electronics

MDPI

Article

Low-Delay AES Key Expansion Units Based on DDBT Structure †

Xinxing Zheng [1,2], Han Yan [3,4], Zhiwei Peng [3,4] and Xiaoqiang Zhang [3,4,*]

[1] School of Information and Artificial Intelligence, Wuhu Institute of Technology, Wuhu 241006, China; xingxin2113@whit.edu.cn
[2] Wuhu Engineering Technology Research Center of Vehicle Intelligent Product Development and Application, Wuhu 241006, China
[3] Anhui Engineering Research Center of Vehicle Display Integrated Systems, Anhui Polytechnic University, Wuhu 241000, China; yanhan@stu.ahpu.edu.cn (H.Y.); pengzhiwei@stu.ahpu.edu.cn (Z.P.)
[4] Joint Discipline Key Laboratory of Touch Display Materials and Devices in Anhui Province, Wuhu 241000, China
* Correspondence: zhangxiaoqiang@ahpu.edu.cn
† This paper is an extended version of our paper published in Zhang, X.; Peng, Z.; Yan, H.; Zheng, X.; Xu, M. A Low-Delay Circuit Structure Construction Method for AES Key Expansion Units. In Proceedings of the IEEE 19th Conference on Industrial Electronics and Applications, Kristiansand, Norway, 5–8 August 2024.

Abstract: Advanced Encryption Standard (AES) key expansion unit is usually implemented by chain structure with a long critical path length. That makes key expansion unit become the bottleneck of high-speed AES implementations. In this paper, a design method of low-delay AES key expansion unit is proposed. The proposed design method is based on a delay-drive binary tree (DDBT) structure, which has been proven that it has the shortest critical path length. Based on the proposed design method, a low-delay AES encryption key expansion unit and a low-delay AES encryption/decryption unified key expansion unit are designed in this paper. Both hardware complexity analysis and integrated circuit synthesis indicate that our DDBT-structure-based designs can reduce the delay greatly compared to traditional chain structures. Furthermore, compared to previous works, our designs can achieve the largest throughput.

Keywords: AES; key expansion unit; critical path delay; low-delay circuit structure

check for updates

Academic Editor: Alexander Barkalov

Received: 4 October 2024
Revised: 2 January 2025
Accepted: 17 January 2025
Published: 19 January 2025

Citation: Zheng, X.; Yan, H.; Peng, Z.; Zhang, X. Low-Delay AES Key Expansion Units Based on DDBT Structure. *Electronics* **2025**, *14*, 384. https://doi.org/10.3390/electronics14020384

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Advanced Encryption Standard (AES) is the latest block cipher standard published by the National Institute of Standards and Technology (NIST) in 2001 [1], and it is widely used in various systems of information security now, not only in the resource-constrained applications such as wireless sensor networks area applications [2–4] and radio frequency identification applications [5,6], but also high-throughput applications such as internet security applications [7,8] and high-speed storage applications [9]. In resource-constrained applications, hardware implementations prioritize the area and power consumption, whereas in high-throughput applications, the focus is on minimizing delay.

Key expansion is a crucial operation in the AES cryptographic algorithm [10,11], which involves adding the round key and the state matrix. To reduce area cost in hardware implementations, the key expansion unit is generally implemented by the chain structure. In this structure, the latter output signal reuses the operation resources of the previous output signal to reduce the area cost in hardware implementations. But the critical path of the chain structure is longer i.e., the delay of the circuit based on the chain structure is

larger. Therefore, the key expansion unit often becomes a bottleneck in high-speed AES implementations [12].

In this paper, a method to construct a low-delay AES key expansion unit structure is proposed. In the proposed method, the expressions of the key expansion operation are derived at first. Subsequently, a delay-aware common subexpressions elimination (DACSE) algorithm is used to extract the common subexpressions (CS) in the expressions. Last, the low-delay key expansion unit is constructed by delay-drive binary tree (DDBT) structure.

It has been proven that DDBT structure has the shortest critical path for the linear operations [13]. But the area cost is larger when the circuit is constructed directly by the expressions of linear operations. By extracting the CS from the expressions before construction, the circuit area can be reduced effectively, while the critical path will be increased [14]. To keep the shortest critical path unchanged in the DDBT structure, the DACSE algorithm is used to extract the CS from expressions of key expansion in this paper, as the DACSE algorithm can extract the CS under a delay constraint [15]. Altogether, the key expansion unit constructed by the proposed method will achieve less delay and less area cost.

A low-delay AES encryption key expansion unit and a low-delay AES encryption/decryption unified key expansion unit are constructed by the proposed method. And the units are synthesized by Synopsys (Sunnyvale, CA, USA) integrated circuit (IC) synthesis tool Design Compiler (DC) Tool with SMIC 0.18 μm technology and AMD (Santa Clara, CA, USA) FPGA synthesis tool Vivado 2019.2 with xc7vx485T.. The synthesized results show that, compared to traditional chain structures, the key expansion units constructed by the proposed method have less delay. Compared with previous works, our key expansion designs achieve the maximum throughput.

2. AES Key Expansion Unit Based on Chain Structure

AES processes data blocks of 128-bits, which can be regarded as 4×4 bytes state matrices, and the entire AES encryption/decryption operation is completed through *Nr* rounds of transformation operations, where *Nr* represents the number of round operations and it depends on the initial key size [1]. The initial key length can be 128 bits, 192 bits, and 256 bits, and the corresponding AES cryptographic algorithms can be denoted as AES-128, AES-192, and AES-256 respectively [1]. We only take AES-128 as an example to illustrate the construction method of low-delay key expansion unit structure, and the construction method can be easily extended to AES-192 and AES-256 key algorithms.

In the encryption process of AES, there are four operations in a round transformation, i.e., *SubBytes*, *ShiftRows*, *MixColumns*, and *AddRoundKey*, except for the last round, there is no *MixColumns* operation. The decryption process of AES performs the reverse data flow of the encryption process, and the round transformation in the decryption process performs four inverse operation of encryption process, i.e., *InvSubBytes*, *InvShiftRows*, *InvMixColumns*, and *AddRoundKey*, except for the first round, which has no *InvMixColumns* operation.

Among these operations, *AddRoundKey* uses the XOR operation to complete the addition operation between the state matrix and the round key, and the round key is generated by the key expansion unit. The key expansion unit uses the initial key to generate the round key for each round transformation according to the key extension algorithm in [1].

2.1. AES Encryption Key Expansion Unit

In the process of AES-128 encryption operation, the 128-bit subkeys used in each round is obtained from the previous round of subkeys according to the key expansion algorithm.

In AES-128, the subkey of each round can be expressed as a 4×4 bytes matrix. The subkey matrix is shown below [1].

$$\begin{bmatrix} k^i_{0,0} & k^i_{0,1} & k^i_{0,2} & k^i_{0,3} \\ k^i_{1,0} & k^i_{1,1} & k^i_{1,2} & k^i_{1,3} \\ k^i_{2,0} & k^i_{2,1} & k^i_{2,2} & k^i_{2,3} \\ k^i_{3,0} & k^i_{3,1} & k^i_{3,2} & k^i_{3,3} \end{bmatrix} \Rightarrow \begin{bmatrix} K^i_0 & K^i_1 & K^i_2 & K^i_3 \end{bmatrix} \tag{1}$$

where $k^i_{m,n}$ is the subkey byte of the nth column mth row of the ith round subkey matrix, K^i_n is the subkey word of the nth column of the ith round subkey matrix, and the word vector K^i_n is a 32-bit word.

According to the key expansion algorithm in [1], the expansion operation of the ith round subkey can be expressed as follows [1].

$$\begin{cases} K^i_0 = S_W + R_c + K^{i-1}_0 \\ K^i_1 = K^i_0 + K^{i-1}_1 \\ K^i_2 = K^i_1 + K^{i-1}_2 \\ K^i_3 = K^i_2 + K^{i-1}_3 \end{cases} \tag{2}$$

where $S_W = SubWord\left(RotWord\left(K^{i-1}_3\right)\right)$, $R_c = Rcon(i)$, and $Rcon(i)$ is a constant vector shown below [1].

$$Rcon(i) = \begin{bmatrix} \{02\}^{i-1} \\ \{00\} \\ \{00\} \\ \{00\} \end{bmatrix} \tag{3}$$

where {00} and {02} are the elements in Galois field $GF(2^8)$, *RotWord* is a word shift operation, *SubWord* is a word substitute operation which is equivalent to four *SubBytes* operations defined in [1], and the addition operation is defined as 32-bit exclusive OR (XOR).

