International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

Design of Approximate Polar Maximum-Likelihood Decoder

Kadala Divyavani, Mamatha Samson, K.Swaraja, Padmavati Kora, Meenakshi.K

Abstract*: Polar codes, presented by Arikan, accomplish the ability to acquire nearly error-less communication for any given noisy channel of symmetry with "low encoding and decoding complexities" on a huge set of fundamental channels. As of late, polar code turned into the best ideal error-correcting code from the perspective of information theory because of its quality of channel achieving capacity. Though the successive cancellation decoder with approximate computing is efficient, the proposed ML-based decoder is more efficient than the former. As it is equipped with the Modified Processing Element which shows the better performance with the properties of Median Filter. The proposed ML-based decoder diminishes the area and power consumed and logic utilization. In the present paper, effective polar decoder architecture is structured and executed on FPGA utilizing Vertex 5. Here we examine the proposed unique construction that is appropriate for decoding lengthy polar codes with less equipment multifaceted nature.

Keywords* : SC Decoding, Approximate computing, IAOU, Media filter, ML decoding.

I. INTRODUCTION

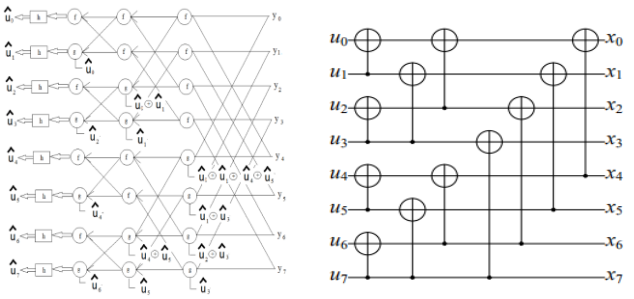
Polar codes are the principal verifiable rate of reliable communication achievable forward error correction (FEC) codes. Polar codes experience an issue with error correction performance of short and moderate codes. To enhance FEC performance of Polar codes over LDPC (Low-Density Parity Check) codes, SC(Successive Cancellation) decoding, SSC(Simplified Successive Cancellation) decoding, ML(Maximum Likelihood) decoding, etc are used. With the advancements in VLSI innovation, the integration of the number of components on a solitary chip are expanding intensely thus also expanding the feasibility of fault occurrence. Thus, this is an important research area. This includes Channel polarization, construction of code sequence for symmetric capacity which is efficient at information-theoretic problems. But higher the no.of iterations higher the complexity and latency. There are several journals discussed regarding polar decoder realization in the VLSI environment[1]. Talk about a VLSI method for area examination for a memory-less channel based decoder. This portrays any calculations that can be measured using a new Channel polarization unit. Avoids memory conflicts and graph-routing issues midst implementation but lead to higher memory requirements and latency[2]. The reevaluation for the last phase of SC decoding is done which inclines to a significant reduction in the minimum time required for the completion of an operation and the number of components required in the last phase of the SC algorithm and there is a simultaneous decoding of two bits instead of one bit which thereby giving rise to a decline in the latency with the absence of performance loss [3]. A semi-parallel consumed SC decoder with the Transpose-algorithm for SC approach and trellis modulation is proposed and the tree is collapsed into a single stage. This is extended into the frame -overlap design but at the expense of additional memory. Then, A low-power method is proposed since a decoder has soaring power dissipation in SC methods [4]. In addition to power dissipation, authors have studied on speed and delay evaluation in [5] and hardware efficiency and area utilization in. All these works of literature are focused to develop polar decoder system by using successive-cancellation technology, but the problem of this method is the architecture was then refined by collapsing the tree into a single-stage composed of N/2 PEs(Processing elements), keeping the pipeline registers intact, but increasing logic utilization significantly. This design was itself extended into a frame-overlapped variant allowing decreasing frames to be decoded serially by making use of more processing elements, thereby increasing the capacity of data transmission of the decoder at the cost of additional memory. But SC operates in series which is time-consuming and to improve the performance of error correction SSC, SC-ML, SSC list are proposed [10-12]. With high parallelism relatively low clock frequency balancing both throughput energy efficiency [20]. This paper comprises of the following. Section- II presents a concise analysis of Polar codes, polar encoding, and decoding. Section- III engraves SC (Successive Cancellation) decoding, the paper relies on. Section- IV engraves the Approximation of Successive Cancellation(SC) decoding. Section- V describes the proposed work ML-based SC decoding. Section- VI describes the implementation outcomes. Section- VII finalizes.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
Kadala Divyavani*, ECE, GRIET, Hyderabad, India.
Mamatha Samson, ECE, GRIET, Hyderabad, India.
Swaraja. K, ECE, GRIET, Hyderabad, India.
Padmavati Kora, ECE, GRIET, Hyderabad, India.
Meenakshi. K, ECE, GRIET, Hyderabad, India.

II. POLAR CODES

The POLAR algorithm was presented in 1967 as an effective strategy for decoding convolutional codes [1], extensively exploited in communication systems [2].
Utilization of this algorithm for decoding the codes is exploited in distinct applications in addition to radio relay, communication through satellite and cellular.

