ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.11, No.22, 1–8

An optimized delay-aware common subexpression elimination algorithm for hardware implementation of binary-field linear transform

Xiaoqiang Zhang[a)], **Ning Wu**[b)], **Fang Zhou, and Xin Chen**

College of Electrical and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

a) *zxq198111@qq.com*

b) *wunee@nuaa.edu.cn*

Abstract: When implementing non-multiplier linear systems, delay-aware common subexpression elimination (DACSE) is a critical algorithm for optimizing the area efficiency under a given timing constraint. In this paper, we propose an optimized DACSE algorithm for the hardware implementation of binary-field linear transform (BFLT). In order to achieve the shortest critical path delay (CPD), the proposed algorithm uses fast-binary-tree structure to implement the BFLT circuit before sharing common subexpressions (CSs). However, as the delays of involved signals are different after sharing CSs, the *delay-driven-binary-tree* (DDBT) structure is adopted to further optimize the critical path of the BFLT logics. The CPD of the DDBT based circuit is evaluated for each case with an eliminated CS, and the CS elimination will be abandoned if the case cannot meet the given timing constraint. Moreover, the proposed algorithm provides all of the design trade-offs, from the shortest feasible CPD to the smallest area, to designers, offering them the maximum design space. Experiments are carried out to verify the proposed algorithm and the results show that the proposed DACSE is more efficient in area reduction than the previous works, especially under a stringent timing constraint.

Keywords: binary-field linear transform, common subexpression elimination, critical path delay

Classification: Integrated circuits

References

[1] M. M. Wong and M. L. D. Wong: Tenth International Conference on Information Sciences, Signal Processing and their Applications (ISSPA 2010) (2010) 452. DOI:10.1109/ISSPA.2010.5605445

[2] J. L. Imaña, J. M. Sánchez and F. Tirado: IEEE Trans. Comput. **55** (2006) 520. DOI:10.1109/TC.2006.69

[3] M. E. Pellenz, R. D. Souza and M. S. P. Fonseca: Telecomm. Syst. **44** (2010) 61. DOI:10.1007/s11235-009-9222-5

DOI: 10.1587/elex.11.20140934
Received September 29, 2014
Accepted October 23, 2014
Publicized November 7, 2014
Copyedited November 25, 2014

[4] O. Song and J. Kim: J. Electr. Eng. Technol. **6** (2011) 418. DOI:10.5370/JEET.2011.6.3.418
[5] L. Fu, X. Shen, L. Zhu and J. Wang: Secur. Commun. Networks **7** (2014) 365. DOI:10.1002/sec.723
[6] A. Hosangadi, F. Fallah and R. Kastner: International Workshop of Logic and Synthesis (IWLS) (2005).
[7] R. Maheshand and A. P. Vinod: IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **29** (2010) 275. DOI:10.1109/TCAD.2009.2035548
[8] N. Petra, D. D. Caro and A. G. M. Strollo: IEEE Trans. Comput. **56** (2007) 1470. DOI:10.1109/TC.2007.70741
[9] M. Martínez-Peiró, E. I. Boemo and L. Wanhammar: IEEE Trans. Circuits Syst. II, Exp. Briefs **49** (2002) 196. DOI:10.1109/TCSII.2002.1013866
[10] N. Chen and Z. Y. Yan: IEEE International Symposium on Circuits and Systems (ISCAS 2009) (2009) 2906. DOI:10.1109/ISCAS.2009.5118410
[11] P. Cappello and K. Steiglitz: IEEE Trans. Acoust. Speech Signal Process. **32** (1984) 1037. DOI:10.1109/TASSP.1984.1164433
[12] X. Zhang and K. K. Parhi: IEEE Trans. Circuits Syst. II, Exp. Briefs **53** (2006) 1153. DOI:10.1109/TCSII.2006.882217

1 Introduction

A common binary-field linear transform (BFLT) operation can be expressed as $\boldsymbol{Y} = \boldsymbol{MX}$, where $\boldsymbol{Y}$ and $\boldsymbol{X}$ are n- and m-dimensional binary column vectors, respectively, and $\boldsymbol{M}$ is an $n \times m$ binary constant matrix. The linear transform $\boldsymbol{Y} = \boldsymbol{MX}$ can also be expressed as a set of bit-level equations that only contain bit-wise additions. In binary field, an addition is performed by a XOR operation [1]. Therefore only two-input XOR gates are required in the pure combinational logic implementations of BFLT. At present, BFLT has been widely used in modern cryptography algorithms and error correcting codes [2]. However, cryptography algorithms and error correcting codes are often used in resource-limited applications, such as wireless sensor networks [3, 4] and radio frequency identifiers [5]. So an efficient hardware implementation of BFLT is highly desirable in these applications.

