Hardware Efficient Mixed Radix-25/16/9 FFT for LTE Systems

Jienan Chen, *Student Member, IEEE*, Jianhao Hu, *Member, IEEE*, Shuyang Lee, and Gerald E. Sobelman, *Senior Member, IEEE*

***Abstract*— In this paper, we propose a hardware-efficient mixed generalized high-radix (GHR) reconfigurable fast Fourier transform (FFT) processor for long-term evolution applications. The GHR processor based on radix-25/16/9 uses a 2-D factorization scheme as the high-radix unit and a 1-D factorization method as the system data routing technology. The 2-D factorization scheme is implemented by an enhanced delay element matrix structure, which supports 25-, 16-, 9-, 8-, 5-, 4-, 3-, and 2-point FFTs. Two different designs were implemented. One design (called discrete Fourier transform core) supports 34 different transform sizes from 12 to 1296 points, while the other design (called FFT core) supports five different power-of-two sizes from 128 to 2048 points. The 1-D factorization method is performed by a coprime accessing technology, which accesses the data in parallel without conflict using a RAM. The GHR combines 2-D and 1-D factorization techniques and improves the throughput by a factor of two to four with comparable hardware cost compared with the previous designs. The speed–area ratio of the proposed scheme is nearly two times better than that of previous FFT processors. Application-specified integrated circuit implementation results based on a 0.18-μm technology are also provided.**

***Index Terms*— Fast Fourier transforms (FFTs), generalized high radix (GHR), long-term evolution (LTE), reconfigurable.**

I. Introduction

THE FAST Fourier transform (FFT) has been investigated extensively for decades [1], and it also has been applied in many orthogonal frequency division multiplexing systems, such as 802.11 and terrestrial digital video broadcasting [2]. The in-place memory-based processor has several advantages over a pipeline FFT processor [3], such as a low hardware area and reduced power consumption [4]. However, a data accessing conflict will occur when both the in-place policy and a high-radix structure are adopted. Several methods have been proposed to solve this problem for a 2^n-point FFT [5]–[7]. In addition, several high-performance reconfigurable FFT processors were proposed in [8]–[10] for the case of 2^n-point FFTs.

Manuscript received August 8, 2012; revised January 5, 2013, March 24, 2013, and August 25, 2013; accepted January 16, 2014. Date of publication March 4, 2014; date of current version January 30, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61101033 and Grant 61070696, in part by the National High Technology Project of China under Grant 2012AA011702, and in part by the Fundamental Research Funds for the Central Universities.

J. Chen, J. Hu, and S. Lee are with the University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: neonanme@163.com; jhhu@uestc.edu.cn; cjn_zhy@sina.com.cn).

G. E. Sobelman is with the University of Minnesota, Minneapolis, MN 55455 USA (e-mail: sobelman@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2014.2304834

The traditional optimized methods for 2^n-point FFTs in [5]–[10] are difficult to apply to a general mixed radix FFT design. In long-term evolution (LTE) uplink precoding [11], for example, transform sizes ranging from 12 points up to 1296 points are required. Several research works have been conducted to address this issue. In [12], a 2-D discrete Fourier transform (DFT) factorization structure was proposed based on using a delay element matrix (DEM) to perform a generalized radix DFT. The DEM can access data directly without requiring memory accessing operations. The generalized mixed radix (GMR) method proposed in [4] is a 1-D DFT factorization algorithm using multiple RAM banks. A variable-length processor for LTE applications was proposed for field-programmable gate array (FPGA) platforms in [12]. The methods proposed in [4] and [12] can be also used in the LTE uplink precoding DFT.

The 2-D DFT factorization method, such as DEM, can achieve a high throughput with a simple data routing scheme. However, a large-scale register matrix is required for large size DFTs, which occupies much more hardware area than a RAM-based scheme in an application-specific integrated circuit (ASIC) implementation. Moreover, the DEM is not suitable for variable-length DFT designs. On the other hand, the processing speed of a 1-D DFT processor is slower than that of a 2-D DFT processor with its complex data routing operations.

In this paper, we combine the 1-D and 2-D DFT factorization methods together and implement a new hardware efficient generalized high-radix (GHR) reconfigurable DFT processor. Several optimization techniques are applied to the radix-25/16/9 DFTs. In the proposed design, the 2-D factorization method is implemented by an enhanced DEM (EDEM), which supports high- and variable-radix DFTs (25-, 16-, 9-, 8-, 5-, 4-, 3- and 2-point DFTs). The system data routing is implemented by a novel coprime accessing technology (CAT), which is a 1-D factorization method based on using a RAM. The proposed DFT processor has much higher throughput than the designs in [4] and [13]. The speed–area ratio of the proposed method is nearly two times better than [4] and three times better than [13]. Moreover, the GHR processor supports an in-place continuous flow (CF) of data within a single clock domain. We have implemented two different designs: one of them (called the DFT core) supports 34 different transform sizes (including nonpower-of-two sizes), while the second one (called the FFT core) supports only 2^n-point transforms. The DFT design methodology,

EDEM, and CAT can also be used for implementing general DFT sizes. The processor has been taped out using a 0.18-μm technology, and chip measurement results are provided.

The remainder of this paper is organized as follows. The DFT factorization algorithm is described in Section II, and the design of the GHR processor with EDEM is presented in Section III. The CAT technique and the structure of the complete DFT processor are given in Section IV, and a performance analysis is presented in Section V. Finally, Section VI gives our conclusion.

II. DFT Factorization Algorithm

A. Factorization Method

In an LTE system, the number of points (N) of the DFT can be specified on a frame-by-frame basis and expressed as

$$N = 2^{\alpha} \cdot 3^{\beta} \cdot 5^{\gamma} \tag{1}$$

where α, β, and γ are positive integers. The 34 specific DFT sizes can be found in [11].

It is well known that a higher radix-based DFT processor can reduce the number of computation cycles (CCs). However, a higher radix unit will cost more hardware resources. Moreover, the parallel data accessing due to the high radix will be a critical issue for the RAM-based DFT processor. The 2-D-based DFT processor can solve the parallel data accessing problem effectively, but the area cost for the register matrix is much larger than for a RAM. In this paper, we propose a mixed 2-D and 1-D DFT processor, in which the 2-D technology serves as a radix unit and a 1-D method is used as the system data routing technology.

