Programmable CRC Computation on High Speed Parallel LFSR Architectures based on Improved State Space Structure

Roshan Anna Joseph[1]
Department of Electronics & Communication Engineering
Saintgits College of Engineering,
Kottayam, India

Hanna Mathew[2]
Department of Electronics & Communication Engineering
Saintgits College of Engineering,
Kottayam, India

Abstract— **Linear Feedback Shift Register (LFSR) is a shift register with its input bit as a linear operation of its previous state. Most commonly, this function is a Boolean exclusive OR(XOR) .LFSRs are as such used in cyclic error correcting codes like BCH and CRC encoders. Parallelization of LFSR increases the system throughput rate, thus makes it useful for high-speed communication. State-space transformation technique reduces the circuit complexity of parallel architectures and the encoder consists of matrix multiplications, with searching of transformation matrices. The new technique is the construction of a modified transformation matrix and the corresponding searching algorithms for the desirable transformation matrix which has both speed and area advantages. The register with a well chosen feedback produces sequence of bits which exhibits psuedo-random behavior thus making it applicable for very fast generation of pseudo-random sequence such as in cryptography, digital broadcasting, fast counters, test-pattern generation, radio jamming systems etc.**

Keywords*— *Component, BCH, CRC, LFSR, state-space transformation.

INTRODUCTION

In digital processing, a Linear Feedback Shift Register (LFSR) is a shift register whose input bit is a direct capacity of its past state. The most generally utilized straight capacity of single bits is exclusive OR (XOR)[1]. Accordingly, an LFSR is regularly a shift register whose input bit is driven by the XOR of certain bits of the general register blocks. LFSRs are widely applied as such in BCH and CRC encoders, generally to compute the remainder polynomials. A traditional encoder or a sequential LFSR architecture can work at extremely high frequency, but it experiences the innate sequential in and sequential out restriction. At the point when the throughput of this sequential/serial architecture can't get up with the framework information rate, parallel processing must be considered to meet the general necessities of fast interchanges and will provide a higher throughput rate. For example, as in optical transmission, rates beyond Gigabits per second are needed.[2]

Many sorts of parallel LFSR structures have been proposed in the related writings for BCH and CRC encoders. In [3], a parallel CRC encoder has been proposed through scientific finding. Tree-organized calculation and sub-expressions sharing are used to improve the cascaded rationale parts. Zhang and Parhi [4] proposed an improved fast BCH encoder intended to dispense with the fanout bottleneck condition. This parallel LFSR is effective as far as it accelerates the calculation, but its equipment cost is high. A sort of state-space transformation technique is applied in [5] and [6] to decrease the multifaceted nature of the traditional parallel CRC circuits. By embracing straight network change, a full speedup factor can be accomplished at the expense of an extra hardware outside the criticism circle. Parhi [7] and Jung [8] proposed another sort of parallel LFSR dependent on IIR channel topology what's more, its leverage is that the pipeline method can be connected to accomplish some improvement in the equipment proficiency of LFSR.

In state-space transformation, the encoder is made out of some grid multiplications, with the most exertion committed to searching of transformation networks that may accomplish the minimal-area circuits. In this short, another change matrix development and an inexact searching strategy are proposed. Utilizing this surmised calculation, the transformation can be found in an a lot shorter time than thorough searching. A low-multifaceted nature and rapid parallel LFSR engineering can be inferred by applying the state-space change with the alluring arrangement. Accordingly, the equipment complexity can be viably decreased without yielding execution or speed

The initial value of the LFSR is called the seed and since the operation of the register is deterministic, the stream of values produced by the register is completely created by the register is totally dictated by its current (or past) state. Similarly, in light of the fact that the register has a finite number of possible states, it should inevitably enter a repeating cycle. In any case, an LFSR with a well-picked feedback function can produce a sequence of bits that shows up random nature and has a long cycle. The mathematics of cyclic redundancy check, utilized to give a brisk check against transmission errors, are firmly identified with those of an LFSR.[5] LFSRs can be implemented in hardware, and this makes them helpful in applications that require a quick generation of a pseudo-irregular and random sequence. LFSRs have additionally been utilized for producing an estimate of repetitive sound in different programmable sound generators. [8] Complete LFSR are commonly used as pattern generators for exhaustive testing, since they cover all possible inputs for an n-input circuit. In built-in self-test (BIST) techniques, it is accomplished with a multiple-input signature register (MISR or MSR), which is a type of LFSR.

