SIMPLIFIED MULTI-BIT SC LIST DECODING FOR POLAR CODES

Jiangxue Han, Rongke Liu, Runxin Wang

School of Electronic and Information Engineering, Beihang University, Beijing, P. R. China

ABSTRACT

In this paper, we propose a simplified multi-bit successive cancellation list (SMSCL) decoding method for polar codes. In the proposed SMSCL decoding, frozen bits are ignored, and multiple information bits are intermediately decoded at each decision step. In addition, a modified two-stage pruning network is proposed to reduce the complexity of path pruning in SMSCL decoding. Compared to the existing multi-bit SCL decoders, the SMSCL decoder can achieve significant latency and complexity reduction over a wide range of code rates.

Index Terms— polar codes, successive cancellation list decoding, multi-bit decision, pruning technique.

1. INTRODUCTION

Polar codes, discovered by Arıkan in 2009, are a family of capacity-achieving error control codes under successive cancellation (SC) decoding algorithm [1]. However, polar codes perform poorly at short to moderate lengths. In order to solve this problem, SC list (SCL) decoding [2] is proposed, in which multiple decoding paths, rather than a single decoding path in SC decoding, are considered.

Although SCL decoding has notable performance gain over SC decoding [2], it comes at the cost of higher complexity. A series of researches has been done on hardware architecture design for SCL decoders [3–5]. Log-likelihood-ratios (LLRs) based SCL decoders are proposed to further reduce the complexity of SCL decoders [6, 7].

SCL decoders also suffer from long latency and low throughput due to the serial nature of their SC component decoders. Recently, a multi-bit SCL (MSCL) algorithm [4, 8] is presented to reduce the latency by intermediately decoding M bits simultaneously at each decision step, where M is an integer parameter. Other fast-SCL algorithms [9–11] improve the throughput by simplifying decoding process of consecutive frozen bits and consecutive information bits.

In this work, we first propose a simplified multi-bit SCL (SMSCL) decoding method to reduce the latency and complexity of SCL decoding. In the proposed SMSCL decoding, frozen bits are ignored, and at least M bits are intermediately

decoded at each decision step. Compared to the M-bit MSCL decoding, the number of bits which are intermediately decoded at each decision step is increased, whereas the complexity of path pruning remains the same. Then a partial sum network (PSN) is well-designed for the SMSCL decoder, since the number of input bits is non-fixed. The proposed PSN is designed by the inherent properties, which can reduce the complexity and critical path delay over the polar-encoder-like architecture. Finally, we proposed a modified two-stage pruning network (MTS-PN). Compared to the two-stage pruning network (TS-PN) in [12, 13], the MTS-PN improves the error performance with the same complexity. Simulations show that SMSCL decoders can reduce 6% to 80% decoding clock cycles over MSCL decoders depending on the code rate.

2. BACKGROUND

2.1. SC and SCL Decoding

In this paper, we define a_1^N as an N-tuple row vector $(a_1, a_2, ..., a_N)$. Denote c_1^N as an N-tuple row vector of constant c. Let A_N be an $N \times N$ matrix, $A_N(i, :)$ be the i-th row of A_N, $A_N(i_1 : i_2, :)$ be the sub-matrix of A_N by selecting the rows $i_1, i_1 + 1, \cdots, i_2$, and $A_N(i_1 : i_2, i_3 : i_4)$ be the sub-matrix of $A_N(i_1 : i_2, :)$ by selecting the columns $i_3, i_3 + 1, \cdots, i_4$. Define the function $\psi(x) = \frac{1}{2}(1 - \text{sgn}(x))$.

A polar code can be represented as $(N, K, \mathcal{A})$, where $N = 2^n$ is the code length, K is the number of information bits, and $\mathcal{A}$ is the index set of the information bits. The remaining $(N - K)$ bits are frozen bits, and are usually set to 0s. Polar codes are encoded by $x_1^N = u_1^N G_N$, where x_1^N is the codeword, $G_N = F^{\otimes n}$ is the generator matrix, and $\forall u_i \in \{0, 1\}$. $F^{\otimes n}$ is the n-th Kronecker power of the kernel matrix $F \triangleq \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

