48-Mode Reconfigurable Design of SDF FFT Hardware Architecture Using Radix-3^2 and Radix-2^3 Design Approaches

Xin-Yu Shih, Yue-Qu Liu, and Hong-Ru Chou

***Abstract*— In this paper, we propose a reconfigurable (RC) fast Fourier transform (FFT) design in a systematic design scheme. The RC design bricks are mainly proposed to arbitrarily concatenate to support FFT-point required. Meanwhile, we show three developed design techniques, including six-type RC processing element, systematic first-in first-out reuse arrangement, and section-based twiddle factor generator to elaborate our FFT design. In a design/implementation example, it can support up to 2187 FFT-point manipulation and 48 RC modes. It also supports 32 operating modes defined in 3GPP-LTE standard. In application-specified integrated circuit implementation with TSMC 90-nm CMOS technology, our design work occupies a core area of 1.664 mm^2 and consumes 35.2 mW under maximal clock frequency of 188.67 MHz. This paper also has outstanding design performance in terms of speed-area ratio and power-frequency ratio for comparison reference.**

***Index Terms*— 3GPP-LTE, fast Fourier transform (FFT), reconfigurable (RC), single-path delay feedback (SDF).**

I. Introduction

FAST Fourier transform (FFT) is a widely used and popular circuit design technique in the communication fields. It is a reduced form of discrete Fourier transform (DFT) in mathematics nature. A regular hardware-oriented design methodology, single-path delay feedback (SDF) FFT [1], is developed in 1996, mostly focusing on the radix-2 FFT design [2]–[6]. It has already become an interesting main research direction both in academic domain and modern industry world. Later, radix-4 [7]–[9] and radix-8 [10] FFT hardware designs are discussed to expand the similar design concepts. Besides, in order to achieve lower computation complexity, radix-2^2 [11]–[13], radix-2^3 [14]–[20], radix-2^4 [21], [22], and radix-2^k [23], [24] FFT circuits are developed in sequence. Furthermore, it moves the trends to apply the analogous approaches on radix-3 [25]–[27] and radix-6 [28], [29] FFT hardware systems.

Manuscript received November 3, 2016; revised December 13, 2016 and January 3, 2017; accepted January 13, 2017. Date of publication February 14, 2017; date of current version May 25, 2017. This work was supported by the Ministry of Science and Technology, R.O.C., under Grant MOST 104-2218-E-110-010-MY2. This paper was recommended by Associate Editor Y. Pu.

X.-Y. Shih was with Mediatek, Hsinchu 300, Taiwan. He is now with the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 804, Taiwan (e-mail: xyshih@mail.ee.nsysu.edu.tw).

Y.-Q. Liu and H.-R. Chou are with National Sun Yat-sen University, Kaohsiung 804, Taiwan (e-mail: tom@snsd.ee.nsysu.edu.tw; bob.chou@snsd.ee.nsysu.edu.tw).

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/TCSI.2017.2654451

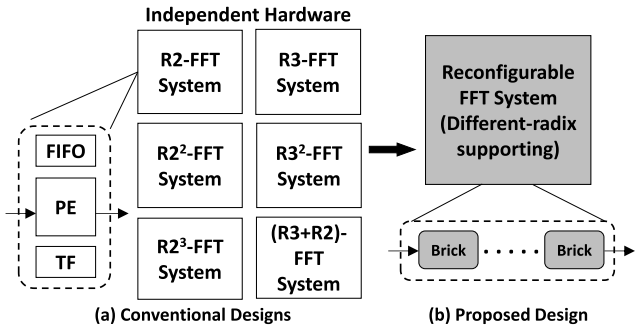

Fig. 1. Comparison of conventional designs and the proposed design.

In addition to supporting only one radix in an individual FFT system, several state-of-the-art works [30]–[36] are used to provide mixed-radix operation. In an independent FFT system, certain portion of circuits are responsible for one radix whereas other parts are utilized for another radix. In other words, it can make different-radix concatenation/combination possible. But one particular processing element (PE) or one specified first-in first-out (FIFO) storage is still served for only one-radix handling, which causes less circuit flexibility and larger design efforts. Instead, in order to overcome the difficulties, we can demonstrate the proposed design ideas, as shown in Fig. 1.

The conventional SDF FFT system design only provides a single-radix SDF FFT operation without any circuit flexibility [see Fig. 1(a)]. For another different-radix supporting needed, it should develop a completely new hardware architecture and take large design time efforts to implement. Take something for examples. If there are S different-radix computations required, most of the design approaches in the current research literature need to develop S different independent hardware circuits, such as $S = 6$. Instead, we take more insights into three basic composition, including PE, FIFO, and twiddle factor (TF) parts. As for different-radix supporting, the according circuits can be developed in a reuse and systematic way. Therefore, we propose a reconfigurable (RC) SDF FFT system [see Fig. 1(b)]. All of the subcircuits are encapsulated as several RC design bricks, arbitrarily concatenated to build up a system with target FFT-point on demand.

