ISSN 2303-4521
Periodicals of Engineering and Natural Sciences *Original Research*
Vol. 10, No. 2, March 2022, pp.90-100

Forward error correction in 5G heterogeneous network

Rusul M. Kanona[1], **Zainab N. Al-Qudsy**[2], **Shaymaa Azzam Alyawer**[3]
[1,2,3]Computer Sciences Department, Baghdad College of Economic Sciences University, Baghdad, Iraq

ABSTRACT

In this research, the feasibility of developing a complete polar FEC chain of 5th generation cellular mobile communication standard in software. Specifically, on general purpose processors. Paperwork attempts to achieve stringent latency requirements through software, algorithmic and platform specific optimizations. Many algorithms in FEC chain are optimized for hardware implementations. Direct implementation of these algorithms in software results in poor performance. To obtain best performance in terms of latency on general purpose processors, these algorithms are modified or reformulated to suit processor architecture and software implementation. Initially both encoding and decoding FEC chains are implemented naively without any optimization. Code profiling is performed on this naive implementation to identify the significant latency contributors. The research split algorithms of significant latency contributing components into primitive operations. These primitive operations are optimized either with software optimizations or mapped to specialized functional units of a general-purpose processor to achieve best performance using CRC calculation in 5G cellular networks. Optimizations reduced the worst-case latency of the encoding FEC chain from 158µs which is more than 10x reduction in latency with communication rate**.**

Keywords: latency, FEC chain, optimization, encoding, decoding, 5th generation, cyclic redundancy check

Corresponding Author:

Rusul M. Kanona
Computer Sciences Department
Baghdad College of Economic Sciences University
Baghdad, Iraq
E-mail: rusulmkanona@baghdadcollege.edu.iq

1. Introduction

The research concentrates on polar encoding and decoding forward error correction (FEC) chain which are used to transmit and receive control information in 5th generation communication. Latency contributing components are identified. Algorithms of those components are reformulated to avoid or to reduce latency contributing operations. Major latency contributors in encoding FEC chain are the cyclic redundancy check (CRC) calculation, the polar code construction itself and polar encoding. Today computing devices and wireless systems have become integral parts of our society. They allow communication between people even from remote areas. The invention of the internet has enabled people to have access to a world of information in their fingertips [1]. Until recently, wireless devices were primarily used for information exchange between people. It is up to software engineer to efficiently harness this computational power as mentioned in [2]. The ability to perform computations has evolved tremendously from the day the first computer was invented by Charles Babbage in the 19th century. By the end 19th century another important event occurred, in 1897 an Italian inventor and engineer [3] demonstrated radio's ability to maintain continuous contact with ships in the English Channel. A major breakthrough happened in the development of computers and wireless systems in 1948 when scientists at Bell Labs achieved groundbreaking results [4]. Researchers in [5] published his paper “A mathematical theory of communication”. John Bardeen, Walter Brattain, and William Shockley announced the invention of the transistor effect. These two landmark events paved way for the widespread adoption of computers and wireless communication systems in numerous applications. Since then, the telecommunication industry has grown manifold fueled by the advancements in RF and transistor fabrication techniques, miniaturization and very large scale integration as mentioned in [6]. These technological advances made computing devices smaller, cheaper and more reliable. Recent advances in wireless communication have allowed not only short distance communication such as cellular communication but also deep space communication with billions of kilometers distance.

Today’s wireless applications are entering new avenues such as industrial automation, telemedicine, Autonomous driving. These applications demand ultra-reliability and ultra-low-latency. Latest mobile communication standard 5G took a giant step to- wards providing service for such mission-critical applications. 5G has adopted several techniques to service stringent latency requirements. To name few, different OFDM numerologies, flexible frame structure et cetera [7]. Traditionally, to achieve stringent latency requirements wireless communication stacks are implemented in hardware, specifically in FPGAs/ASICs in 5G networks. Hardware implementations make use of implicit hardware-concurrency [8]. However, hardware implementations come with inherent non-flexibility, huge cost and high development time. Due to the latest technological advancements in general-purpose computing, modern processors come with tremendous computation power.

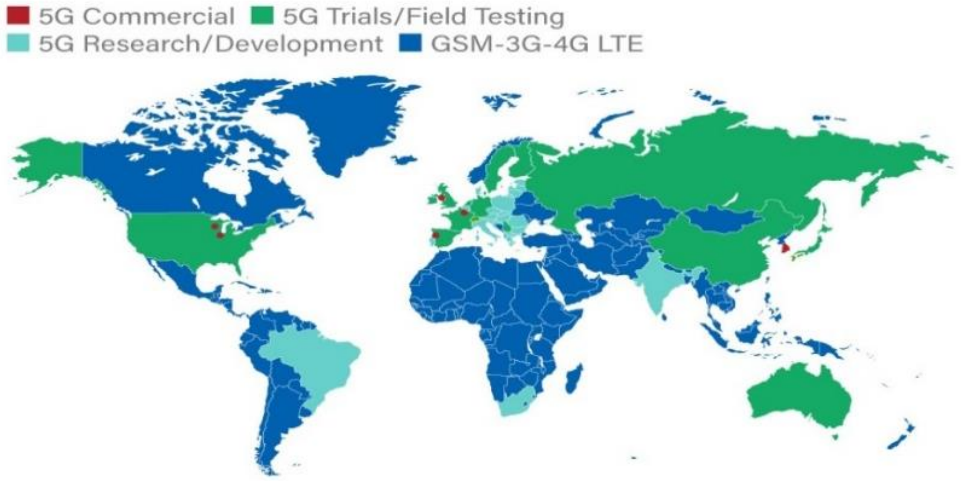

Figure 1. The usage and deployment of 5G cellular network around the world [8]

