Architecture of Parallel CRC Encoder using State Space Transformations

Lekshmi S Prabha
Department of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, India
Email: lekshmisprabha@gmail.com

Geethu R S
Department of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham
Amritapuri, India
Email: geethu.amrita@gmail.com

***Abstract*—Linear Feedback shift Registers (LFSRs) are widely used in encoders like Cyclic Redundancy Check (CRC) for generating error detecting codes. In order to achieve high speed communication, parallel processing is performed in the serial CRC. Since this method increases the circuit complexity, the speed gets limited. State space transformation is a method that can be used to reduce the circuit complexity. Therefore, an efficient transformation matrix is needed in this method. In this paper, a method to construct a transformation matrix and an approximate searching algorithm to generate certain vectors, which are used in transformation matrix are implemented.**

***Keywords*—LFSR, CRC encoder, Parallel CRC, State Space Transformation, Approximate Search Algorithm.**

I. Introduction

Cyclic codes have wide use in communication systems since they exhibit good error detecting and correcting properties. Cyclic Redundancy Check and Bose Chaudhuri Hocquenghem codes[1] are the least difficult and most regularly used cyclic codes.

CRC generation can be implemented using the basic Linear Feedback Shift Registers (LFSRs)[2],[3],[4]. Their simple structure is suitable for working at higher clock pulses. However, they experience the constraint that the data stream must be bit-serial which restricts their throughput to keep up with the system data rate. In order to achieve high speed communication and high throughput, parallel implementation of CRC circuits[5],[6] can be adopted. In parallel implementation[7] the input message is divided into blocks containing equal no.of bits. No.of blocks depends on the no.of bits given into the parallel structure at a time.

In serial CRC structure, the computational time is equal to k clock cycles, where k is the no.of message bits. By adopting such a parallel architecture[8], CRC can be calculated in k/m clock cycles, where m is the no.of inputs given into the architecture at a time. Such an architecture can increase the throughput and allows high speed communication. But, the increase in hardware cost and longer critical path leads to area complexity. As the critical path length increases, the speed of operation of the circuit also diminishes and hence the throughput rate accomplished by the parallel architecture will get reduced. Pipelining can not be applied to these circuit to reduce the critical path, since they contain feedback paths.

In order to reduce the hardware complexity, state space transformation[9],[10] is proposed. They can shift the complexity out of the feedback. Hence a speed-up can be achieved at cost of an additional circuitry. This additional circuitry outside the feedback loop can be pipelined, and a parallel architecture having area complexity less than the parallel CRC structure can be achieved.

In the state space transformation, the CRC encoder is represented using matrix multiplications. In such representation, searching a suitable transformation matrix needs more effort. Construction of transformation matrix needs an efficient algorithm also. If the degree of generator polynomial is n, then the dimension of transformation matrix will be n×n. Transformation matrix is constructed using an n×1 dimensional vector c. Using approximate search[11], different c vectors are generated and an efficient c vector is selected and transformation matrix is constructed. The c vector must be picked with the end goal that transformation matrix is non-singular. In this work an approximate search algorithm is implemented to generate a set of c vectors and transformation matrix is constructed using each c vector. Then from that, an efficient transformation matrix is chosen in order to achieve a less complex and high throughput parallel CRC architecture.

The paper is organized as follows: section I contains the introduction and section II contains the background study. Section III describes the methodology adopted and IV discusses the implementation and results. Section V discusses the conclusion.

II. Background Study

A. Serial CRC

A serial CRC [12] contains shift register and a feedback path. The position of xor gates is defined by a generator polynomial. The generator polynomial will be in the form, P(x)=$p_0 + p_1x^1 + ... + p_nx^n$ where n corresponds to the no.of flip flops in the CRC circuits. The coefficients p_n=1 and p_n=0 decides the connection and disconnection between the xor and feedback path. The state of the flip flops is represented as $C_0(t)$, $C_1(t)$, ..., $C_n(t)$, where $C_i(t)$ represents the state of i^{th} flip flop.

The CRC, C(x) is calculated by dividing $x^nK(x)$ by generator polynomial P(x). CRC is appended with message sequence in order to identify transmission errors.