The rest of this paper is organized as follows. Section II shows that we deduce SDF FFT algorithms with different-radix bases, which become the fundamental hardware design of our proposed RC FFT system. Section III presents our

proposed RC hardware architecture with three developed design techniques, including: 1) six-type RC PE (6T-RC-PE); 2) systematic FIFO reuse arrangement (SFRA); and 3) section-based TF generator (STFG). Section IV demonstrates our application-specified integrated circuit (ASIC) chip implementation with both synthesis and post-automatic place and route (APR) results. The design comparison with other works is shown in Section V. Finally, Section VI concludes our design work.

II. Deduced SDF FFT Algorithm With Different-Radix Bases

In this section, we want to deduce SDF FFT algorithms with different-radix bases. First, assume that an SDF FFT system has $N = r^P$ points, where r and P are positive integers. In the following, we deduce a general-form algorithm with radix-r^x ($x > 1$)-based mathematical equations. Also, we verify that applying radix-r^x ($x > 1$) basis has better performance than only applying radix-r basis in terms of hardware resource, such as the number of multipliers required and TF table entry size.

DFT equation is originally defined as

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

Assume that n and k are represented with alternative forms as (2) and (3), respectively. Also, we use radix-r^2 basis to derive

$$n = \frac{N}{r} n_1 + \frac{N}{r^2} n_2 + n_3 \tag{2}$$

$$k = k_1 + r k_2 + r^2 k_3 \tag{3}$$

where $r > 1$ and $\{n_1, n_2, n_3, k_1, k_2, k_3\}$ are nonnegative integers.

Replace (1) with (2) and (3) simultaneously

$$\begin{aligned} X[k_1 + rk_2 + r^2 k_3] &= \sum_{n_3=0}^{\frac{N}{r^2}} \sum_{n_2=0}^{r-1} \sum\nolimits_{n_1=0}^{r-1} x\left(\frac{N}{r} n_1 + \frac{N}{r^2} n_2 + n_3\right) \\ &\quad \times W_N^{\left(\frac{N}{r} n_1 + \frac{N}{r^2} n_2 + n_3\right)(k_1 + rk_2 + r^2 k_3)} \end{aligned} \tag{4}$$

$$\begin{aligned} &= \sum_{n_3=0}^{\frac{N}{r^2}} \sum_{n_2=0}^{r-1} \left[\sum_{n_1=0}^{r-1} x\left(\frac{N}{r} n_1 + \frac{N}{r^2} n_2 + n_3\right) W_N^{\frac{N}{r} n_1 k_1} \right] \\ &\quad \times W_N^{\left(\frac{N}{r^2} n_2 + n_3\right)(k_1 + rk_2 + r^2 k_3)}. \end{aligned} \tag{5}$$

We can rewrite summation term in (5), $\sum_{n_1=0}^{r-1} x((N/r)n_1 + (N/r^2)n_2 + n_3) W_N^{(N/r)n_1 k_1}$, as (6). It is considered as a general form of *radix-r*-based algorithm

$$B_{\frac{N}{r}}^{k_1}\left(\frac{N}{r^2} n_2 + n_3\right) = \sum_{n_1=0}^{r-1} x\left(\frac{N}{r} n_1 + \frac{N}{r^2} n_2 + n_3\right) W_N^{\frac{N}{r} n_1 k_1}. \tag{6}$$

Also, we rewrite the other TF term in (5) as (7)

$$\begin{aligned} &W_N^{\left(\frac{N}{r^2} n_2 + n_3\right)(k_1 + rk_2 + r^2 k_3)} \\ &\quad = W_N^{\frac{N}{r^2} n_2 k_1} W_N^{\frac{N}{r} n_2 k_2} W_N^{N n_2 k_3} W_N^{n_3 k_1} W_N^{r n_3 k_2} W_N^{r^2 n_3 k_3} \\ &\quad = W_N^{\frac{N}{r^2} n_2 (k_1 + rk_2)} W_N^{n_3 (k_1 + rk_2)} W_N^{r^2 n_3 k_3}. \end{aligned} \tag{7}$$

Replace (5) with (6) and (7)

$$\begin{aligned} &X[k_1 + rk_2 + r^2 k_3] \\ &\quad = \sum_{n_3=0}^{\frac{N}{r^2}} \left[\sum_{n_2=0}^{r-1} B_{\frac{N}{r}}^{k_1}\left(\frac{N}{r^2} n_2 + n_3\right) W_N^{\frac{N}{r^2} n_2 (k_1 + rk_2)} \right] \\ &\qquad \times W_N^{n_3 (k_1 + rk_2)} W_N^{r^2 n_3 k_3}. \end{aligned} \tag{8}$$

