Two-parallel Reed-Solomon based FEC architecture for optical communications

Seungbeom Lee, Chang-Seok Choi, and Hanho Lee^{a)}

*School of Information and Communication Engineering, Inha University,
Incheon, 402–751, Republic of Korea*

a) hhlee@inha.ac.kr

Abstract: This paper presents a high-speed Forward Error Correction (FEC) architecture based on two-parallel Reed-Solomon (RS) decoder for 10 and 40-Gb/s optical communication systems. A high-speed two-parallel RS(255, 239) decoder has been proposed and the derived structure can also be applied to implement the 10 and 40-Gb/s RS FEC architectures. The implementation results show that 16-Ch. RS FEC architecture can operate at a clock frequency of 160 MHz and has a throughput of 41 Gb/s for the Xilinx Virtex4 FPGA. Also, RS FEC operates at a clock frequency of 400 MHz and has a throughput of 102 Gb/s for 0.18- μ m CMOS technology.

Keywords: Reed-Solomon code, forward error correction, two-parallel

Classification: Integrated circuits

References

- [1] “Forward Error Correction for Submarine Systems,” Telecommunication Standardization Section, International Telecom. Union, ITU-T Recommendation G.975, Oct. 2000.
- [2] L. Song, M.-L. Yu, and M. S. Shaffer, “10 and 40-Gb/s Forward Error Correction Devices for Optical Communications,” *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1565–1573, Nov. 2002.
- [3] H. M. Shao, T. K. Truong, L. J. Deutsch, J. H. Yuen, and I. S. Reed, “A VLSI Design of a Pipeline Reed-Solomon Decoder,” *IEEE Trans. Comput.*, vol. C-34, no. 5, pp. 393–403, May 1985.
- [4] H. Lee, “High-Speed VLSI Architecture for Parallel Reed-Solomon Decoder,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 11, no. 2, pp. 288–294, April 2003.
- [5] S. Lee, H. Lee, J. Shin, and J.-S. Ko, “A High-Speed Pipelined Degree-Computationless Modified Euclidean Algorithm Architecture for Reed-Solomon Decoders,” *2007 IEEE International Symposium on Circuits and Systems (ISCAS 2007)*, pp. 901–904, May 2007.

1 Introduction

Reed-Solomon (RS) codes have been widely used in a variety of communication systems such as space communication links, digital subscriber loops,

and wireless systems, as well as in networking communications [1, 2, 3, 4, 5]. Due to increasing demand for high capacity of optical communications, the high-speed low-power implementations of RS decoders are desirable to meet higher data rate for system-level integration. An 8-byte error-correcting RS code is recommended by ITU-T for optical submarine systems [1].

The very high-speed data transmission techniques that have been developed for the fiber optical networking systems have necessitated the implementation of high-speed FEC architectures to meet the continuing demands for ever higher data rates [2, 4]. Currently, the RS(255, 239) code is commonly used in high-speed (40-Gb/s and beyond) fiber optic systems. However, as the data rates approach 40-Gb/s and beyond, all existing RS decoders using a systolic-array structure [2, 3, 4] cause relatively huge hardware complexity and power consumption, which cause difficulties in system-level integration.

In this paper, we present two-parallel RS decoder architecture and high-speed low-complexity 10 and 40-Gb/s Reed-Solomon based FEC (RS FEC) architectures on Xilinx Virtex4 FPGA. Also, we will describe the key ideas applied to two-parallel RS FEC design, especially those for achieving high throughput and reducing complexity.

2 Two-parallel Reed-Solomon decoder

2.1 Two-parallel Syndrome computation block

The syndrome computation block calculates all the syndromes S_i ($0 \leq i \leq 15$) by putting the roots of generator polynomial $G(x)$ into the received codeword polynomial $R(x)$. As shown in Fig. 1 (a), proposed two-parallel syndrome computation block is implemented by following Eq. (4).

$$R(x) = r_{254}x^{254} + r_{253}x^{253} + \cdots r_1x + r_0 \quad (1)$$

$$G(x) = (x - \alpha^0)(x - \alpha^1) \cdots (x - \alpha^{14})(x - \alpha^{15}) \quad (2)$$

$$S_i = R(\alpha^i) = r_{254}(\alpha^i)^{254} + r_{253}(\alpha^i)^{253} + \cdots r_1(\alpha^i) + r_0 \quad (3)$$

$$S_i = R(\alpha^i) = ((\cdots (0 + r_{254}\alpha^i)\alpha^i + r_{253}\alpha^i + r_{252})\alpha^i + r_{251}\alpha^i + r_{250})\alpha^i + \cdots r_1\alpha^i + r_0 \quad (4)$$

