World Engineering & Applied Sciences Journal 6 (2): 78-84, 2015
ISSN 2079-2204

DOI: 10.5829/idosi.weasj.2015.6.2.22135

A High Speed Partial Sum Network for Low Latency Successive Cancellation Polar Decoder

Challa Vidya Tejaswini and R. Vijayabhasker

Department of ECE,
Anna University Regional Centre, Coimbatore, Coimbatore, India

Abstract: Polar codes have emerged as an important error correction codes due to their capacity achieving property and also seen them as a major breakthrough in coding theory. For (n, k) polar codes, the delay in partial sum network (PSN) implementation are a bottleneck for designing high speed polar decoder. Therefore, an efficient PSN updating algorithm and its corresponding architecture was proposed, which achieves delay performance irrespective of code lengths are introduced. Our work has significantly reduces critical path delay and hardware complexity. Simulation results show that high speed PSN with 2-bit successive cancellation pre-computation decoder achieves 40% hardware efficiency and 1.05 times improved in speed by comparing with the prior least latency SC decoder.

Key words- polar codes, Partial Sum Update, Frozen, Free, Successive cancellation

INTRODUCTION

A polar code is a linear block error correcting code developed by "Erdal Arýkan". Different from other well-known capacity-approaching codes, such as Turbo codes and LDPC codes, polar code is the first family of codes known to achieve channel capacity with explicit construction. Besides achieving the capacity for binary-input symmetric memory less channels, polar codes have also been proved to be able to achieve the capacity for any discrete and continuous memory less channel. They form a family of error correcting codes with an explicit and efficient construction encoding and decoding algorithms [1]. Moreover, based on the successive cancellation decoder (SCD), the channel capacity is asymptotically achieved by length polar codes with a low encoding and encoding complexity. To date, they are the first codes to provably achieve channel capacity with tractable decoding complexity. Polar codes are the only known solution which is both explicit and efficient in some information theoretic applications, such as achieving the secrecy capacity of the wiretap channel in the general case. They are therefore seen as a major breakthrough in coding and information theory. From a practical point of view, however, polar codes come close to achieving the channel capacity only for very large code lengths.

These are the first error-correcting codes with an explicit implementation to provably achieve the symmetric capacity of memory less channels. They have two properties that are of interest to data storage systems: (a) Very low error-floor due to their large stopping distance (b) Low complexity implementations.

A polar code decoder can be roughly divided into three types: Successive cancellation decoder (SCD) in [2], Belief propagation decoder (BPD) in [3] and SCD with multiple codeword candidates such as list decoder, stack decoder and chase decoder. In this work, the long latency of SC algorithm which was reduced using 2-bit decoding has been used. This 2-bit sc decoding has the following benefits. Firstly, hardware complexity and critical path in the last stage of SC algorithm is reduced significantly and at each clock cycle 2bits can be decoded simultaneously instead of 1 bit. As a result, this new decoder, referred to as 2-bit SC decoder, reduces latency from (2n-2) to (1.5n-2) without performance loss. With the use of overlapped scheduling technique, further reduces the latency to (n-1) [4, 5], referred to as the 2-bit SC

Correspondig Author: Challa Vidya Tejaswini, Department of ECE,
Anna University Regional Centre, Coimbatore, Coimbatore, India.

Overlapped scheduling decoder. With the use of pre-computation technique in [6, 7], further reduces the latency from to (n-1) to (3n/4-1), referred to as the 2-bit SC pre-computation decoder.

Generally two classes of operations, the arithmetic and the logical operations, are executed in a SC decoder. The arithmetic operation in SCD is min-sum operation. The logical operation is executed to generate binary hard values which determine the amount of interference to be cancelled in the arithmetic operation and are passed through an XOR network to generate these binary hard value bits which are also known as the partial-sums. Because of dependency of the arithmetic operations on the partial-sums, to avoid the idling of the processing elements, the partial-sums are required to be generated in the same cycle of the arithmetic operation. In this work, low-latency PSN architecture is proposed based on the polar codes property. Regardless of the code length, the proposed PSN architecture has a constant updating complexity for each partial-sum. As a result, its critical path is short, consisting of only one XOR gate and a 2-input AND gate.