We can rewrite summation term in (8), $\sum_{n_2=0}^{r-1} B_{(N/r)}^{k_1}((N/r^2)n_2 + n_3) W_N^{(N/r^2)n_2(k_1 + rk_2)}$, as (9). It is considered as a general form of *radix-r^2*-based algorithm

$$H(k_1, k_2, n_3) = \sum_{n_2=0}^{r-1} B_{\frac{N}{r}}^{k_1}\left(\frac{N}{r^2} n_2 + n_3\right) W_N^{\frac{N}{r^2} n_2 (k_1)} W_N^{\frac{N}{r^2} n_2 (rk_2)}. \tag{9}$$

Finally, replace (8) with (9)

$$X[k_1 + rk_2 + r^2 k_3] = \sum_{n_3=0}^{\frac{N}{r^2}} \left[H(k_1, k_2, n_3) W_N^{n_3 (k_1 + rk_2)} \right] W_N^{r^2 n_3 k_3} \tag{10}$$

$$= \sum_{n_3=0}^{\frac{N}{r^2}} \left[\left[\sum_{n_2=0}^{r-1} B_{\frac{N}{r}}^{k_1}\left(\frac{N}{r^2} n_2 + n_3\right) W_N^{\frac{N}{r^2} n_2 (k_1)} W_N^{\frac{N}{r^2} n_2 (rk_2)} \right] \times W_N^{n_3 (k_1 + rk_2)} \right] W_N^{r^2 n_3 k_3} \tag{11}$$

$$= \sum_{n_3=0}^{\frac{N}{r^2}} \left[\left[\sum_{n_2=0}^{r-1} B_{\frac{N}{r}}^{k_1}\left(\frac{N}{r^2} n_2 + n_3\right) \boldsymbol{W_{r^2}^{n_2 (k_1 + rk_2)}} \right] \times W_N^{n_3 (k_1 + rk_2)} \right] W_N^{r^2 n_3 k_3} \tag{12}$$

where the particular TF term, $W_{r^2}^{n_2 (k_1 + rk_2)}$, is trivial and independent of the total FFT point, N. The trivial TF term means that the possible values belong to a certain small set, without the full range of N equal segments on a unit circle. The corresponding complex multiplier operation is tremendously simplified if only utilizing the controlled MUXes and adders, instead of the actual multiplier.

Table I presents different-radix bases for dealing with the same N-point SDF FFT system. As for radix-2^2 basis (using $r = 2$ in the previously derived equations), the particular TF term, $W_4^{t1}(t1 = 0, 1)$, only has two kinds of special values. Therefore, radix-2^2 basis only has $(P/2 - 1)$ nontrivial multipliers whereas radix-2 basis has $(P - 1)$ nontrivial multipliers. In other words, the radix-2^2 basis has not only less number

TABLE I
COMPARISON OF DIFFERENT-RADIX BASES FOR THE SAME n-POINT SDF FFT SYSTEM

FFT Point	Type of Radix Base	Number of Stage	Number of Non-Trivial Multiplier	Number of Trivial Multiplier	Trivial Multiplier Value
$N=2^P$	Radix-2	P	P-1	0	None
	Radix-2^2	P/2	P/2-1	P/2	$\{w_4^{t1} \mid t1 = 0, 1\}$
	Radix-2^3	P/3	P/3-1	2P/3	$\{w_4^{t1} \mid t1 = 0, 1\}, \{w_8^{t2} \mid t2 = 0 \sim 3\}$
	Radix-2^4	P/4	P/4-1	3P/4	$\{w_4^{t1} \mid t1 = 0, 1\}, \{w_8^{t2} \mid t2 = 0 \sim 3\}, \{w_{16}^{t3} \mid t3 = 0 \sim 7\}$
$N=3^P$	Radix-3	P	P-1	0	None
	Radix-3^2	P/2	P/2-1	P/2	$\{w_9^{t1} \mid t1 = 0, 1, 2, 4\}$
	Radix-3^3	P/3	P/3-1	2P/3	$\{w_9^{t1} \mid t1 = 0, 1, 2, 4\}, \{w_{27}^{t2} \mid t2 = 0 \sim 8, 10, 12, 14, 16\}$
	Radix-3^4	P/4	P/4-1	3P/4	$\{w_9^{t1} \mid t1 = 0, 1, 2, 4\}, \{w_{27}^{t2} \mid t2 = 0 \sim 8, 10, 12, 14, 16\}$, $\{w_{81}^{t3} \mid t3 = 0 \sim 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52\}$

