PAPER

VLSI Implementation of an Interference Canceller Using Dual-Frame Processing for OFDM-IDMA Systems

Shingo YOSHIZAWA[†a)], *Member*, **Mai NOZAKI**[†], *Nonmember*, *and* **Hiroshi TANIMOTO**[†], *Member*

SUMMARY Due to increasing demand for machine-to-machine (M2M) communication, simultaneous connections for many terminals are requested for current wireless communication systems. Interleave division multiple access (IDMA) has superior multiuser detection performance and attains high data transmission efficiency in multiuser communications. This paper describes the VLSI implementation of an interference canceller for OFDM-IDMA systems. The conventional architecture decreases a throughput in pipeline processing due to wait time occurring in interleave and deinterleave memory units. The proposed architecture adopts dual-frame processing to solve the problem of the wait time and achieves a high utilization ratio in pipeline stage operation. In the implementation results, the proposed architecture has reduced circuit area and power consumption by 25% and 41% for BPSK demodulation and 33% and 44% for QPSK demodulation compared with the conventional architecture on the same throughput condition.

key words: *OFDM-IDMA, interference canceller, VLSI architecture*

1. Introduction

In recent wireless communication, machine to machine (M2M) communication that machines autonomously exchange information via communication networks has attracted attention and become an essential technology to construct information infrastructures such as smart cities and smart grids [1], [2]. Wireless communication systems request simultaneous connections for many terminals to cope with thus increasing communication throughput.

Multiple access, simultaneously connecting multiple user terminals, is classified into orthogonal and non-orthogonal schemes in radio resource allocation. Orthogonal multiple access that each user separates radio resources by time and frequency units as frequency division multiple access (FDMA), time division multiple access (TDMA), and orthogonal frequency-division multiple access (OFDMA), requires scheduling where control information is exchanged among base stations and user terminals. In M2M communication supporting connections of many terminals, orthogonal multiple access might decrease an effective transmission rate due to increasing control information. On the other hand, non-orthogonal multiple access as code division multiple access (CDMA) and interleave division multiple access (IDMA) [3]–[5] does not require the scheduling. Since IDMA is superior to CDMA in multiuser detection (detecting desired signals from interference and noise) on the condition of connecting many users [5], it is thought to be promising in wireless M2M communication. As for practical studies, throughput evaluation of IDMA in cellular communication has been reported in [6]. We have studied hardware development and outdoor experiment of IDMA systems [7], [8]. Moreover, OFDM-IDMA combining IDMA and OFDM can easily perform channel equalization in the frequency domain, which is robust with multipath interference [9].

This paper describes VLSI implementation of an interference canceller that is the most important part in OFDM-IDMA processing. The interference canceller separates users' signals by iterative processing, whose computational cost is proportional to the numbers of users and iterations. As a related work, pipeline and parallel architecture is adopted in an interference canceller, which is used in a partitioned spreading CDMA (PS-CDMA) receiver [10]. Since the interference cancellation of PS-CDMA is similar to that of IDMA, we treat it as conventional architecture. The conventional architecture decreases a throughput processing due to wait time occurring in interleave and deinterleave memory units, whose utilization ratio in circuit operation is half. The proposed architecture adopts dual-frame processing to solve the problem of the wait time and achieves a high utilization ratio, which can provide smaller circuit scale and power consumption than the conventional architecture when comparing on the same throughput condition.

The paper is organized as follows: Sect. 2 explains OFDM-IDMA system. Section 3 describes an interleaver design used in OFDM-IDMA system. Section 4 discusses VLSI architecture of an interference canceller. The VLSI implementation results of conventional and proposed architectures are reported in Sect. 5. Section 6 summarizes our work.

2. OFDM-IDMA System

2.1 Transmitter and Receiver

Block diagrams of OFDM-IDMA transmitter and receiver and uplink channel are shown in Fig. 1. In the transmitter in Fig. 1(a), the coded bit sequence $\boldsymbol{c}_k=[c_k(0),c_k(1),...,c_k(n_\mathrm{c}),...,c_k(N_\mathrm{c}-1)] \in \{0,1\}^{N_\mathrm{c}}$ $(0 \le n \le N_\mathrm{c}-1)$ is generated by N_rep-times repetition codes with a coding rate $R_\mathrm{rep}(=1/N_\mathrm{rep})$ from the information bit sequence $\boldsymbol{b}_k=[b_k(0),b_k(1),...,b_k(n_\mathrm{b}),...,b_k(N_\mathrm{b}-1)] \in \{0,1\}^{N_\mathrm{b}}$ $(0 \le n_\mathrm{b} \le$

Manuscript received June 2, 2014.
Manuscript revised October 27, 2014.

[†]The authors are with the Department of Electrical and Electronic Engineering, Kitami Institute of Technology, Kitami-shi, 090-8507 Japan.
a) E-mail: yosizawa@mail.kitami-it.ac.jp

DOI: 10.1587/transfun.E98.A.811

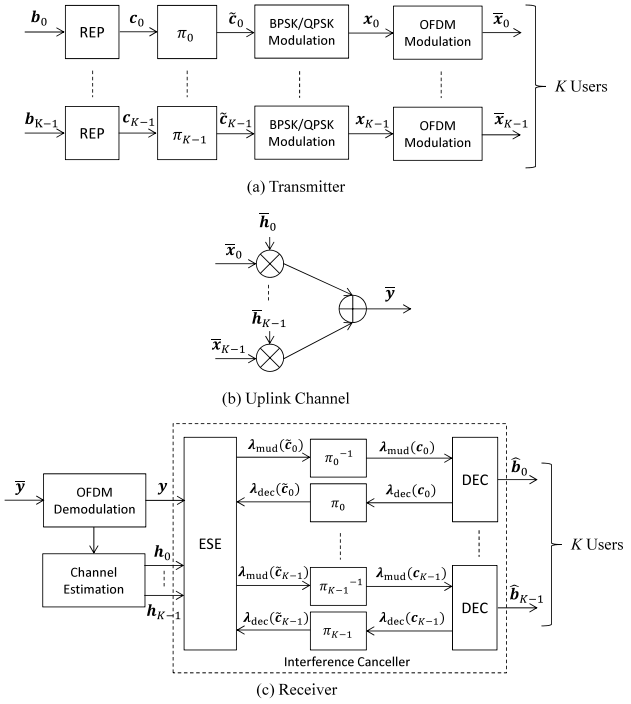

Fig. 1 Block diagrams of OFDM-IDMA transmitter and receiver and uplink channel.