The organization of this paper is as follows. Section II discusses about the architecture for conventional polar encoder and decoder. Section III discusses about the existing PSN architectures. Section IV discusses about the proposed PSN architecture. Section V provides the simulated results and compared with existing and proposed work. Section VI concludes the paper by describing observations and also gives the scope of future work [8].

Polar Codes and its Prelimnaries

Polar Encoding: With an efficient construction approach, the reliability of decoded bits will be polarized based on their different positions at the source data. An efficient polar-based encoder shown in Figure 1 can be constructed based on the following principles:

Sending required "message" bits at "good" positions, which can strongly guarantee the reliability of transmission.

Sending fixed "0" at "bad" positions, since after the transmission any decoded bits at these "bad" positions are highly unreliable.

In [3], those "0" bits are called "frozen" bits since these are fixed and their positions are known at both the encoder and the decoder. Similarly, call the non-frozen information bits as "free" bits in this project.

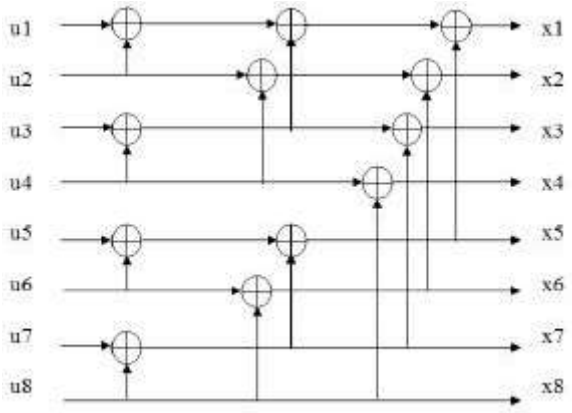


Fig. 1: Polar encoder with n = 8.

Accordingly, an (n,k) polar code contains information ("free") bits and (n-k) "frozen" bits. An (n,k) polar code can be constructed in two steps from the original k-bit message bits C_k^l . For this, denote the free and frozen bits positions.

The first encoding step for to get the polar code from original message is, to construct an n-bit source data vector, as $U = u_k^l$.

The second encoding step is to compute the transmitted codeword as, $X = X = {}_k^l$, by the generator matrix G.

$$x = uG,\ G=F^{\otimes n},\ F=\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix} \quad (1)$$

Polar 2-bit Sc Decoding: At the receiver end, corrupted by the transmission noise, the received codeword will no longer be x, but change to $y = \hat{y}_n^l$ and the goal of polar decoding is to recover u from y. In this recovery can be accomplished by the SC algorithm. With a recursive computation procedure, the SC algorithm can use the log-likelihood ratios (LLRs), in [9]. This estimated u is denoted as $\hat{u} = \hat{u}_l^n$.

Processing Element (Pe) For F and G Nodes: As shown in Figure 2, p nodes are used in stage- m and f and g nodes are used in other stages to calculate the propagated LLR values [10]. For simplicity of hardware design, the functions of f and g nodes are always implemented by unified processing elements (PEs) in Figure 3 shows the architecture of this PE based on the LLR version. Here S2C is the block that performs the