Traditional approaches factorize DFTs to 3-, 4-, and 5-point sizes in LTE applications. We factorize the DFTs to a higher radix, such as 25, 16, and 9, which reduces the required number of CCs. Furthermore, smaller radix sizes of 2–5 and 8 are also included in the proposed scheme to support all of the possible DFT sizes in an LTE system. Thus, (1) can be rewritten as follows:

$$N = 16^{a_1} \cdot 8^{a_2} \cdot 4^{a_3} \cdot 2^{a_4} \cdot 9^{b_1} \cdot 3^{b_2} \cdot 25^{c_1} \cdot 5^{c_2} \tag{2}$$

where

$$\begin{aligned} \alpha &= 4a_1 + 3a_2 + 2a_3 + a_4 \quad (a_1 \geq 0, \quad a_2, a_3, a_4 = 0, 1) \\ \beta &= 2b_1 + b_2 \quad (b_2 \geq 0, \ b_2 = 0, 1) \\ \gamma &= 2c_1 + c_2 \quad (c_2 \geq 0, \ c_2 = 0, 1). \end{aligned}$$

As shown in Fig. 1, we first factorize the N-point DFT into 2^{α}, 3^{β} and 5^{γ}-point DFTs using prime factor algorithm (PFA) [14] method. Then, the 2^{α}, 3^{β}, and 5^{γ}-point DFTs are decomposed using mixed 16-, 9-, and 25-point DFTs based on the Cooley–Tukey algorithm (CTA) [1]. For example, an 864-point DFT is decomposed into a 16 × 2 × 9 × 3-point DFT. In this manner, the N-point DFT can be performed by a mixed-radix DFT using factors of 25, 16, 9, 8, 5, 4, 3, and 2.

According to the definition of LTE DFT sizes in (1), the radix-5-based DFT size is 25. Hence, we use a 25-point DFT as the maximum radix unit size. A technology reuse approach is incorporated in the radix unit to achieve hardware efficiency.

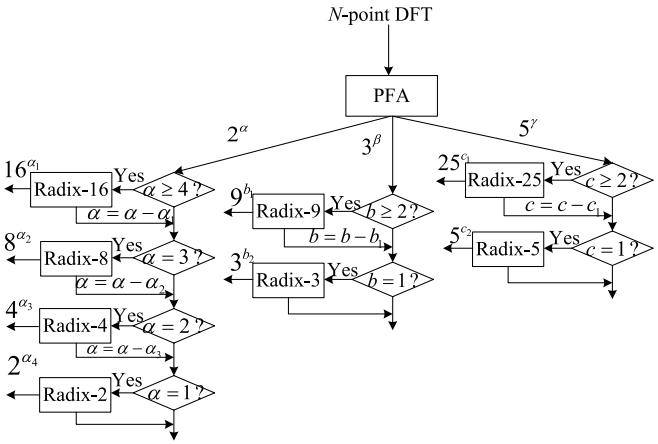

Fig. 1. Factorization flowchart.

Hence, 2^n-point and 3^n-point DFTs are performed by radix-16 and 9, respectively. In addition, the design supports continues pipeline data processing within a single clock domain with the selected radix.

B. Index Mapping Review

The DFT is defined as

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}, \quad W_N^{nk} = \exp(-j2\pi nk/N). \tag{3}$$

The decomposition of a large-size DFT into smaller size DFTs is given by [4]

$$\begin{aligned} n &= (N_2 n_1 + A_2 n_2) \bmod N, \quad n_1, k_1 = 0, 1, \ldots, N_1 - 1 \\ k &= (B_1 k_1 + N_1 k_2) \bmod N, \quad n_2, k_2 = 0, 1, \ldots, N_2 - 1. \end{aligned} \tag{4}$$

Equation (4) maps the indices n and k into index vectors (n_1, n_2), and (k_1, k_2), i.e., from a 1-D range [0, $N-1$] into a 2-D range [0, $N_1 - 1$] × [0, $N_2 - 1$]. If N_1 and N_2 are coprime, A_2 and B_1 satisfy the following equations:

$$\begin{aligned} A_2 &= p_1 N_1 \ , A_2 = q_1 N_2 + 1 \\ B_1 &= p_2 N_2 \ , B_1 = q_2 N_1 + 1 \end{aligned} \tag{5}$$

where p_1 and p_2 are positive integers. Then, the DFT can be represented as

$$\begin{aligned} X[k_1, k_2] &= \sum_{n_2} \sum_{n_1} x[n_1, n_2] W_{N_1}^{n_1 k_1} W_{N_2}^{n_2 k_2} \\ &= \sum_{n_2} y[k_1, n_2] W_{N_2}^{n_2 k_2} \end{aligned} \tag{6}$$

where

$$y[k_1, n_2] = \sum_{n_1} x[n_1, n_2] W_{N_1}^{n_1 k_1}. \tag{7}$$

$y[k_1, n_2]$ can be obtained by an N_1-point DFT. $\sum y[k_1, n_2] W_{N_2}^{n_2 k_2}$ is performed by an N_2-point DFT of $y[k_1, k_2]$. This procedure is the well-known PFA.

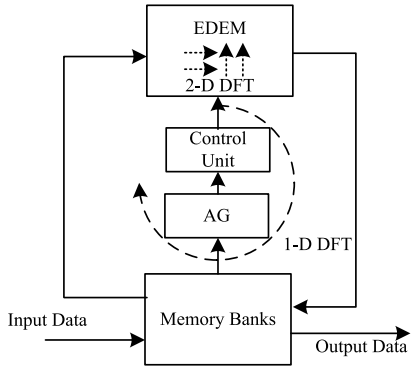

Fig. 2. Structure of the reconfigurable GHR processor.

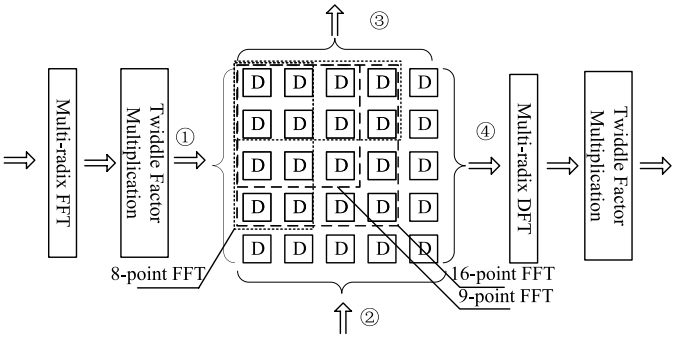

Fig. 3. Proposed EDEM.

On the other hand, if N_1 and N_2 are not relatively prime, we let $A_2 = B_1 = 1$. The DFT is then given by

$$X[k_1, k_2] = \sum_{n_2} \sum_{n_1} x[n_1, n_2] W_{N_1}^{n_1 k_1} W_{N_2}^{n_2 k_2} W_N^{n_2 k_1} = \sum_{n_2} W_N^{n_2 k_1} y[k_1, n_2] W_{N_2}^{n_2 k_2}. \quad (8)$$

In a similar fashion, $y\ [k_1,\ n_2]$ can be obtained by N_1-point DFTs, and the multiplication by $W_N^{n_2 k_1}$ is known as the twiddle factor multiplication. Then, N_2-point DFTs are performed to obtain the final result. This procedure is the well-known CTA.