Let y_1^N be the the noisy signal of x_1^N at the receiver. Let $\hat{u}_i$ be the estimate of u_i, and $L_N^{(i)} = \ln \frac{P(y_1^N, \hat{u}_1^{i-1} | 0)}{P(y_1^N, \hat{u}_1^{i-1} | 1)}$ be the LLR of bit $\hat{u}_i$. Thus SC decoders obtain $\hat{u}_i$ sequentially by

$$\hat{u}_i = \begin{cases} 1, & i \in \mathcal{A} \text{ and } L_N^{(i)} < 0, \\ 0, & \text{otherwise.} \end{cases} \qquad (1)$$

Different from selecting a certain path at each decision step in SC decoding, SCL decoding extends two paths of $u_i = 0$ and 1. The complexity of SCL decoding is constrained by the list size L. In SCL decoding, L most reliable paths are

preserved at each decision step. Let $\hat{u}_i[l]$ be the estimate of u_i in the l-th path, where $l \in \{1, 2, ..., L\}$. We use *path metric* to measure the reliability of paths, and the path metric of $\hat{u}_1^i[l]$ is approximatively computed by

$$PM_l^{(i)} = PM_l^{(i-1)} + c_i[l] \cdot |L_N^{(i)}[l]|, \quad (2)$$

where (i) is the decision step index, $PM_l^{(0)} = 0$, $c_i[l] = 0$ when $\hat{u}_i[l] = \psi(L_N^{(i)}[l])$ and $c_i[l] = 1$ otherwise.

As u_i has two possible values, L paths with smallest path metrics are selected from $2L$ paths at the i-th decision step. After the N-th bit has been decoded, the path with the smallest path metric is regarded as the decoding result.

2.2. MSCL Decoding

A polar code is intermediately decoded bit by bit in standard SCL decoding, which leads to long latency and low throughput. In [4,8], an MSCL decoding approach is proposed, which reduces the latency by performing intermediate decoding of $M = 2^m$ bits simultaneously at each decision step and cutting down the likelihood update in the last m stages. Let $\hat{v}_{(i-1)M+1}^{iM}[l] = \hat{u}_{(i-1)M+1}^{iM}[l]G_M$, and $L_{N/M}^{(i,k)}[l]$ be the LLR of $\hat{v}_{(i-1)M+k}[l]$. The path metric of $\hat{u}_1^{iM}[l]$ in MSCL decoding is computed by

$$PM_l^{(i,M)} = PM_l^{(i-1,M)} + \sum_{k=1}^{M} c_{i,k}[l] \cdot |L_{N/M}^{(i,k)}[l]|, \quad (3)$$

where (i, M) represents that M bits are intermediately decoded at the i-th decision step ($i \in \{1, 2, \cdots, \frac{N}{M}\}$), $PM_l^{(0,0)} = 0$, $c_{i,k}[l] = 0$ when $\hat{v}_{(i-1)M+k}[l] = \psi(L_{N/M}^{(i,k)}[l])$ and $c_{i,k}[l] = 1$ otherwise. Since $\hat{u}_{(i-1)M+1}^{iM}[l] \in \{0,1\}^M$, L paths with smallest path metrics are selected from at most $2^M L$ paths in MSCL decoding.

3. PROPOSED SMSCL DECODING

Since an M-bit MSCL decoder reduces the latency of SCL decoder from $(3N - 2)$ to $(\frac{4N}{M} - 2)$ [4], an MSCL decoder with a larger M can achieve shorter latency and higher throughput. However, since L paths are preserved from up to $2^M L$ paths, a larger M also leads to a more complicated path pruning network. Thus M is usually no more than 8 for practical implementation [4], which is a limit to the throughput of MSCL decoder. In this section, we propose an SMSCL decoding method, which can further reduce the latency over MSCL decodeing by increasing the number of bits intermediately decoded at each decision step.