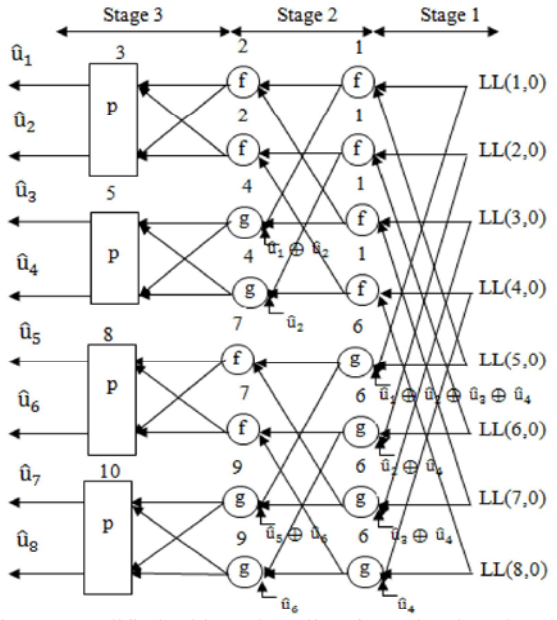


Fig. 2: Modified 2-bit sc decoding for polar decoder with n=8

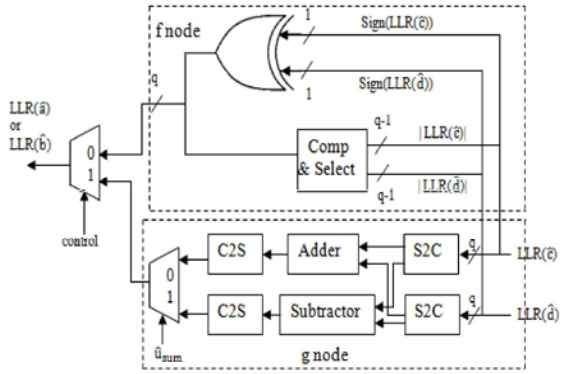


Fig. 3: PE architecture for f and g nodes.

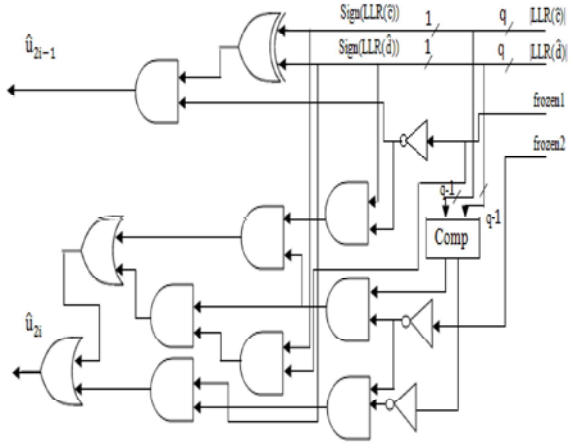


Fig. 4: P node architecture

conversion from sign-magnitude form to 2's complement form, while C2S unit carries out the inverse conversion. Additionally, adder and sub tractor are employed to carry out addition and subtraction between the two inputs. The corresponding sum and difference are selected by the partial sum signal $\hat{u}_{sum}$ from the PSG block. Finally, at the output end of the PE, control signal is used to determine the output as LLR $(\hat{a})$ or LLR $(\hat{b})$ which is propagated to the next stage.

Simply, $f(a,b) \approx \text{sign}(a)\,\text{sign}(b)\,\min(|a|,|b|)$. (2)

$$g(a,b) = a(-1)^{\hat{u}_{sum}} + b \quad (3)$$

Accordingly, the critical path delay of PE is T_{s2c} + T_{adder} + T_{c2s} +2T_{MUX}.

P Node: In this, the decision scheme in p node has been described based on the LLR representation. To implement its function, a straightforward approach is too complex, as shown in Figure 4. So its reformulation states that, the function of a node depends on the frozen conditions of $\hat{u}_{2i-1}$ and $\hat{u}_{2i}$ signals whether they are frozen bits or not. If $\hat{u}_{2i-1}$ is frozen, frozen1 will be 1, otherwise 0. Similarly, frozen2 will be 1 or 0 when $\hat{u}_{2i}$ is frozen or not. Secondly, the sign bits of LLR $(\hat{c})$ and LLR $(\hat{d})$, are employed for simplifying computations. Denoted as sign (LLR $(\hat{c})$) or sign (LLR $(\hat{d})$), these sign bits will be, respectively, 0 or 1 when the corresponding LLR values are non-negative or negative. Furthermore, the comp signal, which is the result of comparison between absolute value of LLR $(\hat{c})$ and LLR $(\hat{d})$, is also employed. When |LLR $(\hat{c})$ |≥|LLR $(\hat{d})$ |, comp will be 1, otherwise 0.