III. GHR Reconfigurable DFT Processor

The GHR DFT processor consists of the EDEM, memory banks, address generator (AG), and control unit, as shown in Fig. 2. The EDEM is a 2-D DFT factorization process, while the data routing is based on memory banks, which is a 1-D DFT process. The proposed EDEM supports 25-, 16-, 9-, 8-, 5-, 4-, 3-, and 2-point DFTs, which are used to calculate the 34 DFT lengths in LTE. Before the presentation of EDEM, we first review the DEM method proposed in [12].

A. Delay Element Matrix

Suppose that an N-point DFT $x_0, x_1, \ldots, x_{N-1}$ is factored to N_1 and N_2 points by (7). The computation is done in two stages: 1) perform an N_1-point DFT N_2 times and 2) perform an N_2-point DFT N_1 times. In the first stage, the input data of N_1-point DFTs are $x[n_1, n_2] n_1 = 0, 1, \ldots, N_1 - 1$, with fixed n_2. The output data are $y[k_1, n_2]$, $k_1 = 0, 1, \ldots, N_1 - 1$. In the second stage, the input data of the N_2-point DFTs are $y[k_1, n_2]$ for $n_2 = 0, 1, \ldots, N_2 - 1$ with fixed k_1. The output data are $X[k_1, k_2](k_1 = 0, 1, \ldots, N_2 - 1)$, which is the result of the N-point DFT. Thus, when the data can be processed concurrently and continuously, the required CCs of the DEM for an N^2-point DFT is $2N$.

B. Enhanced DEM

We optimize the 25-point DFT DEM to support 25-, 16-, 9-, 8-, 5-, 4-, and 2-point DFTs, as shown in Fig. 3. The multiradix DFT block can perform 5-, 4-, 3-, and 2-point DFTs. The 16- and 9-point DFTs are decomposed into 4- and 3-point DFTs, respectively. The 16-point DFT processing uses a 4×4 subarray of delay elements in the EDEM, whereas a 9-point DFT uses a 3×3 subarray. The 8-point DFT is decomposed into two 2-point DFTs and one 4-point DFT. Two clock cycles are required to perform an 8-point DFT. If 5-, 4-, 3-, or 2-point DFTs are required, the delay elements can be bypassed. The computational processes for 9- and 16-point DFTs are similar toN^2-point DFT. We present the data flow of an 8-point DFT here as an example, as shown in Fig. 4. The first frame data $\{x_0, x_1, \ldots, x_7\}$ are injected to the register matrix horizontally for two clock cycles. Then, the data are output vertically to perform a 2-point DFT, while the next frame data x_i' are simultaneously injected to the bottom of the array.

C. Hardware Efficient Multiradix DFT

To reduce the hardware cost of the multiradix DFT block, we implement 2-, 3-, 4-, and 5-point DFTs by reusing adders and multipliers with the Winograd Fourier transform algorithm (WFTA) [15]. The computational process of the WFTA contains three steps: 1) preaddition; 2) multiplication; and 3) postaddition. For 2-, 3-, 4-, and 5-point DFTs, the 5-point WFTA requires the most additions and multiplications. Hence, we can implement the 2-, 3-, and 4-point WFTAs by reusing the structure of the 5-point WFTA shown in Fig. 5. The 5-point WFTA requires ten real multiplications and 34 real additions.

The 5-point WFTA is organized as shown at the bottom of the next page.

On the other hand, the 3-point WFTA is organized as shown in the equation at the bottom of the next page.

Thus, the additions T_{51} and T_{31} can share one adder. In the case of a 5-point WFTA, x_5 is selected, while x_3 is selected for a 3-point WFTA. Next, consider the computations for a 4-point WFTA

First stage:	Second stage:	Third stage:
$T_{41} = x_1 + x_3$		$X_{41} = T_{41} + T_{42}$
$T_{42} = x_2 + x_4$		$X_{42} = T_{42} + M_{41}$
$T_{43} = x_1 - x_3$		$X_{43} = T_{41} - T_{42}$
$T_{44} = x_4 - x_2$	$M_{41} = j \cdot T_{44}$	$X_{44} = T_{42} - M_{41}$.

The addition T_{42} is performed by the same adder that is used for T_{51} and T_{31}. The shared additions in the first stage are shown in Table I.

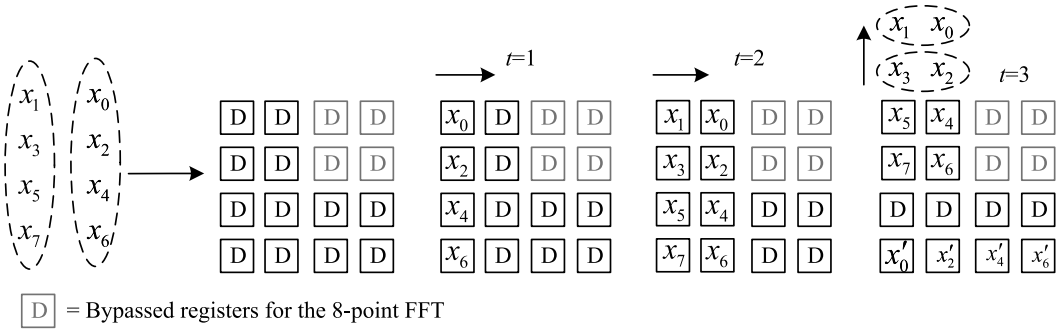

Fig. 4. Data flow for an 8-point FFT.