In the existing literatures [2–6,10,11], the AES encryption key expansion unit is generally implemented in a chain structure according to Equation (2), as shown in Figure 1.

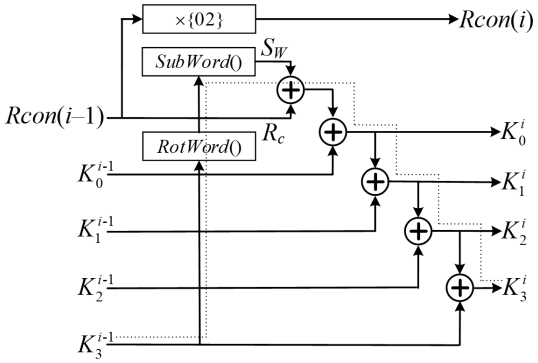

Figure 1. AES encryption key expansion unit based on the chain structure.

In the chain structure, the latter output signal reuses the operation resources of the previous output signal to reduce the area cost in hardware implementations. However, the critical path of the chain structure is also increased due to the sharing of operation resources. The critical path of the chain structure is shown by the dotted line in Figure 1. We mainly focus on the design method of the low-delay structure, so only logic gate delay is considered in this paper. On the critical path, *RotWord* operation can just be implemented by switching bus sequence, therefore, no logic resources are required in *RotWord* unit, then the critical path length of *RotWord* unit is 0. The critical path length of *SubWord* unit is

equivalent to the critical path length of *SubBytes*. As shown in Figure 1, on the critical path, there are 5 adders after *SubWord*. Therefore, the critical path of the AES key expansion unit based on the chain structure is $T_{KE} = T_{SB} + 5T_{ADD}$, where T_{KE}, T_{SB}, and T_{ADD} denote critical path of AES key expansion unit, *SubBytes* and adder, repectively. The area cost of the AES key expansion unit based on the chain structure is $A_{KE} = 4A_{SB} + 5A_{ADD} + A_{CST}$, where A_{KE}, A_{SB}, A_{ADD}, and A_{CST} denote the area cost of the key expansion unit, *SubBytes* unit, adder, and constant multiplier unit, respectively.

In this paper, the delay and area cost of designed circuit are evaluated by logic gate delay and area cost only. The implementation of *SubBytes* proposed in [16] is used to verify the construction method proposed in this paper. According to [16], the critical path length of *SubBytes* is $T_{SB} = 18T_{XOR} + 3T_{AND}$ and $A_{SB} = 89A_{XOR} + 35A_{AND}$, where T_{XOR} and A_{XOR} denote the delay and the area cost of XOR gate, respectively, and T_{AND} and A_{AND} denote the delay and the area cost of AND gate, respectively.

In the Galois field, the adders are implemented by bit-XOR operations, therefore, we can use the T_{XOR} and A_{XOR} to evaluate the critical path length and area cost of adders. There is only one XOR gate on the critical path of adder, then $T_{ADD} = 1T_{XOR}$. As mentioned above, the word vector K_n^i is a 32-bit word, therefore, the adders in Figure 1 are implemented by 32-bit-XOR operations. Except for add R_c, which only requires an 8-bit-XOR operation in the adder. Therefore, a total of 136 XOR gates are required for the adders. The constant multiplier $\times\{02\}$ unit in Figure 1 requires 3 XOR gates. For the AES key expansion unit based on chain structure, the critical path delay is $T_{KE} = T_{SB} + 5T_{ADD} = 23T_{XOR} + 3T_{AND}$, and area cost is $A_{KE} = 4A_{SB} + 5A_{ADD} + A_{CST} = 495A_{XOR} + 140A_{AND}$.

The logic gates in SMIC 0.18 μm technology are used to more accurately evaluate the delay and area cost of the designs in this paper, the parameters of related logic gates are shown in Appendix A. According to Appendix A, the delay and area cost of AES key expansion unit based on the chain structure are 5.31 ns and 4520 Trs, respectively, where Trs is an abbreviation for transistors in this paper.

2.2. AES Decryption Key Expansion Unit

The key expansion operation in decryption is the inverse operation of the encryption key expansion operation. The decryption key expansion can be expressed as follows [1].

$$\begin{cases} K_0^i = S'_W + R'_c + K_0^{i-1} \\ K_1^i = K_0^{i-1} + K_1^{i-1} \\ K_2^i = K_1^{i-1} + K_2^{i-1} \\ K_3^i = K_2^{i-1} + K_3^{i-1} \end{cases} \tag{4}$$

where $S'_W = SubWord\left(RotWord\left(K_2^{i-1} + K_3^{i-1}\right)\right)$, $R'_c = Rcon^{-1}(i)$, $Rcon^{-1}(i)$ is a constant vector shown below [1].

$$Rcon^{-1}(i) = \begin{bmatrix} \{36\} \times \{8D\}^i \\ \{00\} \\ \{00\} \\ \{00\} \end{bmatrix} \tag{5}$$

where $\{36\}$ and $\{8D\}$ are the elements in Galois field $GF(2^8)$. The structure of decryption key expansion unit is shown in Figure 2 [1].

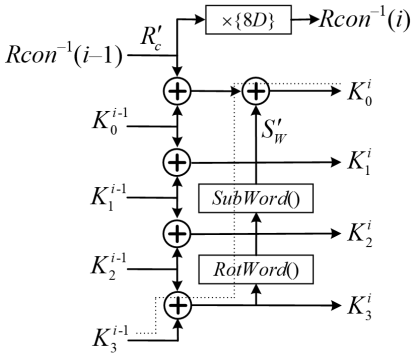

Figure 2. AES decryption key expansion unit.

The whole decryption key expansion unit only requires 5 adders (which include four 32-bit width adders and one 8-bit width adder). The adders and constant multiplier ×{8D} unit in Figure 2 also require 136 XOR gates and 3 XOR gates, respectively. The critical path of the decryption key expansion unit is shown by the dotted line in Figure 2. On the critical path, there is one adder before and after *SubWord*, respectively. Therefore, the critical path of the decryption key expansion unit is $T_{KE} = T_{SB} + 2T_{ADD} = 20T_{XOR} + 3T_{AND}$. The area cost of the decryption key expansion unit is $A_{KE} = 4A_{SB} + 5A_{ADD} + A_{CST} = 495A_{XOR} + 140A_{AND}$. When converted to SMIC 0.18 μm technology, the delay and area cost of the decryption key expansion are 4.68 ns and 4520 Trs, respectively. The decryption key expansion unit shown in Figure 2 is the optimal circuit structure, that is, both area and critical path of the unit achieve the minimum. Therefore, the decryption key expansion unit has no space to be further optimized.

2.3. AES Encryption/Decryption Unified Key Expansion Unit

In many applications, encryption operation and decryption operation are often necessary to integrate into the same hardware. In this case, an AES encryption/decryption unified key expansion unit can be constructed to reduce the area cost, as the same operation units between the AES encryption key expansion operation and decryption key expansion operation are reused in the unified unit [2,11].

In the conventional AES encryption/decryption unified key expansion unit, the chain structure is also employed to reduce the area cost. The AES encryption/decryption unified key expansion unit based on the chain structure is shown in Figure 3.

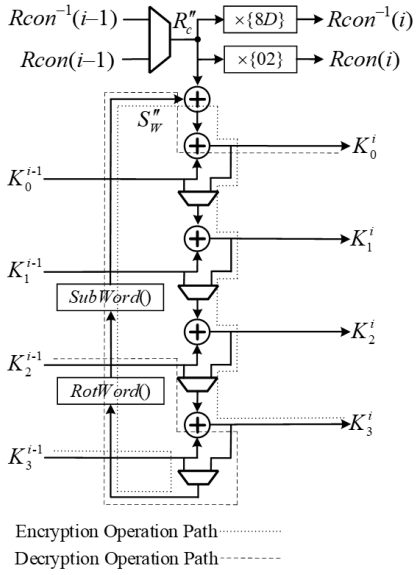

Figure 3. AES encryption/decryption unified key expansion unit based on the chain structure.

