JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE, VOL.15, NO.3, JUNE, 2015
http://dx.doi.org/10.5573/JSTS.2015.15.3.427
ISSN(Print) 1598-1657
ISSN(Online) 2233-4866

High-Throughput Low-Complexity Successive-Cancellation Polar Decoder Architecture using One's Complement Scheme

Cheolho Kim, Haram Yun, Sabooh Ajaz, and Hanho Lee

Abstract—**This paper presents a high-throughput low-complexity decoder architecture and design technique to implement successive-cancellation (SC) polar decoding. A novel *merged processing element with a one's complement scheme*, a *main frame with optimal internal word length*, and optimized *feedback part* architecture are proposed. Generally, a polar decoder uses a two's complement scheme in *merged processing elements*, in which a conversion between two's complement and sign-magnitude requires an adder. However, the novel merged processing elements do not require an adder. Moreover, in order to reduce hardware complexity, optimized *main frame* and *feedback part* approaches are also presented. A (1024, 512) SC polar decoder was designed and implemented using 40-nm CMOS standard cell technology. Synthesis results show that the proposed SC polar decoder can lead to a 13% reduction in hardware complexity and a higher clock speed compared to conventional decoders.**

Index Terms—**Polar code, successive-cancellation, decoder, one's complement, high-throughput, low-complexity**

I. INTRODUCTION

Polar codes, proposed by Arikan in 2009 [1], have attracted a lot of attention because of their excellent capacity-achieving property over a binary-input discrete memoryless channel (B-DMC). Due to their explicit structure and low-complexity encoding/decoding scheme, polar codes have emerged as one of the most important in coding theory.

To date, much of the work has addressed several theoretical aspects of polar codes and is aimed at improving the error correction performance of polar codes of moderate lengths [2-7]. However, few publications have reported implementation of polar decoders. Pamuk [8] reported an FPGA implementation of a polar decoder based on the belief-propagation (BP) algorithm. Although a BP decoder has particular advantages in parallel design, due to the requirement for a large number of processing elements (PEs), the BP decoder is not attractive for practical applications. Several researchers have viewed the successive-cancellation (SC) algorithm as a good candidate for hardware design of polar decoders due to its low complexity [10-13]. The *semi-parallel SC decoder* from Leroux et al. [10] has a very low processing complexity, while memory complexity remains similar to previous architectures, also from Leroux et al. [9]. However, due to the inherent serial nature of the SC algorithm, these SC decoders have significant disadvantages with respect to both long latency and low throughput. Since SC decoding has low intrinsic parallelism, look-ahead techniques [11] were proposed to reduce decoding latency and increase throughput of SC decoders, while using limited extra hardware resources. Yuan and Parhi [13] presented a 2b-SC-precomputation decoder that

Manuscript received Oct. 15, 2014; accepted May. 28, 2015
Dept. of Information and Communication Engr. Inha University, Incheon, 402-751, Korea
E-mail : hhlee@inha.ac.kr

reduces decoding latency without performance loss. However, these low-latency architectures do not show detailed implementation results and bit error rate (BER) performance. The *list SC decoder* from A. Balatsoukas-Stimming et al. [14] shows higher decoding performance than SC decoder [11], whereas the list SC decoders has significant disadvantages with respect to both high hardware complexity and low throughput.

In this paper, we propose a high-throughput low-complexity architecture for SC polar decoding. A novel *merged processing element (M-PE) with a one's complement scheme, a main frame with optimal internal word length*, and an optimized *feedback part* architecture are proposed. Generally, the polar decoder uses a two's complement scheme in the M-PE, in which a conversion between two's complement and sign-magnitude requires an adder [11]. However, our novel M-PEs do not require an adder. Moreover, in order to reduce hardware complexity, optimized *feedback part* approaches are also presented.

The rest of this paper is organized as follows. Section II describes design issues related to SC polar decoding and provides analysis of a fixed-point BER simulation. Section III presents the proposed SC polar decoder architecture and the novel design techniques. In Section IV, the results and a comparison are presented. Finally, a conclusion is provided in Section V.

II. Design Issues Related to SC Polar Decoding

1. Construction of Polar Codes

By exploiting channel polarization, polar codes approach the symmetric capacity of a channel as the code length, N, increases. Channel polarization creates N independent channels, W, where, as N → ∞, the probability of error-free transmission approaches either 1 or 0.5 [1]. In other words, as N → ∞, each bit's probability of being successfully estimated approaches 1 (perfectly reliable) or 0.5 (completely useless). The ratio of reliable bits approaches the capacity of the channel. Let $N = 2^n$ ($n > 1$), $\mathbf{u} = (u_0, u_1, \ldots u_{N-1})$ and $\mathbf{c} = (c_0, c_1, \ldots c_{N-1})$, where **u** is the input bit and **c** is the corresponding codeword. A codeword of length N can be represented using a generator matrix **G**[1].