Fig. 1 shows the architecture of a serial CRC. A serial CRC processes one bit at a time. A k bit message is processed and

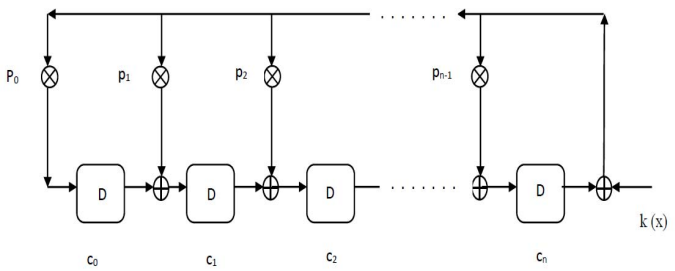

Fig. 1. Serial CRC

CRC value is obtained after k clock cycles. Speed requirement is not met in serial CRC in the case of high speed communication. The net propagation delay, t_p around the feedback path comprises of 2 xor gates which limits the throughput. The maximum available throughput will be $1/t_p$. Serial CRC experiences the constraint that the data stream must be bit-serial, which restricts their throughput to keep up with the system data rate.

B. Parallel CRC

In order to achieve high speed communication[13], a parallel arrangement of serial CRC is used. In this the message is divided into blocks of length k/m bits each. When comparing serial CRC, a parallel CRC can calculate the CRC in k/m clock cycles, and hence throughput can be increased.

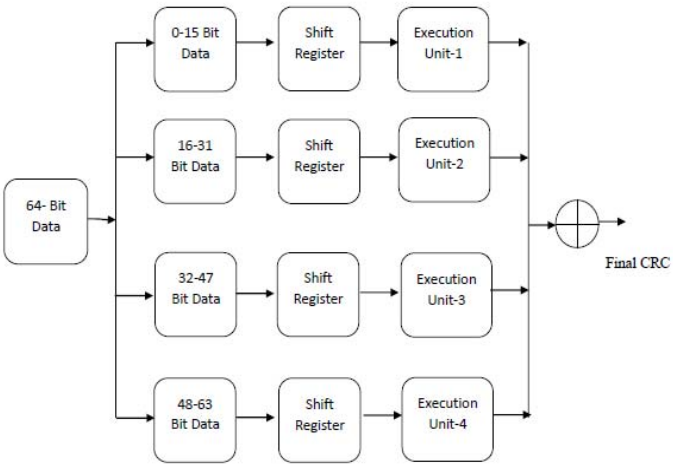

Fig. 2. Parallel CRC

In Fig. 2, a parallel CRC structure is shown which receives 64 bit message (k=16), which is divided four blocks (m=4). For processing the message four execution units are there, into which each message block is given. After 16 clock cycles (k/m=64/4=16) four 16 bit CRC values (if degree n of generator polynomial is 16) will be obtained. The final 16 bit CRC will be the xor of these four CRC values. Parallel processing increases the throughput rate as well as the no.of bits that can be processed at a time. But this architecture causes critical path to increase and hence circuit speed to decrease. The hardware cost also increases, which introduces a complexity in the structure.

C. State Space Transformation

The complexity of parallel CRC architecture can be reduced by applying a kind of state space Transformation. In state space transformation, the encoder consists of some kind of matrix multiplications, and in that transformation matrix, is to be efficient. Using linear transformation, the complexity in the feedback loop of the parallel CRC architecture can be reduced.

If the degree of generator polynomial is n, then the dimension of transformation matrix, TM, will be n×n. A vector c which has a dimension n×1 is used for constructing TM. The vector c should be chosen in such a way that the matrix TM is non singular. Since the feedback path can not be pipelined, the linear transformation moves the complexity inside the feedback path to the blocks outside the feedback path[5]. These blocks can be pipelined and hence the critical path can be reduced. Thus high speed architecture can be achieved along with increased throughput rate.

Using state space model, the remainder polynomial for serial CRC can be expressed as,

$$C(t+1) = B \times C(t) + G \times K(t) \tag{1}$$

where C(t) is the state of flip flops at time t, K(t) is the single bit input at time t. B is the companion matrix defined as,

$$B = \begin{bmatrix} a_{\text{n-1}} & 1 & 0 & . & . & 0 \\ a_{\text{n-2}} & 0 & 1 & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a_1 & 0 & 0 & . & . & 1 \\ a_0 & 0 & 0 & . & . & 0 \end{bmatrix}$$

and G is defined as, G=[a_{n-1}, a_{n-2}, ..., a_1, a_0]^T.
Remainder polynomial at time p can be defined as,

$$C(t+p) = [B^{\text{p-1}}G, .., BG, G] \times \begin{bmatrix} K(0) \\ K(1) \\ . \\ . \\ K(t+p-1) \end{bmatrix} + B^{\text{p}} \times C(t) \tag{2}$$

