https://doi.org/10.31436/iiumej.v23i2.2272

A NEW HARDWARE ARCHITECTURE FOR HIGH-PERFORMANCE PARALLEL TURBO DECODER

SUJATHA ELUKURU[1*], SUBHAS CHENNAPALLI[2] AND GIRIPRASAD MAHENDRA NANJAPPA[2]

[1]Department of Electronics and Communications,
Sree Vidyanikethan Engineering College, Andhra Pradesh, India.
[2]Department of Electronics and Communications,
JNTUA College of Engineering, Andhra Pradesh, India.

[*]Corresponding author: sujathaece88@gmail.com

(Received: 1st January 2022; Accepted: 26th March 2022; Published on-line: 4th July 2022)

ABSTRACT: Recent wireless communications demand maximum achievable data rates without intervention. The channel decoder in the physical layer would support such high data rates with a flexible hardware structure. The turbo channel decoder offers flexible hardware architecture and reliable decoding, but the turbo decoder design is complex, and its hardware architecture consumes more power and area in a communication system. Hence, an optimized high-performance turbo decoder architecture with simplified QPP interleaver is needed for supporting various data rates. In this context, this article presented a new hardware architecture with a three-stage pipeline parallel turbo decoding process and each MAP decoder in the proposed parallel turbo decoder with a three-stage micro pipeline process. The proposed structure optimized the circuit complexity and improved the throughput through parallel pipeline decoding process. Also, this article presents a simplified semi-recursive QPP interleaver, which avoids complex 'mod' operations for a high-performance turbo decoder. The performance analysis has been done using Model sim, Xilinx Vivado design suite, and estimated performance analysis was observed on various 28 nm CMOS technology FPGAs and compared with the conventional designs. Analysis of the proposed design showed improvement throughput up to 55.6% and a reduction in the power consumption up to 43% as compared to the recently reported architectures.

ABSTRAK: Komunikasi tanpa wayar terkini menuntut kadar data maksimum yang boleh dicapai tanpa intervensi. Penyahkod saluran dalam lapisan fizikal akan menyokong kadar data yang tinggi dengan struktur perkakasan fleksibel. Penyahkod saluran turbo menawarkan seni bina perkakasan fleksibel dan penyahkodan yang boleh dipercayai. Tetapi, penyahkod turbo merupakan blok yang kompleks, lebih berkuasa dan menggunakan kawasan yang luas dalam sistem komunikasi. Oleh itu, seni bina penyahkod turbo optimum berprestasi tinggi dengan antara lembar QPP yang mudah diperlukan bagi menyokong pelbagai kadar data. Dalam konteks ini, kajian ini merupakan seni bina perkakas baru dengan proses penyahkod turbo selari bersama salur paip tiga peringkat dan setiap penyahkod MAP yang dicadangkan dalam penyahkod turbo selari bersama proses saluran paip mikro tiga peringkat dibentangkan. Struktur yang dicadangkan dapat mengurangkan kerumitan litar dan meningkatkan daya pemprosesan melalui penyahkodan saluran paip selari. Selain itu, kajian ini merupakan antara lembar mudah QPP rekursif, yang dapat mengelakkan operasi 'mod' yang kompleks bagi penyahkod turbo berprestasi tinggi. Analisis prestasi telah dilakukan menggunakan sim Model, reka bentuk suit Xilinx Vivado, dan analisis prestasi anggaran telah diperhatikan pada pelbagai teknologi FPGA CMOS 28 nm dan dibandingkan dengan reka bentuk konvensional. Analisis reka bentuk yang dicadangkan menunjukkan peningkatan sepanjang 55.6% dan pengurangan penggunaan kuasa sehingga 43% berbanding seni bina laporan terkini.

KEYWORDS: *turbo decoder; MAP decoder; VLSI; interleaver; FPGA*

1. INTRODUCTION

Channel coding techniques are essential for a wireless communication system to achieve a reliable and high-performance transmission between transmitter and receiver, in a noisy channel. State-of-the-art iterative channel codes such as Turbo codes [1], Low-density parity-check codes (LDPC) [2], and Polar codes [3] are often used. Turbo codes offer more flexible architecture for their encoder and decoder than LDPC and polar codes. Also, Turbo codes achieve high diversity, reliable data transmission, and possible large coding gain in fading channels.

The efficient-hardware implementation of Turbo codes, in order to meet real-time constraints, is an active area of research and there is a need for innovation in the VLSI design of high-performance Turbo Decoders in terms of throughput, silicon area, and power-efficiency as well. Hence, the present study is aimed at developing a high-throughput, low area, and low power turbo decoder by modifying the hardware architecture of the decoder, simplifying mathematical computations involved in the decoding and interleaving process, and applying the optimization techniques. Maximum a-posteriori probability (MAP) algorithm introduced by Bahl-Cocke-Jelinek-Raviv (BCJR) [4] for SISO decoders and the simplifications of MAP algorithm called Log-MAP and Max-Log-MAP [5] were studied and Max-log-MAP algorithm is adopted in the design and hardware implementation of the proposed turbo decoder due to its lower complexity than the log-MAP algorithm.

To improve the throughput performance of the turbo decoder, the number of MAP decoders could be increased and all operated in parallel at the cost of degradation in error-correcting performance, especially with higher code rates. Moreover, employing multiple decoders to increase the throughput does not solve the additional challenge of lower latency requirements. The throughput could also be increased by increasing the block size (from 40 to 6144), but this would result in consequent complexities in computational latency, area requirement, and power consumption. The trade-off among the performance parameters could be best compromised by effective hardware design and suitable optimization techniques [6]. Interleaver is an essential part of turbo decoder and is also responsible for BER performance of decoding. The algebraic properties and contention-free property of QPP interleaver [7] guarantee contention-free access to memory and generated addresses.

