Low Energy High Speed Reed-Solomon Decoder Using Two parallel Modified Evaluator Inversionless Berlekamp-Massey

Hazem A. Ahmed, Hamed Salah, Tallal Elshabrawy, and Hossam A. H. Fahmy.

Abstract—**This paper proposes a low power high throughput Reed Solomon decoder designed optimally for handheld devices under the DVB-H standard. This architecture based on Decomposed Inversionless Berlekamp-Massey Algorithm (DiBM), where the error locator and evaluator polynomial can be computed serially. In the proposed architecture, a new scheduling of 6 Finite Field Multipliers (FFMs) is used to calculate the error locator polynomial in a two parallel way and these multipliers are reused to calculate the error evaluator polynomial in a novel architecture called two parallel modified evaluator decomposed inversionless Berlekamp-Massey (MEDiBM) to achieve low energy. This architecture is tested in a pipelined two parallel decoder. This decoder has been implemented by $0.13\mu m$ CMOS IBM standard cells for RS$(204,\,188)$ and gave gate count of $33\,K$ and area of $1.06\,mm^2$. Simulation results show this approach can work successfully at the data rate $100\,Mbps$ with power dissipation of $0.266\,mW$.**

I. Introduction

Among the various kinds of error correcting codes (ECC) in digital communication systems, Reed-Solomon (RS) code is is especially suitable for the situation where long codes are needed, for example, Digital Video Broadcast-Handheld (DVB-H) system.

The conventional RS decoder architecture [1], can be summarized into four steps : 1) calculating the syndromes from the received codeword; 2) computing the error locator polynomial and the error evaluator polynomial; 3) finding the error locations; and 4) computing error values. It can be modeled with the Block diagram shown in Figure 1.

The second step is considered the most complex part in RS decoding, there are two main approaches to compute the error locator and evaluator polynomials [1], the Berlekamp-Massey algorithm, and the Euclidean algorithm. Berlekamp-Massey algorithm gives Lower complexity than the standard Euclidean algorithm [1].

In wireless communication applications most of RS architectures focused on low power consumption with a reasonable area. Figure 1 shows the conventional pipelined architectures for the conventional RS decoder. The bottleneck in this architecture is the syndrome and Chien search blocks where they need n clock cycles to finish, where n is the codeword legnth. The serial architecture [2] was only interest to minimize the area so this architecture suggests 3 FFMs implementation to compute $\sigma(x)$ and $W(x)$ with latency equal to $3t^2 + 3t$ clock cycles, where t is the number of symbols that can be corrected with this code so the throughput of this architecture is controlled by the syndrome circuit n clock cycles. An architecture for syndrome and Chien search blocks are proposed as two parallel syndrome and two parallel Chien search [3] where they need $\frac{n}{2}$ clock cycles to finish which make the bottleneck of the architecture in the algorithm and according to that the throughput will be controlled of the latency of the algorithm, then the modified evaluator architecture [4] reduced the latency of the algorithm to be $2t^2 + 5t$ and according to this modification the throughput increased and controlled by the latency of the algorithm.

In this paper a two parallel modified evaluator architecture for Decomposed inversionless Berlekamp-Massey (DiBM) algorithm is proposed. In this architecture, the locator polynomial is calculated in two parallel way [5] which needs 6 FFMs in its calculations, then the error evaluator polynomial is efficiently implemented by reusing the 6 FFM to reduce the latency of the algorithm to be t^2+4t+2 which transfers the bottleneck to the two parallel syndrome block again and make the throughput controlled by the two parallel syndrome $\frac{n}{2}$ clock cycles, so to support the same throughput for architecture [2] and the proposed and architecture, the proposed architecture needs half the frequency which is used in architecture [2] which makes the power dissipation of the proposed architecture is better.