$N_b - 1$) in the repetition coder (REP). n_b is a bit number in the information bit sequence and $N_c(=N_b N_{rep})$ denotes the number of coded bits, and k ($0 \le k \le K-1$) indicates a user number. The transmit symbol sequence $\tilde{\boldsymbol{c}}_k$ is generated by changing the order of the coded bits by adopting a user specific interleaving pattern π_k. The BPSK/QPSK symbol sequence $\boldsymbol{x}_k=[x_k(0),x_k(1),...,x_k(n),...,x_k(N-1)]$ ($0 \le n \le N-1$) is generated by the BPSK/QPSK modulator. n is a symbol number and $N(=N_c/N_{bps})$ denotes the number of symbols. N_{bps} means the number of bits per symbol, i.e., 1 for BPSK and 2 for QPSK. After the OFDM modulation, the OFDM-IDMA signals $\bar{\boldsymbol{x}}_k$ are transmitted by adding a cyclic prefix (CP).

In the uplink channel in Fig. 1(b), the OFDM-IDMA transmitted signals $\bar{\boldsymbol{x}}_k$ are multiplied by the channel coefficient $\bar{\boldsymbol{h}}_k$ for each user. The OFDM-IDMA received signals $\bar{\boldsymbol{y}}$ are modeled by summing the channel affected signals of all users. In the receiver in Fig. 1(c), the frequency-domain received signals $\boldsymbol{y}$ are obtained by the removal of CPs and the OFDM demodulation. From the training signals located at the head of OFDM-IDMA frame, the channel coefficients of all users $\boldsymbol{h}_0,\boldsymbol{h}_1,...,\boldsymbol{h}_{K-1}$ are computed by the frequency-domain channel estimation. The interference canceller consists of the elementary signal estimator (ESE) and the parallel iterative decoders (DECs) with K users [11]. The number of iterations is given by N_{iter}. The ESE computes the extrinsic value $\lambda_{mud}(\tilde{\boldsymbol{c}}_k)$ for each user by using the received signals $\boldsymbol{y}$ and the channel coefficients of all users $\boldsymbol{h}_0,\boldsymbol{h}_1,...,\boldsymbol{h}_{K-1}$. The DEC accepts the extrinsic value $\lambda_{mud}(\boldsymbol{c}_k)$ after deinterleaving π_k^{-1} and outputs the received bit sequence $\hat{\boldsymbol{b}}_k$ by decoding of repetition codes with BPSK/QPSK demodulation. The reliable extrinsic value $\lambda_{dec}(\boldsymbol{c}_k)$ converts to $\lambda_{dec}(\tilde{\boldsymbol{c}}_k)$ by interleaving π_k. The converted data $\lambda_{dec}(\tilde{\boldsymbol{c}}_k)$ is used as the input of ESE at the next iteration step.

2.2 Interference Canceller

In the receiver of Fig. 1(c), the received signal $\boldsymbol{y}$ after the OFDM demodulation can be expressed by the desired and interference components as

$$y(n) = h_k(n)x_k(n) + I_k(n), \tag{1}$$

where $I_k(n)$ denotes the sum of the noise and interference components in the n-th symbol, expressed as

$$I_k(n) = \sum_{l=0,l\neq k}^{K-1} h_l(n)x_l(n) + z(n), \tag{2}$$

where $z(n)$ is a sample of additive white Gaussian noise (AWGN). The interference cancellation by iterative processing is given by the following procedure:

(A) Generate the inputs of the interference canceller by interleaving the extrinsic values of the decoder outputs.

$$\lambda_{dec}(\tilde{\boldsymbol{c}}_k) = \pi_k(\lambda_{dec}(\boldsymbol{c}_k)), \tag{3}$$

where $\lambda_{dec}(\boldsymbol{c}_k) = \boldsymbol{0}$ is applied in case of the first iteration step.

(B) Compute the expectation and variance values of the desired signals as $E_k(n)$ and $V_k(n)$ and their summations as $\bar{E}(n)$ and $\bar{V}(n)$. For BPSK demodulation,

$$E_k(n) = h_k(n)\tanh(\lambda_{dec}(\tilde{c}_k(n))/2) \tag{4}$$

$$V_k(n) = |h_k(n)|^2 - |E_k(n)|^2 \tag{5}$$

$$\bar{E}(n) = \sum_{k=0}^{K-1} E_k(n) \tag{6}$$

$$\bar{V}(n) = \sum_{k=0}^{K-1} V_k(n). \tag{7}$$

For QPSK demodulation, the symbol index n is divided into odd and even numbers of $2m$ and $2m-1$ ($0 \le m \le N/2-1$), their equations are given by

$$E_k(m) = h_k(m)\{\tanh(\lambda_{dec}(\tilde{c}_k(2m))/2) + j\tanh(\lambda_{dec}(\tilde{c}_k(2m+1))/2)\} \tag{8}$$

$$V_k(m) = (1 - \mathrm{Re}[E_k(m)]^2) + j(1 - \mathrm{Im}[E_k(m)]^2) \tag{9}$$

$$\bar{E}(m) = \sum_{k=0}^{K-1} E_k(m) \tag{10}$$

$$\bar{V}(m) = \sum_{k=0}^{K-1} V_k(m), \tag{11}$$

where $\mathrm{Re}[\cdot]$ and $\mathrm{Im}[\cdot]$ indicate extracting real and imaginary parts.