Also, one of the problems in the implementation of highly parallel decoders is memory contention during decoding, where all the sub-block of MAP decoders simultaneously tries to access the same memory bank on reading or writing the extrinsic information from or into it. To solve this problem, an efficient approach of collision-free parallel interleavers in which data is read or written on the intra-sub block as well as inter-sub block of MAP decoder to achieve low complexity architecture having no additional hardware resources.

Some benchmarked research works on high throughput turbo decoders were discussed here. A high throughput turbo decoder with 8 and 64 parallel radix-2 MAP decoder architecture in 90 nm CMOS technology was proposed [8]. This paper proposed a new ungrouped backward recursion scheme and a new state metric normalization technique to offer retiming and pipelining in architecture for performance improvement. Also, this work adopted a fine-grain clock gating technique to solve the power issue and the throughput achieved is 301 Mbps at 272 mW of power. A highly parallel turbo decoder structure in

2015 was reported [9] to achieve the highest throughput rate of 1.45 Gbps implemented in 90 nm CMOS technology. This work was aimed at improving the decoding efficiency and this improvement was possible by modifying the parallel window MAP decoding algorithm.

A fully parallel turbo decoding [FPTD] algorithm was reported [10] which allows parallel processing to offer higher processing throughput. This novel FPTD algorithm reduced computational complexity by 50% and enhanced its suitability for FPGA implementations. It was concluded that the fully parallel turbo decoder with radix-2 and 6144 parallel MAP decoders resulted in 14.8Gbps but this design utilized 9618 mW of high power at 100 MHz clock frequency. Various VLSI architectures were presented in [11] for the computing blocks of the turbo decoder and made the SISO decoder support Radix- 2/4/8 modes. The design resulted in throughput in the range of 80Mbps to 270 Mbps, reducing power consumption to up to 61% as compared to the other state-of-art designs. A parallel turbo decoder with reverse address generator in interleaver for low latency and high throughput architecture with double buffer technique was proposed in [12] for effective utilization of FPGA resources for broadcasting systems. This work resulted in a throughput performance of 2.12 Gbps at 250 MHz and a latency of 23.2 µsec with 64 parallel map decoders.

A memory-reduced turbo decoder was proposed by a reverse recalculation technique using the Log-MAP algorithm with a focus on power reduction [13]. It was reported that the technique helped to reduce the memory and power consumption as compared to other conventional turbo decoder designs. The Vedic multiplier-based implementation presented in [14] could be preferred in-branch metric calculations in Max-log-MAP algorithms for low latency turbo applications, but the implementation consumes more area. An optimized turbo decoder for performance improvement of turbo decoder, where the parallel computation of state metrics, reusing of memory and single SISO decoder in the hardware implementation was proposed [15]. A low memory turbo decoder with reverse calculation techniques was reported where the trellis diagram was partitioned and the max* operator was simplified [16]. The findings revealed that the architecture achieved a 65% reduction in state metric cache (SMC) capacity with other designs and lower power dissipation. By this motivation, the present study focused on developing a new hardware architecture for parallel turbo decoder to achieve high performance and balanced hardware implementation using optimization techniques.

2. TURBO DECODER DESIGN PERSPECTIVE

The general structure of a turbo decoder consists of two SISO decoders connected through an interleaver and de-interleaver to perform the iterative process of soft bits to provide a-posteriori LLRs after the required number of iterations. The soft-demodulated values of transmitted bits are referred to as a-priori probability values and are fed to constituent SISO decoders as input LLRs, shown in Fig. 1 [17]. Each decoder operates on the systematic and parity bits associated with its constituent encoder and produces soft outputs of the original data bits in the form of a-posteriori probabilities. The extrinsic information is computed using a-posteriori probability values from the SISO decoder, interleaved/non-interleaved a-priori probability values, and interleaved/de-interleaved extrinsic information from another SISO decoder. Such extrinsic information values are shuffled between two SISO decoders and are iteratively processed along with a-priori probability values to produce error-free a-posteriori probabilities of the transmitted bits.

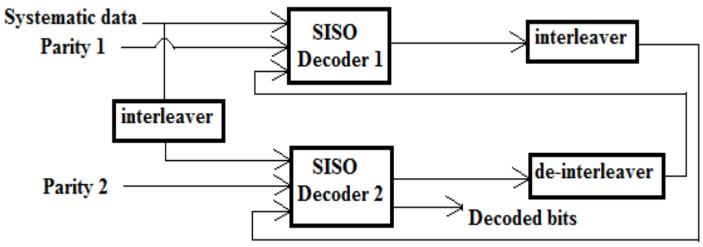


Fig. 1: Block diagram of Iterative Turbo Decoder.

In the iterative process, the MAP algorithm decodes the probabilities for each bit correctly. The complexity of the MAP algorithm has been reduced by operating the algorithm in the log domain variants such as the log-MAP algorithm and max-log-MAP algorithm. In order to realize the high-performance turbo decoder, SISO decoders involved in the turbo decoder should provide high-speed data transmission without significant coding loss. Major tasks of the SISO decoder are computation of branch metrics, state metrics, and LLR computation to extract the final extrinsic information. However, two SISO decoders do not work simultaneously in each half iteration to compute the state metrics. Hence, the present study utilized the turbo decoder with a single SISO decoder for one complete iteration as shown in Fig. 2.

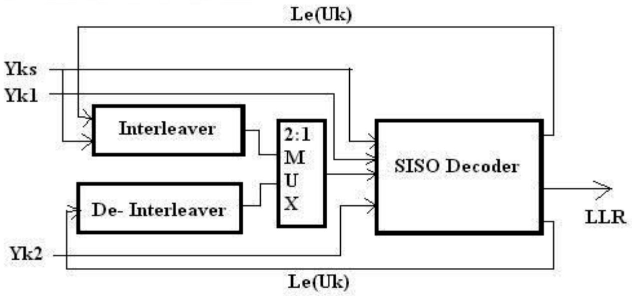


Fig. 2: Block diagram of Turbo Decoder with single SISO Decoder.