The organization of this paper is as follows: Section 2 introduces the fundamental decoding of Reed-Solomon codes and the main blocks of RS decoders. Section 3 presents the proposed architecture of the Key Equation Solver (KES) "two parallel modified evaluator architecture". Section 4 discusses the architecture of the proposed decoder. Section 5 compares between the proposed architectures and other RS(204; 188) architectures. Finally, section 6 gives our conclusions.

II. Reed Solomon Code

RS codes are non-binary cyclic codes. RS$(n,\,k)$ codes on m-bit symbols exist for all n and k for which $0 < k < n \leq 2^m - 1$, where k is the number of data symbols to be encoded, and n called codeword. This means that the RS encoder takes k data symbols and adds parity symbols (redundancy) of $(n-k)$ symbols to make an n symbol codeword in systematic form. For the most conventional RS$(n,\,k)$ code $(n,\,k)\ = (2^m-1, (2^m-1)-2t\,)$ where t is the number of symbols that can be corrected with this code, where t can be expressed as $t = \lfloor (n-k)/2 \rfloor$

The code generator polynomial $g(x)$ of the code is

$$g(x) = \prod_{i=0}^{n-k} (x - \alpha^{i+j}) \tag{1}$$

Where j is an arbitrary integer and α is a primitive element of the field GF(2^m), i.e. is a root of field generator polynomial.

Decoding process can be divided into four steps. The first step is to calculate the syndrome polynomial $S(X)$ with $2t$ coefficients S_i from the received codeword $R(x)$ as shown in equations, 3 and 4.

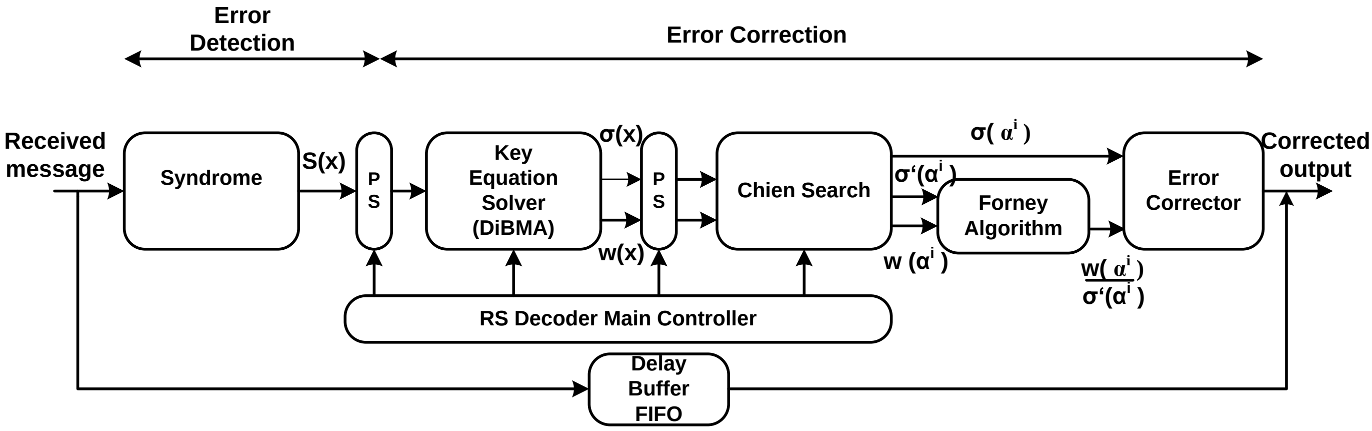

Figure 1. Main Block Diagram of RS Decoder

$$
\begin{aligned}
R(X) &= r_{n-1}x^{n-1}+r_{n-2}x^{n-2}+\cdots+r_1x+r_0 && (2)\\
S(x) &= \sum_{i=1}^{2t} S_i x^{i-1} && (3)\\
S_i &= R(\alpha^i)=r_{n-1}(\alpha^i)^{n-1}+r_{n-2}(\alpha^i)^{n-2}+\cdots\\
&+\ r_1(\alpha^i)+r_0 && (4)
\end{aligned}
$$

where $i = 1, 2, \ldots, 2t$.