It has demonstrated to be an impressive result for a lot of issues associated with digital
estimation. Moreover, the POLAR decoder has constructive sbenefits in the execution of serializer-deserializers (SERDESs) with high-speed (5 to 10 Gb/s) possessing having critical latency requirements. SERDESs can be additionally utilized with 10 Gb/s in optical networks(synchronous) and the local area. Moreover, they are utilized in hard disk drive or DVD storage systems which are magnetic or optical storage systems [3]. The recent advances in polar codes are ship safety for Polar code i.e., operation of International ships in Polar Waters(Polar Codes). This safety includes device, plan, and architecture and working and crewing. It applies to provide safe ship operation working in Polar waters(north and south pole) and thereby protecting the polar environment. The other advancement is the construction of polar codes in 5G new radio. sPolar Encoding: Characterization of polar encoding is done by the generator matrix as the application of a set of linear block codes is error-correcting code (ECC). The accomplishment of the generator matrix is done by the nth power application to the kernel matrix, for the code having length N or 2n, G= $F^{\otimes N}$. Constitution of both input and frozen bits form an information block and is the mapping of the codeword block i.e., Output codeword vector (O) and Length (N) where GN is the generator matrix. Calculation of the codeword vector is done after acquiring the generator matrix by using Out = (In)*GN, where In and Out represent information fed at the input and codeword vectors at the output respectively. Channel polarization is employed by Polar codes. In the case of a channel that is combined, every channel advances to reach a trusty or unreliable channel. Now the reliability of every sub-channel is admitted. So input message bits are traversed through the high capacity channels and frozen bits are traversed through low capacity and the remains of sub-channels are exploited for the construction of polar code. The POLAR algorithm is the process similar to identifying the order of states that are most-likely, resulting in an order of observed events and thus, boasts high efficiency as it comprises of a finite number of possible states [4–6]. It is an impressive exertion of a discrete-time finite-state Markov procedure recognized in memory-less noise and optimality can be accomplished by following the maximum-likelihood criteria [7]. It helps in keeping a track of stochastic procedure state utilizing an optimum recursive strategy that aides in the analysis and execution [8, 9].



(a) (b)

Fig. 1 (a) SC decoder for N=2^3=8
(b) Polar encoder for N=2^3=8 [2]

POLAR DECODING: The codeword is to be decoded to obtain the required information which is the removal of the generator matrix. Thus the output of the decoder is obtained which is the estimated codeword. The most important and widely used decoder is the Successive Cancellation(SC) decoder which is based on the decoding tree. [2].

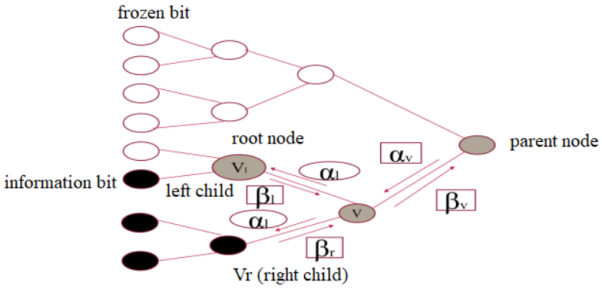


Fig. 2 Successive Cancellation Decoding tree of an (8,3) polar code

III. SUCCESSIVE CANCELLATION DECODING WITH APPROXIMATE COMPUTING

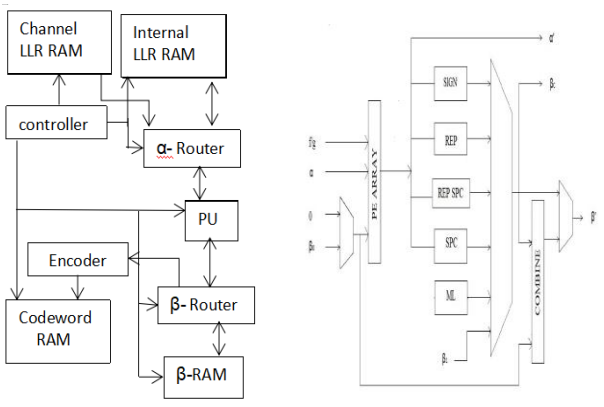
A. Primary design of the decoder

The PU and primary plan of the decoder are illustrated in Fig. 3(a)-(b). The primary structure of the decoder consists of various blocks, in which PU plays a major role. PU consists of a combine unit that carries out the operation as in Eq. (1)[8] to produce the final estimated codeword. Here N_v is the length of the code. Based on the delay there is a reduction in the minimum time required for the completion of operation through PE, SPC, and Combine, which needs a quick implementation, notably for PE and SPC. The operational blocks of the decoder are discussed in the following sections.

$$\beta_v[k] = \begin{cases} \beta_{left}[k] \oplus \beta_{right}[k], & \text{if } k < \frac{N_v}{2}; \\ \beta_{right}\left(k - \frac{N_v}{2}\right), & \text{otherwise.} \end{cases} \quad (1)$$

$$f(a,b) = \text{sign}(a)\text{sign}(b)\min(|a|,|b|) \quad (2a)$$

$$g(\hat{u}_s, a, b) = a(-1)^{\hat{u}_s} + b \quad (2b)$$



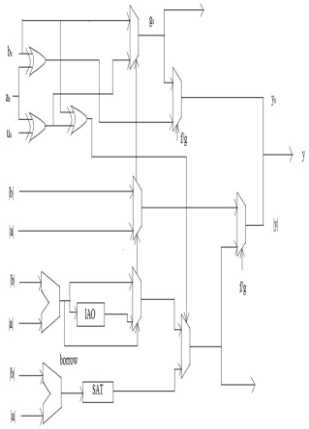
(a) (b)

Fig. 3: (a) Primary structure of the decoder, (b) Structure of the Processing Unit (PU).