The main objective of the present study is to design an efficient parallel turbo decoder that can support higher throughputs using streaming techniques. QPP Interleaver plays a vital role in turbo encoder/decoder error correction. Hardware design of the QPP interleaver involves complex mathematical functions and dependency of previous computations. A semi recursive QPP interleaver is proposed in the present study which simplifies the aforementioned disadvantages of QPP interleaver.

3. QPP INTERLEVER

QPP interleaver is an integral part of the turbo encoder and decoder and it plays a critical role in turbo codes, especially in turbo decoder, for achieving high-speed decoding. For each of the 188 block lengths, a different set of f_1, f_2 parameters were pre-defined in 3GPP LTE [17]. In the recent 3GPP LTE/ LTE-A, QPP interleaver is based on algebraic properties and contention-free properties, providing contention-free memory access for any specified code block size between 40 to 6144. The efficient design of a conflict-free reconfigurable QPP interleaver for turbo encoder and turbo decoder is a pre-eminent task in turbo channel coding scheme. The hardware implementation of QPP interleaver/de-interleaver should support parallel interleaving for the high-performance parallel decoder. This research work proposed the design of reconfigurable semi-recursive QPP interleaver for parallel and direct computation of address locations of all the bits for turbo decoder by semi recursive computation approach as explained below.

3.1 Semi Recursive QPP Interleaver

The mathematical complexity and dependency of the current address location of the previous address location are solved by the semi-recursive computation method. The address locations of interleaved bits/the sequence of numbers of interleaving $\pi(i)$ of current symbol i in QPP interlever is computed as

$$\pi(i) = (f_1 i + f_2 i^2)\ modK \tag{1}$$

In Eq. 1 [17], parameters $f_1 and\ f_2$ depend on $'K'$and all the possible variants of block size K and variables $f_1 and\ f_2$, are defined. In hardware implementation of Eq.1, the address computation of current index i, depends on previous computations recursively and this recursive dependency creates high decoding latency and is not preferable for high-performance turbo decoders. The proposed design does not contain mod operation, as mod operator implementation is complex in the hardware design of the QPP interleaver; it is replaced by an Add-Compare-Select (ACS) unit. The ACS unit is composed of only arithmetic operators like addition and subtraction. Replacing the mod operation by the ACS unit is called the modulo normalization technique.

To simplify the complex interleaver computation and to avoid large storage requirements, the proposed semi recursive computation approach for parallel interleaver supports the independent parallel computation of interleaved addresses. The input sequence (Num) is denoted as Metric Weight (MW) and it is represented as MW (1, K+1) in the first column. The subsequent columns are defined as mentioned below.

Case 1: If mod (Num, 2) ≠0, then Num = Num+1 and

MW (2, K+1) = (Num+1)/2;
MW (3, K+1) = MW (2, K+1) - 1;

Case 2: If mod (Num, 2) =0, then Num= Num;
MW (2, K+1) = Num/2;
MW (3, K+1) = Num - MW (2, K+1);

Then, Value (V) is defined as,
V = δ(0) x MW (2, K+1) + δ(1) x MW (3, K+1),
where, $\delta(0) = f_1 + f_2$ and $\delta(1) = \delta(0) + 2f_2$;

It can be observed from Tables 1 and Table 2, that computation of address locations of 40 bits was done within 5 clock cycles independently. This approach is proposed to minimize the computational complexity and avoid the storage of interleaver tables.

Table 1: Metric weight table in semi recursive order

MW (1, K+1), Num	MW (2, K+1)	MW (3, K+1)	Value (V)	Π(i) = mod (V,40)
0	0	0	0	0
1	1	0	13	13
2	1	1	46	6
3	2	1	59	19
4	2	2	92	12

Table 2: Proposed Parallel Computation of Sub blocks

CLOCK	Sub-block1	Sub-block2	Sub-block3	Sub-block4	Sub-block5	Sub-block6	Sub-block7	Sub-block8
Clock1	0	5	10	15	20	25	30	35
Clock2	1	6	11	16	21	26	31	36
Clock3	2	7	12	17	22	27	32	37
Clock4	3	8	13	18	23	28	33	38
Clock5	4	9	14	19	24	29	34	39

From Tables 1 and 2, it can be observed that the parallel computation of 40 bits has been done with 8 parallel operations. In the first clock cycle, bits 0, 5, 10, 15, 20, 25, 30 and 35 will be computed simultaneously. Similarly in the second, third, fourth, and fifth clock cycles, the parallel computation of the remaining bits is performed in the order shown in Table 2. The proposed method is most suitable for highly parallel turbo decoding architectures. The proposed design and FPGA implementation of a new hardware architecture for a high-performance turbo decoder using streaming techniques is presented below.

4. PARALLEL TURBO DECODER

The parallel decoding approach of turbo decoder with P parallel MAP decoders roughly increases the decoding throughput by a factor of 'P' compared to non-parallel turbo-decoders. Modern parallel hardware architectures can have either spatial or functional parallelization to improve the throughput performance. For a high-performance turbo decoder, this article proposed a new hardware architecture, which is an 8-parallel MAP decoder structure. The proposed architecture is designed in a three-stage pipelined process.

In the first stage, the input LLRs load into the three buffers namely systematic buffer, parity-1 buffer and parity-2 buffer in parallel. Here, the input LLRs could be related to any of 188 block sizes varying from 40 to 6144. In the second stage, the data of eight coded words are processed parallel with the eight BCJR decoders as shown in Fig. 3.

In the second stage, the BCJR decoder is further processed into three micro pipeline stages. Two SISO decoders, named SISO-1 and SISO-2, the first decoder processes the systematic input, parity-1 and a-priori data. Similarly, the second decoder process interleaved systematic input, parity-2, interleaved a-priori data in the micro-pipeline stage is presented below.