Boolean expression of $\hat{u}_{2i-1}$ and $\hat{u}_{2i}$ can be derived as follows:

$$\hat{u}_{2i-1} = \overline{\text{frozen 1}}\ ((\text{sign}(\text{LLR}(\hat{c})) \square \text{sign}(\text{LLR}(\hat{d}))) \quad (4)$$

$$\hat{u}_{2i} = (\overline{\text{comp}}\ \overline{\text{frozen 2}}\ \text{sign}(\text{LLR}(\hat{d}))) + (\text{comp}\ \overline{\text{frozen 1}}\ \overline{\text{frozen 2}}\ \text{sign}(\text{LLR}(\hat{d}))) + (\text{comp frozen1}\ \overline{\text{frozen 2}}\ \text{sign}(\text{LLR}(\hat{c}))) \quad (5)$$

Accordingly, this serial decoding leads to an overall latency of 14 cycles. In general, for (n,k) polar code, the decoding latency of SC decoder is (2n-2).

Overall Architecture for 2-Bit Sc Decoder: Based on the circuits of the PE and the p node in Figures 3 and 4, respectively, the overall 2-bit SC decoder can be

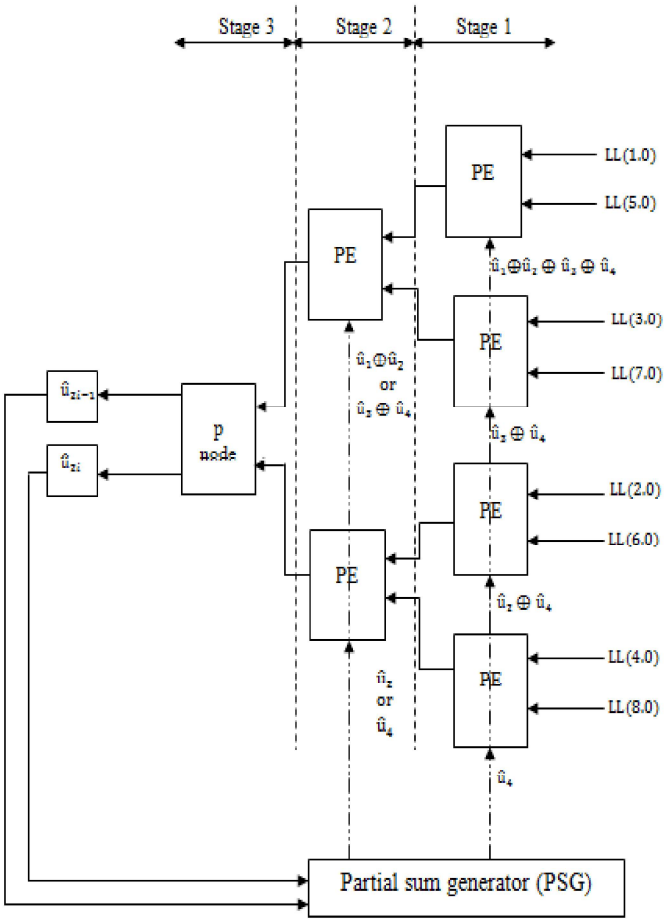


Fig. 5: Tree based structure for modified polar decoder with n=8.

constructed as a butterfly-like architecture (Figure 2) which is not hardware-efficient and at least half of nodes in each stage are always idle during decoding procedure. Therefore, in order to increase hardware utilization, tree-based architecture, is usually used to construct overall SC decoder. Figure 5 shows the architecture of a tree-based 2-bit SC decoder with n=8. In this design, when a particular stage is activated, all the nodes in that stage are activated. Therefore, a total of (n-2) PEs and a single p node are needed.