The second step is the key equation solver (KES) which solves the key equation $S(x)\sigma(x) = W(x) mod\, x^{2t}$ to produce the error locator polynomial $\sigma(x)$ and the error evaluator polynomial $W(x)$ from the syndrome polynomial $S(x)$. The key equation solver technique will be explained in detail in section 3. The third and fourth steps are parallel Chien search and Forney algorithm respectively to produce the error locations and error values from equation $e_l = \frac{W(\alpha^{-l})}{\sigma'(\alpha^{-l})}$, where α^{-l} is the root of $\sigma(x)$. But the two steps can be combined in one block which is called error evaluator block.

III. Architecture Of Two Parallel Modified Evaluator Of DiBM Algorithm

A. Computation of the Error Locator Polynomial $\sigma(x)$

The Error Locator Polynomial $\sigma(x)$ in our architecture is computed in a $2t$ step iterative algorithm. The initial conditions are $D^{(-1)} = 0$, $\delta = 1$, $\sigma^{(-1)}(x) = T^{(-1)}(x) = 1$, and $\Delta^{(0)} = S_1$

where $\sigma^{(i)}(x)$ is the ith step error locator polynomial and $\sigma_j^{(i)}$'s are the coefficients of $\sigma^{(i)}(x)$;$\Delta^{(i)}$ is the ith step discrepancy and δ is a previous nonzero discrepancy; $T^{(i)}(x)$ is an auxiliary polynomial and $D^{(i)}$ is an auxiliary degree variable in ith step. then the algorithm proceeds as

for $(i = 0\ to\ 2t-1)$

$$
\begin{cases}
\sigma^{(i)}(x) &= \delta.\sigma^{(i-1)}(x)+\Delta^{(i)}xT^{(i-1)}(x),\\
\Delta^{(i+1)} &= S_{i+2}\sigma_0^{(i)}+S_{i+1}\sigma^{(i)}+...+S_{i-t+2}\sigma_t^{(i)}
\end{cases}
\qquad (5)
$$

if $(\Delta^{(i)} = 0\, or\, 2D^{(i-1)} \geq i+1)$ then

$D^{(i)} = D^{(i-1)},\ T^{(i)}(x) = xT^{(i-1)}(x)$

else

$D^{(i)} = i+1-D^{(i-1)},\ \delta = \Delta^{(i)},\ T^{(i)}(x) = \sigma^{(i-1)}(x)$

The ith iteration can be decomposed into $(\lceil \frac{t+1}{2} \rceil + 1)$ cycles. In each cycle two coefficients from $\sigma^{(i)}(x)$ are calculated in parallel as shown in equations, 6 and 7, these coefficients need four FFMs. In the same clock cycle two partial results from the discrepancy $\Delta^{(i+1)}$ are calculated in parallel as shown in equation 9, this operation needs two FFMs. So it is clear that we need 6 FFMs in parallel in the proposed architecture to get the error locator polynomial $\sigma(X)$.

Define

$$
\sigma_{2j}^{(i)} = \begin{cases}\delta.\sigma_0^{(i-1)}, & for\, j=0\\ \delta.\sigma_{2j}^{(i-1)} \quad +\Delta^{(i)}T_{2j-1}^{(i-1)}, & for\, 1\leq j\leq t\end{cases} \qquad (6)
$$

$$
\sigma_{2j+1}^{(i)} = \Big\{ \delta.\sigma_{2j+1}^{(i-1)} \quad +\Delta^{(i)}T_{2j}^{(i-1)}, \qquad for\, 0\leq j\leq t \qquad (7)
$$

$$
\Delta^{(i+1)} = \Delta_t^{(i+1)} + \Delta_{t+1}^{(i+1)} \qquad (8)
$$

where,

$$
\begin{aligned}
\Delta_{2j}^{(i+1)} &= S_{i-2j+2}.\sigma_{2j-1}^{(i)}+\Delta_{2j-2}^{(i+1)},\ for\, 0\leq j\leq t/2\\
\Delta_{2j+1}^{(i+1)} &= S_{i-2j+1}.\sigma_{2j}^{(i)}+\Delta_{2j-1}^{(i+1)}\ ,\ for\, 0\leq j\leq t/2 \qquad (9)
\end{aligned}
$$

