Low Latency SC Decoder Architecture for Interleaved Polar Codes

Nandini JALI, Pullakandam MURALIDHAR, Sreehari Rao PATRI

Dept. of ECE, National Institute of Technology, Warangal - 506004, Telangana, India

jalinandini@gmail.com, pmurali@nitw.ac.in, patri@nitw.ac.in

Submitted October 7, 2021 / Accepted May 10, 2022 / Online first August 16, 2022

Abstract. *Interleaved polar (I-Polar) codes, a new facet of polar codes to achieve better channel capacity, is designed by placing the interleaver and deinterleaver blocks midway between the stages of the polar codes. Low latency hardware optimization makes their implementation even more suitable for ultra-reliable low latency applications. This study proposes an optimal hardware design for low latency interleaved polar codes by reframing the last stage of the interleaved successive cancellation decoder. A high-speed adder-subtractor is used to reduce the latency further, thus increasing the speed of operation. Interleaving data in the proposed polar codes augment BER performance compared to conventional (n, k) polar codes. The proposed I-Polar codes are synthesized using Synopsys design compiler (SDC) in CMOS 65-nm technology. Results show that the latency is reduced by 50.5% on average compared to the conventional polar codes as high-speed adder and merged processing elements are used. Moreover, the average gate count and power are reduced by 14% and 40.56%, respectively.*

Keywords

BER, deinterleaver, interleaver, I-Polar, latency, ultra-reliable low latency applications

1. Introduction

High-speed communication with good performance is the primary concern in wireless communication systems usage framework consisting of enhanced mobile broadband communications (eMBC), massive machine type communication (MMTC) and ultra-reliable low latency communications (URLLC). The invention of polar codes is the breakthrough in forward error correction techniques of wireless communication systems. The low-complexity forward error correction codes proposed in [1] are the first to acheive the channel capacity of symmetric binary input memoryless channels. It is well known that polar codes are based on channel polarization for dividing the channel into reliable and unreliable channels by combining and splitting the channels. Unreliable channels are filled with frozen bits '0', whereas reliable channels are filled with information bits.

Successive cancellation (SC) decoding is the standard decoding algorithm for the polar codes derived in [1] with low complex hardware architecture and $O(n \log n)$ operations. It suffers from significant decoding latency and high hardware costs for long code lengths due to its sequential nature.The combinational logic-based SC polar decoder proposed in [2], which operates at low clock frequencies, has a trade-off between the throughput and energy-efficient systems for moderate code lengths. For long code lengths, a hybrid circuit with a combinational circuit and sequential SC was used. To alleviate the delay accrued in standard SC decoding methods, a partial sum update logic was proposed in [3].

An efficient semi-parallel SC decoder using processing elements, which has high-performance network architecture with low critical path delay, was proposed in [4] with XOR and AND gates. One's complement-based merged processing element implemented in [5] removes the adder to convert 1's complement to 2's complement. As a result, the hardware complexity decreases and throughput increases. A low complex SC decoder was presented in [6]. The frozen bit calculation is pruned since the sequence of frozen bits is known at transmitter and receiver. In addition, the SC decoder's sequential structure is used to calculate information bits.

The novel LLR representation scheme for SC polar decoder was presented in [7] by representing in a redundant manner leading to low complexity and high-speed circuitry. Its throughput efficiency is higher than previous decoders. Throughput can be further increased to attain higher speed. A high-speed low complex algorithm for PE using 2's complement representation of logarithmic like hood ratios was implemented in [8]. It reduces the critical path delay and simplifies computation complexity. The throughput and frequency are better for the presented algorithm using 2's complement representation of LLRs. 2's complement representation increases the hardware complexity. A novel parallel polar encoding and decoding is proposed using the cyclic redundancy check for the high throughput implementations was proposed in [9]. It further reduces the latency and throughput compared to the conventional SC polar decoder. Addition of CRC may further increase the complexity.

DOI: 10.13164/re.2022.0398

Successive cancellation list (SCL) decoding, an improved version of SC decoding, maintains 'L' candidates and path metrics to increase the reliability with $O(Ln \log n)$ operations. The final codeword is chosen from the list of codewords using path metrics. Cyclic redundancy checks and parity aided SCL decoding techniques are used to decode the correct codeword. As the list size increases, the delay and complexity of decoding increase, thereby restricting the use of SCL for low latency and high throughput applications. Low latency successive cancellation list decoding was proposed in [10] through selective expansion and double thresholding at the system and algorithmic level, respectively. By reformulating the decoding algorithm, multibit-decision was implemented in [11] to reduce the list successive cancellation decoder's latency. Hardware implementation of $2b$-rSCL and $4b$-rSCL reformulated decoders achieved a notable improvement in hardware efficiency and throughput. SCL decoder with split tree architecture was proposed in [12] by dividing the decoding tree into subtrees that are decoded parallelly by the subtree algorithm. Later, reconciliation of subtrees after each decoding step improves the latency and throughput proportional to split tree. Successive cancellation stack (SCS) decoder is proposed to improve the performance of the successive cancellation decoder. It suffers from high time complexity at low SNR. Segmented cyclic redundancy check aided-SCS (SCA-SCS) and adaptive SCS (ASCS) decoders were presented in [13] to reduce the time and space complexity at low and high SNR. These decoders performs better than the traditional SC decoder.

