Efficient AES cipher on coarse-grained reconfigurable architecture

Chao Wang, Peng Cao, and Jun Yang

National ASIC System Engineering Research Center, Southeast University, Nanjing, China, 210096
a) dragon@seu.edu.cn

Abstract: By exploring the mapping schemes with dataflow graph (DFG) transformation and different granularity of task-level parallelism, we presented various AES implementations on a coarse grained reconfigurable architecture (CGRA) to meet the requirements raging from high performance to low power. In comparison with published AES cipher implementations on programable processors, our AES cipher has 14.7~121.4× higher energy efficiency. Moreover, the design shows the advantage over other CGRAs with 1.3~4.5× energy efficiency improvement.

Keywords: coarse-grained reconfigurable architecture (CGRA), advanced encryption standard (AES), mapping scheme

Classification: Integrated circuits

References

1 Introduction

With the development of information technology, information protection becomes more and more important in daily life. The symmetric block cipher, Rijndael, is standardized by the National Institute of Standards and Technology as the AES (Advanced Encryption Standard) [1], which replaced the original DES (Data Encryption Standard) [2] for better security.

Numerous implementations of AES have been reported to meet different applications’ requirements. Although customized hardware implementations generally offer higher throughput and better energy efficiency [3, 4, 5], these designs are time-consuming and not flexible to upgrade and adapt for future possible protocol changes. Since the Rijndael algorithm is suitable for efficient designs in CPU, by utilizing the specific SSSE3 instructions (Supplemental Streaming SIMD Extensions 3) [6], a bitsliced implementation of AES encryption was presented for Intel Core i7 920 [7] with the speed of 6.92 cycles/byte. Furthermore, the Intel AES-NI instruction set [8] increased the performance of AES encryption to be
surprisingly 0.62 cycles/byte for Intel Core i7-2600K under the frequency of 3.4 GHz [9]. GPU (Graphic Processing Unit) was also chosen as the platform of AES implementation. Manavski [10] implemented AES on GeForce8800 GTX with both OpenCL (Open Computing Language) style and CUDA (Compute Unified Device Architecture) programming model, which achieved the performance of 0.56 cycles/byte. In [11], multi-variant AES cipher was executed in batch by avoiding thread divergence on a GPU device of Tesla K20c, where the speed-up can reach 0.44 cycles/byte when using 4 non-default streams. Besides, an AES engine was implemented on a many core processor array called AsAP [12] by exploring data-level and task-level parallelism [13], which presented various implementations for high performance, throughput per unit of chip area and energy efficiency.

In spite of the above implementation approaches, there is a trend to use CGRA (Coarse Grained Reconfigurable Architecture) to realize the Rijndael algorithm, which can provide satisfying solutions in terms of performance and energy efficiency as well as flexibility. With the intrinsic function set and loop parallelism, AES was mapped on ADRES [14] to process 300 KB data in 3.6 million clock cycles. On the DREAM architecture [15], a maximum performance of 2.39 cycles/byte was achieved for AES with energy efficiency up to 3.03 Mbps/mW owing to its native modular operation support. Cryptoraptor [16] was designed as a high performance, low power, and highly flexible cryptographic processor, which boosts the AES encryption performance by 0.06 cycles/byte.

This paper presents various AES solutions on a CGRA called REMUS-II, which achieve the goal of high performance and low power by mapping with reconstructed and decomposed dataflow graph (DFG) of AES respectively. The reminder of this paper is organized as follows. Section 2 introduces the AES algorithm. Section 3 briefly describes the targeted CGRA platform. In Section 4, different mapping schemes are presented and analyzed. Section 5 discusses the performance and energy efficiency with different granularity of task-level parallelism and compares them with prior arts. Finally, Section 6 concludes the paper.

2 Advanced encryption standard (AES)

AES was originally called Rijndeal, which takes a 128/192/256-bit data block as input and performs several rounds of transforms to generate the cipher block. In this work, we focus on the situation with a 128-bit data block, where the plain text is arranged into a 4-by-4 byte array called as State and executed in 10 rounds.

The DFG of AES is shown in Fig. 1. After the initial step of AddRoundKey, the first 9 rounds consist of four steps including SubBytes, ShiftRows, MixColumns, and AddRoundKey. The final round is different from others by skipping MixColumns. AddRoundKey adds the round key to the State array by exclusive OR operation. SubBytes is a non-linear substitution step executed to each byte of the State over GF ($2^8$), followed by an affine transformation, which is also known as S-Box. In ShiftRows, each row is rotated to the left by a shift step equal to the row index. MixColumns treats each column of the State as the coefficient of a four term polynomial and multiplies with a fixed polynomial over GF ($2^8$).
The targeted platform, REMUS-II [17, 18], is presented as a coarse-grained reconfigurable architecture for computing-intensive tasks with no specific hardware or instructions, which can provide satisfying solutions in terms of both efficiency and flexibility.