1.1. Problem statement

Optimizations for hardware such as recursive formulation, reducing look-up tables (LUTs) and flip-flops are not always relevant in software. Let’s try to understand conflicts in the optimizations targeted for hardware and software as discussed in [9]. Most of the encoder/decoder algorithms are formulated in a recursive form. In hardware implementations, recursive formulations are particularly useful since same the design can be replicated multiple times without significant effort and also with no performance compromise [10]. However, in software implementations recursive implementation incur with significant overhead. Mainly due to a large number of branches, stack allocation/de-allocation, and pipeline flushing. The next optimization steps in hardware targeted implementations are minimizing the required memory and flip-flops [11]. The cost of hardware implementation depends on the amount of memory and number flips-flops required as given in [12]. In contrast, general-purpose computing world can make use of off-the-shelve available cheap memory. Software designer should reduce the number of cache misses and branch miss-predictions as given in [13]. In addition, software implementations should also avoid expensive operations such as multiplications, division, and modulus operations. If not, reformulate them by using inexpensive bitwise operators.

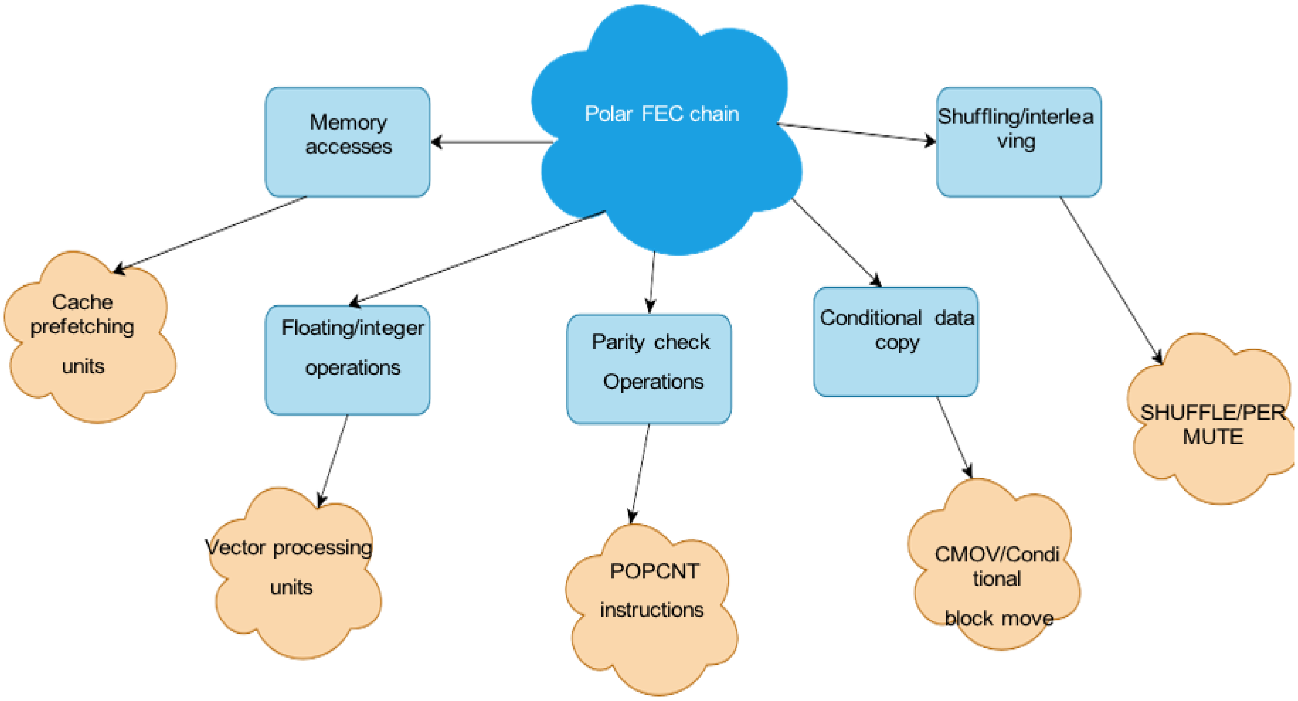

Figure 2. Mapping FEC chain operations to particular functional units [14]

1.2. Algorithm and solution

Modern processors come with special computing units to cater to specific application domains, namely vector processing units for signal processing applications. To achieve stringent latency requirements, it is very important that a software designer makes use of these additional processing elements and available optimization techniques. This work extensively makes use vector processing units through CRC algorithm instruction set extensions in 5G communication. Commonly high-performance signal processing and error correction applications are implemented in hardware in CRC algorithm. Software implementations allow these applications to achieve low latency and high throughput. Algorithms in these applications are developed targeting hardware implementations. These algorithms implemented in software without any reformulation or modification result in poor performance. Therefore, algorithms need to be modified and reformulated to achieve expected performance in software.