A parallel decoding algorithm for linear block codes, generally known as the belief propagation (BP) technique, uses messages passing through the nodes of the factor graph. Consequently, high throughput and performance can be achieved depending on the number of iterations required to get the final information bits. The belief propagation decoder is faster than the SC because of parallel operations. However, as the number of iterations increases, the latency and energy dissipation increase proportionally in the belief propagation decoder. The subfactor graph freezing approach lowers average calculations, iteration count, latency, and energy consumption. Different scheduling schemes like halfway, quarter, and round-trip scheduling were used in early stopping criterion-based belief propagation decoder [14].

Interleaved polar (I-Polar) codes proposed in [15] are the new class of polar codes developed by inserting the interleavers between the intermediate stage of the polar codes, thus giving better performance than standard polar codes for different code lengths. During transmission, bit errors group together due to synchronization and multipath fading of the channel. The group errors that appeared on particular bits are distributed through the block length using the interleaver, which reorders the bit sequence to reduce the error. Deinterleavers are used at the receiver to get back the sequence of bits transmitted. By using the best interleaver, good performance can be attained with a low bit error rate. Concatenated codes with different interleaving schemes like random interleaving and blind interleaving were presented in [16]. Bit error rate performance of concatenated polar codes with blind interleaving is better than with random interleaving. Concatenated polar codes with SCL decoding are better than SC decoding, but the complexity increases when the list size increases.

Till now, most of the researchers have designed low complexity successive cancellation polar decoders. However, the polar codes still suffer from the reliability issues. To increase the reliability of the polar decoders, I-Polar codes were designed in [15]. The authors have proposed processing element based I-Polar codes to increase the throughput. The main contributions are as follows:

- The last stage of the I-Polar decoder is reformulated to decode two bits simultaneously using the processing element.
- A high-speed adder-subtractor block is used in the G function in all the stages to increase the throughput and reduce the delay concurred by the ripple carry adder in G function.

This paper focuses on the hardware architecture of low latency interleaved polar codes to attain good performance and high throughput. The rest of the paper is organized as follows: Section 2 demonstrates the basics of polar codes and I-Polar codes. Section 3 discusses low latency interleaved polar codes. Section 4 provides the simulation results, synthesis results of the proposed I-Polar decoder obtained using Synopsys design compiler (SDC) in CMOS 65-nm technology and comparison with the conventional polar codes. Section 5 concludes the contents discussed in this article.

2. Previous Work

2.1 Polar Codes

Polar codes are the linear block codes (LBC) represented by (n, k), where n (i.e., $n = 2^N, N = 1, 2, 3, 4, \dots$) denotes code length for any positive integer N and k denotes the number of message bits. The ratio of k to n represents the rate, and $(n - k)$ bits are assigned as frozen bits $0 \in I^C$. The encoding uses the general Kronecker matrix '**G**' given in (1).

$$\mathbf{G} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}. \tag{1}$$

The Kronecker matrix can be extended to any N. The codeword to be transmitted in (2) is generated by the input U_n (i.e., $U_n = I + I^C$) and the Kronecker product.

$$T_N = U_N * \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \otimes n. \tag{2}$$

The architecture of polar encoder is shown in Fig. 1. The white circles and black dots at input represent the frozen (unreliable) bits and message (reliable) bits, respectively. Information bits are assigned to the reliable channels, while the frozen bits are assigned to the unreliable channels. Encoded bits T_n are deduced through the recursive XOR operation of the input bits (U_n). They are modulated and transmitted through the channel. The considered physical paramenters of the transmission setup and channel are listed in Tab. 1. The noise that exists in the channel gets added to the encoded bits during transmission. The receiver demodulates the received signal. After that, the SC decoder is used to process the received signals. The probability values obtained at the receiver are to be decoded by the decoder. In practical applications, soft information values are computed based on the likelihood ratio (LR) instead of probability values. To reduce the computational complexity and potential overflow, the likelihood ratios are further transformed to the logarithmic domain and these logarithmic likelihood ratios (LLRs) are processed in the decoder to get the final output vector. The LLR calculation was illustrated in [15] and is represented as given below in (3)–(5).

$$LR(U_1) = \frac{\mathrm{P}(U_1 = 0)}{\mathrm{P}(U_1 = 1)} = \frac{1 + LR(T_1)LR(T_2)}{LR(T_1) + LR(T_2)}, \tag{3}$$

$$LR(T_1) = \frac{\mathrm{P}(T_1 = 0)}{\mathrm{P}(T_1 = 1)}, \quad LR(T_2) = \frac{\mathrm{P}(T_2 = 0)}{\mathrm{P}(T_2 = 1)}, \tag{4}$$

$$LLR(U_1) = 2\tanh^{-1}(\tanh(LR(T_1)/2)\tanh(LR(T_2)/2)). \tag{5}$$

Codes	Linear block codes ((n, k) polar codes)
Code rate (k/n)	1/2
Code length (n, k)	(1024, 512)
Channel	Binary input memoryless channels
SNR (E_b/N_0)	1 dB to 3 dB

Tab. 1. Physical parameters.

