Usage of 64 bit CRC utilizing Parallel Execution for 3G Systems

Parulpreet Singh[1], Rupali Walia[2], Gagandeep Singh Walia[3] , Shelej Khera[4]
[1,3,4] *School of Electronics and Electrical Engineering, Lovely Professional University*
[2]*Department of Computer Applications, Trinity Institute of Management and Technology*
Corresponding Author: parulpreet.23367@lpu.co.in

***Abstract*- Cyclic redundancy check is highly utilized method for checking the presence of error in the data that have been introduced during transmission over a media. As the size of message bit increases in 3G, the CRC computation of the data becomes slow which is a major drawback in a communication process.**

To match with the speed of transmitting data it is significant to optimize the rate of CRC computation. A 64 bit CRC generator provides with more swiftness as compared to the 32 bit CRC for 3G. This paper focuses on implementation of 64 bit CRC on 3G through parallel execution unit that will generate CRC for various packets and will combine them to create the final CRC result. Consequently, this method will immensely reduce the clock cycles to calculate CRC. Based on this concept, the comparative analysis between 32 bit CRC and 64 bit CRC is also included in the paper. The latency decreases for the 64 bit CRC parallel execution unit and throughput increases.

Keywords- Cyclic Redundancy Check, 3G

I. INTRODUCTION

If the communication has to take place wirelessly, the most important concerning factor is see that no errors are found the message which is sent should be reliable. So, for the steady and reliable transmission, we are implementing 64 bit parallel CRC generator for 3G. During the data transmission the errors get introduced. So it becomes necessary that they are detected in order to make the signal error free. Communication system uses many error detection algorithms to achieve reliable communication between sender and receiver. One of the most trusted method to find out the errors is cyclic redundancy check . But it cannot be used to correct the errors. The sender and receiver mutually decide a generator polynomial. Checksum bits are sent along with the transmitted message. The receiver has to decide that whether or not checksum is in accordance with the data. This is to determine the presence of errors. If the error has occurred, the receiver sends the negative acknowledgement (NAK). The NAK is send in two cases

- The data received is incomplete.
- The data is not received at all.

So in such cases, the receiver requests the transmitter to send that message again. The CRC standards differ in many ways other than for the selection of generator polynomial. CRC is used in many systems such as Ethernet, 2G, 3G links, ADSL/VDSL, Wireless LAN , Bluetooth etc.[3] Linear feedback shift register is generally the hardware solution for CRC. But LFSR cannot be efficiently used for increased rate of bits. [3] For removing this bug we make use of parallel multiple execution units. To execute parallel 64 bit CRC calculation in 3G we use 64 bit long polynomial which is given as

$$x^{64}+x^{62}+x^{57}+x^{55}+x^{54}+x^{53}+x^{52}+x^{47}+x^{46}+x^{45}+x^{40}+x^{39}+x^{38}+x^{37}+x^{35}+x^{33}+x^{32}+x^{31}+x^{29}+x^{27}+x^{24}+x^{23}+x^{22}+x^{21}+x^{19}+x^{17}+x^{13}+x^{12}+x^{9}+x^{7}+x^{4}+x+1$$

Here we propose the analogy to create the CRC code based on the execution of the multiple parallel executions to enhance the computation speed which would be helpful in 3G communication.

II. METHODOLOGY

The 64 bit CRC of any input sequence can be calculated either serially or parallel. A serial computation method is slow as CRC is found by shifting the generator polynomial and XORing it with the input sequence for each message bit. But the parallel computation method divides the data in number of smaller execution units and executes them parallel. The resultant CRC of each unit is XORed to obtain the final CRC. A given input sequence is divided into M execution units with N bytes each and independent remainder of each execution unit is calculated. As a result, there are M remainders for M execution units. These remainders are XORed with each other to find the final CRC result which is appended at end of the input sequence. A 64 bit generator polynomial produces a 64 bit CRC result which not only improves the error detection capability but it also makes the process faster as 64 bits of the input sequence are XORed with generator at a time.