Suppose a polar code is divided into D groups, and all the bits in group i are intermediately decoded at the i-th decision step, where $i \in \{1, 2, \cdots, D\}$. Assume group i contains $M_i = 2^{m_i}$ consecutive bits, and there are C_i information bits among them. In group i, M_i and C_i are maximized under the constraints that

$$M_i \geq M, \text{ and } C_i \leq M, \quad (4)$$

Algorithm 1 Code Division

Input: $C_{M\,1}^{\;N/M}$
Output: M_1^D, C_1^D
 1: *Initialize* $i = 1$, $j = 1$
 2: **while** $j \leq N/M$ **do**
 3:     $C_i = C_{M,j}$, $\Delta_C = 0$, $\Delta_S = 1$
 4:     **while** $C_i + \Delta_C \leq M$ && $j < N/M$ **do**
 5:         $C_i = C_i + \Delta_C$
 6:         $\Delta_C = \sum_{k=j+\Delta_S}^{j+2\Delta_S-1} C_k$, $\Delta_S = 2 \cdot \Delta_S$
 7:     **end while**
 8:     $M_i = \Delta_S \cdot M$
 9:     $i = i + 1$, $j = j + \Delta_S$
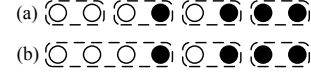10: **end while**

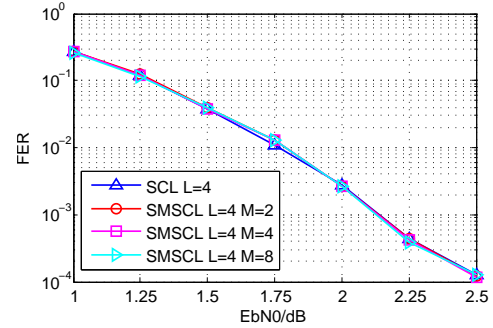**Fig. 1**: Code division of SMSCL decoding.

Fig. 2: FERs of SMSCL decoding method for a (1024,512) polar code with $L = 4$.

where M is a constant parameter. In MSCL decoding, we have $D = \frac{N}{M}$, and for $i \in \{1, 2, \cdots, \frac{N}{M}\}$, $M_i = M$. Let $C_{M,i}$ be the number of information bits in group i in MSCL decoding. Then M_1^D and C_1^D in SMSCL decoding can be obtained from $C_{M\,1}^{\;N/M}$ by Algorithm 1.

A simple example for a $(8, 4)$ polar code division in SMSCL decoding with $M = 2$ is given in Fig. 1, where write and black circles represent frozen and information bits respectively. The code is first divided into 4 2-bit groups in Fig. 1 (a), and the final code division is shown in Fig. 1 (b).

The path metric of SMSCL decoding can be updated in a similar way of MSCL decoding. Assume M_i bits are intermediately decoded at the i-th decision step in SMSCL decoding. Let $d_i = \frac{(\sum_{k=1}^i M_k)}{M_i}$, $\hat{v}_{(d_i-1)M_i+1}^{d_i M_i}[l] = \hat{u}_{(d_i-1)M_i+1}^{d_i M_i}[l]G_{M_i}$, and $L_{N/M_i}^{(d_i,k)}[l]$ be the LLR of $\hat{v}_{(d_i-1)M_i+k}[l]$. The path metric of $\hat{u}_1^{d_i M_i}[l]$ can be computed by

$$PM_l^{(i,M_i)} = PM_l^{(i-1,M_{i-1})} + \sum_{k=1}^{M_i} c_{i,k} \cdot |L_{N/M_i}^{(d_i,k)}[l]|, \quad (5)$$

where $PM_l^{(0,0)} = 0$, $c_{i,k} = 0$ when $\hat{v}_{(d_i-1)M_i+k}[l] = \psi(L_{N/M_i}^{(d_i,k)}[l])$ and $c_{i,k} = 1$ otherwise.

At the i-th decision step in SMSCL decoding, L most reliable paths are selected from 2^{C_i} paths. Since $C_i \leq M$ for all is, path pruning network in SMSCL decoder is identical to the one in MSCL decoder. Furthermore, since M_i bits are

intermediately decoded at the i-th decision steps, likelihood updates in the last m_i stages are cut down. Since $M_i \geq M$, SMSCL decoding reduces the latency over MSCL decoding.

Fig.2 shows FERs of SMSCL decoding for a (1024, 512) polar code with $L = 4$, concentrating with CRC-32. These FER curves indicate that SMSCL decoding has the same performance with SCL decoding.

