Jpn. J. Appl. Phys. Vol. 42 (2003) pp. 1–6
Part 1, No. 4B, April 2003
©2003 The Japan Society of Applied Physics

Implementation of Orthogonal Frequency Division Multiplexing Modem Using Radix-N Pipeline Fast Fourier Transform (FFT) Processor

Jung-yeol Oh*, Jae-sang Cha†, Seong-kweon Kim‡ and Myoung-seob Lim§

Information Communication Research Center, Chonbuk National University, 664-14 Duck-jin dong, Jeon-ju, Korea

(Received October 2, 2002; revised manuscript received December 10, 2002; accepted for publication December 20, 2002)

In this paper, a new Radix-N pipeline fast Fourier transform (FFT) processor for the implementation of IEEE 802.11a baseband orthogonal frequency division multiplexing (OFDM) modem for wireless local area network (WLAN) is proposed. The newly proposed scheme has a simple control structure and multiplication block is designed based on canonic signed digit (CSD) with $N/4$ twiddle factors, which enables both hardware complexity and power consumption to be reduced as about less 33% and 66% respectively than the conventional Radix-4 pipeline and Radix-2 pipeline structures. In order to verify the real time operation, the IEEE802.11a baseband OFDM test-bed with the newly proposed Radix-N pipeline FFT processor is implemented using field programmable gate array (FPGA) devices. [DOI: 10.1143/JJAP.42.dummy]

KEYWORDS: Radix-N, FFT, pipeline, WLAN, OFDM, Radix, complex multiplier, CSD

1. Introduction

Recently, the orthogonal frequency division multiplexing (OFDM) technique has been used widely in the wireless communication applications such as digital broadcasting and short-range wireless local area network (WLAN) system.[1,2] Particularly, the IEEE 802.11a WLAN which is one of the typical OFDM applications could attain the maximum data rate of 54 Mbit/s using interleaver/deinterleaver, M-ary modulation mapper/demodulation demapper, convolutional encoder/Viterbi decoder and inverse fast Fourier transform/fast Fourier transform (IFFT/FFT) based on the baseband OFDM modem block as shown in the Fig. 1.[1] In the implementation of OFDM system, IFFT/FFT processor are one of the most important core components for modulation and demodulation parts. Therefore, more efficient FFT design methods for OFDM system have been developed and several papers[3–6] about the pipeline FFT design methods were presented with the following advantages. The pipeline FFT processor can be characterized by non-stopping process because the process can be completed with the same rate as the sampled input data. Certainly, a lower clock frequency is a clear advantage for pipeline architectures in case that either a high speed processing or a low power solution is necessary. In addition, pipeline structure is highly regular, which can be easily scaled and parameterized when hardware description language (HDL) is used in the design of digital FFT processor. However, because about 80% of total-power consumption is caused by the use of several complex multipliers in the pipeline structure of

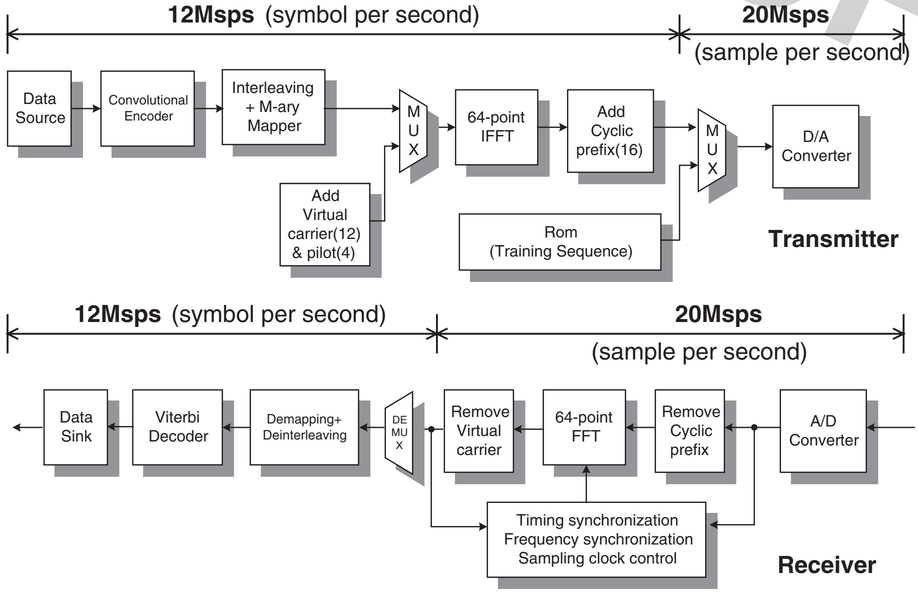


Fig. 1. IEEE 802.11a baseband OFDM modem architecture.