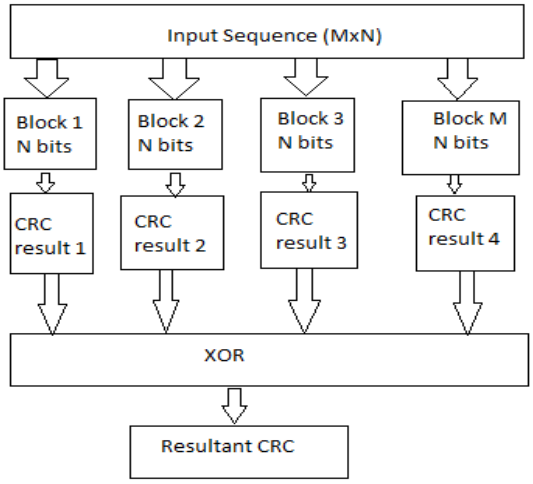

Fig. 1. CRC Calculation using Parallel Execution units

A. Design of CRC generator

In the proposed design of 3G, we have used the 64 bit long polynomial which is given as

$x^{64}+x^{62}+x^{57}+x^{55}+x^{54}+x^{53}+x^{52}+x^{47}+x^{46}+x^{45}+x^{40}+x^{39}+x^{38}+x^{37}+x^{35}+x^{33}+x^{32}+x^{31}+x^{29}+x^{27}+x^{24}+x^{23}+x^{22}+x^{21}+x^{19}+x^{17}+x^{13}+x^{12}+x^{9}+x^{7}+x^{4}+x+1$

This has a hex code 0x42F0E1EBA9EA3693

III. PARALLEL EXECUTION

A high speed 3G requires a faster computation. Consider a message of 2048 bits to be transmitted by 3G network. A serial computation would be slow as it would compute the bits serially. But in parallel computation the data bits are divided into 2, 4,8,16 blocks. For 2048 data bits, 2 blocks will have 1024 bits each, 4 blocks will have 512 bits each, 8 blocks will have 256 bits each and 16 blocks will have 128 bits each. For 2048 bits divided in two blocks, the time required to compute CRC will be same as needed for 1024 bits. Hence computation time is decreased. As the number of blocks increases, the computation time for CRC will decrease. At the same time the components used in the circuit to design such CRC generator will increase

When the message of 2048 bits is divided into two blocks, each block will have 1024 bits. Tw execution units are divided by 64 bit CRC polynomial thus producing two independent 64 bit remainders. These remainders are XORed to obtain the final 64 bit CRC.

When the message of 2048 bits is distributed into 8 blocks, each block will have 258 bits. The 8 execution units will produce 8 independent 64 bit remainders which will be XORed to obtain the final result.

IV. COMPARISON OF 32 BIT AND 64BIT CRC

The 32 bit CRC and 64 bit CRC not only differs in their generator polynomial but also shows difference in throughput, clock cycles, latency and speed. Clock cycles in 32 bit CRC are more as compared to the 64-bit CRC. More the clock cycles slower the system will be. 64-bit CRC has reduced clock cycles which increase the speed. Greater the processing speed better becomes the system. Latency of 32 bit CRC is more than 64 bit CRC which increases the time delay in computation of CRC for 32 bit CRC. Having less latency in 64 bit decreases the processing delay of CRC thus making the system faster. The throughput in case of 32 bit CRC is less than 64 bit CRC. Hence chances of error in the message delivered are more than 64 bit CRC. The increased throughput of 64 bit CRC increases the successful message delivery over the communication system.

We will find out the performance parameters for the calculation of CRC which are given as below.

1) Latency- This in a network symbolifies the delay i.e. the time taken by a packet of data to reach the receiver from the transmitter. Latency is factor that contributes to network speed. Low-latency network experiences small time delays, high-latency networks experiences long time delays. Latency of a network depends upon-

- Transmission- It includes the delay caused by the medium of propagation.
- Router-time taken to change the starting bits (header) of the data.
- Propagation-Time taken to reach the receiver.
- Other delays-The delay due to storage or due to the passing from the bridges and other intermediate devices.