4. PROPOSED PARTIAL SUM NETWORK

The overall architecture of SMSCL decoder is similar to that of MSCL decoder [4, 8, 13]. Compared to MSCL decoders, non-fixed M_i bits, rather than fixed M bits, are intermediately decoded at the i-th decision step in SMSCL decoder. Thus the main difference between SMSCL decoders and MSCL decoders is in the part of PSN. We elaborate the PSN for SMSCL decoders in this section.

4.1. Properties of Polar Codes

We first present several properties of polar codes, which help us to design the PSN. Proofs are omitted due to the page limit.

Property 1: For the generator matrix $G_N = F^{\otimes n}$, the $(k+J)$-th row $G_N(k+J, :)$ can be obtained from the k-th row $G_N(k, :)$ in the manner of

$$G_N(k+J, :) = G_N(k, :) \oplus \{0_1^J, G_N(k, 1:N-J)\}, \quad (6)$$

where J is a power of two, $0 < k \leq N$, $0 < k + J \leq N$, and $G_N(1, :) = \{1, 0_1^{N-1}\}$. $\oplus$ is the element-wise XOR operator.

Property 2: For the generator matrix G_N, consecutive J rows can be obtained from two smaller generator matrices in the manner of

$$G_N(kJ+1 : kJ+J, :) = G_{N/J}(k+1, :) \otimes G_J, \quad (7)$$

where J is a power of two, and $0 \leq k < N/J$.

Property 3: For the generator matrix G_N, the k-th row $G_N(k, :)$ can be obtained from the k/J-th row of a smaller generator matrix $G_{N/J}(k/J, :)$ in the manner of

$$G_N(k, :) = G_{N/J}(k/J, :) \otimes 1_1^J, \quad (8)$$

where J is a power of two, $k \geq J$, and k is divisible by J.

4.2. Partial Sum Network

In [14], an efficient PSN constructed by the inherent properties of polar codes is proposed, and it can reduce the complexity and critical path delay compared to the polar-encoder-like architecture in [15]. In this subsection, based on the properties in 4.1, a PSN with nonfixed-multibit-input (NFM-PSN) is proposed for SMSCL decoders.

For a length-N polar code, at most $\frac{N}{2}$ partial sums are required in SC/SCL decoding. Let

$$\hat{s}^{(i)} = \begin{cases} \hat{u}_1^i G_{N/2}(1:i, :), & 1 \leq i \leq \frac{N}{2}, \\ \hat{u}_{\frac{N}{2}+1}^i G_{N/2}(1:i-\frac{N}{2}, :), & \frac{N}{2} < i \leq N, \end{cases} \quad (9)$$

be the $\frac{N}{2}$-tuple partial sum vector after i bits are decoded. In SMSCL decoder, $d_k M_k$ bits have been intermediately decod-
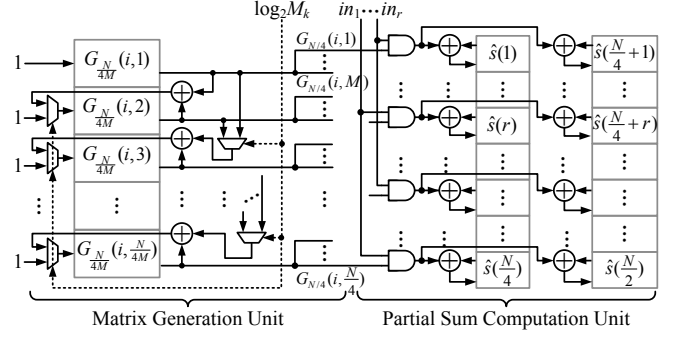

Fig. 3: Architecture of the proposed partial sum network.

ed at the k-th decision step, where d_k and M_k have been defined in Section III-B. Hence, at the k-th decision step, partial sum vector $\hat{s}$ can be updated by

$$\hat{s}^{(d_k M_k)} = \hat{s}^{(d_{k-1} M_{k-1})} \oplus$$
$$\hat{u}_{d_{k-1} M_{k-1}+1}^{d_k M_k} G_{\frac{N}{2}}(d_{k-1} M_{k-1} + 1 : d_k M_k, :), \quad (10)$$

where $\hat{s}^{(0)} = 0_1^{N/2}$, and $\hat{s}^{(\frac{N}{2})}$ is reset to $0_1^{N/2}$ before $u_{N/2+1}$ is intermediately decoded.

