Low-delay parallel Chien search architecture for RS decoder

Xiaoqiang Zhang[1,2a)], **Ning Wu**[1b)], **Fang Zhou**[1], **Jianhua Li**[1], **and Yasir**[1]

[1] *College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*

[2] *College of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, China*

a) *zxq198111@qq.com*

b) *wunee@nuaa.edu.cn*

Abstract: Sharing common subexpressions (CSs) in the logic expressions can reduce the total gates in hardware implementations of parallel Chien search. In this paper, we prove that sharing CSs will increase the delays of the hardware implementation. Based on the proof, a shortest-path-keep common subexpression elimination (SPK-CSE) algorithm is proposed. By using SPK-CSE algorithm, the output delays can be kept unchanged after sharing CSs. The parallel Chien search implemented with the proposed SPK-CSE algorithm can achieve the minimal delay.

Keywords: Chien search, common subexpression elimination, critical path delay

Classification: Integrated circuits

References

[1] Y. Chen and K. K. Parhi: "Small area parallel Chien search architectures for long BCH codes," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **12** (2004) 545 (DOI: 10.1109/TVLSI.2004.826203).

[2] Q. Hu, *et al.*: "Low complexity parallel Chien search architecture for RS decoder," Proc. IEEE Int. Symp. Circuits & Syst. **1** (2005) 340.

[3] J. Cho and W. Sung: "Strength-reduced parallel Chien search architecture for strong BCH codes," IEEE Trans. Circuits Syst. II, Exp. Briefs **55** (2008) 427 (DOI: 10.1109/TCSII.2007.914898).

[4] Y. Lee, *et al.*: "Low-complexity parallel Chien search structure using two-dimensional optimization," IEEE Trans. Circuits Syst. II, Exp. Briefs **58** (2011) 522 (DOI: 10.1109/TCSII.2011.2158709).

[5] X. Li, *et al.*: "Efficient architecture for algebraic soft-decision decoding of Reed-Solomon codes," IET Commun. **9** (2015) 10 (DOI: 10.1049/iet-com.2014.0460).

[6] P. R. Cappello and K. Steiglitz: "Some complexity issues in digital signal processing," IEEE Trans. Acoust. Speech Signal Process. **32** (1984) 1037 (DOI: 10.1109/TASSP.1984.1164433).

[7] N. Petra, *et al.*: "A novel architecture for Galois Fields $GF(2^m)$ multipliers based on mastrovito scheme," IEEE Trans. Comput. **56** (2007) 1470 (DOI: 10.

DOI: 10.1587/elex.13.20160729
Received July 24, 2016
Accepted September 6, 2016
Publicized September 20, 2016
Copyedited October 10, 2016

1109/TC.2007.70741).

[8] A. Hosangadi, *et al.*: "Simultaneous optimization of delay and number of operations in multiplierless implementation of linear systems," Proc. IWLS (2005) 1.

[9] A. Chandrakasan, *et al.*: "Optimizing power using transformations," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **14** (1995) 12 (DOI: 10.1109/43.363126).

[10] C. Paar: "Optimized arithmetic for Reed-Solomonen coders," Proc. IEEE Int. Sym. Information Theory (1997) 250 (DOI: 10.1109/ISIT.1997.613165).

[11] N. Chen and Z. Yan: "Cyclotomic FFTs with reduced additive complexities based on a novel common subexpression elimination algorithm," IEEE Trans. Signal Process. **57** (2009) 1010 (DOI: 10.1109/TSP.2008.2009891).

[12] R. Pasko, *et al.*: "A new algorithm for elimination of common subexpressions," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **18** (1999) 58 (DOI: 10.1109/43.739059).

1 Introduction

Chien search is an important process in Reed-Solomon (RS) decoder. To increase the decoding throughput, the decoders are required to be implemented by parallel architecture, but the parallel realization of Chien search occupies the most area in the overall decoders [1, 2, 3, 4, 5]. A typical p-parallel Chien search architecture is shown in Fig. 1.