(C) Compute the outputs of the ESE as $\lambda_{mud}(\tilde{\boldsymbol{c}}_k)$ by log likelihood ratios (LLRs). For BPSK demodulation,

$$\lambda_{\mathrm{mud}}(\tilde{c}_k(n)) = 2h_k^*(n)\frac{y(n) - \bar{E}(n) + E_k(n)}{\bar{V}(n) - V_k(n) + \sigma^2}, \quad (12)$$

where σ^2 denotes the average noise power in the received signals. For QPSK demodulation,

$$E'_k(m) = x_k(m) - \bar{E}(m) + E_k(m) \quad (13)$$

$$V'_k(m) = \bar{V}(m) - V_k(m) + (1 + j)\sigma^2 \quad (14)$$

$$\lambda_{\mathrm{mud}}(\tilde{c}_k(2m)) = \frac{2|h_k(m)|^2 \cdot \mathrm{Re}[E'_k(m)h_k^*(m)]}{\mathrm{Re}[h_k^*(m)]^2\mathrm{Re}[V'_k(m)] + \mathrm{Im}[h_k^*(m)]^2\mathrm{Im}[V'_k(m)]} \quad (15)$$

$$\lambda_{\mathrm{mud}}(\tilde{c}_k(2m+1)) = \frac{2|h_k(m)|^2 \cdot \mathrm{Im}[E'_k(m)h_k^*(m)]}{\mathrm{Re}[(h_k^*(m)]^2\mathrm{Im}[V'_k(m)] + \mathrm{Im}[(h_k^*(m)]^2\mathrm{Re}[V'_k(m)]}, \quad (16)$$

where the outputs of $\lambda_{\mathrm{mud}}(\tilde{c}_k(2m))$ and $\lambda_{\mathrm{mud}}(\tilde{c}_k(2m+1))$ are merged by the original order of symbol index m.

(D) Generate the DEC inputs as $\lambda_{\mathrm{mud}}(\boldsymbol{c}_k)$ by deinterleaving.

$$\lambda_{\mathrm{mud}}(\boldsymbol{c}_k) = \pi_k^{-1}(\lambda_{\mathrm{mud}}(\tilde{\boldsymbol{c}}_k)) \quad (17)$$

(E) Compute the DEC outputs as $\hat{\boldsymbol{b}}_k$ and the extrinsic values for the next iteration as $\lambda_{\mathrm{dec}}(\boldsymbol{c}_k)$.

$$T_k(n_\mathrm{b}) = \sum_{n_\mathrm{rep}=0}^{N_\mathrm{rep}-1} \lambda_{\mathrm{mud}}(c_k(n_\mathrm{rep} + N_\mathrm{rep}n_\mathrm{b})) \quad (18)$$

$$\hat{b}_k(n_\mathrm{b}) = f_{\mathrm{Demod}}(T_k(n_\mathrm{b})) \quad (19)$$

$$\lambda_{\mathrm{dec}}(\boldsymbol{c}_k) = \{T_k(0), ..., T_k(n_\mathrm{b}), ..., T_k(N_\mathrm{b} - 1)\} - \lambda_{\mathrm{mud}}(\boldsymbol{c}_k), \quad (20)$$

where $n = n_\mathrm{rep} + N_\mathrm{rep}n_\mathrm{b}$ $(0 \le n_\mathrm{b} \le N_\mathrm{b} - 1)$. f_{Demod} generates binary data by BPSK/QPSK demodulation. By repeating the aforementioned procedure, the decoder output sequence $\hat{\boldsymbol{b}}_k$ gradually approaches to the information bit sequence $\boldsymbol{b}_k$.

3. Interleaver

3.1 Random Interleaver

Data sorting of interleaver is realized by the use of memory on hardware. The behavior of interleaver and deinterleaver is illustrated in Fig. 2. First, the data sequence of {A,B,C,D} are stored in the interleaver memory by the write address order of {0,1,2,3}. The sorting of {D,B,C,A} is performed by the read address order of {3,1,2,0}. The sorting of deinterleaver is given by the write address order of {3,1,2,0} and the read address order of {0,1,2,3}. The interleaver and deinterleaver are requested to memorize their address sequences for all users because the sorting patterns are user specific. Memorizing address sequences costs KN memory words if their sequences are randomly generated.

3.2 Algebraic Interleaver

An algebraic interleaver has been presented to reduce the amount of memorized interleaving patterns [12]. The algebraic interleaver generates interleave patterns by algebraic operation according to a user specific constant. The algebraic operation of the Takeshita-Costello method [12] is given by the following equation:

$$a(n) = \frac{S_k n(n+1)}{2} (\mathrm{mod}\ N) \quad (0 \le n \le N - 1), \quad (21)$$

where N is given by the power of 2. S_k is a user specific constant, randomly generated from odd numbers. The Takeshita-Costello method can reduce memory words from KN to K compared with the random interleaver. However, the Takeshita-Costello method might degrade communication performance due to the low-spread property that the similar address sequence patterns are observed among users. Hence, we employ multiple stage algebraic interleaver shown in Fig. 3. The output of the first stage interleaver $a_1(n)$ is connected to the input of the second stage to scramble the interleaved patterns. The number of user specific constants increases to the two of $S_{1,k}$ and $S_{2,k}$, however its increase is small compared with the random interleaver.

The optimal number of stages in algebraic interleaver depends on communication specifications and channel conditions. As one of examples, we append the simulation results of random and algebraic interleavers in Appendix.

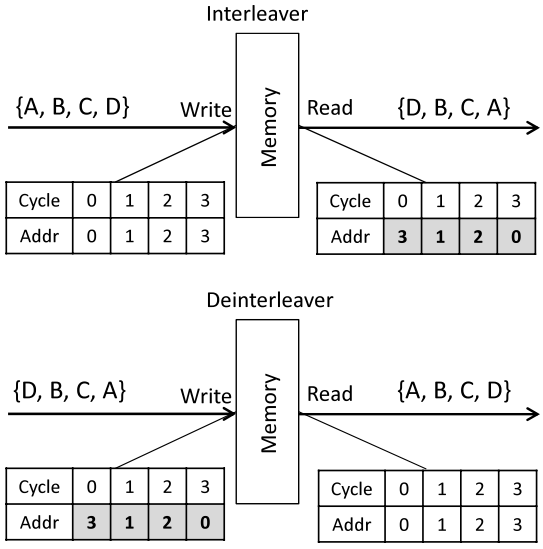

Fig. 2 Behavior of interleaver and deinterleaver.