From equation 9, the computation of $\Delta_{2j}^{(i+1)}$ and $\Delta_{2j+1}^{(i+1)}$ requires $\sigma_{2j-1}^{(i)}$, $\sigma_{2j}^{(i)}$, Δ_{2j-2}^{i+1}, and Δ_{2j-1}^{i+1}, which have been computed at cycle $(j-1)$. Similarly, from equations 6 and 7 at cycle j, the computation of $\sigma_{2j}^{(i)}$ and $\sigma_{2j+1}^{(i)}$ require $\Delta^{(i)}$ which has been computed at cycle 0 and $\sigma_{2j}^{(i-1)}$, and $\sigma_{2j+1}^{(i-1)}$ which have been computed at the $(i-1)$th step.

The proposed architecture shown in Figure 2 computes $\sigma(x)$ and $W(x)$ with latency t^2+4t clock cycles which makes the latency of our proposal lower than that of the serial [2] and the modified [4] architectures for all values of t. This enhancement leads to higher throughput and lower energy at the expense of a slight increase in area.

B. Efficient Computation Of Error Evaluator Polynomial $W(x)$

The conventional way to compute the error evaluator polynomial $W(x)$ using the Berlekamp-Massey algorithm is to do it after the computation of $\sigma(x)$ [1]. From the key equation and the Newton's identity we could derive $W(x)$ as follows [2]:

$$
\begin{aligned}
W(x) &= S(x)\sigma(x)\, mod\, x^{2t}\\
&= (S_1+S_2x+...+S_{2t}x^{2t-1})\\
&\quad .(\sigma_0+\sigma_1x+...+\sigma_tx^t)\, mod\, x^{2t}\\
&= W^{(0)}+W^{(1)}x+...+W^{(t-1)}x^{t-1} && (10)\\
W^{(i)} &= S_{i+1}\sigma_0+S_i\sigma_1+...+S_1\sigma_i, && (11)\\
i &= 0,1,...,t-1.
\end{aligned}
$$

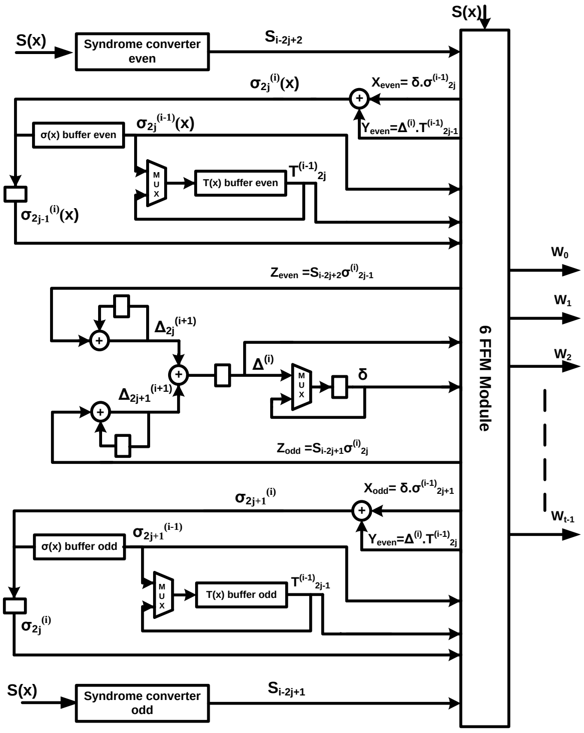

Figure 2. Implementation of the Two parallel decomposed inversionless Berlekamp–Massey algorithm

The computation of $W(x)$ can be performed directly after computation of $\sigma(x)$. Note that the direct computation requires fewer multiplications than the iterative algorithm which computes many unnecessary intermediate results, but it needs a lot of FFMs.

