ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

# Design and Implementation of High Speed CRC Generators

Chidambarakumar.S[1], Thaky Ahmed[2], Ubaidullah.M.M[3], Venketesh.K[4], J.Subhash[5]

UG Scholar, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India[1,2,3,4]

Associate Professor, Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India[5]

**ABSTRACT**: High speed data transmission is a necessary requirement of current smart generation. The networking environment is in an urge to develop extremely fast wireless communication with null occurrence of error. Cyclic redundancy check (CRC) is a promising solution for error detection in data transmission. The high speed data transmission with high speed error checking using parallel CRC generation is presented. The proposed scheme reports parallel CRC architecture composed of 32 bits based on F matrix with a polynomial generator polynomial of order n. The improved efficiency of the proposed technique verified using Xilinx ISE Simulator.

**KEYWORDS**: Cyclic Redundancy Check, Parallel CRC calculation, Linear Feedback Shift Register, LFSR, F matrix

## I. INTRODUCTION

Cyclic redundancy check is widely used in various application areas like data communication, data storage, data compression for reducing errors in data transmission. The conventional hardware implementation of CRC calculations is based on the linear feedback shift registers (LFSRs) where the data is checked in a serial manner with minimum throughput. Parallel CRC computation can be able to rectify these limitations to some extent such that the throughput of the 32-bit parallel calculation of CRC-32 can achieve several gigabits per second. But this minimum performance improvement cannot satisfy the data transmission speed of Ethernet networks which is almost used in a huge manner. Hence an alternative approaching of processing multiple bits in parallel is suggested. Variants of CRCs are used in applications like CRC-16 BISYNC protocols, CRC-32 in Ethernet frame for error detection, CRC-8 in ATM, CRC-CCITT in X-25 protocol, disc storage, SDLC, and XMODEM.Albertengo et al. developed z-transform based architecture for parallel CRC where synthesizable Verilog code generation is impossible. Braun et al. proposed a technique suitable for FPGA implementation with analytical proof of relatively high complexity. Another approach based on Galois field has been proposed by Shieh et al. Campobello et al. suggested 32-bit parallel processing using pre-calculated F matrix whose performance will be affected by a polynomial change. This paper deals with an architecture where 32-bit parallel processing is performed by means of F matrix which can be able to reduce the number of cycles of CRC to 50%.This paper is organized as follows. The conventional serial CRC generation technique based on LFSR is discussed in section II. The parallel CRC generation scheme based on F matrix for 32 bits is presented in section III and IV respectively. Simulation results are provided and discussed in section V. The enhanced performance of proposed technique is concluded in section VI.

## II. SERIAL CRC

Linear feedback shift registers (LFSR) are commonly used for generating serial CRC by performing a simple computation of binary divisions. It can be performed by a sequence of subtractions and shifting operations. In modulo -2 arithmetic, the addition and subtraction are equivalent to bitwise XORs ($\oplus$) and multiplication is equivalent to AND ($\otimes$). The basic architecture of LFSR used for serial CRC calculation is shown in Fig. 1.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

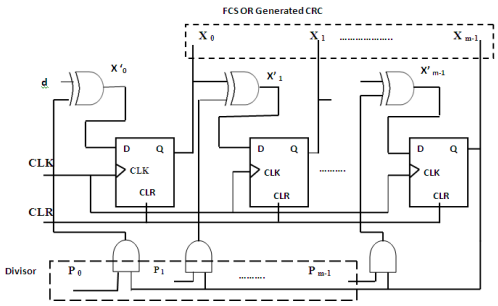


Fig. 1: Basic LFSR Architecture

As shown in above Fig. 'd' is serial input data, 'X' represents present state (generated CRC), 'X' represents next state and 'p' is generator polynomial. The operation of basic LFSR architecture is expressed in terms of following equations.

$$X_0' = (P_0 \otimes X_{m-1}) \oplus d \qquad (1)$$
$$X_i' = (P_0 \otimes X_{m-1}) \oplus X_{i-1}$$

The generator polynomial for CRC-32 is as follows

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + x^0;$$

We can extract the coefficients of G(x) and represent it in binary form as

$$P = \{P32, P31, \ldots\ldots\ldots\ P0\}$$

$$P = \{10000010011000001000111011011 0111\}$$

Frame Check Sequence (FCS) will be generated after 'k+m' cycle, where 'k' denotes the number of bits of data and 'm' denotes the order of generator polynomial. In case of 32 bits serial CRC, the serial CRC generation will happen after 64 cycles when the order of generator polynomial is 32.

## III. PARALLEL CRC

There multiple techniques used for parallel generation of CRC is labeled below.