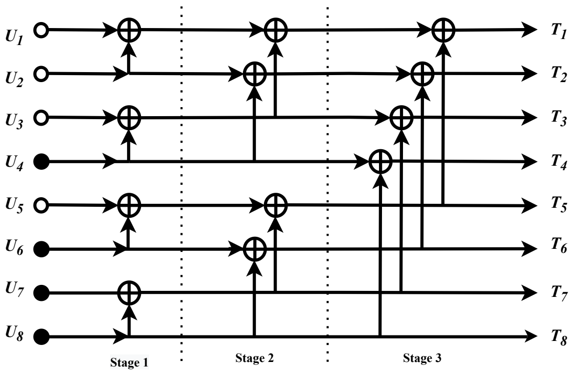

Fig. 1. Polar encoder (8,4).

The polar codes are sequentially decoded by an SC decoder that operates in m (i.e., $m = \log_2 n$) stages and estimates one bit per cycle. It can be seen from Fig. 2 that the SC decoder mainly consists of F and G functions, process the logarithmic likelihood ratios received from the channel $(T_1, T_2, \ldots, T_n)$. The LLR output of the F function is the product of both the signs of input LLRs concatenated with the minimum magnitude of LLRs. Based on the previously decoded bits, the operation of the G function is either addition or subtraction. The functional representation of F and G functions are shown in Fig. 3 and Fig. 4, respectively. F function comprises a comparator and an XOR gate, as depicted in Fig. 3. G function consists of signed magnitude to 2's complement converter (S2C), followed by an adder, subtractor, 2's complement to signed magnitude converter (C2S) and a multiplexer, as illustrated in Fig. 4. The control signal given to the multiplexer in Fig. 4 is the partial sum generated from the previously decoded bits. F and G units in each stage of the SC decoder are computed by (6) and (7), respectively.

$$\begin{aligned} F(T_1, T_2) = \mathrm{sign}(LLR(T_1))\mathrm{sign}(LLR(T_2)) \\ \min|LLR(T_1)|, |LLR(T_2)|, \end{aligned} \tag{6}$$

$$G(T_1, T_2) = LLR(T_1)(-1)^U + LLR(T_2). \tag{7}$$

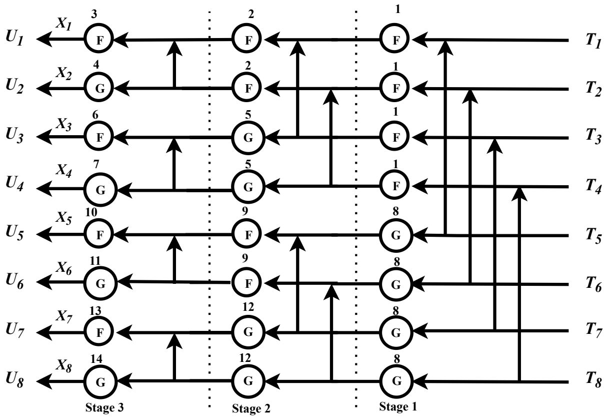

Fig. 2. (8,4) successive cancellation decoder.

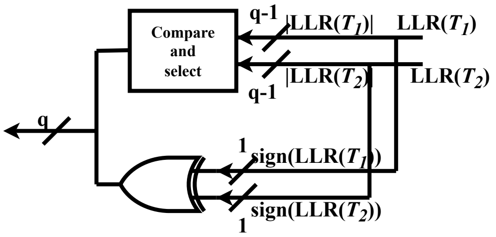

Fig. 3. F function.

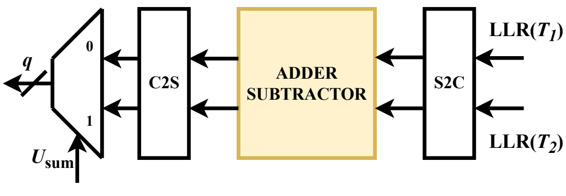

Fig. 4. G function.

Channel values are processed using F and G functions that generate LLR values till $(m-1)$ stages. At the final stage, after all F and G computations, each bit is decoded based on X_i. If the decoded bit is frozen, it is assigned with 0, irrespective of the value of X_i. If the decoded bit is an information bit and $X_i \geq 0$, then the information bit is assigned with '0'. In case $X_i < 0$, the information bit is assigned with '1', as given in (8).

$$U_i = \begin{cases} 1, & \text{if } X_i<0 \\ 0, & \text{if } X_i \geq 0, \text{frozen}. \end{cases} \tag{8}$$

The decoded bits propagate back to the G function. Depending on the XOR operation of previously decoded bits, the LLRs to the G function are either added or subtracted. If the XOR operation of decoded bits yields zero, the LLR values are added. If it yields one, the LLR values are subtracted. This process continues till all the bits are decoded. For example, in the case of an (8,4) successive cancellation decoder, the decimal digits on the top of the F function and the G function in Fig. 2 represent the sequence of operations.

