Area-Efficient VLSI Design of Reed-Solomon Decoder for 10GBase-LX4 Optical Communication Systems

Huai-Yi Hsu[*] and Jih-Chiang Yeo[*] and An-Yeu (Andy) Wu[**]

[*]Graduate Institute of Electronics Engineering,
[**]Graduate Institute of Electronics Engineering, and Department of Electrical Engineering,
National Taiwan University, Taipei 106, Taiwan, R.O.C.
Contact people: An-Yeu (Andy) Wu (E-mail: andywu@cc.ee.ntu.edu.tw)

ABSTRACT

This paper proposes an area-efficient architecture to implement the *Modified Euclidean algorithm* (*MEA*), which is frequently used in Reed-Solomon decoders. We present the new ME architecture to achieve high-throughput rate and reducing hardware complexity. We propose a folding architecture to reduce the hardware complexity about 50% compared to the fully parallel architecture. The Modified Euclidean algorithm has been implemented in 0.18-μm CMOS technology with 1.8V supply voltage. The results show that total number of gates is about 20K and it has a data processing rate of 3.2Gbit/s at clock frequency of 400 MHz. The proposed area-efficient architecture can be readily applied to 10Gbase-LX4 optical communication systems.

Keywords: Forward Error Correcting, Reed-Solomon code, Modified Euclidean Algorithm, 10Gbase-LX4.

1. INTRODUCTION

The capacity of optical transmission systems has rapidly increased over past ten years. When the capacity increased, the transmission quality degrade from noise and pulse distortion has become more serious When the data rate of optical systems arrives to the range of tens of gigabits per second, the impairments of optical channels become more and more serious and limit transmission distance. Therefore, some advanced digital signal processing techniques can be applied to optical systems, such as equalization and forward error correction. The enhanced gain provided by these techniques can be used to reduce channel impairments, improve overall transmission quality, and increase transmission distance. Hence, it can reduce system cost and increase the limit of system transmission capacity.

Forward Error Correcting (*FEC*) codes compensates quality degradation and increases transmission distance and system capacity due to its error-correcting capability. In particular, the RS(255,239) code is now commonly used for Wavelength Division Multiplexing (WDM) systems due to its 8-byte error-correcting capability.

When FEC upon the range of tens of gigabits per second. Designing a high performance RS decoder is more challenging, because the RS decoder consumes the highest power and occupies the largest area of the FEC codec due to its high computational complexity. Therefore, we developed an area-efficient Modified Euclidean architecture, which can reduce the hardware complexity about 50% compared to the fully parallel architecture.

The paper is organized as follows. The typical Reed-Solomon decoding procedure is described in Section 2. Section 3 describes the conventional fully parallel architecture of the MEA. Section 4 presents the area-efficient architecture of the MEA. The comparisons of these RS decoder realizations are illustrated in Section 5. Finally, the conclusions are given in Section 6.

2. FORWARD ERROR CORRECTION

Among various *Forward Error Correcting* (*FEC*) techniques, Reed Solomon codec is one of the most widely applied schemes for error correction. It provides excellent error correction capability for both random and burst errors. Hence, RS has been employed in many practical applications, such as digital audio and video, magnetic and optical recording, computer memory, cable modem, xDSL wireless and satellite communications systems [1], [2].

0-7803-8637-X/04/$20.00©2004 IEEE.

The decoding procedure of RS code based on the syndrome-based architecture essentially consists of three components, as shown in Fig. 1, [3].

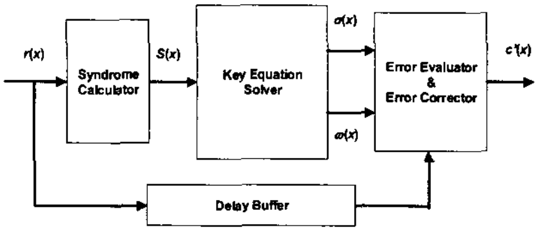


Fig. 1 Syndrome-based Reed-Solomon decoding.

First component is the syndrome calculator that calculates a syndrome polynomial from the received codewords, which is used in the second component for solving the key equation. In the second component, we adopt Modified Euclidean GCD algorithm to solve a key equation for an error-location polynomial and an error-value polynomial. Then in the third component, these two polynomials are used to find out the error locations and the corresponding error values according to the Chien's search and Forney algorithm. In addition, a FIFO memory is used in order to buffer the received symbols according to the latency of these components.

Since the second component of decoding process, which finds the error locator and error magnitude polynomial, involves higher computational complexity; it affects the speed and the hardware complexity of RS decoding. Therefore, we propose an area-efficient architecture to reduce hardware complexity.