[*] E-mail address: jyoh@hslab.chonbuk.ac.kr
[†] Dept. of Information and Communication Eng., Seokyeong University, 16-1 Jung-eung dong, Sungbuk-ku, Seoul 136-704, Korea.
[‡] Research Institute of Electrical Communication, Tohoku University, Katahira 2-1-1, Aoba-ku, Sendai 980-8577, Japan.
[§] E-mail address: mslim@hslab.chonbuk.ac.kr

FFT,[3,7] it is necessary to devise the new design method for low power consumption.

In this paper, a new FFT processor design algorithm is proposed, in which canonic singed digit (CSD) scheme is used for multiplication with minimum number of twiddle factors. The CSD representation multipliers instead of the parallel complex multiplier circuits enables HW complexity to be reduced about less 33% and 66% than that of conventional Radix-4 pipeline and Radix-2 pipeline structures, respectively.[4] The IEEE802.11a baseband OFDM modem test-bed with the newly proposed Radix-N pipeline FFT processor is implemented using the field programmable gate array (FPGA) devices, so that it is demonstrated that the proposed FFT processor works properly in real time.

2. Design of FFT Processor

2.1 *Radix-N pipeline FFT*

The N-point DFT can be expressed in terms of the decimated sequences as follows[9]

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} = F_1(k) + W_N^k F_2(k) \tag{1}$$

where, $F_1(k)$ and $F_2(k)$ are the N/2-point DFTs of the sequences with even terms and odd terms of $x(n)$. The repeated decimation process is resulted as the lower point FFT.

In eq. (1), $W_N^k = \exp(-j2\pi k/N) = [W_N^0, W_N^1, \cdots, W_N^{N-1}]$ is twiddle factor for multiplication and the number of twiddle factors can be reduced to $N/4$ in case of N point FFT. For example, in the case of $N = 8$,

$$\begin{aligned} W_8^k &= \begin{bmatrix} W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \end{bmatrix} \\ &= \begin{bmatrix} W_8^0 & W_8^1 & W_8^2 & W_8^3 & -W_8^0 & -W_8^1 & -W_8^2 & -W_8^3 \end{bmatrix} \\ &= \begin{bmatrix} 1 & W_8^1 & -j & -jW_8^1 & -1 & -W_8^1 & j & jW_8^1 \end{bmatrix} \end{aligned} \tag{2}$$

where j term can be implemented by just exchanging real term and imaginary term without any multiplier. Thus the required twiddle factor for multiplication is only W_8^1. In the proposed structure, Radix-4 scheme is used at the starting stage for making the computation to be simplified. The 8-point DFT can be represented as in the eq. (3).

$$X(k) = \sum_{n=0}^{7} x(n) W_8^{kn} \tag{3}$$

$$\begin{aligned} &= [x(0) + (-j)^k x(2) + (-1)^k x(4) + j^k x(6)] + [x(1) + (-j)^k x(3) + (-1)^k x(5) + j^k x(7)] W_8^k \\ &= \left[\begin{bmatrix} \mathbf{R_4} & \mathbf{0_4} \\ \mathbf{0_4} & \mathbf{R_4} \end{bmatrix} \begin{bmatrix} \mathbf{x(2n)} \\ \mathbf{x(2n)} \end{bmatrix} \right]^T + \left[\begin{bmatrix} \mathbf{R_4} & \mathbf{0_4} \\ \mathbf{0_4} & \mathbf{R_4} \end{bmatrix} \begin{bmatrix} \mathbf{x(2n+1)} \\ \mathbf{x(2n+1)} \end{bmatrix} \right]^T [\boldsymbol{\Lambda}(\boldsymbol{\Omega}_\mathbf{8})] \end{aligned} \tag{4}$$