2.2 I-Polar Codes

Interleaved polar codes (I-Polar) are a new version of polar codes designed by placing the interleavers between the intermediate stages of the polar codes. Interleaver is used in conjunction with error-correcting code to reduce the errors. It redistributes symbols based on a mapping technique. Inverse mapping is used by a corresponding deinterleaver to reconstruct the original symbol sequence. In a communication system, interleaving and deinterleaving can aid in mitigating the burst errors. It disperses a sequence of bits in a bitstream to minimize these burst errors during transmission. It improves the performance of FEC codes by arranging data in a non-contiguous way.

Convolution interleavers and block interleavers are two types of interleaver techniques used with FEC codes. Matrix interleaver, which is a block interleaver technique, is employed in this paper to reduce the errors. It takes row-wise input bit sequence and gives column-wise output bit sequence The advantage of using the interleavers is to reduce the burst errors inserted in the channel. The disadvantage of interleavers is the delay produced by the mapping process of the interleavers. Though there is a slight increase in the delay due to inclusion of interleavers, there is a considerable improvement in the overall performance of I-Polar codes.

The number of interleaver stages in the I-Polar codes is $(m-1)$, where $m = \log_2 n$. They are arranged with a size of 2^N, where $N(N = 1, 2, 3, \ldots, m-1)$ represents stage. As the number of stages increases, the size of the interleaver increases. At each stage, 2^N bits are interleaved and non-interleaved systematically, as shown in Fig. 5.

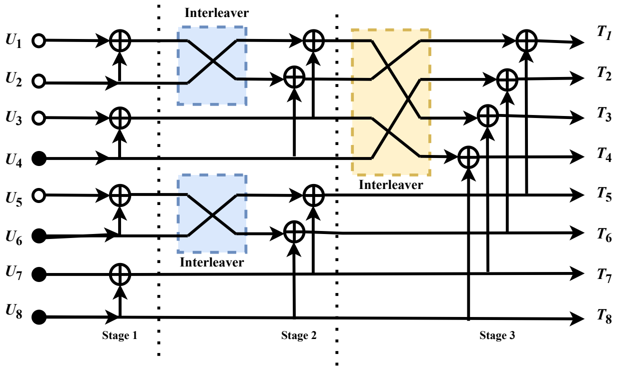

Fig. 5. (8,4) I-Polar encoder.

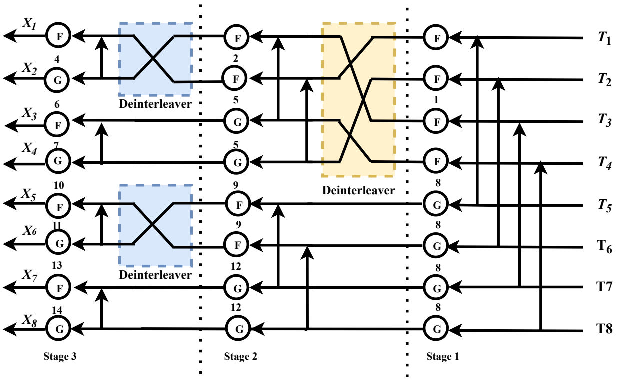

Fig. 6. (8,4) I-Polar decoder.

I-Polar codes are decoded by successive cancellation decoding with deinterleavers placed after the F function in each stage. Deinterleavers, whose operation is exactly reverse to the interleavers, generate the original sequence of bits. The size of deinterleavers placed in the decoder is the decreasing powers of 2^N, where $N = (m-1), \ldots, 3, 2, 1$, as illustrated in Fig. 6.

3. I-Polar Codes Using Processing Element

I-Polar codes decoded by the SC decoder suffer from high latency due to sequential F and G operations. An approach is proposed in the paper to reduce latency and increase throughput. By replacing the adder-subtractor block of G function in $m-1$ stages with the carry look-ahead adder-subtractor block. The output bits are decoded by merging the F and G functions of SC decoders that are present in the last stage. This merging of F and G functions in the last stage is the reformulation of a processing element to decode 2-bits in a single clock cycle. The proposed I-Polar decoder using the processing element is shown in Fig. 7. This approach reduces the number of clock cycles required to decode all the output bits after reformulation of the processing element to $0.75n - 1$ clock cycles. Carry look-ahead block increases the speed of operation by reducing the delay occurred due to the carry and borrow propagation in a ripple carry adder-subtractor. Consequently, it reduces the latency in decoding the output bits.

Inputs					Outputs	
F_1	F_2	S_1(sign(T_1))	S_2(sign(T_2))	Comp.(C)	$U_{(2i-1)}$	$U_{(2i)}$
1	1	X	X	X	0	0
0	0	0	0	X	0	0
		0	1		1	1
		1	0		1	0
		1	1		0	1
0	1	0	0	X	0	0
		0	1		1	
		1	0		1	
		1	1		0	
1	0	0	0	X	0	0
		1	1			1
		0	1	0	0	1
		0	1	1		0
		1	0	0		0
		1	0	1		1

Tab. 2. Truth table for processing element.

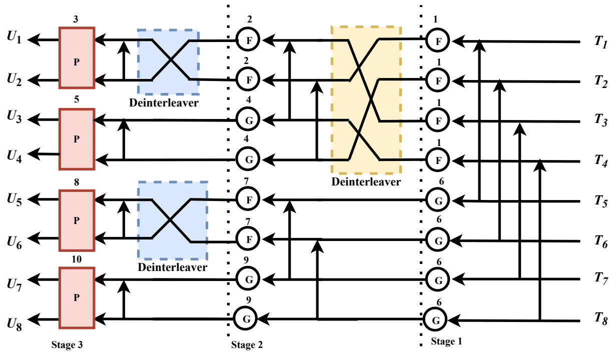

Fig. 7. Proposed (8,4) I-Polar codes using processing element.

