Efficient Implementation of a Pseudorandom Sequence Generator for High-Speed Data Communications

Soo Yun Hwang, Gi Yoon Park, Dae Ho Kim, and Kyoung Son Jhang

A conventional pseudorandom sequence generator creates only 1 bit of data per clock cycle. Therefore, it may cause a delay in data communications. In this paper, we propose an efficient implementation method for a pseudorandom sequence generator with parallel outputs. By virtue of the simple matrix multiplications, we derive a well-organized recursive formula and realize a pseudorandom sequence generator with multiple outputs. Experimental results show that, although the total area of the proposed scheme is 3% to 13% larger than that of the existing scheme, our parallel architecture improves the throughput by 2, 4, and 6 times compared with the existing scheme based on a single output. In addition, we apply our approach to a 2×2 multiple input/multiple output (MIMO) detector targeting the 3rd Generation Partnership Project Long Term Evolution (3GPP LTE) system. Therefore, the throughput of the MIMO detector is significantly enhanced by parallel processing of data communications.

Keywords: pseudorandom sequence generator, linear feedback shift register, matrix multiplication, 3GPP LTE system, MIMO detector.

I. Introduction

Pseudorandom sequences [1] have been widely used in various fields, including communications, navigation, radar technology, cipher technologies, remote control, measurements, and industrial automation [2]. For example, pseudorandom sequences have been used in error-correcting codes [3], spread spectrum communication [4], [5], and system identification and parameter measurements [6], [7]. Other example applications are found in surface characterization and 3D scene modeling [8]. The design of a general purpose pseudorandom sequence generator has matured and has already been commercialized [9]-[11].

Pseudorandom sequences are series of 1's and 0's that lack any definite pattern and look statistically independent and uniformly distributed. The sequences are deterministic, but exhibit noise properties similar to randomness [12]. In particular, a pseudorandom sequence generator is usually made up of shift registers with feedback. By linearly combining elements from taps of the shift register and feeding them back to the input of the generator, we can obtain a sequence of much longer repeat length using the same number of delay elements in the shift register. Therefore, these blocks are also referred to as a linear feedback shift register (LFSR) [13], [14]. The length of the shift register, the number of taps, and their positions in the LFSR are important to generate pseudorandom sequences with desirable auto-correlation properties [15]. However, the output of the conventional pseudorandom sequence generator is limited to 1 bit per clock cycle. This restriction can be a bottleneck for data communications and may cause a delay. To deal with this issue, parallel architectures for a pseudorandom sequence generator

Manuscript received Sept. 4, 2009; revised Dec. 17, 2009; accepted Jan. 4, 2010.

This work was supported by the IT R&D program of MKE/KEIT, Rep. of Korea [2006-S-001-04, Development of Adaptive Radio Access and Transmission Technologies for the 4th Generation Mobile Communications].

Soo Yun Hwang (phone: +82 42 860 5569, email: syhwang@etri.re.kr), Gi Yoon Park (email: gypark@etri.re.kr), and Dae Ho Kim (email: daeho@etri.re.kr) are with the Internet Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Kyoung Son Jhang (email: sun@cnu.ac.kr) is with the Department of Computer Engineering, Chungnam National University, Daejeon, Rep. of Korea.

doi:10.4218/etrij.10.1409.0047

have been proposed [16], [17]. The approaches describe a parallel architecture implementation of a pseudorandom sequence generator for a spread-spectrum communication system and its associated switch minimization algorithm. However, the approaches are somewhat complicated in implementation and require additional memory, control blocks, and switches.

Another way to avoid delay in data communications is to generate the bit sequence in advance in a serial manner and store it in parallel format in an extra buffer before applying it to the actual data. However, this method also requires more area overhead such as memory and memory control blocks. In this paper, we propose an efficient method of implementing a pseudorandom sequence generator for high-speed data communications. Through simple matrix multiplications, we are able to derive an efficient recursive formula in a parallel form and to simply implement a pseudorandom sequence generator with multiple outputs that does not require any control logics or buffers. In addition, we apply the proposed pseudorandom sequence generator with parallel outputs to a 2×2 multiple input/multiple output (MIMO) detector to demonstrate the efficiency of our approach.

The remainder of this paper is organized as follows. In section II, we describe the key idea of the pseudorandom sequence generator with parallel outputs and an example in which to apply our scheme. We present experimental results in section III, and concluding remarks are given in section IV.