In addition to *SubWord* unit, the AES encryption/decryption unified key expansion unit requires a total of 5 adders (which include four 32-bit width adders and one 8-bit width adder), 5 two-to-one bus-multiplexers (which include four 32-bit width bus-multiplexers and one 8-bit width bus-multiplexer), and 2 constant multipliers (both contain 3 XOR gates). The unified unit can operate at two states: encryption key expansion state and decryption key expansion state. The critical path of the unified unit in two operation states is shown in Figure 3. In the encryption key expansion state, the critical path of the unified unit is $T_{KE} = T_{SB} + 5T_{ADD} + 4T_{BMX}$, where T_{BMX} is the critical path of a bus-multiplexer. In the decryption key expansion state, the critical path of the unified unit is $T_{KE} = T_{SB} + 2T_{ADD} + 2T_{BMX}$. Therefore, the critical path of the entire unified unit is $T_{KE} = T_{SB} + 5T_{ADD} + 4T_{BMX}$, where T_{MUX} is the critical path of a two-to-one multiplexer. The area cost of the unified unit is $A_{KE} = 4A_{SB} + 5A_{ADD} + 2A_{con} + 5A_{BMX} = 498A_{XOR} + 140A_{AND} + 136A_{MUX}$, where A_{BMX} and A_{MUX} are the area cost of the bus-multiplexer and two-to-one multiplexer, respectively. When converted to SMIC 0.18 μm technology, the delay and area are 6.07 ns and 5632 Trs, respectively.

According to Figure 3, the encryption/decryption unified key expansion operation can be expressed as follows.

$$\begin{cases} K_0^i = S_W'' + R_c'' + K_0^{i-1} \\ K_1^i = K_0^i / K_0^{i-1} + K_1^{i-1} \\ K_2^i = K_1^i / K_1^{i-1} + K_2^{i-1} \\ K_3^i = K_2^i / K_2^{i-1} + K_3^{i-1} \end{cases} \tag{6}$$

where $S_W'' = SubWord\left(RotWord\left(K_3^{i-1} / \left(K_2^{i-1} + K_3^{i-1}\right)\right)\right)$, $R_c'' = Rcon(i)/Rcon^{-1}(i)$, and '/' denotes two-to-one bus-multiplex operation.

3. The Proposed Low-Delay Structure Construction Method

Although the key expansion unit based on the chain structure has a smaller area, the critical path is much longer, which will lead to greater circuit delay. To solve this problem, a method to construct a low-delay key expansion unit is proposed in this paper. In this paper, the study presented in [17] is expanded upon. The specific construction method is as follows:

Step 1: Derive the expressions of AES key expansion operation.

Step 2: Extract CS from the expressions by the DACSE algorithm.

Step 3: Construct the key expansion unit based on the DDBT structure.

According to the above construction method, low-delay AES encryption key expansion unit and low-delay AES encryption/decryption unified key expansion unit are further constructed.

3.1. The Construction Method of Low-Delay AES Encryption Key Expansion Unit

3.1.1. Deriving the Expressions

According to Equation (2), we can obtain the complete form of encryption expansion operation expressions as follows.

$$\begin{cases} K_0^i = S_W + R_c + K_0^{i-1} \\ K_1^i = S_W + R_c + K_0^{i-1} + K_1^{i-1} \\ K_2^i = S_W + R_c + K_0^{i-1} + K_1^{i-1} + K_2^{i-1} \\ K_3^i = S_W + R_c + K_0^{i-1} + K_1^{i-1} + K_2^{i-1} + K_3^{i-1} \end{cases} \tag{7}$$

The expressions in Equation (7) are the linear operation expressions. It has been proven in [13] that the linear operation unit can achieve the shortest critical path if it was

constructed by the DDBT structure. Before constructing the DDBT structure, it is necessary to obtain the delay of all input signals. Supposed that the delay before AES encryption key expansion unit can be ignored, i.e., the delays of input signals $\left[R_c,\ K_0^{i-1},\ K_1^{i-1},\ K_2^{i-1},\ K_3^{i-1} \right]$ are 0. There is a *Subword* operation before the input signal S_W. As previously mentioned, the delay of the input signal S_W is $18T_{\mathrm{XOR}} + 3T_{\mathrm{AND}}$, and only the delay of XOR gates is taken into consideration in the construction process. The construction method can be found in Figure 4 in [13]. For ease of understanding, the construction method of the DDBT structure is redescribed as follows.

We take the circuit of output K_3^i as an example to illustrate the construction method of the DDBT structure. The input signal set of K_3^i is $\left\{ S_W, R_c, K_0^{i-1}, K_1^{i-1}, K_2^{i-1}, K_3^{i-1} \right\}$. Firstly, the input signals are sorted in ascending order of delay, and the sorted set is shown in the first row of Table 1. In the first iteration, R_c and K_0^{i-1}, which are the two signals with the smallest delay, are extracted from the set to construct the circuit $D_0 = R_c + K_0^{i-1}$. According to the delay calculation method in the 7th line of Algorithm 1, the delay of signal D_0 is $1T_{\mathrm{XOR}}$.

Algorithm 1. The Construction Method of the DDBT Structure.

```
for j = 1 to N_out
    All input signals form a set S
    for i = 1 to N_in − 1
        Choose the two signals s_1 and s_2 with the least delay from S
        Construct an addition circuit s_new = s_1 + s_2
        Remove s_1 and s_2 from S
        Compute the delay of s_new with t_new = max(t_1, t_2) + 1
        Add s_new to the set S
    end
end
```

where N_{in} and N_{out} denote the number of input signals and output signals, respectively, and t_{new}, t_1, and t_2 denote the delay of s_{new}, s_1, and s_2, respectively. The construction method always chooses the two signals with the least delay to construct the circuit in each iteration of the construction process.

Table 1. The DDBT structure construction process of signal K_3^i.

No.	Input Signal Set	Constructed Circuit
1	$\left\{ R_c(@0),\ K_0^{i-1}(@0),\ K_1^{i-1}(@0),\ K_2^{i-1}(@0),\ K_3^{i-1}(@0),\ S_W(@18) \right\}$	$D_0(@1) = R_c(@0) + K_0^{i-1}(@0)$
2	$\left\{ K_1^{i-1}(@0), K_2^{i-1}(@0), K_3^{i-1}(@0), D_0(@1), S_W(@18) \right\}$	$D_1(@1) = K_1^{i-1}(@0) + K_2^{i-1}(@0)$
3	$\left\{ K_3^{i-1}(@0), D_0(@1), D_1(@1), S_W(@18) \right\}$	$D_2(@2) = K_3^{i-1}(@0) + D_0(@1)$
4	$\{ D_1(@1), D_2(@2), S_W(@18) \}$	$D_3(@3) = D_1(@1) + D_2(@2)$
5	$\{ D_3(@3), S_W(@18) \}$	$K_3^i(@19) = D_3(@3) + S_W(@18)$

Repeat the above process at subsequent iterations until all input signals are used to construct the circuit. The process of the DDBT structure is shown in Table 1. After $N - 1$ iterations, where N is the number of input signals, the construction process of the DDBT structure was completed, and the DDBT structure of K_3^i is shown in Figure 4.

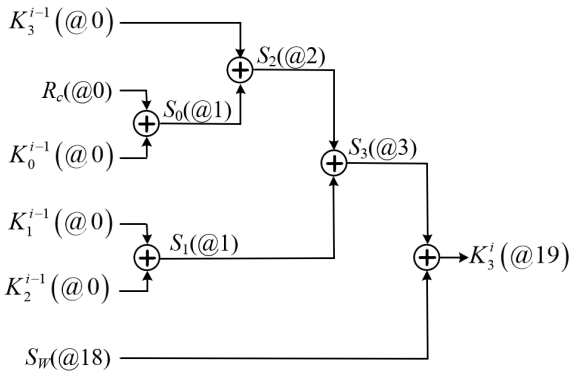

Figure 4. The DDBT structure of input signal K_3^i.

The circuit of other three output signals are also constructed in the same way. After construction, the AES encryption key expansion unit based on the DDBT structure can be constructed, as shown in Figure 5.