of complex multiplier, but also smaller total TF table entry size. If taking $r = 3$ design case for further discussion, the number of nontrivial multiplier in radix-3^2 basis is also only $(P/2 - 1)$, which is decreased from that in radix-3 basis, $P - 1$. Furthermore, x is not restricted to 2. We can derive the analogous mathematical equations for extending x to 3, 4, . . ., and so on. As for radix-2^3 basis ($r = 2$ and $x = 3$), two particular TF terms are $W_4^{t1}s(t1 = 0, 1)$ and $W_8^{t2}(t2 = 0-3)$. The number of nontrivial multiplier is further reduced to $(P/3 - 1)$ accordingly. So do the other radix cases listed in Table I. We can prove that less complex multiplier resource for the same N-point SDF FFT system is required while increasing the applied values of r or x.

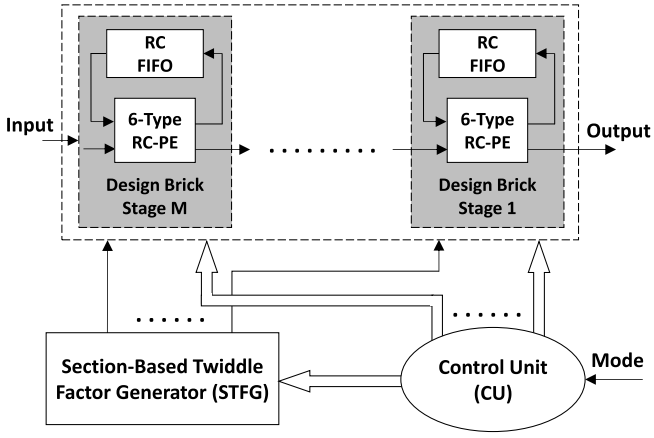

Fig. 2. Proposed RC FFT hardware architecture.

III. PROPOSED RECONFIGURABLE HARDWARE ARCHITECTURE

In this section, we propose an RC FFT hardware architecture and also develop three design techniques as follows:

1) 6T-RC-PE;
2) SFRA on RC FIFO;
3) STFG.

In accordance with the general-form analysis introduced in Section II, different (r, x) combinations can result in various extents of circuit complexity reducing and chip area decreasing. All of the following circuit discussions regarding proposed hardware architecture and three design techniques are not restricted to a certain (r, x) combination. We can make a possible extension to other (r, x) combinations for improving design performance. In order to show a design prototype, we just employ a demonstrated design case, which is mainly based on radix-3^2 and radix-2^3 design approaches.

A. Reconfigurable FFT Hardware Architecture

Fig. 2 shows the proposed RC FFT hardware architecture. First, we can build up an RC design brick with two core parts, including the proposed RC PE (utilizing 6T-RC-PE) and RC FIFO (using SFRA), which are introduced in more detail in Sections III-B and III-C, respectively. In the following, one design brick can be easily concatenated with another to support the number of target brick stage, M. The concatenation is only related to the direct same-wordlength input and output connections between two successive brick stages without any further complex control circuits. This design concept of connected design bricks fully realizes the RC property.

Once completing the design brick connection, STFG is proposed to provide the necessary TF values to the target brick stage. STFG is a distinctive design technique, which is different from the traditional direct lookup table (LUT) approach. We make the detailed discussion and circuit analysis in Section III-D. Finally, the control unit (CU) is to generate the corresponding control signals to all the subcircuits by the operating mode (Mode) asserted.

According to most of the advanced communication standards (such as WiFi, 3GPP-LTE, and next generation 5G systems), industry products, and academic research topics, the applied FFT systems have around several hundreds to two thousands of FFT points. We can choose $M = 4$ due to two reasons. One is to accommodate 32 operating modes defined in 3GPP-LTE applications. $M = 3$ choice (too short case)