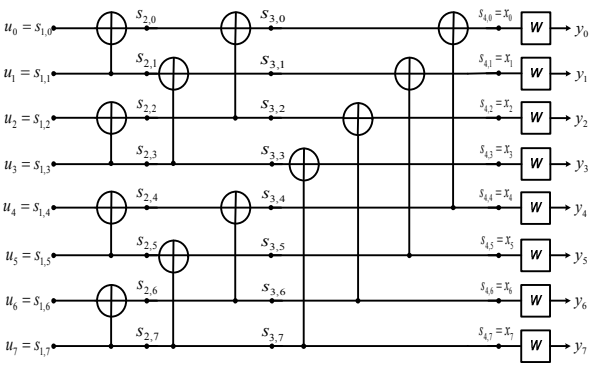

Fig. 1. Polar codes encoder with N = 8 [10].

$$c = uG \tag{1}$$

Here $\mathbf{G} = \mathbf{F}^{\otimes \mathbf{m}}$, where $\mathbf{F}^{\otimes \mathbf{m}}$ denotes the *m*-th Kronecker power of $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. For example, for n = 3,

$$\mathbf{F}^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{2}$$

The equivalent graph representation of $\mathbf{F}^{\otimes 3}$ is illustrated in Fig. 1, where $\mathbf{u} = u_0^7$ represents the information-bit vector and $\mathbf{x} = x_0^7$ is the codeword sent over the channel [10]. We use the same notation for vectors as that of [1], namely u_a^b consists of bits u_a .... u_b of the vector **u**. An (N, *k*) polar code uses *k* bits among the N most reliable bits to transmit the information's *k* bits and force the remaining N - *k* bits, called frozen bits, to zero. The location of the information and frozen bits is determined using the method described by Tal and Vardy [5].

2. Conventional SC Decoding Algorithm

The SC decoding algorithm [1] works by successively estimating the bits $\hat{u}_i$, i = 0, ... N-1, using the channel output **y** and the previously estimated bits $\hat{u}_0$ to $\hat{u}_{i-1}$, as shown in Fig. 2. As demonstrated by Leroux et al. [9],

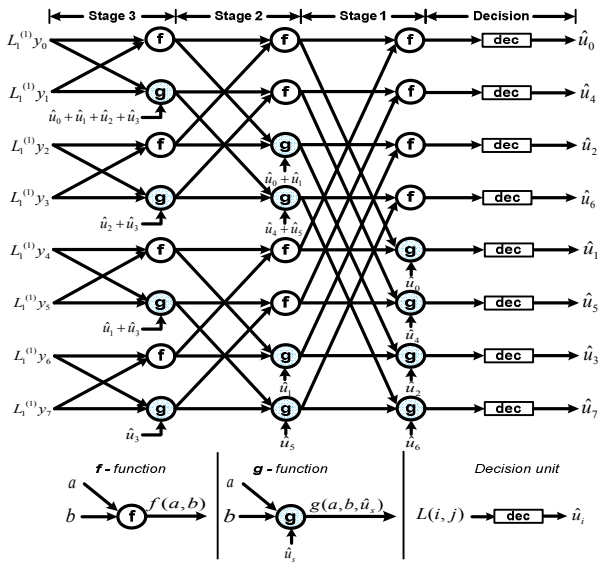

Fig. 2. Decoding procedure of polar codes with N = 8-bit.

it can be carried out without the use of multiplication or division by expressing probabilities as log-likelihood ratios (LLRs), denoted L(*i, j*) from the node at row *i* and stage *j*, and by resorting to min–sum approximation. With this new notation, LLR values for the received bit L(*i, j*) can be denoted as L(*i*, 0). Hence, the decision rule for becomes

$$\hat{u}_i = \begin{cases} 0, & if \quad L(i,0) > 0 \\ 1, & otherwise \end{cases} \tag{3}$$

It is noted that the LLRs with **f** and **g** functions can be generated by recursively applying Eqs. (4) and (5), respectively.

$$\mathbf{f}(a, b) = \text{sign}(a)\text{sign}(b)\min(|a|, |b|) \tag{4}$$

$$\mathbf{g}(a, b, \hat{u}_s) = (-1)^{\hat{u}_s} a + b \tag{5}$$

Function **f** can be computed as soon as a and b are available. On the other hand, the computation of **g** requires knowledge of $\hat{u}_s$. Notice that $\hat{u}_s$ is the module-2 sum of partial previously decoded bits, which can be computed by using the construction of a polar code. The need for partial sum computations causes strong data dependencies in the SC decoding algorithm. This constrains, the dependencies in which the decoded bit, can be computed in the Fig. 1. For instance, in Fig. 1, $s_{3,0}$ is estimated by propagating $\hat{u}_0^3$ in the polar code

CC	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Stage 3	**f**							**g**						
	f							**g**						
	f							**g**						
	f							**g**						
Stage 2		**f**			**f**				**f**			**g**		
		f			**f**				**f**			**g**		
Stage 1			**f**	**g**		**f**	**g**			**f**	**g**		**f**	**g**
Out			$\hat{u}_0$	$\hat{u}_1$		$\hat{u}_2$	$\hat{u}_3$			$\hat{u}_4$	$\hat{u}_5$		$\hat{u}_6$	$\hat{u}_7$

