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.
have been proposed \cite{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:

\[
0 \leq n < K, \quad c(n + K) = \text{mod} \left( \sum_{0 \leq k < K} a_k \cdot c(n + k), 2 \right).
\]

(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 \cite{18, 19}.

\[
c(n + 1, K) = A \cdot c(n; K),
\]

(2)

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

\[
A = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & & & & \\
0 & a_0 & a_1 & a_2 & \cdots & a_{K-1}
\end{bmatrix}
\]

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

\[
c(n + M; K) = A^M \cdot c(n; K),
\]

(3)

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 \( A^M \) 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 \( B \) 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 \( A \), 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 \( A^M \) (\( K \)-by-\( K \)) and \( B \) (\( M \)-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
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 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 $a(n)$ 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, $x_{k}(n)$ and $x_{l}(n)$ [23].

The gold sequence generator consists of two pseudorandom sequence generators, and the existing structure has a 1 bit output $a(n)$ as shown in Fig. 3. This restriction may cause a delay in data communications. Therefore, we apply our scheme to the 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 $x^2 + x + 1$ and $x^3 + x^2 + x + 1$ [24] as shown in Fig. 4, where the mask stack $A^M$ includes exclusive-OR gates, and each connection line except for shift register array and the mask stack $B$ 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.

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
The gold sequence 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.

### Table 1. Descriptions of input signals.

<table>
<thead>
<tr>
<th>Signal name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>srstb</td>
<td>System reset, active low signal</td>
</tr>
<tr>
<td>sclk</td>
<td>System clock</td>
</tr>
<tr>
<td>seed_in</td>
<td>Input valid signal for dataseed0 and dataseed1</td>
</tr>
<tr>
<td>din_valid</td>
<td>Output valid signal for the generated sequences</td>
</tr>
<tr>
<td>dataseed0</td>
<td>Initial value of upper pseudorandom sequence generator</td>
</tr>
<tr>
<td>dataseed1</td>
<td>Initial value of bottom pseudorandom sequence generator</td>
</tr>
</tbody>
</table>

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
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 on physical channels as shown in Fig. 8.
Table 2. Simulation configurations for each channel type.

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

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

\[
\text{Throughput} = \frac{N_{\text{transactions}} \cdot N_{\text{NData}} \cdot N_{\text{bit}}}{T},
\]

where \(N_{\text{transactions}}\) is the number of transactions, \(N_{\text{NData}}\) denotes the NData, \(N_{\text{bit}}\) 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 generators 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


Soo Yun Hwang received the BS degree in computer engineering from Hanyang 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 Kyungpook 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.