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

Hanho Lee*, Chang-Seok Choi*, Jongyoon Shin**, Je-Soo Ko**
*Department of Information and Communication Engineering
Inha University, Incheon, 402-751, Korea
**Electronics and Telecommunications Research Institute (ETRI)
161, Gajeong-dong, Yuseong-gu, Daejeon, 305-700, Korea
E-mail: hhlee@inha.ac.kr

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.

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 Recommendation G.709.

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 [2]-[6]. 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 [2]-[5]. 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]-[5] 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 [2]-[7].

In this paper, we present the 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.

Section II shows the proposed architecture for the three-parallel RS decoder, especially three-parallel syndrome computation block and Chien search & error correction block. Section III describes the architecture design of 100-Gb/s RS-FEC using three-parallel processing. In Section IV, implementation results and a performance comparison are presented. Finally, conclusions are provided in Section V.

II. THREE-PARALLEL REED-SOLOMON DECODER

A. Three-Parallel Syndrome Computation Block
of 3, so that proposed syndrome computation block can 
codeword is consists of 255 symbols which are multiple 
flop (1). Multiplexer selection (3) and (4) becomes 1 
becomes 1 
During 85 clock cycles, syndromes $\bar{S}_i$ are stored. 
compute new received codeword. 
The syndrome computation block calculates all the 
syndromes $\bar{S}_i$ ($0 \leq i \leq 15$) by putting the roots of 
generator polynomial $G(x)$ into the received codeword 
polynomial $R(x)$. As shown in Fig. 1, proposed three-
parallel syndrome computation block is implemented by 
following equation (4).

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

$$G(x) = (x - \alpha^n) (x - \alpha^1) \cdots (x - \alpha^{i-1}) (x - \alpha^{i+1}) \cdots (x - \alpha^{254})$$  \hspace{1cm} (2)

$$\bar{S}_i = R(\alpha^i) = r_{254}(\alpha^i)^{254} + r_{253}(\alpha^i)^{253} + \cdots + r_1(\alpha^i) + r_0$$  \hspace{1cm} (3)

$$S_i = \bar{S}_i - \bar{S}_i(\alpha^{i+1}) = r_{254}(\alpha^i)^{254} + r_{253}(\alpha^i)^{253} + \cdots + r_1(\alpha^i) + r_0$$  \hspace{1cm} (4)

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 
compute 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 $r_{254}(\alpha^i)^{254} + r_{253}(\alpha^i)^{253} + \cdots + r_1(\alpha^i) + r_0$ stored in the flip-flop (1). At the next clock cycle, The flip-flop (1) is 
multiplied by $(\alpha^i)^{254}$, and then added with $r_{253}(\alpha^i)^{253} + r_{252}(\alpha^i)^{252}$. This iterative process will be 
performed during 85 clock cycles, syndromes $S_i$ is stored in flip-
flop (1). Multiplexer selection (3) and (4) becomes 1 
every 85th clock cycle, to shift syndromes $\bar{S}_i$ to the flip-
flop (2) and to compute new received codeword. 

B. Key Equation solver Block

The pipelined degree-computationless modified 
Euclidean (pDCME) algorithm and architecture [6] is 
used to obtain the error locator polynomial $\sigma(x)$ and 
the error value polynomial $\alpha(x)$ by solving the key equation 
$\sigma(x)=S(x)\alpha(x) \mod x^5$. The detailed explanation of 
pDCME algorithm and architecture was addressed in 
our previous paper [6]. To minimize the hardware 
complexity, three 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 $\alpha(x)$ are obtained by the KES block. Let 
$X_1 = \alpha^m$ and $Y_1 = e_{m1}$. Equation (5) is transformed to 
equation (6), where $X_i$ and $Y_i$ are the possible error 
location and the possible error value, respectively. 
"$\sigma(\alpha^i)=0$" means that $r^{254}$ is corrupted by error. At first 
$\alpha^i$ is put into $\sigma(x)$ because the first symbol of 
received codeword is $r^{254}$. In equation (10), $\sigma'(x)$ 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 Chien search 
and Forney algorithm block is implemented as shown in Fig. 
2. In equation (10), dividing operation is implemented by 
256x8 ROM in which the inverse of field elements are 
stored.