Fig. 3. Scheduling for the SC decoding algorithm with N = 8-bit.

encoder: $\hat{u}_0 \oplus \hat{u}_1 \oplus \hat{u}_2 \oplus \hat{u}_3$. This partial sum of $\hat{u}_0^3$ is then used to compute L(2, 3). Fig. 3 shows the scheduling of the decoding algorithm. At each clock cycle (CC), LLRs are evaluated by computing function **f** function or **g** function. It is assumed here that those functions are calculated as soon as the required data is available. Once the channel information $\hat{y}_0^{N-1}$ is available on the left hand side of the decoder, the bits $\hat{u}_i$ are successively estimated by updating the appropriate nodes of the graph, from left to right. When the bits $\hat{u}_i$ are estimated, all partial sums involving $\hat{u}_i$ are updated, allowing future evaluations of **g** function to be carried out. Therefore, the decoded bits can only be computed in a successive manner.

3. Analysis of Internal Word Length

In this section, the effects of internal word length parameters in terms of error-correction performance of SC polar codes are described. The optimal internal word length for a SC decoder can be decided.

Fig. 4 shows the BER performance comparison of various SC polar codes in which the code length from 2^9 to 2^{11} using q = 5 quantization bits in LLR and internal word length w bits are used. As show in Fig. 4, BER performance shows that the fixed-point operations with an optimum internal word length have decoding performance close to a floating-point operation. That is, the optimum internal word length W for (1024, 512) size and (2048, 1024) size was decided to 14 bits and 15 bits, respectively, since the BER degradation with respect to the floating point and to the hardware complexity is very small. However, w = 13 bits for (1024, 512) and w = 14

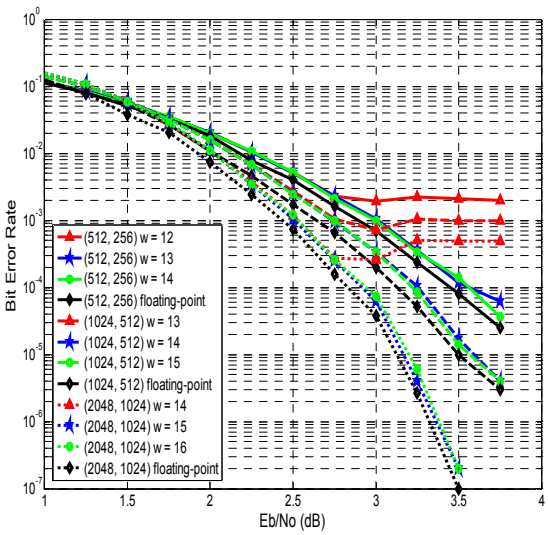

Fig. 4. BER performance of various SC polar codes.

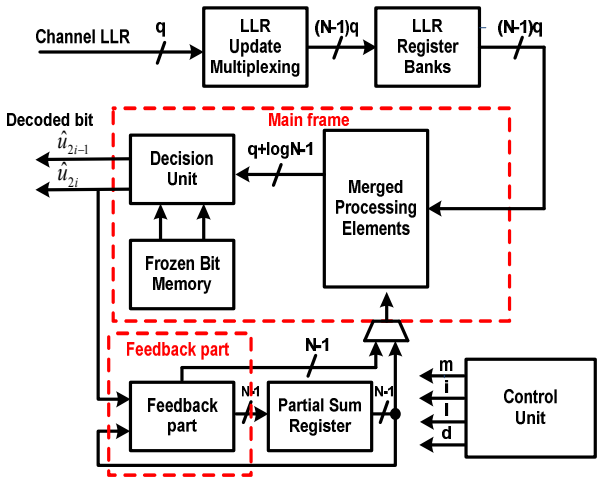

Fig. 5. Proposed SC polar decoder architecture.

bits for (2048, 1024) show the error floor due to overflow. In order to avoid the overflow in the decoding operation, M-PEs always calculate Eqs. (4) and (5) with an optimum internal word length of W bits.

III. Proposed SC Polar Decoder Architecture

As shown in Fig. 6, the proposed SC polar decoder architecture consists of two major units: *main frame* and *feedback part*. The *main frame* consists of M-PEs, a decision unit and frozen bit memory. The *feedback part*, which is defined with a power of 2 recursively, computes the partial sums required by the M-PEs to calculate Eq. (5).