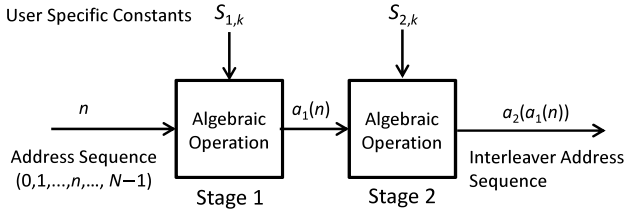

Fig. 3 Multiple stage algebraic interleaver.

4. VLSI Design

4.1 IDMA Decoder

A circuit structure of an IDMA decoder at the inside of an OFDM-IDMA receiver is illustrated in Fig. 4. After storing the received signals $\boldsymbol{y}$ and channel coefficients $\boldsymbol{h}$ in memory, the IDMA decoder performs iterative processing by parallel interference cancellers of all users. The block of "Interference Canceller" includes the processing of "ESE" and "DEC" in Fig. 1. The interference canceller executes pipeline processing by a block unit of N symbols. Since the summations of means and variances in (6) and (7) ((10) and (11)) cannot be computed at the inside of the interference canceller, the "Mean/Var Summation" block takes mean and variance values of all users and delivers their summations to the interference cancellers.

4.2 Conventional Architecture

Conventional architecture base on pipeline processing that has been used in PS-CDMA system [10] is illustrated in Fig. 5. All circuit blocks operate by pipeline processing and the feedback datapath of $\lambda_{\mathrm{dec}}(\boldsymbol{c}_k)$ realizes iterative processing. We have modified the mean and variance computation block ("Mean/Var") and the LLR computation block ("LLR") for IDMA. Also, we use the algebraic interleaver, where the address generation is executed in the "Algebraic Operation" block. When the outputs of the extrinsic value computation block "Ext. Calc" go back to the input of the interleaver memory ("Int. RAM"), the next iterative process is started.

The timing chart of the conventional architecture is illustrated in Fig. 6. The output data of "Extrinsic Calculation" becomes the input data of "Interleaver" at the next iteration step. Since all the blocks execute pipeline processing, a utilization ratio should be almost one on the assumption that those blocks process valid data anytime. However, the utilization ratio of conventional architecture is about half, where all the blocks are forced to wait their operations until the next iteration step comes.

4.3 Improvement of Utilization by Dual-Frame Processing

Let us consider the reason that the conventional architecture has the low utilization in pipeline processing. Taking notice of the dashed circle in Fig. 6, it is obvious that memory writing and reading are not simultaneously executed. We explain their memory read and write operations by the example of data sequence of {A, B, C, D} in Fig. 2. The memory read and write operations in the interleaver memory for every clock cycle are expressed in Table 1, however the other operations after interleaving are omitted. $\{\mathrm{A}_1, \mathrm{B}_1, \mathrm{C}_1, \mathrm{D}_1\}$ denotes the input data at the first iteration step. The read operation of $\{\mathrm{D}_1, \mathrm{B}_1, \mathrm{C}_1, \mathrm{A}_1\}$ cannot start until the symbol of "D_1" has been buffered in memory. Similarly, the write operation of $\{\mathrm{A}_2, \mathrm{B}_2, \mathrm{C}_2, \mathrm{D}_2\}$ at the second iteration step cannot start until the symbol of "A_1" has been fed back to the input. The interleaver memory essentially needs idle cycles in memory read and write operations (given by the number of symbols N) for data buffering.

Our new idea to dissolve the aforementioned idle cycles is to apply dual-frame processing in the interleaver memory. As long as the parameters of IDMA such as the number of symbols N and the repetition code rate R_{rep} are identical between two frames, another data sequence can be processed by substituting its operation cycles for the idle cycles. The memory read and write operations by dual-frame processing are expressed in Table 2. Another data sequence of $\{\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1\}$ can be interleaved without data collision

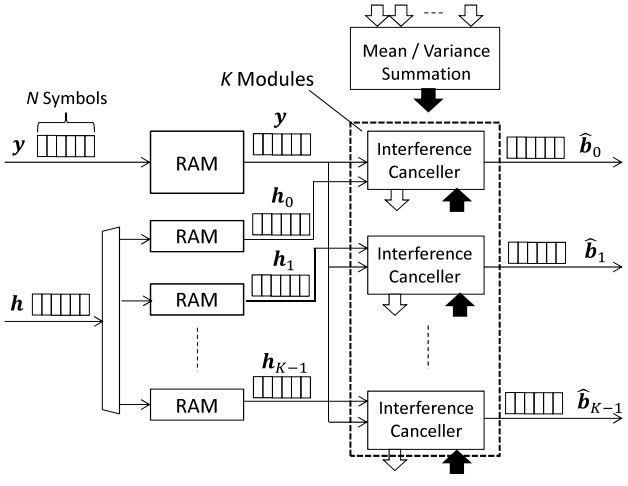

Fig. 4 Circuit structure of IDMA decoder.

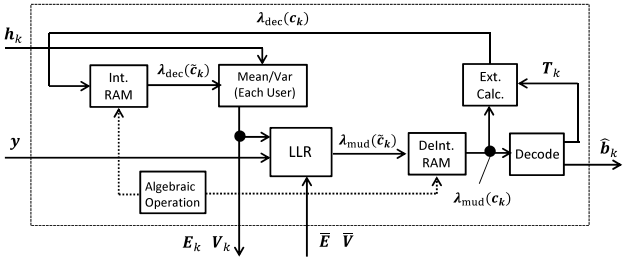

Fig. 5 Conventional architecture.

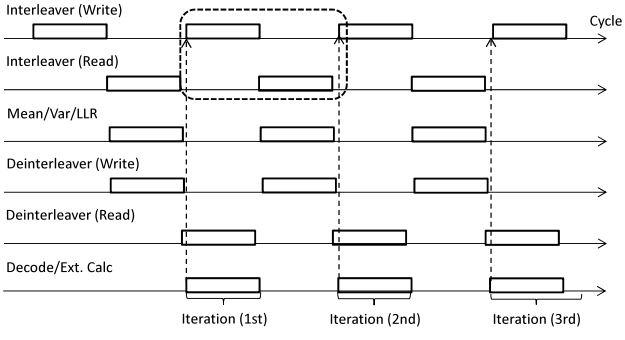

Fig. 6 Timing chart of conventional architecture.