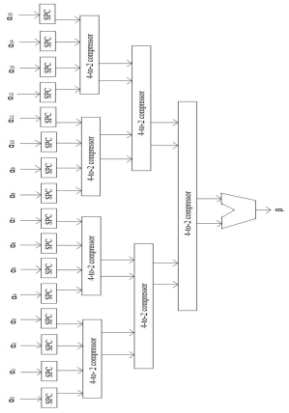
B. Structure of the PU block

1)PE: The PE array is used for the calculation of 'f' and 'g' nodes i.e., from eq 2(a) and 2(b). 'a' and 'b' are inputs with Q-bit and $\hat{u}_s$ is the previously decoded bit.

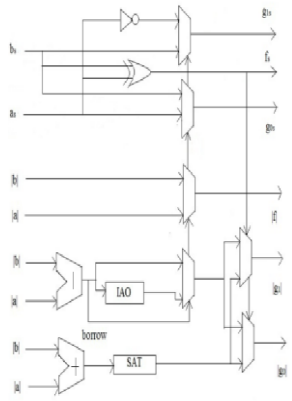
Calculate 1) $|g| = |a| + |b|$ if as $\oplus \hat{u}_s \oplus b_s = 0$; Otherwise, $|g| = ||a| - |b||$. 2) $g_s = \hat{u}_s \oplus a_s$ if $|a| \geq |b|$; Otherwise, $g_s = b_s$. The structure of the PE is presented in Fig. 4(a), here selection line 'f' or 'g' indicates the corresponding node on which the output y is the based. Here IAO is an Invert add one unit and SAT is the saturation block. Results obtained show the when compared separation of g block [8] and the merged PE [17] PE is faster. The concept of node merges is realized by extracting and feeding $|g|$ and g_s to the SPC block and Sign block [8].



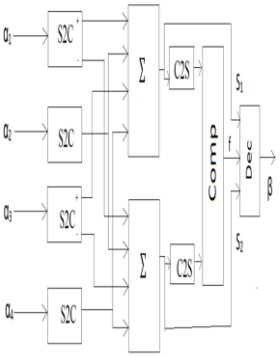
(a) Processing Element



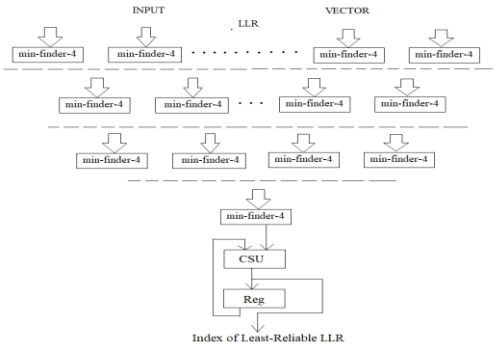
(b) Repetition



(c) Modified Processing Element



(d) Maximum Likelihood



(e) basic section of Single Parity Check

Fig. 4: Internal structures of Processing Unit block

2) REP Block: Initially there is a quantization of input into Q-bit in SM representation which is not greater than 16. The Q-bit inputs are fed to sign-to-2's complement blocks, whose outputs are in turn fed as inputs to the 3-stage 4-to-2 compressors [18]. The 4-to-2 compressor consists of two full adders. The output of the compressor is concatenated with an accumulator, as depicted in fig. 4(b). β is either all zeros or all ones depending on the sum obtained from all input LLRs

3) SPC Block: The inputs vectors are fed to the minimum finder blocks stages as depicted in fig4(e) which finds the minimum of the 4 numbers using pair-wise comparison to identify the minimal-trusty one among the 4 inputs. For P = 64, it is implemented by a 3 stage min-finder-4. For the faster implementation, there is an insertion of pipe-lined registers between all three stages. The register is utilized to deal with the situation when $N_v > P$. The use of pair-wise comparison improves the fastness of the decoder.

4) ML Block: The ML nodes have a code (0, u_1, 0, u_3) form in which there is a restriction to the code length and it is 4. Once the encoding process is done, there is a possibility of only 4 codewords, (0000), (1111), (1010), and (0101). As the codewords (0000) and (1111) have opposite reliabilities. Similarly to the codewords (1010) and (0101). By this observation, ML block with a reduction in complexity was proposed. As depicted in Fig. 4(d), α_0, α_1, α_2, α_3 are the inputs of Q-bit LLRs which are in SM representation. The 2's complement of $+\alpha_0$ and $-\alpha_0$ is generated by the S2C section which is at the top left. The reliability of the codeword (0000) is calculated by the upper accumulator and the reliability of the codeword (1010) is calculated by the lower accumulator. After calculating the reliabilities, the sign bits s_1 and s_2 are extracted. 'f' which represents the signal of a flag gives the size comparison of the magnitudes of the 2 reliabilities. Now s_1, s_2, f are fed as input to the Dec unit which is used to generate the most likely codeword based on Table 1.