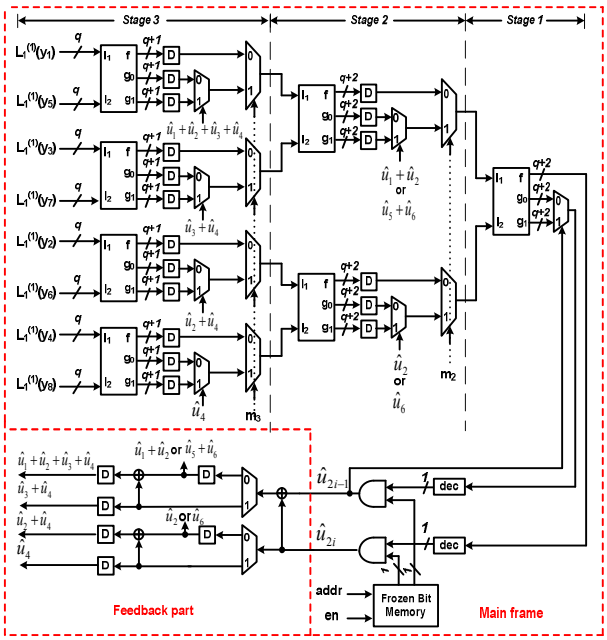

Fig. 6. Proposed N = 8-bit SC polar decoder architecture with optimal internal word length.

1. Main Frame Architecture with Optimum Internal Word Length

In order to reduce hardware complexity and avoid overflow, the optimum internal word length can be found to design the efficient main frame architecture. The main frame architecture employs tree-based architecture. In order to reduce hardware complexity of main-frame architecture, the internal word length of M-PE in each stage can be determined by exploiting the fact that the optimum internal word length in subsequent M-PE is increased by one bit per stage.

2. Merged Processing Element with an One's Complement Scheme

In this section, we discuss the detailed hardware architectures of the proposed M-PE architecture using one's complement scheme, which is the main arithmetic component of the decoder. It contains the arithmetic logic that carries out the likelihood estimations using Eqs. (4) and (5). The hardware complexity of the M-PE is significantly higher than that of other blocks, which results in very high hardware complexity for the SC decoder. Thus, minimizing the hardware complexity of the M-PE is critical to reducing the hardware complexity

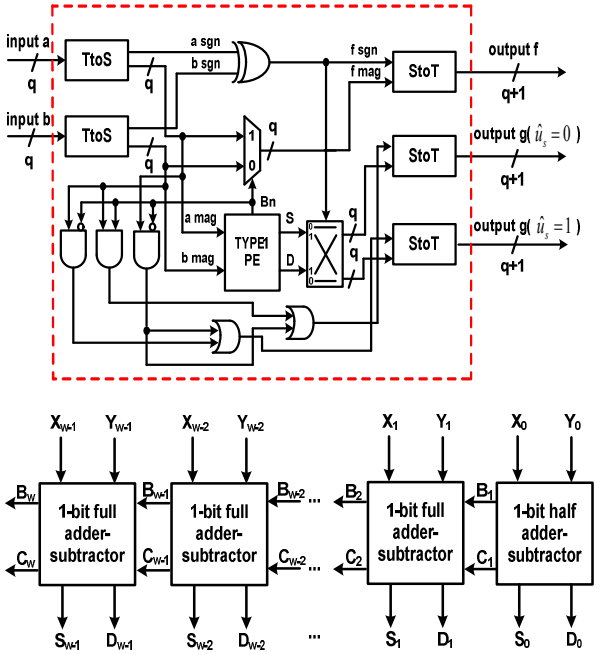

Fig. 7. (a) Conventional structure of the merged processing element (M-PE) [12], (b) Conventional structure of the Type I PE [12].

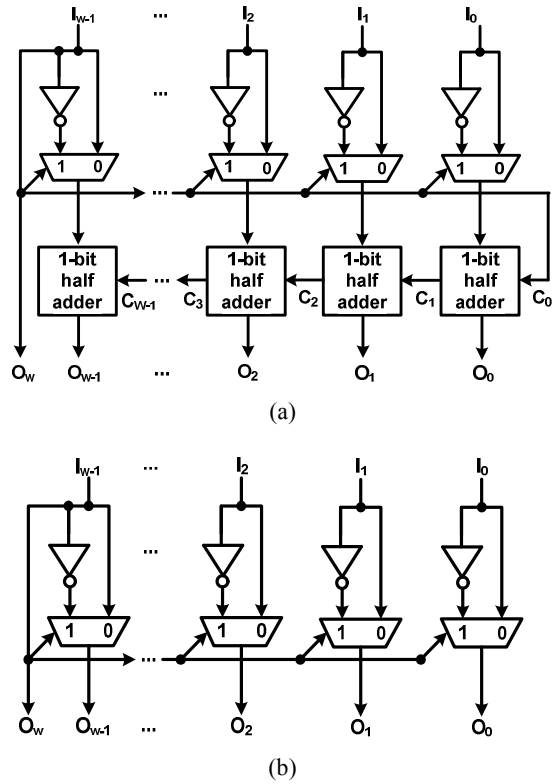

Fig. 8. (a) Conventional structure of TtoS block [12], (b) Proposed structure of OtoS block.