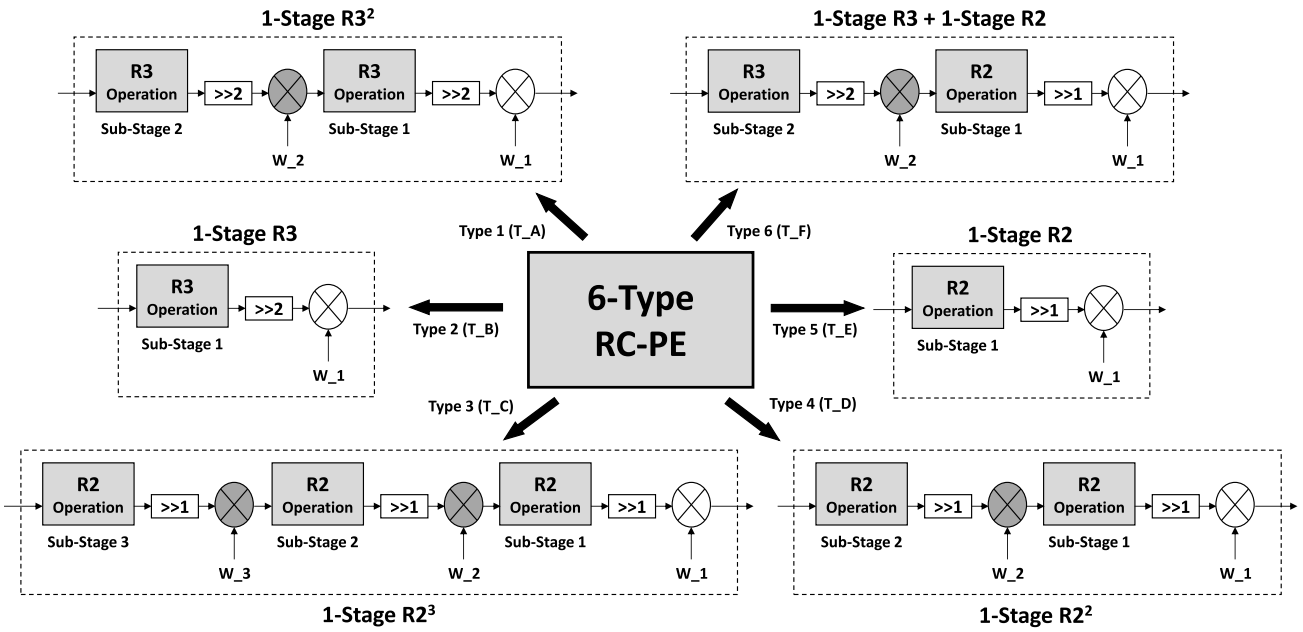

Fig. 3. Design configuration of the proposed 6T-RC-PE.

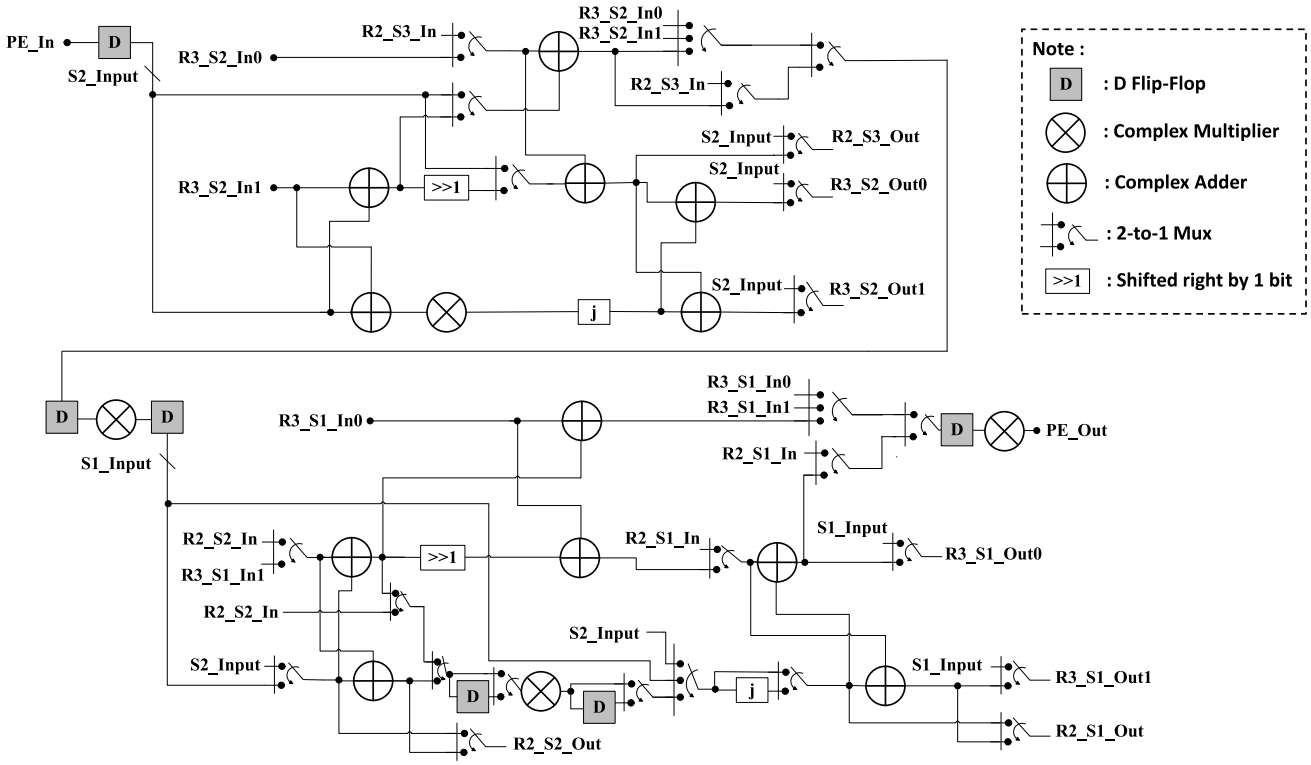

Fig. 4. Proposed hardware circuit of 6T-RC-PE.