In this concise, a new transformation matrix development and an approximate looking strategy are proposed. Utilizing this estimated calculation, the desirable transformation matrix framework can be found in a lot shorter time than thorough pursuits. A low complexity and rapid parallel LFSR architecture can be inferred by applying the state-space change with the new arrangement. Therefore, the hardware complexity can be viably decreased without relinquishing execution or speed.The rest of this brief is organized as follows. In Section I,the basic serial and parallel LFSR designs are reviewed briefly. State-space transformation and IIR filter based architectures are mentioned as parallel schemes. Section II briefs about the basic searching algorithm for the state-pace method the and problems with existing architecture. Section III introduces a modified algorithm for the transformation and CRC check for the modified transformation matrix, and the corresponding searching algorithms. Section IV gives the experimental results and makes a comparison. Conclusions are drawn in Section V.

I. BASIC LFSR ARCHITECTURES

LFSRs can be implemented based on serial and parallel architectures. The initial value of the LFSR is called the seed and since the operation of the register is deterministic, the stream of values produced by the register is completely created by the register is totally dictated by its current (or past) state. Similarly, in light of the fact that the register has a finite number of possible states, it should inevitably enter a repeating cycle.The mathematics of cyclic redundancy check, utilized to give a brisk check against transmission errors, are firmly identified with those of an LFSR.[5]Few LFSR architectures are discussed below.

A. Serial LFSR Architecture

A conventional serial LFSR architecture for BCH(n, k) code is illustrated in figure 1. There is XOR operation and finite-field multiply operation.[1] In binary codes, the coefficient of the generator polynomial equals either 0 or 1, then the corresponding multiply can be simplified to a connection or disconnection. Registers represent the remainder polynomial.Two kinds of typical parallel LFSR architectures are introduced next.

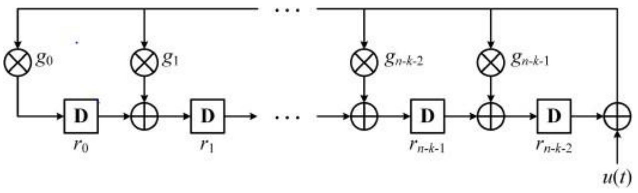
g_0
g_1
$g_{n\text{-}k\text{-}2}$
$g_{n\text{-}k\text{-}1}$
D
D
D
D
r_0
r_1
$r_{n\text{-}k\text{-}1}$
$r_{n\text{-}k\text{-}2}$
$u(t)$

Figure 1: Conventional serial LFSR architecture

Although a serial LFSR architecture can operate at very high frequency, it suffers from the inherent serial-in and serial-out limitation.[4][5] When the throughput of this serial architecture cannot catch up with the system data rate, parallel processing must be considered to meet the general requirements of high-speed communications and realize a higher throughput rate. There are many kinds of parallel LFSR architectures. A kind of state-space transformation can reduce the complexity of the conventional parallel CRC.

B. Parallel LFSR Architecture based on state-space transformation

In state-space transformation, the encoder is composed of some matrix multiplications, with the most effort devoted to searching of transformation matrices that may achieve the minimal-area circuits.[1] In this brief, a new transformation matrix construction and an approximate searching method are proposed. Using this approximate algorithm, the desirable transformation matrix can be found in a much shorter time than exhaustive searches. A low-complexity and high-speed parallel LFSR architecture can be derived by applying the state-space transformation with the desirable solution. As a result, the hardware complexity can be effectively reduced without sacrificing performance or speed.[5] Supposing an n-k dimensional state vector $R(t) = [r_{n-k-1}(t), r_1(t), r_0(t)]$, where $r_{i(}t)$ represents the state of the ith remainder registers at time t and $u(t)$ represents the tth single-bit input. $R(t + 1)$ can be expressed as $R(t + 1) = A * R(t) + b *u(t)$ where matrix A is

$$\mathbf{A} = \begin{pmatrix} g_{n-k-1} & 1 & 0 & \cdots & 0 \\ g_{n-k-2} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ g_1 & 0 & 0 & 0 & 1 \\ g_0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

and vector b is $b = (g_{(n\text{-}k\text{-}1)}, \ . \ . \ . \ , g_1, g_0)^T$.$R(t+1)$ can be derived from recursive deductions. Assuming p message bits are input at the same time,