II. Parallel Pseudorandom Sequence Generator

1. Description of the Proposed Parallel Pseudorandom Sequence Generator

Figure 1 shows the structure of a conventional pseudorandom sequence generator based on LFSR with degree K. In the figure, pseudorandom sequence $c(n)$ is defined using a linear recurrence equation:

$$c(n+K)=\operatorname{mod}\left(\sum_{0\le k<K} a_k\cdot c(n+k),2\right). \tag{1}$$

The feedback taps are taken from cells corresponding to the exponents in the polynomial. Consequently, LFSR has taps from cells indexed by k such that a_k is nonzero.

The matrix formula (2) is obtained from the existing pseudorandom sequence generator with a single output shown in Fig. 1, since the pseudorandom sequence is based on linear operations [18], [19].

$$\mathbf{c}(n+1;K)=\mathbf{A}\cdot\mathbf{c}(n;K), \tag{2}$$

where vectored sequence $\mathbf{c}(m;L)$ denotes a sequence of L-dimensional row vector $[c(m)\cdots c(m+L-1)]^t$ and K-by-K

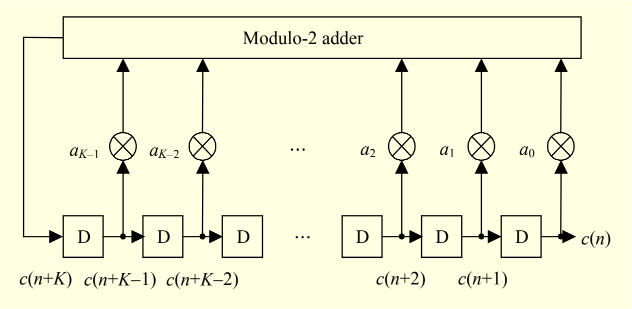


Fig. 1. Conventional pseudorandom sequence generator based on LFSR with degree K.

matrix

$$\mathbf{A}=\begin{bmatrix} 0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ & & & \vdots & \\ a_0 & a_1 & a_2 & \cdots & a_{K-1}\end{bmatrix}. \tag{3}$$

By mathematical induction, an M-shifted sample version of vectored sequences is calculated as

$$\mathbf{c}(n+M;K)=\mathbf{A}^M\cdot\mathbf{c}(n;K), \tag{4}$$

where M is any non-negative integer, and matrix multiplications are induced from multiplication and addition of a Galois field (2).

Actually, the r-th row of matrix $\mathbf{A}^M$ amounts to a mask for shifting (M+r–1) samples. Note that for M = 1, all rows except the last one degenerate into a trivial masking pattern or the selection of one element.

Figure 2 shows the architecture of the pseudorandom sequence generator with M-bit outputs, where the r-th row of M-by-K matrix $\mathbf{B}$ corresponds to the mask for shifting (r–1) samples. In particular, if M is not greater than K, the row vectors degenerate into selection patterns regardless of $\mathbf{A}$, and the additional delay is applied by simply adopting other mask patterns.

The parallel architecture has two mask stacks for each maximal length sequence generator as shown in Fig. 2. The operations of the switches for the mask stacks are determined by each element in the matrices $\mathbf{A}^M$ (K-by-K) and $\mathbf{B}$ (M-by-K). If the elements have a 1, the connection of the switches is achieved; otherwise, the switches are disconnected. In addition, the stacks at the feedback path update the states of the shift registers and depend on processing rates M, while those at the forward path transform the states into output samples with constant delays. These mask stacks are generated by the generating polynomials of the pseudorandom sequence generator.