Table- I: The most likely codeword according to s1, s2, and f.

S1	S2	f	The most likely codeword
0	-	0	0
1	-	0	-1111
-	0	1	-1010
-	1	1	-101

5) REP-SPC Block: There is a restriction to the code length of this node which is 8. A modified PE (MPE) is depicted in fig. 4(c), which contributes to the result of the g operation in eq. (2b) is used to calculate g_0 and g_1. g_0 is calculated assuming $\hat{u}_s$ is 0 and g_1 assuming $\hat{u}_s$ is 0. Here α is an input vector with Q-bit LLRs in Sign-Magnitude representation. α_{r0} is calculated supposing β_1 is (0000). Similarly, α_{r1} assuming β_1 is (1111). α_1, α_{r0}, and α_{r1} are calculated using four MPEs. There is a necessity of two small SPC units and an accumulator and to advance the progress α_1, α_{r0}, and α_{r1}, to keep the consistency with [8].

A. Approximate Comparator (Apx-Comp)

Given A and B are two n-bit inputs with magnitude, the smaller one between them can be identified just by verifying their binary strings beginning at the MSB till the LSB. Several comparisons are possible by the use of only some bits close to the MSB, which is taken as an inspiration and led to the structure of Apx-Comp, as depicted in Fig. 5.

In this, the m-LSBs of P and Q are disregarded. f=0 when $P^m_{n-1} \geq Q^m_{n-1}$, else f = 1. The Apx-Comp evaluates that P ≥ Q, if $P^m_{n-1} = Q^m_{n-1}$ and $P^0_{m-1} < Q^0_{m-1}$, but P < Q which results in a fault. There is an analysis of the quantified tie that binds between the error rate (ER) and the parameter k in the subsequent section to recognize the precision of the Apx-Comp. For uniform distribution, the inputs are assumed to be random numbers which also shows ease thus for i_0^{n-1} = 0, 1, 2,· · ·, n-1 $Pr[p_i=q_i]=\frac{1}{2}$. So $Pr[p^0_{m-1} < Q^0_{m-1}] = \frac{1-\left(\frac{1}{2}\right)^m}{2} = \frac{2^m-1}{2^{m+1}}$, $Pr[P^m_{n-1} = Q^m_{n-1}] = \left(\frac{1}{2}\right)^{n-m}$. [21]. The Apx-Comp's ER is calculated as shown below

$$ER = \left(\frac{1}{2}\right)^{n-m} \cdot \frac{2^m-1}{2^{m+1}} = \frac{2^m-1}{2^{n+1}}.$$

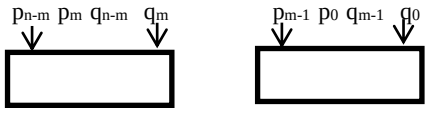


Fig: 5 Structure of the Apx-Comp

B. Apx Inversion-and-adding-one unit(Apx-IAOU)

The decoder has several data format transformations. The principal operation is the bit-wise inversion of a number and adding one to it. P an n-bit unsigned input inverted bit-wise and one is added to it and S denotes the number corresponding to it. This work has an approximate scheme introduced to acquire S(m-1, · · ·, 0). Initially, P is 1's complemented and the result is acquired and 1 is added to the m-LSBs. The carry-out bit generated after the add 1 function is discarded without proliferating to the next stage of bits. S^0_{m-1} are saturated, if $P^1_{m-1} = 0$. By taking 1's complement of P^m_{n-1}, S^m_{n-1} is directly obtained. The Apx-IAOUs truth tables are shown in Fig. 6. From Boolean algebra, the expressions of the corresponding bits can be derived.

p_0	$\overline{p_0}$	s_0
0	1	1
1	0	1

(a) m=1

$p_2p_1p_0$	$\overline{p_2}\,\overline{p_1}\,\overline{p_0}$	$s_2s_1s_0$
000	111	111
001	110	111
010	101	110
011	100	101
100	011	100
101	010	011
110	001	010
111	000	001

(b) m=2

p_1p_0	$\overline{p_1}\,\overline{p_0}$	s_1s_0
00	11	11
01	10	11
10	01	10
11	00	01

(c) m=3

FIG. 6: TRUTH TABLES OF THE APX-IAOUS

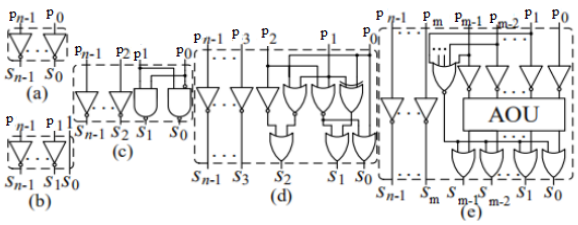


Fig. 7: Structure of the Apx-IAOUs: (a) m=0, (b) m=1, (c) m=2, (d) m=3, and (e) m>3 [22]

For m= 1, s_0= 1. For m= 2, $s_0=\overline{p_1p_0}$, $s_1=\overline{p_1p_0}$. For m= 3, $s_0=\overline{p_2}+p_1+p_0$, $\overline{s_1}=\overline{p_2}+p_1+p_1\oplus p_0$, $\overline{s_2}=\overline{p_2+p_1}+p_0$. The structures of the Apx-IAOUs for m=0, m=1, m=2, m=3, and m> 3 are depicted in Fig.7(a)-(e). A carry signal is generated when each bit of P^0_{m-1} is 0, but not propagated which leads to a fault.