supports only up to 729 points whereas $M = 5$ selection (too long case) provides up to 59 049 points. The other is to provide a prototyping design for easier description and clearer design technique illustrating. In case of the future standard changes, we can extend to other M values in a similar manner.

All the combination set ($M = 4$) is listed in Table II. There are totally 48 different operating modes, depending on the valid (a, b) combinations. All of the operating modes are commonly divided into eight categories based on the value of "a," which ranges from 0 to 7. The maximal FFT-point supporting is 2187 ($=3^7$) under all radix-3^2 basis configuration. In additional, by setting as all radix-2^3 basis configuration, the utilized system performs 2048 ($=2^{11}$) FFT points maximally. As for the other cases, we can deliver various FFT-point calculating by setting proper (a, b) combinations. Even we can set the bypassed modes on certain design brick stages for

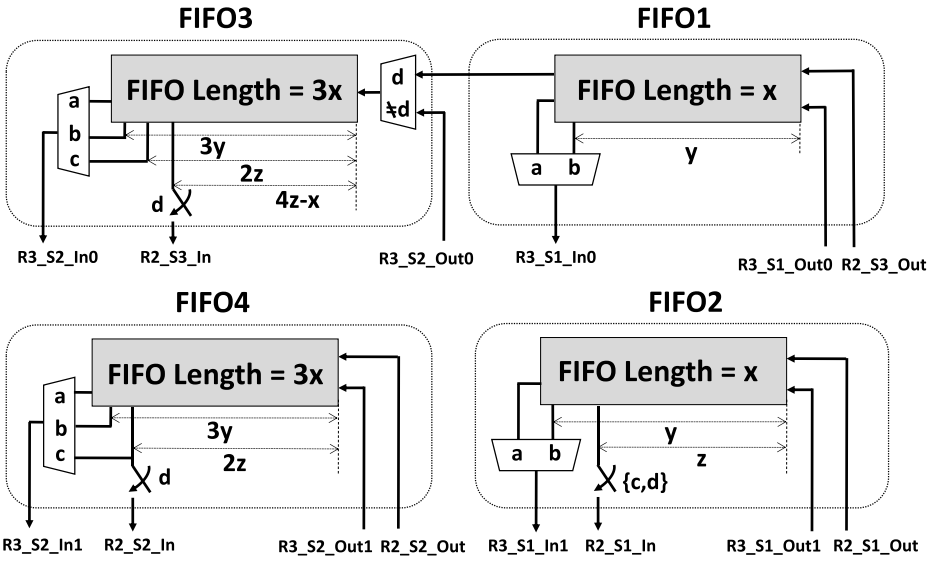

Fig. 5. Proposed hardware circuit of SFRA, FIFOs in the ith brick stage (FBS$-i$).

TABLE II
48-MODE CONFIGURATION SUPPORTING OF THE PROPOSED FFT HARDWARE ARCHITECTURE

Note : (a, b) represents $3^a * 2^b$- FFT point

Category	FFT Profile	Possible Combination Set Based on Radix-3^2 and Radix-2^3 Bases										# of FFT Size
1	(a, b)	(7,0)										1
	FFT Point	2187										
2	(a, b)	(6,1)	(6,0)									2
	FFT Point	1458	729									
3	(a, b)	(5,3)	(5,2)	(5,1)	(5,0)							4
	FFT Point	1944	972	486	243							
4	(a, b)	(4,4)	(4,3)	(4,2)	(4,1)	(4,0)						5
	FFT Point	1296	648	324	162	81						
5	(a, b)	(3,6)	(3,5)	(3,4)	(3,3)	(3,2)	(3,1)	(3,0)				7
	FFT Point	1728	864	432	216	108	54	27				
6	(a, b)	(2,7)	(2,6)	(2,5)	(2,4)	(2,3)	(2,2)	(2,1)	(2,0)			8
	FFT Point	1152	576	288	144	72	36	18	9			
7	(a, b)	(1,9)	(1,8)	(1,7)	(1,6)	(1,5)	(1,4)	(1,3)	(1,2)	(1,1)	(1,0)	10
	FFT Point	1536	768	384	192	96	48	24	12	6	3	
8	(a, b)	(0,11)	(0,10)	(0,9)	(0,8)	(0,7)	(0,6)	(0,5)	(0,4)	(0,3)	(0,2)	(0,1)	11
	FFT Point	2048	1024	512	256	128	64	32	16	8	4	2	
												Total	48

Note : (gray-colored area) indicates the FFT points defined in 3GPP-LTE standard (32 modes).

providing the smaller FFT size on demand, such as 12–192 points. In summary, the RC FFT hardware architecture can perform as many sizes as possible when M is increasing.

