논문 2009-46SD-11-7

100*Gb/s*급 광통신시스템을 위한 3-병렬 Reed-Solomon 기반 FEC 구조 설계

(Three-Parallel Reed-Solomon based Forward Error Correction Architecture for 100*Gb/s* Optical Communications)

최 창 석*, 이 한 호**

(Chang-Seok Choi and Hanho Lee)

요 약

본 논문에서는 차세대 100-*Gb/s*급 광통신 시스템을 위한 3-병렬 Reed-Solomon (RS) 디코더 기반의 고속 Forward Error Correction (FEC) 구조를 제안한다. 제안된 16채널 RS기반 FEC 구조는 4개의 신드롬 계산 블록이 1개의 Key Equation Solver (KES) 블록을 공유하는 3-병렬 4채널 RS 기반 FEC 구조 4개로 구성되어 있다. 제안하는 100-*Gb/s* RS 기반 FEC는 1.2*V*의 공급전압의 0.13*μm* CMOS 공정을 이용하여 구현하였다. 구현 결과 제안된 RS기반 FEC 구조는 300*MHz*의 동작 주파수에서 115-*Gb/s* 의 데이터 처리율을 가지며, 기존의 RS 기반 FEC 구조에 비해 높은 데이터 처리율과 낮은 하드웨어 복잡도를 보여주고 있다.

Abstract

This paper presents a high-speed Forward Error Correction (FEC) architecture based on three-parallel Reed-Solomon (RS) decoder for next-generation 100-*Gb/s* optical communication systems. A high-speed three-parallel RS(255,239) decoder has been designed and the derived structure can also be applied to implement the 100-*Gb/s* RS-FEC architecture. The proposed 100-*Gb/s* RS-FEC has been implemented with 0.13-μm CMOS standard cell technology in a supply voltage of 1.2V. The implementation results show that 16-Ch. RS-FEC architecture can operate at a clock frequency of 300MHz and has a throughput of 115-*Gb/s* for 0.13-μm CMOS technology. As a result, the proposed three-parallel RS-FEC architecture has a much higher data processing rate and low hardware complexity compared with the conventional two-parallel, three-parallel and serial RS-FEC architectures.

Keywords: Reed-Solomon code, forward error correction, architecture, optical communications, CMOS.

I. INTRODUCTION

Driven by high-definition video and the penetration of high-speed broadband access, the increasing amount of consumer IP traffic will bolster the overall IP growth rate so that it sustains a fairly steady growth rate, e.g., nearly doubling the IP traffic in Korea from 2005(424.5 *Gb/s*) to 2007(721.7 *Gb/s*).

The *IEEE* 802.3ba task force is currently discussing the use of 40*Gb/s* Ethernet (40*GbE*) as the next generation high speed interface for server and storage applications, and the use of 100*Gb/s* Ethernet (100*GbE*) for network aggregation applications[1]. In ITU-T SG15, a 100*GbE* optical transport network (OTN4) for 100*GbE* clients is also under discussion according to ITU-T

* 학생회원, ** 평생회원, 인하대학교 정보통신공학부
(School of Information and Communication Engineering, Inha University)
※ 이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2008-313-D00746)
접수일자: 2009년 7월29일, 수정완료일: 2009년10월26일

Recommendation G.709[2].

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[3~7]. The very high speed data transmission techniques that have been developed for the fiber optical networking systems have necessitated the implementation of high speed Forward Error Correction (FEC) architectures to meet the continuing demands for ever higher data rates[3~6]. 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[3~6] cause relatively huge hardware complexity and power consumption, which cause difficulties in system-level integration.

The RS decoder can be implemented using the modified Euclidean (ME) algorithm to solve a key equation. A syndrome based RS decoder consists of three components, which are syndrome computation block, key equation solver (KES) block and Chien search & error correction block[3~9].

In this paper, we present three parallel RS decoder architecture and the high speed low complexity 100*Gb/s* RS decoder based FEC (RS-FEC) architecture for next generation 100*Gb/s* Ethernet. Also, we will describe the key ideas applied to three parallel RS-FEC design, especially those for achieving high throughput and reducing complexity.

The remainder of the paper is organized as follows. Section II provides the three parallel RS encoder architecture. And Section III shows the proposed architecture for the three parallel RS decoder, especially three parallel syndrome computation block and Chien search & error correction block. Section IV describes the design of 100*Gb/s* RS-FEC architecture using three parallel processing. In Section V, implementation results and performance comparison are presented. Finally, conclusions are provided in Section VI.

