Ultra Low-Cost 3.2Gb/s Optical-Rate Reed Solomon Decoder IC Design

Huai-Yi Hsu  
Graduate Institute of Electronics Engineering  
National Taiwan University  
Taipei, Taiwan (R.O.C.)

Jih-Chiang Yeo  
Graduate Institute of Electronics Engineering  
National Taiwan University  
Taipei, Taiwan (R.O.C.)

An-Yeu (Andy) Wu  
Graduate Institute of Electronics Engineering  
National Taiwan University  
Taipei, Taiwan (R.O.C.)

Abstract—Reed-Solomon (RS) code is a widely used Forward Error Correction (FEC) technology to improve the channel impairments in the fiber communication systems. The typical parallel FEC architecture requires huge hardware cost to achieve very-high-speed transmission data rate for the optical systems. This paper presents an ultra low-cost VLSI architecture of RS decoder by using a novel Just-in-Time Folding Modified Euclidean Algorithm (JIT-FMEA). The JIT-FMEA VLSI architecture can greatly reduce the hardware complexity by about 50% from the fully expanded parallel architecture, while it can achieve very high throughput rate for the 10Gbase-LX4 optical communication system. The proposed RS decoder architecture has been designed and implemented by using 0.18μm CMOS standard cell technology at the supply voltage of 1.8V. Our design only requires about 21K gates and can achieve the data throughput rate of 3.2Gbit/s at clock frequency of 400MHz.

I. INTRODUCTION

The capacity of optical transmission systems has been drastically increased to reach the range of tens gigabits per second over the past ten years, the channel impairments become more and more severe, and the degradation of the optical signals limits the data transmission distance. Therefore, advanced digital signal processing (DSP) techniques have been applied to enhance the transmission capacity in the optical systems, such as Equalization (EQ) and Forward Error Correction (FEC) codec [1].

The Reed-Solomon (RS) code provides excellent error correcting capability for both random and burst errors [2]. The system simulation of the optical system shows that the RS(255, 239) code can provide approximately 5.5dB coding gain to reduce the bit error rate (BER) from $10^{-4}$ to $10^{-15}$ for correcting random errors [3].

For optical applications, the transmission rate reaches several gigabits per second. Conventionally, many high-speed RS decoder designs have adopted fully expended parallel architectures to achieve the requirement of high throughput rate [5]-[9]. However, the hardware utilization is not efficient. In this paper, we develop a new decoding algorithm called Just-in-Time Folding Modified Euclidean Algorithm (JIT-FMEA), which can construes an ultra low-cost VLSI architecture and achieve optical throughput rate. The key idea is to use folding and pre-computation scheme (PCS) to reduce hardware cost and to eliminate the idle cycles. Besides, the retiming ability of JIT-FMEA architecture can overcome the critical paths of the bottleneck in a RS decoding procedure to achieve high throughput rate in optical systems.

The folding RS decoding architecture can reduce the hardware complexity about 50% from the conventional fully parallel architecture [11]. We implement the design in 0.18μm CMOS technology. Our chip can work up to 400MHz and the throughput rate can achieve 3.2Gbit/s to meet 10Gbase-LX4 optical communication systems. Moreover, based on the developed RS decoder, we can easily construct the 4-way parallel FEC decoder architecture for the application of 10Gbase-LX4 optical transmission systems as in Fig. 1.

II. PROPOSED JUST-IN-TIME FOLDING MODIFIED EUCLIDEAN ARCHITECTURE

The syndrome-based RS decoding scheme consists of three components, the Syndrome Calculator (SC), the Key Equation Solver (KES), and the Error Corrector (EC) [4].

Since the KES involves the highest computational complexity among the three components of RS decoding
procedure, it dominates the speed and the hardware complexity of the RS decoders. Hence, the throughput bottleneck in a RS decoder is in the KES block. Other blocks of the RS decoder can be simply pipelined due to their feed-forward structures. To meet the optical-rate requirement, the target RS decoder need to provide two design features—high data processing rate but at low hardware complexity.