The efficiency of the hardware implementations is usually measured by area and critical path delay (CPD) [2]. It is well known that the direct implementation of BFLT leads to a lot of redundant gates. For improving area efficiency, the common subexpression elimination (CSE) algorithm is usually adopted to eliminate the redundant gates. The CSE algorithm identifies the common subexpressions (CSs) that occur more than once in bit-level equations, and replaces each of them with a new signal. With the replacement, each of the CSs is computed only once. Thus the number of gates is reduced in hardware implementations [1].

A shorter CPD indicates a higher throughout. For a specified circuit with a certain gate count, the CPD is only decided by the adopted structure. For a circuit that contains only one kind of two-input gate, it constructed with *Fastest-Binary-Tree* (FBT) structure has the shortest CPD [6, 7] on condition that the input signals have the same input delay. On the contrary, if the input signals have different input delays, the circuit constructed with *Delay-Driven-Binary-Tree* (DDBT) structure has the shortest CPD [8]. Obviously, the FBT structure can be regarded as a special

DOI: 10.1587/elex.11.20140934
Received September 29, 2014
Accepted October 23, 2014
Publicized November 7, 2014
Copyedited November 25, 2014

case of the DDBT structure. Suppose that the delays before input signals are ignored. Then a direct implementation of BFLT with FBT structure has the shortest CPD.

However, sharing CSs will increase the CPD, since the FBT structure is destroyed [6]. To avoid destroying the FBT structure, only the CSs which contain 2^i (i is a positive integer) *original* signals are selected to be eliminated [1, 9]. But the area reduction is limited, since the selected sharing CSs is limited [6].

In order to achieve different trade-offs between area and CPD, Delay-Aware CSE (DACSE) algorithms are proposed in [6, 10]. In [6], the delays of the design are evaluated and controlled during the processing of selecting CSs. So the scope of CS selection is extended. After sharing CSs, the FBT structure is adopted to construct the circuit. Nevertheless, the FBT structure is not the optimal structure in this case, as the new signals appended for replacing the CSs have different delay values. To achieve the shortest CPD after sharing CSs, the DDBT is adopted in [10]. The major drawback in [10] is that delays are evaluated only after all CSs are eliminated. If the CPD cannot meet with the given timing constraint, the eliminations of CSs are cancelled, and the algorithm reverts back to resort to other strategies of CSE. If CPD still cannot meet with the given timing constraint after all solutions are tried, the operation of CSE is abandoned and the circuit is implemented in the direct way.

To overcome the drawbacks in the above works, this paper proposes an optimized DACSE algorithm. First, to achieve the shortest CPD after sharing CSs, the proposed DACSE algorithm adopts the DDBT structure to construct the optimized circuit. Second, to avoid a violation of meeting the given timing constraint in the backtracking method, the delay of DDBT based circuit is evaluated when each CS is selected to be eliminated. The elimination of the selected CS will be abandoned if it leads the CPD violating the given timing constraint, The proposed DACSE algorithm provides a broad range of design trade-offs between area and CPD, from the shortest feasible CPD to the smallest area. To verify the proposed algorithm, several experiments are made and the results show that the proposed DACSE is more efficient in area reduction than the existing DACSE algorithms.

2 The proposed DACSE algorithm

In this section, the principles of the proposed DACSE algorithm are presented. As a BFLT circuit only contains XOR gates, the area is measured in terms of the number of XOR gates (which denote as XORs) and the delays are measured in terms of the total XOR gate delays (which denote as T_{XOR}) in this paper. Suppose the delays before input signals can be ignored in the BFLT operations mentioned in the following. By using the proposed DACSE algorithm, the implementation of an $n \times m$ BFLT operation involves the following steps.