Telecom is preparing to undergo a profound transformation in its fundamentals: 5G will use a CRC calculation and an estimated delivery date of 2021. One development is the demise of the long-held belief that the future of frequencies and reception is only different frequency bands on the same channel, as It is described in [15]. The transmission and reception time of transmission in wireless communications enables this next generation, which leads to significant gains. In addition, when it appears in a photo, the gain is greater than expected after a few years. Use systems in a variety of situations. The financial resources available in these networks of the Telecommunications Authority in the structures of the OFDM. Improving Usage Efficiency in a Cellular Network, Power Densification This study explores co-investment, investment interfaces, and pioneering micro-interfaces in a multi-investment small-to-mass (MIMO) environment [17]. Using the network module's LAN to encrypt vehicle communications [18].

1.3. Aim of contribution

This work fills a research gap using both forward error checking and cyclic redundancy check (CRC) calculations to validate 5G technology that is field-validated and untested under various discovered conditions. This study explores the potential of 5G cellular as a different means of communication, as well as the ability to adjust input parameters for various FEC series network deployment configurations. This example can be seen as a high-level approach to 5G network communications under various scenarios, including the use of 5G network propagation models and the massive cyclic redundancy check (CRC) calculation algorithm, and possibly the 5G communication output solution used. networks, which is acceptable from a perspective. In this identification of the most advanced methodologies for calculating RCS and 5G network coverage and capacity models. In addition, the principles of cell network planning should be considered, as well as the importance of using the 5G network in terms of calculation/volume and sharing/mix profile during the day with natural impact.

2. Methodology

The 5G physical channels which use polar codes. The 5G standard adopted polar codes for uplink and downlink control channels. Uplink control channels carry information about channel quality indicators, acknowledgments. In downlink control channels carry resource allocation information, uplink power control instructions and the information required for the user equipment (UE) to access the network. Following sections explain each of these uplink and downlink control channels and their polar FEC chain parameters. In the modern processors, a fast memory called cache is used to reduce the average access time of main memory also called RAM (Random Access Memory). The cache minimizes the number of accesses to RAM by storing frequently accessed data in it, hence avoiding huge penalty of reading data frequently from RAM which operates at a much lower frequency than the CPU. When a memory location is accessed for the first time it is copied from the RAM to the cache, future accesses to the same location is done via cache. This fast memory is placed between RAM and processor. In modern processors instead of single cache, multi-level caches are present. The main idea behind having multi-level caches is that if the data is not found in the first level, then second level is checked if not then the third level until the last level, still, if the data is not found then RAM is accessed. This model significantly reduces the probability of accessing the RAM compared to having a single level cache. Complete memory hierarchy of the modern processors is shown in Figure 3.