According to Property 2 and the identity $(AC) \otimes (BD) = (A \otimes B)(C \otimes D)$, we can obtain

$$\hat{s}^{(d_k M_k)}$$
$$= \hat{s}^{(d_{k-1} M_{k-1})} \oplus \hat{u}_{d_{k-1} M_{k-1}+1}^{d_k M_k} \left(G_{\frac{N}{2M_k}}(d_k, :) \otimes G_{M_k} \right) \quad (11)$$
$$= \hat{s}^{(d_{k-1} M_{k-1})} \oplus \left(G_{\frac{N}{2M_k}}(d_k, :) \otimes \hat{v}_{d_{k-1} M_{k-1}+1}^{d_k M_k} \right).$$

where $\hat{v}_{d_{k-1} M_{k-1}+1}^{d_k M_k} = \hat{u}_{d_{k-1} M_{k-1}+1}^{d_k M_k} G_{M_k}$.

In addition, taking advantage of the property $G_N = \begin{bmatrix} G_{N/2} & 0 \\ G_{N/2} & G_{N/2} \end{bmatrix}$, partial sum vector $\hat{s}$ can be updated by

$$\hat{s}^{(d_k M_k)} = \hat{s}^{(d_{k-1} M_{k-1})} \oplus \left(G_{\frac{N}{4M_k}}(d_k, :) \otimes \hat{v}_{d_{k-1} M_{k-1}+1}^{d_k M_k} \right). (12)$$

Notice that $\hat{s}$ should be reset to $0_1^{N/2}$ before u_1 and $u_{N/2+1}$ are decoded, and $\hat{s}_{N/4+1}^{N/2}$ is reset to $0_1^{N/4}$ before $u_{N/4+1}$ and $u_{3N/4+1}$ are decoded.

Since $\forall M_k \geq M$, according to Property 3, we have

$$G_{\frac{N}{4M}}(\frac{M_k d_k}{M}, :) = G_{\frac{N}{4M_k}}(d_k, :) \otimes 1^{\frac{M_k}{M}}, \quad (13)$$

that is,

$$G_{\frac{N}{4M_k}}(d_k, :) = G_{\frac{N}{4M}}(\frac{M_k d_k}{M}, b). \quad (14)$$

where b is a $\frac{N}{4M_k}$-tuple vector, and $b = \frac{M_k}{M} \cdot (1, 2, \cdots \frac{N}{4M_k})$. Forecast (14) to (12), we can obtain

$$\hat{s}^{(d_k M_k)} = \hat{s}^{(d_{k-1} M_{k-1})} \oplus \left(G_{\frac{N}{4M}}(\frac{d_k M_k}{M}, b) \otimes \hat{v}_{d_{k-1} M_{k-1}+1}^{d_k M_k} \right). (15)$$

(15) shows that partial sum vector $\hat{s}$ can be updated by a small generator matrix $G_{\frac{N}{4M}}$ and $\hat{v}_{d_{k-1} M_{k-1}+1}^{d_k M_k}$.

Assume information bits which are intermediately decoded at the k-th decision step are limited in the last r_k bits. Define $r = 2^{\lceil \log_2 \max_k (r_k) \rceil}$, where $\lceil x \rceil$ is the smallest integer not less than x. When $M_k > r$, $\hat{u}_{d_{k-1} M_{k-1}+1}^{d_k M_k - r}$ are frozen bits. Since frozen bits are set to 0s and do not change $\hat{s}$, r-bit inputs of $\hat{t}_{d_k M_k - r+1}^{d_k M_k}$ are enough, where $\hat{t}_{d_k M_k - r+1}^{d_k M_k} =$

Table 1: Resource consumption for partial sum networks.

Component	FFA [15]	HPPSN [14]	Proposed PSN
XOR	$\frac{N}{2} - M$	$\frac{3N}{4}$	$\frac{N}{2} + \frac{N}{4M}$
REG	——	$\frac{3N}{4}$	$\frac{N}{2} + \frac{N}{4M}$
AND	——	$\frac{N}{4}$	$\frac{N}{4}$
MUX	$\frac{N}{2} - 2M$	——	$\frac{N}{2M} - 3$
RAM	$\frac{N}{2} - 2M$	——	——

$\hat{u}_{d_k M_k - r+1}^{d_k M_k} G_r$.