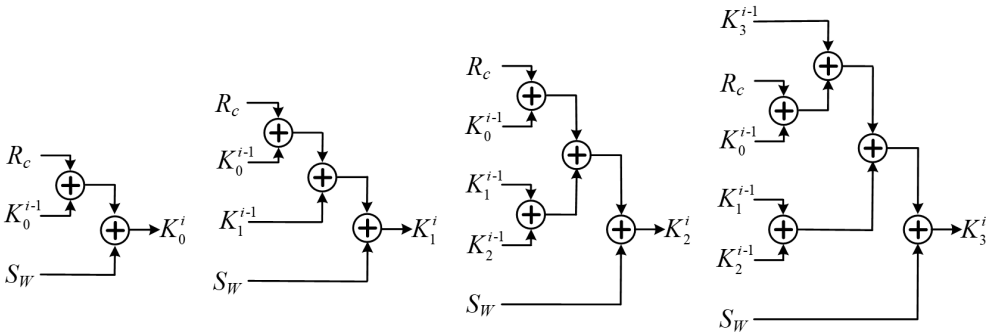

Figure 5. The AES encryption key expansion unit based on the DDBT structure.

The critical path length of the DDBT structure can be calculated as follows [13–15]:

$$T_{\mathrm{DDBT}} = \max_{j=0,\cdots,n-1}(T_j) = \max_{j=0,\cdots,n-1}\left(\left\lceil \log_2 \sum_{N_j} 2^{t_{k,j}} \right\rceil\right) \tag{8}$$

where T_j is the delay of jth output signal, $t_{k,j}$ is the delay of kth input signal of jth output signal, N_j is the number of input signals of jth output signal. Note that T_j and $t_{k,j}$ are calculated by the number of logic gates on the path. According to Equation (8), we can determine the critical path length of AES encryption key expansion unit based on the DDBT structure is $T_{\mathrm{KE}} = T_{\mathrm{SB}} + 1T_{\mathrm{ADD}} = 19T_{\mathrm{XOR}} + 3T_{\mathrm{AND}}$.When converted to SMIC 0.18 μm technology, the delay is 4.47 ns. In the DDBT structure, there is only one addition operation after the input signal S_W, the critical path is reduced by $4T_{\mathrm{XOR}}$ compared to the chain structure. When converted to SMIC 0.18 μm technology, the delay is reduced by 0.84 ns, the reduction is up to 15.82%.

A total of 14 adders (which include ten 32-bit width adders and four 8-bit width adders) are required in the DDBT structure in Figure 5, which is much more than that in the chain structure, as there is no shared unit among the operations of the output signal. The area cost of the DDBT structure in Figure 5 is $A_{\mathrm{KE}} = 4A_{\mathrm{SB}} + 14A_{\mathrm{ADD}} + A_{\mathrm{CST}} = 711A_{\mathrm{XOR}} + 140A_{\mathrm{AND}}$. When converted to SMIC 0.18 μm technology, the area cost is 6248 Trs, the area cost increases by 1728 Trs, the increase is up to 38.23%.

The number of adders in the DDBT structure in Figure 5 is much more than that in the chain structure, a total of 14 adders are required, as there is no shared unit among the operations of output signal. The number of adders can be reduced by sharing CS in the operation expressions, and common subexpressions elimination (CSE) algorithms are the

effective algorithms to extract CS. But it has been proven that sharing CS may increase critical path length in the hardware implementations [14]. The DACSE algorithm proposed in [15] can extract CS while keeping the shortest critical path unchanged in the DDBT structure, as it can extract CS under a delay constrain. In this paper, the CS in expressions of the AES encryption key expansion operation is further extracted by the DACSE algorithm.

3.1.2. Extracting the CS

The DACSE algorithm can be found in [15], we rewrite it as follows for ease of understanding.

As shown in Algorithm 2, the DACSE algorithm calculates the T_{KE} at each iteration of CS extraction. If the extraction of the CS causes T_{KE} to exceed the delay constraint T_{con}, the extraction of the CS is abandoned. If the extraction of the CS keeps T_{KE} unchanged, the extraction of the CS is valid. The optimization process of the AES encryption key expansion unit by Algorithm 2 is shown as follows:

Algorithm 2. The DACSE Algorithm.

Input the operation expressions $\boldsymbol{E}$ and delay constraint T_{con}
Create two empty sets $\boldsymbol{S}_{CS}$ and $\boldsymbol{S}_{IV}$
Count the occurrence frequency of each CS in $\boldsymbol{E}$
Find out the max occurrence frequency O_{max}
while $O_{max} > 1$
    Create new operation expressions $\boldsymbol{E}_{\boldsymbol{CS}} = \boldsymbol{E}$
    Select a CS $(s_1 + s_2)$ with occurrence frequency O_{max} randomly
    Replace the CS $(s_1 + s_2)$ in $\boldsymbol{E}_{\boldsymbol{CS}}$ with a new signal s_{com}
    Compute the delay of s_{com} with $t_{com} = \max(t_1, t_2) + 1$
    Compute the whole operation unit delay T_{KE} with Equation (8)
    if $T_{KE} = T_{con}$
        Add the eliminated CS $s_{com} = s_1 + s_2$ to set $\boldsymbol{S}_{CS}$
        Replace $\boldsymbol{E}$ with $\boldsymbol{E}_{\boldsymbol{CS}}$, that is $\boldsymbol{E} = \boldsymbol{E}_{\boldsymbol{CS}}$
    else
        Add the CS $s_1 + s_2$ to set $\boldsymbol{S}_{IV}$
    end if
    Count the occurrence frequency of each CS in $\boldsymbol{E}$ other than the CS in set $\boldsymbol{S}_{IV}$
    Find out the max occurrence frequency O_{max}
end while

where t_{com}, t_1, and t_2 denote the delay of s_{com}, s_1, and s_2, respectively. The DACSE algorithm requires the input of a delay constraint T_{con} at first. According to the above, the critical path length of the AES encryption key expansion unit based on the DDBT structure is $T_{KE} = 19T_{XOR} + 3T_{AND}$, only T_{XOR} is considered in the optimization process. Therefore, when the expressions of the AES encryption key expansion operation are optimized by DACSE, the delay constraint is set as $T_{con} = 19$. With this constraint, the shortest critical path length can remain unchanged during the extraction of CS.

At the first iteration of Algorithm 2, the occurrence frequency of each CS in Equation (7) is counted. According to the counting results, CS $(S_W + R_c)$, $(S_W + K_0^{i-1})$, and $(R_c + K_0^{i-1})$ have the max occurrence frequency, they all occur 4 times. A CS with max occurrence frequency is selected randomly, supposed that CS $(S_W + R_c)$ is selected to be eliminated.

Supposed that $C_0 = S_W + R_c$, according to line 9 in Algorithm 2, the delay of C_0 is $T_{C0} = 19T_{\mathrm{XOR}}$. The CS ($S_W + R_c$) in Equation (7) is replaced by C_0 as shown in the following expressions.

$$\begin{cases} K_0^i = C_0 + K_0^{i-1} \\ K_1^i = C_0 + K_0^{i-1} + K_1^{i-1} \\ K_2^i = C_0 + K_0^{i-1} + K_1^{i-1} + K_2^{i-1} \\ K_3^i = C_0 + K_0^{i-1} + K_1^{i-1} + K_2^{i-1} + K_3^{i-1} \end{cases} \tag{9}$$

The delay of output signals can be calculated by Equation (8), the delay of all output signals is $20T_{\mathrm{XOR}}$, which has exceeded the delay constraint $T_{\mathrm{con}} = 19$. Therefore, the elimination of CS ($S_W + R_c$) is abandoned, and CS ($S_W + R_c$) is added into a set $\boldsymbol{S}_{\mathrm{IV}}$, which contains the checked CSs. However, selecting CS ($S_W + K_0^{i-1}$) to eliminate will also increase the critical path delay. Supposed that CS ($R_c + K_0^{i-1}$) is selected to eliminate, and supposed that $C_0 = R_c + K_0^{i-1}$, the delay of C_0 is $T_{C0} = 1T_{\mathrm{XOR}}$. The CS ($R_c + K_0^{i-1}$) in Equation (7) is replaced by C_0 as shown in the following expressions.

$$\begin{cases} K_0^i = S_W + C_0 \\ K_1^i = S_W + C_0 + K_1^{i-1} \\ K_2^i = S_W + C_0 + K_1^{i-1} + K_2^{i-1} \\ K_3^i = S_W + C_0 + K_1^{i-1} + K_2^{i-1} + K_3^{i-1} \end{cases} \tag{10}$$

After the elimination of CS ($R_c + K_0^{i-1}$), the delays of all output signals are $19T_{\mathrm{XOR}}$, which equal to the delay constraint $T_{\mathrm{con}} = 19$. Therefore, the CS ($R_c + K_0^{i-1}$) is eliminated at the first iteration.