II. THREE-PARALLEL REED-SOLOMON ENCODER

The systematic RS encoding can be described as equation (1), where $U(x)$ is encoded polynomial. The message polynomial $M(x)$ is multiplied by X^{n-k} after then added parity polynomial $P(x)$. The following $P(x)$ can be written as equation (2). To apply three parallel structure, the equation (2) is needed to modify. The $M(x)$ is not multiple of three, so we assumed one zero symbol is padded in front of M(x). As a result, three parallel based $P(x)$ can be described equation (3) and we can derive following partial generator polynomials as shown in equation (4).

$$U(x) = x^{n-k}m(x) + p(x) \tag{1}$$

$$p(x) = x^{n-k}m(x) \bmod g(x) \tag{2}$$

$$\begin{aligned} m(x)x^{16} \bmod g(x) = [\cdots\{[0 \bullet x^{18} + m_{238} \bullet x^{17} + m_{237} \bullet x^{16}) \bmod g(x) \bullet x^3 + \\ [(m_{236} \bullet x^{18} + m_{235} \bullet x^{17} + m_{234} \bullet x^{16}) \bmod g(x)]\} \bullet x^3 + \cdots] \\ \bullet x^3 + (m_2 \bullet x^{18} + m_1 \bullet x^{17} + m_0 \bullet x^{16}) \bmod g(x) \end{aligned} \tag{3}$$

$$\begin{aligned} g_0(x) = x^{16} \bmod g(x) \;\; g_1(x) = x^{17} \bmod g(x) \\ g_2(x) = x^{18} \bmod g(x) \end{aligned} \tag{4}$$

The proposed three parallel RS encoder is shown in Fig 1. The three parallel message symbol (M2, M1, M0) is inputted during 80 clock cycle and divided with each partial generator polynomial. Finally, the parity symbol is generated through

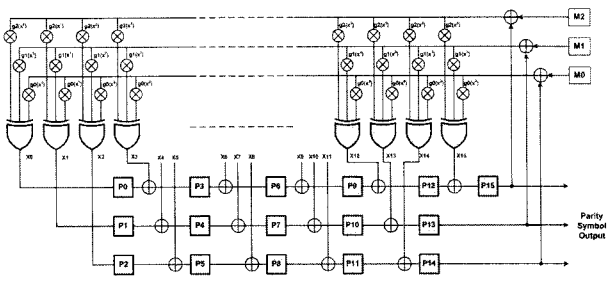


그림 1. 3-병렬 리드 솔로몬 부호기
Fig. 1. Three-parallel Reed-Solomon encoder.

표 1. 신드롬 연산 블록의 입력 패턴
Table 1. Input patterns of syndrome computation cell.

Clock	1st	2nd	···	85th	1th
InputA	r_{254}	r_{251}	···	r_2	r_{254}
InputB	r_{253}	r_{250}	···	r_1	r_{253}
InputC	r_{252}	r_{249}	···	r_0	r_{252}

Linear Feedback Shift Register (LFSR). In equation (3), the assumption was one zero symbol is padded, so that desired input pattern is[0, m238, m237],[m236, m235, m234]···[m2, m1, m0]. To solve this problem, D flip-flop was added after M2 input port, as a result the input symbol can be delayed by one clock cycle.

III. THREE-PARALLEL REED-SOLOMON DECODER

A. Syndrome Computation Block

The syndrome computation block calculates all the syndromes Si ($0 \le i \le 15$) by putting the roots of generator polynomial $G(x)$ into the received codeword polynomial $R(x)$. As shown in Fig. 2, proposed three parallel syndrome computation block is implemented by following equation (8).

$$R(x) = r_{254}x^{254} + r_{253}x^{253} + \cdots r_1 x + r_0 \quad (5)$$

$$G(x) = (x-\alpha^0)(x-\alpha^1)\cdots(x-\alpha^{14})(x-\alpha^{15}) \quad (6)$$

$$S_i = R(a^i) = r_{254}(\alpha^i)^{254} + r_{253}(\alpha^i)^{253} + \cdots r_1(\alpha^i) + r_0 \quad (7)$$

$$\begin{aligned} S_i = R(a^i) = ((\cdots(r_{254}(\alpha^i)^2 + r_{253}(\alpha^i)^1 + r_{252})(\alpha^i)^3 + \\ r_{251}(\alpha^i)^2 + r_{250}(\alpha^i)^1 + r_{249})(\alpha^i)^3 + \cdots)(\alpha^i)^3 \\ + r_2(\alpha^i)^2 + r_1(\alpha^i)^1 + r_0) \end{aligned} \quad (8)$$