$$R(t + 1) = Ap * R(t) + Bp * U\,p(t).$$

In the case of parallel input, where matrix Bp is an $n - k$, p matrix,

$$Bp = (A_{(p-1)}b, ..., Ab, b).$$

To reduce the complexity of the system,a linear transformation of R(t) is applied through a non singular rmatrix T, $R(t) = T * R_T(t)$.Then the state equation can be rewritten as ,

$$R_T(t + 1) = Ap_T * R_T(t) + Bp_T * U_p(t)$$

where,

$$R(t + 1) = T * R_T(t + 1)$$

$$A_{pT} = T^{-1} * A^p T$$

$$B_{pT} = T^{-1} * B_p.$$

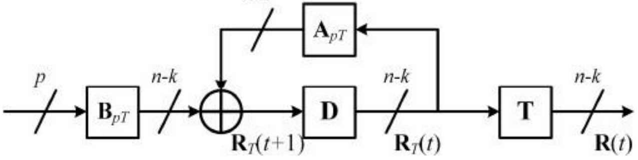

Figure 2: Block diagram for parallel LFSR architecture.

The figure 2 shows block diagram of the system described by this method. The transformation matrices are chosen such that matrix A_{pT} is a companion matrix, which will simplify the feedback loop of the parallel architecture. Unfortunately, when A_p is transformed into a companion matrix A_{pT} , matrix T and B_{pT} may become complicated and dense with many 1s. Even the feedback loop is fast and of low complexity as expected, the other parts of the encoder may have a longer critical path with high complexity. After applying pipelining and retiming techniques to reduce the critical path, the data-path timing is still not satisfying and brings extra hardware cost as well.To further optimize the hardware implementations with state-space transformations and to find a better transformation matrix, some constraints are needed to be defined.

- The transformation matrix T must be reversible to make the state-space transformation workable.
- The total number of 1s in matrices T , A_{pT} , and B_{pT} should be as small as possible. The number of 1s in these matrices determines the number of XOR gates of the encoder directly.
- The method to find the transformation matrix needs to be efficient. The whole searching space is too huge for exhaustive search.

Based on these three constraints, a new method to construct the transformation matrix is introduced in the next section.

C. Parallel LFSR Architecture based on IIR filter method

A kind of parallel LFSR architecture can be implemented based on IIR filter, to mitigate the impractical hardware complexity of the parallel architecture.[7] The serial LFSR encoder shown in figure 1 is treated as an IIR filter, and the loop update equations can be derived by adopting the look-ahead scheme. A parallel BCH encoder can be implemented as in figure 3 based on these equations. The past input and output values are combined to generate the output y(n).[7]

.

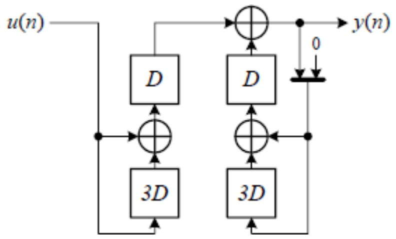

Figure 3: Example for an LFSR using IIR filter method

II. BASIC SEARCHING ALGORITHM FOR STATE-SPACE MODEL OF LFSR

Few CRC or BCH codes are to be referenced . Their generator polynomials are to be considered. In order to find the desirable transformation matrix T, an exhaustive search is performed in the vector space of b. The parallel level p or the input size is chosen as the degree of generator polynomials for a clear comparison with previous architectures. The proposed method can be applied at any parallel level. Since the matrices T, A_{pT}, and B_{pT} actually imply the connection of the coupling circuits, it follows that the hardware complexity is related to the total number of 1s in these three matrices, which is denoted as TN in the following. Consequently, it is needed to find a vector $b2$ that minimizes TN. The basic searching algorithm to be used is as follows.

- First, the coupling matrices A and B_p can be computed for the generator polynomials with input size p = nk.
- Then traverse all the possible values of b2 to construct transformation matrix T, and subject it to state-space transformation to obtain transformed coupling matrices A_{pT} and B_{pT} .
- Afterward, the TN can be counted. The least one indicates the lowest complexity of hardware and the corresponding vector b2 can be regarded as the best solution. TN is regarded as the number of XOR gates.
- For n − k = 32 or more, the vector space of b is too enormous and this exhaustive searching algorithm requires more than one month to obtain the results. This restricts the application of this searching algorithm for high-order generator polynomials.