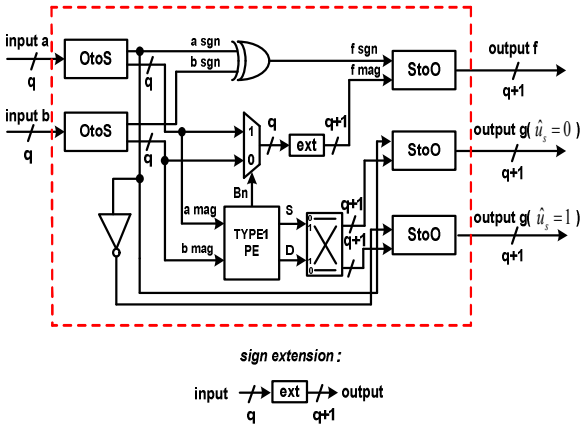

Fig. 9. Proposed structure of the merged processing element (M-PE).

of the SC decoder.

The conventional M-PE [11] consists of the TYPE1 PE, two's complement to sign-magnitude (TtoS) and sign-magnitude to two's complement (StoT), as depicted in Fig. 7. The computation of Eq. (4) requires an XOR gate and a multiplexer (MUX), while B_n is already available from the TYPE1 PE. The TYPE1 PE is in charge of calculating the two possible **g** outputs Eq. (5) in parallel and consists of an adder-subtractor, as shown in Fig. 7(b). For the full adder, the sum and carry-out bit are represented by S and C_{out}. The difference and borrow-out produced by the full subtractor are denoted by D and B_{out}. The TtoS block performs the conversion from two's complement to sign-magnitude representation. The StoT block performs the reverse conversion. The conventional architecture of the TtoS block [12] is illustrated in Fig. 8(a).

In this work, we propose one's complement to sign-magnitude (OtoS) and sign-magnitude to one's complement (StoO) blocks for the merged processing element that employs the one's complement scheme, as shown in Fig. 9. The StoO block is similar to the OtoS block in Fig. 8(b). We see that the proposed OtoS block is much simpler than the conventional TtoS block. This leads to two benefits. First, since all the half adders are removed, the hardware complexity of the M-PE is reduced compared to the conventional TtoS. Second, the critical path of TtoS is only $T_{inv} + T_{MUX}$, which is much shorter than that of the conventional TtoS. The conventional two's complement M-PE architecture of Zhang et al. [11] calculates Eqs. (4) and (5) by converting two's complement to sign-magnitude representation. On the other hand, the proposed one's

Table 1. Calculation of transformed **f** function and **g** function based on one's complement format

Case	**tf**(a, b)	Case	**tg**(a, b, =0)	**tg**(a, b, =1)
$a \geq 0, b \geq 0$	**f**	$a \geq 0, b \geq 0$	**g**$_0$	**g**$_1$ **- 1**
sign(a)≠sign(b) | a | ≤ | b |		$a < 0, b > 0$		
sign(a)≠sign(b) | a | > | b |	**f - 1**	$a > 0, b < 0$	**g**$_0$ **+ 1**	**g**$_1$
$a < 0, b < 0$		$a < 0, b < 0$		

complement M-PE architecture is not required to add the carry in the conversion from one's complement to sign-magnitude representation. In this section, we describe how input LLR values in the sign-magnitude conversion operation can be calculated, leading us to update Eqs. (4, 5) using the proposed OtoS. Here, Eqs. (4, 5) using two's complement are denoted as **f** = **f**(a, b), **g**$_0$ = **g**(a, b, $\hat{u}_s$ =0) and **g**$_1$ = **g**(a, b, =1), where a and b are LLR values and $\hat{u}_s$ is the partial sum of the decoded bit.

Table 1 shows the transformed equations from Eqs. (4, 5) for the one's complement scheme, which are used for BER simulation for the proposed one's complement scheme. These transforms depend on the sign of a and b. Therefore, there are four possible examples, as follows:

1) a is positive and b is positive in **tf**(a , b), ($a \geq 0, b \geq 0$)

When both a and b are positive, their sign-magnitude forms are the same as the two's complement forms. As a result, the proposed OtoS and StoO in the M-PE will not be used in this case.

2) a is negative and b is negative in **tf**(a , b), ($a < 0, b < 0$)

When both a and b are negative, **tf** > 0; thus, the magnitude of **tf** is just the absolute minimum value of a and b, whereas the sign of **tf** is always positive. Since a and b are negative, the sign-magnitude needs to subtract 1 in the proposed OtoS. However, since the result of **tf** is always positive, StoO is not used in this case. As a result, the transformed f function is defined as follows:

$$\mathbf{tf}(a, b) = \mathbf{f}(a, b) - 1 \text{ when } a < 0, b < 0 \tag{6}$$

3) a is positive and b is negative in **tg**(a, b, $\hat{u}_s$ =0) (a > 0, b < 0)