B. Six-Type Reconfigurable Processing Element

The dedicated PEs in the current literature can only support one radix manipulation. Instead, our proposed PE (6T-RC-PE) is RC to perform six different-radix configuration types. As demonstrated in Fig. 3, six configuration types are denoted as T_A, T_B, …, T_E, and T_F. Take T_A type (one-stage radix-3^2) for illustration. The multiplier computed with W_1 is nontrivial. But, the multiplier between two radix-3 substages is considered as the trivial multiplier with the TF, W_2. The possible values of W_2 are listed in Table III. Each substage of radix-3 operation is followed by a shifting process (shifted right by 2 b). It is not only to keep the same wordlength to connect two successive radix-3 substage operation circuits, but also to improve signal-to-quantization-noise effects due to fixed-point nature. Totally, operating at T_A type needs 12 complex adders and 4 complex multipliers, which achieve 100% of hardware reuse ratio.

In addition, as for T_B type, it is considered as one-stage radix-3 processing without any trivial multiplication. T_C type is served as one-stage radix-2^3 calculation with two trivial multipliers regarding W_2 and W_3. The shifting circuit is

TABLE III
EACH CONFIGURATION TYPE PROFILE OF 6T-RC-PE

Type Index	Circuit Configuration	Activated Adder		Activated Multiplier		# of TTFM Stage	Trivial Multiplier Value
		Number	Ratio	Number	Ratio		
T_A	Radix-3^2	12	100.0%	4	100.0%	1	$W\_2 = \{w_9^0, w_9^1, w_9^2, w_9^4\}$
T_B	Radix-3	6	50.0%	2	50.0%	0	None
T_C	Radix-2^3	6	50.0%	3	75.0%	2	$W\_2 = \{w_8^0, w_8^1, w_8^2, w_8^3\}$, $W\_3 = \{w_4^0, w_4^1\}$
T_D	Radix-2^2	4	33.3%	2	50.0%	1	$W\_2 = \{w_4^0, w_4^1\}$
T_E	Radix-2	2	16.7%	1	25.0%	0	None
T_F	Radix-3 + Radix-2	8	66.7%	3	75.0%	1	$W\_2 = \{w_6^0, w_6^1, w_6^2\}$

Note : TTFM = Trivial Twiddle Factor Multiplier

TABLE IV
DIFFERENT MODE SETTING OF 48-MODE DESIGN EXAMPLES SELECTED

Case Index	FFT Point	Configuration Type of 6T-RC-PE				{x, y, z} FIFO Parameter Setting															
						Stage 4				Stage 3				Stage 2				Stage 1			
		Stage 4	Stage 3	Stage 2	Stage 1	Cond.	x	y	z	Cond.	x	y	z	Cond.	x	y	z	Cond.	x	y	z
1	2187	T_B	T_A	T_A	T_A	a	729	-	-	a	81	-	-	a	9	-	-	a	1	-	-
2	2048	T_D	T_C	T_C	T_C	d	729	-	512	d	81	-	64	d	9	-	8	d	1	-	1
3	1024	T_E	T_C	T_C	T_C	d	729	-	512	d	81	-	64	d	9	-	8	d	1	-	1
4	729	(Bypass)	T_A	T_A	T_A	(Bypass)				a	81	-	-	a	9	-	-	a	1	-	-
5	648	(Bypass)	T_A	T_A	T_C	(Bypass)				b	81	72	-	b	9	8	-	d	1	-	1
6	576	(Bypass)	T_A	T_C	T_C	(Bypass)				b	81	64	-	d	9	-	8	d	1	-	1
7	512	(Bypass)	T_C	T_C	T_C	(Bypass)				d	81	-	64	d	9	-	8	d	1	-	1
8	486	(Bypass)	T_A	T_A	T_F	(Bypass)				b	81	54	-	b	9	6	-	c	1	-	1
9	432	(Bypass)	T_A	T_F	T_C	(Bypass)				b	81	48	-	c	9	-	8	d	1	-	1
10	384	(Bypass)	T_F	T_C	T_C	(Bypass)				c	81	-	64	d	9	-	8	d	1	-	1
11	12	(Bypass)	(Bypass)	T_B	T_D	(Bypass)				(Bypass)				b	9	4	-	d	1	-	1
12	2	(Bypass)	(Bypass)	(Bypass)	T_E	(Bypass)				(Bypass)				(Bypass)				d	1	-	1