Similar to tree-based 2-bit SC architectures, the 2-bit SC version of overlapped and pre-computation techniques can be used to reduce output decoding latency satisfactorily.

2-Bit Sc Overlapped Scheduling Architecture: In this, it is observed that the p node has shorter critical path than the PE and this can be exploited to reduce the overall latency to (n-1).

After a careful examination of the decoding sequence, it is found that the computations of p and g nodes can be overlapped.

The new decoding sequence with overlapped scheduling has the computations of the p and g nodes are carried out in the same clock cycle; therefore, one cycle can be saved each time. Considering p node is activated for 0.5 cycles, this overlapped scheduling approach reduces the overall latency to (1.5n - 2) - (0.5n - 1) = (n-1).

2-Bit Sc Pre-computation Architecture: In [3], pre-computation technique was exploited to reduce the overall latency of the original SC algorithm. The essential idea of this method is to merge the computation of f and g nodes in the same stage. In each clock cycle, the computations of f and g nodes are carried out at the same time.As a result, the overall latency is 50% less than that of the conventional scheme. Moreover, in order to implement the pre-computation scheme, [3] proposed to employ merged PEs (Figure. 6).

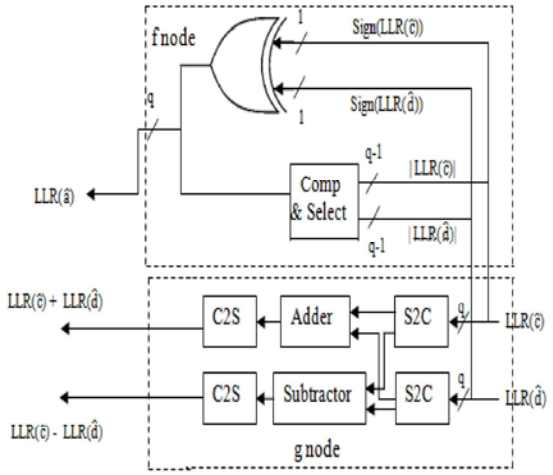


Fig. 6: Merged PE architecture for modified SC pre-computation decoding.

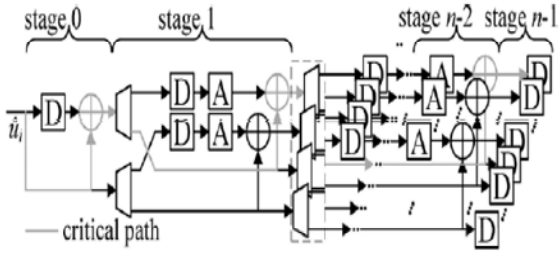
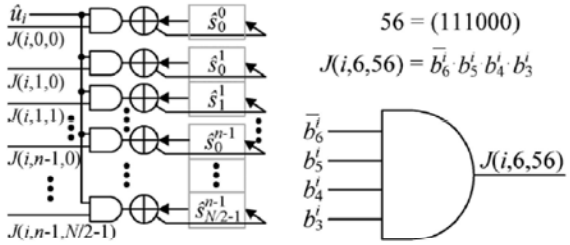


Fig. 7: Feed-forward architecture.



(a) (b)

Fig. 8: (a) partial-sum updating logic, 8(b) J (i,6,56) generation logic.

Based on this new scheme, the overall latency is further reduced from (n-1) to (3n/4 - 1).