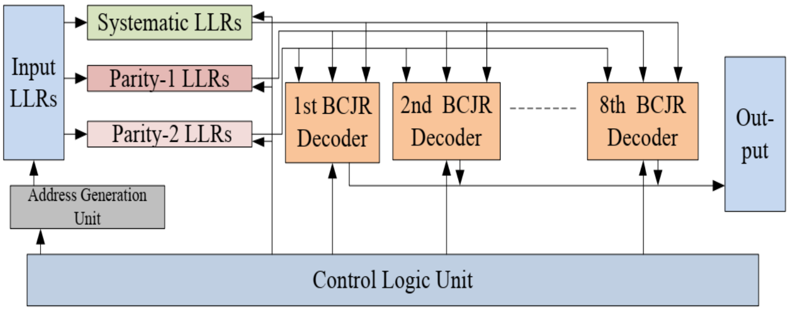


Fig. 3: Block diagram of proposed three-stage pipeline parallel turbo decoder.

In the first micro pipeline stage, all the SISO decoders are processed in parallel with the given two inputs and then extrinsic information is produced as the output of the SISO decoder. In the second micro-pipeline stage, the produced output information is processed to interleaved/ de-interleaved block. Finally, in the third micro-pipeline stage, the third input of de-interleaved a-priori data to SISO decoder blocks to process the extrinsic information. This three-stage micro pipeline process continues for 8 number of iterations. This process is depicted in Fig. 4. Then the third pipeline stage of the parallel '8' turbo decoder continues until maximum convergence is achieved and the output LLRs are processed into the output buffer.

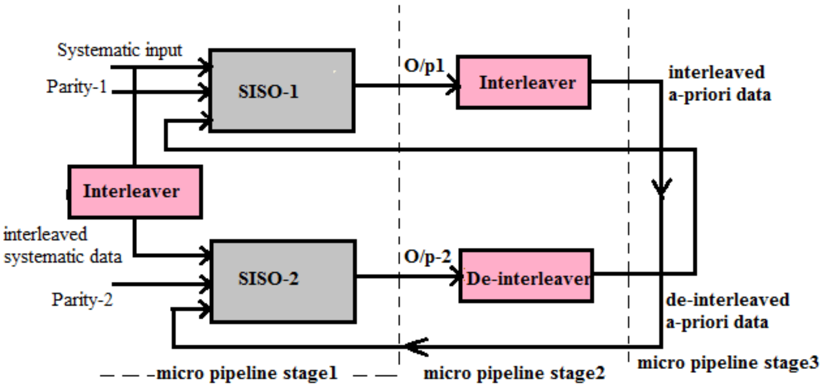


Fig. 4: Block diagram of three stages micro pipeline Turbo Decoder.

4.1 Simplified Computation of Soft-output

The soft output L can be computed as shown in Eq. 2 [18] from the state metrics and branch metrics to find maximum value as,

$$\begin{aligned} L = \max\,(&\alpha_0' + \beta_0 + \gamma_{00}, \alpha_1' + \beta_4 + \gamma_{00}, \alpha_2' + \beta_5 + \gamma_{01}, \alpha_3' + \beta_1 + \gamma_{01}, \\ &\alpha_4' + \beta_2 + \gamma_{01}, \alpha_5' + \beta_6 + \gamma_{01}, \alpha_6' + \beta_7 + \gamma_{00}, \alpha_7' + \beta_3 + \gamma_{00}) - \\ \max\,(&\alpha_0' + \beta_4 + \gamma_{11}, \alpha_1' + \beta_0 + \gamma_{11}, \alpha_2' + \beta_1 + \gamma_{10}, \alpha_3' + \beta_5 + \gamma_{10}, \\ &\alpha_4' + \beta_6 + \gamma_{10}, \alpha_5' + \beta_2 + \gamma_{10}, \alpha_6' + \beta_3 + \gamma_{11}, \alpha_7' + \beta_7 + \gamma_{11}) \end{aligned} \tag{2}$$

where, α_0' *to* α_7' denotes the forward state metrics, β_0 to β_7 denotes backward state metrics of 8 states and γ_{00} *to* γ_{11} denotes branch metrics.

Equation 2 is further simplified as Eq. 3 in our proposed simplification for computing soft output (L) with common γ_{00} *to* γ_{11} ,

$$L = \max\,(\max(s_0, s_1) + \gamma_{00}, \max(s_2, s_3) + \gamma_{01} - \max(t_0, t_1) + \gamma_{11}, \max(t_2, t_3) + \gamma_{10}\ ) \tag{3}$$

where,

$$s_0 = \max\,(\alpha_0' + \beta_0, \alpha_1' + \beta_4)$$

$$s_1 = \max\,(\alpha_6' + \beta_7, \alpha_7' + \beta_3)$$

$$s_2 = \max\,(\alpha_2' + \beta_5, \alpha_3' + \beta_1)$$

$$s_3 = \max\,(\alpha_4' + \beta_2, \alpha_5' + \beta_6)$$

$$t_0 = \max\,(\alpha_0' + \beta_4, \alpha_1' + \beta_0)$$

$$t_1 = \max\,(\alpha_6' + \beta_3, \alpha_7' + \beta_7)$$

$$t_2 = \max(\alpha_2^{'} + \beta_1, \alpha_3^{'} + \beta_5)$$

$$t_3 = \max(\alpha_4^{'} + \beta_6, \alpha_5^{'} + \beta_2)$$

The extrinsic information/ a-posteriori information $\lambda_{out}(k)$ can be calculated as in Eq. 4 [18], with the aid of L(k), x(k) and y(k) as,

$$\lambda_{out}(k) = \frac{1}{2}L(k) - x(k) - \lambda_{in}(\mathrm{k}) \tag{4}$$

where, $L(k)$ denote soft-output, x(k) is the received soft systematic information, λ_{in}(k) is a-priori information.