A. Problems With The Existing Architecture.

The transformation matrix used in [4] is chosen such that matrix A_{pT} is a companion matrix, that simplifies the feedback loop of the parallel architecture. When A_p is transformed into a companion matrix A_{pT}, matrix T and B_{pT} may become complicated and dense with many 1s. Even the feedback loop is fast and of low complexity as expected. The other parts of the encoder may have a longer critical path with high com- plexity. After applying pipelining and retiming techniques to reduce the critical path, the data-path timing is still not satisfying and brings extra hardware cost as well. As well the number of clocks needed will be much larger and only synchronous clock could be provided for an optimal output. This difference indicates that only a minority of possibilities of T are considered if using b to generate T. The transformation matrix derived from optimal b may not be the optimal one among all the possibilities of T [11]. Since there are different kinds of matrices that may change the circuits into increasingly proficient structures, a few endeavors can be committed to improve the technique for developing matrix T and to additionally advance the equip- ment usage with state-space changes. So as to locate a superior change network, a few imperatives need to be characterized here. To start with, the change in matrix T must be reversible to make the state-space change useful. [5] Second, the all out number of 1s in networks T , A_{pT} , and B_{pT} ought to be as little as could reasonably be expected. The number of 1s in these networks decides the quantity of XOR entryways of the encoder straight forwardly. Third, the strategy to observe the change framework should be effective. The entire scanning space is unreasonably tremendous for comprehensive look [11].

In view of these limitations, another strategy to choose the transformation matrix is proposed as pursues. In this, encoder is composed of some matrix multiplications. Most effort is devoted to the searching of transformation matrices. and it helps to achieve the minimal-area circuits. An alternative searching algorithm is designed to obtain the desired transformation matrix more quickly.

III. MODIFIED SEARCHING ALGORITHM FOR STATE-SPACE MODEL OF LFSR

Based on the approximate searching algorithm a modified state space model can be implemented with a different choices for T matrix. The generation of T matrix is an area of research. Efficient choice of T matrix yields in generating A_{pT} B_{pT} transformed matrices very faster. The generated T must be reversible. Number of 1s in matrices be small, which in turn determines number of XOR gates in the architecture. So the method to find *T* needs to be efficient. The table 4.1 shows a few suggestions for *T* matrix and corresponding values for A_{pT} and $B_{PT.}$

Table 1: Few suggestions for T Matrix

Choice1	Choice2	Choice3
T=I $A_{pT} = A^p$ $B_{pT} = B_p$	$T=[A^p]^{-1}$ $A_{pT} = [A^p]^{-1}$ $B_{pT} = A_p B_p$	$T=[B_p]^{-1}$ $A_{pT} = I$ $B_{pT} = B_p A^{p-1}$

A. *Implementation of the Proposed Method In CRC Encoding*

Linear feedback shift registers (LFSRs) are applied as such in BCH and CRC encoders to compute the remainder polynomial. Many types of parallel LFSR architectures have already been presented for BCH and CRC encoders. In this, a parallel CRC implementation designed through mathematical deduction using the modified state-space model. Tree-structured computation and sub-expressions are used to optimize the cascaded logic parts[12].

Table 2: Generator polynomials referenced for the programmable CRC.

CRC Name	Polynomials
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + 1$
CRC-16	$x^{16} + x^{15} + x^2 + 1$
CRC-32	$x^{32} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

B. *Programmable CRC for16 and 32bit*

The generator polynomial CRC-16 and CRC-32 is given in table 2. A CRC-16 bit code, detects single and double bit errors in the message. The burst errors are of length 16 bit or less. All errors are with an odd number of bits and most errors are for longer bursts. It is used in Ethernet IEEE 802.3 standard. Remainder polynomial is computed to min- imize errors, get random values and lengthy pattern. The initial value, called as the seed, can be chosen to all 1 or 0.At the reset condition every flip-flop must be set to high.

Table 3: CRC-16: Hexadecimal output of some random values.

Input	CRC-Output
0000ABCD	F8A4
0000E3DD	48C2
0000A82D	70E7
00008A25	3CD4
000042CD	8EA8
00003DFD	8C0E

Table 4: CRC-32: Hexadecimal output of some random values.