If m bits of message are processed at a time and k is divisible by m, the the parallel input can be represented as,

$$K_{\text{m}}(t) = (K(tm), K(tm+1), ..., K(tm+m-1))^{\text{T}} \tag{3}$$

Then,

$$C(t+1) = B^{\text{m}} \times C(t) + A_{\text{m}} \times K_{\text{m}}(t) \tag{4}$$

where A_m = [B^{p-1}G,. . ., BG, G] which is an n×m matrix. Linear transformation is applied to (4) through a non singular matrix TM, in order to reduce the complexity of parallel CRC.

$$C(t) = TM \times C_{\text{T}}(t) \tag{5}$$

$$C(t+1) = TM \times C_{\text{T}}(t+1) \tag{6}$$

The equation (4) can be written as,

$$C(t+1) = B_{\text{mT}} \times C(t) + A_{\text{mT}} \times K_{\text{m}}(t) \tag{7}$$

where B_{mT}= $TM^{-1} \times B^m \times TM$
and A_{mT}= $TM^{-1} \times A_m$

In prior works, exhaustive search was used to find a good transformation matrix. Since it was a time consuming method, a new and time saving approximate algorithm is introduced in this work. This method reduces the search space and a good transformation matrix can be calculated within a time less than that required in an exhaustive search.

D. Contributions

In this work, an approximate search algorithm is implemented to find an optimum vector through which an efficient transformation matrix is constructed. Use of this efficient transformation matrix in the state space model of the parallel CRC encoder, reduces the area complexity. The area and power analysis of such an architecture is also done.

III. Methodology

A. Construction of Transformation matrix

A transformation matrix of dimension n×n is constructed using c, where c is an n dimensional vector. The vector c is chosen in such a way that $TM \times TM^{-1}$= I. When the matrix B^m is transformed to form the matrices B_{mT}, A_{mT} and TM, there may a possibility that they become complex. They may become complex with large no.of ones. The total no.of ones in the three matrices B_{mT}, A_{mT} and TM decides the complexity of the parallel CRC structure. To construct a less complex TM, a less complex c vector is needed, which is constructed through an approximate search algorithm.

In this method first bit of c vector is kept as 1 and is subjected to right shift to form successive rows of the transformation matrix. Let the vector c be, c=(1,c_1,...,c_n), then

$$TM = \begin{bmatrix} 1 & c_1 & c_2 & . & . & c_n \\ 0 & 1 & c_1 & . & . & c_{n-1} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & 1 \end{bmatrix}$$

Under an alternate decision of vector c, the matrices B_{mT}, A_{mT} and TM will be different. In this manner, the area complexity depends on the value of c. N, is the total number of ones present in these three matrices, which has to be kept low as possible in order to achieve an efficient parallel CRC structure.

B. Approximate search algorithm

The vector c is having a length same as the degree of the generator polynomial chosen. All possible combinations of vectors can be represented using a tree structure having n layers as shown in Fig. 3. Using each of the vectors, B_{mT}, A_{mT} and TM are constructed and N is counted. Here the operation is progresses through each layers. The repeating nodes and the nodes possessing higher N are discarded. Hence their child node also gets discarded (If father node has more N, it will lead to large N for the child node also) through which time can be saved.

The algorithm is as follows:

1) Root node (c vector) will be having a 1 followed by n-1 zeroes.
2) The next layer consists of all the possible child nodes of the nodes present in previous layer.
3) The N value of each vectors in the current layer is calculated and nodes having higher N are discarded along with their child nodes, then moves to the next layer.
4) Repeating nodes are also discarded.
5) This procedure is repeated on all the layers. Finally, nodes having higher N from the selected X nodes of each layer are discarded, and an optimum vector is chosen which has least N.

X is used to select a specific number of nodes from each layer in case higher degree polynomials.

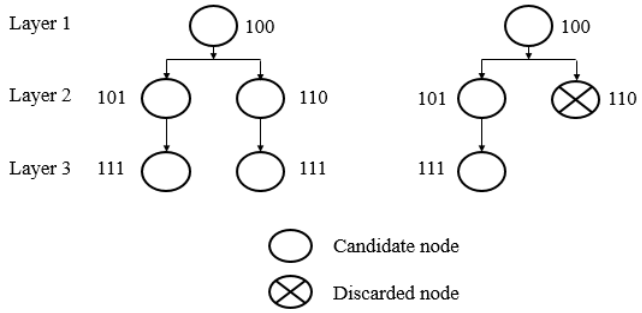

Fig. 3. Generation of all possible vectors and application of approximate search algorithm

IV. Implementations and Results

A polynomial of degree 3 is used for the construction of serial and parallel CRCs. Modelsim SE 6.5 was utilized for obtaining simulation results of the design. The simulation results of the approximate search algorithm and the state space representation are also obtained. The proposed architecture is synthesized using in Xilinx Planahead 14.2 version. xc3s100evq100-5 device is chosen with a package of VQ100 in the Spartan-3E family. The Cadence Encounter RTL Compiler with global synthesis technology is also used through which 90 nm technology mapping is done. It is also used to evaluate various parameters of the architecture like area, power etc.