1. A Table-Based Algorithm for Pipelined CRC Calculation.
2. Fast CRC Updation.
3. F matrix based parallel CRC generation.
4. Algorithm for Unfolding, Re-timing and Pipelining.

LUT based architecture provides minimum memory, whereas the high pipelining table based architecture has multiple inputs such as LUT1, LUT2 and LUT3. LUT3 contains CRC values for the input followed by 12 bytes of zeros, 8 bytes of LUT2, and 4 bytes of LUT4. Higher throughput has been achieved through this algorithm. But its performance is limited by pre-calculating CRC and storing it in LUT, because LUT must be changed with respect to polynomial change. Pipelining algorithm can be used to reduce the critical path with the integration of delay element.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

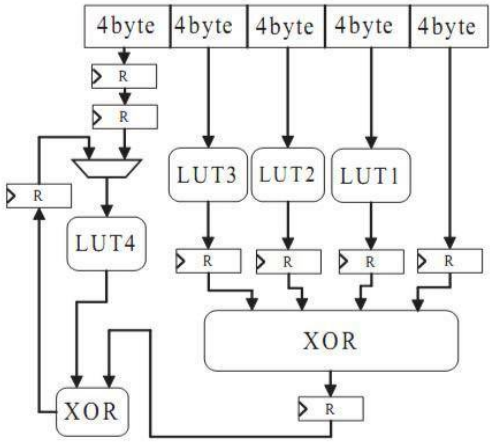


Fig. 2: LUT based architecture

Parallel processing used to increase the throughput by producing the number of output at the same time. Retiming is used to increase the clock rate of the circuit by reducing the critical path computation time. The fast CRC update technique involves the calculation of CRC for the bits which undergoes any change instead of calculating CRC each time for all the data bits.

There are different approaches to generate the parallel CRC having advantages and disadvantages for each technique. Table based architecture required pre-calculated LUT, so, it will not be used for generalized CRC, fast CRC update technique required buffer to store the old CRC and data. In unfolding architecture increases the no. of iteration bound. The F matrix based architecture more simple and low complex.

### *A. ALGORITHM FOR F MATRIX BASED ARCHITECTURE.*

Algorithm and Parallel architecture for CRC generation based on F matrix is discussed in this section. As shown in
fig. 2 it is basic algorithm for F matrix based parallel CRC generationParallel data input and each element of F matrix, which is generated from given generator polynomial is ANDed, result of that will XORing with present state of CRC checksum. The result generated after (k+ m) /w cycle.

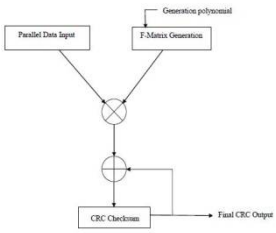


Fig. 3: Algorithms for F matrix based architecture

### *B. F MATRIX GENERATION*

F matrix is generated from generator polynomial.Where, $\{p_0 \ldots\ldots p_{m-1}\}$ is generator polynomial.For example, the generator polynomial for CRC4 is {1, 0, 0, 1, 1} and w bits are parallelly processed.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

$$\mathbf{F} = \begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{matrix}$$

$$\mathbf{F} = \begin{matrix} \mathbf{Pm\text{-}1} & 1 & 0 & 0 & 0 \\ \mathbf{Pm\text{-}2} & 0 & 1 & 0 & 0 \\ \mathbf{Pm\text{-}3} & 0 & 0 & 1 & 0 \\ \mathbf{Pm\text{-}4} & 0 & 0 & 0 & 1 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ \mathbf{P0} & 0 & 0 & 0 & 0 \end{matrix}$$

### *C. PARALLEL ARCHITECTURE*

Parallel architecture based on F matrix illustrated in fig. 3. As shown in fig. 1, d is data that is parallel processed (i.e. 32bit), X' is next state, X is current state (generated CRC), F(i)(j) is the $i^{th}$ row and $j^{th}$ column of F matrix. If $X = [x_{m-1} \ldots . x_1\ x_0]^T$ is utilized to denote the state of the shift registers, in linear system theory, the state equation for LFSRs can be expressed in modular-2 arithmetic as follow.

$$X_i' = (P_0 \otimes X_{m-1}) \oplus X_{i+1} \qquad (2)$$

Where, X(i) represents the $i^{th}$ state of the registers, X(i + 1) denotes the $(i + 1)^{th}$ state of the registers, d denotes the one-bit shift-in serial input.
F is an m x m matrix and G is a 1 x m matrix.