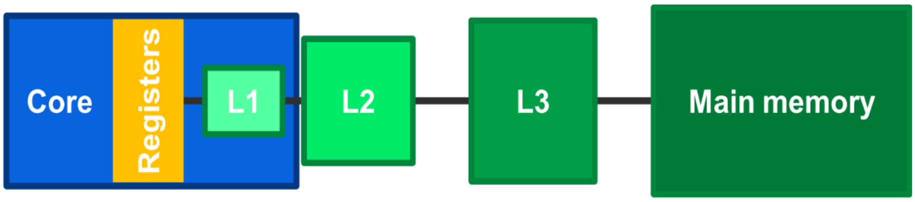

Figure 3. Memory hierarchy in systems

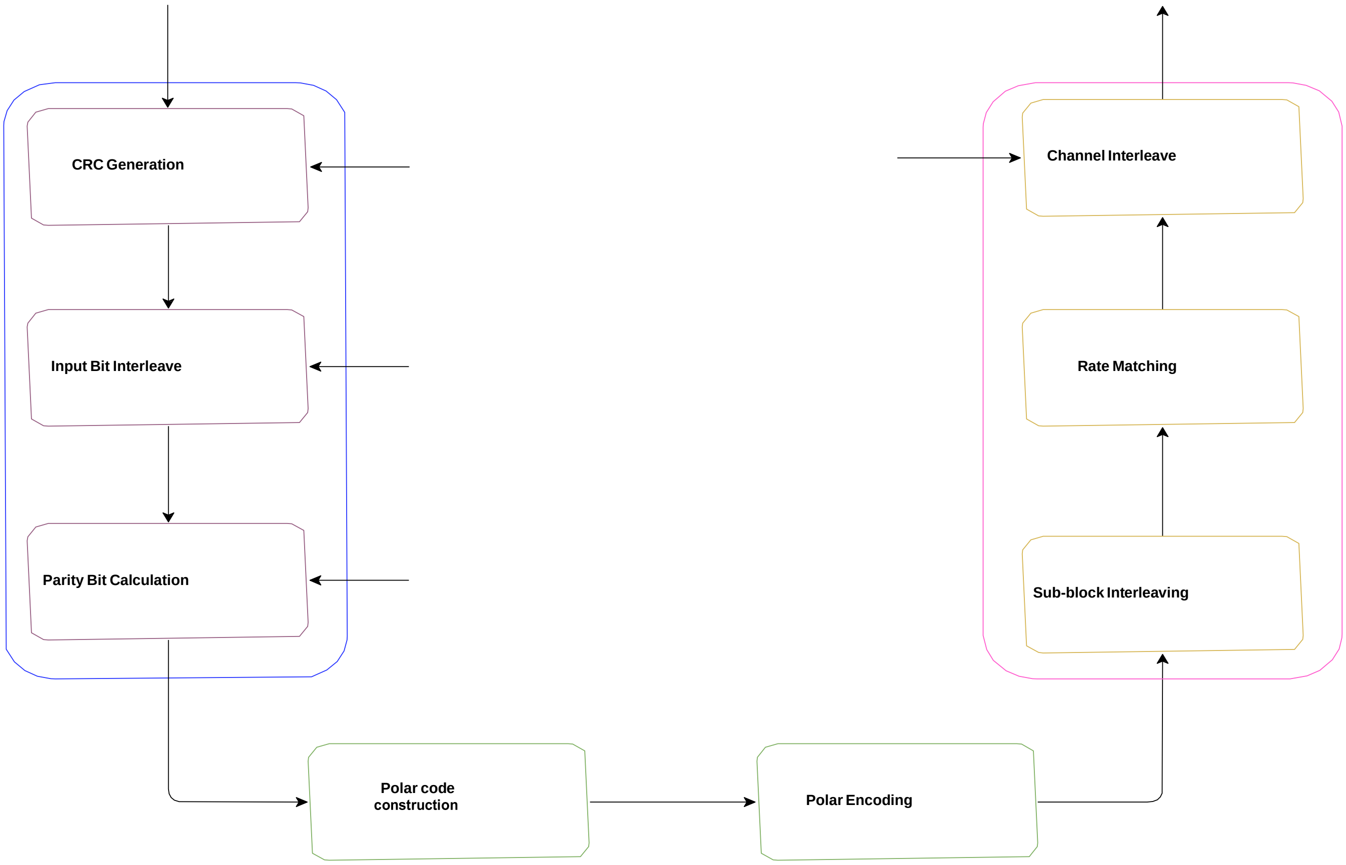

Figure 4. Generic Encoding FEC of 5G

To better understand the bottlenecks and optimizations performed in the software implementation of 5G FEC chain, it is necessary to understand the fundamentals of general-purpose processors architecture. This section gives necessary background about cache memory systems, instruction pipelining, branch predictors, vector processing units and recursive function calling mechanism.

2.1. Physical downlink control channel (PDCCH)

The PDCCH is another downlink control channel which uses polar codes. Resources requested by the UE are assigned by the base station. This resource allocation information is transmitted via PDCCH channel. PDCCH also carries information related to uplink power control, downlink resource grant and system paging information. The PDCCH contains a message called Downlink Control Information (DCI) which carries all the control information of UE. Payload size of PDCCH is not fixed. It varies based on the format of DCI, consequently, values of *A*, *N* and *E* vary. Type of DCI is configured from the higher layer. Except *A*, *N* and *E*, other parameters of the PDCCH polar FEC chain are same as PBCH.

2.2. Physical uplink control channel (PUCCH)

In uplink, PUCCH contains Uplink Control Information (UCI) similar to DCI in the downlink. UCI carries channel state information, acknowledgments, scheduling request. The payload size of PUCCH varies based on the PUCCH formats. PUCCH uses different channel coding techniques depending on payload size. When payload size A ≥ 12 polar codes are used. PUCCH polar FEC chain parameters also vary depending on the values of A and E.

• There are three different cases for PUCCH polar FEC chain parameters based on the values of A and E as presented below.

• Case 1. A ≥ 20

L = 11, nmax = 10, IIL = 0, IBIL = 1, and nP C = nwm = 0.

• Case 2. 12 ≤ A ≤ 19 and E − A ≤ 175

L = 6, nmax = 10, IIL = 0, IBIL = 1, and nP C = 3, nwm = 0.

• Case 3. 12 ≤ A ≤ 19 and E − A ≥ 175

L = 6, nmax = 10, IIL = 0, IBIL = 1, and nP C = 3, nwm = 1.

2.3 Physical Uplink Shared Channel (PUSCH)

PUSCH is another uplink channel which transmits UCI. In PUSCH, LDPC codes are used for encoding user data and polar codes for control information. The UE transmits UCI in PUCCH, if UE is not transmitting any user data to the base station. When UE is transmitting user data through PUSCH, UCI is also transmitted with PUSCH using the same modulation parameters. In other words, if PUSCH is using 64-QAM then same modulation technique is applied to UCI.

The construction of polar codes involves the identification of channel reliability values. Information bits are placed in the K (number of information bits) high reliable bit indices out of N (block-length) positions and remaining bits are set to zero. These N bits are passed through a polar encoding circuit to get the encoded bits. Selection is of reliability indices is done based on the code length and channel signal-to-noise ratio. Due to varying code length and channel conditions in 5G systems, a significant effort has been put into identifying the reliable indices which have good error correction performance over different code length and channel conditions. The 5G parameter indicates how unreliable the channel is, it is easy see that Z (W) takes values between [0, 1] better the channel smaller is the Z (W). Polarization for N → ∞ creates channels with either Z (W) → 0 or Z (W) → 1.