A. Simulation Results

A serial CRC is designed for the polynomial, $P(x)=1+x+x^3$. The message inputted is 100111101. The seed is set as 111. After 9 clock cycles the CRC is obtained as 101, which is used as the error detecting code. The CRC calculation of a serial CRC can also be done using state space model which is represented using (1).

In the parallel CRC architecture, three serial CRCs are kept parallel. Each of the CRCs are having same generator polynomial $P(x) = 1+x+x^3$. A 9-bit message input message = 100111101 is given and each serial CRC circuit is initialized to a seed value 111. The message is divided as three blocks, 1001, 1110 and 1010 and each CRC will receive each bit of message block on each clock cycle. After 3 clock cycles four

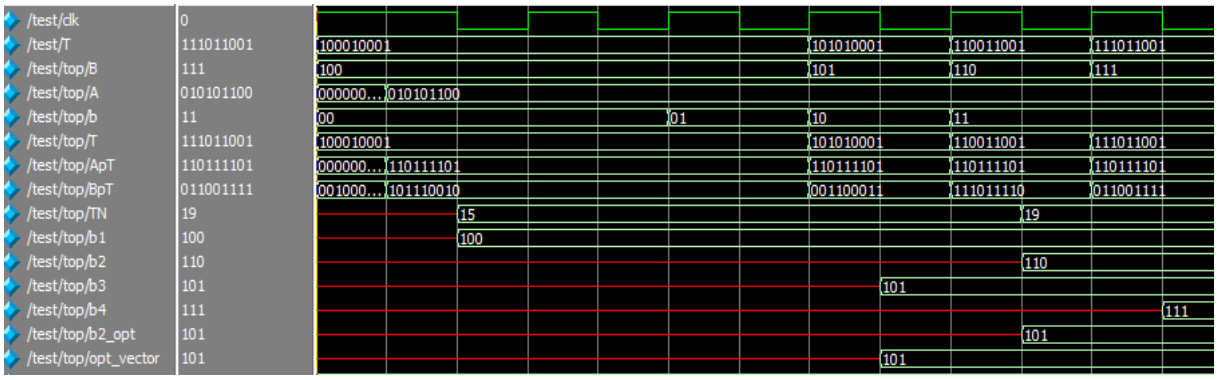

Fig. 4. Optimum vector calculation

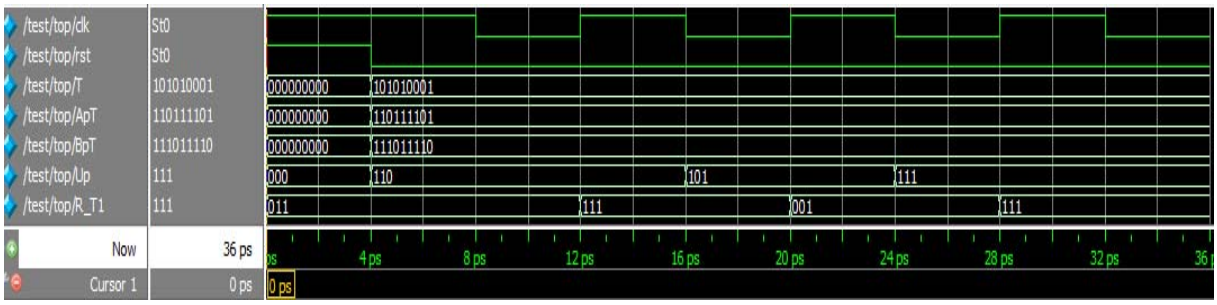

Fig. 5. Simulation results of proposed parallel architecture

3-bit CRC values will be obtained. The final CRC will be the xor of these three CRC values. The four CRC values obtained are, c1=000, c2=001, and c3=111. The final 3-bit CRC value obtained is crc 110.