Table 1 Memory read and write operations by single frame processing.

Cycle	Write	Buffered Data	Read
1	A_1		
2	B_1	$\{\mathbf{A_1}\}$	
3	C_1	$\{A_1,\mathbf{B_1}\}$	
4	D_1	$\{A_1,B_1,\mathbf{C_1}\}$	
5		$\{A_1,B_1,C_1,\mathbf{D_1}\}$	D_1
6		$\{A_1,B_1,C_1,D_1\}$	B_1
7		$\{A_1,B_1,C_1,D_1\}$	C_1
8		$\{A_1,B_1,C_1,D_1\}$	A_1
9	A_2	$\{A_1,B_1,C_1,D_1\}$	
10	B_2	$\{\mathbf{A_2},B_1,C_1,D_1\}$	
11	C_2	$\{A_2,\mathbf{B_2},C_1,D_1\}$	
12	D_2	$\{A_2,B_2,\mathbf{C_2},D_1\}$	
13		$\{A_2,B_2,C_2,\mathbf{D_2}\}$	D_2
14		$\{A_2,B_2,C_2,D_2\}$	B_2
15		$\{A_2,B_2,C_2,D_2\}$	C_2
16		$\{A_2,B_2,C_2,D_2\}$	A_2

Table 2 Memory read and write operations by dual-frame processing.

Cycle	Write	Buffered Data	Read
1	A_1		
2	B_1	$\{\mathbf{A_1}\}$	
3	C_1	$\{A_1,\mathbf{B_1}\}$	
4	D_1	$\{A_1,B_1,\mathbf{C_1}\}$	
5	a_1	$\{A_1,B_1,C_1,\mathbf{D_1}\}$	D_1
6	b_1	$\{A_1,B_1,C_1,D_1\}\{\mathbf{a_1}\}$	B_1
7	c_1	$\{A_1,B_1,C_1,D_1\}\{a_1,\mathbf{b_1}\}$	C_1
8	d_1	$\{A_1,B_1,C_1,D_1\}\{a_1,b_1,\mathbf{c_1}\}$	A_1
9	A_2	$\{A_1,B_1,C_1,D_1\}\{a_1,b_1,c_1,\mathbf{d_1}\}$	d_1
10	B_2	$\{\mathbf{A_2},B_1,C_1,D_1\}\{a_1,b_1,c_1,d_1\}$	b_1
11	C_2	$\{A_2,\mathbf{B_2},C_1,D_1\}\{a_1,b_1,c_1,d_1\}$	c_1
12	D_2	$\{A_2,B_2,\mathbf{C_2},D_1\}\{a_1,b_1,c_1,d_1\}$	a_1
13	a_2	$\{A_2,B_2,C_2,\mathbf{D_2}\}\{a_1,b_1,c_1,d_1\}$	D_2
14	b_2	$\{A_2,B_2,C_2,D_2\}\{\mathbf{a_2},b_1,c_1,d_1\}$	B_2
15	c_2	$\{A_2,B_2,C_2,D_2\}\{a_2,\mathbf{b_2},c_1,d_1\}$	C_2
16	d_2	$\{A_2,B_2,C_2,D_2\}\{a_2,b_2,\mathbf{c_2},d_1\}$	A_2

although the memory size has doubled. The simultaneous read and write operations can be implemented by dual-port memory having independent read and write address ports.

We should consider whether the dual-frame processing is acceptable for OFDM-IDMA communication. Figure 7 shows single-frame and multi-subframe transmission schemes. For the single-frame transmission such as wireless LAN, the dual-frame processing is not efficient because of taking a long time for the next frame reception. The multi-subframe transmission such as mobile communications in long-term evolution (LTE) and WiMAX, consisting of multiple subframes for each frame, has a chance to apply the dual-processing when their subframes are assigned to IDMA frames. Although a frame format of OFDM-IDMA communication has not been standardized yet, Matsumoto et al. presented the resource allocation of OFDM-IDMA [6] where IDMA symbols are mapped into time and frequency bins in the inside of a subframe as illustrated in Fig. 8.

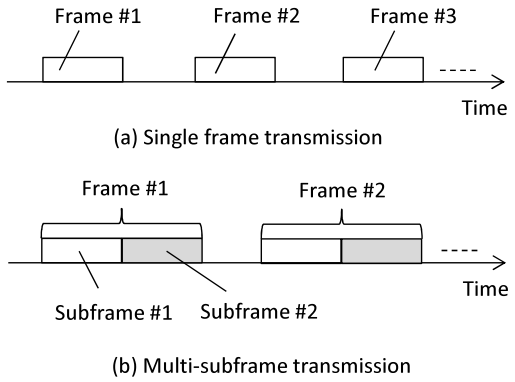

Fig. 7 Single-frame and multi-subframe transmission schemes.

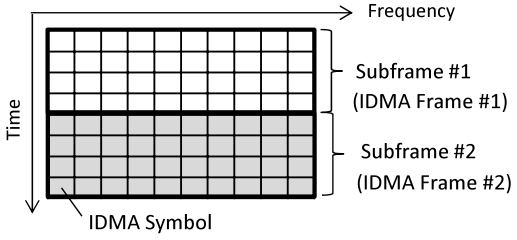

Fig. 8 Resource allocation of OFDM-IDMA.

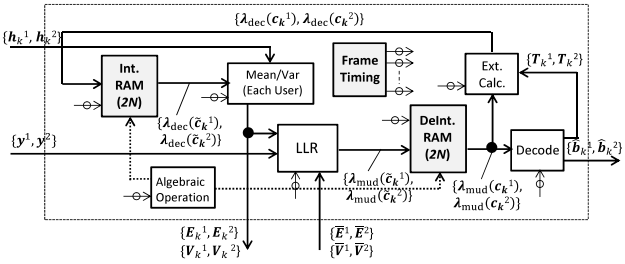

Fig. 9 Proposed architecture.