The input patterns of the three parallel syndrome computation cell are shown in Table I. The received codeword is consists of 255 symbols which are multiple of 3, so that proposed syndrome computation block can process the syndromes during 85 clock cycles. At the first clock, the received codeword (r_{254}, r_{253},r_{252}) are inputted parallel, and then computes following partial syndromes $r_{254}(\alpha^i)^2 + r_{253}(\alpha^i)^1 + r_{252}$ stored in the flip-flop (1). At the next clock cycle, The flip-flop (1) is multiplied by $(\alpha^i)^3$, and then added with $r_{251}(\alpha^i)^2 + r_{250}(\alpha^i)^1 + r_{249}$. This iterative process will be performed during 85 clock cycles, syndromes Si is stored in flip-flop (1). Multiplexer selection (3) and (4) becomes 1 every 85^{th} clock cycle, to shift syndromes Si, to the flip-flop (2) and to compute new received codeword. Therefore the syndromes S_0, S_1,···,S_{15} are outputted serially to the KES block.

B. Key Equation Solver Block

Pipelined degree-computationless modified Euclidean (pDCME) algorithm and architecture[7] is used to obtain the error locator polynomial $\sigma(x)$ and the error value polynomial $\omega(x)$ by solving the key equation $\omega(x)=S(x)\sigma(x) \mod x^{2t}$. The detailed explanation of pDCME algorithm and architecture was addressed in our previous paper[7]. To minimize the hardware complexity, three-level pipelined processing element (PE) was used in pDCME architecture.

C. Three-Parallel Chien Search and Error Correction 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 $Y_l=e_{m_l}$. Equation (9) is transformed to equation (10), where X_l and Y_l are the possible error location and the possible error value, respectively. "$\sigma(\alpha^l)=0$" means that $r^{254}$is corrupted by error. At first α^l is putted into σ(x) because the first symbol of received codeword is r^{254}. In equation (13), $\sigma'(x)$ is the derivative of $\sigma(x)$. Rewriting $\sigma(x)$ as the sum of the even terms $\sigma_{even}(x)$and the odd terms $\sigma_{odd}(x)$, we have $\sigma_{odd}(x) = x \cdot \sigma'(x)$. Therefore, the proposed Chien search and Forney

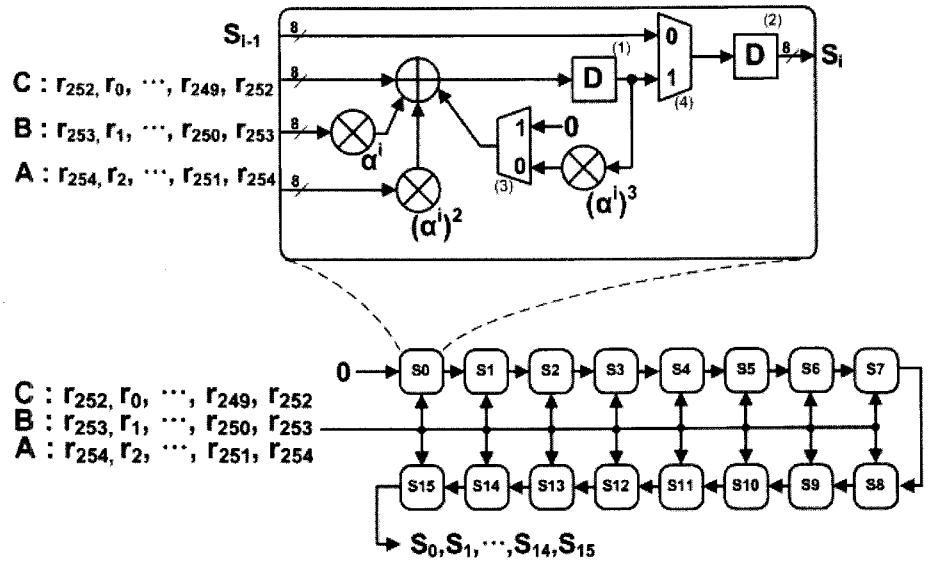


그림 2. 제안하는 3-병렬 신드롬 계산 블록
Fig. 2. Proposed three-parallel syndrome computation block.