where,

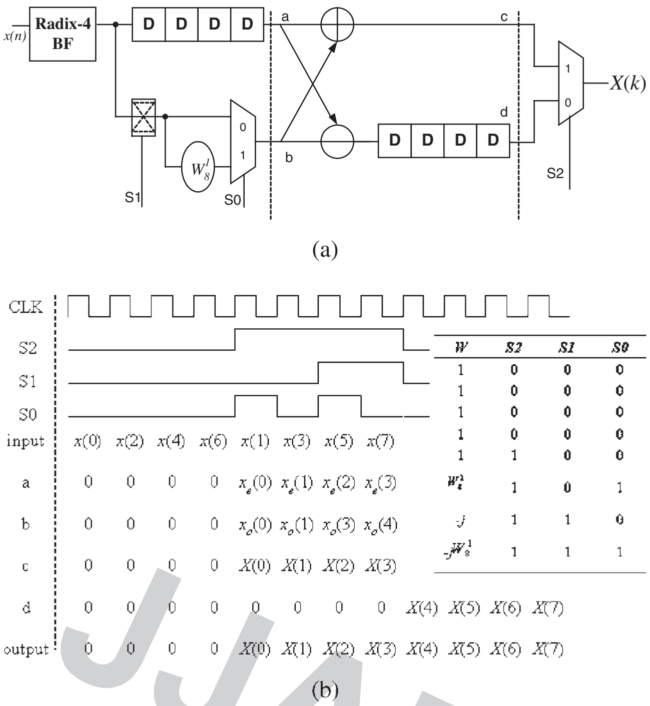
$$\mathbf{R_4} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{pmatrix}, \quad \mathbf{0_4} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{\Lambda}([a\ b\ c\ d]) = \begin{pmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{pmatrix},$$

$$\mathbf{x(2n)} = \begin{bmatrix} x(0) & x(2) & x(4) & x(6) \end{bmatrix}^T, \quad \boldsymbol{\Omega}_\mathbf{8} = \begin{bmatrix} W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \end{bmatrix}$$

Equation (4) represents the first 4 outputs of the 8-point DFT can be given by addition between the 4-point DFT outputs of even index and the 4-point DFT outputs of odd index. Similarly, the last 4 outputs of the 8-point DFT can be given by subtraction.

3. Newly Proposed Pipeline Architecture

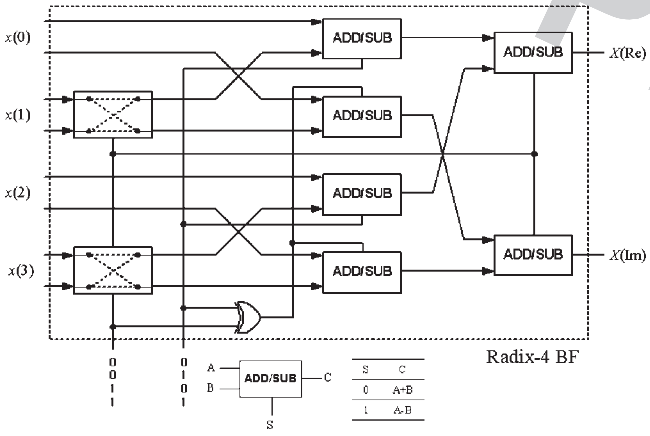
Figure 2(a) shows the new implemented architecture related with eq. (4). In the Fig. 2, the input signals $x(n)$ are reordered as the even index part and the odd index part and transformed by the Radix-4 butterfly (BF) unit. Figure 3 shows the block diagram of Radix-4 BF unit. Figure 2(b) shows the timing diagram about processed data and control signal flows. First of all, the first sequence of four outputs, $x_e(0)$, $x_e(1)$, $x_e(2)$, $x_e(3)$ from the radix-4 BF are directed to the upper path with delay elements. The second sequence of four outputs, $x_o(4)$, $x_o(5)$, $x_o(6)$, $x_o(7)$ are directed to the lower path and multiplied with twiddle factor. $X(0)$, $X(1)$, $X(2)$, $X(3)$ are sequentially generated through the addition of upper path and lower path and then $X(4)$, $X(5)$, $X(6)$, $X(7)$ through subtraction, respectively. Note that the twiddle factor W_8^1 is just only one used in this architecture. In the newly proposed scheme, the multiplication with twiddle factor is designed as a fixed size multiplier using the minimum number of twiddle factor instead of parallel complex multiplier, which enables both HW complexity and power consumption to be reduced. Figure 4 shows the revised memory efficient structure where memory for delay in Fig. 2 is used for upper path during $N/2$ clocks and again is used for lower path during next $N/2$ clocks, which enables the required memory to be reduced half. And, the required latency is $N-4$ as shown in the timing diagram of Fig. 2(b). and is same as the other schemes. When a coefficient of multiplier is a constant value, the constant multiplier can be designed by only adders by just adding the partial product terms corresponding to the non-zero bit positions in the constant coefficient. Moreover, in order to reduce the area and power consumption, the constant coefficient can be