4.2 Performance Analysis

The performance analysis of the channel decoder can be done by decoding delay/latency and the throughput obtained. But a hardware digital system/circuit performance will be measured in three parameters called power, area, and throughput. This analysis can be done when the proposed architecture is synthesized by hardware design tool like Xilinx ISE/Vivado.

For the proposed design of turbo decoder, the decoding delay is calculated as Eq. 5 and 6 [17] for block sizes less than 264 and from 264 to 6144,

If K< 264,

$$D = (26 + (2f(K,N) + 14)2I) \tag{5}$$

If K≥ 264,

$$D = (26 + (f(K,N) + 46)2I) \tag{6}$$

where, K denote block size, N denote number of decoders and I denote number of iterations and

$$\mathrm{f\,(K,N)} = \begin{cases} \frac{\mathrm{K}}{\mathrm{N}} \text{if K is divisible by N} \\ \frac{\mathrm{K}}{8} \text{if K is not divisible by N} \end{cases}$$

Decoding latency (L) is calculated as

$$L = \frac{D}{f_{max}}\ sec \tag{7}$$

The throughput (T) is calculated as

$$T = \frac{[K * f_{max}]}{D} bps \tag{8}$$

where, f_{max}denote the maximum operating frequency, which effects both latency and throughput as in Eq. 7 and 8 [17].

For instance, if the operating frequency of this hardware is about 250 MHz, then the throughput for the block size of 40 bits is 24.38 Mbps and for block size of 6144 bits is 117.7 Mbps.

5. RESULTS AND DISCUSSION

In order to get a higher throughput and lower latency, the most commonly adopted design methodology is to improve the level of parallelism. A new architecture consisting of an 8-parallel decoder structure has been proposed for the high-performance turbo decoder proposed in the present study. The proposed hardware architecture of the turbo decoder is designed into the three-stage pipeline and three-stage micro-pipeline procedures for high performance. The high-level block diagram is shown in Fig. 5, the simulation waveform and performance analysis of the proposed design are discussed below.

The three-stage pipeline and three-stage micro-pipeline procedure in the proposed parallel turbo decoder is to improve the speed of data processing in the whole structure to improve the throughput and to reduce the latency. The proposed architecture has been designed, simulated in MAT Lab and Modelsim for functionality verification and the simulation waveform is shown in Fig. 6. From the simulation diagram, the throughput latency, or the time taken to produce the first output for the given input, is 1.9 ns.

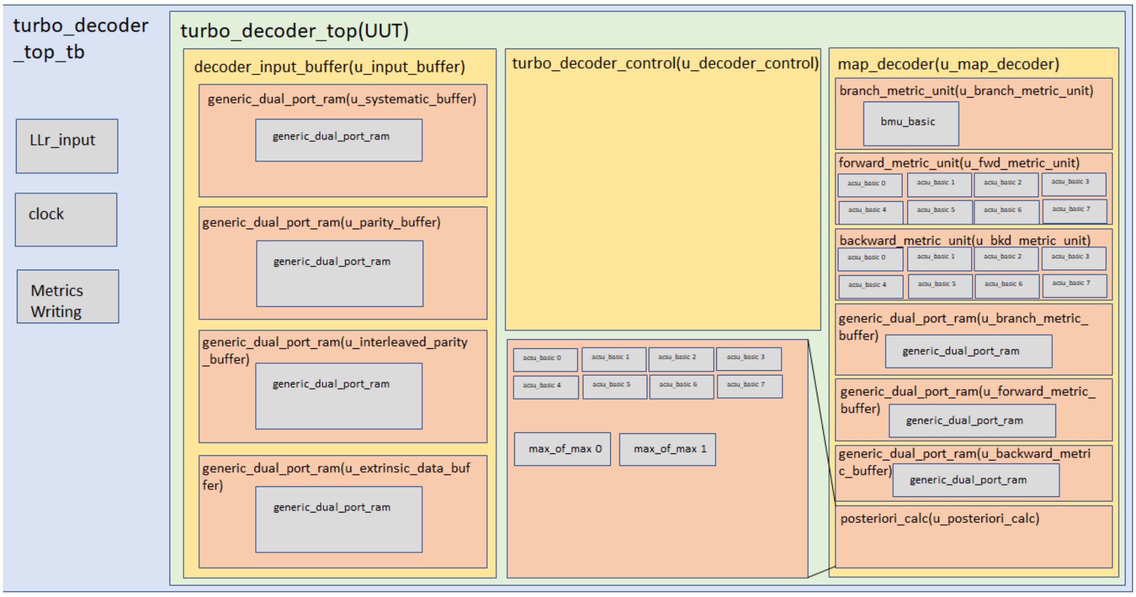
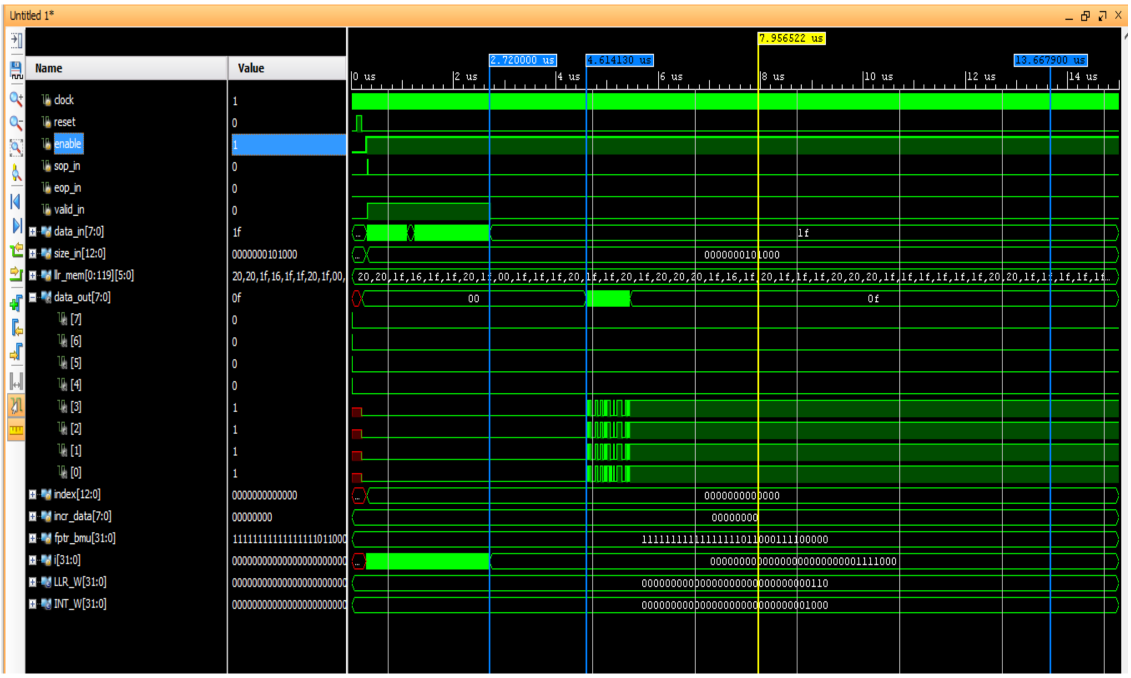