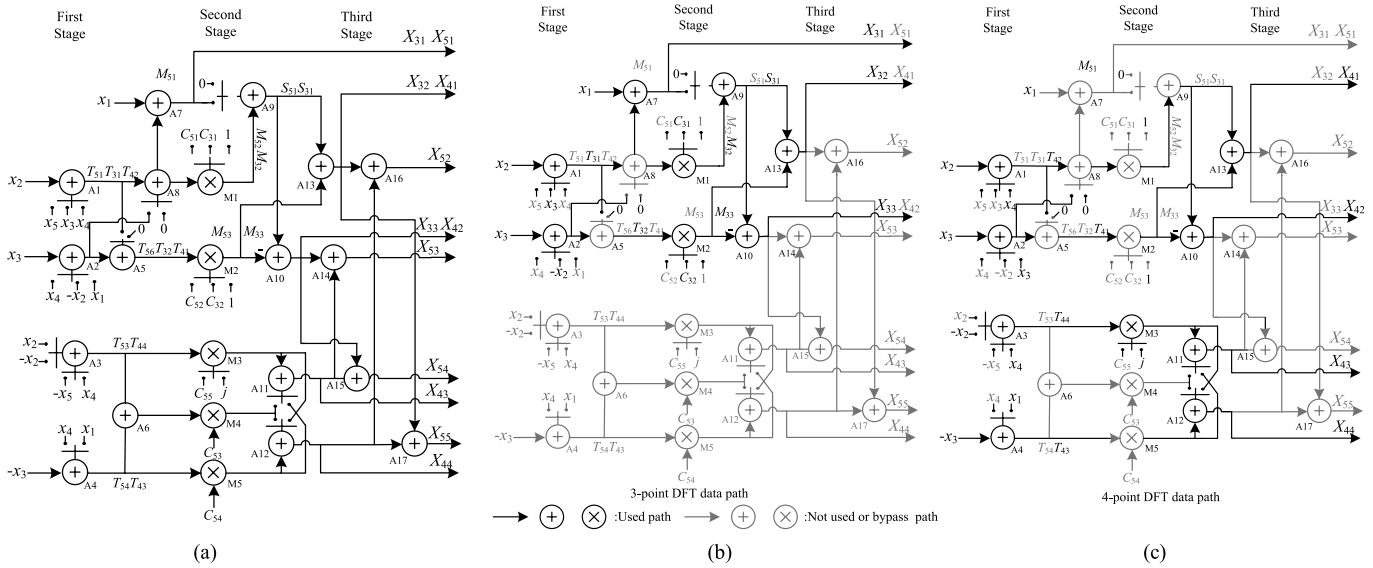

Fig. 5. Proposed hardware-efficient multiradix DFT block. (a) Complete multiradix FFT structure (all adders and multipliers used the 5-point WFTA). (b) Configurations for the 3-point WFTA paths. (c) Configurations for the 4-point WFTA data paths.

Finally, note that a 2-point DFT only requires two operations, x_1+x_2 and x_1-x_2, which are implemented using adders A1 and A3. Thus, the proposed hardware-efficient 2-, 3-, 4-, and 5-point WFTA module only requires the cost of a 5-point WFTA together with eight additional 3:1 multiplexers and 16 additional 2:1 multiplexers, as shown in Fig. 5. So, the total cost of the proposed EDEM consists of 44 real multipliers (RMs), 108 real adders (RAs), the multiplexers, and the 5×5 array of delay elements. In a similar fashion, one can obtain shared multiplications in the second stage and shared additions in the third stage, as shown in Table II.

IV. Data Routing Scheme

A. Address Generator

The PFA is first applied to factorize an N-point DFT into coprime DFTs in the proposed DFT processor. Then, (4) and (5) are applied to generate the data access addresses.

5-point WFTA	First Stage: $T_{51}=x_2+x_5, T_{55}=T_{51}+T_{52}$ $T_{52}=x_3+x_4, T_{56}=T_{51}-T_{52}$ $T_{53}=x_2-x_5, T_{57}=T_{53}+T_{54}$ $T_{54}=x_4-x_3, M_{51}=x_1+T_{55}$	Second Stage: $M_{52}=C_{51}\cdot T_{55}$ $M_{53}=C_{52}\cdot T_{56}$ $M_{54}=C_{53}\cdot T_{57}$ $M_{55}=C_{54}\cdot T_{54}$ $M_{56}=C_{55}\cdot T_{53}$	Third Stage: $S_{51}=M_{51}+M_{52}\ X_{51}=M_{51}$ $S_{52}=S_{51}+M_{53}\quad X_{52}=S_{52}+S_{53}$ $S_{53}=M_{54}-M_{55}\ X_{53}=S_{54}+S_{55}$ $S_{54}=S_{51}-M_{53}\quad X_{54}=S_{54}-S_{55}$ $S_{55}=M_{54}+M_{56}\ X_{55}=S_{52}-S_{53}$
3-point WFTA	First stage: $T_{31}=x_2+x_3$ $T_{32}=x_3-x_2$ $M_{31}=x_1+T_{31}$	Second stage: $M_{32}=C_{31}\cdot T_{31}$ $M_{33}=C_{32}\cdot T_{32}\ S_{31}=M_{31}+M_{32}$	Third stage: $X_{31}=M_{31}$ $X_{32}=S_{31}+M_{32}$ $X_{33}=S_{31}-M_{33}.$

TABLE I
SHARED ADDITIONS IN THE FIRST STAGE

WFTA / ADDER	5-point WFTA	3-point WFTA	4-point WFTA
A1	$T_{51}=x_2+x_5$	$T_{31}=x_2+x_3$	$T_{42}=x_2+x_4$
A2	$T_{52}=x_3+x_4$	$T_{32}=x_3-x_2$	$T_{41}=x_1+x_3$
A3	$T_{53}=x_2-x_5$	N/A	$T_{42}=-x_2+x_4$
A4	$T_{54}=-x_3+x_4$	N/A	$T_{41}=x_1-x_3$

TABLE II
SHARED MULTIPLICATIONS IN THE SECOND STAGE

WFTA / MUL	5-point WFTA	3-point WFTA	4-point WFTA
M1	$M_{52}=C_{51}\cdot T_{55}$	$M_{52}=C_{31}\cdot T_{31}$	$M_{41}=j\cdot T_{44}$
M2	$M_{53}=C_{52}\cdot T_{56}$	$M_{52}=C_{32}\cdot T_{32}$	N/A
M3	$M_{56}=C_{55}\cdot T_{54}$	N/A	N/A

After that, the CTA factorizes 2^{α}, 3^{β}, and 5^{γ}-point DFTs into smaller DFTs if required. For convenience of discussion, we give an example of a 1296-point DFT. The other sizes of DFT are obtained by a similar method.

The 1296-point DFT is factorized to 16-, 9-, and 9-point DFTs according to Fig. 1. Suppose the 16-point DFT is computed in the first stage, the 9-point DFT in the second stage, and another 9-point DFT in the third stage. According to (4), the decomposition equations are given by

$$\begin{cases} n = (81n_1 + 1216\tilde{n}_2) \bmod 1296 & n_1, k_1 = 0, 1, \ldots, 15 \\ k = (81k_1 + 16\tilde{k}_2) \bmod 1296 & \tilde{n}_2, \tilde{k}_2 = 0, 1, \ldots, 80 \end{cases} \tag{9}$$

$$\begin{cases} \tilde{n}_2 = (9n_2 + n_3) \bmod 81 & n_2, k_2 = 0, 1, \ldots, 8 \\ \tilde{k}_2 = (k_2 + 9k_3) \bmod 81 & n_3, k_3 = 0, 1, \ldots, 8. \end{cases} \tag{10}$$

The index mapping for the first stage is shown in Table III. The data samples in each row of Table III are used to perform 16-point DFTs. Then, the data in each column of Table III are used to calculate 81-point DFTs. Each of these 81-point DFTs is decomposed into 9-point by 9-point DFT by (9).