1 Compute the lower limit of timing constraint T_{Cmin}. The lower limit of timing constraint T_{Cmin} is given in the first step of the algorithm. The given timing constraint T_C must be satisfied with $T_C \geq T_{Cmin}$. T_{Cmin} should be equal to the shortest feasible CPD achieved by the BFLT circuit. As sharing CSs will increase

DOI: 10.1587/elex.11.20140934
Received September 29, 2014
Accepted October 23, 2014
Publicized November 7, 2014
Copyedited November 25, 2014

the delays, the BFLT circuit that is direct implemented with FBT structure has the shortest feasible CPD. Let T_{FBT} denote the shortest feasible CPD, then $T_{\mathrm{Cmin}} = T_{\mathrm{FBT}}$, and T_{FBT} is given by

$$T_{\mathrm{FBT}} = \max_{i=0,\cdots,n-1}(T_i) = \max_{i=0,\cdots,n-1}\left(\lceil \log_2 N_i \rceil\right) \tag{1}$$

where T_i is the delay of output signal y_i, N_i is the number of involved signals in the expression of output signal y_i, and $\lceil x \rceil$ represents the smallest integer which is larger than or equal to x. The unit of T_{FBT} is T_{XOR}.

2 Select a CS to be eliminated. If the given timing constraint T_{C} is satisfied with $T_{\mathrm{C}} \geq T_{\mathrm{Cmin}}$, the algorithm starts to identify CSs in the equations. As the simplest CS consists of two terms and a *multi-term* CS can be expressed recursively as *two-term* CSs, only *two-term* CSs are taken into account in the algorithm. However, how to select a CS to be eliminated to achieve maximal area reduction is a NP-complete problem [11]. In our algorithm, the CS with the highest occurrence frequency is selected to be eliminated.

3 Evaluate the delays when a CS is selected. Once a CS is selected to be eliminated, the delays of the design after sharing the selected CS should be evaluated. If the CPD of the design larger than the given timing constraint T_{C}, the elimination of selected CS is abandoned. First of all, the delay of the new signal that is appended for replacing with the selected CS is computed, since the new signal is involved in the design. Suppose that selected CS $x_p + x_q$ is replaced with the new signal x_{new}, the delay of x_{new} is given by

$$t_{new} = \max(t_p, t_q) + 1 \tag{2}$$

where t_{new}, t_p, and t_q are the delays of x_{new}, x_p, and x_q, respectively. Then, the delays of the design after sharing the selected CS are computed. After sharing CSs, the delays of the signals involved in the design are different, so the DDBT structure is adopted to construct the circuit. The constructed method of DDBT structure is described in detail in [8]. Let T_{DDBT} denote the CPD of the DDBT based circuit. Then T_{DDBT} is given by

$$T_{\mathrm{DDBT}} = \max_{i=0,\cdots,n-1}(T_i) = \max_{i=0,\cdots,n-1}\left(\left\lceil \log_2 \sum_{N_i} 2^{t_k} \right\rceil\right) \tag{3}$$

where t_k is the delay of input signal x_k. The elimination of the selected CS should keep T_{DDBT} less than or equal to the given timing constraint T_{C}, and thus the following inequality should hold true:

$$T_C \geq \left\lceil \log_2 \sum_{N_i} 2^{t_k} \right\rceil \tag{4}$$

If inequality (4) is not satisfied, the elimination of the selected CS is canceled. Inequality (4) can be rewritten in the following way:

$$2^{T_C} \geq \sum_{N_i} 2^{t_k} \tag{5}$$

Inequality (5) can be implemented by simple operations, such as shifts, additions and comparisons.

4 If all CSs have been checked, end the algorithm; if not, then return to the Step 2.

DOI: 10.1587/elex.11.20140934
Received September 29, 2014
Accepted October 23, 2014
Publicized November 7, 2014
Copyedited November 25, 2014

As a matrix is easier to handle by a computer program, the constant matrix $\boldsymbol{M}$ is used as input vector for the proposed DACSE algorithm. The output vector is the transformed form of $\boldsymbol{M}$. The computational complexity of the proposed DACSE algorithm only depends on the dimensions of $\boldsymbol{M}$. As the dimensions of transformed matrix $\boldsymbol{M}$ are $n \times (m+s-1)$ at the s^{th} iteration, the computational complexity of the s^{th} iteration is $O(n \times (m+s-1)) + O(nC^2{}_{s+m-1}) + O(n(m+s-1)) \approx O(n(m+s)^2)$. As there are at most $C^2{}_m$ identified CSs, i.e., there are at most $C^2{}_m$ iterations in the proposed algorithm, the worst case computational complexity for the proposed algorithm is $\sum_{s=1}^{m(m-1)/2} O(n(m+s)^2) \approx O(nm^6)$. As the candidate pattern at an iteration is often more than one, the first CS is selected to be evaluated in the proposed algorithm. Moreover, a greedy search can be used to check all candidate patterns to search the optimized results for a small-scale BFLT operation [1]. Then the worst case computational complexity for greedy search is $O(C^2{}_m! \times nm^6)$.