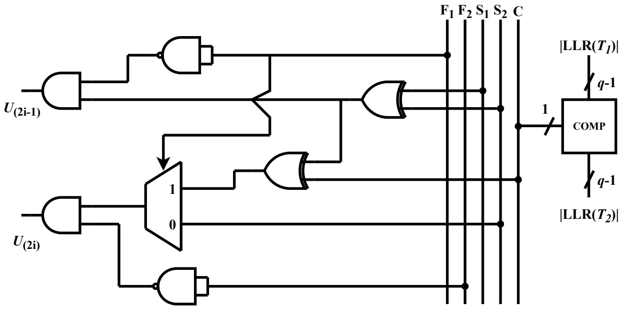

Fig. 8. Processing element.

The decoding procedure and placement of deinterleavers are the same as the I-Polar decoder until the final stage. The size of deinterleavers for an (n, k) code length varies from 2^N to the size of 2-bit deinterleavers. After 2-bit deinterleavers, the processing element removes the F and G functions in the final stage.

The processing element (PE) inputs are frozen input 1(F_1), frozen input 2(F_2) and LLR values from the previous stage. The hardware of the processing element consists of a comparator, two XOR gates, two NAND gates, two AND gates and one multiplexer whose control input is frozen input 1(F_1). Comparator generates high output if |LLR(T_1)|≥|LLR(T_2)|. The gate-level representation of the PE is shown in Fig. 8. The critical path delay of the processing element is expressed in (9), where T_{Comp}, T_{XOR}, T_{Mux} and T_{AND} are the delay accumulated by comparator, XOR gate, multiplexer and AND gate respectively.

$$T_{\text{Critical path\_PE}} = T_{\text{Comp}} + T_{\text{XOR}} + T_{\text{Mux}} + T_{\text{AND}}. \qquad (9)$$

The logic representing the operation of the processing element is given in (10) and (11).

$$U_{(2i-1)} = \begin{cases} S_1 \oplus S_2, & \text{if } F_1 = 0 \\ 0, & \text{if } F_1 = 1, \end{cases} \qquad (10)$$

$$U_{(2i)} = \begin{cases} S_2, & \text{if } F_1 = 0 \quad \text{and} \quad F_2 = 0 \\ 0, & \text{if } F_2 = 1 \\ C \oplus S_1 \oplus S_2, & \text{if } F_1 = 1 \quad \text{and} \quad F_2 = 0. \end{cases} \qquad (11)$$

The truth table for the possessing element is shown in Tab. 2. Depending on F_1 and F_2 states, it can be decided that whether $U_{(2i-1)}$ and $U_{(2i)}$ are information bits or frozen bits. $U_{(2i-1)}$ is considered as information bit if F_1 is low. In contrast, it is considered as a frozen bit if F_1 is high. $U_{(2i)}$ is considered as an information bit if F_2 is low. In comparison, it is considered as a frozen bit if F_2 high. If both F_1 and F_2 inputs are low, $U_{(2i-1)}$ and $U_{(2i)}$ are considered information bits, thereby $U_{(2i-1)}$ depends on sign(T_1) and sign(T_2), $U_{(2i)}$ depends on sign(T_2). If $F_1$is low and F_2 is high, output ($U_{(2i-1)}$) depends on the sign(T_1) and sign(T_2), whereas $U_{(2i)}$ is zero. If F_1 is high and F_2 is low, output ($U_{(2i)}$) depend on the comparator output C in Fig. 8, whereas $U_{(2i-1)}$ is zero. If both input F_1 and F_2 are high, outputs (i.e., $U_{(2i-1)}$ and $U_{(2i)}$) are zero irrespective of the sign and comparator values. The mathematical expressions of $U_{(2i-1)}$ and $U_{(2i)}$, which are formed from the truth table, are given in (12) and (13), respectively.

$$U_{2i-1} = \overline{F_1}\left(S_1 \oplus S_2\right), \qquad (12)$$

$$U_{2i} = \overline{F_2}S_1\left(\overline{F_1} + \overline{C}\right) + F_1\overline{F_2}S_1C. \qquad (13)$$

The algorithmic representation of the interleaved polar decoder using a processing element is shown below.