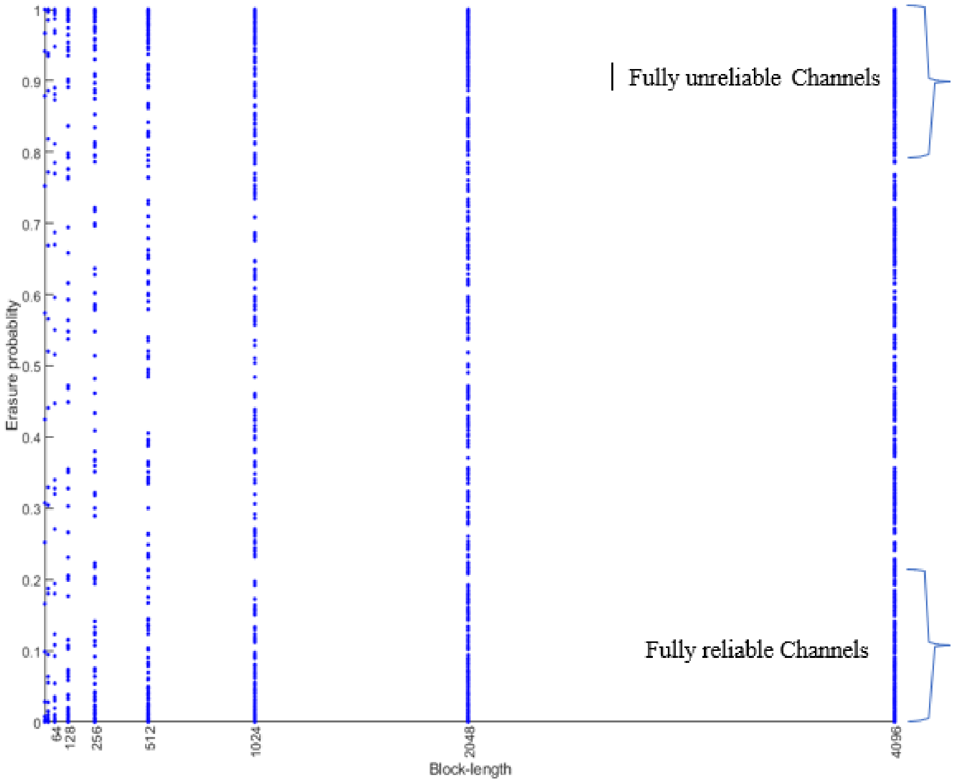

Figure 5. Channel polarization example for binary erasure channel

2.4 Cyclic redundancy checks (CRC) calculation

This solution presents the details of the polar encoding FEC chain in 5G with a block diagram. Future will explain the functionality and potential latency contribution of individual components in the FEC chain. Each of these individual components is extensively profiled to identify expensive operations and latency contribution. After identifying the bottlenecks, both algorithmic and software optimization techniques are employed. Algorithm optimizations include reformulation of the problem to avoid expensive operations, encoder tree pruning using lookup tables etc. Huge latency reduction is achieved through software optimizations as well. Some of the major software optimization methods are unrolling an encoder function, exploiting data parallelism with CRC, avoiding exponentially complex operations and finally reformulation of polar code construction to avoid expensive remove, erase and copying operations.

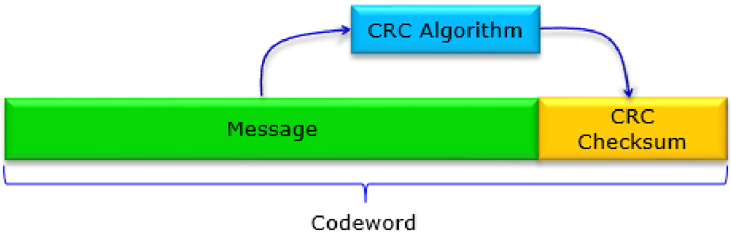

Figure 6. The CRC algorithm flow diagram for processing a message

Typically, in software implementations, for clarity and ease implementation each bit of information is represented with 32-bit or 64-bit integers. Due to the presence of only one bit of information in each integer, if 1024 bits need to be encoded or decoded then 1024 integers are involved in encoding or decoding process. However, this isn’t the case in hardware implementations since each bit can be processed in hardware description languages (HDL). Representing one bit of information using a 32 or 64-bit integer has the following disadvantages.

- Increased memory footprint: For 1024 bits, 64 • 1024 bits’ memory need to be allocated. It is equivalent to 8 kilobytes. Allocating and initializing this memory can introduce significant latency.

- Results in more cache misses: If more memory is allocated, more data needs to accessed from RAM which can result in more cache misses.
- Serializes encoding/decoding: General purpose processors have a data path width of 64-bit. If each bit is represented using a 64-bit integer, research is not using the capability of processing 64 bits simultaneously. Instead, each bit is processed sequentially. This can make encoding or decoding sequential although the processor is capable of processing multiple bits in parallel.

To avoid the above disadvantages and to enable data parallelism, this work tries to pack multiple bits into a single integer. Although packing multiple bits to a single integer has advantages, for some operations such as bitwise interleaving accessing each bit efficiently is very important. To exploit the advantages of bit packing as well as the advantages of accessing each bit separately, it is necessary to convert between the two. This is where the power of CRC instructions in modern processors comes into play. These processors come with special hardware instructions which help to efficiently pack and unpack data. Data bits are used in packed format when data parallelism needs to be exploited and in unpacked format when certain operations require bits to be accessed individually. These pack/unpack instructions are very efficient and have low latency. Details of the AMD processor's instructions with corresponding latencies.