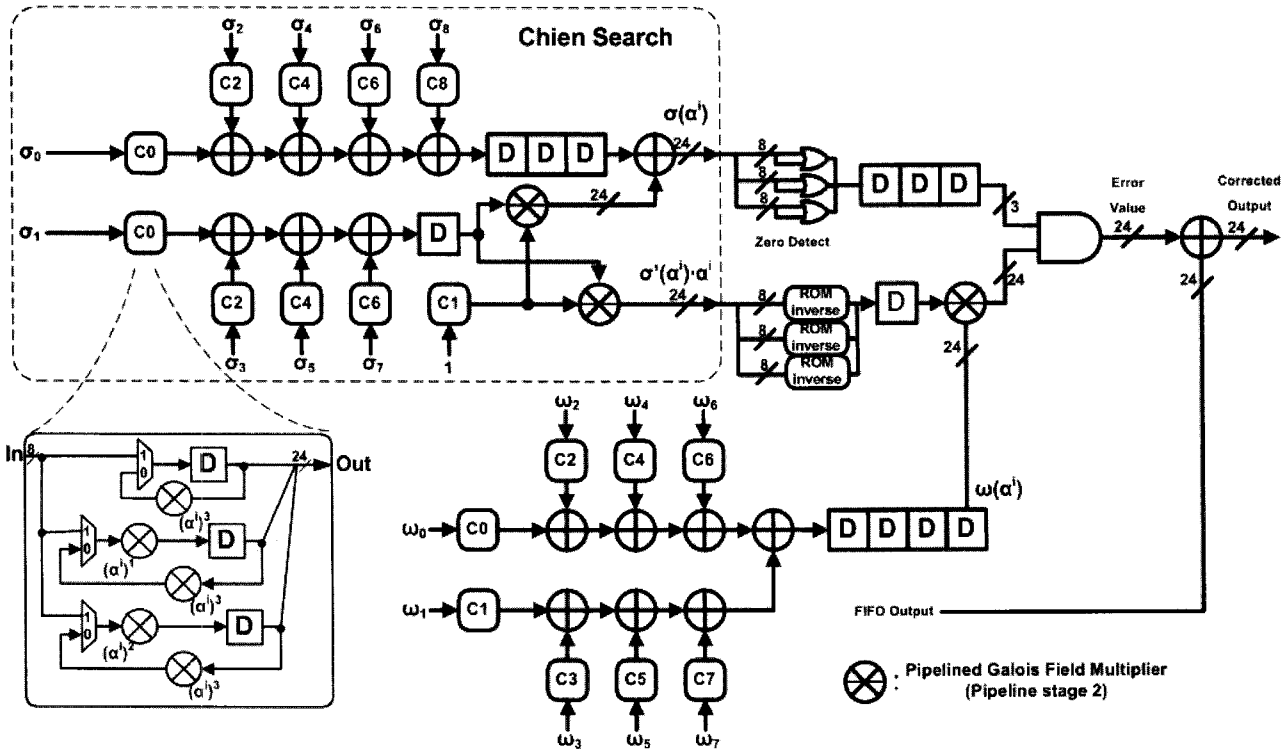


그림 3. 제안하는 3-병렬 Chien search 와 Forney 알고리즘 블록
Fig. 3. Proposed three-parallel Chien search and Forney algorithm block.

algorithm block is implemented as shown in Fig. 3. In equation (13), dividing operation is implemented by 256×8 ROM in which the inverse of field elements are stored.

$$S_i = r(\alpha^i) = e(\alpha^i) = \sum_{l=1}^{v} e_{m_l} \alpha^{m_l \cdot i} \tag{9}$$

$$S(x) = \sum_{i=0}^{15} S_i x^i = \sum_{i=0}^{15} \sum_{l=1}^{v} Y_l X_l{}^i x^i \tag{10}$$

$$\sigma(x) = (1 - xX_1)(1 - xX_2)\cdots(1 - xX_v) \tag{11}$$

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

$$= \sum_{l=1}^{v} Y_l \cdot \prod_{n=1, n\neq l}^{v} (1 - xX_n) \tag{12}$$

$$Y_l = \omega(X_l^{-1}) / ((-X_l^{-1})\cdot\sigma'(X_l^{-1})) \tag{13}$$

As shown in Fig. 3, serial Chien search cell is expanded into three parallel Chien search cell, because the following Chien search and Forney algorithm block should calculate three locations of error at each clock cycle. At the first clock $\sigma(\alpha^1)$, $\sigma(\alpha^2)$, $\sigma(\alpha^3)$ are calculated and at the last clock cycle $\sigma(\alpha^{253})$, $\sigma(\alpha^{254})$, $\sigma(\alpha^{255})$ are calculated during 85 clock cycles.