4.4 Proposed Architecture

In accordance with the idea of applying dual-frame processing, we present the proposed architecture, illustrated in Fig. 9. The proposed architecture expands the interleaver and deinterleaver memory sizes to $2N$ words and inserts the frame timing block that supplies the start timings of two frames to the other processing blocks. Most of arithmetic units are the same as those in the conventional architecture. The timing chart of proposed architecture is illustrated in Fig. 10. Highlighted in the dashed circles, the dual-frame can increase the utilization radio in pipeline processing to almost one.

Table 3 compares conventional and proposed architectures in the number of operation cycles, throughput, and memory size per user. F denotes clock frequency (Hz) and

Table 3 Comparison of architectures.

	Conventional	Proposed
Number of operation cycles	$N_{\text{iter}}(2N+\tau)+N$	$N_{\text{iter}}(2N+\tau)+N$
Throughput (bits/s)	$FN_{\text{bps}}N_{\text{b}}/(N_{\text{iter}}(2N+\tau)+N)$	$2FN_{\text{bps}}N_{\text{b}}/(N_{\text{iter}}(2N+\tau)+N)$
Memory size (bits)	$2N_{\text{bps}}W_{\text{D}}N$	$4N_{\text{bps}}W_{\text{D}}N$

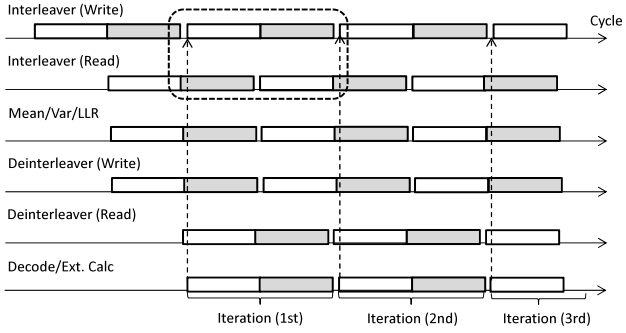

Fig. 10 Timing chart of proposed architecture.

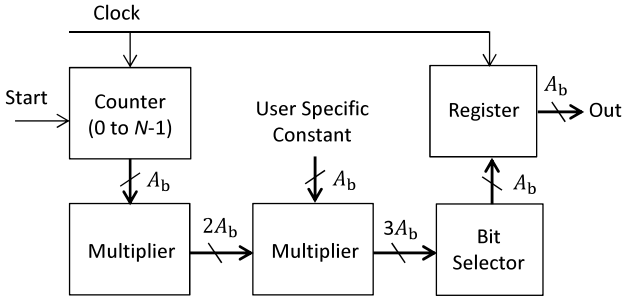

Fig. 11 Circuit structure of algebraic interleaver.

τ is the number of latency cycles caused by pipeline stages. W_{D} denotes a bit length in fixed-point operation. The proposed architecture provides a throughput of twice higher than the conventional architecture. The proposed architecture costs a double memory size, however fewer hardware resources than the conventional architecture on the same throughput condition.

4.5 Other Circuits

We describe the other circuit blocks used in the interference canceller. These blocks have been implemented in both conventional and proposed architectures.

4.5.1 Algebraic Interleaver

A circuit structure of an algebraic interleaver is shown in Fig. 11. According to (21), two multiplications and modulo operation are required. Since N is the power of two, the modulo operation can be realized by selecting lower bits corresponding to fewer digits than N. The width of address counter bus is given by A_{b}=$\log_2 N$. The multiple stage interleaver is constructed by connecting the same circuits together with inserting pipeline registers.

4.5.2 Hyperbolic Tangent Function

The direct circuit implementation of hyperbolic tangent function in (4) is complicated and costs computation cycles. Therefore, we apply the following approximation as

$$\tanh\alpha \approx \begin{cases} 1, \text{if } \alpha \geq 3 \\ -1, \text{if } \alpha \leq -3 \\ f_{\text{table}}(\alpha), \text{otherwise}, \end{cases} \tag{22}$$

where f_{table} denotes a function approximation by ROM table. The ROM values are calculated in advance. Although the approximation accuracy depends on the number of ROM words, the simulation of fixed-point arithmetic has indicated that at least 512 words are enough to have the same accuracy as the direct computation.

4.5.3 Complex Arithmetic Unit

The computations of mean and variance values in (8)–(11) and LLRs in (13)–(16) for QPSK require complex operations in arithmetic units. We define real and imaginary parts of $\boldsymbol{y}$ and $\boldsymbol{h}_k$ as

$$y_{\text{r}}(m) = \text{Re}[y(m)] \tag{23}$$
$$y_{\text{i}}(m) = \text{Im}[y(m)] \tag{24}$$
$$h_{\text{r}}(k,m) = \text{Re}[h_k(m)] \tag{25}$$
$$h_{\text{i}}(k,m) = \text{Im}[h_k(m)]. \tag{26}$$

We convert the complex operations to only real operations that can be implemented by fixed-point arithmetic units, expressed by the following equations:

$$T_{\text{r}}(k,m) = h_{\text{r}}(k,m)\tanh(\lambda_{\text{dec}}(\tilde{c}_k(2m))/2)) \tag{27}$$
$$T_{\text{i}}(k,m) = h_{\text{i}}(k,m)\tanh(\lambda_{\text{dec}}(\tilde{c}_k(2m+1))/2)) \tag{28}$$
$$E_{\text{r}}(k,m) = h_{\text{r}}(k,m)T_{\text{r}}(k,m) - h_{\text{i}}(k,m)T_{\text{i}}(k,m) \tag{29}$$
$$E_{\text{i}}(k,m) = h_{\text{r}}(k,m)T_{\text{i}}(k,m) + h_{\text{i}}(k,m)T_{\text{r}}(k,m) \tag{30}$$
$$V_{\text{r}}(k,m) = h_{\text{r}}^2(k,m)(1-T_{\text{r}}^2(k,m)) + h_{\text{i}}^2(k,m)(1-T_{\text{i}}^2(k,m)) \tag{31}$$
$$V_{\text{i}}(k,m) = h_{\text{i}}^2(k,m)(1-T_{\text{r}}^2(k,m)) + h_{\text{r}}^2(k,m)(1-T_{\text{i}}^2(k,m)) \tag{32}$$
$$\bar{E}_{\text{r}}(m) = \sum_{k=0}^{K-1} E_{\text{r}}(k,m) \tag{33}$$
$$\bar{E}_{\text{i}}(m) = \sum_{k=0}^{K-1} E_{\text{i}}(k,m) \tag{34}$$
$$\bar{V}_{\text{r}}(m) = \sum_{k=0}^{K-1} V_{\text{r}}(k,m) + \sigma^2 \tag{35}$$

Table 4 Implementation results of interference cancellers.