2. Application Example

We apply our scheme to a gold sequence generator as an

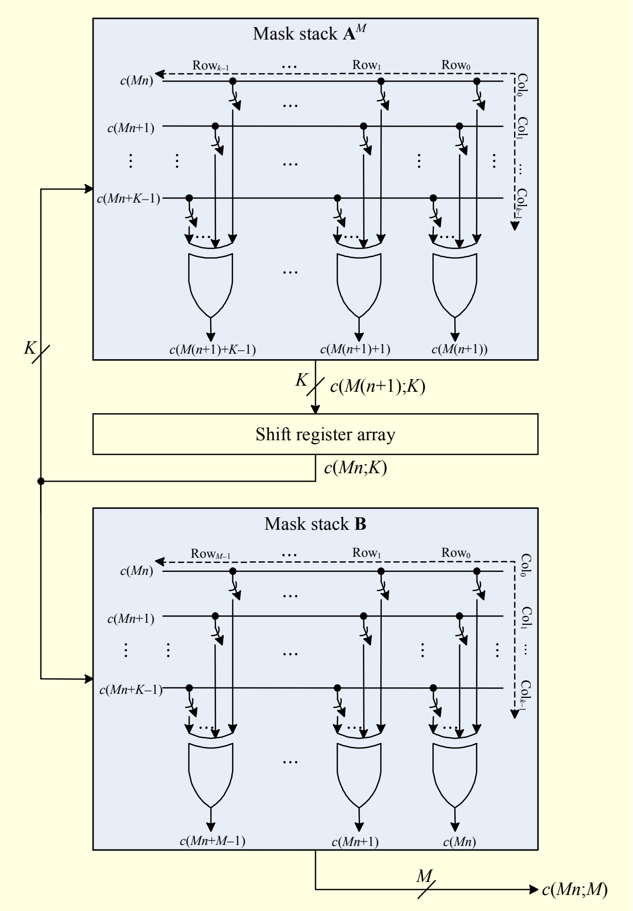


Fig. 2. Proposed architecture for the pseudorandom sequence generator with M-bit outputs.

application example. A gold sequence generator is a representative example employing a pseudorandom sequence generator. Gold sequences are a set of specific sequences found in systems employing a spread spectrum or code division multiple access (CDMA) techniques. These systems are often used in communications equipment, such as cellular phones, GPS devices, and very small aperture satellite terminals (VSATS) [20]-[22].

Figure 3 shows the structure of the existing gold sequence generator with degree N. The gold sequence $d(n)$ belongs to a family of codes with well-behaved cross-correlation properties that are constructed using a modulo-2 addition of the specific relative phases of a preferred pair of pseudorandom sequences, $x_0(n)$ and $x_1(n)$ [23].

The gold sequence generator consists of two pseudorandom sequence generators, and the existing structure has a 1 bit output $d(n)$ as shown in Fig. 3. This restriction may cause a delay in data communications. Therefore, we apply our scheme to an existing gold sequence generator with a 1 bit output to increase the data throughput.

In this paper, we implement a 6-dimensional gold sequence generator with generating polynomials $x^{25}+x^3+1$ and $x^{25}+x^3+x^2+x+1$ [24] as shown in Fig. 4, where the mask stack $\mathbf{A}^M$ includes exclusive-OR gates, and each connection line except for shift register array and the mask stack $\mathbf{B}$ is applied as a trivial case. The output of a 6-dimensional gold sequence generator with a degree of 25 is 6 bits as shown in the figure.

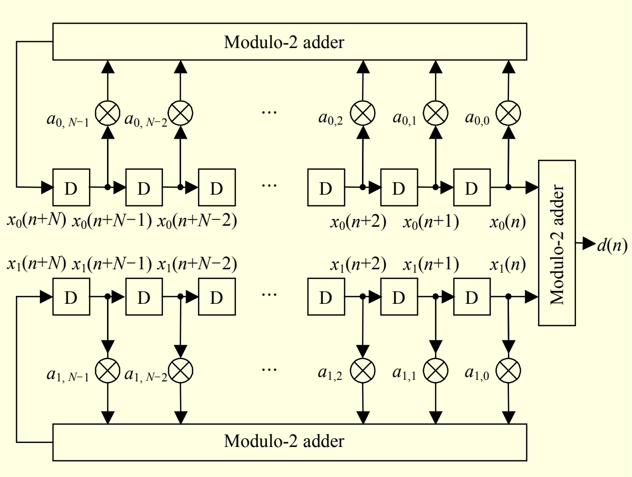


Fig. 3. Structure of the existing gold sequence generator with degree N.

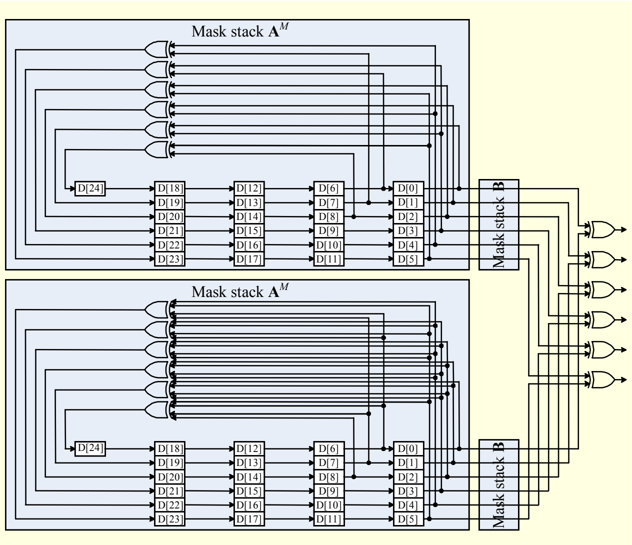


Fig. 4. Structure of a 6-dimensional gold sequence generator with a degree of 25 ($M = 6$).