IV. 25 AND 100-GB/S RS-FEC ARCHITECTURE WITH THREE-PARALLEL PROCESSING

Fig. 4 and Fig. 5 show the proposed architecture and the timing chart for three parallel four channel RS-FEC architecture. The KES block[7] accepts the syndromes every 18 clock cycles, whereas three parallel syndrome computation block accepts the received codeword every 85 clock cycles. Therefore, syndrome generation and application of correction have to be instantiate 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. 5, the sharing of the pDCME algorithm requires phase shifting between each single channel.

The syndrome computation block provides $2t$ syndromes after the processing delay of 85 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. The KES block outputs the $\sigma(x)$ and $\omega(x)$ polynomials in parallel feeding to the

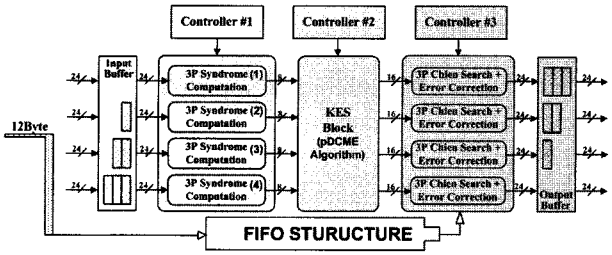


그림 4. 3-병렬 4-채널 RS기반 FEC 구조
Fig. 4. Block diagram of three-parallel 4-Ch. RS-FEC architecture.

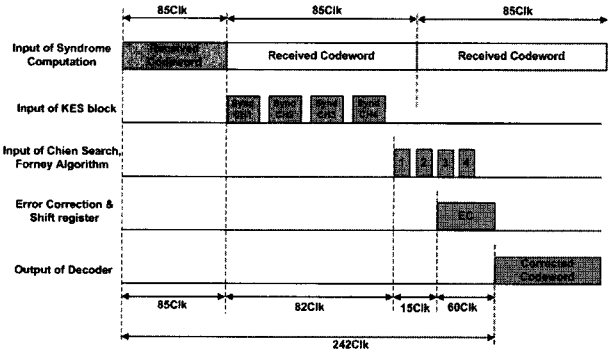


그림 5. 3-병렬 4-채널 RS 기반 FEC의 타이밍도
Fig. 5. Timing chart for three-parallel 4-Ch. RS-FEC architecture.

Chien search block after 82 clock cycles. The latency of KES block is $16t$, which is 80 clock cycles, and the latency of added in/out buffer is 2 clock cycles. The proposed RS-FEC continuously takes in code blocks, performs the appropriate coding operation, and outputs the data with a fixed latency of 242 clock cycles. Each channel has shift registers to synchronize the inputs and outputs. The three parallel 4-channel RS-FEC architecture is used to design *100Gb/s* 16-channel RS-FEC architecture. It achieves a payload throughput *115Gb/s* with an internal clock rate of *300MHz*.

The block diagram of the three parallel *100Gb/s* 16-channel RS-FEC architecture is shown in Fig. 6, which consists of 16 three parallel syndrome computation blocks, 4 shared KES blocks, 16 three parallel Chien search & error correction blocks and input/output buffers.

V. RESULTS AND COMPARISON

The proposed three parallel *100Gb/s* 16-Ch.