Input	CRC-Output
ABCD1234	3005573B
1234ABCD	5C380B83
AFCF1374	A9DEF39B
1A35ABD3	D4954474

A programmable CRC checker is implemented using the modified State-Space trans- formation model which selects between CRC-16 and CRC-32.Asynchronous clock is given for the reset condition, and for the modulo-2 operations. The implemented CRC checker generates the CRC coded output for each input bits. The tables shows the corresponding CRC outputs of 16 and 32 bit CRC input values. The values were cross- checked using Online CRC Calculator.

IV. RESULTS AND DISCUSSION

The proposed, programmable CRC computation on high-speed parallel LFSR architectures based on improved state space structure have been designed using Xilinx. The result of simulation is shown in the following sections. For the experimental evaluation, a Spartan3a Starter board xc3s700a fg484 was used. The Verilog code of the modified State-Space methodology was generated and simulated with the Model Sim PE 5.5e to extract the output codes for the CRC bits. Following simulation, methodology is synthesized with ISE Design Suite 14.7. The Xilinx XPower analyzer was then used to determine power consumption, using the design netlist, the design constraints, and the simulation activity.

A. *Simulation Results*

The output values of CRC-16 and CRC-32 obtained through the simulation of Verilog code was cross-checked with values in table 4.5 which was obtained from online CRC calculator(www.ghi.de).

B. *Analysis of the modified programmable crc with existing architectures*

Outputs of CRC checksum were cross-checked using Online CRC checker for different values. Synthesis results shows this architecture outperforms the previous designs. Post place & route timing analysis shows that the clock needed to setup on destination clock=2.461 ns. Maximum clocking frequency, assuming 50 % duty cycle, T= 2.461nS was found to be 406.33888 MHz . CRC computations per second, assuming 4 clocks per CRC computation is 101584721 CRC. Throughput, assuming 4 clocks per CRC computation is calculated as 3.0274Gbps. Latency time (assuming 4 clocks per CRC computation),time taken for the input to reach the output is found as 9.844nS. Number of clocks needed were very less compared to the conventional structures. Asynchronous clock was given for the programmable CRC. Clocks were provided for the reset condition and loading the transformation matrix values. Delay was significantly reduced in the modified programmable structure. The table 5.2 shows the delay, number of clocks, and power analysis of the existing architectures and the modified programmable CRC.

Table 5: Analysis of various LFSR architectures with the modified architecture using CRC Check.

Model	CRC-16 series	CRC-32 series	CRC-16 parallel	CRC-32 parallel	PCC Using Modified SST
Delay	8.158 ns	15.705 ns	2.885 ns	4.795 ns	5.202 ns(total)
Clock (no.s)	17	33	4	4	4 (asynch.)
Power	73.62 mW	92.41mW	13.73 mW	17.42 mW	20.35 mW

V. CONCLUSION

The A linear feedback shift register (LFSR) is the shift register including input bit which is the direct XOR operation of its old block. The LFSR including well-chosen feedback operation is able to generate the series of bits that looks random that includes lengthy cycle. Through this project the basic serial and parallel LFSR designs were reviewed and analyzed. The parallel architectures can achieve hardware designs with the minimal XOR gates and the same CPD. LFSRs can be implemented in hard- ware, and this makes them useful in applications that require very fast generation of a pseudo-random sequence, such as direct-sequence spread spectrum radio. LFSRs have also been used for generating an approximation of white noise in various programmable sound generators. The LFSR including well-chosen feedback operation is able to generate the series of bits that looks random that includes lengthy cycle. Modified state-space transformation architectures and the corresponding searching algorithms were introduced.

Through this project a modified method to choose the transformation matrix in the state-space model was computed. A CRC check for the modified transformation ma- trix, and the corresponding searching algorithms were done. A programmable CRC checker was implemented based on the modified State-Space transformation technique of parallel LFSR. Based on this improved model, reduced-complexity parallel architectures is obtained. Experimental results shows that the throughput of the modified parallel architecture is much higher than those of serial LFSR architecture. Choosing of the transformation matrices is an area of research. Number of clocks were reduced which reduces the overall delay. There is a deterministic improvement in the throughput and speed and a significant reduction in power consumed.

REFERENCES

[1] G. Hu, J. Sha and Z. Wang,”High-Speed Parallel LFSR Architectures Based on Improved State-Space Transformations”,*IEEE Transaction on Very Large Scale In- tegration (VLSI) Systems*,vol.25, 1159-1163,2017 .