A. Drawback of Existing RS Architectures

In order to achieve high data throughput rate, many RS decoders employ the pipelined and/or parallel architecture to improve the throughput rate. For example, [6][9] use the pipelined architecture to improve maximum operating frequency. However, it results in much more registers to handle the timing matching issue, which greatly increase latency and hardware cost.

In order to reduce hardware complexity, some other RS decoders have adopted resource-sharing and/or time-multiplexing scheme to reduce the hardware cost. In general, the directly folding-by-2r method finishes the KES operation in $(2^r)^2 = 256$ cycles, which is greater than the received code length $n (=255)$. Since the directly folding-by-2r method needs one more cycle, it is not suitable to be applied in continuous real-time processing.

B. Just-in-Time Folding Modified Euclidean Algorithm

Due to the disadvantages of the existing RS decoding architectures, in this paper, we propose an ultra low-cost architecture compared with parallelism MEA architecture to achieve optical throughput rate.

We employ the regular MEA parallel architecture and fold it by $2t$ [11]. The folding architecture only requires one processing element (PE) to solve the KES operation, which is $1/2t$ of what the full parallel architecture requires. Hence, we can greatly reduce the total hardware cost. Moreover, we adopted the pre-calculation scheme (PCS) to eliminate the idle cycles in the folding MEA algorithm to save the first iteration, i.e., we can save $2t$ operating cycles in the KES block. Therefore, the PCS method can not only save first iteration from MEA, but also diminish the operating complexity and the power consumption at no increase of extra circuit.

At each iteration operation, we employ the additional cycle to lock the leading coefficients, $a_i$ and $b_i$, to ensure the correct results. Moreover, its control flow can be simplified. Consequently, each iteration operation needs $2t+1$ symbol cycles. Therefore, FMEA architecture need $(2t-1)(2t+1) = (2^r)^2 -1 = (2-8)^2 -1 = 255$ cycles, which equals to $n$ symbols, to accomplish the KES procedure. We call this method Just-in-Time Folding Modified Euclidean Algorithm (JIT-FMEA). It does not have any idle cycles during the RS decoding procedure. Therefore, this algorithm can provide fully hardware utilization as compared to conventional parallel architecture.

The new timing chart of the proposed JIT-FMEA is shown in Fig. 2. We can see that no idle cycle occurs in during the KES procedure. With the modification, now the decoding method can be easily applied to the 10GBase-LX4 system, which doesn’t require additional buffers at input and output terminals to handle the synchronization problem [10].

C. Proposed JIT-MEA Architecture

In [11], we have proposed a scalable parallel MEA architecture as shown in Fig. 3. It can be easily folded by $2t$. Moreover, according to the JIT-FMEA algorithm, we can easily construct the regular datapath for JIT-MEA architecture as shown in Fig. 4, which is very regular and can be easily controlled.

![Fig. 2. New timing chart without idle cycles for JIT-FMEA algorithm.](image1)

![Fig. 3. (a) The parallel Modified Euclidean Division architecture. (b) The parallel Modified Euclidean Multiplication architecture](image2)

![Fig. 4. The proposed JIT-FMEA architecture.](image3)
The JIT-FMEA algorithm consists of two major operations: One is the RQ part for Modified Euclidean Division (MED) operation. It performs the long division operation to obtain the error magnitude polynomial \( \omega(x) \). The other is the LU part for Modified Euclidean Multiplication (MEM) operation. It performs the multiplication and accumulation in the polynomial domain. It is used to obtain the error location polynomial \( \sigma(x) \).

The KES operation can be separated into two modes, exchange and un-exchange. All odd iterations are operating in the exchange mode; and all even iterations are operating in the un-exchange mode. All iterations are performed alternatively between the exchange and the un-exchange modes until the stop condition is satisfied. Finally, we can give the error magnitude polynomial from the MSB of R-type registers and the error location polynomial from the MSB of L-type registers, \( \omega(x) \) and \( \sigma(x) \), respectively. The control flow of the JIT-FMEA architecture shows in Fig. 5.