(a) BPSK demodulation

	Conventional (one unit)	Conventional (two units)	Proposed (one unit)
Clock frequency (MHz)	357	357	357
Circuit area (μm^2)	274,580	549,160	415,720
Logic gate count	68,645	137,290	103,930
Power consumption (mW)	34.6	69.2	41.0
Throughput ratio	0.5	1	1

(b) QPSK demodulation

	Conventional (one unit)	Conventional (two units)	Proposed (one unit)
Clock frequency (MHz)	357	357	357
Circuit area (μm^2)	517,059	1,034,118	691,249
Logic gate count	129,260	258,520	172,810
Power consumption (mW)	55.4	110.8	61.5
Throughput ratio	0.5	1	1

$$\bar{V}_{\rm i}(m) = \sum_{k=0}^{K-1} V_{\rm i}(k,m) + \sigma^2 \tag{36}$$

$$R_{\rm r}(k,m) = y_{\rm r}(k,m) - \bar{E}_{\rm r}(m) + E_{\rm r}(k,m) \tag{37}$$

$$R_{\rm i}(k,m) = y_{\rm i}(k,m) - \bar{E}_{\rm i}(m) + E_{\rm i}(k,m) \tag{38}$$

$$W_{\rm r}(k,m) = \bar{V}_{\rm r}(m) - V_{\rm r}(k,m) \tag{39}$$

$$W_{\rm i}(k,m) = \bar{V}_{\rm i}(m) - V_{\rm i}(k,m) \tag{40}$$

$$\lambda_{\rm mud}(\tilde{c}_k(2m)) = 4R_{\rm r}(k,m) \times \frac{(h_{\rm r}(k,m)+R_{\rm i}(k,m)h_{\rm i}(k,m))(h_{\rm r}^2(k,m)+h_{\rm i}^2(k,m))}{W_{\rm r}(k,m)h_{\rm r}^2(k,m)+W_{\rm i}(k,m)h_{\rm i}^2(k,m)} \tag{41}$$

$$\lambda_{\rm mud}(\tilde{c}_k(2m+1)) = -4R_{\rm r}(k,m) \times \frac{(h_{\rm i}(k,m)+R_{\rm i}(k,m)h_{\rm r}(k,m))(h_{\rm r}^2(k,m)+h_{\rm i}^2(k,m))}{W_{\rm r}(k,m)h_{\rm r}^2(k,m)+W_{\rm i}(k,m)h_{\rm i}^2(k,m)}. \tag{42}$$

5. VLSI Implementation

The interference cancellers based on conventional and proposed architectures have been implemented on CMOS 90-nm standard cell library whose voltage supply is 1.0 V. We have designed digital circuits by Verilog hardware description language and used memory macros with 16/32-bit buses and 512/1024 words. The bit length in fixed-point arithmetic units has been set to $W_{\rm D}$=16. The number of latency cycles τ is 32 for BPSK and 39 for QPSK. Table 4 summarizes the implementation results. The target clock frequency was set to 357 MHz in logic synthesis. We have measured power consumption from the synthesized gate-level circuits including switching activities. In order to compare the two architectures on the same throughput condition, we assume a case that two identical units concurrently operate in the conventional architecture. The notation of "Conventional (two units)" in Table 4 indicates this case. On the same throughput condition, the proposed architecture has reduced circuit area and power consumption by 25% and 41% for BPSK and by 33% and 44% for QPSK compared with the conventional architecture.

The implementation results of IDMA decoders (Fig. 4) based on the proposed architecture are presented in Table 5. The IDMA decoders have implemented parallel interference cancellers according to the number of users, set to K=20. Since the IDMA decoders require memory units storing received signals and channel coefficients of all users, the values of circuit area and power consumption are larger than those of the 20 times in the interference canceller.

Table 5 Implementation results of IDMA decoders.

(a) BPSK demodulation

Clock frequency (MHz)	333
Circuit area (μm^2)	13,231,160
Logic gate count	3,307,790
Power consumption (W)	1.28

(b) QPSK demodulation

Clock frequency (MHz)	333
Circuit area (μm^2)	19,968,450
Logic gate count	4,992,160
Power consumption (W)	1.84

6. Conclusion

This paper has presented the VLSI implementation of interference canceller for OFDM-IDMA systems. We have presented an algebraic interleaver for memory size reduction and proposed dual-frame processing architecture to solve a problem of decreasing an utilization ratio in interleave and deinterleave memory blocks.

In the VLSI implementation, the proposed architecture has showed smaller circuit area and power consumption than the conventional architecture on the same throughput condition. VLSI implementations of OFDM-IDMA transmitter and receiver will be studied in our future work.

References

[1] S. Whitehead, "Adopting wireless machine-to-machine technology," IEE Computing and Control Engineering, vol.15, no.5, pp.40–46, Oct. 2004.
[2] I. Cha, Y. Shah, A.U. Schmidt, A. Leicher, and M.V. Meyerstein, "Trust in M2M communication," IEEE Veh. Technol. Mag., vol.4, no.3, pp.69–75, Sept. 2009.
[3] L. Ping, L. Liu, K. Wu, and W.K. Leung, "Interleave-division multiple access," IEEE Trans. Wireless Commun., vol.5, no.4, pp.938–947, April 2006.

