanding area efficiency and energy efficiency in the domain specific area [1][2][3]. However, with the increasing complexity of the application and the performance requirements of the algorithms, the capacity of the configuration becomes larger. The large capacity of configuration not only leads to the area overhead of configuration memory, but also affects the time overhead of reconfiguration transmission. Therefore, it is important to develop a feasible configuration scheduling scheme to improve the area and energy efficiency of CGRA.

The configuration which is used to configure the function of the computing resources in CGRA, is usually stored in the configuration memory. Reducing the area and time overhead used to implement the configuration needs to research the manner of configuration and the organization of configuration memory. To reduce the size of the configuration memory, a variety of configuration memory structures have been proposed. The early CGRA is mainly used to accelerate the application at kernel level, so the simple structure of the single configuration memory [4] can be fine to meet the requirements of performance. Multi-configuration structure [5] have been proposed to improve the configurable efficiency, when the CGRA should support more complicated applications. However, the complexity of the application supported by CGRA is further improved, a great deal of configuration is hard to be stored on the chip, the structure of the configuration cache [6] is proposed, which uses small chip configuration storage resources to obtain fast configurable transmission speed. At the same time, the researchers put forward Hierarchical Configuration Context (HCC) [7], Hierarchical Context Organization (HCO) [8] and other schemes to further reduce the capacity and the time of configuration. HCC scheme is based on the similarity of the kernels in the application, which reduces the capacity of the configuration by storing configuration hierarchically. HCO scheme switches the configuration row
by row during the calculation, and the pipeline technology is used to reduce configuration time. Nevertheless, the similarity of the Processing Elements (PEs) configuration does not receive sufficient attention, which has potential to enhance the configuration efficiency of CGRA targeted at Block ciphers.

This paper proposed an efficient organization of configuration for CGRAs. The main contributions is that the configuration is organized according to the function of heterogeneous PEs to reduce the capacity of configuration. Then, it is applied in a reconfigurable cryptographic processor, which supports several famous block ciphers. The processor is implemented under TSMC 40nm CMOS technology. Compared with other similar work, configuration capacity of the algorithm is reduced by 90.93% to 96.91%, when the proposed organization of configuration is used.

2. Organization scheme of Configuration

2.1. Analysis of block ciphers and overview of reconfigurable architecture

Block ciphers treat a block of data as an operating unit. One block of data usually contains 64 bits or 128 bits. Multiple iterative operations, usually called round function, are used to increase the security of ciphers. The round functions in one cipher is similar or even the same. As is shown in Fig.1, according to the features of the round function, block ciphers can be divided into two broad categories, Feistel structure and SP (substitution & permutation) structure.

![Figure 1. The Feistel and SP Structure.](image)

Although round functions of different ciphers are diverse, their elementary operations are similar. Based on the analysis of more than 100 common used block ciphers, we found that round functions can be induced as four basic operations, which are arithmetic, logic or shift, permutation and substitution box. It should be noticed that few ciphers which take multiplication as nonlinear function instead of S-box, such as IDEA [9], are excluded from the analysis in this paper.

As is shown in Fig.2, the coarse-grained reconfigurable architecture contains two parts which are configuration controller and computing array. The configuration controller, which consists of configuration memory and a scheduling module, is designed for storing and scheduling the configuration of the algorithm. The basic operations discussed above are implemented in the computing array as different Processing Elements (PEs). The PEs are organized by row and rows relate to cross-bar connection. Since the look-up table for S-box leads to much memory cost and it is not necessary to be implemented each row, every four rows share a look-up table for S-box operation. When we are executing the cryptographic algorithm, the configuration controller is used to send the configuration to the computing array step by step. Then the computing array configures the PEs to corresponding functions to perform the operations.
Configuration efficiency is the key factor of the performance of the whole reconfigurable architecture. So, this paper considers the design of the configuration controller. The purpose scheme is to reduce the configuration time and decrease the area cost of configuration memory.

2.2. Organization of Configuration based on Heterogeneous PEs

In most block ciphers, the size of the Data Flow Graphs (DFGs) is larger than the reconfigurable computing array, so they need to be divided into several smaller DFGs, usually called sub-DFGs, to fit the computing array. These sub-DFGs of block ciphers will generate lots of configuration when they are mapping into the computing array. Therefore, an Organization of Configuration based on heterogeneous PEs is proposed in this chapter to reduce the configuration capacity.

The configuration of the computing array contains three parts, which are external data transmission configuration for exchanging data between computing array and outside data memory, internal data transmission configuration for exchanging data between PEs and data memory in the computing array, function configuration for changing operations of PEs in the computing array. Since the number of PEs is large, the capacity of function configuration takes a dominant share of the capacity of total configuration. When an algorithm is divided into sub-DFGs, not all sub-DFGs have external and internal data transmission configuration. In our solutions, configuration is organized as three layers and the external/internal configuration and function configuration are separated. The function configuration is defined as the bottom layer, and the middle layer contains the index of function configuration and the internal data transmission configuration, called sub-DFG configuration. The bottom and middle configuration is stored in the on-chip memories. The top layer configuration, which is generated during the runtime of block ciphers, consists of data transmission configuration and the index of sub-DFG configuration.