For a third degree polynomial all the possible combinations of vectors are generated. Then using approximate search algorithm T,A_{mT} and B_{mT} of the vectors in each layer is calculated and N is counted. The vectors having higher N are discarded and an optimum vector is calculated which is obtained as 101 and has N=15. The A_{mT} is obtained as 110111101, B_{mT} is obtained as 001100011 and T will be 101010001. The simulation results are shown in Fig. 4.

Fig. 5 shows the simulation results of proposed parallel CRC architecture using state space transformation. The optimum vector, 101, obtained through the application of approximate search algorithm is applied in the equation 7 and CRC is calculated. The CRC is obtained as 111.

B. Schematic Synthesized Design

The Fig. 6 to 8 shows the schematic of serial, parallel and proposed parallel CRC architectures after synthesis using xilinx planahead. It gives a clear analysis of components involved in the architecture.

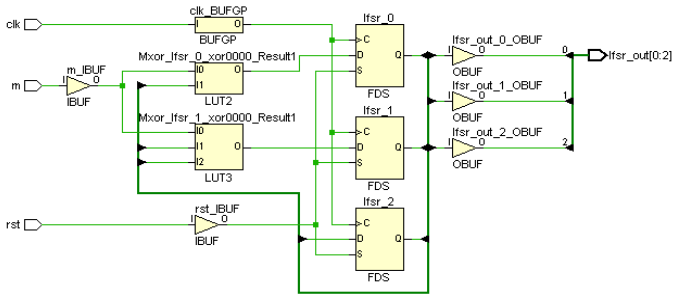

Fig. 6. Schematic of Synthesized serial CRC

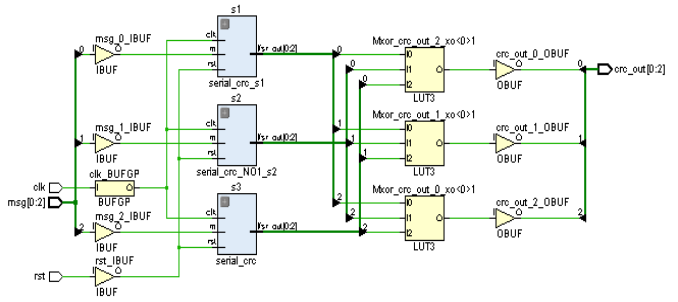

Fig. 7. Schematic of Synthesized Parallel CRC

C. Implemented Utilization

Fig. 9 and 10 gives the graphical representation of implemented utilization which includes slice flipflops, LUTs, slices,IO, BUFGMUX etc used in all the three architectures.

D. Synthesis using RTL compiler

Serial CRC, parallel CRC and the proposed parallel CRC structure are synthesized using cadence encounter RTL compiler, in which mapping to a 90 nm technology is done. The