The 16-point DFT is decomposed into 4-point DFT by CTA. Hence, the index mapping equations are given by

$$\begin{cases} n = (81 \cdot (4n_1 + n_2) + 1216\tilde{n}_2) \bmod 1296 \\ \qquad\qquad n_1, n_2, k_1, k_2 = 0, 1, 2, 3 \\ k = \left(81 \cdot (4k_1 + k_2) + 16\tilde{k}_2\right) \bmod 1296 \\ \qquad\qquad \tilde{n}_2, \tilde{k}_2 = 0, 1, \ldots, 80. \end{cases} \tag{11}$$

The data are output concurrently to perform the 4-point DFT with indices $n_1 = 0, 1, 2, 3$. Then, the sequence of index mapping patterns is shown in Table IV. The values of n are used as addresses in the memory banks. The hardware implementation of the AG is shown in Fig. 6(a).

The mod N unit is implemented by the subtractor and multiplexer to generate the address the range $[0, 2N-1]$.

TABLE III
INDEX MAPPING FOR THE FIRST STAGE

	n_1=0	n_1=1	...	n_1=15
$\tilde{n}_2=0$	$x[0]$	$x[81]$	...	$x[1215]$
$\tilde{n}_2=1$	$x[1216]$	$x[1]$	...	$x[1135]$
...	...	...	...	...
$\tilde{n}_2=80$	$x[80]$	$x[161]$	...	$x[1295]$

TABLE IV
MAPPING INDEX PATTERNS FOR THE SECOND STAGE

Time	n_1=0	n_1=1	n_1=2	n_1=3		
$t=1$	$x[0]$	$x[324]$	$x[648]$	$x[972]$	n_2=0	$\tilde{n}_2=0$
$t=2$	$x[81]$	$x[405]$	$x[729]$	$x[1053]$	n_2=1	
			⋮			
$t=5$	$x[1216]$	$x[244]$	$x[568]$	$x[892]$	n_2=0	$\tilde{n}_2=1$
$t=6$	$x[1]$	$x[325]$	$x[649]$	$x[973]$	n_2=1	
			⋮			
$t=324$	$x[323]$	$x[647]$	$x[971]$	$x[1295]$	n_2=3	$\tilde{n}_2=80$

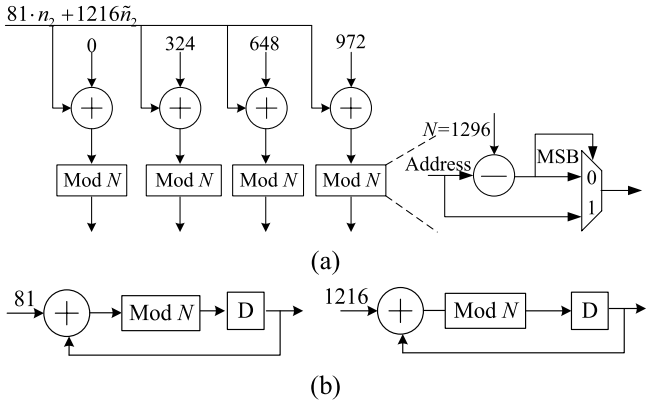

Fig. 6. Implementation of the AG.

The addresses, $81 \cdot n_2$ and $1216\tilde{n}_2$, are generated by the accumulator, as shown in Fig. 6(b).

The initial addresses, 0, 324, 648, and 972, are calculated by the accumulator. Hence, we only keep values of addresses 324, 81, and 1216 to perform the 16-point DFT in the first stage. In the remaining stages, the addresses can be obtained from (4) and (5) and in the final form of

$$n = (n_1 P_1 + n_2 P_2 + \cdots + n_m P_m) \bmod N \tag{12}$$

which can also be implemented by a similar scheme to that of Fig. 6. The range of m determines the number of adders, and $n_1 P_1, n_2 P_2, \ldots, n_m P_m$ can be implemented by the accumulators. The values that are required to be stored are $P_1, P_2, \ldots, P_m$.

B. Coprime Accessing Technology

Data with the address given by (11) from $n_1 = 0, 1, \ldots, N_1-1$ are read in parallel to perform N_1-point DFTs concurrently in each stage. We propose a coprime access technology to avoid data accessing conflicts in the memory arrays.

According to (11), the data accessing address vector is given by

$$\mathbf{A_c} = \Big([0, P_1, 2P_1, \ldots, (N_1-1)P_1] + n_2P_2 + \cdots + n_mP_m\Big) \bmod N. \quad (13)$$

Since the values of $P_1, P_2, \ldots, P_m$ are multiples of the factors two, three, or five, Lemma 1 can be used to avoid conflicts.

Lemma 1: Define a set

$$\mathbf{x}=\{x(i), x(i)=t+i\cdot s \,|i=0,1,\ldots,D-1\} s\in[1, N_1] \quad (14)$$

where s is coprime with D. Set $\mathbf{y}$ is given by

$$\mathbf{y}(i) = \mathbf{x}(i) \bmod D.$$

Then, for any element in $\mathbf{y}$

$$\mathbf{y}(i) \neq \mathbf{y}(j) \quad i \neq j.$$

The proof can be found in the Appendix.

According to Lemma 1, we have, if L is coprime with N_1 and $L > N_1$, the remainder vector $\mathbf{R}$ of $\mathbf{A_c}$ divided by P satisfies

$$\mathbf{R} = \mathbf{A_c} \bmod L$$
$$\mathbf{R}(i) \neq \mathbf{R}(j) \quad i \neq j. \quad (15)$$

Thus, the index vector A_c is converted to 2-D indices, a remainder vector $\mathbf{R}$ and a quotient vector $\mathbf{Q}$

$$\mathbf{A_c} = \mathbf{Q}\cdot L + \mathbf{R}. \quad (16)$$

The elements in the remainder vector $\mathbf{R}$ are different from each other. Hence, $\mathbf{R}$ is used as a RAM index vector to avoid accessing conflicts. The quotient vector $\mathbf{Q}$ is used as the accessing address for each RAM bank.

Since P_1 is a multiple of three, four, or five, we select L equal to seven to satisfy Lemma 1. K equals to N_1 for the N_1-point DFT. The structure of the CAT is shown in Fig. 7, which contains four parts: address converter, $K \rightarrow L$ router, $L \rightarrow K$ router, and RAM banks. The address vector $\mathbf{A_c}$ is reformulated into $\mathbf{R}$ and $\mathbf{Q}$ by the address converter, which is realized by a mod-7 calculator. Then, data are routed by the $K \rightarrow L$ router according to $\mathbf{R}$. The data from port i are sent to port $\mathbf{R}(i)$. The $L \rightarrow K$ data router is implemented by K instances of L:1 multiplexer. The $K \rightarrow L$ data router, which performs the inverse operation, is implemented by L instances of K:1 multiplexer. The accessing addresses are equal to the quotient vector $\mathbf{Q}$, and vice versa for writing data.