C. Apx-Adder

In the Repetition block, several LLRs are assembled in which the desired part is the sign bit which is obtained from the sum, whereas the remaining section is worthless. This was taken as an inspiration that put forward the Apx-Adder to conserve the hardware resource. P and Q are n-bit inputs. First, the m-LSBs of addends P and Q are excluded and the remaining bits i.e., n-m bits are given as input to the adder. An error occurs when the sum calculated from the rest of the section provides the chain propagation of carries of addition and a carry is developed at the excluded section.

$$\text{Error Rate} = P_r[(\prod_{j=m}^{n-1} x_j)=1]\,(P_r[g_{m-1}=1] + P_r[x_{m-1}g_{m-2}=1] + \cdots + P_r[(\prod_{j=1}^{m-1} x_j)g_0=1]) = \frac{2^m-1}{2^{n+1}} \quad (4)$$

where $g_i=p_iq_i$, $x_i=p_i\oplus q_i$. Assume $P_r[g_i=1]=P_r[p_i=1,q_i=1]=P_r[p_i=1]P_r[q_i=1]=\frac{1}{4}$, $P_r[p_i=1]=P_r[p_i=1, q_i=0]+P_r[p_i=0, q_i=1]=P_r[p_i=1]P_r[q_i=0]+P_r[p_i=0]P_r[q_i=1] = \frac{1}{2}$. $g_i=1$ and $x_j=1$ (i ≠ j) are independent events. P and Q are distributed uniformly.

IV. ML DECODER WITH APPROXIMATION

In this section, we will discuss about ML-based SC Decoder. Some codes want to correct as many errors as viable, for the reason that there's due to the lack of re-transmission, there is a necessity of correcting as many errors as possible. To achieve this the standard approach is Maximum Likelihood Decoding(ML). In Maximum Likelihood Decoding(ML), the received word is compared with all the possible code words and one closest codeword is the maximum corrected codeword. ML decoder works on this property of this decoding. If an error is detected, to correct the mistake we identify the most probable error and undo this error. Occurrence of the greatest probability of the word is received using ML. ML decoder finds the shortest path. Generally, ML decoder uses Hamming distance to correct the error. Initially, the data is encoded and this encoded data that is to be decoded is received by ML. Now ML compares the received codeword with all the possible codewords and calculates the Hamming distance (using xor operation) which is the presence of the number of 1s and thus the error detection is done. Finally, this error is corrected. This paper presents a decoder that is identical to the previously discussed decoder but the only difference is the ML block. PU(Processing Unit) which is the sub-block of the top-level architecture of decoder is modified fig. 8. Initially in PU block, the input is fed to the PE array. Now the PE array block is implemented the output of PE(Processing Element) array is fed as input to the Median Filter fig. 9.

Median block is a data array signal processing block. It operates as an interface between the PE array and the remaining blocks. It consists of high-quality filters. The size of the filters can be variable depending on the size of the data. The Median filter consists of exchange nodes as shown in Fig. 10 and each node has two outputs, one low(L) and the other high(H). In the Median Block first, data sorting is done and the sorted data is sent to the exchange nodes. Each exchange node takes two bits as input. The exchange node consists of two 2x1 multiplexers and a comparator. When the input is fed to the exchange node, initially the comparator is activated and the output of the comparator is the control signal to the multiplexers. Based on this control signal the two multiplexers outputs H and L outputs. Likewise, all the exchange nodes operate which traverse the data and finally give a median value(M). The data below the Median value is considered error less while the data above it has an error. Now, this output is fed to Sign block, REP, REP-SPC and SPC blocks. The approximated outputs (Section V) of REP(Repetition) block, REP-SPC(Modified PE) block, SPC(Single Parity Check) block are fed as input to ML(Maximum Likelihood) block. The output of the ML block along with Sign block is fed as input to a multiplexer, based on the critical path gives the estimated codeword.