input	$x(0)$	$x(2)$	$x(4)$	$x(6)$	$x(1)$	$x(3)$	$x(5)$	$x(7)$				
a	0	0	0	0	$x_e(0)$	$x_e(1)$	$x_e(2)$	$x_e(3)$				
b	0	0	0	0	$x_o(0)$	$x_o(1)$	$x_o(3)$	$x_o(4)$				
c	0	0	0	0	$X(0)$	$X(1)$	$X(2)$	$X(3)$				
d	0	0	0	0	0	0	0	0	$X(4)$	$X(5)$	$X(6)$	$X(7)$
output	0	0	0	0	$X(0)$	$X(1)$	$X(2)$	$X(3)$	$X(4)$	$X(5)$	$X(6)$	$X(7)$

W	*S2*	*S1*	*S0*
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
1	1	0	0
W_8^1	1	0	1
$-j$	1	1	0
$-jW_8^1$	1	1	1

Fig. 2. (a) Newly proposed architecture of 8 point FFT. (b) Timing diagram of signals and control signals.



S	C
0	A+B
1	A-B

Fig. 3. Radix-4 BF(Butterfly) unit.

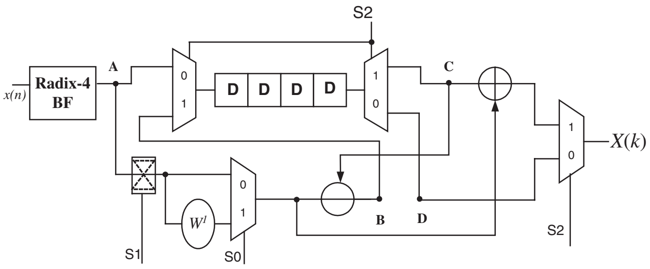


Fig. 4. Memory efficient model of 8 point FFT.

encoded such that it contains the fewest number of non-zero bits, using CSD representation.[10] For example, Table I shows the 8 bit binary number representations of real part in twiddle factors, where three twiddle factors $W_{16}^1 = 0.9239 - j0.3827$, $W_{16}^2 = 0.7071 - j0.7071$, $W_{16}^3 = 0.3827 - j0.9239$ are necessary for processing 16-point DFT. Since twiddle factor can be represented to have the minimal number of non-zero bits, one real multiplier can be designed by just 3 adders according to the number of '1' shown in Table I even in the worst case with the successive non-zero bits.

Table I. Binary representation of twiddle factors of 16 point DFT. ($\bar{1}$ indicates -1, the twiddle factor is represented by 8bit.)

Decimal representation	2's complement	CSD representation	Design representation
0.9239	01110110	$1000\bar{1}0\bar{1}0$	$1000\bar{1}0\bar{1}0$
0.7071	01011010	$10\bar{1}0\bar{1}010$	$10\bar{1}0\bar{1}010$
0.3827	00110000	*Not used*	00110000

A fractional number X can be represented as a CSD representation in eq. (5).

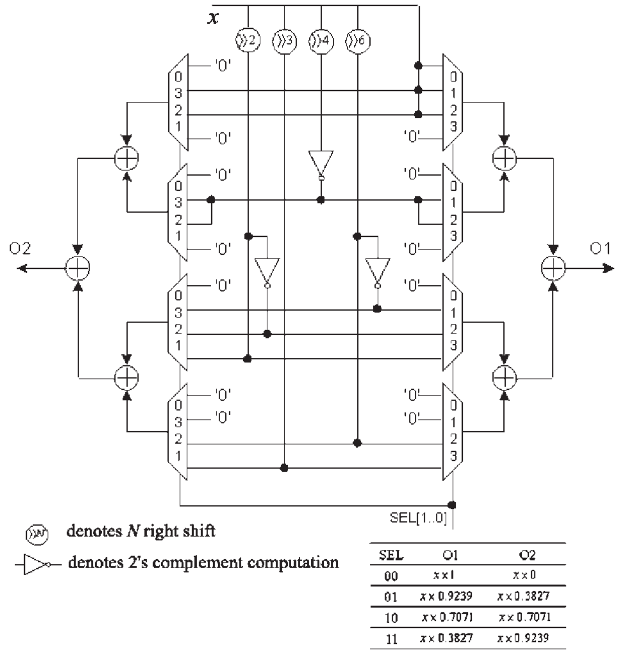
$$X = \sum_{k=0}^{M} S_k 2^{-p_k} \tag{5}$$