III. Experiments

1. Implementation

We implement the gold sequence generators with 1, 2, 4, and 6 bit outputs (abbreviated as GSG_1, GSG_2, GSG_4, and GSG_6, respectively). The implemented gold sequence

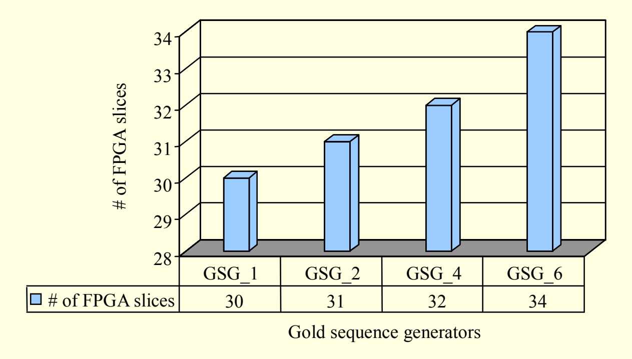


Fig. 5. Synthesis results of gold sequence generators.

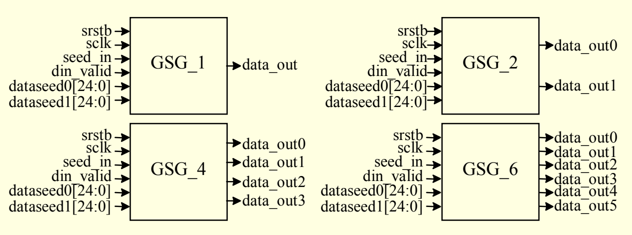


Fig. 6. Block diagrams of gold sequence generators.

generators have generating polynomials $x^{25}+x^3+1$ and $x^{25}+x^3+x^2+x+1$ [24]. The gold sequence generators are designed using a synthesizable RTL Verilog targeting XILINX FPGA (XC2VP100-6ff1704), and the XILINX design tool (ISE 8.2i) is used to measure the total area. Figure 5 shows the synthesis results of the gold sequence generators with various types of outputs.

The total areas of the gold sequence generators with parallel outputs (GSG_2, GSG_4, and GSG_6) are 3% to 13% larger than that of the gold sequence generator with a single output (GSG_1) since the gold sequence generator based on a parallel architecture requires additional exclusive-OR gates to handle the parallel processing. However, we consider this to be non-critical because the gold sequence generator occupies a very small fraction of the total FPGA chip area. Actually, the gold sequence generators take less than 1% of the total area in the case of XC2VP100.

2. Simulation

A ModelSim II simulator is utilized to measure the performance of the gold sequence generators with various types of outputs. Figure 6 shows block diagrams of each gold sequence generator for performance simulation.

In Fig. 6, GSG_1, GSG_2, GSG_4, and GSG_6 have 1 (data_out), 2 (data_out0 and data_out1), 4 (from data_out0 to data_out3), and 6 (from data_out0 to data_out5) bit outputs, respectively, while the input signals of all gold sequence generators are equal. Table 1 gives the descriptions of the input signals.

Table 1. Descriptions of input signals.

Signal name	Descriptions
srstb	System reset, active low signal
sclk	System clock
seed_in	Input valid signal for dataseed0 and dataseed1
din_valid	Output valid signal for the generated sequences
dataseed0	Initial value of upper pseudorandom sequence generator
dataseed1	Initial value of bottom pseudorandom sequence generator

The total number of bit sequences generated by each gold sequence generator is 72, chosen as an example, and the clock frequency for simulation is 100 MHz. Figure 7 shows the simulation results.

In Fig. 7, the throughputs of GSG_2, GSG_4, and GSG_6 are improved by 2, 4, and 6 times compared with the existing gold sequence generator with a single output (GSG_1), respectively. These throughputs are enhanced by changing the data transmission type from serial to parallel schemes. In the figure, GSG_1, GSG_2, GSG_4, and GSG_6 take 72, 36, 18, and 12 clock cycles to generate 72 sequences, respectively.

3. Evaluation

We apply the gold sequence generators employing the proposed scheme to a 2×2 MIMO detector based on the 3rd Generation Partnership Project Long Term Evolution (3GPP LTE) system [24]-[26] to show the efficiency of our approach.

The MIMO detector is based on the minimum mean square error-successive interference cancellation (MMSE-SIC) detection algorithm [27]. In particular, latency is one of the critical factors that decides the overall system performance in an SIC receiver [28]. In addition, recent communication systems usually adopt a high-order modulation scheme, such as 64-QAM, to increase the spectral efficiency. However, the descrambling module, which has become a mandatory building block for interference mitigation, forces system developers into serializing the demodulated bits, leading to a possible degradation of system throughput. Therefore, we employ the proposed scheme to accelerate the descrambling module of an MIMO detector. Figure 8 shows the overall structure of the MIMO detector.