3 An example

We will take an example to illustrate the proposed algorithm in this section. Consider a 3×6 BFLT operation as follows:

$$\begin{cases} y_0 = x_0 + x_1 + x_2 + x_3 + x_4 + x_5 \\ y_1 = x_0 + x_1 + x_2 + x_3 + x_5 \\ y_2 = x_0 + x_1 + x_2 \end{cases} \tag{6}$$

Let T_i denote the delay of output signal y_i, and t_j denote the delay of input signal x_j. The circuit of computation in (6) that is direct implemented with FBT structure is illustrated in Fig. 1. According to (1), the shortest feasible CPD T_{FBT} is obtained as $3T_{\text{XOR}}$. Suppose that the given timing constraint $T_{\text{C}} = 3T_{\text{XOR}}$, by using the proposed DACSE algorithm, the elimination of CSs is expressed as follows:

$$\begin{cases} y_0(@3) = x_0 + x_1 + x_2 + x_3 + x_4 + x_5 \\ y_1(@3) = x_0 + x_1 + x_2 + x_3 + x_5 \\ y_2(@2) = x_0 + x_1 + x_2 \end{cases} \rightarrow \begin{cases} y_0(@3) = x_2 + x_3 + x_4 + x_5 + x_6 \\ y_1(@3) = x_2 + x_3 + x_5 + x_6 \\ y_2(@2) = x_2 + x_6 \\ x_6(@1) = x_0 + x_1 \end{cases}$$

$$\rightarrow \begin{cases} y_0(@3) = x_3 + x_4 + x_5 + x_7 \\ y_1(@3) = x_3 + x_5 + x_7 \\ y_2(@2) = x_7 \\ x_6(@1) = x_0 + x_1 \\ x_7(@2) = x_2 + x_6 \end{cases} \rightarrow \begin{cases} y_0(@3) = x_4 + x_7 + x_8 \\ y_1(@3) = x_7 + x_8 \\ y_2(@2) = x_7 \\ x_6(@1) = x_0 + x_1 \\ x_7(@2) = x_2 + x_6 \\ x_8(@1) = x_3 + x_5 \end{cases} \tag{7}$$

At the first iteration, CS $x_0 + x_1$ is replaced with a new signal x_6, as it is one of the CSs with the highest occurrence-frequency. According to (2), delay t_6 is obtained as $1T_{\text{XOR}}$. After sharing CS $x_0 + x_1$, delays T_0, T_1 and T_2 are all satisfied with inequality (5). In the same way, CS $x_2 + x_6$ is replaced with signal x_7 at the second iteration, and delay t_7 is obtained as $2T_{\text{XOR}}$. There are three CSs with the highest occurrence-frequency at the third iteration, only after sharing CS $x_3 + x_5$,

DOI: 10.1587/elex.11.20140934
Received September 29, 2014
Accepted October 23, 2014
Publicized November 7, 2014
Copyedited November 25, 2014

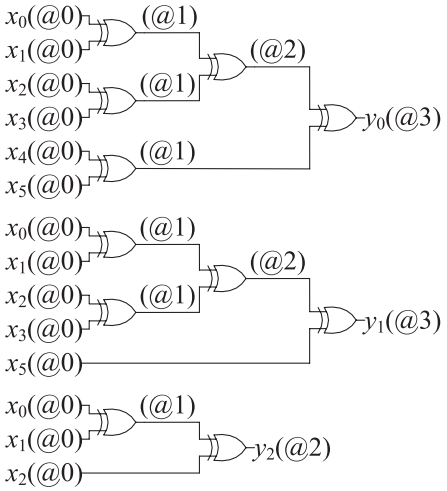

Fig. 1. The direct implementation with FBT structure

the delays T_0, T_1 and T_2 are all satisfied with inequality (5). Signal x_8 replaces CS $x_3 + x_5$, and it can be obtained that $t_8 = 1T_{\mathrm{XOR}}$. CS $x_7 + x_8$ is the only CS at the fourth iteration, but the elimination of it will lead to T_0 unsatisfied with inequality (5), and thus the elimination is abandoned.