3. MODIFIED EUCLIDEAN ALGORITHM

Assume that ν symbols error occurred in data transmission; we can define the error location polynomial as

$$\sigma(x)=\prod_{l=1}^{\nu}(1+xX_l)=\sigma_0+\sigma_1x^1+\cdots+\sigma_l x^{\nu} \quad (1)$$

Besides, we also can define the error magnitude polynomial, it denoted by $\omega(\underline{x})$, which can be written as

$$\omega(x)=\omega_0+\omega_1x^1+\cdots+\omega_{l-1}x^{\nu-1} \quad (2)$$

Furthermore, we define the syndrome polynomial, it denoted by $S(x)$, which can be written as

$$S(x)=\sum_{i=0}^{2t-1}S_ix^i=S_0+S_1x^1+\cdots+S_{2t-1}x^{2t-1} \quad (3)$$

We can obtain the error locator and error magnitude polynomials by solving the key equation, as shown in Eq. (4). The error locator and error magnitude polynomials can be obtained by using Euclidean GCD algorithm.

$$S(x)\sigma(x)=\omega(x)\bmod x^{2t} \quad (4)$$

To solve the key equation, Euclidean GCD algorithm is an iteration procedure for finding the error location polynomial and the error magnitude polynomial [2]. The algorithm can be explained using the following iteration equations:

A. Initial conditions:

$$\begin{cases}R_0(x)=x^{2t}\\Q_0(x)=S(x)\end{cases} \quad \textit{and} \quad \begin{cases}L_0(x)=0\\U_0(x)=1\end{cases} \quad (5)$$

B. In i-th iteration:

$$\begin{cases}R_i(x)=s_{i-1}\left(\begin{bmatrix}b_{i-1}R_{i-1}(x)\\a_{i-1}Q_{i-1}(x)\end{bmatrix}+x^l\begin{bmatrix}a_{i-1}Q_{i-1}(x)\\b_{i-1}R_{i-1}(x)\end{bmatrix}\right)\\Q_i(x)=s_{i-1}\left(\begin{bmatrix}Q_{i-1}(x)\\R_{i-1}(x)\end{bmatrix}\right)\end{cases}, \quad (6)$$

$$\begin{cases}L_i(x)=s_{i-1}\left(\begin{bmatrix}b_{i-1}L_{i-1}(x)\\a_{i-1}U_{i-1}(x)\end{bmatrix}+x^l\begin{bmatrix}a_{i-1}U_{i-1}(x)\\b_{i-1}L_{i-1}(x)\end{bmatrix}\right)\\U_i(x)=s_{i-1}\left(\begin{bmatrix}U_{i-1}(x)\\L_{i-1}(x)\end{bmatrix}\right)\end{cases}, \quad (7)$$

C. Stop condition:

$$\deg(R_{i-1}(x))<t\,. \quad (8)$$

a_{i-1} and b_{i-1} denote the leading coefficients of $R_{i-1}(x)$ and $Q_{i-1}(x)$, respectively. Finally, the algorithm stops condition is satisfied, then $\omega(x)=R_{i-1}(x)$ and $\sigma(x)=Q_{i-1}(x)$. The conventional architecture of the Modified Euclidean algorithm has two parts: the Euclidean multiplier and divider.

3.1. Modified Euclidean Division Module

Modified Euclidean Division (*MED*) operation is to compute Eq. (6). It performs the long division operation to obtain error magnitude polynomial, $\omega(x)$. In performing the mutual multiplication, the polynomial $Q_{i-1}(x)$ multiplies with the leading coefficient of $R_{i-1}(\underline{x})$, a_{i-1}. On the other hand, the polynomial $R_{i-1}(x)$ multiplies with the leading coefficient of $Q_{i-1}(\underline{x})$, b_{i-1}. We can obtain new two polynomials, $R_i(x)$ and $Q_i(x)$. By using the iterative method, we can get the *Greatest Common Divisor* (*GCD*) of both $S(x)$ and x^{2t}. This polynomial division architecture includes one elemental component, the *RQ* modules in Fig. 2. The Euclidean divider module architecture is shown in Fig. 3.

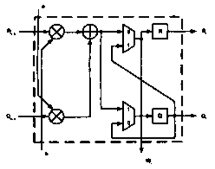


Fig. 2 RQ module for Modified Euclidean Division.

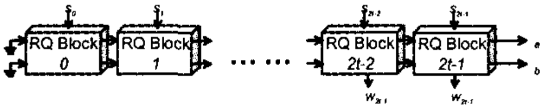


Fig. 3 The block diagram of the multi-mode MED architecture for *t* correction capability.