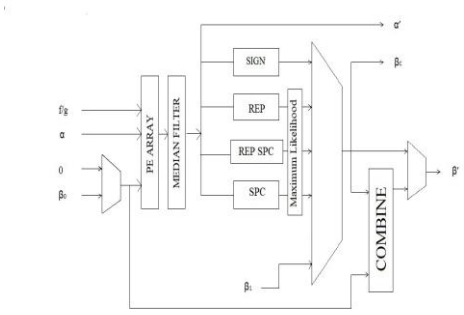


Fig. 8: Modified PU block (ML-based)

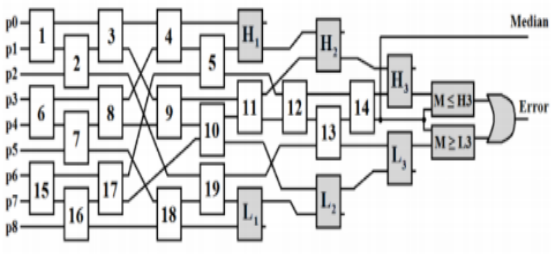


Fig. 9: Minimum exchange network scheme for nine input (Median Filter)

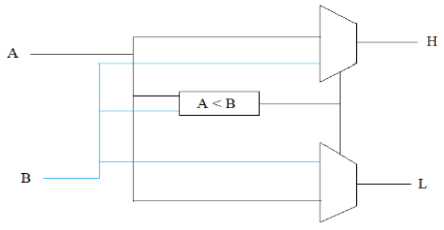


Fig. 10: Internal diagram of one exchange node [10]

V. SIMULATION RESULTS

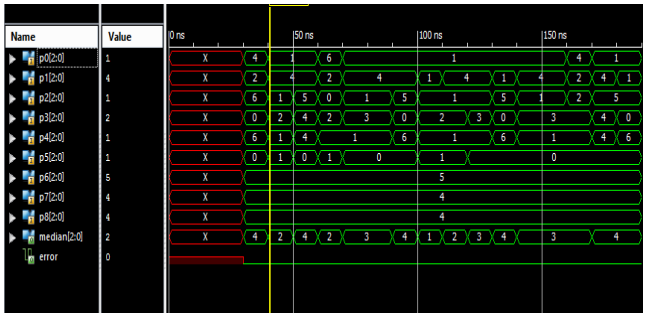


Fig. 11.1: Output of Median filter

Here p0,p1,p2,…..,p8 are the 3-bit inputs to the exchange node of the median filter. The median value is indicated by "median" which is 3-bit and is the output of the Median filter which is in turn fed as input to the SIGN, REP, REP SPC, and SPC blocks.

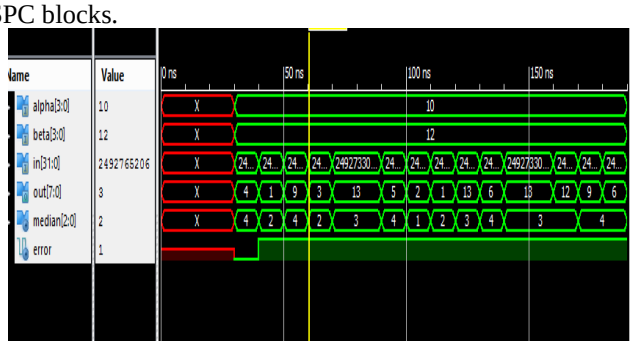


Fig. 11.2: Output of PU (Processing Unit) block

Here is the output of PU block. "alpha", "beta", "in", "median" are the inputs to the PU block. "alpha" and "beta" are 4-bit inputs. "in" is 32-bit input. "median" a 3-bit input which is the output of median block is also fed as input to PU block. Finally "out" an 8-bit output is obtained.

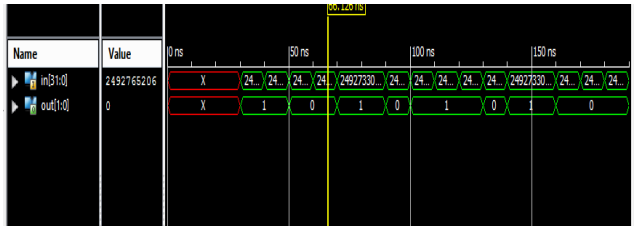


Fig. 11.3: Output of REP block (Repetition)

Fig.11.3 shows the output of the REP block(Repetition). Here "in" is a 32-bit input which utilizes 4-to-2 compressors and an accumulator to produce the final output of REP block. "out" is a 2-bit output of REP block.

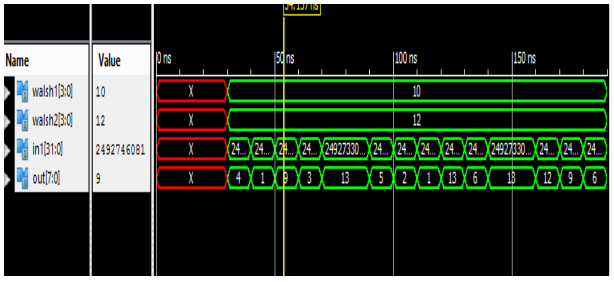


Fig 11.4: Output of REP-SPC block

Fig. 11.4 shows the output of REP-SPC block. “walsh1”, “walsh2”, “in” are the inputs to REP-SPC block. “walsh1”, “walsh2” are 4-bit inputs and “in” is 32-bit input. “out” is an 8-bit which is the output of REP-SPC block.