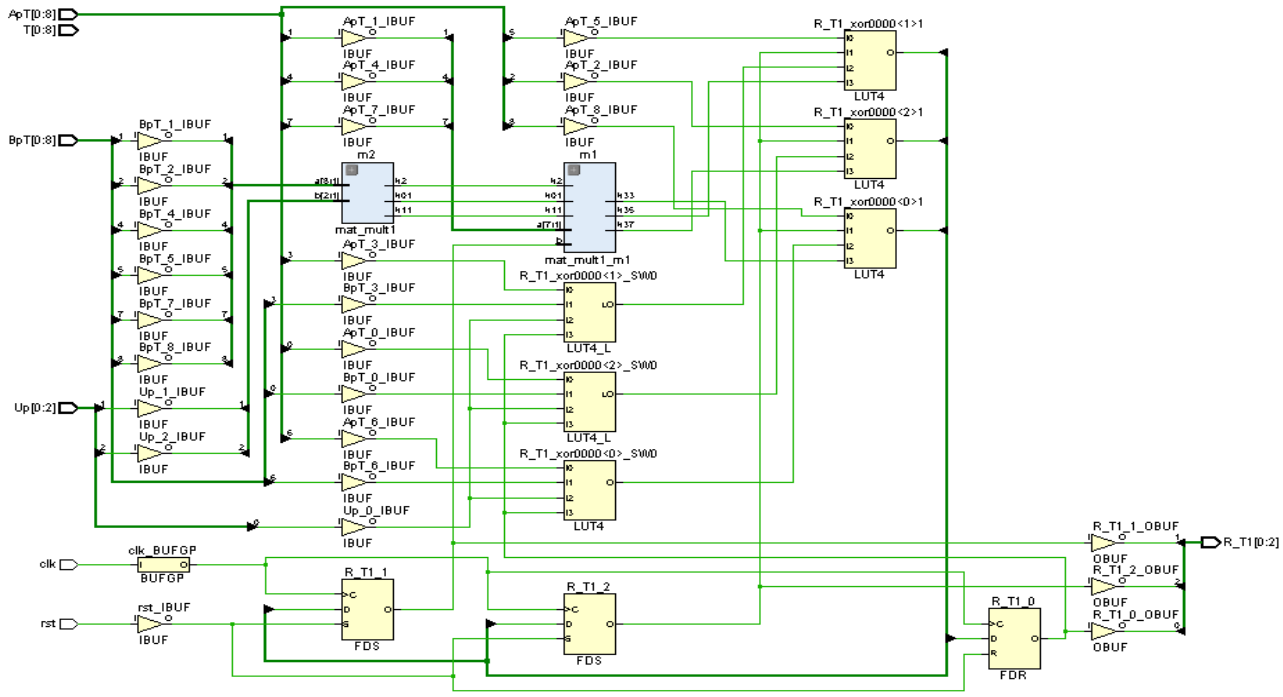

Fig. 8. Schematic of Proposed Parallel CRC after synthesis

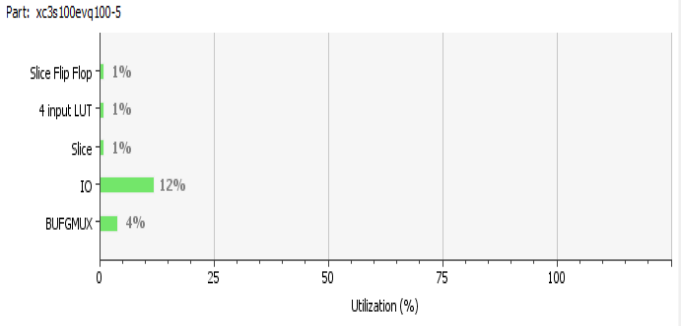

Fig. 9. Implemented utilization of parallel CRC

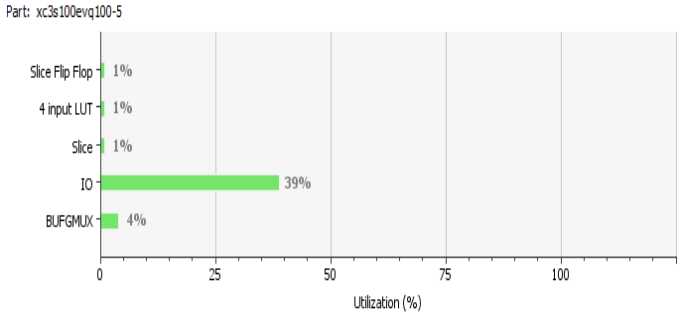

Fig. 10. Implemented utilization of proposed parallel CRC

schematic of the technology mapped architectures are shown in Fig. 11 to 13.

E. Performence Evaluation

The cell report of the serial, parallel and proposed CRC structures are shown in Table I.

TABLE. I. CELL REPORT FOR DIFFERENT 3 DEGREE CRC ARCHITECTURES AFTER SYNTHESIS IN RTL COMPILER

Type of architecture	Type	Instances	Area(nm^2)	area %
Serial CRC	sequential	3	16.416	66.7
	Inverter	1	0.684	2.8
	Logic	5	7.524	30.6
	Total	9	24.624	100
Parallel CRC	sequential	9	49.248	51.1
	Inverter	3	2.052	2.1
	Logic	24	45.144	46.8
	Total	36	96.444	100
Proposed CRC	sequential	3	16.416	21.4
	Logic	36	60.192	78.6
	Total	39	76.608	100

The power consumption of all the three architectures are listed in Table II. The proposed architecture consumes more power than the existing architectures.

TABLE. II. POWER REPORT FOR DIFFERENT 3 DEGREE CRC ARCHITECTURES AFTER SYNTHESIS IN RTL COMPILER

Instance	Cells	Leakage Power (nW)	Dynamic Power (nW)	Total Power (nW)
Serial CRC	9	1.196	1313.375	1314.571
Parallel CRC	36	4.816	4262.127	4266.944
Proposed CRC	39	3.503	4487.162	4490.665

The area, power and minimum period analysis of all the three architectures is done using RTL compiler which is shown in Table III. It shows that area complexity of parallel CRC is reduced from 96.444 nm^2 to 76.608 nm^2 with a tradeoff in power consumption, using the proposed architecture. The