3.2. Modified Euclidean Multiply Module

Modified Euclidean Multiplication (MEM) operation is used to compute Eq. (7) to perform the multiplication and accumulation in the polynomial domain. It is used to obtain the error location polynomial $\sigma(x)$. Similar to the MED method, we can easily obtain the error location polynomial. The polynomial multiply architecture includes one major component of the LU module, as shown in Fig. 4. The MEM module architecture is shown in Fig. 5.

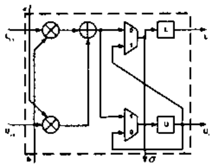


Fig. 4 LU module for Modified Euclidean Multiplication.

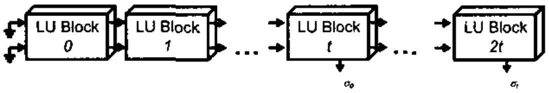


Fig. 5 The block diagram of the multi-mode MED architecture for t correction capability.

4. AREA-EFFICIENT MEA ARCHITECTURE

The fully parallel architecture requires only a total of *n-k* symbols time to complete MEA. A full parallel implementation of MEA has only a latency of 2t cycles with a throughput rate of one set polynomial per cycle. But, this way needs large hardware cost in high-speed operation. Moreover, the timing chart is show as Fig. 6. The parallel implementation consumes huge area-cost. And hardware cost is expensive. Hence, we propose an area-efficient architecture, which uses time-cycles instead of space-cost.

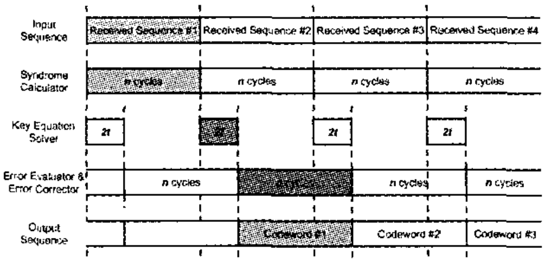


Fig. 6 The timing chart for parallel architecture of MEA.

We employ the regularly MEA parallel architecture to fold by 16. In ideally, the folding architecture need only 1/16 hardware cost to solve the key equation. The folding architecture is shown as Fig 7.

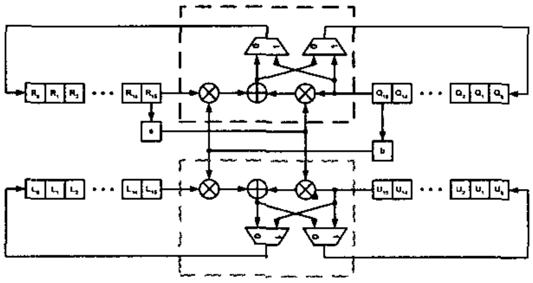


Fig. 7 The proposed folding architecture of MEA.

In general, using a single cell recursively requires only a total of $(n\text{-}k)^2$ symbol time to complete the MEA. In the case of a RS(255,239) code, the calculation of syndrome only need 255 cycles, so it is difficult to difficult to design the whole decoder into pipeline. Hence, we need to reduce the number of symbol times, let it can fixed 255 cycles. We have developed a new ME cell. The proposed ME cell have an improvement, which reduce the number of symbol times significantly as compared to conventional architecture.

The improvement can eliminate first iteration computation by using the pre-calculation scheme, thus total processing time can be reduced to 2t-1 iterations computation. The pre-calculation scheme uses the new initial values of MEA, which are

$$\begin{cases} R_0(x) = xS(x) \bmod x^{2t} \\ Q_0(x) = S(x) \end{cases} and \begin{cases} L_0(x) = x \\ U_0(x) = 1 \end{cases} \qquad (1)$$

Using additional cycle to lock the leading coefficients, a_i and b_i, ensure getting correcting results. Hence, each iteration needs *2t+1* symbol cycles.

Therefore, our architecture need $(2t)^2$-1 symbols, which equals to n symbols. Consequently, the. proposed architecture can be pipelined to arrive high throughput rate. The new timing chart is shown as Fig. 8.