[2] M. Y. Hsiao and K. Y. Sih, ”Serial-to-parallel transformation of linear-feedback shift-register circuits”,*Trans. Electron. Comput.*,vol.6,738-740,2007.

[3] M. Ayinala and K. K. Parhi,”High-speed parallel architectures for linear feedback shift registers”,*IEEE Trans. Signal Process.*,vol.59,4459–4469,2011. [8] M. Ayinala and K. K. Parhi, “Efficient Parallel VLSI Architecture for Linear Feedback Shift Registers,” in Proc. IEEE Workshop on SiPS, Oct. 2010, pp. 52–57.

[4] J. Jung, H. Yoo, Y. Lee, and I.-C. Park,”Efficient parallel architecture for lin- ear feedback shift registers”, *IEEE Trans. Circuits Syst. II, Express Briefs*,vol.62, 1068–1072,2015.

[5] K. K. Parhi, ”Eliminating the fanout bottleneck in parallel long BCH en- coders”,*IEEE Trans. Circuits Syst. I, Reg. Papers*,vol.51,512-516,2004.

[6] T.-B. Pei and C. Zukowski,”High-speed parallel CRC circuits in VLSI”, *IEEE Trans.Commun.*,vol.40,653–657,2004.

[7] K. K. Parhi and D. G. Messerschmitt, ”Pipeline interleaving and parallelism in re- cursive digital filters. II. Pipelined incremental block filtering”,*IEEE Trans. Acoust., Speech, Signal Process.*,vol.37,1118–1134,2000.

[8] J. H. Derby,”High-speed CRC computation using state-space transformations”, *in Proc. IEEE GLOBECOM,* 166–170,2001.

[9] X. Zhang and K. K. Parhi, ”High-speed architectures for parallel long BCH en- coders”,(IEEE Trans. Very Large Scale Integr. VLSI Syst.),13,872–877,2005.

[10] C. Kennedy and A. Reyhani-Masoleh, ”High-speed parallel CRC circuits”, (in Proc. 42nd Annu. Asilomar Conf. Signals, Syst., Comput),1823–1829,2008.

[11] L. Song, M.-L. Yu, and M. Shaffer, “10- and 40-Gb/s forward error correction de- vices for optical communications”,IEEE J. Solid-State Circuits,11,1565–1573,2002.

[12] Y.-M. Lin, C.-H. Yang, C.-H. Hsu, H.-C. Chang, and C.-Y. Lee, “A MPCN-based parallel architecture in BCH decoders for NAND flash memories” ,*IEEE Trans. Cir- cuits and Sys. II*,58,682-686,2011.

[13] H.-C. Chang, C.-Y. Lee, Y.-M. Lin, and C.-H. Yang, “Apparatus and method of processing cyclic codes”, *U.S. Patent 20 110 292 681 A1*,12, 2011.

[14] H. Chen, “CRT-Based high-speed parallel architecture for long BCH encoding”,IEEE Trans. Circuits Syst. II, Exp. Briefs,56,684–686,20

[15] R. Cherukuri, “Agile encoder architecture for strength-adaptive long BCH codes”,in Proc. IEEE GLOBECOM,1900–1904,2010.

[16] S. Lin and D. J. Costello, ”Error control coding: Fundamentals and Applica- tions”,*2nd ed. Englewood Cliffs, NJ: Prentice-Hall Inc.*,2004.

[17] H. Yoo, J. Jung, J. Jo, and I.-C. Park, “Area-Efficient Multimode Encoding Ar- chitecture for Long BCH Codes”,*IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 60*,12,872–876,2013.

[18] S. Gregori, A. Cabrini, O. Khouri, and G. Torelli, ”On-chip error correction tech- niques for new-generation flash memories”,*Proc. IEEE*,91,602–616,2003.

[19] M. Sprachmann,“Automatic Generation of Parallel CRC Circuits”,*IEEE Des.*Test. Comput., vol. 18,3,108–114,2001.

[20] J. Zhang, Z. Wang, Q, Hu, and J. Xiao, “Optimized Design for High-Speed Parallel BCH Encoder”, *in Proc. IEEE Int. Workshop VLSI Des. Video Tech- nol.(IWVDVT)*,179–182,2005.