As shown in Fig. 2, REMUS-II consists of a reconfigurable processor unit (RPU) and a micro-processor unit (μPU) for speeding up computing-intensive and control-intensive tasks as well as a RISC processor to host both RPU and μPU. The μPU incorporates four micro processors to dynamically manage the reconfiguration process of the RPU via the FIFO write channel. The RPU contains four reconfigurable arrays (RCA), which are composed of 8 × 8 processing elements (PE) coupled with temporary registers (TR) and interconnected by a 2D-mesh network. The PE supports the logical and arithmetic operations with the granularity of 8 bits whereas the TR is used to store intermediate data and offer bypass data path.

During the process of reconfiguration, μPU is activated by the RISC processor firstly to generate a sequence of configuration words for the task kernels of a certain application, before they are fed into RPU in sequence via the FIFO write channel. After that, RPU is instructed by the configuration words to fetch the corresponding configuration contexts. By parsing the configuration context, the PE functions and interconnections within RPU are reconfigured to execute the computation of task kernels. Due to the pipeline fashion of processing in REMUS-II as shown in Fig. 3, the configuration delays in RISC, μPU, the FIFO write channel and RPU are overlapped, which is mainly dominant by the most time-consuming context parsing procedure in RPU.
The reconfiguration procedure impairs the performance due to two reasons. Firstly, the procedure of data processing is decomposed into several steps to compute the intermediate results during configuration switch, so that the throughput is reduced to a fraction of the number of required configuration tasks. Secondly, because of the processing time diversity between configuration and computation, pipeline bubbles might be unavoidable, leading to additional delay between successive tasks. As illustrated in Fig. 3, the computation of task2 is delayed because the configuration of task2 is more time-consuming than the computation of task1.

The REMUS-II architecture has been fabricated in 65 nm CMOS technology, which occupies 21.6 mm² and can work under the frequency of 200 MHz.

4 AES mapping schemes on REMUS-II

In this section, the hardware overhead required for each step in AES is analyzed first and then two mapping schemes of AES on the REMUS-II platform are presented for high performance and low hardware cost respectively.

4.1 Hardware overhead analysis

The hardware overhead of each step of AES is listed in Table I in terms of the number of PEs, where MixColumns is mapped with the data size of a block (128 bits) or a column (32 bits) whereas the other three are mapped with the whole data block. As shown in Table I, only two rows of PE in RCA are needed to implement the transforms of AddRoundKey and SubBytes by exclusive OR operation and lookup table operation. ShiftRows can be realized with the router between PE rows by combining with other steps. As for MixColumns, because of complex modulo computation, it is mapped onto one RCA for one block or two PE rows for one column by decomposing it into multiple logical operations.

<table>
<thead>
<tr>
<th>Step</th>
<th>Data Size</th>
<th># of PE used</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddRoundKey</td>
<td>Block</td>
<td>2 rows (1/4 RCA)</td>
</tr>
<tr>
<td>SubBytes</td>
<td>Block</td>
<td>2 rows (1/4 RCA)</td>
</tr>
<tr>
<td>ShiftRows</td>
<td>Block</td>
<td>0</td>
</tr>
<tr>
<td>MixColumns</td>
<td>Block</td>
<td>8 rows (1 RCA)</td>
</tr>
<tr>
<td>MixColumns</td>
<td>Column</td>
<td>2 rows (1/4 RCA)</td>
</tr>
</tbody>
</table>
4.2 Mapping scheme for high performance

It can be seen from Fig. 1 that in the first 9 rounds, the most complicated step, MixColumns, is carried out in the middle of each round. Hence, when implementing the loop of the first 9 rounds, the DFG of each round should be divided into three sub graphs so as to be mapped onto RCA, which increases the configuration complexity and impairs the performance.

In order to improve the configuration efficiency, the DFG of the round is reconstructed by unrolling the first 9 rounds and combining with the initial AddRoundKey step to form the loop, as illustrated in Fig. 4. In the reconstructed DFG, the loop with 9 iterations are executed in the sequence of AddRoundKey, SubBytes, ShiftRows and MixColumns. Owing to the low PE utilization of the first three steps, they can be mapped within one sub graph, resulting in significant configuration complexity reduction.

4.3 Mapping scheme for low hardware cost

By comparing the required PEs for different steps in Table I, it can be found that although the computation complexity of MixColumns is far higher than other steps, the hardware cost for one column is equivalent with that of others for one block.

In order to map the AES encryption with minimal reconfigurable resources, the DFG of MixColumns can be decomposed into columns owing to data independence so that it can be mapped iteratively for each column onto the same two rows of PEs. Moreover, the occupied reconfigurable resources can be reconfigured for other steps. The decomposed DFG is shown in Fig. 5, where the steps are carried out in sequence with individual configuration contexts, suffering from frequent dynamic reconfiguration and performance degradation.

4.4 Mapping scheme comparison

The configuration overhead of the mapping schemes with reconstructed DFG and decomposed DFG can be compared with the original one in Table II in terms of the total number of reconfigurations required for the whole AES encryption. The ratios...
of PE utilized in RCA and the number of reconfigurations for each sub graph are also listed in Table II with their functions described.