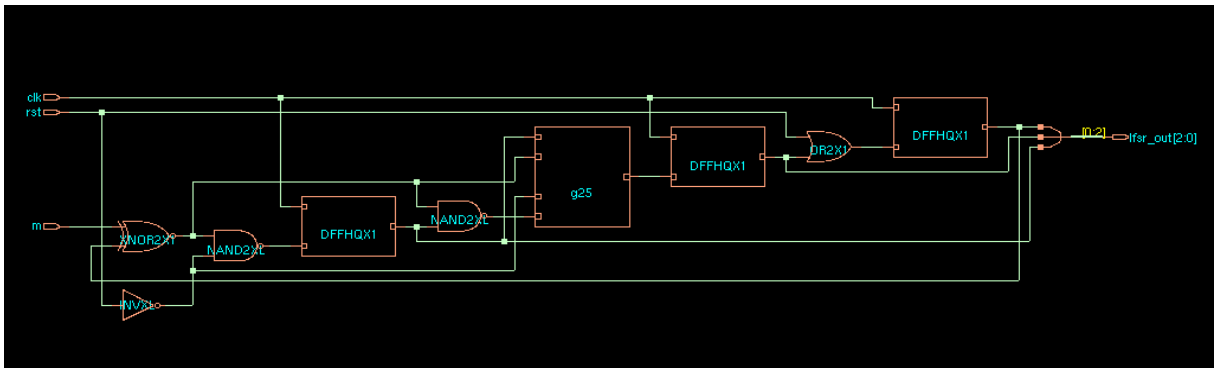

Fig. 11. Schematic of synthesized serial CRC using RTL compiler

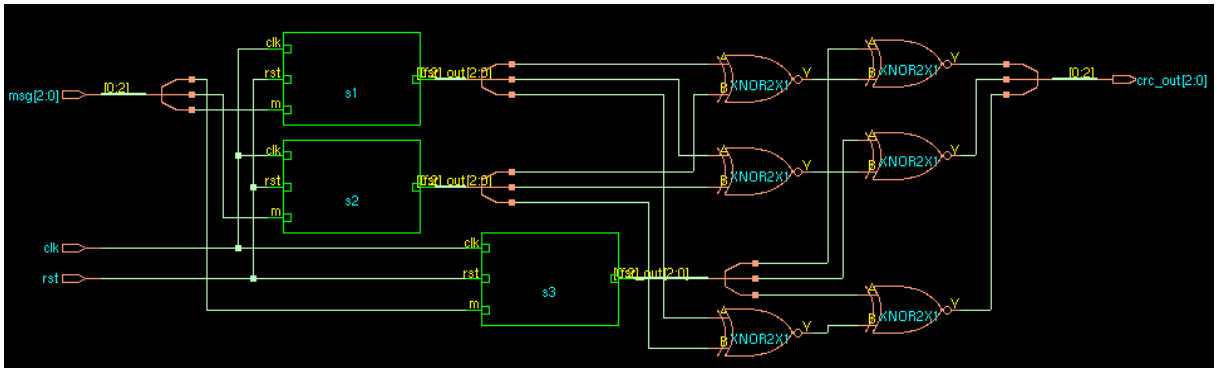

Fig. 12. Schematic of synthesized parallel CRC using RTL compiler

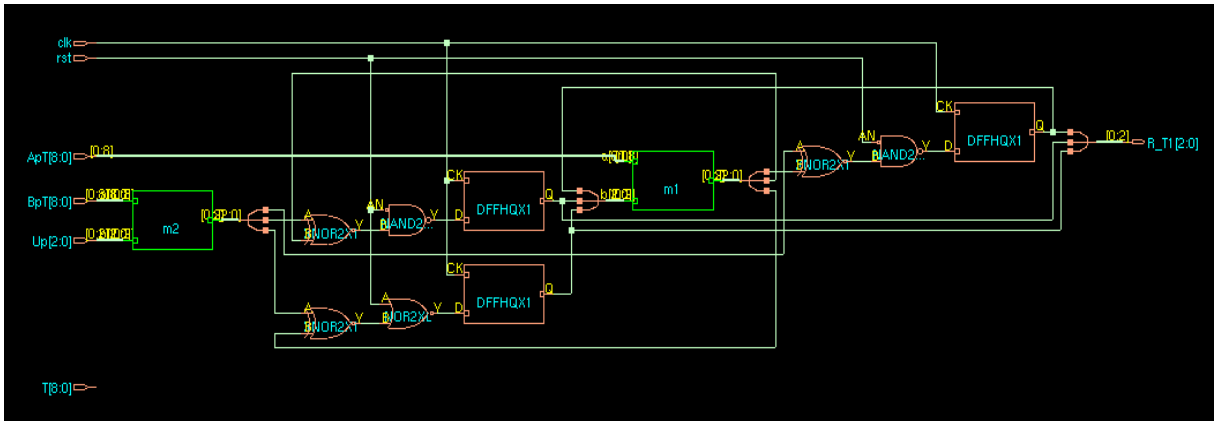

Fig. 13. Schematic of proposed parallel CRC after synthesis using RTL compiler

power consumption of both parallel and proposed architectures are high, when compared to serial CRC. The no.of flipflops in the proposed architecture is less than the parallel architecture but is same as that of a serial CRC.

TABLE. III. ASIC REPORT FOR DIFFERENT 3 DEGREE CRC ARCHITECTURES AFTER SYNTHESIS IN RTL COMPILER

Parameters	Serial CRC	Parallel CRC	Proposed Parallel CRC
Total area(nm^2)	24.624	96.444	76.608
Power (nW)	1314.571	4266.944	4490.665
Minimum period (ns)	1.914	1.914	2.895
No.of flipflops	3	9	3

V. Conclusion and Futurescope

In this work, an area efficient parallel CRC architecture is implemented using state space transformation. An approximate searching algorithm is used to obtain an optimum vector and to construct a good transformation matrix, in order to reduce the area complexity as well as the throughput rate of the parallel architecture. Since, the power consumption of the proposed architecture increases when compared to the existing architectures, in future, designs to decrease the power consumption along with the area reduction can be included. Thus an area efficient and power efficient parallel CRC encoder can be achieved.

REFERENCES

[1] V. Sudharsan and Dr. Yamuna B., *"Support vector machine based decoding algorithm for BCH codes"*, Journal of Telecommunications and Information Technology, vol. 2016, pp. 108-112, 2016.

[2] K. N. Devika, Ramesh Bhakthavatchalu,*"Design of reconfigurable LFSR for VLSI IC testing in ASIC and FPGA,"* 2017 International Conference on Communication and Signal Processing (ICCSP)