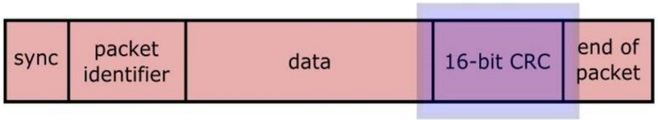

Figure 7. The single message distribution with 16-bit CRC

Information bits concatenated with CRC increases the error correction performance of polar codes significantly. CRC is used for selecting the correct code word out of potential candidates in the list. With CRC aided decoding, polar codes performance better than LDPC and Turbo codes at short block-lengths. To reduce the latency of encoding FEC chain CRC needs to be calculated very efficiently. A naive implementation of CRC calculation uses a shift register. This method calculates the CRC sequentially for one bit at a time.

2.5 Polar code construction

Polar code construction is the process of identifying information and frozen bit position, i.e. K out of N positions. This step determines the error correction performance of polar codes. There are many methods in the literature to construct polar codes. The parameter as reliability metric for Binary Erasure Channels (BEC) then deriving reliability values using simulation. For other channels, use more accurate density evolution (DE) methods but it suffers huge complexity. The CRC based Gaussian Approximation (GA) to reduce the complexity of DE with approximations. Still, the GA method has a high computational complexity which scales linearly with code block-length, therefore, it is unacceptable for varying SNR, block-length and code rate. In use cases such as 5G, where the channel is continuously varying. it is not feasible to construct polar codes on the fly due to stringent latency requirements of both encoder and decoder. The polar code construction in 5G takes a suboptimal approach, instead of constructing polar codes for every different SNR, block-length and code rate, construction is carried out in such a way that the constructed code performs sufficiently good over a large range of SNR, block-length and code rate. 5G polar code construction method is based on the contribution from Huawei which uses a β-expansion method with universal partial order (UPO) property of channel reliability.

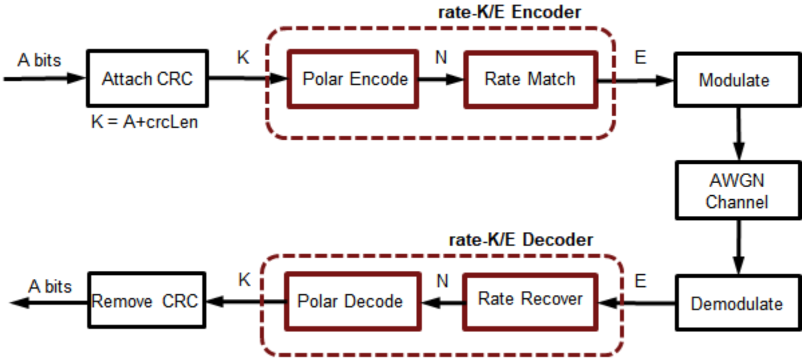

Figure 8. The block diagram represents the polar code conversion using CRC in 5G network

The 5G standard has adopted five different polar code block-lengths. Block-length sizes N are by N = {32, 64, 128, 256, 512, 1024}. For each of the block lengths, reliability indices values. The polar code construction also depends on the rate matching mode since it affects the reliability of bit indices. The polar code construction is straightforward when rate matching output E greater than or equal to block-length N. In such a case, code construction involves selection of K most reliable indices for information bits remaining positions are frozen since bit reliability not affected by rate matching.

$$E(N) = x_1$$

$$\frac{dy(t)}{dt} = x_2 = x_1$$

$$\frac{d^2y(t)}{dt^2} = x_3 = x_2$$

$$\frac{d^{n-1}y(t)}{dt^{n-1}} = x_n = \dot{x}_{n-1}$$

$$\frac{d^ny(t)}{dt^n} = \dot{x}_n$$

And $u(t) = u$

Then, the generic equation for 5G network is given by:

$$Bandwidth(N) = \dot{x}_n + kx_n + \ldots + k_1x_2 + kx_1$$

Rewriting the above state equation for 5G as given by.

$$\dot{x}_n = -k_0x_1 - k_1x_2 - .. - k_{n-1}x_n + B_0U_1(t)$$

The output equation is given by:

$$y(t) = y = x_1$$

Pertaining inversing the bandwidth transformation on both the sides.

$$\frac{d^2yt}{dt^2} + \frac{dy(t)}{dt} + y(t) = u(t)$$

Let

$$y(t) = x_1$$

$$\frac{dy(t)}{dt} = x_2 = x_1$$

And $K(t) = K$

Then, the state equation is

$$\dot{K}_N = -K_1 - K_2 + Q^i$$

The output equation is

$$E(N) = K = K_1$$

Where 'i' can be carried from 1 to 31.