$$G = [0\ 0 \ldots\ldots\ldots 0\ 1]^T \qquad (3)$$

Finally, equation (3) can be written in matrix form as

$$X' = F^W \otimes X \oplus d \qquad (4)$$

Equation (4) can be expanded as,

$$\begin{aligned} X_3' &= X_2 \oplus X_1 \oplus X_0 \oplus d_3 \\ X_2' &= X_3 \oplus X_2 \oplus d_2 \\ X_1' &= X_3 \oplus X_2 \oplus X_1 \oplus d_1 \\ X_0' &= X_3 \oplus X_2 \oplus X_1 \oplus X_0 \oplus d_0 \end{aligned} \qquad (6)$$

The dividend is divided into three 4-bit fields, acting as the parallel input vectors D(0), D(1), D(2), respectively. The initial state is $X(0) = [0\ 0\ 0\ 0]^T$.

$$X(4) = F^4 \otimes X(0) \oplus D(0)$$

$$X(12) = F^4 \otimes X(8) \oplus D(2) \qquad (7)$$

calculation of the next state X' by matrix $F^w$, current state X and parallel input D, make the 32-bit parallel input vector suitable for any length of messages besides the multiple of a sequence of 32-bit parallel calculation, the final remaining number of bits of the message could be 8; 16,or 24. For all these situations, an additional parallel calculation w = 8; 16; 24 is needed by choosing the corresponding $F^w$. If the length of the message is

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

not the multiple of the number of parallel processing bits w = 4 i.e. data bit is11011101011. Then last two more bits D(3) need to be calculated after getting X (12).

For calculating the extra bits if the data message length is not the multiple of w, the number of parallel processing bits. It is worth to notice that in CRC-32 algorithm, the initial state of the shift registers is preset to all `1's. Therefore, X(0) = 0xFFFF. However, the initial state X (0) does not affect the correctness of the design. For better understanding, the initial state X (0) is still set to 0x0000 when the circuit is implemented.

## IV. PROPOSED DESIGN

In proposed architecture w= 32 bits are parallelly processed and order of generator polynomial is m= 16. As discussed in section 3, if 32 bits are processed parallelly then CRC-32 will be generated after (k □ m)/w cycles. If we increase number of bits to be processed parallelly, number of cycles required to calculate CRC can be reduced. Proposed architecture can be realized by below equation.

$$Xtemp = F^{W} \otimes D(0to15) \oplus D(16to32)$$
$$X' = F^{W} \otimes X \oplus Xtemp \qquad (11)$$

Where,

D (0 to 15) =first 16 bits of parallel data input D (0 to 32) = next 16 bits of parallel data input
X'=next state,X=present state

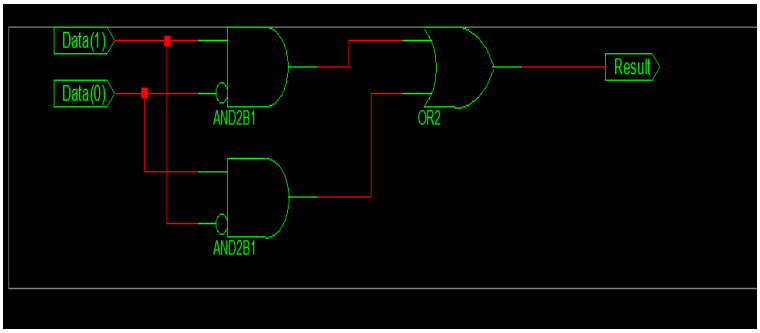


Fig. 4: Circuit diagram of parallel calculation of CRC-32

In proposed architecture $d_i$ is the parallel input and F(i)(j) is the element of $F^{16}$ matrix located at $i^{th}$ row and $j^{th}$ column. As shown in figure 3 input data bits $d_0 \ldots d_{15}$anded with each row of $F^{W}$ matrix and result will be xored individually with $d_{16}$, $d_{17}$ …….$d_{32}$. Then each xored result is then xored with the X' (i) term of CRC-32. Finally X will be the CRC generated after (k □ m)/w cycle, where w=3