We also use the first stage of a 1296-point DFT as a case study. At $t = 2$, the accessing addresses are {81, 405, 648, 972}. After the address converter, the addresses are converted to the remainders {4, 6, 1, 3} and quotients {11, 57, 104, 150}. Then, $x[81]$, $x[405]$, $x[729]$, and $x[1053]$ are read from RAM-4, RAM-6, RAM-1, and RAM-3 with address 11, 57, 104, and 150. Hence, an $K \rightarrow L$ data router is required to move the data into correct positions. The set of memory arrays consists of seven RAMs of depth $\lceil N/7 \rceil$.

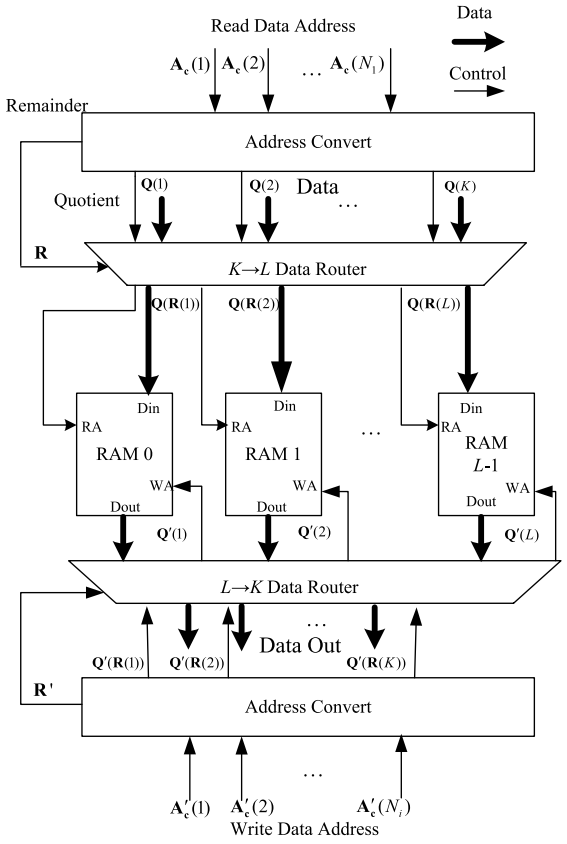

Fig. 7. Structure of the proposed CAT.

TABLE V
CCs OF PROPOSED DESIGN

Radix Size	Radix-25	Radix-16	Radix-9	Radix-8
Computation Cycles	5	4	3	2
Radix Size	Radix-5	Radix-4	Radix-3	Radix-2
Computation Cycles	1	1	0.5*	0.5

* 0.5 means two 3-point FFTs or two 2-point FFTs are computed in one cycle

C. Control Unit

The control unit prestores the DFT size of the EDEM and the initial value of the accessing address in each iteration process. For example, the control unit sends signals to the EDEM to perform the constituent DFTs to calculate the 1296-point DFT. It also sends the initial values to the AG so that the corresponding data accessing addresses are generated. Approximately 1035 initial values are prestored in the control unit for 34 different DFT sizes. Since the PFA algorithm is applied in our method, the twiddle factor multiplication is free between radix-2, -3, and 5. The total number of twiddle factors prestored is 200 for the 34 types of DFTs.

V. PERFORMANCE ANALYSIS

A. Computation Cycle

The number of CCs for the GHR is related to the processing speed of the EDEM. The CCs of the proposed design are shown in Table V.

TABLE VI
NUMBER OF REQUIRED CCS FOR LTE APPLICATIONS

Length	Factorization	Computation Cycles					
		Proposed		GMR [4]		DFT IP Core [13]	
		C.C	Lat	C.C	Lat	C.C	Lat
12	3×4	11	22	13	25	62	75
144	9×16	61	72	168	180	444	457
216	9×3×8	162	173	224	236	719	732
384	16×8×3	256	267	512	524	1145	1158
480	16×2×3×5	416	427	616	628	1496	1509
540	9×3×4×5	513	524	783	795	1760	1773
648	9×9×4×2	594	605	1188	1200	2212	2225
864	16×2×9×3	864	875	1512	1524	2842	2265
1200	16×3×25	740	751	1480	1492	3779	3792
1296	16×9×9	1188	1199	2376	2388	4318	4331

C.C: Computation Cycle
Lat: Latency

For the N-point FFT in (2), the required number of CCs is given by

$$\begin{aligned} C.C &= 4a_1 \cdot 16^{a_1-1} \cdot U_{16} + 2a_2 \cdot 8^{a_2-1} U_8 \\ &\quad + a_3 \cdot 4^{a_3-1} U_4 + a_4 \cdot 2^{a_4-2} U_2 \\ &\quad + 3b_1 \cdot 9^{b_1-1} U_9 + b_2 \cdot 3^{b_2-1} U_3/2 \\ &\quad + 5c_1 \cdot 25^{c_1-1} U_{25} + c_2 \cdot 5^{c_2-1} U_5 \\ U_i &= N/i^k \end{aligned} \tag{17}$$

where

$$k = \begin{cases} a_1 \text{ when } i = 16;\ a_2 \text{ when } i = 8 \\ a_3 \text{ when } i = 4\ ;\ a_4 \text{ when } i = 2 \\ b_1 \text{ when } i = 9\ ;\ b_2 \text{ when } i = 3 \\ c_1 \text{ when } i = 25;\ c_2 \text{ when } i = 5. \end{cases}$$

The required number of CCs for LTE is shown in Table VI. The proposed method requires the fewest CCs compared with methods in [4] and [13]. Moreover, the CCs for the proposed method are less than or equal to the length of the FFT in LTE. Hence, the proposed method supports the CF of data without the need for frequency doubling.

If two memory banks are used, the flow of data is shown in Fig. 8 for different methods. We can observe that data loss occurs due to memory accessing conflicts for the method in [4] and [13], while the proposed architecture is conflict free.

B. Hardware Cost

We have implemented the proposed DFT processor, GMR [4], and Xilinx IP core [13] on a Xilinx Virtex-5 FPGA to compare the hardware costs. The results are shown in Table VII. For the proposed method, the 3-, 4-, and 5-point FFT modules require ten RMs and 34 RAs. The EDEM contains two of these modules, with two twiddle factor multiplication modules, so that the total hardware cost is 34 RMs and 108 RAs.

The proposed method requires seven RAMs with $2N$ words to support the CF of data. The maximum throughput rate (TR) is 2.36 and the minimum TR is one for the proposed design.