turbo_decoder
_top_tb
LLr_input
clock
Metrics Writing
turbo_decoder_top(UUT)
decoder_input_buffer(u_input_buffer)
generic_dual_port_ram(u_systematic_buffer)
generic_dual_port_ram
generic_dual_port_ram(u_parity_buffer)
generic_dual_port_ram
generic_dual_port_ram(u_interleaved_parity_buffer)
generic_dual_port_ram
generic_dual_port_ram(u_extrinsic_data_buffer)
generic_dual_port_ram
turbo_decoder_control(u_decoder_control)
acsu_basic 0
acsu_basic 1
acsu_basic 2
acsu_basic 3
acsu_basic 4
acsu_basic 5
acsu_basic 6
acsu_basic 7
max_of_max 0
max_of_max 1
map_decoder(u_map_decoder)
branch_metric_unit(u_branch_metric_unit)
bmu_basic
forward_metric_unit(u_fwd_metric_unit)
acsu_basic 0
acsu_basic 1
acsu_basic 2
acsu_basic 3
acsu_basic 4
acsu_basic 5
acsu_basic 6
acsu_basic 7
backward_metric_unit(u_bkd_metric_unit)
acsu_basic 0
acsu_basic 1
acsu_basic 2
acsu_basic 3
acsu_basic 4
acsu_basic 5
acsu_basic 6
acsu_basic 7
generic_dual_port_ram(u_branch_metric_buffer)
generic_dual_port_ram
generic_dual_port_ram(u_forward_metric_buffer)
generic_dual_port_ram
generic_dual_port_ram(u_backward_metric_buffer)
generic_dual_port_ram
posteriori_calc(u_posteriori_calc)

Fig. 5: High level block diagram of proposed turbo decoder.



Name	Value
clock	1
reset	0
enable	1
sop_in	0
eop_in	0
valid_in	0
data_in[7:0]	1f
size_in[12:0]	0000000101000
llr_mem[0:119][5:0]	20,20,1f,16,1f,1f,20,1f,00,
data_out[7:0]	0f
[7]	0
[6]	0
[5]	0
[4]	0
[3]	1
[2]	1
[1]	1
[0]	1
index[12:0]	0000000000000
incr_data[7:0]	00000000
fptr_bmu[31:0]	1111111111111111011000
i[31:0]	0000000000000000000000
LLR_W[31:0]	0000000000000000000000
INT_W[31:0]	0000000000000000000000

Fig. 6: Simulation waveform of proposed parallel turbo decoder.

Then, the RTL schematic shown in Fig. 7 is observed for the proposed architecture in detail for hardware components utilized. Also, the submodules of the proposed parallel decoder, like branch metrics, parallel state metric computations, and LLR computations, are run to find the maximum value of the computed posteriori LLRs to finalize whether the decoded bit belongs to either "0" or "1".

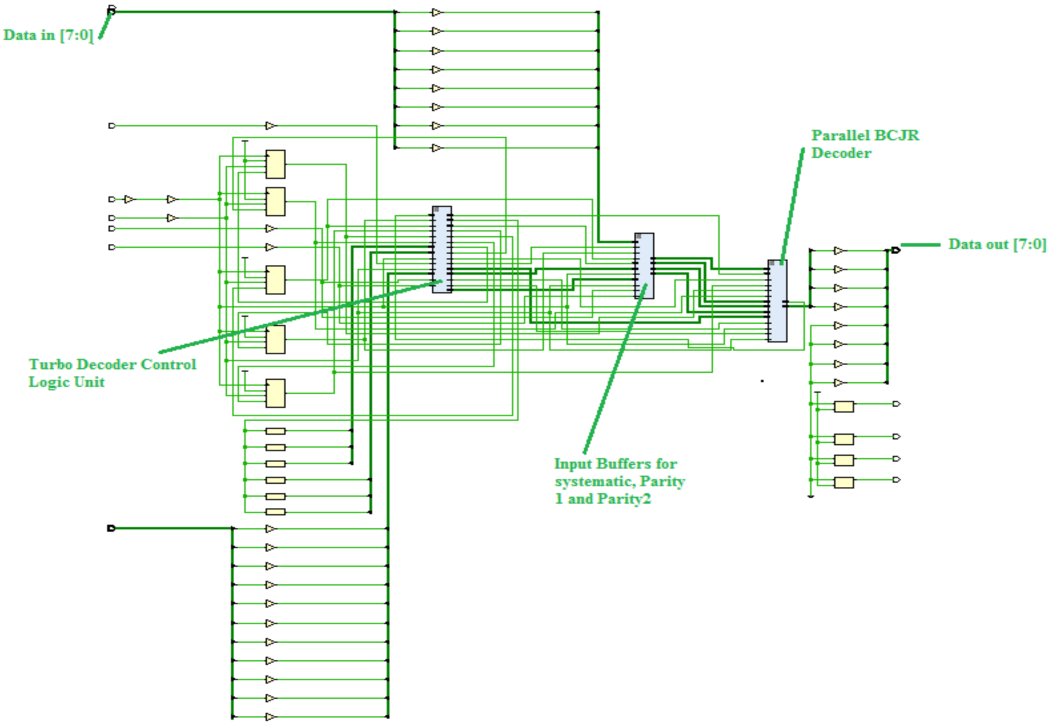

Data in [7:0]
Parallel BCJR Decoder
Data out [7:0]
Turbo Decoder Control Logic Unit
Input Buffers for systematic, Parity 1 and Parity2

Fig. 7: RTL schematic of proposed parallel turbo decoder using Xilinx VIVADO.