Existing PSN Architectures

Feed-forward PSN Architecture: A straightforward manner to implement PSN is to input to the encoding signal flow graph shown in Figure. 7. The intermediate outputs at the corresponding XOR nodes and "split" nodes are the partial-sums required. FFA implements this method in a hardware-efficient way, based on the fact that is serially generated in the SCD. The critical path of FFA consists of n-1 XOR gates and n-2 DEMUXes,

Partial-Sum Update Logic: In [8], A parallel updating logic is adopted here. Generally, partial-sum $\hat{s}_j^t$ can be viewed as the XOR combination of a supporting subset of $\hat{u}_N$. Based on this, an auxiliary variable, J(i, t, j), is defined. If $\hat{u}_i$ belongs to the supporting subset of $\hat{s}_j^t$, J(i, t, j) = 1. Otherwise, J(i, t, j) = 0. Thus, $\hat{s}_j^t = \sum_{i=0}^{N-1} J(i,t,j)\,\hat{u}_i$. When a new $\hat{u}_i$ is decoded, $\hat{s}_j^t$ is serially updated in the following way

$$\hat{S}_j^t(I) = \hat{S}_j^t(i-1) \square (J(i,t,j).\,\hat{u}_i) \tag{6}$$

where $\hat{s}_j^t = \sum_{k=0}^{i} J(i,t,j).\,\hat{u}_k$. is the partially updated $\hat{s}_j^t$, when $\hat{u}$ (0 : i) is available.

The most complicated part for the PSUL logic is the evaluation of J(i, t, j).

Compared with the FFA, the delay of the PSUL is enhanced by the parallel updating logic. However, the complexity of the J(i, t, j) logic also scales up logarithmically with the code length N.

In summary, both the FFA and the PSUL have a complexity depending on N.

Proposed High Speed Psn Architecture: After a particular bit is decoded, let $\hat{u}_{N/2}$, only some partial sums $\hat{s}_j^t$'s with $j \geq N/2$ are required for the remaining g nodes.

At the same time, for the g functions that are required for the decoding of $\hat{u}_i$ where $0 \leq i \leq N/2$, only partial sums of $\hat{s}_j^t$'s where $j < N/2$ are required. These can be given as,

$$\hat{s}^t(0:N/2-1) = \hat{s}^{t-1}(0:N/2-1) \oplus (\hat{u}_i.G_N(i,0:N/2-1)) \tag{7}$$

for $0 \leq i \leq N/2$.

$$\hat{s}^t(N/2:N-1) = \hat{s}^{t-1}(N/2:N-1) \oplus (\hat{u}_i.G_N(i,N/2:N-1)) \tag{8}$$

for $N/2 \leq i \leq N$.

where $\hat{s}^i$ (N/2:N-1) for $0 \leq i \leq N/2$ is a zero vector because of the lower triangular structure of G_N.

According to these equations, $\hat{s}^i(j)$ and $\hat{s}^i(j+N/2)$, where $0 \leq j \leq N/2$, their calculations are independent and $\hat{s}^i(j+N/2)$ won't start until $\hat{s}^i(j)$ is used. This shows that, they can share same register for storing the intermediate value and hence only N/2 registers are needed, denoted by $r_{N/2}$. Comparing with feed-forward and partial sum update logic, the number of registers required to store is reduced to half.

Table 1: Comparison Analysis

Design Parameters	Conventional Sc Polarfa Decoder	Modified SC Polar Decoder with High Speed PSN		
		2-BIT SC	Overlapped SC	Pre-Computation SC
Gate counts	413	1306	1121	823
Output decoding latency (cycle)	14	10	7	5
Clock frequency (MHz)	750	750	750	750
Area of PSN	126	64	64	64
Power(mW)	167.32	332.97	323.28	315.23

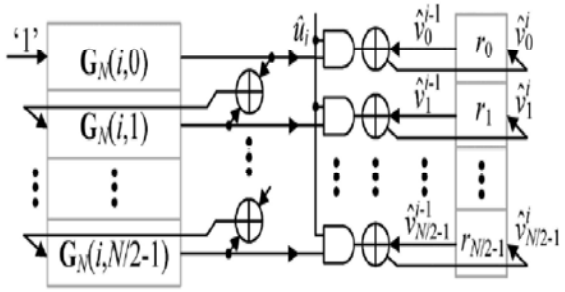


Fig. 9: Proposed high speeds PSN

The logic used here includes the encoding of G_N. Whenever a new bit is decoded, $\hat{S}^i$ is updated from $\hat{S}^{i-1}$ based on above equations and the critical path of the PSN architecture is independent of N and is equal to the sum of the delay of a 2-input AND gate and an XOR gate, as shown in Figure 9.