At the second iteration, the occurrence frequency of each CS in Equation (10) is counted. The CS $S_W + C_0$ has max occurrence frequency, it occurs 4 times. However, the elimination of CS $S_W + C_0$ will increase the critical path delay, thus, the CS $S_W + C_0$ will be added into set $\boldsymbol{S}_{\mathrm{IV}}$. Then CS ($C_0 + K_1^{i-1}$) is selected to be eliminated as it occurs 3 times, which has max occurrence frequence except for CS $S_W + C_0$. Supposed that $C_1 = C_0 + K_1^{i-1}$, the delay of C_1 is $T_{C1} = 2T_{\mathrm{XOR}}$. The CS ($C_0 + K_1^{i-1}$) in Equation (10) is replaced by C_1 as shown in the following expressions.

$$\begin{cases} K_0^i = S_W + C_0 \\ K_1^i = S_W + C_1 \\ K_2^i = S_W + C_1 + K_2^{i-1} \\ K_3^i = S_W + C_1 + K_2^{i-1} + K_3^{i-1} \end{cases} \tag{11}$$

After the elimination of CS ($C_0 + K_1^{i-1}$), the delays of all output signals keep $19T_{\mathrm{XOR}}$ unchanged.

At the third iteration, the elimination of CS $S_W + C_1$ with max occurrence frequency will also increase the critical path delay. Then, CS ($C_1 + K_2^{i-1}$) is selected to be eliminated. Supposed that $C_2 = C_1 + K_2^{i-1}$, the delay of C_2 is $T_{C2} = 3T_{\mathrm{XOR}}$. The CS ($C_1 + K_2^{i-1}$) in Equation (11) is replaced by C_2 as shown in the following expressions.

$$\begin{cases} K_0^i = S_W + C_0 \\ K_1^i = S_W + C_1 \\ K_2^i = S_W + C_2 \\ K_3^i = S_W + C_2 + K_3^{i-1} \end{cases} \tag{12}$$

After the elimination of CS ($C_1 + K_2^{i-1}$), the delays of all output signals still keep $19T_{\mathrm{XOR}}$ unchanged. There is no CS in Equation (12), therefore, the optimization process of

Algorithm 2 is terminated. After the optimization of Algorithm 2, the extracted CSs are shown as follows.

$$\begin{cases} C_0 = R_c + K_0^{i-1} \\ C_1 = C_0 + K_1^{i-1} \\ C_2 = C_1 + K_2^{i-1} \end{cases} \tag{13}$$

3.1.3. Constructing the Unit

After the extraction of CS, the AES encryption key expansion unit is also constructed by the DDBT structure, as shown in Figure 6.

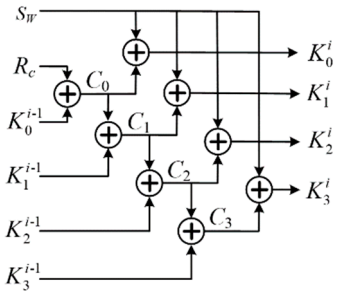

Figure 6. The AES encryption key expansion unit after extraction of CS.

After the extraction of CS, the critical path length of the AES encryption key expansion unit still is $T_{KE} = T_{SB} + 1T_{XOR} = 19T_{XOR} + 3T_{AND}$, but only 8 adders (which include seven 32-bit width adders and one 8-bit width adders) are used in the unit. The area cost of the AES encryption key expansion unit in Figure 6 is $A_{KE} = 4A_{SB} + 8A_{ADD} + A_{CST} = 591A_{XOR} + 140A_{AND}$. When converted to SMIC 0.18 μm technology, the delay and the area cost of the optimized DDBT structure in Figure 6 are 4.47 ns and 5288 Trs, respectively. Compared with the chain structure, the delay is reduced by 15.82% (0.84 ns) at a cost of 16.99% (768 Trs) area increase. Compared with un-optimized DDBT structure in Figure 5, a total of 6 adders (which include three 32-bit width and three 8-bit width) are reduced by the extraction of CS. When converted to SMIC 0.18 μm technology, 960 Trs are reduced, the reduction is up to 15.36%.

3.2. The Construction Method of Unified Key Expansion Unit

3.2.1. The Expressions of Unified Unit

According to the encryption key expansion expressions in Equation (7) and the decryption key expansion expressions in Equation (4), we can obtain the expressions for unified encryption/decryption key expansion operation as follows.

$$\begin{cases} {}^{+}K_0^i = S''_W + R''_c + K_0^{i-1} \\ {}^{+}K_1^i = S''_W + R''_c + K_0^{i-1} + K_1^{i-1} \\ {}^{+}K_2^i = S''_W + R''_c + K_0^{i-1} + K_1^{i-1} + K_2^{i-1} \\ {}^{+}K_3^i = S''_W + R''_c + K_0^{i-1} + K_1^{i-1} + K_2^{i-1} + K_3^{i-1} \\ {}^{-}K_0^i = S''_W + R''_c + K_0^{i-1} \\ {}^{-}K_1^i = K_0^{i-1} + K_1^{i-1} \\ {}^{-}K_2^i = K_1^{i-1} + K_2^{i-1} \\ {}^{-}K_3^i = K_2^{i-1} + K_3^{i-1} \end{cases} \tag{14}$$

where $S''_W = SubWord\left(RotWord\left(K_3^{i-1} / \left(K_2^{i-1} + K_3^{i-1}\right)\right)\right)$, $R''_c = Rcon(i)/Rcon^{-1}(i)$, $\left[{}^{+}K_0^i, {}^{+}K_1^i, {}^{+}K_2^i, {}^{+}K_3^i\right]$ is the output vector of the encryption key expansion operation, $\left[{}^{-}K_0^i, {}^{-}K_1^i, {}^{-}K_2^i, {}^{-}K_3^i\right]$ is the output vector of decryption key expansion operation. As shown in Equation 14), the input signals include $\left[S''_W, R''_c, K_0^{i-1}, K_1^{i-1}, K_2^{i-1}, K_3^{i-1}\right]$. Ex-

cept for the input signal S''_W, the delay of other input signals is 0. There is an adder, a two-to-one bus-multiplexer and a *Subword* operation before the input signal S''_W. Then, the delay of the input signal S''_W is $T_{XOR} + T_{MUX} + T_{SB} = 19T_{XOR} + 3T_{AND} + T_{MUX}$. According to Equation (8), we can determine that the critical path length of the DDBT structure is $T_{KE} = 20T_{XOR} + 3T_{AND} + T_{MUX}$.

3.2.2. Extracting the CS

The expressions in Equation (14) are also optimized by the DACSE algorithm, the delay constrain of the DACSE algorithm is $T_{con} = 23$. After optimization, the expressions for the unified encryption/decryption key expansion operation are as follows:

$$\begin{cases} {}^{+}K_0^i = C_3 \\ {}^{+}K_1^i = S''_W + C_0 + K_1^{i-1} \\ {}^{+}K_2^i = S''_W + C_2 \\ {}^{+}K_3^i = S''_W + C_2 + K_3^{i-1} \\ {}^{-}K_0^i = C_3 \\ {}^{-}K_1^i = K_0^{i-1} + K_1^{i-1} \\ {}^{-}K_2^i = C_1 \\ {}^{-}K_3^i = K_2^{i-1} + K_3^{i-1} \end{cases} \tag{15}$$

where $[C_0, C_1, C_2, C_3]$ is the extracted CS.

$$\begin{cases} C_0 = R''_c + K_0^{i-1} \\ C_1 = K_1^{i-1} + K_2^{i-1} \\ C_2 = C_0 + C_1 \\ C_3 = S''_W + C_0 \end{cases} \tag{16}$$

3.2.3. Constructing the Unit

The low-delay AES encryption/decryption unified key expansion unit is constructed by the DDBT structure according to (15), as shown in Figure 7. In the encryption key expansion state, the critical path of the unified unit is $T_{KE} = T_{SB} + 1T_{ADD} + 2T_{BMX}$. In the decryption key expansion state, the critical path of the unified unit is $T_{KE} = T_{SB} + 2T_{ADD} + 2T_{BMX}$. Therefore, the critical path of the whole unified unit is $T_{KE} = T_{SB} + 2T_{ADD} + 2T_{BMX} = 20T_{XOR} + 3T_{AND} + 2T_{MUX}$. Compared with the chain structure, 3 adders and 2 bus-multiplexers on the critical path length are reduced in the DDBT structure. When converted to SMIC 0.18 μm technology, the delay of the unified unit based on the DDBT structure is 5.06 ns, 1.01 ns are reduced compared with the chain structure, the reduction is up to 16.64%.