The architecture is implemented over Xilinx Vivado for 28 nm CMOS technology Kintex 7, Vertex-7, and Zynq-7000 Zed FPGA evaluation boards for its performance analysis. The hardware utilization is summarized in Table 3. It can be observed from Table 3 that a much smaller number of logic cells and memory cells are occupied by the proposed design with VLSI optimization techniques than the standard design. As ACS units have been used for metric computation, instead of many arithmetic/logical units, the hardware resource utilization has been reduced. It is evident from the observation that hardware utilization is less at post-implementation than post-synthesis of the design.

Table 3: Hardware resource utilization of parallel turbo decoder

Hardware Resource		Utilization (%) Post-Synthesis	Utilization (%) Post-Implementation	Available
Utilization (%)	FF	809 (0.76%)	809 (0.76%)	106400
	LUT	1072 (2.02%)	1059 (1.99%)	53200
	I/O	38 (19%)	34 (17%)	200
	BRAM	32 (22.86%)	32 (22.86%)	140
	BUFG	1 (3.12%)	1 (3.12%)	32
Power consumption (in Watt)			0.157	

Once the functionality is proven, then the netlist of the design is ready for further processing. Synthesized-netlist has been placed, routed, and checked for timing violations.

The timing report was generated for the proposed design and the critical path delay of 3.04ns and the respective maximum operating clock frequency obtained was 329MHz, as presented in Table 4.

Table 4: Throughput, latency, and power utilization of the proposed turbo decoder

Platform	Critical path delay ns	Max.Clock frequency f_{max} MHz	Block size K	Latency L µs	Throughput T Mbps
Kintex-7 28 nm	3.04	329	40	1.34	32
CMOS	3.04	329	6144	39.67	155

The proposed parallel turbo decoder on Xilinx Kintex-7 FPGA, achieved a throughput of 155 Mbps and 32 Mbps, and the latency of 39.67 µs and 1.34 µs for the block lengths of 6144 and 40, respectively. Furthermore, maximum clock frequency f_{max} of 329 MHz was observed as listed in Table 4. It can be seen from Table 4 that this parallel design achieved 155 Mbps of throughput at maximum flock frequency of 329 MHz and 39.67 µs of latency for block size 6144 on 28 nm CMOS Kintex-7 FPGA.

The proposed parallel architecture with these techniques gives reduction in energy consumption of the proposed architecture compared to the general architecture. The estimated performance analysis of the proposed turbo decoder on various Xilinx FPGA and the comparison of obtained results with other recent turbo decoder designs are shown in Table 5. It is observed that the present work provides a balanced design between performance parameters of speed, area, and power. It is evident from the results that for similar Algorithm, block size, and approximately the same number of interactions, the proposed turbo decoder gives a much better throughput.

Table 5: Comparison of the proposed Turbo decoders with other reported works

Parameter	Z Yan 2016 [19]	Hua. L 2017 [18]	Vadim B 2017 [20]	Rahul 2018 [11]	Farzana 2019 [21]	Farzana 2019 [21]	**Present work**	**Present work**
Target device/ FPGA family	130 nm CMOS	28 nm Vertex-7	Virtex-7 28nm	28 nm Zynq	28 nm Vertex-7	28 nm Vertex-7	**28 nm Vertex-7/Zynq**	**Kintex-7**
Parallelism/Radix	08-Apr	64	-	8/2	8	-	**8**	**8**
Algorithm	Max-Log MAP	Max-Log MAP	Max-log-MAP	Max-log-MAP	Max-log-MAP	Max-log-MAP	**Max-log-MAP**	**Max-log-MAP**
Block size	6144	6144	6144	6144	6144	6144	**6144**	**6144**
Number of iterations	5.5	8	5	8	8	8	**8**	**8**
Maximum clock rate (MHz)	290	250	270.9	276	86.3	86.3	**252.5**	**329**
Throughput (Mbps)	384.3	2120	5	80	86.3	10.7	**118**	**155**

6. CONCLUSIONS

The present study highlights the concept of a new architecture with a three-stage pipelined parallel turbo decoder and three-stage micro-pipelined MAP decoder. These techniques have specifically improved the throughput and operating clock frequency by pipelined parallel implementation of the turbo decoder and shortened the critical path delay in the whole design. Algorithmic approximation and architectural optimization like pipelining and parallelizing were used to minimize the critical path and attain a higher throughput. However, the hardware complexity advances linearly as the number of sub-blocks or iterations increases and increased recursions in architecture of the MAP decoder normally limit the throughput of the turbo decoder. The estimated performance has been observed by implementing the proposed parallel turbo decoder at 28 nm CMOS technology Xilinx Kintex7 FPGA and achieved a maximum estimated throughput of 155 Mbps with 8 iterations, which is suitable for 3GPP-LTE-Advanced, as per its specification. The proposed design improved throughput to the tune of 55.6% as compared to other recently reported designs.

From the performance analysis of the proposed turbo decoders and comparison with other recent turbo decoder designs, it is evident that the proposed architecture provides a balanced design among performance parameters, speed, and area. It can be concluded that throughput increases for the optimized turbo decoder and parallel turbo decoder architectures as compared to the standard design. However, the area requirement or power consumption increases proportionately with the throughput.