The architecture of UFMPSN is shown in Fig. 3, which consists of a matrix generation unit (MGU) and a path metric computation unit (PMCU). The MGU generates $G_{\frac{N}{4M}}$ according to Property 1 under the control of $\log_2 M_k$, and the PMCU updates $\hat{s}$ according to (15). Both $G_{\frac{N}{4M}}$ and $\hat{s}$ are stored in registers. The inputs $in_1^r = \hat{t}_{d_k M_k - r+1}^{d_k M_k}$ when $M_k > r$, and $in_1^r = \left(\hat{v}_{d_{k-1}M_{k-1}+1}^{d_k M_k}, \cdots, \hat{v}_{d_{k-1}M_{k-1}+1}^{d_k M_k} \right)_{r/M_k}$ otherwise.

Resource consumption for different PSNs is shown in Table 1. A large amount of RAMs are required in the feedforward architecture (FFA) [15]. The high-performance (HP)-PSN in [14] merely match with standard SC/SCL decoders. Therefore, the proposed UFMPSN is more efficient than the existing PSNs in hardware implementation.

5. MODIFIED TWO-STAGE PRUNING NETWORK

For the SMSCL decoder with parameter M, L paths should be selected from $2^M L$ paths in the worst case. In the TS-PN [12, 13], q most reliable paths are selected among 2^M candidates of each existing path in the 1st stage, and L most reliable paths are selected from the remaining qL paths in the 2nd stage. The TS-PN is efficient when $q < L$. However, when q is not large enough, it will lead to huge performance loss.

In the MTS-PN, we maintain qL paths after the 1st stage, yet choose a larger q-value for more reliable path and vice versa, then the performance can be improved with the same complexity. Suppose the indices of the L existing paths are $\{1, 2, \cdots, L\}$ respectively, and the path with a smaller index has a higher reliability. Then the q-value for the l-th ($l \in \{1, 2, \cdots, L\}$) existing path is calculated by

$$q_l = round(\frac{2q(L-l)}{L+1}). \qquad (16)$$

Fig.4 shows FERs of SMSCL decoders of a (1024, 512) polar code with TS-PN and MTS-PN, concentrating with CRC-32. When L=4 and M=4, MTS-PN has no performance loss, whereas TS-PN leads to 0.1dB loss at 2.25dB. When L=8 and M=4, MTS-PN and TS-PN lead to 0.1dB and 0.2dB loss at 2.25dB. Thus the proposed MTS-PN achieves higher performance than the TS-PN with the same complexity.

6. LATENCY ANALYSIS

We analyse the decoding latency of SMSCL decoder in this section. At the i-th decision step in SMSCL decoder, $M_i = 2^{m_i}$ bits are intermediately decoded. Compared to standard

(a) L=4, M=4   (b) L=8, M=4
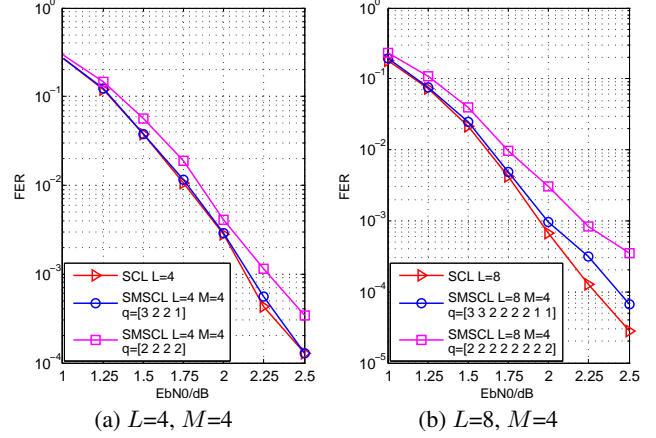
Fig. 4: FERs of SMSCL decoders for a (1024, 512) polar code with different pruning networks.

Table 2: Latency of decoders for length-1024 polar codes.