## V. RESULT

### *A. SYNTHESIS REPORT*

The proposed architecture is synthesized in Xilinx-9.2i and simulated in Xilinx ISE Simulator, which required half cycle then the previous 32-bit design. In our programming in Verilog by specifying only generator polynomial, it directly gives F matrix useful for parallel CRC generation that is not available in previous methods. Hardware utilization is compared in table I for different approaches for different parameter like LUT, CPD and cycle.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

| Device Utilization Summary | | | | |
|---|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** | **Note(s)** |
| Number of Slice Flip Flops | 356 | 1,536 | 23% | |
| Number of 4 input LUTs | 430 | 1,536 | 27% | |
| **Logic Distribution** | | | | |
| Number of occupied Slices | 336 | 768 | 43% | |
| Number of Slices containing only related logic | 336 | 336 | 100% | |
| Number of Slices containing unrelated logic | 0 | 336 | 0% | |
| **Total Number of 4 input LUTs** | 430 | 1,536 | 27% | |
| Number of bonded IOBs | 153 | 124 | 123% | OVERMAPPED |
| IOB Flip Flops | 164 | | | |
| Number of GCLKs | 4 | 8 | 50% | |
| **Total equivalent gate count for design** | 6,752 | | | |
| Additional JTAG gate count for IOBs | 7,344 | | | |

fig. 7: Synthesis report for parallel CRC-32.

```
Data Path: crc1/data_in_14 to crc1/lfsr_q_13
                           Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   FDC:C->Q             9   0.626   1.250  crc1/data_in_14 (crc1/data_in_14)
   LUT3:I0->O           1   0.479   0.704  crc1/lfsr_c<13>4_SW1 (N6118)
   LUT4_D:I3->O         1   0.479   0.704  crc1/lfsr_c<13>4 (crc1/lfsr_c<13>_bdd6)
   LUT4:I3->O           1   0.479   0.000  crc1/lfsr_c<13>58 (crc1/lfsr_c<13>)
   FDPE:D                   0.176          crc1/lfsr_q_13
  ----------------------------------------
  Total                     4.897ns (2.239ns logic, 2.658ns route)
                                    (45.7% logic, 54.3% route)

=========================================================================
Timing constraint: Default period analysis for Clock 'clk2'
  Clock period: 4.862ns (frequency: 205.666MHz)
  Total number of paths / destination ports: 559 / 145
-------------------------------------------------------------------------
```

Fig. 8: Total delay of parallel CRC-32.

| CRC W parallel bits | Clock cycle | LUTs | CPD |
|---|---|---|---|
| CRC32 w=16bit[1 ] | 17 | 162 | 7.3ns |
| CRC32 w=32bit[8 ] | 17 | 220 | 30.5ns |
| CRC32 w=32bit(Proposed) | 9 | 430 | 4.8ns |

Fig. 9: Comparison table for different algorithms of CRC

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 3, March 2017

5th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '17)

13th-14th March 2017

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India.

### *B. SIMULATION*

From the table it is observe that, the architecture proposed by require 17 clock cycle to generate CRC as per equation (k+ m)/w and for proposed architecture it required only 9 cycle, CPD for proposed architecture is less than the architecture, only the disadvantage for proposed architecture is the no. of LUT get increased, so area also get increase .

## VI.CONCLUSION

32-bit parallel architecture required 17 ((k + m)/w) clock cycles for 32-byte data. Proposed design (32bit) required only 9 cycles to generate CRC with same order of generator polynomial. So, it drastically reduces computation time to 50% and same time increases the throughput. Pre-calculation of F matrix is not required in proposed architecture. Hence, this is compact and easy method for fast CRC generation.

## REFERENCES

[1] Campobello, G.; Patane, G.; Russo, M.; "Parallel CRC realization," *Computers, IEEE Transactions on* , vol.52, no.10, pp. 1312- 1319, Oct.2003.

[2] M.D.Shieh et al., “A Systematic Approach for Parallel CRC Computations,” Journal of Information Science and Engineering, May 2001.

[3] Braun, F.; Waldvogel, M.; "Fast incremental CRC updates for IP over ATM networks," *High Performance Switching and Routing,2001 IEEE Workshop on*, vol., no., pp.48-52, 2001.

[4] Weidong Lu and Stephan Wong, “A Fast CRC Update Implementation”, IEEE Workshop on High Performance Switching and Routing, pp. 113-120, Oct. 2003.

[5] RameshwarMurade, MD Manan Mujahid, M.A.M. Sabir, “The design and implementation of a Programmable CRC computation circuit architecture using FPGA” 2013.

[6] S.R. Ruckmani, P. Anbalagan, “High Speed cyclic Redundancy Check for USB” Reasearch Scholar, Department of Electrical Engineering, Coimbatore Institute of Technology, Coimbatore-641014, DSP Journal, Volume 6, Issue 1, September, 2006.

[7] Yan Sun; Min Sik Kim; "A Pipelined CRC Calculation Using Lookup Tables," *Consumer Communications and NetworkingConference (CCNC), 2010 7th IEEE*, vol., no., pp.1-2, 9-12 Jan.2010

[8] Sprachmann, M.; "Automatic generation of parallel CRC circuits," *Design & Test of Computers, IEEE*, vol.18, no.3, pp.108-114, May 2001.