The proposed architecture suggests a 6 FFM implementation to evaluate $\sigma(x)$ and $W(x)$. The error locator polynomial is evaluated by using 6 FFMs. However, the error evaluator polynomial $W(x)$ reuses these 6 FFMs, as each $W^{(i)}$can be calculated in one clock cycle where $(i \leq 6)$ then the remaining two coefficients each one is calculated in two clock cycles. as shown in Figure 2. Compared to the previously proposed parallel architectures [5] our architecture reduces the hardware complexity significantly. Compared to a previously proposed serial and serial modified evaluator architecture [2, 4] respectively, our architecture reduces the latency significantly because of the reduction of number of clock cycles which transfer the bottelneck of the pipelined architecture to the two parallel syndrome circuit and for the same throughput, the proposed architecture need half the frequency of architecture [2], and $\sim$ 0.6 the frequency of architecture [4], which make our design more efficient in power consumption.

IV. Two Parallel RS Decoder architecture

In this paper a pipelined two parallel RS(204,188) decoder using two parallel Modified evaluator DiBM is presented. The decoder architecture consists of two parallel syndrome [3] block which calculates the syndromes from the received codewords in 102 clock cycles as shown in Figure 3(a). Figure 3(b) presents each two parallel syndrome cell. From the syndromes, the key equation solver (KES) block uses two parallel modified evaluator DiBM architecture to produce the error locator polynomial then the error evaluator polynomial as discussed in the previous section, the KES latency is 98 clock cycles. From the error locator and evaluator polynomials a Chien search algorithm is used to produce the error locations as shown in Figure 4(a). The two parallel Chien search [3] circuit is used and Figure 4(b) presents the two parallel Chien search cell with latency 102 clock cycles, but in each clock cycle a new codeword is corrected, then Forney algorithm is used to calculate the error values. The two blocks are combined in one block as an error corrector block and Figure 4(c) presents the circuit diagram of the complete error corrector.

The bottleneck for this architecture is in the syndrome computation block as the two parallel syndrome circuit needs 102 clock cycles and the KES latency is 98 clock cycles and each Chien search and Forney need only one clock cycle to correct one codeword, therefore the total latency of our architecture will be 202 clock cycles.

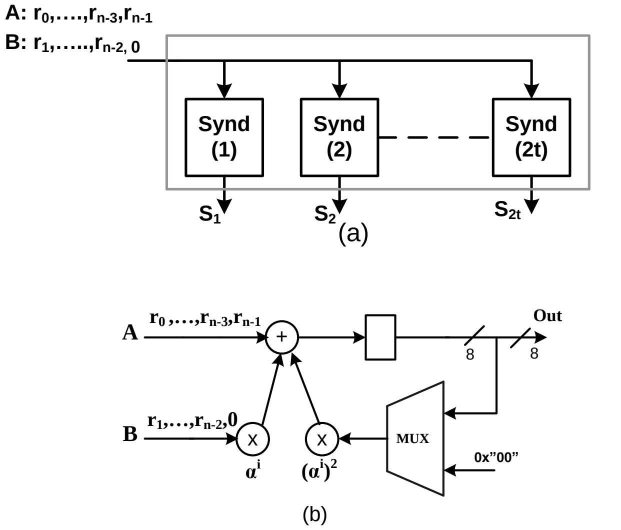

Figure 3. Syndrome Circuit

V. Results and Comparison

The architecture was modeled in VHDL 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 on $0.13\mu m$ CMOS technology and optimized for a $1.2V$ supply voltage, we used this technology to make our comparison fair with the previously published architectures. The total number of gates for the proposed decoder is $33,000$ from the synthesized results excluding the FIFO memory, and the clock frequency up to $660MHz$. Simulation results show this approach can work successfully at the data rate $100\,Mbps$ with power dissipation of $0.266\,mW$.