Algorithm 1. Decoding algorithm for I-Polar codes using processing element.

```
Input: LLR($T_1$), LLR($T_2$), LLR($T_3$), ..., LLR($T_N$).
Output: $U_1, U_2, U_3, \dots, U_N$.
    $F_1$, $F_2$ are frozen bits or information bits, $S_1$ & $S_2$ are sign bits of LLR($T_1$) & LLR($T_2$).
    Computation of $F$ functions → $F(T_1,T_2) = S_1S_2\min(|\text{LLR}(T_1)|, |\text{LLR}(T_2)|)$.
    $2^N (N = (m-1), \dots, 2, 1$ respectively at each stage) deinterleaving till final stage.
    Final stage: computation by processing element
Case 1: $F_1 = 0, F_2 = 0$
→ $U_{2i-1} = S_1 \oplus S_2, U_{2i} = S_2$
Case 2: $F_1 = 0, F_2 = 1$
→ $U_{2i-1} = S_1 \oplus S_2, U_{2i} = 0$
Case 3: $F_1 = 1, F_2 = 0$
if $S_1$ = +Ve & $S_2$ = +Ve then
    $U_{2i-1} = 0, U_{2i} = 0$
else if $S_1$ = +Ve & $S_2$ = −Ve then
    $U_{2i-1} = 0, U_{2i} = S_2 \oplus C$
    where $C = 1$ if $|\text{LLR}(T_1)| > |\text{LLR}(T_2)|$
else if $S_1$ = −Ve & $S_2$ = +Ve then
    $U_{2i-1} = 0, U_{2i} = S_2 \oplus C$
else if $S_1$ = −Ve & $S_2$ = −Ve then
    $U_{2i-1} = 0, U_{2i} = 1$
end if
Case 4: $F_1 = 1, F_2 = 1$
→ $U_{2i-1} = 0, U_{2i} = 0$
Computation of $G$ functions → $G(T_1,T_2) = LLR(T_1)(-1)^U + LLR(T_2)$.
Computation of $F$ and $G$ functions till last stage and final stage is computed using processing element.
Final output vector U ($U_1, U_2, U_3, \dots, U_N$).
```

3.1 I-Polar Decoder Architecture

The precomputation SC decoder architecture of I-Polar codes reduces the latency of the conventional SC algorithm.It consists of merged F and G, processing element and partial sum generator (PSG) logic. The partial sum from the previously decoded bits is generated by the PSG block.

The merged F and G depicted in Fig. 9 includes the F function with deinterleaver and the G function with carry look-ahead adder-subtractor. Ripple carry adder-subtractor in $(m-1)$ stages is replaced with the high-speed carry look-ahead adder-subtractor. Thus, the speed of operation increases, and latency decreases. As a result, throughput increases. The tree-based precomputation architecture of proposed I-Polar codes is shown in Fig. 10. Look ahead reformulation with processing elements, partial sum block and outputs from previous stages, shown in Fig. 11, is designed to decode the output bits quickly by decreasing the latency.

The decoding clock cycles of (8,4) I-Polar codes with clock, different stages of operation (i.e., stage 1, stage 2 and stage 3) and output is depicted in Tab. 3.

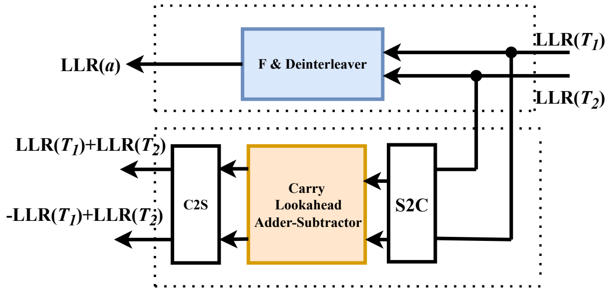

Fig. 9. Merged F and G.

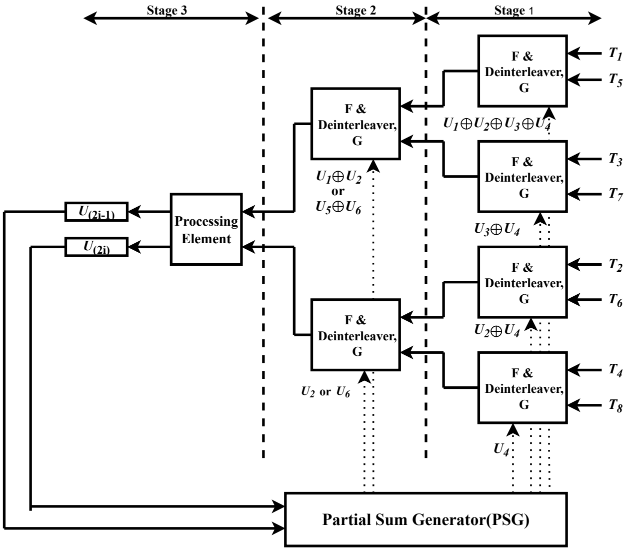

Fig. 10. (8,4) I-Polar tree-based architecture.

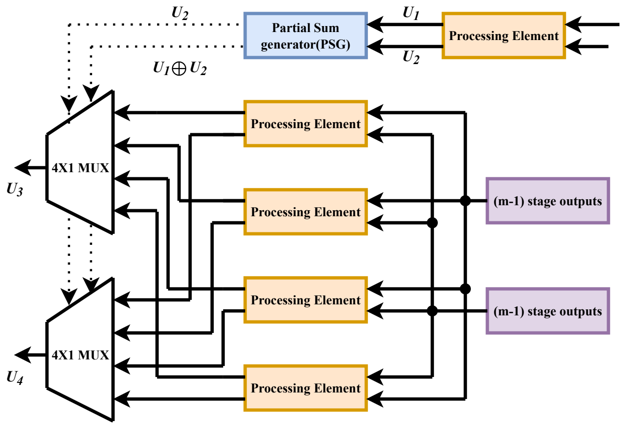

Fig. 11. Carry lookahead reformulation with processing elements.