Fig. 11.5 shows the output of ML(Maximum Likelihood) block. “in” is an 8-bit input. “pos0”, “pos2”, “neg0”, “neg1”, “neg3” are the outputs of intermediate blocks which are 2-bit each. “add1”, “add2” are the outputs of adder block which are 4-bit each. “m1”, “m2” is 4-bit which decides the most likely codeword. “out” is 4-bit and is the final most likely codeword which is the output of ML block.

Fig. 11.5: Output of ML(Maximum Likelihood) block

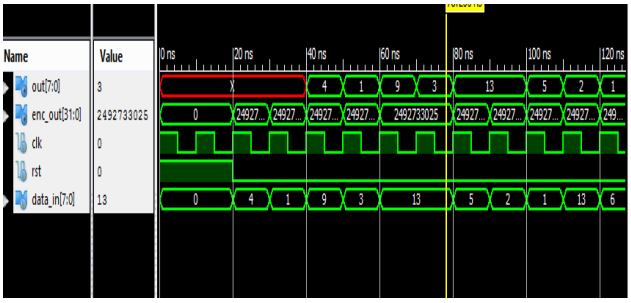


Fig. 11.6: Simulation output

Fig. 11.6 represents the simulation output of the approximate ML-based decoder. Here ‘data_in’ is 8-bit data which is to be encoded to get ‘enc_out’. ‘enc_out’ is the encoded operand which is 32-bit. ‘Out’ is the decoded error-free output data which is 8 bit. Now ‘data_in’ which is 8-bit is the same as ‘out’ which is 8-bit delayed by 1 clock cycle after reset initialized to zero.

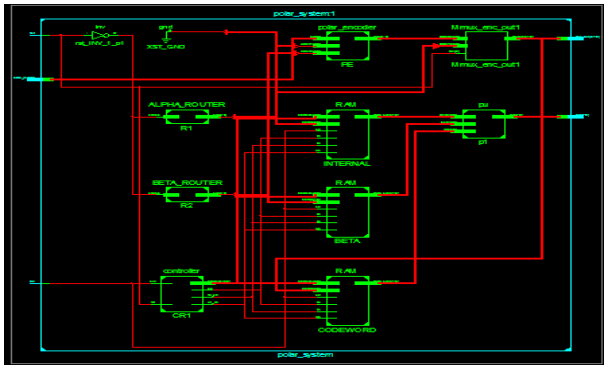


Fig. 11.7 RTL Schematic

The above figure shows the RTL layout, which has all sub-blocks and their interconnections.

TABLE- II Design Summary

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	91	16640	0%
Number of Slice Flip Flops	112	33280	0%
Number of 4 input LUTs	150	33280	0%
Number of bonded IOBs	69	309	22%
Number of GCLKs	1	24	4%

TABLE- III COMPARISON OF THE THREE METHODS

Parameter	SC decoder without Approximation	SC decoder with Approximation	ML decoder with Approximation
Time delay	4.115 ns	3.9771 ns	1.018 ns
Power utilized	1.065 µw	0.143 µw	0.065 µw
Slice registers	63	31	45
LUTs	263	214	84
LUT Flip Flop	57	30	26
IOBs	450	50	43

VI. CONCLUSION

In this paper, several implementations of CLA, SC, approximate computing are utilized to develop a new approximate polar ML decoder based architecture. The application of various algorithms lead to the design of this approximate polar ML decoder based architecture. The proposed ML-based architecture uses almost one-third of the number of device utilization ratio, compared to the results obtained using Xilinx Virtex-5. There is an appreciable reduction in delay when compared to the existing methods. Here ML is used to increase the accuracy and reduce the delay. Apart from these, the proposed architecture exploits a much less area utilization ratio. A wide variety of applications depends on this criterion.

REFERENCES

1. E. Arıkan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memory-less channels,” IEEE Transactions on Information Theory, volume 55, no. 7, pp. 3051–3073, July. 2009.
2. C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, “A semi-parallel successive-cancellation decoder for polar codes,” IEEE Transactions on Signal Process., volume 61, no. 2, pp. 289–299, January 2013.
3. B. Yuan and K. K. Parhi, “Low-latency successive-cancellation polar decoder architectures using 2-bit decoding,” IEEE Transactions on Circuits and Systems I, Regular Papers, vol. 61, no. 4, pp. 1241–1254, April 2014.
4. A. J. Raymond and W. J. Gross, “A scalable successive-cancellation decoder for polar codes,” IEEE Transactions on Signal Process., volume 62, no. 20, pp. 5339–5347, October 2014.
5. Y. S. Park, Y. Tao, S. Sun, and Z. Zhang, “A 4.68Gb/s belief propagation polar decoder with bit-splitting register file,” in Proc.