Table I shows a comparison between different architectures of RS$(255, 239)$ decoders. It is clear from the table that architectures [2], [9], [10], [11], and [12] have smaller area than the proposed architecture, but the proposed architecture has higher throughput so we can define another parameter that shows the value of the proposed design which is its "Efficiency". It is defined as follows:

Efficiency = (throughput / # Gates).

A higher efficiency is better, as it comes from higher throughput and lower area. These results show that the proposed design is much better than most designs, The only design that has higher efficiency than the proposed design is [2]. Table II shows specific comparison between the proposed architecture and [2] in terms of power consumption for a constant throughput

Table I
IMPLEMENTATION RESULTS OF THE RS(255,239) DECODERS

Architecture	Technology (μm)	Total # of Gates	Clock (MHz)	Latency (clocks)	Latency (ns)	Throughput (Mb/s)	Efficiency
Proposed	0.13	33,000	660	226	434.6	10,500	0.318
[2]	0.13	15,000	700	475	679	5,600	0.373
[4]	0.13	37,600	606	298	491.7	7,357	0.19566
[6]	0.18	49,200	200	512	2560	1,550	0.0315
[7]	0.13	53,200	660	355	537.9	5,300	0.09962
[8]	0.13	44,700	300	287	956.7	2,400	0.05369
[9]	0.18	20,614	400	512	1280	3,200	0.15523
[10]	0.18	18,400	640	519	811	5,022	0.273
[11]	0.13	24,600	625	513	820	5000	0.203
[12]	0.18	20,614	400	513	1283	3,200	0.155

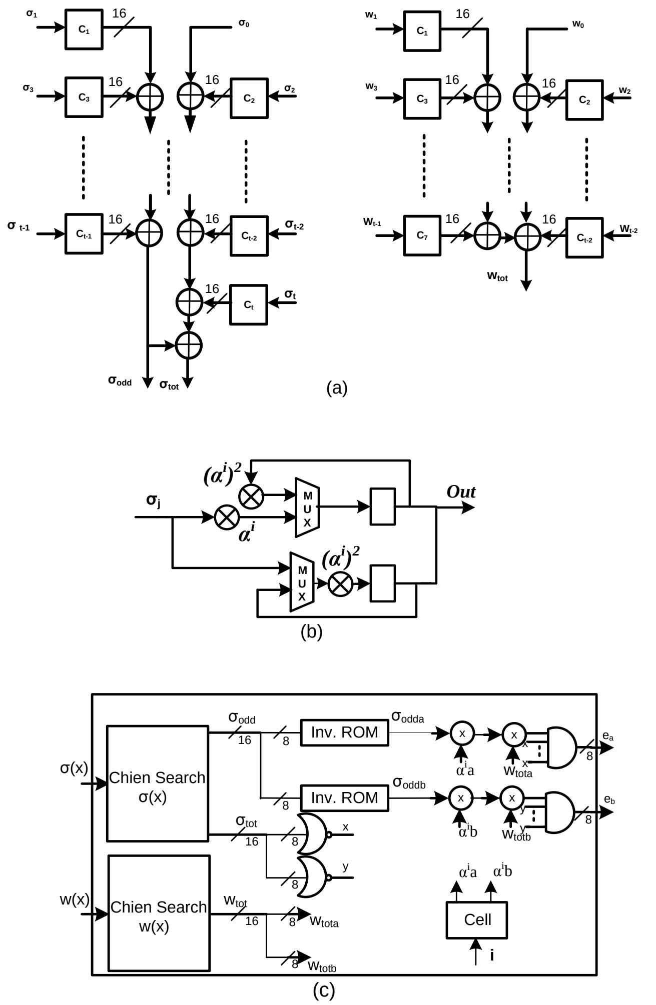

Figure 4. Error Corrector Block

$100\,Mbps$ to be suitable for DVB-H application. It is clear that the proposed architecture in terms of power consumption is lower than [2] by 17% and this is very useful for the handheld devices in the DVB-H standard.