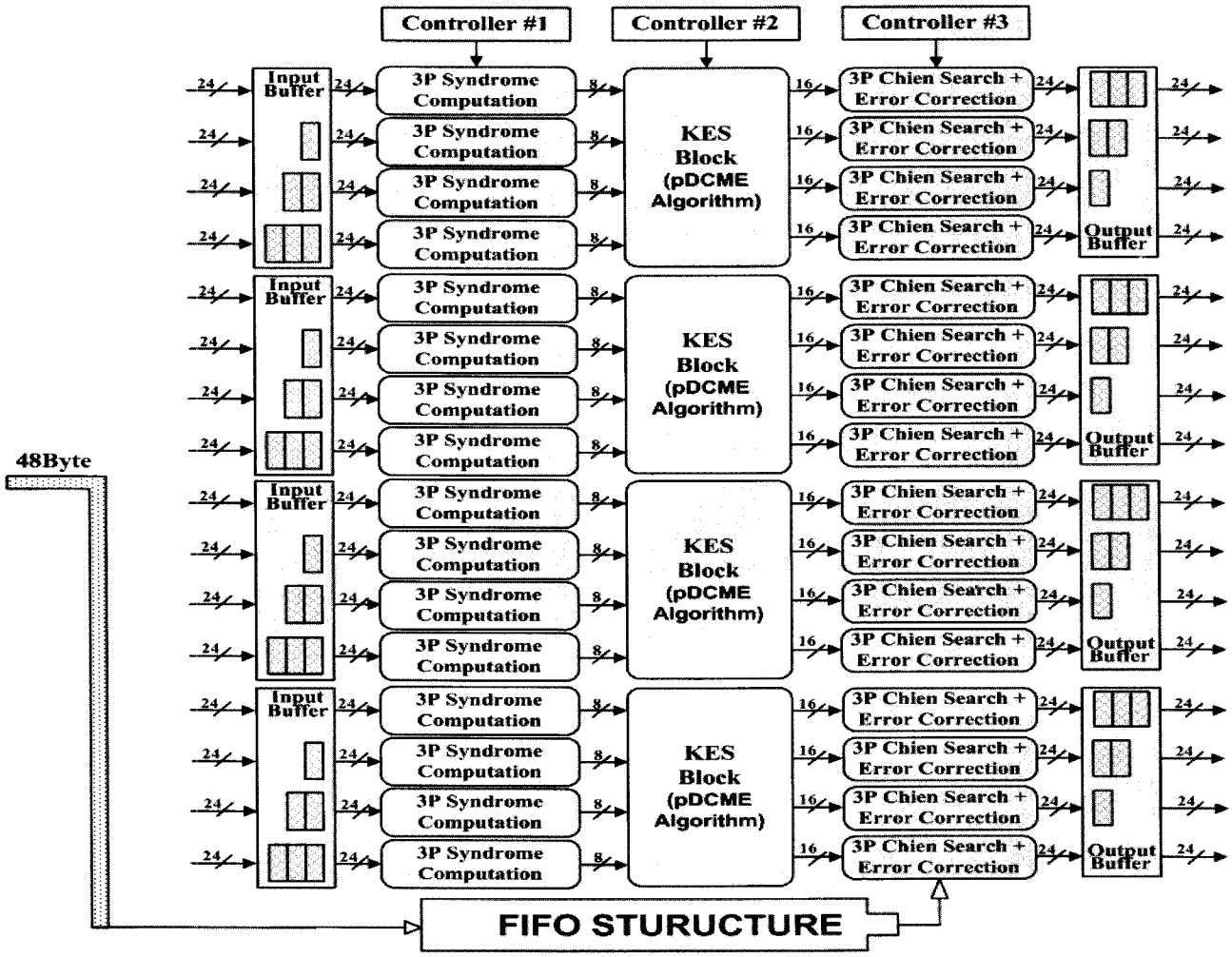


그림 6. 100-*Gb/s*급 16-채널 RS 기반 FEC 구조
Fig. 6. 100-*Gb/s* 16-Ch RS-FEC architecture.

RS-FEC architecture was modeled in Verilog HDL and simulated to verify its functionality. After complete verification of the design functionality, it was then synthesized using appropriate time and area constraints. Both simulation and synthesis steps were carried out using SYNOPSYS design tool and 0.13μm CMOS technology optimized for a 1.2V supply voltage. The total number of gates is 378,000 from the synthesized results excluding the ROM and FIFO. From pre-layout simulation, the proposed RS-FEC architecture can operate at a clock frequency of 300MHz and has a data processing rate of 115 *Gb/s* in 0.13μm CMOS technology.

Table II compares the performance of the several RS-FEC architectures for high data rates. Sharing of the KES block among 4-Ch. 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. But parallel RS decoder causes higher hardware complexity for syndrome computation block and Chien search block. Proposed three parallel RS-FEC is implemented by pDCME algorithm block, which reduces the hardware complexity of KES block. In the two-parallel architecture[8], error polynomial buffer and different clock cycles caused larger gates count compared to proposed three-parallel architecture. Table II shows that the proposed three parallel RS-FEC architecture has a much higher data processing rate and low

표 2. 16채널 RS 기반 FEC 구조의 구현 결과
Table 2. Implementation Results of the 16-Ch. RS-FEC Architecture.