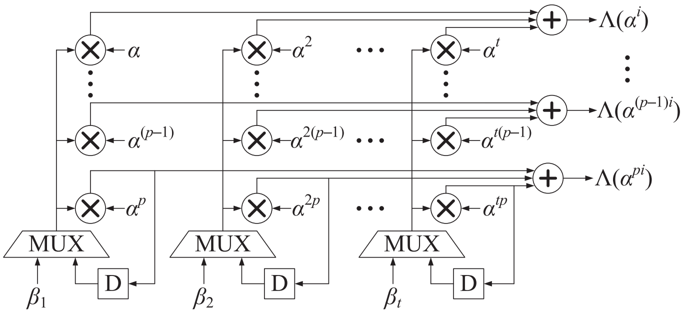

Fig. 1. Typical p-parallel Chien search architecture

In the architecture, p is called parallel factor, and $\Lambda(\alpha^i)$ is called error-locator polynomial which can be expressed as [1]:

$$\Lambda(\alpha^i) = \sum_{j=1}^{t} \alpha^{ij}\beta_j + 1 \tag{1}$$

where α^{ij} is the constant elements, β_j is the input element. In constant finite field multipliers (CFFMs) $\alpha^{ij}\beta_j$, only XOR gates are required [1]. Sharing common subexpressions (CSs) in the logic expressions of CFFMs can reduce the total gates in the hardware implementations, and common subexpressions elimination (CSE) algorithm is an effective algorithm to explore the CSs in the logic expressions. How to select the CSs to achieve the minimum area is an NP-complete problem [6]. Different CS-selecting strategies are proposed for Chien search implementations. However, these strategies proposed in previous works only focus on the area reduction without taking the delay into consideration.

DOI: 10.1587/elex.13.20160729
Received July 24, 2016
Accepted September 6, 2016
Publicized September 20, 2016
Copyedited October 10, 2016

In this paper, not only area but also delay, which is an important parameter in the high-speed applications, is taken into consideration in CS selection. As the delay of a circuit is mainly related with adopted structure, we first introduce the shortest path structures for the XOR network circuits. Based on the shortest path structures, we prove that sharing CSs will increase the delay of the XOR networks. Then a sufficient condition to keep the shortest path unchanged is deduced, and a shortest-path-keep CSE (SPK-CSE) algorithm is proposed based on the sufficient condition. In the last, the parallel Chien search in RS$(255, 239)$ code is implemented with the proposed SPK-CSE algorithm, and the implementation achieves the minimal delay.

2 Shortest path structures

For a N inputs XOR network, suppose the delays of inputs $\{x_0, x_1, \ldots, x_{N-1}\}$ is $\{d_0, d_1, \ldots, d_{N-1}\}$, it has been proven in [7] that the minimal delay $t_{\min}$ for the XOR network is given by

$$t_{\min} = \left\lceil \log_2 \sum_{i=0}^{N-1} 2^{d_i} \right\rceil \tag{2}$$

The units of both $\{d_0, d_1, \ldots, d_{N-1}\}$ and $t_{\min}$ are T_{XOR}, where T_{XOR} denotes the normalized delay of a XOR gate. It also has been proven in [7] that the XOR network constructed with delay-driven-binary-tree (DDBT) structure can achieve the minimal delay given by (2). The method to construct the DDBT structure is shown in Fig. 4 in [7]. At each iteration of the constructing algorithm, two elements x_i and x_j with the minimum delay in the signal-set S, which is initialized by being composed of all input signals $\{x_0, x_1, \ldots, x_{N-1}\}$, are taken to construct the circuit with an XOR gate, and the output signal $x_k = x_i + x_j$ is inserted into the S.