The MIMO detector consists of a lattice decoder, symbol demapper, descrambler, and symbol encoder. The implemented MIMO detector has four 14-bit lattice points (LPs), eight 14-bit

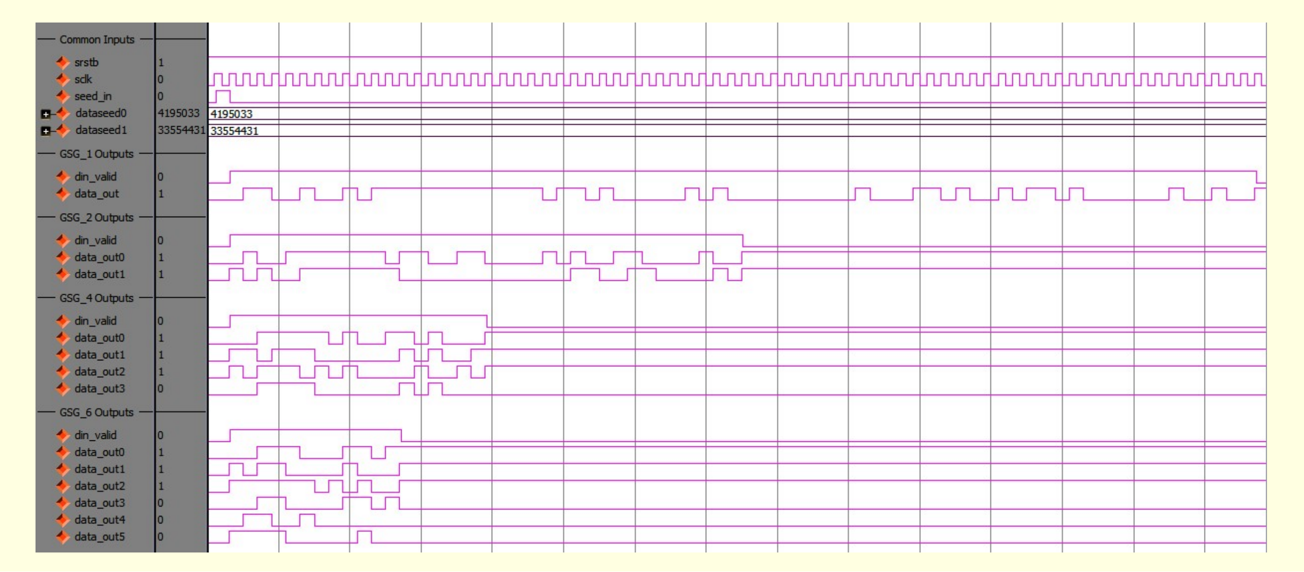


Fig. 7. Throughput for GSG_1, GSG_2, GSG_4, and GSG_6.