M	SCL	MSCL	SMSCL at different code rates				
			0.1	0.3	0.5	0.7	0.9
2		2046	381	866	1237	1590	1923
4	3070	1022	217	441	630	809	957
8		510	98	234	331	420	488

SCL decoder, path pruning is performed once instead of M_i times, reducing $L_1 = M_i - 1$ clock cycles. Suppose $m_i - m$ pipelines are inserted for path metric computation, then $L_2 = m_i - m + 1$ extra clock cycles are required. LLR updates of the last m_i stages are cut down, and $L_3 = 2M_i - 2$ clock cycles are reduced. Hence, $L_1 - L_2 + L_3 = 3M_i - m_i + m - 4$ clock cycles are reduced at the i-th decision step in total. Since standard SCL decoder requires $3N - 2$ clock cycles [4], the decoding clock cycles of SMSCL decoder is computed by

$$\mathcal{L} = (3N - 2) - \sum_i (3M_i - m_i + m - 4). \qquad (17)$$

Decoding clock cycles of SMSCL decoder is mainly affected by code rate K/N. We adapted polar codes of length $N = 1024$ constructed for AWGN channel with $E_b/N_0 = 2$ dB to show decoding clock cycles of different SCL decoders.

Table 2 represents latency of SCL, MSCL and SMSCL decoders with $M = 2, 4, 8$ at different rates. Clock cycles of MSCL decoder is calculated by $\frac{4N}{M} - 2$ [4]. Compare different columns of Table 2, SMSCL decoders can reduce 6% to 80% decoding clock cycles in contrast with SCL and MSCL decoders with the same M. The reduced clock cycles of SMSCL decoder decreases as R grows.

7. CONCLUSION

In this work, a simplified multi-bit SCL decoding method together with the corresponding hardware architecture are proposed, which can reduce the decoding latency. We also proposed a modified two-stage pruning network, which can enhance the performance over the two-stage pruning network with the same complexity.

8. REFERENCES

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[2] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[3] A. Balatsoukas-Stimming, A. J. Raymond, W. J. Gross, and A. Burg, "Hardware architecture for list successive cancellation decoding of polar codes," *IEEE Trans. Circuits Syst. II*, vol. 61, no. 8, pp. 609–613, 2014.

[4] B. Yuan and K. K. Parhi, "Low-latency successive-cancellation list decoders for polar codes with multibit decision," *IEEE Trans. VLSI Syst.*, vol. 23, no. 10, pp. 2268–2280, 2014.

[5] J. Lin and Z. Yan, "An efficient list decoder architecture for polar codes," *IEEE Trans. VLSI Syst.*, pp. 1–11, 2015.

[6] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, 2015.

[7] B. Yuan and K. K. Parhi, "Successive cancellation list polar decoder using log-likelihood ratios," in *2014 48th Asilomar Conf. on Signals, Systems and Computers*, 2014, pp. 548–552.

[8] B. Yuan and K. K. Parhi, "Reduced-latency LLR-based SC list decoder for polar codes," in *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*, 2015, pp. 107–110.

[9] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Increasing the speed of polar list decoders," in *IEEE Workshop on Signal Process. Syst. (SiPS)*, 2014, pp. 1–6.

[10] J. Lin, C. Xiong, and Z. Yan, "A reduced latency list decoding algorithm for polar codes," in *IEEE Workshop on Signal Process. Syst. (SiPS)*, 2014, pp. 1–6.

[11] G. Sarkis, P. Giard, A. Vardy, C. Thubeault, and W. J. Gross, "Unrolled polar decoders, part II: Fast list decoders," arXiv:1505.01466 [Online].

[12] C. Xiong, J. Lin, and Z. Yan, "Symbol-based successive cancellation list decoder for polar codes," in *IEEE Workshop on Signal Process. Syst. (SiPS)*, 2014, pp. 1–6.

[13] C. Xiong, J. Lin, and Z. Yan, "Symbol-decision successive cancellation list decoder for polar codes," *IEEE Trans. Signal Process.*, vol. 64, no. 3, pp. 675–687, 2016.

[14] Y. Fan and C. Y. Tsui, "An efficient partial-sum network architecture for semi-parallel polar codes decoder implementation," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3165–3179, 2014.

[15] C. Zhang and K. K. Parhi, "Low-latency sequential and overlapped architectures for successive cancellation polar decoder," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2429–2441, May 2013.