Latency is calculated by [1]

$$Latency(s) = \frac{NComp + (M-1)Comb}{F_{System}}$$

Here,

N represents number of blocks per byte
M represents number of divisions of input data
Comp represents cycles for calculation of CRC
Comb represents cycles to find modulo 2 of all remainders
F_{system} represents clock frequency of the system which has been kept permanent for Spartan 3 kit as 50 MHz

2) Throughput: It is the speed at which something can be processed. i.e the amount of data successfully transferred from one place to another within a given time slot. It is calculated bit per second or data per second. Throughput depends on –

- Limitation of physical medium in which it travels.
- Processing power of the system.

Throughput of a system is given by [1]

$$Throughput\left(\frac{bits}{s}\right) = F_{system}\frac{M*N*8}{N*Comp+(M-1)Comb}$$

Here,

N represents number of blocks per byte
M represents number of divisions of input data
Comp represents cycles for calculation of CRC
Comb represents cycles to find modulo 2 of all remainders
F_{system} represents clock frequency of the system which has been kept permanent for Spartan 3 kit as 50 MHz

Speed: 32-bit CRC can process 33-bits of data a time, whereas 64-bit CRC can process 65-bits of data which is comparatively a faster processing rate as compared to the 32-bit CRC. Higher the speed better performance it will give. Clock cycles in 64-bit CRC are reduced as compared to 32-bit CRC which increases the speed and performance.

V. RESULT

The simulation result for 64 bit CRC and 2048 input data is computed by Xilinx ISE 9.1i. using parallel execution units are formed by 2, 4, 8, 16 blocks.

Simulation result for 16 execution unit:

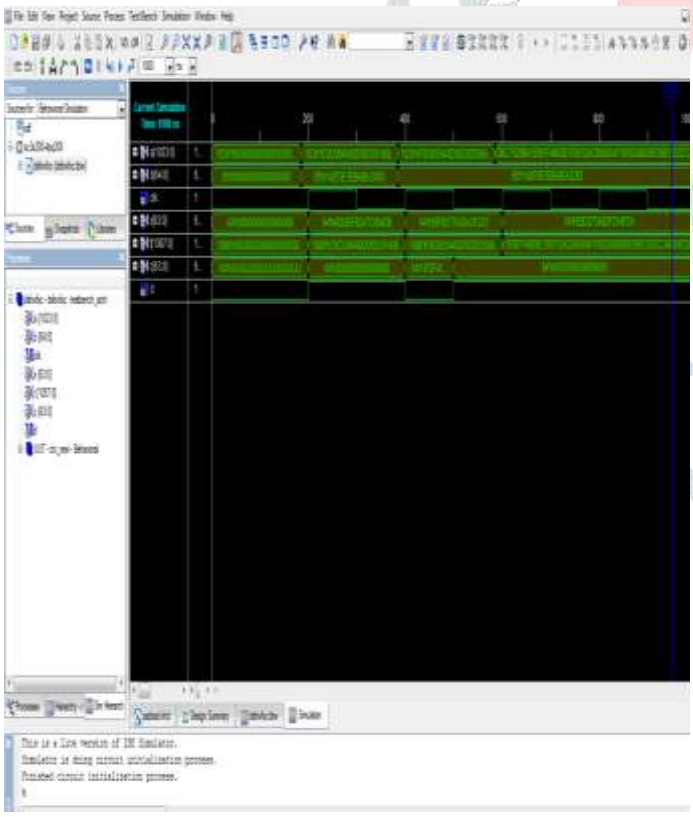

Fig. 2. Simulation waveform for 64 bit CRC generator

The latency and throughput for various execution units is given in following table:

TABLE I. The latency and throughput for different execution units.

Execution Unit	N	M	Comp	Comb	Latency	Through put MHz
1 EU	256	1	2048	1	10485 us	0.195
2 EU	128	2	1024	1	2621 us	0.78
4 EU	64	4	512	1	655 us	3.12
8 EU	32	8	256	1	164 us	12.48
16 EU	16	16	128	1	41 us	44.64

The table shows that latency decreases as the number of execution unit increases which in turn decreases the delay in the system. Hence parallel execution becomes faster than serial execution with the increase in execution units.