As shown in Figure 7, it requires a total of 5two-to-one bus-multiplexers (which include four 32-bit width bus-multiplexers and one 8-bit width bus-multiplexer), 11 adders (which include ten 32-bit width adders and one 8-bit width adder), and 2 constant multipliers in the unified unit based on the DDBT structure. The area cost is $A_{KE} = 4A_{SB} + 11A_{ADD} + 2A_{con} + 5A_{BMX} = 690A_{XOR} + 140A_{AND} + 136A_{MUX}$. When converted to SMIC 0.18 μm technology, the area cost is 7168 Trs, compared with the chain structure, the area cost is increased to 1536 Trs, the increase is up to 27.27%.

Compared with the DDBT structure, a total of 8 adders (which include four 32-bit width adders and four 8-bit width adders) are reduced by the DACSE algorithm. When converted to SMIC 0.18 μm technology, 1280 Trs are reduced, the optimization rate is up to 15.15%.

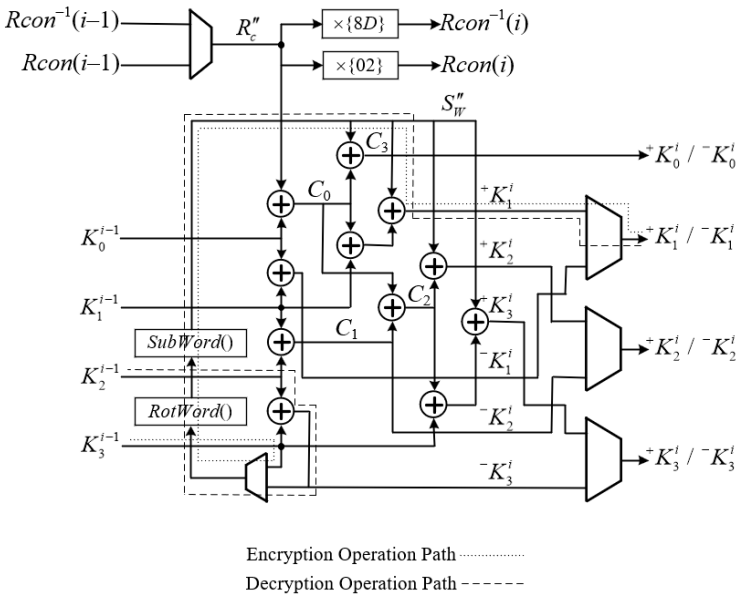

Figure 7. AES encryption/decryption unified key expansion unit based on the DDBT structure.

3.3. The Impact of Proposed Optimization Method on the Systems

As mentioned above, the delay is reduced at the cost of increased area using the proposed design method. But the impact of the increased area will be much smaller in the entire AES encryption/decryption system. An AES encryption/decryption circuit is shown in Figure 8.

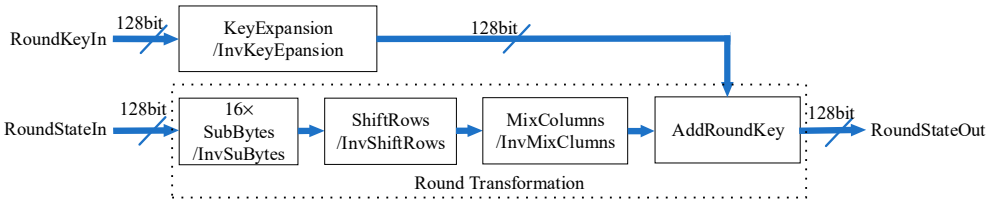

Figure 8. An example of AES encryption/decryption circuit.

We only take the round transformation unit into considerations in the AES encryption/decryption circuit, and the other units, such as the system controller, auxiliary circuits, and sequential logic circuits, are ignored. As shown in Figure 8, the round transformation unit includes *SubBytes*/*InvSubBytes*, *ShiftRows*/*InvShiftRows*, *MixColumns*/*InvMixColumns*, and *AddRoundKey*. Among these operations, *ShiftRows*/*InvShiftRows* does not require any logical operations, it can be achieved by simply shifting the bus sequence.

In AES encryption, the critical path delay of round transformation operations before *AddRoundKey* is $T_{SB} + T_{MC}$, where T_{MC} is the delay of *MixColumns*, and the minimum value of T_{MC} is $3T_{XOR}$ [18]. Therefore, the critical path delay is $T_{SB} + T_{MC} = 21T_{XOR} + 3T_{AND}$, and it is 4.89 ns in SMIC 0.18 µm technology, which is less than the delay of the key expansion unit based on chain structure (5.31 ns). Therefore, the key expansion unit becomes the critical path of the whole circuit. After adopting the DDBT structure, the delay of the key expansion unit is 4.47 ns, which is less than the delay of the round transformation unit, and the round transformation unit becomes the critical path. According to the area evaluation based on SMIC 0.18 µm technology, the area increase of the DDBT structure only accounts for about 2.63% of whole circuit.

In AES encryption unified unit, the critical path delay of round transformation operations before *AddRoundKey* is $T_{ISB} + T_{IMC} + T_{BMX} = 23T_{XOR} + 3T_{AND} + T_{MUX}$, and it is 5.50 ns in SMIC 0.18 µm technology, and it is also less than the delay of the key expansion

unit based on the chain structure (6.07 ns) and greater than the one of the DDBT structure (5.06 ns). Therefore, the key expansion unit becomes the critical path of the whole circuit. And the area increase of the DDBT structure only accounts for about 2.83% of the whole circuit.

Based on the above analysis, the round transformation unit becomes the critical path when the key expansion units are constructed by the DDBT structure. How to design a low-delay round transformation unit should be further studied in the further works.

4. Verification and Comparisons

4.1. Verification by Integrated Circuit Synthesis

Low-delay AES encryption key expansion unit and low-delay AES encryption/decryption unified key expansion unit are constructed in this paper based on the DDBT structure and the hardware complexities of these designs are theoretically analyzed. The comparisons of hardware complexities between the DDBT structure and the chain structure are summarized in Table 2. To evaluate accurately, the hardware complexities in Table 2 are converted to SMIC 0.18 μm technology.

Table 2. The comparisons of hardware complexities between the DDBT structure and the chain structure.

Unit	Structure	Area (Trs)	Delay (ns)	Frequency (MHz)
Encryption	Chain	4520	5.31	188.32
	DDBT	5288	4.47	223.71
Unified	Chain	5632	6.07	164.74
	DDBT	7168	5.06	197.63

As shown in Table 2, in both AES encryption key expansion unit and AES encryption/decryption unified key expansion unit, the DDBT structure has a lower delay. Compared with the chain structure, the delay of the DDBT structure is reduced by 15.82% at a cost of 16.99% area increase in the AES encryption key expansion unit, and the delay is reduced by 16.64% at a cost of 27.27% in the AES encryption/decryption unified key expansion unit. Note that, only the hardware complexities of combinational logic circuits are considered in the analysis in Table 2, and sequential logic circuits have not been taken into account. Our low-delay designs are further verified by the Synopsys (Sunnyvale, CA, USA) DC Tool with SMIC 0.18 μm technology and AMD (Santa Clara, CA, USA) Vivado 2019.2 with xc7vx485T. DC synthesis results and Vivado 2019.2 synthesis results are listed in Tables 3 and 4, respectively. In IC synthesized results, the delay of the DDBT structure is reduced by 15.23% at a cost of 10.08% area increase, in the AES encryption key expansion unit, and the delay of the DDBT structure is reduced by 67.52% at a cost of 12.25% area increase only in AES encryption/decryption unified key expansion unit. In FPGA synthesized results, the delay of the DDBT structure is reduced by 11.49% at a cost of 1.85% area increase in the AES encryption key expansion unit. In AES encryption/decryption unified key expansion unit, the delay of the DDBT structure is reduced by 76.64% at a cost of 6.83% area increase only. The synthesis results show that the delay of our designs is significantly reduced at a cost of less area increase.