where $S_k \in \{-1, 0, 1\}$, $P_k = 0, 1, 2, \ldots, M$ and $M + 1$ is the coefficient of word-length.[11)]

Figure 5 shows the example for complex multiplier with twiddle factor based on CSD representation. Shift operators corresponding to non-zero bits of twiddle factors are designed and can be shared by twiddle factor which has same non-zero bit. The processed data are output sequentially using multiplexer with select signal.

$$\begin{aligned}(x + jy) \times W_N^k &= (x + jy) \times (c - js)\\ &= (xc + ys) + j(yc - xs) = (O_1 + O_4) + j(O_3 - O_2)\end{aligned} \tag{6}$$

Equation (6) shows the processed data after multiplying input signal $x + jy$ with twiddle factor $W_N^k = c - js$. To implement two real multipliers by this method, total 6 adders are required. Therefore, total 14 adders are necessary in



SEL	O1	O2
00	$x \times 1$	$x \times 0$
01	$x \times 0.9239$	$x \times 0.3827$
10	$x \times 0.7071$	$x \times 0.7071$
11	$x \times 0.3827$	$x \times 0.9239$

Fig. 5. The example of multiplier by shifters and adders for twiddle factors of 16-point DFT.

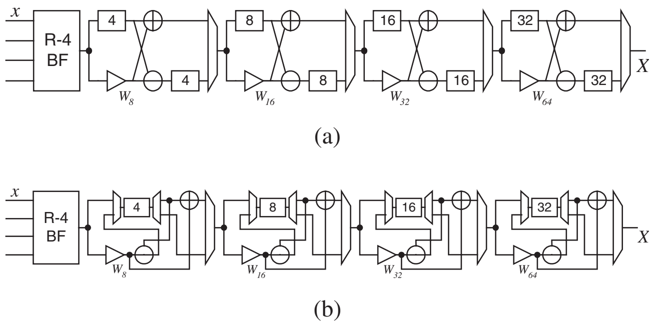
order to implement a complex multiplier with adders, because one complex multiplier requires 4 real multipliers and 2 adders. Equation (7) shows a radix-N pipeline structure of 64-point FFT as follows.

$$\begin{aligned}[X(0)X(1)\cdots X(63)] &= [\Re_{32}\{\mathbf{x(2n)}\}]^T + [\Re_{32}\{\mathbf{x(2n+1)}\}]^T \boldsymbol{\Lambda}_{\mathbf{64}} \\ &= \{\boldsymbol{\alpha} + \boldsymbol{\beta}\boldsymbol{\Lambda}_{\mathbf{32}}\} + \{\boldsymbol{\gamma} + \boldsymbol{\delta}\boldsymbol{\Lambda}_{\mathbf{32}}\}\boldsymbol{\Lambda}_{\mathbf{64}}\end{aligned} \qquad (7)$$