The input patterns of the two-parallel syndrome computation cell without a dummy zero input symbol are shown in Fig. 1 (a). The input B is delayed one clock cycle to compute $r_{254}(\alpha^i)^2 + r_{253}\alpha^i + r_{252}$ at the same clock cycle. At the first clock, r_{254} is stored in flip-flop(5) by multiplexer select signal (MSS)(2) and then $r_{254}(\alpha^i)^2 + r_{253}\alpha^i + r_{252}$ is computed at the next clock cycle. The input r_{254} is the first input symbol of the new received codeword at the 128th clock. Syndromes S_i is stored in flip-flop(5) and the first symbol is stored in flip-flop(4) at the 128th clock. MSS(3) becomes 1 to compute $r_{254}(\alpha^i)^2 + r_{253}\alpha^i + r_{252}$ at the 129th clock. After the 128th clock, MSS(1) becomes 1 to multiply the input A by α^i . Two codeword are processed during one iterative cycle. When MSS(7) is 1, syndromes S_i is stored in buffer register(8) and MSS(7) becomes 1 at 1st clock and 129th clock. When MSS(6) is 1, syndromes S_i is shifted to output.

Fig. 1. (a) Proposed two-parallel syndrome computation block, and (b) two-parallel Chien search and Forney algorithm block.

2.2 Key equation solver block

The pipelined degree-computationless modified Euclidean (p DCME) algorithm and architecture [5] is used to obtain the error locator polynomial $\sigma(x)$ and the error value polynomial $\omega(x)$ by solving the key equation $\omega(x) = S(x)\sigma(x) \bmod x^{2t}$. The detailed explanation of p DCME algorithm and architecture was addressed in our previous paper [5].

2.3 Two-parallel Chien search and Forney algorithm block

The error locator polynomial $\sigma(x)$ and error value polynomial $\omega(x)$ are obtained by the KES block. Let $X_l = \alpha^{m_l}$ and $Y_l = e_{m_l}$. Eq. (5) is transformed to Eq. (6), where X_l and Y_l are the possible error location and the possi-

ble error value, respectively. “ $\sigma(\alpha^1) = 0$ ” means that r^{254} is corrupted by error. At first α^1 is putted into $\sigma(x)$ because the first symbol of received codeword is r^{254} . In Eq. (10), $\sigma'(x)$ is the derivative of $\sigma(x)$. Rewriting $\sigma(x)$ as the sum of the even terms $\sigma_{\text{even}}(x)$ and the odd terms $\sigma_{\text{odd}}(x)$, we have $\sigma_{\text{odd}}(x) = x \cdot \sigma'(x)$. Therefore, the Chien search and Forney algorithm block is implemented as shown in Fig. 1 (b). In Eq. (10), dividing operation is implemented by 256×8 ROM in which the inverse of field elements are stored. First, conventional two-parallel Chien search block computes $\sigma(\alpha^1)$ and $\sigma(\alpha^2)$. And then $\sigma(\alpha^{255})$ and $\sigma(\alpha^{256})$ are outputted after 128 clock cycle. In this case the $\sigma(\alpha^{256})$ is a dummy symbol.

$$S_i = r(\alpha^i) = e(\alpha^i) = \sum_{l=1}^v e_{m_l} \alpha^{m_l \cdot i} \quad (5)$$

$$S(x) = \sum_{i=0}^{15} S_i x^i = \sum_{i=0}^{15} \sum_{l=1}^v Y_l X_l^i x^i \quad (6)$$

$$\sigma(x) = (1 - xX_1)(1 - xX_2) \cdots (1 - xX_v) \quad (7)$$

$$\omega(x) = S(x) \cdot \sigma(x) \bmod x^{2t} \quad (t = 8) \quad (8)$$

$$= \sum_{l=1}^v Y_l \cdot \prod_{n=1, n \neq l}^v (1 - xX_n) \quad (9)$$

$$Y_l = \omega(X_l^{-1}) / ((-X_l^{-1}) \cdot \sigma'(X_l^{-1})) \quad (10)$$

The Chien search block should calculate $\sigma(\alpha^1)$ not $\sigma(\alpha^{256})$ at the 128th clock. $\sigma(\alpha^1)$ and $\sigma(\alpha^2)$ are calculated at the first clock cycle because both of MSS(2) and MSS(3) are 1, and then $\sigma(\alpha^{253})$ and $\sigma(\alpha^{254})$ are calculated at 127th clock. At the next clock cycle, $\sigma(\alpha^{255})$ is obtained when MSS(2) is 0, and the new $\sigma(\alpha^1)$ is obtained when MSS(3) is 1. When both of MSS(2) and MSS(3) are all zero, MSS(1) can be either 0 or 1.