[4] P. Wang, J. Xiao, and L. Ping, "Comparison of orthogonal and non-orthogonal approaches to future wireless cellular systems," IEEE Veh. Technol. Mag., vol.1, no.3, pp.4–11, Sept. 2006.
[5] K. Kusume, G. Bauch, and W. Utschick, "IDMA vs. CDMA: Analysis and comparison of two multiple access schemes," IEEE Trans. Wireless Commun., vol.11, no.1, pp.78–87, Jan. 2012.
[6] T. Matsumoto, Y. Hatakawa, and S. Konishi, "Performance analysis of interleave-division multiple access for uplink in multi-cell environment," IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), pp.376–381, July 2011.
[7] S. Yoshizawa, Y. Hatakawa, T. Matsomoto, S. Konishi, and Y. Miyanaga, "Hardware implementation of an interference canceller for IDMA wireless communications," IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), pp.645–650, Nov. 2013.
[8] Y. Hikiyama, H. Iwaizumi, S. Yoshizawa, Y. Hatakawa, S. Konishi, H. Tsutsui, and Y. Miyanaga, "Channel evaluation of IDMA in outdoor transmission experiment," Proc. 2014 IEICE General Conference, B-5-45, pp.440, March 2014.
[9] L. Ping, Q. Guo, and J. Tong, "The OFDM-IDMA approach to wireless communication systems," IEEE Wireless Commun., vol.14, no.3, pp.18–24, June 2007.
[10] R. Dodd, C. Schlegel, and V. Gaudet, "DS-CDMA implementation with iterative multiple access interference cancellation," IEEE Trans. Circuits Syst. I, Regular Papers, vol.60, no.6, pp.222–231, Jan. 2013.
[11] L. Ping, L. Liu, and W.K. Leung, "A simple approach to near-optimal multiuser detection: Interleave-division multiple-access," IEEE Wireless Communications and Networking Conference (WCNC), vol.1, pp.391–396, March 2003.
[12] O.Y. Takeshita and D.J. Costello, Jr., "New classes of algebraic interleavers for turbo codes," IEEE International Symposium on Information Theory (ISIT), p.419, Aug. 1998.
[13] C.R.N. Athaudage, "Infinite series computation of probability of error in BPSK/QPSK OFDM systems with post-FFT phase error," IEEE 8th International Conference on Communication Systems (ICCS), vol.1, pp.47–51, Nov. 2002.

Appendix: Optimal Number of Stages in Algebraic Interleaver

The optimal number of stages in algebraic interleaver depends on communication specifications and channel conditions. Table A· 1 shows the simulation parameters as one of examples for OFDM-IDMA. The channel coefficients for all users are estimated from the training signals at the head of the OFDM-IDMA frame. One-path channel model assumes that random phases are multiplied by the transmitted signals.

Figure A· 1 shows the simulation results in BPSK and QPSK modes, where averages of bit error rates (BERs) for all users are plotted. The one-stage algebraic interleaver degrades a BER, especially in the QPSK mode. The three-stage algebraic interleaver shows almost the same BERs as those in the random interleaver.

Due to the channel estimation error, the received signal is affected by its amplitude and phase errors even after channel equalization. It is considered that the phase error causes the differences of BERs between BPSK and QPSK in less than 13 dB of E_b/N_0, as probability of error in BPSK/QPSK OFDM systems with post-FFT phase error has been reported in [13].

Table A· 1 OFDM-IDMA simulation parameters.

Modulation	BPSK / QPSK
Frame data size [bit] (N_b)	512
Repetition code length (N_{rep})	16
Number of symbols (N)	8192 / 4096
Number of users (K)	20
Number of iterations in interference canceller (N_{iter})	10
Channel model	One-path
Channel estimation	Training signals

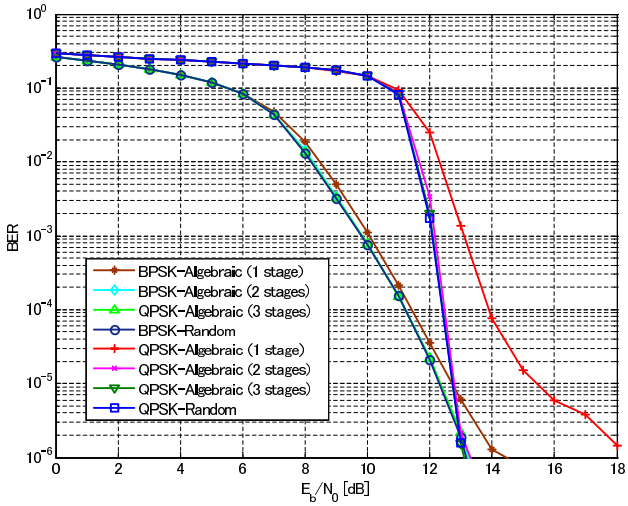

Fig. A· 1 Simulation results.

Shingo Yoshizawa received the B.E., M.E., and Ph.D. degrees from Hokkaido University, Japan in 2001, 2003, and 2005, respectively. He was an Assistant Professor in the Graduate School of Information Science and Technology, Hokkaido University from 2006 to 2012. He is currently an Associate Professor in the Department of Electrical and Electronic Engineering, Kitami Institute of Technology. His research interests are wireless communication and VLSI architecture. He had been an associate editor for IEICE Electron Express. He also served as a guest editor for IEICE Trans. Fundamentals. He is a member of IEEE and Research Institute of Signal Processing Japan.

Mai Nozaki received the B.E. degree from Kitami Institute of Technology, Japan in 2014. She is currently studying at the Graduate School of Information Engineering, Kitami Institute of Technology. Her research interests are wireless communication and digital circuits.

Hiroshi Tanimoto received the B.E., M.E., and Ph.D. degrees in Electronic Eng. all from Hokkaido University, in 1975, 1977, and 1980, respectively. In 1980 he joined the Research & Development Center, Toshiba Corp., Kawasaki, Japan, where he was engaged in research and development of telecommunication LSIs. Since 2000, he has been a Professor in the Dept. of Electrical and Electronic Eng., Kitami Institute of Technology, Kitami, Japan, where he is a department head since 2012. His main research interests include analog integrated circuit design, analog signal processing, and circuit simulation algorithms. He had been an associate editor for IEICE Trans. Fundamentals, IEEE TCAS-II, and IEICE Trans. Electron. He also served as a guest editor for IEICE Trans. Fundamentals, and IEICE Trans. Electron. He was the past chair of IEEE Circuits and Systems Society Japan Chapter. He served as a TPC chair of 2012 International Conference on Analog VLSI Circuits. Dr. Tanimoto is a member of IEICE, IEEJ, and IEEE.