The direct implementation of the BFLT requires 11XORs. After optimized by the proposed DACSE, the implementation only requires 6XORs. As in DDBT structure, two signals with minimal delay are first taken to be implemented [8]. For output signal y_0 in (7), the operation $x_4 + x_8$ is first taken to construct the circuit, as shown in Fig. 2(a). In [6], as the delays are not taken into consideration in the constructing circuit, the operation $x_4 + x_7$ is first taken to construct the circuit, as shown in Fig. 2(b). That will lead to T_0 unsatisfied with inequality (5), and thus the elimination of CS $x_3 + x_5$ at the third iteration is abandoned. Only CS $x_0 + x_1$ and CS $x_2 + x_6$ are eliminated by the DACSE algorithm proposed in [6].

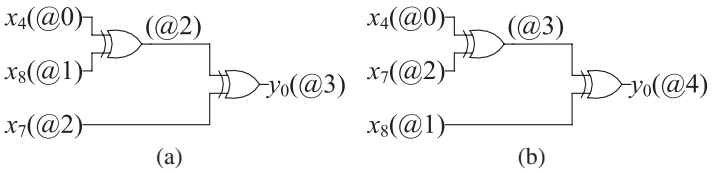

Fig. 2. The optimized circuit for output signal y_0: (a) constructed with DDBT structure; (b) constructed with FBT structure

For keeping the shortest feasible CPD, only two-term non-recursive CSs, i.e., the CSs containing two *original* signals, are selected to be eliminated by the Non-Recursive CSE (NR-CSE) algorithm proposed in [9]. Then non-recursive CS $x_0 + x_1$ and non-recursive CS $x_2 + x_3$ are eliminated by the NR-CSE algorithm. In [1], not only two-term non-recursive CSs but also the particular recursive CSs, which contain 2^i (i is a positive integer) *original* signals, are selected to be eliminated by the CSE algorithm. Then recursive CS $x_6 + x_7$ will be further eliminated by the CSE algorithm proposed in [1]. The corresponding elimination is expressed as

DOI: 10.1587/elex.11.20140934
Received September 29, 2014
Accepted October 23, 2014
Publicized November 7, 2014
Copyedited November 25, 2014

$$\begin{cases} y_0(@3) = x_0 + x_1 + x_2 + x_3 + x_4 + x_5 \\ y_1(@3) = x_0 + x_1 + x_2 + x_3 + x_5 \\ y_2(@2) = x_0 + x_1 + x_2 \end{cases} \rightarrow \begin{cases} y_0(@3) = x_2 + x_3 + x_4 + x_5 + x_6 \\ y_1(@3) = x_2 + x_3 + x_5 + x_6 \\ y_2(@2) = x_2 + x_6 \\ x_6(@1) = x_0 + x_1 \end{cases}$$

$$\rightarrow \begin{cases} y_0(@3) = x_4 + x_5 + x_6 + x_7 \\ y_1(@3) = x_5 + x_6 + x_7 \\ y_2(@2) = x_2 + x_6 \\ x_6(@1) = x_0 + x_1 \\ x_7(@1) = x_2 + x_3 \end{cases} \rightarrow \begin{cases} y_0(@3) = x_4 + x_5 + x_8 \\ y_1(@3) = x_5 + x_8 \\ y_2(@2) = x_2 + x_6 \\ x_6(@1) = x_0 + x_1 \\ x_7(@1) = x_2 + x_3 \\ x_8(@2) = x_6 + x_7 \end{cases} \quad (8)$$

In DACSE algorithm proposed in [10], the delays are only calculated after all CSs are eliminated. By using this DACSE algorithm, all solutions of CS elimination are failed to construct the circuit satisfied with the given timing constraint $T_C = 3T_{XOR}$. So the circuit will be implemented in the direct way.

The results optimized by the works are listed in Table I. As shown in Table I, under timing constraint $T_C = 3T_{XOR}$, the DACSE algorithm proposed in this paper has more efficient in area reduction than previous works. Under timing constraint $T_C = 4T_{XOR}$, the smallest area (which is 5 XORs) is achieved by using the DACSE algorithm proposed in this paper and in [6, 10], respectively. However, the area reduction can not be further improved under a looser delay constraint for the CSE algorithms proposed in [1, 9].