The next optimization is avoiding copying sub-block interleaving pattern to frozen in- dices array in case of shortening. Instead, a sub-block interleaving pattern is directly used from the lookup table to mark the reliability indices as removed. In addition to above-mentioned optimizations, minor ones such as avoiding dynamic memory allocation instead reserving required memory in advance and employing pointer operations to avoid copying are performed. Finally, information bit positions are obtained from iterating the reliability table from the end (since indices are sorted in ascending order of reliabilities) and extracting K unmarked positions. These optimizations reduced the latency of polar code construction from 10.25 µs to 21.70 µs. Polar Code Construction Example for N = 32, K = 16 and E = N. Let's take an example with N = 32 channel reliability values are extracted from the reliability are given by:

Q^{31} = {0, 1, 2, 6, 3, 7, 9, 16, 4, 8, 11, 17, 13, 19, 20, 26, 5, 10, 12, 18, 14, 21, 24, 27, 15, 23, 22, 28, 25, 29, 30, 31}

2. Results

In this work, the algorithm is reformulated to avoid searching, copying and memory de-allocation while removing incapable bit indices. To avoid search operations, a lookup table is built whose values indicate the position of particular reliability value. After identifying the position, it is marked as removed instead of removing. Marking has two advantages first one is avoiding memory de-allocation and copying, the second one is keeping the same order of elements which is particularly useful for using the same lookup table for finding the next incapable bit index. After all the incapable bit indices are marked as removed, only the unmarked elements are considered as reliable bit positions for placing information bits.

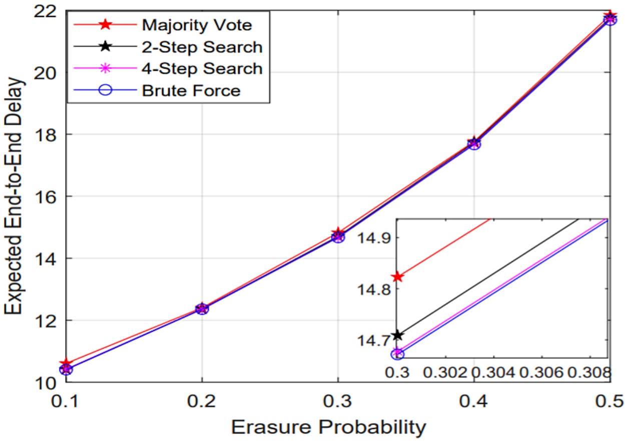

Figure 9. The extended end-to-end delay in communication with possible erasure probability

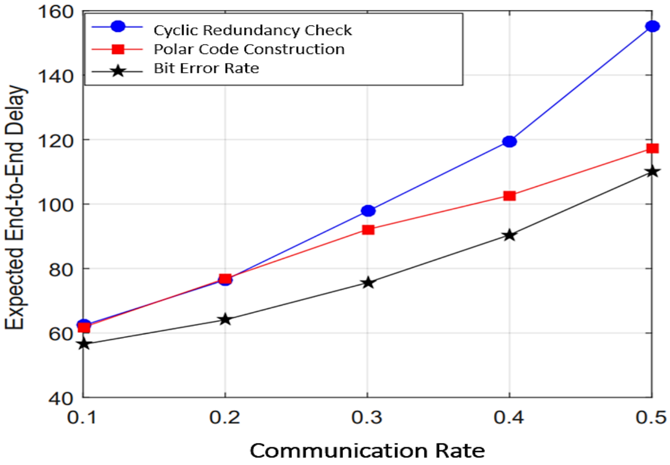

Figure 10. The communication rate in 5G with respect to proposed CRC technique

3. Discussion

As an outlook, for the above stated decoding FEC chain, decoder is developed with fast-SSC algorithm. This algorithm has much lower error correction performance than similar block-length LDPC and Turbo counterparts. As part of this work, CRC-Aided Successive Cancellation List (CRC-SCL) decoding algorithm is also implemented, however, it is not optimized for software. The CRC ideally suits very low SNR scenarios such as mm-Wave communication. It has approximately 1.5dB gain over fast-CRC algorithm for N = 2048 and list size L = 8. Ideal continuation of this work would be to extend the decoding chain by incorporating CRC algorithm to the FEC chain. It would be interesting to see the latency values of this algorithm, which has expensive sort and copying operations.

Table 1. Comparison of proposed technique with existing literature

Article	Technique	Communication Rate
[19]	Non-Orthogonal Multiple Access	137µs
[20]	Device-to-Device discovery scheme	96µs
[21]	Cross-link interference management	143µs
Proposed	Cyclic Redundancy Check	158µs

3. Conclusion

The objective of this research is to study the feasibility of developing polar FEC chain of 5G in software on general-purpose-processor while satisfying stringent latency requirements. In other words, all the components of encoder and decoder FEC chain are developed on general purpose AMD processor. The software satisfies latency constraint of less than 50µs. In the first part of the paper, research provides necessary background about polar encoding/decoding and computer architecture. In the second part, research develops encoding and decoding FEC chains and optimize them to satisfy the necessary latency constraints. To begin with, research provided necessary mathematical background about polar code construction, polar encoding, and decoding. Including different polar decoding algorithms. To understand FEC chain development in software it is necessary to know the basics of modern computer architecture. Computer architecture section talks about pipelining, cache memory and vector processing units in modern general-purpose processors. Research justifies about the details of polar encoding FEC chain. Research analyzed the different components of the FEC chain to identify latency contributors. Each of these latency contributors is further studied to reformulate the algorithm to avoid costly operations. Algorithms are reformulated to fit into specialized functional units of modern processors such as vector processing units. Vector processing units allow data parallelism in addition to supporting very fast mathematical computations. The encoding, major latency contributors were polar code construction, CRC calculation, encoding, and rate matching. A wide range of optimization techniques is employed to reduce the latency both algorithmic and platform specific. Namely, reducing algorithm complexity, using lookup tables, compiler hints for better instruction scheduling, vector processing instructions for data parallelism and avoiding superfluous copy operations et cetera. Optimizations reduced the worst-case latency of the encoding FEC chain from 158µs which is more than 10x reduction in latency with communication rate.