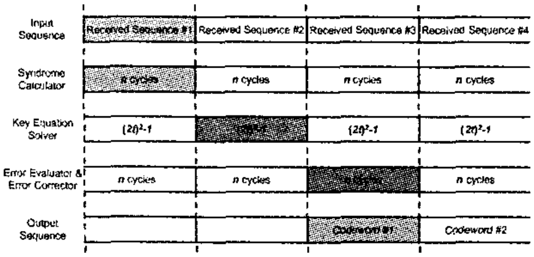
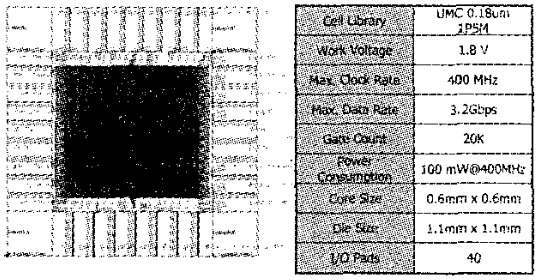


Fig. 8 The timing chart for proposed architecture.

The proposed architecture can reduce hardware-cost enormously. From synthesis results, we can save 50% hardware cost of MEA more efficient than the parallel architecture. Moreover, it keeps the speed performance.

5. IMPLEMENTATION RESULT

We propose the VLSI architecture that has only 20,614 gate count and core size is only 600×600 um^2. From post-layout simulation, the chip can operate at a clock frequency of 400 MHz and has a data processing rate of 3.2 Gbps in 0.18um CMOS technology at 1.8 V. The chip summary is shown in Fig. 9.



Cell Library	UMC 0.18um 1P5M
Work Voltage	1.8 V
Max. Clock Rate	400 MHz
Max. Data Rate	3.2Gbps
Gate Count	20K
Power Consumption	100 mW@400MHz
Core Size	0.6mm x 0.6mm
Die Size	1.1mm x 1.1mm
I/O Pads	40

Fig. 9 The chip layout of the Reed Solomon codec.

We make comparison between our design and other existing chip solutions as listed in Table 1. The proposed architecture has smallest cost. And, the high throughput performance meets current high-speed applications of RS codes. We can see from this comparison table that the efficient performance of our RS chip is highest. The efficiency is 2 to 3 times exceeding modern literatures.

Table 1 The comparison table of performance.

	Tech. (um)	Latency (Cycles)	Clock (MHz)	Total of #Gates	Throughput (Gbit/s)	Efficiency (Mbit/s/gate)
Our RMEA	0.18	$n+(2t)^2-1$	400	20,614	3.2	152.38
ISVLSI'03 [4]	0.13	287	300	44,700	2.4	53.70
ASIC-SOC'01 [5]	0.13	355	770	102,500	6.16	60.10
ISCAS'00 [7]	0.25	n+3t+42	75	55,240	0.6	10.86
ICCSC'02 [8]	0.25	n+16	100	118,303	0.8	6.76

6. CONCLUSIONS

In this paper, we have developed VLSI architecture of the area-efficient Reed Solomon decoder, which can be used in the 10GBase-LX4 optical communication system. The folding architecture can reduce the hardware complexity about 50% compared to the fully parallel architecture. Hence, we proposed new architecture greatly improve area-efficient about double or triple. Our chip can work up to 400MHz and throughput rate can arrive 3.2G/s to meet 10Gbase-LX4 optical communication systems. Moreover, total gate count is about 20K. Comparing with others, our design has smaller size, higher throughput rate and area-efficient.

7. REFERENCES

[1] S. Lin and D. J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Englewood Cliffs, NJ: Prentice-Hall, 1983.

[2] R. Blahut, Theory and Practice of Error Control Codes, Addison-Wesley Co., 1983.

[3] A. Raghupathy and K. J. R. Liu, "Algorithm-Based Low-Power/High-Speed Reed-Solomon Decoder Design," IEEE Trans. on CAS-II, vol.47, no. 11, Nov. 2000, pp. 1254-1270.

[4] Hanho Lee, "An Area-Efficient Euclidean Algorithm Block for Reed-Solomon Decoder," Proceedings of the IEEE Symposium on VLSI, 2003.

[5] Hanho Lee, "A VLSI Design of A High-Speed Reed-Solomon Decoder," 14th Annual IEEE International, ASIC/SOC Conference, 12-15 Sept. 2001. pp.316-320

[6] Hanho Lee, "Modified Euclidean algorithm block for high-speed Reed-Solomon decoder," IEE 5th Electronics Letters, vol. 37, no.14, pp. 903-904, July 2001.

[7] Hanho Lee, Meng-Lin Yu, and Leilei Song, "VLSI Design of Reed-Solomon Decoder Architectures," IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland

[8] L. Song, M. L. Yu, and M. S. Shaffer, "10- and 40-Gb/s Forward Error Correction Devices for Optical Communications," IEEE Journal on Solid-State Circuits, vol. 37, no. 11, Nov. 2002. pp. 1565-1573.