Suppose all delays of inputs are zeros, then (2) can rewrite as

$$t_{\min} = \lceil \log_2 N \rceil \tag{3}$$

where N is the number of inputs. In this case, fast-binary-tree (FBT) structure can achieve the minimal delay given by (3) [8, 9]. Obviously, the FBT structure is a special form of the DDBT structure. In the FBT structure constructing process, the delays are not required taking into considerations. First, every two signals in set S_0, which consists of all input signals $\{x_0, x_1, \ldots, x_{N-1}\}$, are taken to construct the circuit with an XOR gate, and the output signal of XOR gate is inserted into a new set S_1. If the number of signals N in S is odd, the last signal x_{N-1} is also inserted into S_1. Then, the signals in S_1 are taken to construct the circuits in the same way, until there is only one signal in the new set.

It can be concluded from (3) that the number of inputs N satisfies $N \le 2^t$, where t is the delay of the XOR network. In this paper, if $N = 2^t$, then the FBT structure is named as full-tree FBT structure, otherwise, the FBT structure is named non-full-tree FBT structure.

Let us take an example to illustrate the shortest path structures. Suppose the delays of inputs $\{x_4, x_3, x_2, x_1, x_0\}$ are $\{1, 2, 0, 2, 1\}T_{\text{XOR}}$, respectively, the minimal delay of the XOR network is $4T_{\text{XOR}}$ according to (2). As shown in Fig. 2(a), the DDBT-based XOR network can achieve the minimal delay. If all delays of inputs are zeros, FBT-based XOR network can achieve the minimal delay in this case,

DOI: 10.1587/elex.13.20160729
Received July 24, 2016
Accepted September 6, 2016
Publicized September 20, 2016
Copyedited October 10, 2016

ELectronics EXpress

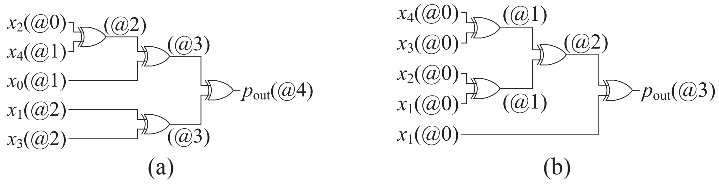

Fig. 2. Diagrams of shortest path structures: (a) DDBT structure; (b) FBT structure

as shown in Fig. 2(b), and the delay of the FBT-based XOR network is $3T_{\text{XOR}}$ according to (3). The FBT structure in Fig. 2(b) is a non-full-tree FBT structure.

3 SPK-CSE algorithm

Suppose the delays of inputs in a CFFM can be ignored, then the CFFM constructed with FBT structure can achieve the shortest path in the direct implementation. After sharing CSs, as the delays of CSs are not the same, in this case, the CFFM constructed with the DDBT structure can achieve the shortest path. The influence of sharing CSs on delay of an XOR network in the CFFM is given by the following theorem.

Theorem 1. *For an XOR network in CFFM, let T_d to denote the minimal output delay of direct implementation and T_o to denote the minimal output delay of CSE-optimized implementation, then the following inequality holds true:*

$$T_o \geq T_d \tag{4}$$

Proof. Consider an N inputs XOR network $p_d = x_{N-1} + \ldots + x_1 + x_0$, suppose it includes m original inputs and n CSs after sharing CSs, i.e., $p_o = (x_{m-1} + \ldots + x_1 + x_0) + (c_{n-1} + \ldots + c_1 + c_0)$, then $N = m + \sum_{j=1}^{n} N_j$, where N_j is the number of inputs in the CS c_j. Therefore, T_d and T_o are calculated in the following way:

$$\begin{cases} T_d = \lceil \log_2 N \rceil \\ T_o = \left\lceil \log_2 \left(m + \sum_{j=1}^{n} 2^{t_j} \right) \right\rceil \end{cases} \tag{5}$$

where t_j is the delay of CS c_j. The CS c_j is also generated by a XOR network, and the FBT structure in CS c_j may be destroyed, then t_j satisfies $t_j \geq \lceil \log_2 N_j \rceil$, and it can be further obtained that $2^{t_j} \geq 2^{\lceil \log_2 N_j \rceil} \geq N_j$. Furthermore, it can be deduced that

$$\left(m + \sum_{j=1}^{n} 2^{t_j} \right) \geq \left(m + \sum_{j=1}^{n} N_j \right) \tag{6}$$

Inequality (4) holds true according to (5) and (6). □

According to Theorem 1, sharing CSs may increase the delay. In the following, we give a sufficient condition to keep the shortest path unchanged.