where,

$$\begin{aligned}\boldsymbol{\alpha} &= \big([\Re_4\{x(16n)\}]^T + [\Re_4\{x(16n+8)\}]^T \boldsymbol{\Lambda}_{\mathbf{8}}\big) \\ &\quad + \big([\Re_4\{x(16n+4)\}]^T + [\Re_4\{x(16n+12)\}]^T \boldsymbol{\Lambda}_{\mathbf{8}}\big)\boldsymbol{\Lambda}_{\mathbf{16}} \\ \boldsymbol{\beta} &= \big([\Re_4\{x(16n+2)\}]^T + [\Re_4\{x(16n+10)\}]^T \boldsymbol{\Lambda}_{\mathbf{8}}\big) \\ &\quad + \big([\Re_4\{x(16n+6)\}]^T + [\Re_4\{x(16n+14)\}]^T \boldsymbol{\Lambda}_{\mathbf{8}}\big)\boldsymbol{\Lambda}_{\mathbf{16}} \\ \boldsymbol{\gamma} &= \big([\Re_4\{x(16n+1)\}]^T + [\Re_4\{x(16n+9)\}]^T \boldsymbol{\Lambda}_{\mathbf{8}}\big) \\ &\quad + \big([\Re_4\{x(16n+5)\}]^T + [\Re_4\{x(16n+13)\}]^T \boldsymbol{\Lambda}_{\mathbf{8}}\big)\boldsymbol{\Lambda}_{\mathbf{16}} \\ \boldsymbol{\delta} &= \big([\Re_4\{x(16n+3)\}]^T + [\Re_4\{x(16n+11)\}]^T \boldsymbol{\Lambda}_{\mathbf{8}}\big) \\ &\quad + \big([\Re_4\{x(16n+7)\}]^T + [\Re_4\{x(16n+15)\}]^T \boldsymbol{\Lambda}_{\mathbf{8}}\big)\boldsymbol{\Lambda}_{\mathbf{16}}\end{aligned}$$

where $n = 0, 1, 2, 3$ and $\Re_N\{\cdot\}$ means Radix-N butterfly processing unit and a 64×1 matrix as a result of multiplication between 64×64 matrix which $\mathbf{R_4}$ matrix as in eq. (4) is repeated along the diagonal direction and 64×1 matrix which a group of $\mathbf{x}(\cdot)$s is repeated 16 times. $\boldsymbol{\Lambda}_{\mathbf{N}}$ is a 64×64 diagonal matrix which the latter half part of N elements has the same elements but negative sign as the former part and those N elements are repeated along the diagonal axis. Figure 6(a) shows the block model of Radix-N pipeline 64-point FFT, which corresponds to eq. (7). And, the memory efficient model is shown in Fig. 6(b).



(a)

(b)

Fig. 6. (a) New Radix-N pipeline FFT structure (64-point FFT). (b) Memory efficient model of Radix-N pipeline FFT.

4. Comparisons of the Proposed Scheme and Conventional Schemes

Pipeline FFT processor has highly regular property, which can be easily scaled and parameterized when HDL is used in the design. There are several schemes such as Radix-2 multipath delay commutator (R2MDC), Radix-2 single-path delay feedback (R2SDF),[12] Radix-4 single-path delay feedback (R4SDF),[12] Radix-4 multi-path delay commutator (R4MDC),[13] Radix-4 single-path delay commutator (R4SDC),[5] Radix-2^2 single-path delay feedback (R2^2SDF).[8] Generally, in the pipeline architecture, the complex multipliers used to compute twiddle factors in each stage dissipate 80% of total power as well as big portion of HW size.[3,7] In this paper, the proposed scheme is compared with the conventional pipeline schemes by converting HW size to the number of full adder required to implement complex multipliers in each method. Table II shows comparison of the number of complex multiplier and memory between conventional and proposed schemes. And, Table II shows the comparison of the number of full adder in case of $N = 64$ and W(word length) $= 12$ bit, where one complex multiplier is designed by 4 real multipliers and 2 adders. In case of parallel multiplier, W[bit] $\times$ W[bit] multiplication based on the 2's complement representation needs at least $(W-1)^2$ of full adders. However, in case of constant multiplier, the required number of full adder becomes $W \times \kappa$

Table III. Design specifications of implemented test-bed of IEEE802.11a baseband OFDM modem.

Components	Design specifications
IFFT/FFT Processor	• Processing speed: 20 Mbps • data rate: 12 Mbps • input/output bits: complex 8 bits • (de)modulation scheme: QPSK
Convolutional encoder	• code rate: 1/2 • constraint length: 7
Viterbi decoder	• register exchange method • majority voting employ
Synchronizer	• synchronization using short, long sequence • signal detection • symbol synchronization
DAC/ADC	• DAC(AD9708), ADC(AD9283) • bit resolution: complex 8 bits • output coding: offset binary code

Table II. Comparative analysis of proposed scheme and conventional schemes. (* is in case of $W = 12$ bit, $N = 64$)