### Table II. Configuration overhead comparison between different mapping schemes

<table>
<thead>
<tr>
<th>Sub Graph No.</th>
<th>Function</th>
<th>RCA Utilization</th>
<th># of Reconf.</th>
<th>Total # of Reconf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>AK</td>
<td>25%</td>
<td>10</td>
<td>29</td>
</tr>
<tr>
<td>2</td>
<td>SB &amp; SR</td>
<td>25%</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MC</td>
<td>100%</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SB &amp; SR &amp; AK</td>
<td>50%</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Recons. DFG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>AK &amp; SB &amp; SR</td>
<td>50%</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MC</td>
<td>100%</td>
<td>9</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>AK &amp; SB &amp; SR &amp; AK</td>
<td>75%</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Decomp. DFG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>AK</td>
<td>25%</td>
<td>11</td>
<td>57</td>
</tr>
<tr>
<td>2</td>
<td>SB &amp; SR</td>
<td>25%</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MC</td>
<td>25%</td>
<td>36</td>
<td></td>
</tr>
</tbody>
</table>

It can be seen that in the mapping scheme with reconstructed DFG, the total number of reconfigurations is reduced dramatically by 34.5% with higher RCA utilization based on one RCA. By mapping with decomposed DFG, only 25% PEs in RCA are required for the implementation of AES with roughly 2× reconfiguration cost.

### 5 Experimental result and comparison

The fabricated chip for REMUS-II was integrated on a verification board and run at the working frequency of 200 MHz with the supply voltage of 1.2 V. When AES is mapped, the execution time for a certain amount of plain text can be monitored with the interrupt signal from REMUS-II which indicates the starting and ending time of encryption whereas the current value during the process of encryption can be recorded via its power supply pin by an Agilent data acquisition unit. Based on chip measurement results, the performance and energy efficiency of the proposed mapping schemes are investigated and compared in this section.

Though the reconstructed DFG and decomposed DFG are targeted on one RCA and 1/4 RCA respectively, they can be implemented in parallel on the REMUS-II architecture to achieve higher performance. The implementation results of the AES encryption with different scale of reconfigurable resources are shown in Fig. 6, where the number of PE used are those in 1/4 RCA, 1/2 RCA, one RCA, two RCAs and four RCAs respectively.

It can be seen that with different mapping scheme and task level parallelism, the REMUS-II platform can meet various requirements raging from high performance and low power consumption. The mapping scheme with decomposed DFG has the advantage of low power when targeting on less than one RCA due to the minimal reconfigurable resource requirement. When multiple RCAs are used, the mapping scheme with reconstructed DFG is superior in energy efficiency with higher
performance and lower power cost. The reason is two fold. Firstly, when occupying the same amount of PEs, the scheme with decomposed DFG suffers from more power consumption due to full utilization of PE and frequent reconfiguration. Secondly, in this situation, the decomposed sub graphs increase the reconfiguration overhead and lead to performance degradation.

Although the reconstructed DFG requires only one third as many configurations as the decomposed DFG, its performance is only double of the decomposed one instead of three times, which is degraded by more pipeline bubbles between sub-graphs. It can be seen in Table II that the complexities of the three sub-graphs in the reconstructed configuration are greatly unbalanced from the aspect of RCA utilization, resulting in different configuration time and computation time. The 2nd sub-graph of MC is mapped with full RCA and 8-stage pipeline whereas the 1st one is configured to execute on half RCA with 4-stage pipeline. During configuration switch, more time is consumed by the configuration of the 2nd sub-graph than the computation of the 1st sub-graph, resulting in the pipeline bubble between data processing of the two sub-graphs. As for the decomposed DFG, all sub-graphs occupy the same amount of reconfigurable resources, avoiding pipeline bubbles during configuration switch.

Since the AES implementations presented in this work are based on a reconfigurable platform without any application-specific hardware, those programmable platforms are chosen for comparison instead of specific hardware. A comprehensive comparison of the state-of-the-art AES implementations is summarized in Table III, where our design was mapped with reconstructed DFG on all RCAs in REMUS-II. In this table, we use the metrics of throughput (Gbps) and throughput per power consumption (Mbps/mW) to compare the performance and energy efficiency of various designs.

It can be seen that compared to other programmable platforms including CPU, GPU and the many core processor, the proposed AES cipher shows a energy efficiency improvement of one to two order of magnitude (14.7~121.4x). Compared with other CGRAs, the implementation on the REMUS-II architecture is
Although some designs boost the throughput at the cost of much more power consumption, this work achieves good balance between performance and energy efficiency.

### 6 Conclusion

This paper proposed two mapping schemes for AES by analyzing the hardware requirement on the CGRA platform. High performance and low power solutions can be obtained by the mapping schemes with reconstructed DFG and decomposed DFG respectively. Compared with other programmable platforms including CPU, GPU, many core processor and CGRA, our design shows the advantage in terms of performance and energy efficiency.

### Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 61404028, 61574033 and 61550110244), Natural Science Foundation of Jiangsu Province (No. BK20161147) and China Scholarship Council.