When a is positive and b is negative, a has the same value of the sign-magnitude and two's complement form. Thus, only the conversion of b is required to perform the transformed **g**$_0$ function. In this case, since b is negative for the one's complement, its magnitude value is calculated by subtracting 1; that is, | b | - 1. As a result, the transformed **g**$_0$ function can be derived as follows:

$$a = |a|, \qquad b = -(|b| - 1) \tag{7}$$

$$\begin{aligned} \mathbf{tg}_0 &= a + b \\ &= |a| - |b| + 1 \\ &= \mathbf{g}(a, b, \hat{u}_s = 0) + 1 \text{ when } a \geq 0, b < 0 \end{aligned} \tag{8}$$

4) a is negative and b is positive in **tg**(a, b, $\hat{u}_s$ =1) (a < 0, b > 0)

When a is negative and b is positive, only the conversion of a is required to perform the transformed **g**$_1$ function, as follows:

$$a = -(|a| - 1), \qquad b = |b| \tag{9}$$

$$\begin{aligned} \mathbf{tg}_1 &= -a + b \\ &= |a| - 1 + |b| \\ &= \mathbf{g}(a, b, \quad = 0) - 1 \text{ when } a < 0, b > 0 \end{aligned} \tag{10}$$

Summarizing the above four cases, we conclude that instead of using two's complement TtoS and StoT, the sign-magnitude–based addition or subtraction can still be carried out with slight modification of Eqs. (4, 5), which is one's complement OtoS and StoO.

3. Feedback Part

Every computation of Eq. (5) requires a specific input corresponding to the sum of a subset of the previously estimated bits. Zhang and Parhi [12] proposed recursive construction of a partial sum unit called the *feedback part*. In general, for an N-bit length decoder, since the architecture of the *feedback part* is defined recursively for powers of 2, the general parallel pipelined architecture can be constructed with the recursive relationship in Fig. 10(a). Here, module U_n can be constructed based on module U_{n-1} and (n - 1) extra XOR-pass elements. However, we note that for U_n, the number of corresponding registers significantly increases with a complexity of 2^n and the feedback part has very high hardware complexity, which is impractical for polar codes with length over 2^{10}. One possible approach to reduce the hardware complexity is to employ a RAM instead of flip-flops. However, storing the partial sums in

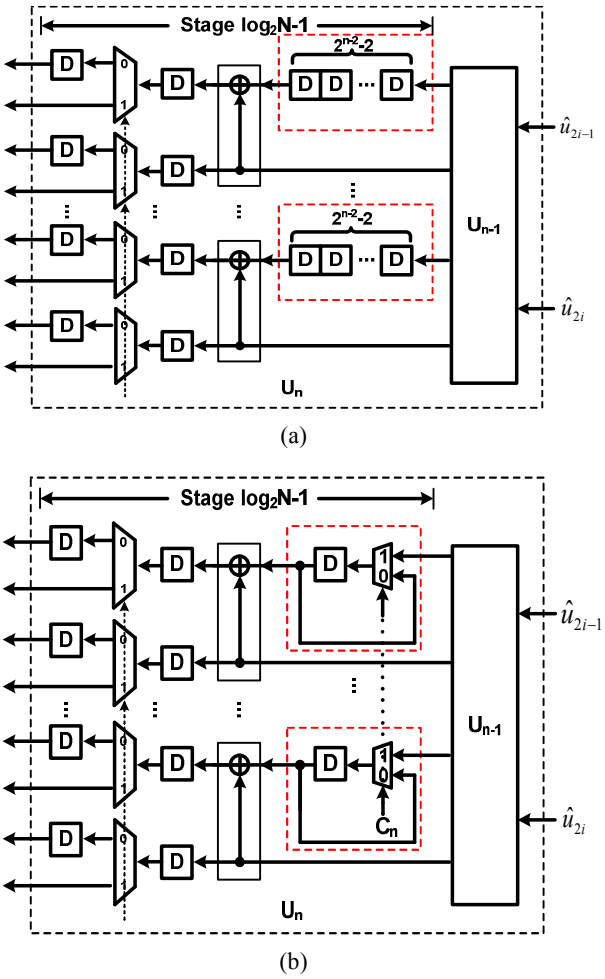

Fig. 10. (a) Conventional structure of feedback part [9], (b) Optimized structure of the feedback part.

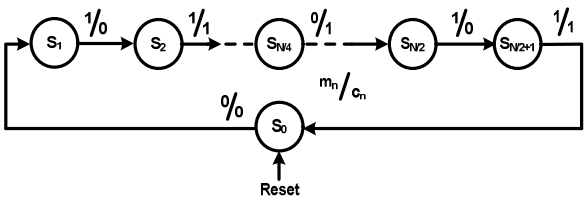

Fig. 11. Finite state machine (FSM) to generate the control signal cn for the feedback part.

a RAM would lead to scattered memory accesses requiring multiple clock cycles. To avoid lowering the throughput of the decoder, we instead store them in D-FF.