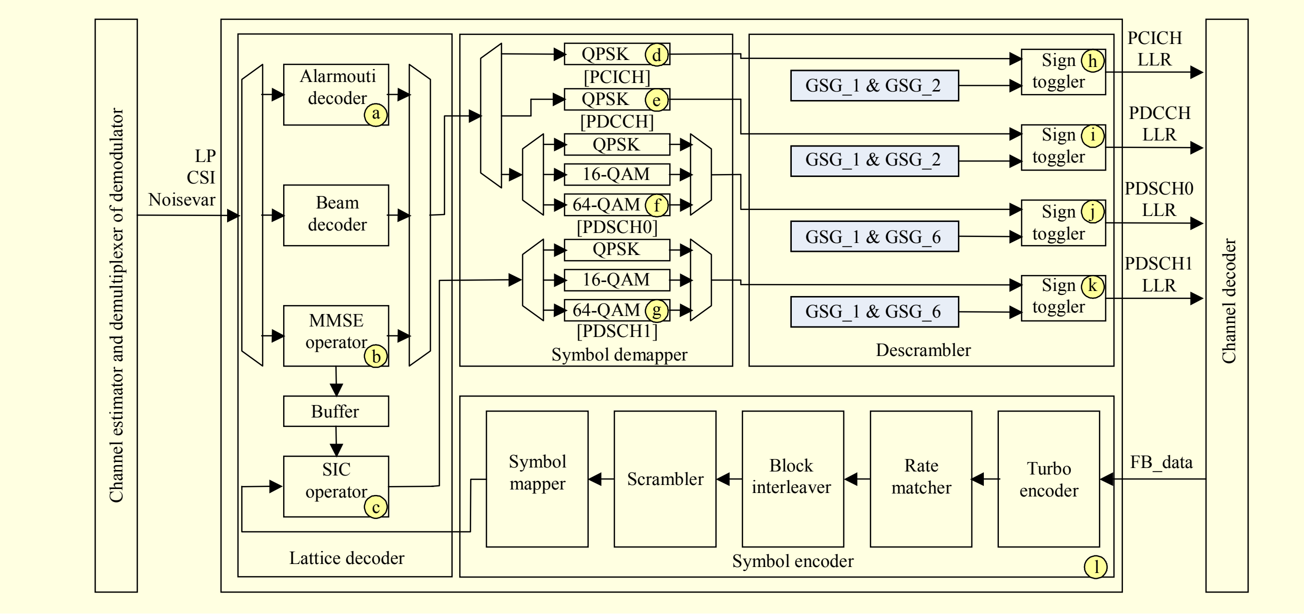


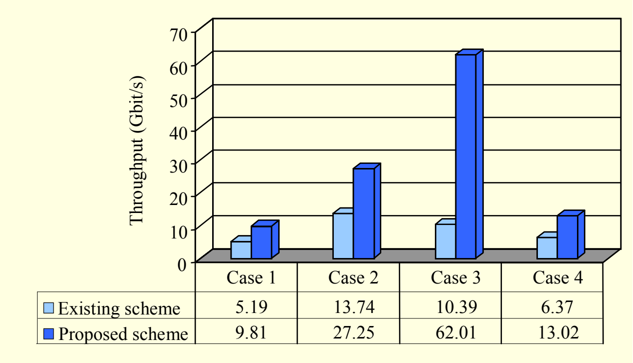
Fig. 8. Overall structure of 2×2 MIMO detector.

channel-state information (CSI), a 32-bit Noisevar, a 2-bit FB_data, 8-bit PCICH LLR, 8-bit PDCCH LLR, 8-bit PDSCH0 LLR, and 8-bit PDSCH1 LLR buses. The LP, CSI, and Noisevar are the lattice points through the physical channels, channel gain matrixes of the corresponding lattice points, and thermal noise variance, respectively, while the FB_data indicates the feedback data for SIC operations. The outputs of the MIMO detector are the descrambled bit LLRs of each channel. The gold sequence generators are used in the descrambler to toggle the scrambled bit LLRs (output signals of the symbol demapper) of a binary codeword through physical channels as shown in Fig. 8.

The control channels (PCICH and PDCCH) are based on quadrature phase-shift keying (QPSK) or 4-quadrature amplitude modulation (QAM). The modulation orders of the data channels (PDSCH0 and PDSCH1) can be QPSK, 16-QAM, or 64-QAM. Thus, the number of scrambled bit LLRs of the control channels is 2 per symbol, and that of the data channels is 2, 4, or 6 per symbol according to the modulation order. We apply GSG_2, GSG_4, and GSG_6 to the corresponding modulation orders, respectively. Next, we compare the proposed scheme with the existing method based

Table 2. Simulation configurations for each channel type.

Case	Channel type	Data path	NData	# of transactions
1	PCICH	a→d→h	288	2,400
2	PDCCH	a→e→i	4,800	2,400
3	PDSCH0	b→f→j	33,120	2,400
4	PDSCH1	l→c→g→k	33,120	2,400



	Case 1	Case 2	Case 3	Case 4
□ Existing scheme	5.19	13.74	10.39	6.37
■ Proposed scheme	9.81	27.25	62.01	13.02

Fig. 9. Comparison of throughput results.

on the gold sequence generator with a single output (GSG_1) in throughput. When using GSG_1, additional buffers and control logics are required to generate a scrambled bit LLR in advance in a serial manner and store it in a parallel format within an extra buffer before applying it to the actual data. Gold sequence generators with parallel outputs such as GSG_2, GSG_4, or GSG_6, however, do not need any area overhead such as extra buffers or control logics.

In Fig. 8, the data paths of the control channels are fixed, while the data paths of the data channels are diverse according to the operation mode, such as SFBC, beamforming, single-user MIMO or multi-user MIMO, and the modulation order, such as QPSK, 16-QAM, or 64-QAM. Therefore, we use the 64-QAM-based single-user MIMO mode with the longest latency among operation modes for performance comparisons of the data channels. Table 2 summarizes the simulation configurations of each channel type for the experiments. In the Table 2, NData is the unit of data transactions.