3 10 and 40-Gb/s RS FEC architectures with two-parallel processing

Fig. 2 shows a proposed architecture and a timing chart for two-parallel 4-channel RS FEC architecture. The KES block [5] accepts the syndromes every 18 clock cycles, whereas two-parallel syndrome computation block accepts the received codeword every 127 or 128 clock cycles. Therefore, syndrome generation and application of correction have to be instantiated independent for each of the four decoding channels, while the KES block can be shared between all channels. As seen from the resulting timing shown in Fig. 2, the sharing of the p DCME algorithm block requires phase shifting between each single channel. The syndrome computation block provides $2t$ syndromes after the processing delay of 128 clock cycles required for computing the syndrome polynomial. Since four syndrome computation blocks are connected by one KES block, $2t$ syndrome values are enter into the KES block sequentially. After 128 clock cycles, KES block outputs the $\sigma(x)$ and $\omega(x)$ polynomials in parallel feeding to the Chien search block. Error polynomial buffer transforms serial output from KES block into parallel input. The proposed RS

Fig. 2. (a) Block diagram, and (b) timing chart of two-parallel 4-Ch. RS FEC architecture.

FEC continuously takes in code blocks, performs the appropriate coding operation, and outputs the data with a fixed latency of 260 clock cycles.

The proposed two-parallel RS decoder is used to design 40-Gb/s 16-Ch. RS FEC architecture. The block diagram of the two-parallel 40-Gb/s 16-Ch. RS FEC architecture consists of four two-parallel 4-Ch. RS FEC architectures.

4 Result and comparison

The proposed two-parallel 10 and 40-Gb/s RS FEC architectures were modeled in Verilog HDL and then were synthesized for the Xilinx Virtex4 FPGA implementation. The total number of gates is 434,800 from the synthesized results excluding the FIFO. The 16-Ch. RS FEC architecture can operate at a clock frequency of 160 MHz and has a data throughput rate of 41-Gb/s for the Xilinx Virtex4 FPGA implementation. Also, the RS FEC can operate at a clock frequency of 400 MHz and has a data processing rate of 102-Gb/s in 0.18- μ m CMOS technology.

Table I compares the performance of the several RS FEC architectures for high-data rates. Sharing of the KES block among 4-Ch. n -parallel RS decoder leads to substantial hardware savings as the KES block requires about 60 ~ 80% of total gates of RS decoder. Also, parallel RS decoder reduces latency and enhances data throughput. Proposed two-parallel RS FEC is implemented by p DCME algorithm block, which reduces the hardware com-

Table I. Implementation result of the 16-Ch. RS FEC architecture.

Design	Proposed Two-Parallel		Three-Parallel [2]	Serial [5]
Syndrome	100,800		40,000	60,000
KES	156,000		84,000	280,000
Chien, Error	178,000		240,000	132,000
Total # of Gates	434,800		364,000	472,000
Clock Rate (MHz)	400	160	112	625
Latency	260 clocks (650 ns)	260 clocks (1.6 μ s)	168 clocks (1.5 μ s)	355 clocks (568 ns)
Throughput (Gb/s)	102	41	43	80
Technology	0.18 μ CMOS 1.8 V	Xilinx Virtex4 FPGA	0.16 μ m CMOS 1.5 V	0.13 μ m CMOS, 1.2 V

plexity of KES block. Generally two-parallel RS(255, 239) decoder requires 1 byte dummy zero input because codeword size is odd. However, the proposed architecture does not require 1 byte dummy input and it has no loss of code rates. The proposed two-parallel RS FEC has a much higher data processing rate compared with the conventional serial and three-parallel RS FEC architectures, and low hardware complexity compared with the serial RS FEC architecture.

5 Conclusion

This paper presents the design and implementation of two-parallel RS decoder, and 10 and 40-Gb/s RS FEC architecture for optical communications. Two-parallel processing is used to achieve 40-Gb/s throughput on the Xilinx FPGA implementation. A high-speed low-complexity p DCME algorithm block is applied to the RS decoder. Two-way parallelizing for syndrome computation and error correction allow the inputs to be received and error correction allow the inputs to be received at very high fiber optic rates and the outputs to be delivered at correspondingly high rates with a minimum delay. Resource sharing is used to reduce the hardware complexity. As a result, the proposed 40-Gb/s RS FEC using two-parallel processing has a hardware complexity that is comparable to a previously published 40-Gb/s RS FEC design. The proposed two-parallel RS FEC has potential applications in the next generation FEC devices for optical communications with a data rate of 40-Gb/s and beyond.

Acknowledgments

This work was partly supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC support program supervised by the IITA (IITA-2008-C1090-0801-0019) and by grant No. R01-2006-000-10596-0 from the Basic Research Program of the Korea Science & Engineering Foundation.