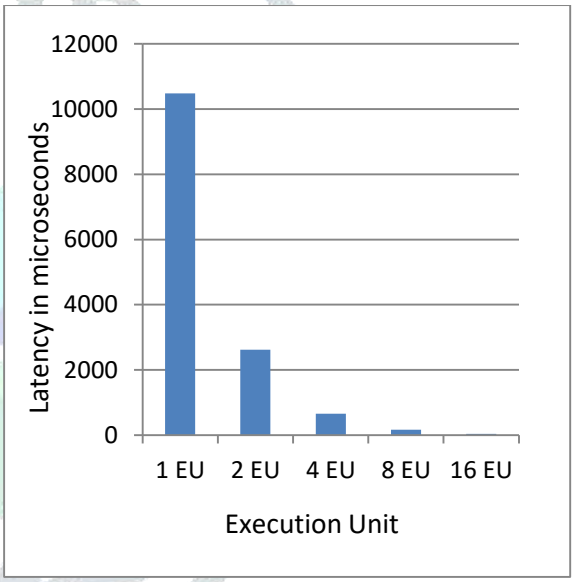

Fig. 3. Latency for different execution units

Similarly the throughput of the system increases. Hence successful message delivery rate increases and the rate of error in the data decreases. The throughput increases with the increase in the execution units.

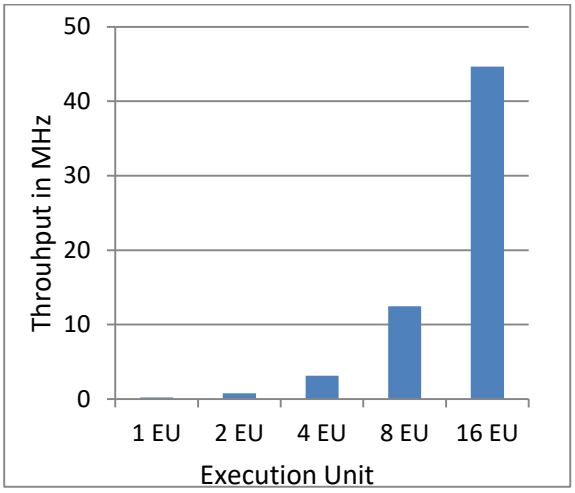

Fig. 4. Throughput for different execution units

VI. CONCLUSION

The quick CRC generator utilizing equal different execution units has been structured and results are confirmed. The exhibition correlation of quick CRC utilizing single execution unit and 2, 4, 8, 16 execution units are gotten in term of latency and throughput and assets usage. The clock cycles for 64- bit crc is reduced as compared to 32-bit crc.the crc result is generated at faster rate. This has been concluded that with the increase in number of multiple parallel execution units , there is a decrease in latency and increase in throughput thus is really good for finding the errors while transmitting data with fast CRC.

REFERENCES

[1] G S.-R. Yoon, S. Seo, M.L. Huang and S.-C. Park : Multi-processor based CRC computation scheme for high-speed wireless LAN design. IEEE ELECTRONICS LETTERS 27th May 2010 Vol. 46 No. 11.

[2] Tomas Henriksson and Dake Liu:Implementation of Fast CRC Calculation 2003 IEEE.

[3] H. Michael Ji, and Earl Killian: Fast Parallel CRC Algorithm and Implementation on a Configurable Processor. 2002 IEEE.

[4] Sanjay M. Joshi, Pradeep K. Dubey, Marc A. Kaplan: A New Parallel Algorithm for CRC Generation. 2002 IEEE

[5] Guido Albertengo, Riccardo Sisto: Parallal CRC generation. IEEE Micro. Special Features.

[6] A. Perez, Byte-wise CRC Calculations, IEEE MICRO, June ,1983.

[7] Weidong Lu and Stephan Wong : A Fast CRC Update Implementation,2004 IEEE,Vol 40, No 113-120.

[8] CRC-Digital Hardware Design Concept : Ishak Suleiman,19July,2004

[9] Alberto Leon-Garcia & Indra Widjaja : Interactive E text, Communication Networks Fundamental concepts and keys architectures.