In order to reduce the complexity of the *feedback part*, the D-FF and MUX with feedback are used instead of a shift-register, as shown in Fig. 10(b). Since data lifetime is 2^{n-2}-2 clock cycles, where n = $\log_2$ N, the additional control signal c_n can be determined accordingly. Fig. 11 shows the finite state machine (FSM) to generate the

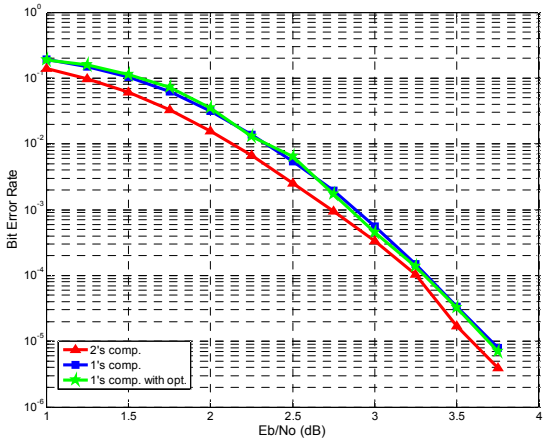

Fig. 12. BER performance for a (1024, 512) SC polar decoder using a two's complement scheme, a one's complement scheme, and a one's complement scheme with an optimal internal word length.

control signal c_n for the *feedback part*. The control signal c_n is generated by m_n, which is the MUX control signal for the n-stage in the *main frame* architecture. The initial state of FSM is S_0 and if FSM arrives into input of m_n, output c_n of FSM is 0; "c_n = 0" means the partial sum values are stored. If k = 2 and S_k assumes "m_n = 1", output c_n of FSM is 1 until the next m_n is set to 1; "c_n = 1" means the partial sum values are transferred. Until state $S_{N/4}$, there is no change in c_n. If k = {N/2, N/2+1} and S_k assumes "m_n = 1", S_1 and S_2 states are repeated.

IV. Results and Comparison

1. BER Performance

Fig. 12 shows the BER performance comparison for a (1024, 512) SC polar decoder using a two's complement scheme, a one's complement scheme with the same internal word length, and the proposed one's complement scheme with an optimum internal word length.

The quantization bits q denotes the number of bit used for the quantization of the channel LLRs and represents the maximum channel symbol magnitude. The quantization bit of channel LLR q = 5 bits was chosen for BER performance comparison, because the BER performance of fixed-point operation with q = 5 bits is sufficient to approach the BER performance of floating-point operation [10]. Therefore, the quantized LLR inputs of polar decoder were limited to the dynamic range [-16, 15]. The result shows that there is almost

Table 2. Synthesis results of different (1024, 512) SC polar decoders

Design	[10]	[13]	Proposed
Architecture	Line-based	Tree-based	Tree-based
Scheme	Two's	Two's	One's
CMOS Tech.	65 nm	45 nm	40 nm
Total gate count (XOR)	85,748†	151,292†	131,474
Critical path	-	$T_{comp}+2T_{AND}+2T_{OR}+2T_{XOR}$	$4T_{MUX}+6T_{XOR}$
Clock speed (MHz)	500	750	1,000
Decoding latency (Cycles)	2,080 (4,160 ns)	767 (1,150 ns)	1,023 (1,023 ns)
Throughput (Mbps)	246	1,000	1,000

† Gate count is calculated based on the area information in [10] and [13], unit gate area in TSMC 65 nm CMOS library and FreePDK 45 nm library.

0.125 dB degradation of 10^{-5} BER between the proposed one's complement and the two's complement scheme, as shown in Fig. 12, because the one's complement scheme offers one less quantization level than the two's complement. However, we can notice that the proposed one's complement scheme with an optimum internal word length provides almost similar BER performance to the one's complement scheme with the same internal word length in each stage.

2. Implementation Results

The proposed polar decoder was modeled in the Verilog HDL and simulated to verify its functionality. After complete verification of the design functionality, it was then synthesized using appropriate time and area constraints. Both synthesis and layout steps were carried out using the SYNOPSYS design tool and 40-nm CMOS technology. The estimated total number of XOR gates is 131,474 from the synthesized results, and the clock speed is 1 GHz for the proposed polar decoder. In addition, the proposed decoder also occupies only 0.4 mm^2 of core area using TSMC 40-nm CMOS technology.

Table 2 lists the implementation results of the reported polar (1024, 512) SC decoders. The estimated total number of XOR gates is 131,474 from the synthesized results. By applying the one's complement scheme, the critical path is reduced and the clock speed improves compared to the two's complement scheme. The critical path is $4T_{MUX}+6T_{XOR}$ and the clock speed is 1 GHz for the proposed polar decoder. In addition, the proposed decoder occupies only 0.4 mm^2 of core area using TSMC 40-nm CMOS technology. From Table 2, it can be seen that the proposed SC polar decoder can achieve a 13% reduction in gate count as well as low latency and 1 Gbps throughput.