$$S_i = r(\alpha^i) = e(\alpha^i) = \sum_{l=1}^{15} \alpha^{m_i}$$  \hspace{1cm} (5)

$$S(x) = \sum_{i=0}^{15} S_i x^i = \sum_{i=0}^{15} \sum_{l=1}^{15} X_i x^l$$  \hspace{1cm} (6)

$$\sigma(x) = (1-xX_1)(1-xX_1) \cdots (1-xX_m)$$  \hspace{1cm} (7)

$$\sigma(x) = S(x) \cdot \sigma(x) \mod x^{25} \hspace{1cm} (t=8)$$  \hspace{1cm} (8)

$$= \sum_{i=1}^{15} X_i \prod_{x \neq x_i, x \neq x_i}$$  \hspace{1cm} (9)

$$Y_i = \sigma(X_i^{-1})/((-X_i^{-1}) \cdot \sigma'(X_i^{-1}))$$  \hspace{1cm} (10)

In Fig. 2, serial Chien search block [5] 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 $\sigma(\alpha^i)$, $\sigma(\alpha^j)$, $\sigma(\alpha^k)$ are 
calculated and at the last clock cycle $\sigma(\alpha^{253})$, $\sigma(\alpha^{254})$ 
are calculated during 85 clock cycles.

III. 25 AND 100-Gb/S RS-FEC ARCHITECTURE WITH 
THREE-PARALLEL PROCESSING

Fig. 3 and Fig. 4 show a proposed architecture and 
a timing chart for three-parallel four channel RS-FEC 
architecture. The KES block [6] accepts the syndromes 
every 18 clock cycles, whereas three-parallel syndrome 
computation block accepts the received codeword every
Figure 2. Proposed three-parallel Chien search and Forney algorithm block.

Figure 3. Block diagram of three-parallel 4-Ch. RS FEC architecture.

Figure 4. Timing chart for three-parallel 4-Ch. RS-FEC architecture.

85 clock cycles. Therefore, syndrome generation and application of correction have to be instantiated independently 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. 4, the sharing of the pDCME algorithm requires phase shifting between each single channel.

The syndrome computation block provides $2t$ syndrome values 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. After 82 clock cycles, KES block outputs the $\sigma(x)$ and $\omega(x)$ polynomials in parallel feeding to the Chien search block. 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 100-Gb/s 16-channel RS-FEC architecture. It achieves a payload throughput 115-Gb/s with an internal clock rate of 300MHz. The block diagram of the three-parallel 100-Gb/s 16-channel RS-FEC architecture is shown in Fig. 5, 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.

**IV. RESULTS AND COMPARISON**

The proposed three-parallel 100-Gb/s 16-Ch. 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 0.13-µm 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 0.13-µm 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. Table II shows that the proposed three-parallel RS-FEC architecture has a much higher data processing rate and low hardware complexity compared with the conventional two-parallel, and three-parallel RS-FEC architectures.

Table II. Implementation Results of the 16-Ch. RS Decoder Based FEC Architectures.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Syndrome</td>
<td>58,000</td>
<td>100,800</td>
<td>40,000</td>
<td>47,000</td>
</tr>
<tr>
<td>KES</td>
<td>108,200</td>
<td>135,000</td>
<td>84,000</td>
<td>130,000</td>
</tr>
<tr>
<td>Chien, Error</td>
<td>212,800</td>
<td>178,000</td>
<td>240,000</td>
<td>72,000</td>
</tr>
<tr>
<td>Total # of Gates</td>
<td>378,000</td>
<td>434,000</td>
<td>364,000</td>
<td>249,000</td>
</tr>
<tr>
<td>Clock Rate (MHz)</td>
<td>360</td>
<td>400</td>
<td>112</td>
<td>625</td>
</tr>
<tr>
<td>Latency (Clocks)</td>
<td>242 (600ns)</td>
<td>260 (650ns)</td>
<td>168 (1.3μs)</td>
<td>355 (568ns)</td>
</tr>
<tr>
<td>Throughput (Gb/s)</td>
<td>115</td>
<td>102</td>
<td>43</td>
<td>80</td>
</tr>
<tr>
<td>Technology</td>
<td>0.13μm CMOS 1.8V</td>
<td>0.18μm CMOS 1.8V</td>
<td>0.16μm CMOS 1.5V</td>
<td>0.13μm CMOS 1.2V</td>
</tr>
</tbody>
</table>

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 parallelizing for syndrome computation 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 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.

ACKNOWLEDGMENT

This work was partly supported by the IT R&D program of MKE/ITTA, Korea (2008-F-017-01, 100Gbps Ethernet and optical transmission technology development), and partly supported by the MKE, Korea, under the ITRC support program supervised by the ITTA (ITTA-2008-C1090-0081-0019).

REFERENCES