According to the initial values of the PCS method, we need to initialize all types of registers, \( R, Q, L \) and \( U \), at the starting phase of the KES block. Two major operations: MED operation and MEM operation can be folded by 2t into RQ and LU computational blocks. The architecture is shown in Fig. 4. Moreover, using additional cycle to lock the leading coefficients can simplify the control flow.

### III. CHIP IMPLEMENTATION

The maximum operation frequency of the proposed RS decoder can achieve 400MHz. Therefore, the timing issue is very critical. Consequently, we must be considered the timing issues in whole IC design flow carefully. Such as, we adopt Design Compiler and Physical Compiler to optimize the critical path to meet the system requirement. Moreover, employing static timing analyzer (STA) tool to make sure overall paths can fit the timing constraint at each stage check. Finally, the post-simulation can achieve 400MHz at three corners (SS, TT, and FF).

The proposed RS decoding chip based on JIT-FMEA algorithm has only 20,614 gate count, and the core size is only 600×600 μm², which can operate at the clock frequency of 400MHz and has a data throughput rate of 3.2Gbps in 0.18um CMOS technology at 1.8V. Consequently, the proposed area-efficient architecture can achieve the data throughput rate of 10Gbase-LX4 system. The chip feature summaries are shown in Fig. 6.

### IV. MEASUREMENT RESULTS

We are using PC-based Pattern Generator (PG) to generate testing signals, and using Tektronix Logic Analyzer (LA) to catch output signals of JIT-MEA RS chip. The testing platform is shown in Fig. 7. As the measurements, we sequentially input the encoded codeword to this chip, whether this chip can successfully to correct various number of errors.

The measurement results of functionality are shown in Fig. 8. Fig. 8 (a)–(h) denote the numbers of errors are from one to eight. The signal of ZeroDetect means the error occurring if its value is equal to 1. And, the signal of EV means the error magnitude. As we can see, the functional testing is correct. The academic equipments only support to 100MHz for digital IC testing. Moreover, the limitation of IO pad is 200MHz. Therefore, we join the Phase-Lock-Loop (PLL) and the Build-In-Self-Test (BIST) techniques for this design at next version, and we had tape-out new version on February in this year.
V. COMPARISON

We make comparisons between our design and other existing chip solutions as listed in Table II. For fair comparison, we define the Area-Efficiency (AE) is the maximum data throughput rate to over total gates. In order to eliminate the factor of different fabrication technology, we adopt the Normalize Index. The Normalized Area-Efficiency (NAE) is the silicon area normalized to a 0.13µm technology, as shown below:

\[
\text{NAE} = \frac{\text{Throughput Rate}}{\# \text{ of Total Gates}} \times \text{technology} \\
\text{0.13µm}
\]

![Table II Comparison Results of the Area Efficiency](image)

As we can see from Table II, the proposed architecture has the smallest cost. Moreover, the high throughput performance meets current high-speed applications of RS codes. The area-efficiency performance of our RS chip is the highest. The normalized area-efficiency is 3 to 4 times exceeding modern existent literatures.

VI. CONCLUSIONS

In this paper, we have developed ultra low-cost Reed Solomon decoder IC design, which can be used in the 10GBase-LX4 optical communication system. The JIT-FMEA architecture can reduce the hardware complexity about 50%, which is more efficient compared to the fully parallel architecture. Comparing with others, our design has smaller size, higher throughput rate and area-efficiency.

ACKNOWLEDGMENT

This work was supported in part by MediaTek Inc. and National Science Council, R.O.C, under grants NSC 92-2220-E-002-012. Moreover, we thank the support by National Chip Implementation Center (CIC) in IC design.

REFERENCES