Corollary 1. *A sufficient condition for $T_o = T_d$ is that any CS s_j is constructed by full-tree FBT structure.*

Proof. In the circuit of CS s_j with full-tree FBT structure, $N_j = 2^{t_j}$, then inequality (6) satisfies the equality condition, and then $T_o = T_d$. According to (5), $T_o = T_d$ can hold true even if inequality (6) does not satisfy the equality condition, therefore, Corollary 1 is only sufficient condition but not a necessary condition. □

DOI: 10.1587/elex.13.20160729
Received July 24, 2016
Accepted September 6, 2016
Publicized September 20, 2016
Copyedited October 10, 2016

Based on Corollary 1, we propose a SPK-CSE algorithm, which only extract the CSs that satisfy the Corollary 1. The proposed SPK-CSE is described in Fig. 3.

```
let $S$ be a signal set, and $S$ is initialized with all input signals $\{x_0, x_1, ..., x_{N-1}\}$:
  1) take two any signals $x_i$ and $x_j$ from set $S$ to compose a CS $c_k$=$x_i$+$x_j$;
  2) calculate the occurrence frequency $f_k$ of the CS $c_k$ in the logic expressions;
  3) repeat Steps 1-2 until all the signal-combinations in $S$ are checked.
  4) find the highest occurrence frequency $f_{\max}$;
  5) if $f_{\max}$>1, execute Steps 6-8, otherwise execute Steps 9;
  6) select a CS $c_n$ with the highest occurrence frequency $f_{\max}$ randomly;
  7) replace the selected CS in the logic expressions with the signal $c_n$;
  8) insert the CS signal $c_n$ into a new set $S_{\text{new}}$, and go to Step 1;
  9) if number of signals in $S_{\text{new}}$ is more than 1, replace $S$ with $S_{\text{new}}$ to repeat Steps 1-8;
  10) stop the algorithm.
```

Fig. 3. The SPK-CSE algorithm

4 An example

In this section, we take a CFFM to illustrate the influence of sharing CSs on the delays, and to evaluate our SPK-CSE. The example of CFFM is

$$\begin{cases} p_7 = x_7 + x_3 + x_2 + x_0 \\ p_6 = x_7 + x_6 + x_3 + x_2 + x_1 + x_0 \\ p_5 = x_5 + x_3 + x_0 \\ p_4 = x_7 + x_6 + x_3 + x_2 + x_1 + x_0 \\ p_3 = x_4 + x_3 + x_2 + x_0 \\ p_2 = x_7 + x_6 + x_5 + x_4 + x_3 + x_2 + x_1 + x_0 \\ p_1 = x_7 + x_3 + x_2 + x_1 + x_0 \\ p_0 = x_7 + x_4 + x_1 + x_0 \end{cases} \tag{7}$$

Suppose the delays of inputs can be neglected, then in the direct implementations, the circuits of $p_7{\sim}p_0$ constructed with FBT structure achieve the shortest path, and the delays are listed in Table I, which are calculated according to (3).

We take the CSE algorithm proposed in [10] (notes as CSE-[10]) to eliminate CSs in the logic expressions. The CSE-[10] algorithm takes the two-term CS with the highest occurrence frequency to be eliminated iteratively. After optimized by using CSE-[10] algorithm, total six CSs are eliminated. The eliminated CSs are $\{c_0 = x_3 + x_0,\ c_1 = c_0 + x_2,\ c_2 = c_1 + x_7,\ c_3 = c_2 + x_1,\ c_4 = c_3 + x_6,\ c_5 = c_4 + x_4\}$, the CSs are formed in overlapping way. The delays of CSs are $\{1, 2, 3, 4, 5, 6\}T_{\text{XOR}}$, respectively. After sharing CSs, the output p_k not only contain the original inputs x_i, but also contain the CSs c_j, therefore the circuits should be constructed with DDBT structure that can achieve the minimal delay. The delays of outputs are also listed in Table I. All output delays have been increased, except for p_5 and p_0, and the critical path delay (CPD) is also increased.