Although HCC and HCO also organize the configuration as three layers, they choose the function configuration of the whole array as the minimum granularity of the comparing items when comparing the function configuration between different sub-DFGs. It is a common solution of configuration organization used by the CGRA, such as [10].

However, as is mentioned in section 2.1, the function configuration of each PE in the bottom layer configuration is chosen as the minimal comparing item in our proposed solution, which is a clear difference with the literature [7].

Although HCC and HCO also organize the configuration as three layers, they choose the function configuration of the whole array as the minimum granularity of the comparing items when comparing the function configuration between different sub-DFGs. It is a common solution of configuration organization used by the CGRA, such as [10].

However, as is mentioned in section 2.1, the function configuration of each PE in the bottom layer configuration is chosen as the minimal comparing item in our proposed solution, which is a clear difference with the literature [7].

The details are shown in Fig.3. The main difference between other solutions is that the configuration of heterogeneous PEs is separated for a greater degree of compression of the configuration capacity. Benefitting from the similarity of round function of the block cipher and the likeness of the basic operation in different block ciphers, the capacity of bottom layer configuration is reduced significantly. The proposed organization is the fundamental prerequisite and its advantage is described in the next section.
Figure 3. The hierarchy of the organization of configuration.

3. Simulation Results and Comparison

The proposed organization of configuration is applied in a reconfigurable cryptographic processor, which supports several mainstream block ciphers. Several block ciphers are implemented on the cryptographic processor to compare with other architecture. Table I shows the reduction of configuration capacity when the proposed organization of configuration is used. Configuration capacity of the algorithm is reduced by 90.93% to 96.91%.

| Algorithm | Unorganized (bit) | Organized (bit) | Reduction (%) |
|-----------|------------------|----------------|---------------|
|           | Total            | Top Layer      | Middle Layer  | Bottom Layer  | Total |          |
| AES       | 64383            | 180            | 746           | 4914          | 5840  | 90.929   |
| DES       | 128766           | 180            | 1136          | 7287          | 8603  | 93.318   |
| SM4       | 193149           | 180            | 1526          | 6799          | 8505  | 95.596   |
| Serpent   | 171688           | 180            | 1396          | 6918          | 8494  | 95.052   |
| Twofish   | 128766           | 180            | 1136          | 2669          | 3985  | 96.905   |

When the number of the block ciphers increases, the capacity of the configuration, which needs to be stored on the chip, will become high. As is shown in Fig.4, the normalized configuration capacity of HCC [7] and HCO [8] are increase rapidly, because they treat the configuration of one sub-DFG, which is rarely the same in different algorithms, as a minimal unit in the bottom layer. The minimal unit is the configuration of the heterogeneous PEs in this paper, which is very likely the same since the round function and basic operation in block ciphers are similar. Benefitting from this, the normalized configuration capacity using proposed organization is increasing slowly.

Figure 4. The configuration capacity growth with the number of Algorithms.
4. Conclusion

Based on the analysis of different block ciphers, an efficient organization of configuration is proposed in this paper. The size of configuration memory and the time of configuration transmission are reduced a lot. It is applied to a reconfigurable processor, on which several block ciphers are implemented. Compared with other similar works, configuration capacity of the algorithm is reduced by 90.93% to 96.91%. Thus, it is very suitable for the reconfigurable architecture, which is applied to the implementation of ciphers.

References

[1] R. Hartenstein: “A Decade of Reconfigurable Computing: A Visionary Retrospective,”, Design, Autom. Test Eur. Conf. Exhibit (2001) 642.
[2] F. Campi, A, et al: “A dynamically adaptive DSP for heterogeneous reconfigurable platforms,”, Design, Autom. Test Eur. Conf. Exhibit (2007) 1.
[3] B. Mei, B. Sutter, T. Aa, M. Wouters, A. Kanstein, S. Dupont: “Implementation of a Coarse-Grained Reconfigurable Media Processor for AVC Decoder,”, Journal of Signal Processing Systems 51 (2008) 225.
[4] S. Banerjee, E. Bozorgzadeh, N. Dutt: “Integrating Physical Constraints in HW-SW Partitioning for Architectures with Partial Dynamic Reconfiguration,”, Journal of Signal Processing Systems 51 (2008) 225.
[5] T. Sano, et al: “Instruction buffer mode for multi-context Dynamically Reconfigurable Processors,”, Proc. Int. Conf. Field Program. Logic. Appl. (2008) 215
[6] J. Shield, et al: “Dynamic Cache Switching in Reconfigurable Embedded Systems,”, Proc. Int. Conf. Field Program. Logic Appl. (2009) 111
[7] W. Yansheng, L. Leibo, Y. Shouyi, Z. Min, C. Peng, Y. Jun, W. Shaojun: “ION-Chip Memory Hierarchy in One Coarse-Grained Reconfigurable Architecture to Compress Memory Space and to Reduce Reconfiguration Time and Data-Reference Time,”, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 22 (2014) 983.
[8] W. Bo, L. Leibo: “Dynamically reconfigurable architecture for symmetric ciphers,”, Science China Information Sciences.  59 (2016) 42403.
[9] B. Schneier: “Applied cryptography, second edition: protocols, algorithms, and source code in C,”, Government Information Quarterly. 13 (1997) 336
[10] Y. Shouyi, et al: “Reducing configuration contexts for coarse-grained reconfigurable architecture,”, IEEE Symp. Circuits and System. (2012) 121