Table 3. The IC synthesis results of key expansion units.

Unit	Structure	Area (gates/Trs)	Delay (ns)	Frequency (MHz)
Encryption	Chain	2550/7650	6.37	156.99
	DDBT	2807/8421	5.40	185.19
Unified	Chain	3264/9792	23.12	43.25
	DDBT	3664/10,992	7.51	133.16

Table 4. The FPGA synthesis results of key expansion units.

Unit	Structure	Area (Slices)	Delay (ns)	Frequency (MHz)
Encryption	Chain	108	6.725	148.70
	DDBT	110	5.952	168.01
Unified	Chain	117	28.285	35.35
	DDBT	125	6.607	151.35

4.2. Comparisons with Previous Works

All implementations of AES key expansion unit in previous works [2–6,10,11,19–37] are constructed by chain structure for saving the hardware resources. These implementations use different data width structures for different applications. In [2,10,11,19–24], 128-bit width structures are used for high data throughput applications. And 8-bit width structures are used in [3,5,6,25–28] for resources limited applications. To make a compromise between speed and area, 32-bit width structures are constructed in [4,29–33]. Besides, a 256-bit width structure is proposed in [34] for AES-256 key expansion operation. In 8-bit width structures and 32-bit width structures, AES key expansion unit needs to iterate several times to complete a round of subkey expansion operations. In 128-bit width structures and 256-bit width structures, the subkey data are processed in parallel to speed up the data processing. Our designs are compared with previous works in Table 5.

As we adopt 128-bit width to construct the DDBT structures in this paper, the previous works with 128-bit width structures [2,10,11,19–24] and 256-bit width structures [34] are listed in Table 5 only.

The frequencies from previous works in Table 5 are the maximum frequencies, except for the ones in [21,24], which are the work frequencies. As shown in Table 5, compared with previous works, the key expansion units proposed in this paper have the largest throughput. The throughput in Table 5 is computed by the following equation.

$$Throughput = \frac{SubkeyLength \times Rounds \times Frequency}{Cycles} \quad (17)$$

AES-128 inputs 128-bit initial key, and generates 10 rounds of 128-bit subkeys for round transformations through key expansion operation, while AES-256 input 256-bit initial key to generate 14 rounds 128-bit subkeys. The key expansion units proposed in this paper need 10 cycles to complete 10 round subkey expansion operations. As the critical path length is larger in the chain structure, pipeline stages are inserted into key expansion unit to speed up the working frequency in [2,10,11,19–21]. But with more pipeline stages, more clock cycles are required to implement the key expansion operations, which results in the decrease in throughput. The key expansion unit proposed in [34] also has a higher throughput due to the more advanced IC process and the 256-bit width structure adopted.

Table 5. Comparisons of previous works and our works.

Unit	Works	Platform	Area (gates)	Frequency (MHz)	Cycles	Throughput (Gbps)
Encryption	[10]	AMD Virtex	—	93.50	30	3.99
			—	168.40	70	3.08
	[19]	AMD Virtex-4	—	208.49	30	8.89
			—	247.19	50	6.33
	Ours	0.18 µm SMIC	2807	185.19	10	23.70
		AMD Virtex-7	—	168.01	10	21.51
Unified	[2]	AMD Virtex II	—	177.3	51	4.45
	[11]	AMD Virtex II	—	281.3	50	7.20
			—	305.1	90	4.34
	[20]	0.11 µm standard	3072	131.24	50	3.36
	[21]	0.25 µm TSMC	—	100	21	6.10
	[22]	0.6 µm standard	3071	64	10	8.19
	[23]	AMD Virtex-E	—	26.40	10	3.38
		AMD Spartan-3	—	28.01	10	3.58
	[24]	0.6 µm standard	3201	50	10	6.40
	[34]	90 nm standard	—	131.8	14	16.87
	Ours	0.18 µm SMIC	3664	133.16	10	17.04
		AMD Virtex-7	—	151.35	10	19.37

5. Conclusions and Future Works

In this paper, we propose a low-delay circuit structure construction method to construct low-delay AES key expansion units for high-speed AES implementations. An AES encryption key expansion unit and an AES encryption/decryption unified key expansion unit with low-delay and high-throughput are constructed by using the proposed construction method.

Side channel attacks are the most commonly used attacks against cryptographic chip currently. Among them, error analysis and power analysis are the two most effective side channel attacks [35–37]. Countermeasures at different abstract levels against error analysis and power analysis are proposed, including chip level, algorithm level, gate level, and transistor level. However, these measures introduce extra cost in terms of hardware resources and performance. How to apply our construction method on the AES key expansion units with countermeasures is the focus of our future research, especially on the algorithm level countermeasures, which are directly added to the source code of the cryptographic algorithms in hardware implementations [35].

In many applications, the encryption/decryption and key expansion are not operated in the same time. In these applications, the *SubBytes* units are often shared between the round transformation and key expansion to reduce the implementation area [2,20,22,24]. A low-delay unified structure for round transformation and key expansion can also be constructed based on our construction method proposed in this paper.

Author Contributions: Conceptualization, X.Z. (Xiaoqiang Zhang) and X.Z. (Xinxing Zheng); methodology, X.Z. (Xinxing Zheng); software, Z.P. and H.Y.; validation, Z.P. and H.Y.; formal analysis, X.Z. (Xinxing Zheng); investigation, X.Z. (Xinxing Zheng); resources, X.Z. (Xiaoqiang Zhang); data cura-

tion, H.Y.; writing—original draft preparation, X.Z. (Xinxing Zheng); writing—review and editing, X.Z. (Xiaoqiang Zhang); visualization, X.Z. (Xinxing Zheng); supervision, X.Z. (Xiaoqiang Zhang); project administration, X.Z. (Xiaoqiang Zhang); funding acquisition, X.Z. (Xiaoqiang Zhang) and X.Z. (Xinxing Zheng). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Program for the Talents Project of Wuhu Institute of Technology (No. 37 of WuZhiYuanRen[2021]), the Science and Technology Plan Projects of Wuhu (No. 2023yf091), the Scientific Research Project of Wuhu Institute of Technology (No. wzyzrzd202302), the Open Project of Anhui Engineering Research Center of Vehicle Display Integrated Systems (No. VDIS2023A01 and VDIS2023D02), the Excellent Scientific Research and Innovation Teams of Anhui Province (No. 2022AH010059), Anhui Provincial Quality Engineering Project for Institutions of Higher Education (No. 2021jyxm1699 and 2023xjzlts041).

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. The parameters of related logic gates in SIMC 0.18 μm technologies.