RESULTS AND DISCUSSION

Polar encoder with n=8 i.e, 4 information bits (1111) and 4 free bits (0000) have given as input. And the source data vector formed is 11101000, using the first encoding step. The corresponding codeword obtained is 10010110.

Polar decoder output was observed as 11101000, which is same as the encoder input (11101000) i.e. source data vector, thereby achieving capacity achieving property, with corresponding output decoding latency of 14 clock cycles using FFA.

Finally, the successive cancellation polar decoder has been designed with use of high speed PSN and counters in order to enable each PE stage accordingly. Here also the inputs to the decoder are Log-Likelihood Ratios, which have been calculated manually and converted them into equivalent hexadecimal single precision using floating point converters. It was observed the decoder output as 11101000, which is same as the encoder input (11101000) i.e. source data vector, thereby achieving capacity achieving property, with corresponding output decoding latency of 10 clock cycles.

After applying the overlapped scheduling technique for the modified 2-bit sc decoder, the output decoding latency is reduced to 7 clock cycles.

After applying the pre-computation scheduling technique for the modified 2-bit sc decoder, the output decoding latency is further reduced to 5 clock cycles. This has been done using verilog HDL in quartus II. The comparisons of different sc decoders for n=8 polar code are shown in Table 1.

CONCLUSION

In this paper, work included in implementing hardware architectures for the polar codes. By comparing with the conventional polar decoder using FFA with proposed work on implementing it with the efficient pre-computation high speed PSN architecture, reduces the latency with decreased hardware complexity. Future work includes in modifying the architecture with error detection block to improve the efficiency.

REFERENCES

1. Arikan, E., 2009. Channel polarization: A method for constructing capacity- achieving codes for symmetric binary-input memoryless channels, IEEE Trans. Inf. Theory, 55(7): 3051-3073.
2. Korada, S.B., E. Sasoglu and R. Urbanke, 2010. Polar codes: Characterization of exponent, bounds and constructions, IEEE Trans. Inf. Theory, 56(12): 6253-6264.
3. Bo Yuan, 2014. Student Member, IEEE and Keshab K. Parhi, Fellow, IEEE, Low-Latency Successive-Cancellation Polar Decoder Architectures Using 2-Bit Decoding,IEEE Trans. On Circuits and Systems—I: Regular Papers, 61(4).
4. Moriand, R. and T. Tanaka, 2009. Performance of polar codes with the construction using density evolution, IEEE Commun. Lett., 13(7): 519-521.
5. Tal, I. and A. Vardy, 2011. How to construct polar codes,arXiv: 1105.6164v1.

6. Zhang, C., B. Yuan and K.K. Parhi, 2012. Reduced-latency SC polar decoder architectures, in Proc. Int. Conf. Commun., pp: 3471-3475.
7. Zhang, C. and K.K. Parhi, 2013. Low-latency sequential and overlapped architectures for successive cancellation polar decoder, IEEE Trans. Signal Processing, 61(10): 2429-2441.
8. Berhult, G., C. Leroux, C. Jego and D. Dallet, 2013. Partial sums computation in polar codes, arXiv:1310.1712[Online].Available:http://arxiv.org/abs/1310.1712
9. Leroux, C., I. Tal, A. Vardy and W.J. Gross, 2011. Hardware architectures for successive cancellation decoding of polar codes, in Proceedings, IEEE ICASSP, pp: 1665-1668.
10. Leroux, C., A.J. Raymond, G. Sarkis and W.J. Gross, 2013. A semi-parallel successive-cancellation decoder for polar codes, IEEE Trans. Signal Processing, 61(2): 289-299.