Design	Proposed Three-Parallel	Two-Parallel[8]	Three-Parallel[3]	Serial[6]
Syndrome	58,000	100,800	40,000	47,000
KES	108,200	156,000	84,000	130,000
Chien+Error	211,800	178,000	240,000	72,000
Total # of Gates	378,000	434,800	364,000	249,000
Clock Rate (MHz)	300	400	112	625
Latency (Clocks)	242 (800ns)	260 (650ns)	168 (1.5μs)	355 (568ns)
Throughput (Gb/s)	115	102	43	80
Technology	0.13μm CMOS 1.2V	0.18μm CMOS 1.8V	0.16μm CMOS 1.5V	0.13μm CMOS, 1.2V

hardware complexity compared with the conventional two parallel, and three parallel RS-FEC architectures.

VI. Conclusions

This paper presents the design and implementation of three parallel RS decoder and 100*Gb/s* 16-channel RS-FEC architecture for next generation optical communication systems. Three parallel processing is used to achieve 100*Gb/s* data throughput and low hardware complexity. A high speed low complexity pDCME algorithm block is applied to the RS decoder. Three way parallel processing for syndrome computation and error correction allows 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 three parallel RS-FEC architecture has a much higher data processing rate and low hardware complexity compared with the conventional two parallel, three parallel and serial RS-FEC architectures. The proposed RS-FEC has potential applications in the next generation FEC devices for optical communications with a data rate of 100*Gb/s* and beyond.

REFERENCES

[1] IEEE P802.3ba 4IEEE P802.3ba 40Gb/s and 100Gb/s Ethrnet Task Force.
[2] ITU-T G.709/Y.1331 "Interfaces for the Optical Transport Network(OTN)" Mar. 2003
[3] L. Song, M-L. Yu and M. S. Shaffer, "10 and 40-Gb/s Forward Error Correction Devices for Optical Communications," IEEE Journal of Solid-State Circuits, vol. 37, no. 11, pp. 1565-1573, Nov. 2002.
[4] S. B. Wicker, "Error Control Systems for Digital Communication and Storage," Prentice Hall, 1995.
[5] 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. on Computers, vol. C-34, no.5, pp.393-403, May 1985.
[6] H. Lee, "High-Speed VLSI Architecture for Parallel Reed-Solomon Decoder," IEEE Trans. on VLSI Systems, vol. 11, no.2 pp. 288-294, April 2003.
[7] S. Lee, and H. Lee, "A High-Speed Pipelined Degree-Computationless Modified Euclidean Algorithm Architecture for Reed-Solomon Decoders," IEICE Transactions on fundamentals of Electronics, Communications, and Computer Sciences, vol. E91-A, no. 3, pp. 830-835, March 2008.
[8] S. Lee, C-S. Choi and H. Lee, "Two-parallel Reed-Solomon based FEC Architecture for Optical Communications," IEICE Electronics Express, vol. 5, no. 10, pp. 374-380, May 2008.
[9] 이승범, 이한호, "광통신시스템을 위한 40Gb/s Forward Error Correction 구조 설계," 전자공학회 논문지, 제45권 SD편 제2호, 101-111쪽, 2008년 2월.

저 자 소 개

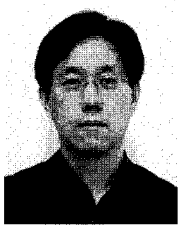
최 창 석(학생회원)

2005년 한신대학교 정보통신공학 학사

2007년 인하대학교 정보통신공학 석사

2007년~현재 인하대학교 정보공학과 박사 재학

<주관심분야 : 통신용 VLSI설계, SoC설계>



이 한 호(정회원)

1993년 충북대학교 전자공학과 학사

1996년 Univ. of Minnesota 전기컴퓨터공학 석사

2000년 Univ. of Minnesota 전기컴퓨터공학 박사

2002년 Member of Technical Staff, Lucent Technologies, USA.

2004년 Assistant Prof. Dept. of Electrical and Computer Engineering, Univ of Connecticut, USA

2004년~현재 인하대학교 정보통신공학부 부교수

<주관심분야 : 통신용 VLSI설계, SoC설계>