Architectures	# of complex multipliers	Memory	# of Full Adders(*)	Memory(*)	Total gate(*)	Control complexity
R2MDC	$\log_2(N) - 2$	$3N/2 - 2$	2032	1152	3184	simple
R2SDF	$\log_2(N) - 2$	$N - 1$	2032	756	2788	simple
R4MDC	$3(\log_4(N) - 1)$	$5N/2 - 4$	3048	1872	4920	simple
R4SDF	$\log_4(N) - 1$	$N - 1$	1016	756	1772	complex
R4SDC	$\log_4(N) - 1$	$2N - 2$	1016	1512	2528	complex
R2^2SDF	$\log_4(N) - 1$	$N - 1$	1016	756	1772	medium
Proposed	$\log_2(N) - 2$	$N - 4$	672	720	1392	simple

(κ is the number of non-zero bits in the coefficient).[10,14] In this paper, after reducing the twiddle factors by N to $N/4 - 1$ in each stage, the multiplication with twiddle factor is implemented by shifters and adders based on CSD number representation. As shown in Table II, the proposed scheme can save 33% of HW complexity in the size of complex multiplications compared with Radix-4 and 66% with Radix-2 pipeline schemes. In the view point of VLSI power consumption, it can be calculated by eq. (8).[15)

$$P = \alpha \times n \times C_{out} \times V_{dd}^2 \times f \qquad (8)$$

where,
α = Activation rate,
n = Number of logic gate,
C_{out} = Capacitive load per on gate,
V_{dd} = Voltage swing,
f = Clock frequency
Under the assumption that other conditions are equal, it is possible to reduce the power consumption of 33% and 66% with Radix-4 pipeline and Radix-2 pipeline scheme respectively by reducing the gate count.[15] The control of the proposed scheme is very efficient to be compared with the complex control of Radix-4 pipeline schemes. Figure 5 displays the number of full adders required to design the multiplier according to the each FFT point length. As the point length increases, the gain about HW size gets better remarkably, as shown in the Fig. 5.

5. An Implementation Example

The implemented test-bed of IEEE 802.11a baseband OFDM modem block is shown in the Fig. 6. Short and long training sequence for symbol synchronization are transmitted at initialization stage. The design specifications of each components of the implemented test-bed are presented in the Table III. The key processor of the implemented test-bed is a Radix-N pipeline IFFT/FFT processor in which symbol-rate of 12 Msps can be processed. The external data word length for both input and output is adopted to $W = 8$ bits real and imaginary part, individually. The number of complex multiplier circuits is decreased by 33% and 66% than that of conventional Radix-4 pipeline and Radix-2 pipeline structures,[4,13] respectively. This property enables high-speed IEEE802.11a baseband OFDM modem to be efficiently implemented. In order to evaluate operation, the baseband OFDM modem test-bed with the proposed radix-N pipeline IFFT/FFT using Altera[17] FPGA device EPF10K200SRC240-1 is implemented. Figure 8 shows the measured OFDM symbols modulated by the proposed IFFT processor, demodulated by the proposed FFT processor. It is certified that the operation of IFFT/FFT processor works well in the test-bed of IEEE 802.11a baseband OFDM modem.

6. Conclusions

In this paper, a new Radix-N FFT pipeline processor to implement IEEE 802.11a WLAN baseband modem is proposed. The proposed scheme is not only easy to be designed, but also hardware complexity can be reduced because the multiplication is designed by CSD with the use of the minimum number of twiddle factors instead of parallel complex multiplier. The power consumption of the proposed FFT processor can be decreased by 33% and 66% than that of conventional Radix-4 pipeline and Radix-2 pipeline structures, respectively. Furthermore, the real time operation of the proposed IFFT/FFT processor are measured and certified with the implemented test-bed of IEEE 802.11a baseband OFDM modem using FPGA.

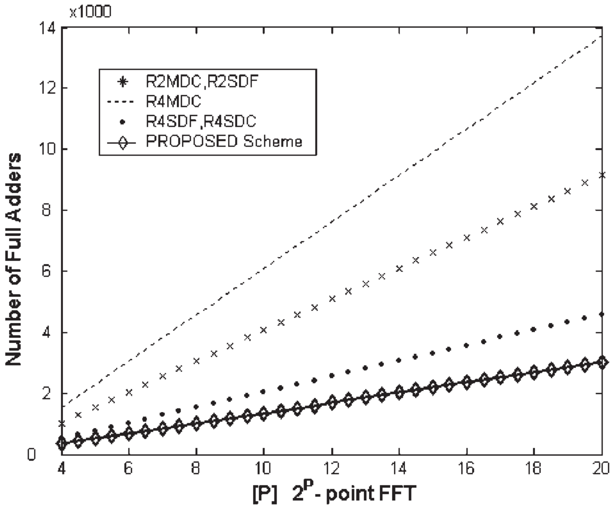


Fig. 7. Comparison of full adders with conventional schemes.