V. CONCLUSIONS

In this paper, a novel merged processing element with a one's complement scheme, *a main frame* with an optimum internal word length, and an optimized *feedback part* for an SC polar decoder are proposed. Based on these techniques, a high-throughput low-complexity SC polar decoder architecture was presented. The estimated total number of XOR gates is 131,474 from the synthesized results, and the throughput is 1 Gbps for the proposed polar decoder. In addition, the proposed decoder requires 0.4 mm^2 of core area using 40-nm CMOS technology. Analysis shows that the proposed architecture has significant advantages with respect to both throughput and hardware complexity.

ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the NRF funded by the Ministry of Science, ICT and future Planning (2013R1A2A2A01068628) and the MSIP, Korea, under the ITRC support program (NIPA-2014-H0301-14-1042) supervised by the NIPA.

REFERENCES

[1] E. Arikan, "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, Jul. 2009.

[2] S. B. Korada, E. Sasoglu, and R. Urbanke, "Polar codes: Characterization of exponent, bounds, and constructions," *IEEE Trans. Inf. Theory,* vol. 56, no. 12, pp. 6253–6264, Dec. 2010.

[3] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Commun. Lett.,* vol. 15, no. 12, pp. 1378-1380, Dec. 2011.

[4] R. Mori and T. Tanaka, "Performance of polar codes with the construction using density evolution," *IEEE*

Commun. Lett., vol. 13, no. 7, pp. 519–521, Jul. 2009.

[5] I. Tal and A. Vardy, "List decoding of polar codes," *in Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pp. 1–5, May. 2011.

[6] K. Niu and K. Chen, "Stack decoding of polar codes," *Elect. Lett.*, vol. 48, no. 12, pp. 695–696, Jun. 2012.

[7] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, pp. 6562-6582, Oct. 2013.

[8] A. Pamuk, "An FPGA implementation architecture for decoding of polar codes," in *Proc. 8th Int. Symp. on Wireless Commun. Syst.(ICWCS)*, pp. 437-441, Nov. 2011.

[9] C. Leroux, I. Tal, A. Vardy, and W. J. Gross, "Hardware architectures for successive cancellation decoding of polar codes," in *Proc. IEEE ICASSP*, pp. 1665–1668, May. 2011.

[10] C. Leroux, A. J. Raymond, G. Sarkis, and W. J. Gross, "A semi-parallel successive-cancellation decoder for polar codes," *IEEE Trans. Signal Processing,* vol. 61, no. 2, pp. 289-299, Jan. 2013.

[11] C. Zhang, B. Yuan, and K. K. Parhi, "Reduced-latency SC polar decoder architectures," in *Proc. Int. Conf. Commun.*, pp. 3471-3475, Jun. 2012.

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

[13] B. Yuan and K. K. Parhi, "Low-latency Successive-Cancellation Polar Decoder Architectures Using 2-Bit Decoding," *IEEE Trans. Circuits and Systems I*, pp.1-14, Oct. 2013.

[14] A. Balatsoukas-Stimming, A. J. Raymod, W. J. Gross, and A. Burg, "Hardware Architecture for List Successive Cancellation Decoding of Polar Codes," *IEEE Trans. On Circuits and Systems-II*, Express Briefs, Vol. 61, No. 8, pp. 609-613, Aug. 2014

Cheolho Kim received the B.S and M.S degrees, both in Information & Communication Engineering from Inha University, Incheon, Korea, in 2012 and 2014, respectively. His research interests VLSI and SOC architecture design for digital signal processing and communication systems.

Haram Yun received the B.S degree in Information & Communication Engineering in 2014, from Inha University, Incheon, Korea, where he is currently working toward the M.S degree. His research interests VLSI and SOC architecture design for digital signal processing and communication systems.

Sabooh Ajaz received the B.S degree in electronic engineering from NED University of Engineering and Technology Karachi, in 2006 and the M.S degree from University of Wollongong, Wollongong, Australia, in 2010. Since 2011, he is currently pursuing the Ph.D. degree in Information & Communication Engineering from Inha University, Incheon, Korea. His research interests VLSI and SOC architecture design for digital signal processing and communication systems.

Hanho Lee received Ph.D. and M.S. degrees, both in Electrical & Computer Engineering, from the University of Minnesota, Minneapolis, in 2000 and 1996, respectively. In 1999, he was a Member of Technical Staff-1 at Lucent Technologies, Bell Labs, Holmdel, New Jersey. From April 2000 to August 2002, he was a Member of Technical Staff at the Lucent Technologies (Bell Labs Innovations), Allentown. From August 2002 to August 2004, he was an Assistant Professor at the Department of Electrical and Computer Engineering, University of Connecticut, USA. Since August 2004, he has been with the Department of Information and Communication Engineering, Inha University, where he is currently Professor. He was a visiting researcher at Electronics and Telecommunications Research Institute (ETRI), Korea, in 2005. From August 2010 to August 2011, he was a visiting scholar at Bell Labs, Alcatel-Lucent, Murray Hill, New Jersey, USA. His research interest includes VLSI architecture design for digital signal processing, forward error correction architectures, cryptographic systems, and communications.