[3] M. I. Shiny and M. Devi, N., *"LFSR Based Secured Scan design Testability Techniques"*, in Procedia Computer Science, 2017, vol. 115, pp. 174-181.

[4] Na Haridas and Dr. Nirmala Devi M., *"Efficient linear feedback shift register design for pseudo exhaustive test generation in BIST"*, ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology, vol. 1. Kanyakumari, pp. 350-354, 2011.

[5] M. Ayinala and K. K. Parhi, *"High-speed parallel architectures for linear feedback shift registers,"* IEEE Trans. Signal Process., vol. 59, no.9, pp. 44594469, Sep. 2011.

[6] J. Jung, H. Yoo, Y. Lee, and I.-C. Park, *"Efficient parallel architecture for linear feedback shift registers,"* IEEE Trans. Circuits Syst. II, Express Briefs, vol. 62, no. 11, pp. 10681072, Nov. 2015.

[7] M. Ayinala and K. K. Parhi, *"Efficient parallel VLSI architecture for linear feedback shift registers,"* in Proc. IEEE Workshop Signal Process. Syst., Oct. 2010, pp. 5257.

[8] D.R.K. Sagar,C. Karthik, *"32-Bit Parallel CRC Generation using LFSR,"* International journal of research in advanced engineering technologies,Volume 2, Issue 1 SEP 2014.

[9] J. H. Derby, *"High-speed CRC computation using state-space transformations,"* in Proc. IEEE GLOBECOM, Nov. 2001, pp. 166170.

[10] C. Kennedy and A. Reyhani-Masoleh, *"High-speed CRC computations using improved state-space transformations,"* in Proc. IEEE Int. Conf.Electro/Inf. Technol., Jun. 2009, pp. 914.

[11] Guanghui Hu, Jin Sha, and ZhongfengWang, *"High-Speed Parallel LFSR Architectures Based on Improved State-Space Transformations,"*IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 3, march 2017.

[12] M. Y. Hsiao and K. Y. Sih, *"Serial-to-parallel transformation of linear-feedback shift-register circuits,"* IEEE Trans. Electron. Comput., vol. EC-13, no. 6, pp. 738740, Dec. 1964.

[13] T.-B. Pei and C. Zukowski, *"High-speed parallel CRC circuits in VLSI,"* IEEE Trans. Commun., vol. 40, no. 4, pp. 653657, Apr. 1992.