Logic Gate	Gate Type	Area (μm^2/Trs)	Delay (ns)
XOR	XOR2X1	26.6112/8	0.21
AND	AND2X1	13.3056/4	0.16
NAND	NAND2X1	9.9792/3	0.04
two-to-one Multiplexer	MX2X1	26.6112/8	0.19
NOT	INVX1	6.6528/2	0.03

References

1. NIST. FIPS PUB 197, Announcing the Advanced Encryption Standard (AES). 2001. Available online: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (accessed on 26 November 2001).
2. Yi, L.; Zou, X.; Liu, Z.; Dan, Y.; Zou, W. A low-cost compact AES architecture for wireless sensor network. *High Technol. Lett.* **2010**, *16*, 184–188.
3. Song, O.; Kim, J. Compact Design of the Advanced Encryption Standard Algorithm for IEEE 802.16.4 Devices. *J. Electr. Eng. Technol.* **2011**, *6*, 418–422. [CrossRef]
4. Hämäläinen, P.; Hännikäinen, M.; Hämäläinen, T. Efficient hardware implementation of security processing for IEEE 802.16.4 wireless networks. In Proceedings of the Midwest Symposium on Circuits and Systems, Covington, KY, USA, 7–10 August 2006; pp. 484–487.
5. Feldhofer, M.; Dominikus, S.; Wolkerstorfer, J. Strong Authentication for RFID Systems Using the AES Algorithm. In Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems, Cambridge, MA, USA, 11–13 August 2004; pp. 357–370.
6. Kim, M.; Ryou, J.; Choi, Y.; Jun, S. Low Power AES Hardware Architecture for Radio Frequency Identification. In Proceedings of the International Workshop on Security, Kyoto Terrsa, Kyoto, Japan, 23–24 October 2006; pp. 353–363.
7. Ali, L.; Aris, I.; Hossain, F.S.; Roy, N. Design of an ultra high speed AES processor for next generation IT security. *Comput. Electr. Eng.* **2011**, *37*, 1160–1170. [CrossRef]
8. Lin, S.Y.; Huang, C.T. A High-Throughput Low-Power AES Cipher for Network Applications. In Proceedings of the 2007 Asia and South Pacific Design Automation Conference, Yokohama, Japan, 23–26 January 2007; pp. 595–600.
9. Wang, Y.; Kumar, A.; Ha, Y. FPGA-based High Throughput XTS-AES Encryption/Decryption for Storage Area Network. In Proceedings of the International Conference on Field-Programmable Technology (FPT), Shanghai, China, 10–12 December 2014.
10. Zhang, X.; Parhi, K.K. High-Speed VLSI Architectures for the AES Algorithm. *IEEE Trans. VLSI Syst.* **2004**, *12*, 957–967. [CrossRef]
11. Hammad, I.; El-Sankary, K.; El-Masry, E. High-Speed AES Encryptor With Efficient Merging Techniques. *IEEE Embed. Syst. Lett.* **2010**, *2*, 67–71. [CrossRef]

12. Kundi, D.-S.; Aziz, A.; Ikram, N. A high performance ST-Box based unified AES encryption/decryption architecture on FPGA. *Microprocess. Microsyst.* **2016**, *41*, 37–46. [CrossRef]
13. Petra, N.; Caro, D.D.; Strollo, A.G.M. A Novel Architecture for Galois Fields $GF(2^m)$ Multipliers Based on Mastrovito Scheme. *IEEE Trans. Comput.* **2007**, *56*, 1470–1483. [CrossRef]
14. Zhang, X.; Wu, N.; Zhou, F.; Li, J. Low-delay parallel Chien search architecture for RS decoder. *IEICE Electron. Express* **2016**, *13*, 20160729. [CrossRef]
15. Zhang, X.; Wu, N.; Zhou, F.; Chen, X. An optimized delay-aware common subexpression elimination algorithm for hardware implementation of binary-field linear transform. *IEICE Electron. Express* **2014**, *11*, 20140934. [CrossRef]
16. Zhang, X.; Wu, N.; Zhou, F.; Ge, F. Optimization of Area and Delay for Implementation of the Composite Field Advanced Encryption Standard S-Box. *J. Circuits Syst. Comput.* **2016**, *25*, 1650037. [CrossRef]
17. Zhang, X.; Peng, Z.; Yan, H.; Zheng, X.; Xu, M. A Low-Delay Circuit Structure Construction Method for AES Key Expansion Units. In Proceedings of the IEEE 19th Conference on Industrial Electronics and Applications, Kristiansand, Norway, 5–8 August 2024; pp. 1–4.
18. Zhang, X.; Yang, F.; Zheng, X.; Zhang, X.; Wu, N. A Full Matrix Joint Optimization Method for Hardware Implementation of AES MixColumns/InvMixColumns. *IEICE Electron. Express* **2020**, *24*, 20200391. [CrossRef]
19. Vinh, T.Q.; Park, J.-H.; Kim, Y.; Kim, K.-O. An FPGA implementation of 30Gbps security module for GPON systems. In Proceedings of the IEEE International Conference on Computer and Information Technology, Sydney, NSW, Australia, 8–11 July 2008; pp. 868–872.
20. Satoh, A.; Morioka, S.; Takano, K.; Munetoh, S. A compact Rijndael hardware architecture with S-box optimization. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, 9–13 December 2001; pp. 239–245.
21. Lu, C.C.; Tseng, S.Y. Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In Proceedings of the IEEE International Conference on Application—Specific Systems, Architectures, and Processors, San Jose, CA, USA, 17–19 July 2002; pp. 277–286.
22. Mangard, S.; Aigner, M.; Dominikus, S. A highly regular and scalable AES hardware architecture. *IEEE Trans. Comput.* **2003**, *52*, 483–491. [CrossRef]
23. Good, T.; Benaissa, M. AES on FPGA from the fastest to the smallest. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Edinburgh, UK, 29 August–1 September 2005; pp. 438–451.
24. Pramstaller, N.; Mangard, S.; Dominikus, S.; Wolkerstorfer, J. Efficient AES Implementations on ASICs and FPGAs. In Proceedings of the 4th International Conference on AES, Bonn, Germany, 10–12 May 2004; pp. 98–112.
25. Hämäläinen, P.; Alho, T.; Hännikäinen, M.; Hämäläinen, T.D. Design and Implementation of Low-area and Low-power AES Encryption Hardware Core. In Proceedings of the the 9th Euromicro Conference on Digital System Design, Cavtat, Croatia, 30 August—1 September 2006; pp. 577–583.
26. Good, T.; Benaissa, M. 692-nW Advanced Encryption Standard (AES) on a 0.13-µm CMOS. *IEEE Trans. VLSI Syst.* **2010**, *18*, 1753–1757. [CrossRef]
27. Chen, M.-C.; Li, W.-T. An 8-Bit ROM-Free AES Design for Low-Cost Applications. *Math. Probl. Eng.* **2013**, *1*, 55554477. [CrossRef]
28. Kim, H.K.; Sunwoo, M.H. Low Power AES Using 8-Bit and 32-Bit Datapath Optimization for Small Internet-of-Things (IoT). *J. Sig. Proc. Syst.* **2019**, *91*, 1283–1289. [CrossRef]
29. Chodowiec, P.; Gaj, K. Very Compact FPGA Implementation of the AES Algorithm. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Cologne, Germany, 8–10 September 2003; pp. 319–333.
30. Rouvroy, G.; Standaert, F.X.; Quisquater, J.J.; Legat, D.J. Compact and Efficient Encryption/Decryption Module for FPGA Implementation of the AES Rijndael Very Well Suited for Small Embedded Applications. In Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA, 5–7 April 2004; pp. 583–587.
31. Yu, N.; Heys, H.M. Investigation of compact hardware implementation of the advanced encryption standard. In Proceedings of the Canadian Conference on Electrical and Computer Engineering, Saskatoon, SK, Canada, 1–4 May 2005; pp. 1069–1072.
32. Jia, Z.; Zeng, X.; Han, J.; Chen, J. Very Low-cost VLSI Implementation of AES Algorithm. In Proceedings of the IEEE Asian Solid-State Circuits Conference, Hangzhou, China, 13–15 November 2006.
33. Li, Z.-R.; Zhuang, Y.-Q.; Zhang, C.; Jin, G. Low-power and area-optimized VLSI implementation of AES coprocessor for Zigbee system. *J. China Univ. Posts Technol.* **2009**, *16*, 89–94. [CrossRef]
34. Liu, P.-C.; Chang, H.-C.; Lee, C.-Y. A 1.69 Gb/s area-efficient AES crypto core with compact on-the-fly key expansion unit. In Proceedings of the European Solid-State Circuit Conference, Athens, Greece, 14–18 September 2000; pp. 404–407.
35. Marzouqi, H.; Al-Qutayri, M.; Salah, K. Review of gate-level differential power analysis and fault analysis countermeasures. *IET Inf. Secur.* **2014**, *8*, 51–66. [CrossRef]

36. Karaklajić, D.; Schmidt, J.-M.; Verbauwhede, I. Hardware Designer's Guide to Fault Attacks. *IEEE Trans. VLSI Syst.* **2013**, *12*, 2295–2306. [CrossRef]
37. Barenghi, A.; Breveglieri, L.; Koren, I.; Naccache, D. Fault Injection Attacks on Cryptographic Devices: Theory, Practice, and Countermeasures. *Proc. IEEE* **2012**, *100*, 3056–3076. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.