Figure 9 shows a comparison of the results. In this study, throughput is defined as

$$\text{Throughput} = N_{\text{transactions}} * N_{\text{NData}} * N_{\text{bit}} / T,$$

where $N_{\text{transactions}}$ is the number of transactions, N_{NData} denotes the NData, N_{bit} denotes the data bit width, and T denotes the completion time of a data transmission.

In cases 1 and 2, our method improves the throughput by about 2 times compared with the existing method. Also, the proposed scheme enhances the throughput by about 6 times compared with the existing scheme in case 3. The throughputs of cases 1, 2, and 3 are almost the same as those in section III.2 since the latencies of a lattice decoder and symbol demapper are considerably shorter than that of a descrambler. However, in case 4, our approach improves the throughput by only 2 times compared with the existing method even though the modulation order is 64-QAM because a symbol encoder for SIC operations has a long latency. Therefore, there is some degradation of the performance improvements.

As a result, we conclude that the throughput of the MIMO detector is remarkably improved through parallel processing of data communications.

IV. Conclusion

In this paper, we presented a method to improve the structure of a pseudorandom sequence generator for high-speed data communications. With the simple matrix manipulations, we can obtain efficient recursive formulas in parallel form as well as implement parallel-structure-based pseudorandom sequence generators that do not require any control logics or memories. Experimental results show that although the total area of the proposed scheme is 3% to 13% larger than that of the existing scheme, our method improves the throughput by 2, 4, and 6 times compared with the existing method based on a single output.

We also applied our scheme to a 2×2 MIMO detector based on the 3GPP LTE system. The performance simulation results demonstrate that the throughput of the MIMO detector is significantly improved by parallel processing of the data communications. We expect that it would be very useful to apply our scheme to data communication systems that require high throughput with low latency.

References

[1] Z.G. Xiao, *Pseudo-Random Sequence and Its Applications*, Beijing, China: Nat. Defence Ind., 1985.

[2] L. Xu and X. Li, "Dual-Channel Pseudorandom Sequence Generator with Precise Time Delay Between Its Two Channels," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 12, Dec. 2008, pp. 2880-2884.

[3] C.H. Yen and B.F. Wu, "An Error-Correcting Stream Cipher Design with State-Hopping Architecture," *J. Chin. Inst. Eng.*, vol. 28, no. 1, 2005, pp. 9-16.

[4] X.G. Wang et al., "Spread-Spectrum Communication Using Binary Spatiotemporal Chaotic Codes," *Phys. Lett. A*, vol. 334, no. 1, Jan. 2005, pp. 30-36.

[5] H.J. Kim et al., "PN Sequence Generation from 2-D Array of Shift Registers," *ETRI J.*, vol. 27, no. 3, June 2005, pp. 273-279.

[6] T. Johnsen et al., "Simultaneous Use of Multiple Pseudo Random Noise Codes in Multistatic CW Radar," *Proc. IEEE Nat. Radar Conf.*, 2004, pp. 266-270.

[7] D.K. Rollins et al., "A Quantitative Measure to Evaluate Competing Designs for Non-linear Dynamic Process Identification," *Can. J. Chem. Eng.*, vol. 84, no. 4, 2006, pp. 459-468.

[8] H.J.W. Spoelder et al., "Some Aspects of Pseudo Random Binary Array-Based Surface Characterization," *IEEE Trans. Instrum. Meas.*, vol. 49, no. 6, Dec. 2000, pp. 1331-1336.

[9] R. Shaltiel and C. Umans, "Simple Extractors for All Min-entropies and a New Pseudorandom Generator," *Proc. Annu. Symp. Found. Comput. Sci.*, 2001, pp. 648-657.

[10] A.H. Tan and K.R. Godfrey, "The Generation of Binary and Near-Binary Pseudorandom Signals: An Overview," *Proc. IEEE Instrum. Meas. Technol. Conf.*, vol. 2, 2001, pp. 766-771.

[11] J. Szczepanski et al., "Biometric Random Number Generators," *Comput. Secur.*, vol. 23, no. 1, Feb. 2004, pp. 77-84.

[12] P. Alfke, "Efficient Shift Registers, LFSR Counters, and Long Pseudo-Random Sequence Generators," *Application Note, Xilinx Corp.*, Aug. 1995.

[13] *Gold Code Generator Reference Design*, Altera Application Note 295, Mar. 2003.

[14] F. Principe et al., "Rapid Acquisition of Gold Codes and Related Sequences Using Iterative Message Passing on Redundant Graphical Models," *Proc. Int. Conf. Military Commun.*, 2006, pp. 1-7.

[15] X.D. Lin and K.H. Chang, "Optimal PN Sequence Design for Quasisynchronous CDMA Communication Systems," *IEEE Trans. Comm.*, vol. 45, no. 2, Feb. 1997, pp. 221-226.