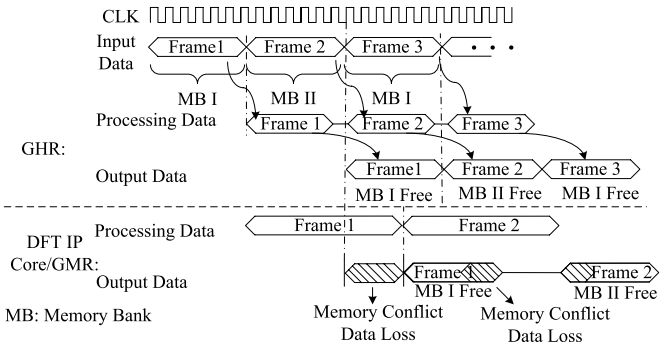

Fig. 8. CF of data in the proposed DFT processor and memory conflicts in previous designs.

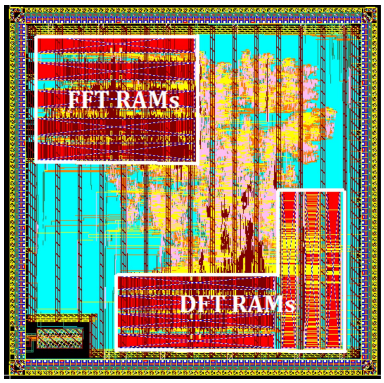

Fig. 9. Layout plot of the ASIC implementation.

Hence, the proposed method supports the CF data at 1× clock rate, which is not possible for the methods in [4] and [13].

We can also use a speed–area ratio to compare the performance of the FFT processors, which is defined as

$$\text{Speed–Area} = \frac{\text{Throughput rate}}{\text{Logic cost}}. \tag{18}$$

To provide a fair comparison of the hardware costs, the multiplier is implemented using lookup tables rather than with an IP core. According to Table VIII, the speed–area ratio for the proposed architecture is three times that of the FFT IP core and nearly two times that of the GMR.

We have also implemented the proposed design in a 0.18-μm CMOS technology. The chip contains two independent cores, called DFT core and FFT core. The DFT core supports all 34 DFT sizes. The FFT core supports FFT sizes of 128/256/512/1024/2048. A summary of the results is given in Table VIII. The proposed FFT core requires a lower gate count than the processor in [8] based on the same TR. As shown in Table VIII, the proposed design has the highest speed–area ratio among the three FFT cores. The normalized energy-efficiency value for the chip is larger (i.e., worse) than those of [8] and [9]. However, the power consumption measurement includes the power used by both the DFT core and the FFT core running concurrently. Moreover, the proposed method supports a much larger set of transform sizes than the designs of [8] and [9]. A layout plot is shown in Fig. 9. The testing platform is shown in Fig. 10.

TABLE VII
HARDWARE COMPARISONS

	RMs	RAs	Memory Size	MAX T.R.	MIN T.R.	Number of LUT	Speed Area Ratio	To support C.F. of data	Critical path	Frequency
Proposed	44	108	7 Block RAMs total 2N words	2.36	1	7791[*1]+16016[*2]	70.5	1x	1 real multiplier	122.88Mhz
GMR [4]	26	82	5 Block RAMs total 2N words	0.96	0.54	6687[*1]+9464[*2]	46.4	2x	1 real multiplier	122.88Mhz
IP Core [13]	16	79	7 Block RAMs total 2N words	0.32	0.19	5746[*1]+5824[*2]	21.6	4x	1 real multiplier	122.88Mhz

Note: *1 The cost of control unit, adder, address generator and data routing unit. *2 The cost of the multiplier, suppose one multiplier occupy 364 LUTs. N=1296 for the LTE application

TABLE VIII
SUMMARY OF THE DFT/FFT ASIC CHIP

	DFT Core	FFT Core	FFT Core[8]	FFT Core[9]
Support FFT Sizes	From 12 to 1296; All 34 LTE sizes.	128/256/512/1024/2048	128/256/512/1024/1536/2048	128/256/512/1024/1536/2048
Clock Frequency	122.88 Mhz		1.25-20Mhz	35Mhz
DEM Sizes	5x5	4x4	Null	Null
Support FFT Radixes	Radix-25,16,9,8,5,4,3,2	Radix-16,8,4,2	Raidx-16/3	Radix-4/3
RAM Banks	7 RAMs	5 RAMs	Null	8 RAMs
Throughput rate	1X		1X	0.25X
Throughput	122.88 M Symbols/s		20M Symbol/s	8.72M Symbols
Support Continues Flow of data	Yes		Yes	No
Core Area (Normalized)	5x5 mm^2		1.25x1.1 mm^2(3.4x3 mm^2)	1.932 mm^2
Gate Count	798K		1,100K	98K
	482K	316K		
RAM Size	41kb Register File	64kb Register File	48kb Register File	64kb Register File
Word-Length[1]	16Bit@23dB		12Bit@22dB	16 Bits@23dB
Power Consumption	320mW		8.55mW	11.29mW
Technology	SMIC 0.18 μm		65nm	0.18 μm
Speed Area Ratio[2]	2.7	6.67	1.9	5.1
Normalized Energy-Efficency[3][16]	2.6		1.92	1.29

1. Input SNR=25dB

2. Speed-Area=$\frac{\text{Throughput rate}}{\text{Gate Count}}$

3. Normailized Energy-Efficiency=$\frac{\text{Power Consumption}}{\text{Throughputput}} \cdot \frac{0.18}{\text{Technology}} \cdot \frac{V_{dd}}{V'_{dd}}$

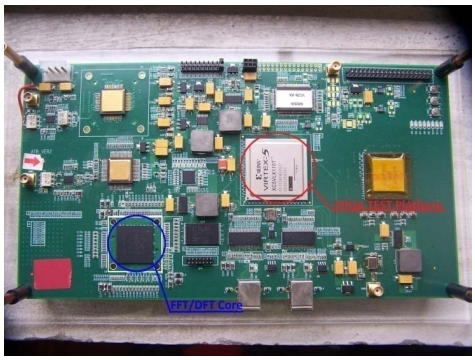

Fig. 10. Functional testing platform for the fabricated ASIC chip.

VI. CONCLUSION

We have proposed a hardware-efficient GHR reconfigurable FFT processor. Compared with previous FFT processors, the GHR design has several important advantages. First, it supports GMR DFTs having eight radices and 34 DFT lengths. Second, the radix size is as high as radix-25, which improves the processing speed and the hardware efficiency. In addition, the proposed GHR is reconfigurable and it supports a continuous data flow within a single clock domain. FPGA and ASIC implementation results and comparisons have also been presented.