Table I. The results optimized by the CSE algorithms

		Direct	[9]	[1]	[6]	[10]	Ours
$T_C = 3T_{XOR}$	Area (XORs)	11	8	7	7	11	6
	Eliminated CSs	—	$x_6 = x_0 + x_1$ $x_7 = x_2 + x_3$	$x_6 = x_0 + x_1$ $x_7 = x_2 + x_3$ $x_8 = x_6 + x_7$	$x_6 = x_0 + x_1$ $x_7 = x_2 + x_6$	Null	$x_6 = x_0 + x_1$ $x_7 = x_2 + x_6$ $x_8 = x_3 + x_5$
$T_C = 4T_{XOR}$	Area (XORs)	11	8	7	5	5	5
	Eliminated CSs	—	$x_6 = x_0 + x_1$ $x_7 = x_2 + x_3$	$x_6 = x_0 + x_1$ $x_7 = x_2 + x_3$ $x_8 = x_6 + x_7$	$x_6 = x_0 + x_1$ $x_7 = x_2 + x_6$ $x_8 = x_3 + x_5$ $x_9 = x_7 + x_8$	$y_0 = x_4 + y_1$ $y_1 = x_3 + x_5 + y_2$ $y_2 = x_0 + x_1 + x_2$	$x_6 = x_0 + x_1$ $x_7 = x_2 + x_6$ $x_8 = x_3 + x_5$ $x_9 = x_7 + x_8$

4 Experiments

In this experiment, a number of BFLT operations in implementation of Advanced Encryption Standard (AES) S-box are used to further evaluate the efficiency of the proposed DACSE algorithm. Three 8×8 BFLT operations are used in implementation of AES S-box based on the composite field arithmetic [1]: isomorphism matrix, inverse-isomorphism matrix, and affine matrix. Inverse-isomorphism matrix is always combined with affine matrix for reducing area [1]. Eight different isomorphism matrices ($\boldsymbol{\beta}_0 \sim \boldsymbol{\beta}_7$), which are generated by the **Algorithm 1** in [12] in the case of $\{\Phi = (10)_2, \lambda = (1100)_2\}$, and their corresponding Inverse-Isomorphism-Affine (IIA) matrices are used in this experiment. Our algorithm and

DOI: 10.1587/elex.11.20140934
Received September 29, 2014
Accepted October 23, 2014
Publicized November 7, 2014
Copyedited November 25, 2014

other related works are implemented by MATLAB. The total area of isomorphism matrix and IIA matrix, which are optimized by our algorithm and other related works under $T_C = 3T_{XOR}$ and $T_C = 4T_{XOR}$, are listed in Table II. Compared with other works, the DACSE algorithm proposed in this paper have more efficient in area reduction, especially under a tight timing constraint.

Table II. The total area of isomorphism matrix and IIA matrix optimized by the works

Constraints	Works	Total Area (XORs)									Reduction (%)
		β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	Average	
$T_C = 3T_{XOR}$	Direct	47	48	45	50	43	46	53	47	47.38	—
	[9]	32	34	30	31	30	32	35	32	32	32.45
	[1]	29	32	28	30	28	30	31	30	29.75	37.20
	[6]	31	29	27	30	27	31	36	30	30.13	36.41
	[10]	28	30	27	41*	38*	29	30	31	31.75	32.99
	Ours	28	29	27	30	27	28	30	30	28.63	39.58
$T_C = 4T_{XOR}$	[6]	28	29	26	27	27	28	29	29	27.88	41.16
	[10]	28	30	27	27	28	28	30	30	28.5	39.84
	Ours	28	29	26	27	27	28	29	29	27.88	41.16

*All solutions of CS elimination are failed to meet the given timing constraint in optimization of IIA matrix, and the circuit of IIA matrix is implemented in the direct way.

5 Conclusions

This paper presents an optimized DACSE algorithm for the hardware implementations of BFLT. The proposed DACSE overcomes the drawbacks of the traditional DACSE algorithms, and thus achieves better area efficiency under stringent timing constraint. Moreover, our approach also provides a broad range of trade-offs between the area and the CPD, offering the designer the maximum design space to explore. In order to show the effectiveness of the proposed algorithm, we implement several 8×8 BFLT operations in AES S-box. Experimental results show that the proposed DACSE is more efficient in area reduction than the previous studies, especially under tight timing constraints. Although we only illustrate the BFLT operations in this paper, the proposed DACSE algorithm is also applicable to general multiple constant multiplication operations, which are usually used in a large set of DSP applications.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61376025, No. 61106029), the Industry-academic Joint Technological Innovations Fund Project of Jiangsu (No. BY2013003-11), the Funding of Jiangsu Innovation Program for Graduate Education (No. KYLX_0273), and the Fundamental Research Funds for the Central Universities.

DOI: 10.1587/elex.11.20140934
Received September 29, 2014
Accepted October 23, 2014
Publicized November 7, 2014
Copyedited November 25, 2014