Table II
PERFORMANCE COMPARISONS FOR CONSTANT RATE $100Mbps$

Technology (μm)	architecture [2]	proposed
# Gates	15000	33000
Latency (clk cycles)	425	204
clk (MHz)	12.5	6.25
power (μW)	320	266

VI. Conclusion

This paper presents a new architecture for a low energy high-speed pipelined two parallel RS(204, 188) decoder. In this proposed architecture a two parallel syndrome and Chien search circuits are used. The KES block includes 6 FFMs which make our design between the serial architecture which uses 3 FFM and the parallel architectures which uses multiples of t FFMs. These 6 FFMs are scheduled in a clever way to lower the latency of the KES with a slight increase in area. This scheduling of multipliers has reduced the energy per symbol significantly. We have investigated hardware gate count, throughput, and energy per symbol for RS decoders. It is clear that the proposed architecture has the lowest latency and highest throughput compared to previous architectures. So our architecture optimizes the latency, throughput, and power consumption.

References

[1] R. E. Blahut, "Theory and Practice of Error Control codes", Addison Wesley, 1983.

[2] H. Chia Chang and C. Shung, "New Serial Architecture for the Berlekamp–Massey Algorithm," IEEE Trans. on communications, vol. 47, no. 4, April 1999.

[3] S. Lee,Chang-Seok Choi, and H. Lee, "Two-parallel Reed-Solomon based FEC architecture for optical communications," IEICE Electronics Express, May. 2008.

[4] H. Ahmed, H. Salah, T. Elshabrawy, and H. A. H. Fahmy, "A low Energy High Speed Reed-Solomon Decoder using Decomposed Inversionless Berlekamp-Massey Algorithm," Proceedings of IEEE Asilomar conference on Signals, Systems and Computers, 2010.

[5] T. Park, "Design of the (248,216) Reed-Solomon Decoder with Erasure Correction for Blu-ray Disc," IEEE Trans. on Consumer Electronics, vol. 51, no. 3, August 2005.

[6] Y. Lu, M. Shieh, "Low-complexity Reed-Solomon decoder for Blu-ray Disc applications," Proceedings of the IEEE International Symposium on VLSI Design Automation and Test (VLSI-DAT), 2010.

[7] S. Lee, H. Lee, J. Shin and J. Ko, "A High-Speed Pipelined Degree-Computationless Modified Euclidean Algorithm Architecture for Reed-Solomon Decoders," IEEE International symposium on Circuits and System (ISCAS), 2007.

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

[9] H. Yi Hsu, A. Yeu (Andy) Wu, and J. Yeo, "Area-Efficient VLSI Design of Reed–Solomon Decoder for 10GBase-LX4 Optical Communication Systems," IEEE Trans. on Circuits and Syst. II, Exp. Briefs, vol. 53, no. 11, November 2006.

[10] B. Yuan, Z. F. Wang, L. Li, M. L. Gao, J. Sha, and C. Zhang, "Area efficient Reed-Solomon Decoder Design for Optical Communications," IEEE Trans. on Circuits and Syst. II, Exp. Briefs, vol. 56, no. 6, June 2009..

[11] H. Lee, "A High-speed Low-complexity Reed-Solomon Decoder for Optical Communications," IEEE Trans. Circuits and Syst. II, Exp. Briefs, vol. 52, no. 8, August 2005.

[12] H. Y. Hsu, A. Y. Wu and J. C. Yeo, "Area-efficient VLSI Design of Reed-Solomon Decoder for 10GBase-LX4 Optical Communication Systems," IEEE Trans. Circuits and Syst. II, Exp. Briefs, vol. 43, no. 4, November 2006.