Symposium on VLSI Circuits Digest of Technical Papers, June. 2014, pp. 1–2.
6. A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive cancellation decoder for polar codes," IEEE Communications Letters., volume. 15, no. 12, pp. 1378–1380, December 2011.
7. G. Sarkis and W. J. Gross, "Increasing the throughput of polar decoders," IEEE Communications. Letters., volume. 17, no. 9, pp. 725–728, April. 2013.
8. G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: algorithm and implementation," IEEE J. Selected Areas in Communications., volume. 32, no. 5, pp. 946–957, May 2014.
9. I. Tal and A. Vardy, "List decoding of polar codes," IEEE Transactions on Information Theory, volume. 61, no. 5, pp. 2213–2226, March 2015.
10. J. Lin and Z. Yan, "Efficient list decoder architecture for polar codes," in Proc. IEEE International Symposium on Circuits and Systems (ISCAS), June 2014, pp. 1022–1025.
11. ——, "An efficient list decoder architecture for polar codes," IEEE Transactions Very Large Scale Integration (VLSI) Systems., volume. 23, no. 11, pp. 2508–2518, November 2015.
12. J. Lin, C. Xiong, and Z. Yan, "A reduced latency list decoding algorithm for polar codes," in Proc. IEEE Workshop on Signal Processing Systems (SiPS), October 2014, pp. 56–61.
13. J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in Proc. IEEE European Test Symposium (ETS), May 2013, pp. 1–6.
14. O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "RAP-CLA: a re-configurable approximate carry look-ahead adder," IEEE Transactions on Circuits and Systems II, Express Briefs, volume. PP, no. 99, pp. 1–1, November. 2017.
15. B. Yuan and K. K. Parhi, "Successive cancellation decoding of polar codes using stochastic computing," in Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS), May 2015, pp. 3040–3043.
16. Y. Fan, J. Chen, C. Xia, C.-Y. Tsui, J. Jin, H. Shen, and B. Li, "Low latency list decoding of polar codes with double thresholding," in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), April 2015, pp. 1042–1046.
17. A. J. Raymond, "Design and hardware implementation of decoder architecture for polar codes," Master's thesis, McGill University, 2013,[Online].Available:http://digitool.Library.McGill.CA:80/R/-?func=dbin-jump-full&object id=121185&silo library=GEN01.
18. C.-H. Chang, J. Gu, and M. Zhang, "Ultra-low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," IEEE Transactions on Circuits and Systems I, Regular Papers, volume 51, no. 10, pp. 1985–1997, October 2004.
19. P. Giard, G. Sarkis, C. Thibeault, and W. J. Gross, "A 638 Mbps low complexity rate 1/2 decoder on FPGAs," in Proc. IEEE Workshop on Signal Processing Systems (SiPS), October 2015, pp. 1–5.
20. O. Dizdar and E. Arıkan, "A high-throughput energy-efficient implementation of successive cancellation decoder for polar codes using combinational logic," IEEE Transactions on Circuits and Systems I, Regular Papers, vol. 63, no. 3, pp. 436–447, March 2016.
21. Successive cancellation(SC) decoding for general(N, K) polar codes by Professor Andrew Thangaraj Department of Electrical Engineering IIT Madras.
22. A practical introduction to Polar codes by Harish Vangala, Yi Hong, Emanuele Viterbo Monash University, Australia.

AUTHORS PROFILE

Kadala Divyavani, Completed B.Tech In Eee From Jntu Hyderabad In The Year 2016 And Pursuing M.Tech In Vlsi At Griet Hyderabad. Her Domains Of Interest Are Exploring The Vlsi Design, Fault Tolerance In Vlsi Circuits.

Dr.Mamatha Samson, Pursued A Bachelor Of Engineering In Electronics And Communication Engineering From Mysore University In The Year 1991. She Pursued Ms From Bits, Pilani In The Year 1995 And In 2013 Pursued Her Ph.D. From Iiith. At Griet, She Is Currently Working As A Professor. She Has Published Various Research Papers In International Journals And Conferences. Vlsi Design, Mixed-Signal Circuits And, Biomedical Circuits Are Her Domains Of Interest. She Is A Member Of Iste, Institution Of Engineers India And Ieee. She Received A Project Fund From Dst Govt. Of India In The Year 2009.

K. Swaraja, Pursued A Bachelor Of Technology In Electronics And Communication Engineering From Jntu, Anantapur, Ap, India. In Digital Electronics & Communication Systems. She Pursued Her Master Of Technology From The Prestigious Jntu, Anantapur, A.P, India. In The Field Of Digital Image Processing, She Pursued Her Ph.D. From Jntuh, A.P, India. At Griet, She Is Currently Working As A Professor, In Bachupally, Hyderabad, Telangana, India. She Has Around 30 Articles Into Her Credit In National & International Journals And Conferences. Digital Signal, Image And Video Processing Besides Vlsi, Machine Learning And Communication Systems Are Her Domains Of Interest.

Dr. Padmavathi Kora, Has Pursued Her Ph.D. In 2016 And Has A Teaching Experience Of 15 Years. At Griet, She Is Presently Employed As A Professor. In International Journals/Conferences, She Has More Than 36 Articles Into Her Credit On Ecg And Eeg Signal Processing, Image Processing, Machine Learning Are Her Domains Of Interest.

.

Dr. Meenakshi K, Pursued Her Ph.D. In 2018 And Has A Teaching Experience Of 15 Years. At Griet, She Is Currently A Professor. In International Journals/Conferences, She Has Nearly 15 Articles Into Her Credit In Image And Video Watermarking. Machine Learning And Network Security Are Her Other Domains Of Interest.