Clock Cycles	1	2	3	4	5	6	7
Stage 1	Merged F with deinterleaver & G						
Stage 2		Merged F with deinterleaver & G			Merged F with deinterleaver & G		
Stage 3			P	P		P	P
Output			Outputs 1&2	Outputs 3&4		Outputs 5 & 6	Outputs 7 & 8

Tab. 3. Decoding cycles for (8,4) I-Polar codes.

Design	Proposed	[4]	[17]	[18]	[7]	[8]
Code Length (n, k)	(1024, 512)	(1024, 512)	(1024, 512)	(1024, 512)	(1024, 512)	(1024, 512)
Code Rate $[k/n]$	1/2	1/2	1/2	1/2	1/2	1/2
Technology	65 nm	65 nm	180 nm	65 nm	180 nm	180 nm
Frequency [MHz]	750	500	150	670	377	446.7
Latency [cycles]	728	2080	1560	1023	1534	1534
Gate Count	203426	214370	183637	268200	256340	295440
Throughput [Mbps]	712	246	49	670	252	298
Power [mW]	31	59	67	39	–	–
TSNT [scaled to 65 nm]	3.5	1.15	0.74	2.5	2.7	2.79

Tab. 4. Implementation results of polar and proposed I-Polar codes.

Code Length (n, k)	(1024, 512)	(512, 256)	(256, 128)	(128, 64)	(64, 32)	(32, 16)	(16, 8)
Code Rate $[k/n]$	1/2	1/2	1/2	1/2	1/2	1/2	1/2
Latency [cycles]	728	364	182	91	46	23	12
Gate Count	203426	101802	50950	25496	12814	6420	3282
Power [µW]	31426	15818	7986	4026	2016	1030	556

Tab. 5. Implementation results of I-Polar codes for different codeword length.

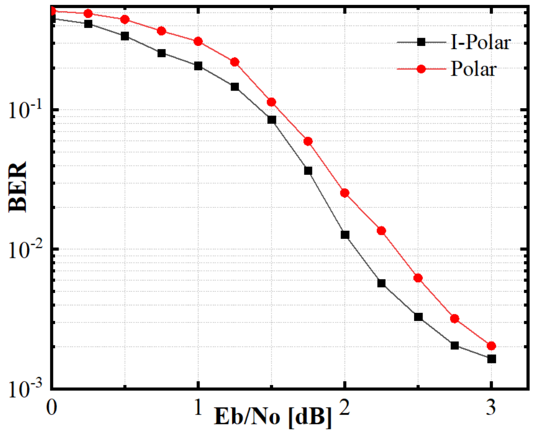

Fig. 12. E_b/N_0 [dB] vs. BER.

4. Results

Bit error rate performances of the conventional polar codes and the proposed I-Polar codes are shown in Fig. 12. Performance comparison reveals that the bit error rate of the proposed I-Polar codes is better than the conventional polar code when E_b/N_0 is less than 3 dB, when $E_b/N_0 > 3$ dB, the BER of both the conventional polar codes and the proposed I-Polar codes are alomost identical.Thus when signal strength is poor, the BER performence is superior in I-Polar codes.

The architectures of conventional polar codes and the proposed I-Polar code are modelled using Verilog HDL. The proposed I-Polar decoder is synthesized in the Synopsys design compiler using CMOS 65 nm technology. Table 4 shows the synthesis results of the proposed (1024, 512) I-Polar decoder and some SC decoders in the literature. The proposed I-Polar decoder with a code rate of 0.5 and (1024, 512) code length has achieved 50.5% average reduction in latency. Furthermore, the average power and gate count are reduced by 40.56% and 14% which are comparatively better than the existing SC decoders. Numerical results for different codeword length to demonstrate the generality are listed in Tab. 5.

5. Conclusion

The highly performant I-Polar codes, which achieves low latency due to the processing element, has been presented. In addition, the carry-lookahead adder-subtractor in the G function in $(m - 1)$ stages further decreases the latency and increases the speed of operation. The synthesis of the proposed I-Polar codes has been done in CMOS 65 nm technology using Synopsys design compiler. Implementation results show improvement in latency, area and throughput. The latency of the proposed decoder is reduced on an average of 50.5% compared to the conventional polar codes. The proposed I-Polar codes has better BER performance with reduced complexity compared to the conventional polar codes.