[16] M. Lowy and K. Anne, "A High-Speed, Low-Power Spread Spectrum Code Generator," *Proc. Int. Symp. MWSCAS*, vol. 1, 1994, pp. 23-26.

[17] M. Lowy, "Low Power Spread Spectrum Code Generator Based on Parallel Shift Register Implementation," *Proc. Int. Symp. Low Power Electron.*, 1994, pp. 22-23.

[18] R. Gold, "Optimal Binary Sequences for Spread Spectrum Multiplexing," *IEEE Trans. Inf. Theory*, 1967, pp. 619-621.

[19] R. Gold, "Maximal Recursive Sequences with 3-Valued Recursive Cross-Correlation Functions," *IEEE Trans. Inf. Theory*, 1968, pp. 154-156.

[20] C.L. Lu et al., "10-Gb/s CMOS Ultra High-Speed Gold-Code Generator Using Differential-Switches Feedback," *Proc. Int. Conf. Microwave Integrated Circuit*, 2007, pp. 239-242.

[21] A.N. Akansu and R. Poluri, "Walsh-Like Nonlinear Phase Orthogonal Codes for Direct Sequence CDMA Communications," *IEEE Trans. Signal Processing*, vol. 55, 2007, pp. 3800-3806.

[22] D. Shiung and J.F. Chang, "Enhancing the Capacity of DS-CDMA System Using Hybrid Spreading Sequences," *IEEE Trans. Comm.*, vol. 52, 2004, pp. 372-375.

[23] P. Markovic and M. Markovic, "FPGA/VLSI Implementation Analysis of PN Sequence Generator for Direct Sequence Spread Spectrum Systems," *Proc. Int. Conf. TELSIKS*, 1999, pp. 574-576.

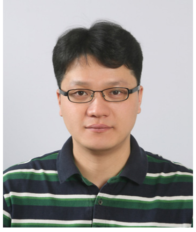
[24] 3GPP TS 36.211, *Physical Channels and Modulation (Release 8)*, 2007.

[25] 3GPP TS 36.201, *LTE Physical Layer-General Description (Release 8)*, 2007.

[26] 3GPP TS 36.212, *Multiplexing and Channel Coding (Release 8)*, 2007.

[27] K.C. Lee et al., "Optimal Lattice-Reduction Aided Successive Interference Cancellation for MIMO Systems," *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, 2007, pp. 2438-2443.

[28] C.H. Tsai et al., "Hybrid MMSE and SIC for Multiuser Detection," *Proc. IEEE Int. Conf. Vehicular Technol.*, vol. 3, 2001, pp. 1779-1783.



Soo Yun Hwang received the BS degree in computer engineering from Hannam University, Daejeon, Korea, in 2002, and the MS and PhD degrees in computer engineering from Chungnam National University, Daejeon, Korea, in 2004 and 2008, respectively. Since 2006, he has been working at ETRI, Daejeon, Korea, where he currently works in a high-speed modem research team as a senior member of engineering staff. He has participated in various projects, including the development of a flexible on-chip-network-based system-on-a-chip platform targeting the H.264 decoder, a user equipment modem apparatus based on the 3rd Generation Partnership Project Long Term Evolution, and adaptive radio access and transmission technologies for the 4th generation mobile communications. His current research interests include CAD for VLSI, system-on-a-chip design methodology, on-chip communication architecture, and high-speed modem designs.

Gi Yoon Park received the BS and MS degrees in electronics and electrical engineering from POSTECH, Pohang, Korea, in 2000 and 2003, respectively. Since 2003, he has been employed at ETRI, Daejeon, Korea, where he is a member of engineering staff. His research interests are in the areas of space-time codes and signal processing for digital communication.

Dae Ho Kim received the BS and MS degrees in electronics engineering from Kyongpook National University, Daegu, Korea, in 1989 and 1991, respectively, and the PhD degree in electronics engineering from Chungnam National University, Daejeon, Korea, in 2005. He is currently a team leader of the High-Speed Modem Research Team with ETRI, Daejeon, Korea. His research interests include broadband wireless communications, DSP and VLSI applications, and multimedia signal processing.

Kyoung Son Jhang graduated from Seoul National University in 1986 with a Bachelor of Computer Engineering degree and received his MS and PhD degrees at the same university in 1988 and 1995, respectively. Upon graduation, he joined Hannam University, Daejeon, Korea, as a faculty member in 1996. He then moved to Chungnam National University where he still works as a professor teaching systems programming and digital hardware design. Currently, his major interests include fault-tolerant hardware design, electronic design automation, and digital system design.