APPENDIX

Lemma 1: Define a set

$$\mathbf{x} = \{x_i, x_i = t + i \cdot s\} \tag{19}$$

where $0 \le i < D$ and s is coprime with D, and suppose set $\mathbf{y}$ is given by

$$\mathbf{y}(i) = \mathbf{x}(i) \bmod D. \tag{20}$$

Then, for any element in $\mathbf{y}$

$$\mathbf{y}(i) \neq \mathbf{y}(j) \quad i \neq j. \tag{21}$$

Proof: Suppose $\exists i \neq j$ and $j > i$ make $\mathbf{y}(i) = \mathbf{y}(j)$, according to (21), $\mathbf{x}$ can be represented as

$$\begin{aligned} \mathbf{x}(i) &= \mathbf{y}(i) + D \cdot k_i \\ \mathbf{x}(j) &= \mathbf{y}(j) + D \cdot k_j. \end{aligned} \tag{22}$$

$\mathbf{x}(i)$ and $\mathbf{x}(j)$ satisfy the following relationship according to the definition of $\mathbf{x}$:

$$\mathbf{x}(j) = \mathbf{x}(i) + s \cdot (i - j). \tag{23}$$

Substitute (23) into (24), and according to $\mathbf{y}(i) = \mathbf{y}(j)$

$$D \cdot k_j = D \cdot k_i + s \cdot (i - j) \tag{24}$$
$$D \cdot (k_j - k_i) = s \cdot (i - j).$$

Since s is coprime with D, (25) holds when

$$i - j = P \cdot D \qquad P \text{ is natural number}$$
$$i = P \cdot D + j > D \tag{25}$$

which does not satisfy the constraint $0 \leq i < D$. Hence, for any element in $\mathbf{y}$, (22) is satisfied.

REFERENCES

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, 1965.

[2] (2012). *The Digital Video Broadcasting Standard* [Online]. Available: http://www.dvb.org

[3] Y.-N. Chang, "An efficient VLSI architecture for normal I/O order pipeline FFT design," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 12, pp. 1234–1238, Dec. 2008.

[4] C.-F. Hsiao, Y. Chen, and C.-Y. Lee, "A generalized mixed-radix algorithm for memory-based FFT processors," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 1, pp. 26–30, Jan. 2010.

[5] B. G. Jo, and M. H. Sunwoo, "New continuous-flow mixed-radix (CFMR) FFT processor using novel in-place strategy," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 5, pp. 911–919, May 2005.

[6] R. Radhouane, P. Liu, and C. Modlin, "Minimizing the memory requirement for continuous flow FFT implementation: Continuous flow mixed mode FFT (CFMM-FFT)," in *Proc. IEEE ISCAS*, May 2000, pp. 116–119.

[7] L. G. Johnson, "Conflict free memory addressing for dedicated FFT hardware," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 5, pp. 312–316, May 1992.

[8] C.-H. Yang, T.-H. Yu, and D. Markovic, "Power and area minimization of reconfigurable FFT processors: A 3GPP-LTE example," *IEEE J. Solid-State Circuits* , vol. 47, no. 3, pp. 757–768, Mar. 2012.

[9] S.-Y. Peng, K.-T. Shr, C.-M. Chen, and Y.-H. Huang, "Energy-efficient 128∼2048/1536-point FFT processor with resource block mapping for 3GPP-LTE system," in *Proc. IEEE ICGCS*, Jun. 2010, pp. 14–17.

[10] C.-P. Hung, S.-G. Chen, and K.-L. Chen, "Design of an efficient variable-length FFT processor," in *Proc. IEEE ISCAS*, May 2000, pp. 833–837.

[11] 3GPP. (2008). *Evolved universal terrestrial radio access (E-UTRA); physical channels and modulation*. Tech. Rep. ETSI TS 136 211 [Online]. Available: http://www.3gpp.org

[12] C. Cheng and K. K. Parhi, "Low-cost fast VLSI algorithm for discrete Fourier transform," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 4, pp. 791–806, Apr. 2007.

[13] *Discrete Fourier Transform V3.1 Product Specification*, Xilinx, San Jose, CA, USA, Mar. 2011.

[14] I. J. Good, "The interaction algorithm and practical Fourier series," *J. Roy. Statist. Soc.*, vol. 20, no. 2, pp. 361–372, 1958.

[15] H. Silverman, "An introduction to programming the Winograd Fourier transform algorithm (WFTA)," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 25, no. 2, pp. 152–165, Apr. 1977.

[16] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1754–1765, Jul. 2011.

Jienan Chen (S'10) received the B.S. degree in electrical and information engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2007, where he is currently pursuing the Ph.D. degree from the National Key Laboratory of Science and Technology on Communications.

He is a Visiting Student with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. His current research interests include very large scale integration circuit designs, low-power circuit designs, and stochastic computation-based system designs.

Jianhao Hu (M'10) received the B.E. and Ph.D. degrees in communication systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1993 and 1999, respectively.

He has been a Professor with the National Key Laboratory of Communication, UESTC, since 2005. He joined the City University of Hong Kong, Hong Kong, from 1999 to 2000 as a Post-Doctoral Researcher. His current research interests include high-speed and low-power digital signal processing technology, very large scale integration, NoC, wireless communications, and software radio. He served as a Senior System Engineer with the 3G Research Center, University of Hong Kong, from 2000 to 2004.

Shuyang Lee received the B.E. and M.S. degrees in communication and information engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2010 and 2013, respectively.

He is currently with SiChuan AirLine Company, Chengdu, for .NET and JSP architecture design. His current research interests include residue number system and low-power consumption circuits design.

Gerald E. Sobelman (M'81–SM'03) received the B.S. degree in physics from the University of California, Los Angeles, CA, USA, and M.S. and Ph.D. degrees in physics from Harvard University, Cambridge, MA, USA.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis. He was a Post-Doctoral Researcher with the Rockefeller University, New York, NY, USA, and he has held Senior Engineering positions with Sperry Corporation, New York, and Control Data Corporation, Minneapolis, MN, USA. He has served as the Director of Graduate Studies for the Graduate Program in computer engineering with the University of Minnesota. He has authored or co-authored more than 120 technical papers and one book, and he holds 12 U.S. patents. He has developed and presented short courses on digital VLSI design at several industrial sites. His current research interests include the VLSI circuit and system design for applications in communications and signal processing.

Prof. Sobelman has been a Distinguished Lecturer of the IEEE Circuits and Systems Society, and he has served on the technical program committees for the IEEE ISCAS, the IEEE SOCC, and the IEEE ICCSC. He was a Chair with the Technical Committee on Circuits and Systems for Communications of the IEEE Circuits and Systems Society, and he has served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: REGULAR PAPERS and for the IEEE Signal Processing Letters. He was a Local Arrangements Chair for the 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing. In addition, he has chaired many sessions at international conferences in the areas of communications and VLSI design.