References

[1] ARIKAN, E. Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 2009, vol. 55, no. 7, p. 3051–3073. DOI: 10.1109/TIT.2009.2021379

[2] DIZDAR, O., ARIKAN, E. 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*, 2016, vol. 63, no. 3, p. 436–447. DOI: 10.1109/TCSI.2016.2525020

[3] FAN, Y., TSUI, C. An efficient partial-sum network architecture for semi-parallel polar codes decoder implementation. *IEEE Transactions on Signal Processing*, 2014, vol. 62, no. 12, p. 3165–3179. DOI: 10.1109/TSP.2014.2319773

[4] LEROUX, C., RAYMOND, A. J., SARKIS, G., et al. A semi-parallel successive-cancellation decoder for polar codes. *IEEE Transactions on Signal Processing*, 2013, vol. 61, no. 2, p. 289–299. DOI: 10.1109/TSP.2012.2223693

[5] CHEOLHO, K., HARAM, Y., SABOOH, A., et al. High-throughput low-complexity successive-cancellation polar decoder architecture using one's complement scheme. *Journal of Semiconductor Technology and Science*, 2015, vol. 15, no. 3, p. 427–435. DOI: 10.5573/JSTS.2015.15.3.427

[6] SHRESTHA, R., BANSAL, P., SRINIVASAN, S. High-throughput and high-speed polar-decoder VLSI-architecture for 5G new radio. In *32nd International Conference on VLSI Design and 18th International Conference on Embedded Systems (VLSID)*. Delhi (India), 2019, p. 329–334. DOI: 10.1109/VLSID.2019.00075

[7] YOON, H., KIM, T. Efficient successive-cancellation polar decoder based on redundant LLR representation. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2018, vol. 65, no. 12, p. 1944–1948. DOI: 10.1109/TCSII.2018.2811378

[8] SHRESTHA, R., SAHOO, A. High-speed and hardware-efficient successive cancellation polar-decoder. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2019, vol. 66, no. 7, p. 1144–1148. DOI: 10.1109/TCSII.2018.2877140

[9] OH, S., LEE, H. High-performance parallel concatenated polar-CRC decoder Architecture. *Journal of Semiconductor Technology and Science*, 2018, vol. 18, no. 5, p. 560–567. DOI: 10.5573/JSTS.2018.18.5.560

[10] FAN, Y., XIA, C., CHEN, J., et al. A low-latency list successive-cancellation decoding implementation for polar codes. *IEEE Journal on Selected Areas in Communications*, 2015, vol. 34, no. 2, p. 303–317. DOI: 10.1109/JSAC.2015.2504318

[11] YUAN, B., PARHIL, K. K. Low-latency successive-cancellation list decoders for polar codes with multibit decision. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2015, vol. 23, no. 10, p. 2268–2280. DOI: 10.1109/TVLSI.2014.2359793

[12] TAO, Y., CHO, S. G., ZHANG, Z. A configurable successive-cancellation list polar decoder using split-tree architecture. *IEEE Journal of Solid-State Circuits*, 2021, vol. 56, no. 2, p. 612–623. DOI: 10.1109/JSSC.2020.3005763

[13] SONG, W., ZHOU, H., NIU, K., et al. Efficient successive cancellation stack decoder for polar codes. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2019, vol. 27, no. 11, p. 2608–2619. DOI: 10.1109/TVLSI.2019.2925029

[14] ABBAS, S. M., FAN, Y., CHEN, J., et al. High-throughput and energy-efficient belief propagation polar code decoder. *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, 2017, vol. 25, no. 3, p. 1098–1111. DOI: 10.1109/TVLSI.2016.2620998

[15] CHIU, M. Interleaved polar (I-Polar) codes. *IEEE Transactions on Information Theory*, 2020, vol. 66, no. 4, p. 2430–2442. DOI: 10.1109/TIT.2020.2969155

[16] ROY, J. S., LAKSHMINARAYANAN, G., KO, S. B. High speed architecture for successive cancellation decoder with split-g node block. *IEEE Embedded Systems Letters*, 2021, vol. 13, no. 3, p. 118–121. DOI: 10.1109/LES.2020.3021144

[17] MISHRA, A., RAYMOND, A. J., AMARU, L. G., et al. A successive cancellation decoder ASIC for a 1024-bit polar code in 180nm CMOS. In *IEEE Asian Solid State Circuits Conference (A-SSCC)*. Kobe (Japan), 2012, p. 205–208. DOI: 10.1109/IPEC.2012.6522661

[18] YUN, H. R., LEE, H. Simplified merged processing element for the successive cancellation polar decoder. *Electronics Letters*, 2016, vol. 52, no. 4, p. 270–272. DOI: 10.1049/EL.2015.3432

About the Authors . . .

Nandini JALI received the B.Tech degree in ECE from SRITW, Hasanparthy, India, in 2013, M.Tech in VLSI and Embedded systems from KITSW, Warangal, India, in 2016. She is currently pursuing a Ph.D. degree at NIT, Warangal. Her research interests include VLSI architectures for forward error correction coding for wireless communication systems.

Pullakandam MURALIDHAR received B.Tech in ECE and M.Tech degree in EI from NIT, Warangal, India, in 1993 and 2004 respectively. Then he has received his Ph.D. degree from NIT Warangal. He is currently working as an Associate Professor in the ECE Department, NIT Warangal. His research interests include design of embedded systems and VLSI architectures for video processing systems.

Sreehari Rao PATRI received a B.E. degree in ECE from Nagarjuna University, India, in 1991, M.E. in Communication Systems from the IIT, Roorkee, India, in 1995 and Ph.D. in Mixed-signal IC design from the National Institute of Technology, Warangal. Currently, he is working as an Associate Professor in the Dept. of ECE at NIT Warangal. His research interests include Analog/Mixed Signal IC Design, Power Management IC Design, RF IC Design, On-chip Smart Sensors, and Communication Systems.