Four CSs are eliminated by using the proposed SPK-CSE algorithm, the eliminated CSs are $\{c_0 = x_3 + x_0,\ c_1 = x_7 + x_1,\ c_2 = x_4 + x_2,\ c_3 = c_1 + c_0\}$. In the first loop, SPK-CSE eliminates the CSs that only contain the original inputs $x_7{\sim}x_0$. The eliminated CSs include $\{c_0, c_1, c_2\}$, and these CSs form a new set S_1. In the second loop, the CSs that only contain the signal in S_1 are eliminated. Only CS c_3 is eliminated in this loop. The circuits of the eliminated CSs can be constructed with full-tree FBT structure. The delays of CSs are $\{1, 1, 1, 2\}T_{\text{XOR}}$, respectively. The circuits of $p_7{\sim}p_0$ are also constructed with DDBT structure, and the delays of

DOI: 10.1587/elex.13.20160729
Received July 24, 2016
Accepted September 6, 2016
Publicized September 20, 2016
Copyedited October 10, 2016

the circuits are also listed in Table I. All delays are kept unchanged after sharing CSs, therefore, the CPD is also kept unchanged, and this is the shortest CPD achieved after sharing CSs according to Theorem 1.

Table I. The hardware complexities of different implementations for the CFFM

Implementation	Area (A_{XOR})	Delays (T_{XOR})								
		p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0	CPD
Direct	33	2	3	2	3	2	3	3	2	3
CSE-[10]	12	3	6	2	5	3	7	4	2	7
SPK-CSE	18	2	3	2	3	2	3	3	2	3

5 Implementations

In this section, low-delay 4-parallel Chien search architecture for RS(255, 239) code, which is used to correct burst errors in optical fiber submarine cable systems [1], is implemented with the proposed SPK-CSE algorithm. In the low-delay implementation, the logic expressions of CFFMs are first optimized by using SPK-CSE algorithm, and then the circuits of CFFMs are constructed with DDBT structure. As the CSs not only exist in a CFFM but also exist among the CFFMs, our SPK-CSE algorithm is combined with group-optimized scheme used in [1, 2] to eliminate the CSs among CFFMs. The hardware complexities of the low-delay implementation with SPK-CSE are listed in Table II.

Table II. Comparisons of SPK-CSE-based low-delay Chien search architecture with other CSE-based implementations

	Direct	CSE-based implementations				
		[1]	[11]	[10]	[12]	Our
Area (A_{XOR})	557	266	252	271	263	301
CPD (T_{XOR})	3	5	6	4	5	3

Our low-delay Chien search architecture is compared with other classic CSE-based implementations in Table II. For fair comparison, other implementations are also constructed with DDBT-structures. In [1], the CSs are selected by iterative matching (IM) strategy, and the IM strategy was further developed in [11] by combining with cancelation property of modulo 2 additions. As shown in Table II, the developed IM (DIM) strategy is more efficient in area reduction, but the efficiency is gained at the expense of increasing delay. In [2, 3, 4, 5], two-term CSs with the highest occurrence frequency are selected to be eliminated, and this strategy was first proposed in [10]. A similar strategy is proposed in [12], which take the most-term with the highest occurrence frequency CSs to be eliminated. However, these strategies do not take the delay into consideration, therefore, the CPD of implementations based on these CSE algorithm are increase. Although the implementation with our SPK-CSE has more area cost, but it can achieve the shortest delay after sharing CSs.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61376025), and the Natural Science Foundation of Jiangsu Province (No. BK20160806).

DOI: 10.1587/elex.13.20160729
Received July 24, 2016
Accepted September 6, 2016
Publicized September 20, 2016
Copyedited October 10, 2016