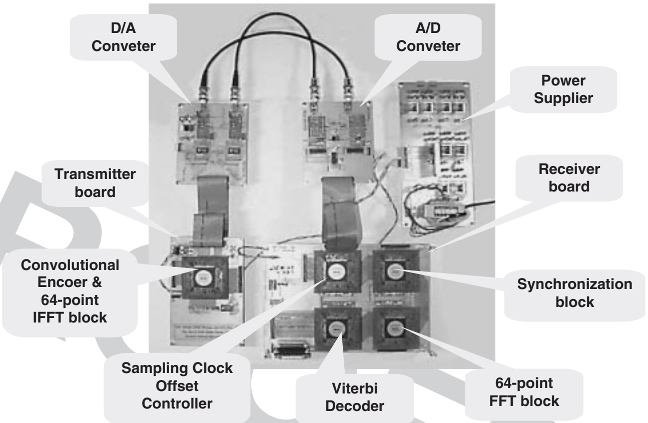


Fig. 8. Implemented test-bed of IEEE802.11a baseband OFDM modem with Radix-N pipeline FFT.

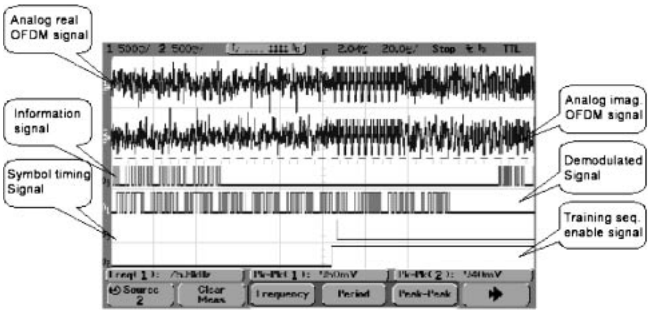


Fig. 9. Measured OFDM transmitted and received symbols.

1) IEEE 802.11a/D7.0: *High Speed Physical Layer in the 5 GHz Band* (1999).
2) J. Choi, S. Park, D. Han and S Park: *IEEE Int. Symp. Circuits & Systems (ISCAS)* (2000) Proc. IEEE Int. Symp. Vol. 5, p. 693.
3) J. Melander: Thesis No. 618, Linköping University, Sweden, 1997.
4) S. He and M. Torkelson: *1998 Union Radio-Scientifique Internationale (URSI) Int. Symp.* (1998) p. 257.
5) G. Bi and E. V. Jones: IEEE Trans. Acoust. Speech & Signal Proc. **37** (1989) p. 1982.
6) Y. Jung, Y. Tak, J. Kim, J. Park, D. Kim and H. Park: *TENCON. Proc. IEEE Region 10 Int. Conf.* (2001) Vol. 2, p. 676.
7) T. Widhe: Thesis No. 619, Linköping University, Sweden, 1997.
8) S. He: Diss. No. 133, Lund University, Sweden, 1995.
9) A. V. Oppenheim and R. W. Schafer: *Discrete-Time Signal Processing* (Prentice-Hall, Englewood Cliffs, 1989).
10) K. K. Parhi: *VLSI Digtal Signal Processing Systems* (Wiley-Interscience, 1999) pp. 505–511, pp. 478–489.
11) X. Xu and B. Nowrouzian: *IEEE Canadian Conf.* (1999) Vol. 2, p. 811.
12) E. H. Wold and A. M. Despain: IEEE Trans. Comp. C-**33** (1984) 414.
13) E. E. Swartzlander, V. K. Jain and H. Hikawa: J. VLSI Signal Proc. **4** (1992) 165.
14) M. Mano: *Computer System Architecture* (Prentice Hall, Upper Saddle River, 1993) 3rd ed.
15) T. Hanyu, M. Kuwahra and T. Higuchi: IEICE Trans. Elec. E**77**-C (1994).
16) H. Schmidt and K.D. Kammeyer: 1st Int. OFDM Workshop (1999).
17) Altera is a trademark of Altera Corporation.