References

[1] F. Jameel, Z. Hamid, F. Jabeen, S. Zeadally, M. Javed. A survey of Device-to-Device communications: Research issues and challenges. IEEE Commun. Surv. Tutor. 2018, 20, 2133–2168.

[2] M. Noura, R. Nordin. A survey on interference management for Device-to-Device (D2D) communication and its challenges in 5G networks. J. Netw. Comput. Appl. 2016, 71, 130–150.

[3] P. Gandotra, R. Jha. Device-to-Device communication in cellular networks: A survey. J. Netw. Comput. Appl. 2016, 71, 99–117.

[4] Y. Zhang, Y. Shen, X. Jiang, S. Kasahara. Mode selection and spectrum partition for D2D inband communications: A physical layer security perspective. IEEE Trans. Commun. 2018, 67, 623–638.

[5] K. Chang, M. Yousaf, S. Hassan, T. Duong. Priority-Based device discovery in public safety D2D networks with full duplexing. In Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Springer Science and Business Media LLC: Berlin, Germany, 2019; Volume 272, pp. 102–108.

[6] Y. He, F. Yu, N. Zhao, H. Yin. Secure Social Networks in 5G Systems with mobile edge computing, caching, and Device-to-Device communications. IEEE Wirel. Commun. 2018, 25, 103–109.

[7] O. Hayat, R. Ngah, S. Hashim, M. Dahri, R. Malik, Y. Rahayu. Device discovery in D2D communication: A survey. IEEE Access 2019, 7, 131114–131134.

[8] Y. Yan, B. Zhang, C. Li. Opportunistic network coding based cooperative retransmissions in D2D communications. Comput. Netw. 2017, 113, 72–83.

[9] J. Zhang, L. Deng, X. Li, Y. Zhou, Y. Liang, Y. Liu. Novel Device-to-Device discovery scheme based on random backoff in LTE-advanced networks. IEEE Trans. Veh. Technol. 2017, 66, 11404–11408.

[10] H. Chour, Y. Nasser, H. Artail, A. Kachouh. VANET aided D2D discovery: Delay analysis and performance. IEEE Trans. Veh. Technol. 2017, 66, 8059–8071.

[11] B. Li, W. Guo, Y. Liang, C. Zhao. Asynchronous device detection for cognitive Device-to-Device communications. IEEE Trans. Wirel. Commun. 2018, 17, 2443–2456.

[12] N. Nguyen, K. Choi, L. Song, Z. Han: An unsupervised indoor peer discovery approach for LTE D2D communications. IEEE Trans. Veh. Technol. 2018, 67, 5069–5083.

[13] M. Luo, Y. Pei, and W. Huang, “Mutual heterogeneous signcryption schemes with different system parameters for 5G network slicings,” Wireless Networks, vol. 27, no. 3, pp. 1901–1912, Jan. 2021.

[14] L. Zhao, H. Wang, X. Zhong. Interference graph based channel assignment algorithm for D2D cellular networks. In IEEE Access 2018, 6, 3270–3279.

[15] S. Shamaei, S. Bayat, A. Hemmatyar. Interference management in D2D-enabled heterogeneous cellular networks using matching theory. IEEE Trans. Mob. Comput. 2018, 18, 2091–2102.

[16] Z. Lv, A. K. Singh, and J. Li, “Deep Learning for Security Problems in 5G Heterogeneous Networks,” IEEE Network, vol. 35, no. 2, pp. 67–73, Mar. 2021.

[17] R. D. Kaliski, “Heterogeneous Network Architecture and Device-to-Device Communications in 5G Cellular Networks,” Wiley 5G Ref, pp. 1–24, May 2020.

[18] S. CHOPRA and A. KAKKAR, “Performance Analysis of Adaptive Beamforming in a MIMO-millimeter Wave 5G Heterogeneous Wireless Network using Machine Learning,” Feb. 2021.

[19] M. Mishra, P. K. Singh, and A. Brahmachari, “A Robust Pest Identification System using Morphological Analysis in Neural Networks,” Period. Eng. Nat. Sci., vol. 7, no. 1, pp. 483–495, 2019.

[20] D. Pradhan, K. C. Priyanka, and Rajeswari, “SDR Network & Network Function Virtualization for 5G Green Communication (5G-GC),” Future Trends in 5G and 6G, pp. 183–203, Nov. 2021.

[21] J. Dion, J. Lallet, L. Beaulieu, P. Savelli and P. Bertin, "Cloud Native Hardware Accelerated 5G virtualized Radio Access Network," 2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2021, pp. 1061-1066, doi: 10.1109/PIMRC50174.2021.9569523.