ACKNOWLEDGEMENT

The authors would like to thank the editors and anonymous reviewers for their insightful comments and constructive suggestions. This work was supported by the Department of Science and Technology, Government of India under women Scientist Scheme-A (WOS-A) (SR/WOS-A/ET-72/2017) and the work was carrying out at Sree Vidyanikethan Engineering College, Tirupati, Andra Pradesh, India.

REFERENCES

[1] Berrou C, Glavieux A, Thitimajshima P. (1993) Near Shannon Limit Error Correcting Coding and Decoding: Turbo-Codes. Proceedings of IEEE International Conference on Communication: pp 1064-1070. doi: 10.1109/ICC.1993.397441

[2] Mackay DJC, Neal RM. (1996) Near Shannon limit performance of low density parity check codes. Electronics Letters, 32(18): 1645-1646. doi: 10.1049/el:19961141

[3] Arıkan E. (2009) Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels. IEEE Transactions on Information Theory, 55(7): 3051-3073. doi: 10.1109/TIT.2009.2021379

[4] Bahl L, Cocke J, Jelinek F, Raviv J. (1974) Optimal decoding of linear codes for minimizing symbol error rate (corresp.). IEEE Transactions on Information Theory, 20(2): 284-287. doi: 10.1109/TIT.1974.1055186

[5] Robertson P, Villebrun E, Hoeher P. (1995) A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain. Proceedings of IEEE Intenational Conference on Communications: pp 1009-1013. doi: 10.1109/ICC.1995.524253

[6] Parhi KK. (1999) VLSI Digital Signal Processing Systems: Design and Implementation. Hoboken, NJ: Wiley.

[7] Nimbalker A, Blankenship TK, Classon B, Fuja TE, Costello DJ. (2008) Contention-free interleavers for High Throughput Turbo Decoding. IEEE Transactions on Communications, 56(8): 1258-1267. doi: 10.1109/TCOMM.2008.050502

[8] Shrestha R, Paily R. (2014) High-Throughput Turbo Decoder with Parallel Architecture for LTE Wireless Communication Standards. IEEE Transactions on Circuits and Systems—I: Regular Papers, 61(9): 2699-2710. doi: 10.1109/TCSI.2014.2332266

[9] Jing-shuin L, Ming-Der S, Chung-Yen L, Der-Wei Y. (2015) Efficient Highly Parallel Turbo Decoder for 3GPP LTE-Advanced. International Symposium on VLSI Design Automation and Test (VLSI-DAT): pp 1-4.

[10] Li A, Xiang L, Chen T, Maunder RG, Al-Hashimi BM, Hanzo L. (2016) VLSI Implementation of Fully Parallel LTE Turbo Decoders. IEEE Access, 4: 323-346. doi: 10.1109/ACCESS.2016.2515719

[11] Shrestha R, Sharma A. (2018) VLSI-Architecture of Radix-2/4/8 SISO Decoder for Turbo Decoding at Multiple Data-rates. Proceedings of IFIP/IEEE International Conference on Very Large Sale Integration (VLSI-SoC): pp 131-136. doi: 10.1109/VLSI-SoC.2018.8644753

[12] Luo H, Zhang Y, Li-ke H, Cosmas J. (2017) Low Latency Turbo Decoder implementation for Future Broad Casting Systems. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting: pp 1-4. doi: 10.1109/BMSB.2017.7986227

[13] Shi Y, Zhan M, Zeng J. (2018) FPGA Implementation and Power estimation of a memory reduced LTE-Advanced Turbo decoder. Proceedings of IEEE Conference on IoT, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Block Chain, Computer and Information Technology: pp 607-611. doi:10.1109/Cybermatics_2018.2018.00124

[14] Narayanan A, Murugan S, Bhakthavatchalu R. (2019) Low Latency Max Log MAP based Turbo Decoder. Proceedings of International Conference on Communication and Signal Processing (ICCSP): pp 721-724. doi: 10.1109/ICCSP.2019.8697955

[15] Sujatha E, Subhas C, Giriprasad MN. (2019) Performance improvement of Turbo decoder using VLSI optimization Techniques. IEEE International Conference on Vision, Towards Emerging Trends in Communication and Networking (ViTECoN): pp 84-90. doi: 10.1109/ViTECoN.2019.8899585

[16] Zhan M, Pang Z, Yu K, Wen H. (2021) Reverse Calculation-Based Low Memory Turbo Decoder for Power Constrained Applications. IEEE Transactions on Circuits and Systems I: Regular Papers, 68(6): 2688-2701. doi: 10.1109/TCSI.2021.3068623

[17] Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and Channel Coding. (2008) 3GPP Technical Specification Group Radio Access Network TS 36.212 Rev. 8.3.0 Release 9.

[18] Hua L, Yue Z, Li-ke H, John C. (2017) Low Latency Turbo Decoder implementation for Future Broad Casting Systems. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Cagliari, Italy: pp 1-4.

[19] Yan Z, He G, He W, Wang S, Mao Z. (2016) High Performance Parallel Turbo Decoder with Configurable Interleaving Network for LTE application: Integration. The VLSI Journal, 52: 77-90. https://doi.org/10.1016/j.vlsi.2015.05.003

[20] Belov V, Mosin S. (2017) FPGA implementation of LTE turbo decoder using MAX-log MAP algorithm. Sixth Mediterranean Conference on Embedded Computing (MECO): pp 1-4. doi: 10.1109/MECO.2017.7977157

[21] Farzana S, Butt MFU, Agha S, Ng SX, Maunder RG. (2019) Performance Analysis of High Throughput MAP Decoder for Turbo Codes and Self Concatenated Convolutional Codes. IEEE Access, 7: 138079-138093. doi: 10.1109/ACCESS.2019.2942152