Note : "Cond." represents FIFO setting condition selected from {a, b, c, d}.

used to shift right by only 1 b instead. In the similar manner, T_D and T_E types are supported for one-stage radix-2^2 and one-stage radix-2, respectively. The sixth type (T_F type) is a mixed-radix computation, concatenating one-stage radix-3 and one-stage radix-2 circuits. Two shifting circuits are shifted by different amounts. One is for shifting right by 2 b and the other is for shifting right by 1 b. The number of activated complex adder and multiplier for all six RC types is summarized in Table III. Also, we can compare the number of trivial TF multiplier stage and corresponding trivial multiplier values for {T_A–T_F} types.

Fig. 4 shows the proposed 6T-RC-PE circuit in more detail. It is a five-stage pipelining hardware design for all of six types. The input and output of 6T-RC-PE are denoted as PE_In and PE_Out, respectively. R2_S3_In stands for the signal from the FIFO output at substage 3 of T_C type. The signal fed into the FIFO at substage 3 of T_C type is denoted as R2_S3_Out. Besides, R3_S2_In0 and R3_S2_In1 indicate the signals from two FIFO outputs at substage 2 of T_A type (or T_F type). The signals fed into two FIFOs at substage 2 of T_A type (or T_F type) are represented as R3_S2_Out0 and R3_S2_Out1. Other signals are shown in the similar manner. As for the detailed design approach regarding systematic FIFO arrangement, we will introduce in Section III-C.

In the hardware resource point-of-view, there entirely are 4 complex multipliers, 12 complex adders, and several controlled MUXes for supporting all of 6 configuration types. In the timing analysis aspect, the critical path is mainly dominated by tracing one multiplier, two adders, and nine controlled MUXes in series.

In order to illustrate the related type setting of 6T-RC-PE, Table IV shows 12 representative design examples, selected out of 48 modes. As for case index of 1, the FFT system is of 2187 ($=3^7$) points. Brick stages 1–4 are asserted as T_A, T_A, T_A, and T_B types, respectively. When dealing with case index of 5, the system performs 648 ($=3^4 \times 2^3$) FFT points and needs 3 worked brick stages in practice. Brick stage 4 is controlled at bypassed mode whereas brick stages 1–3 are chosen as T_C, T_A, and T_A types, respectively. Moreover, the number of utilized brick stage is less while manipulating

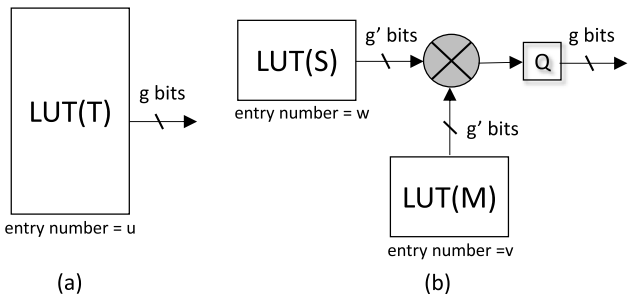

Fig. 6. TF generator. (a) Traditional direct LUT approach. (b) Proposed STFG technique.

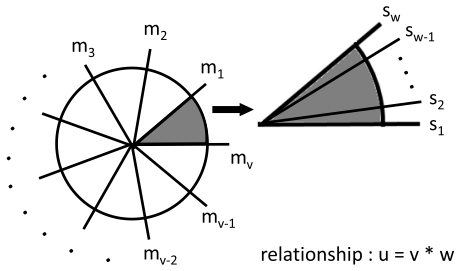

Fig. 7. Design concept of the proposed STFG technique.

smaller FFT size. We can set 12 FFT points as shown in case index of 11. It totally requires only two brick stages, which brick stages 1 and 2 are of T_D and T_B types, respectively. In other words, appropriately performing valid selected types at each brick stage can lead to the possible FFT-point combination. Also, if we want to realize larger FFT size, the system requires more brick stages (change the corresponding value of M).

C. Systematic FIFO Reuse Arrangement

Each brick stage in Fig. 2 has not only PE, but also FIFO, which is used to store the temporal data. In order to overcome complex reading/writing access for so many mode configuration, SFRA is developed to provide an easy and systematic scheme for each brick stage. Fig. 5 shows FIFOs in the ith brick stage, denoted as FBS$-i$, where i = 1–4 for $M = 4$ case. In the hardware resource aspect, FBS$-i$ has a total FIFO length of $8\times$, where $x = 3^{2(i-1)}$. Overall FIFO is separated into four portions, such as FIFO1–FIFO4. FIFO1 and FIFO2 lengths are of x whereas FIFO3 and FIFO4 lengths are of $3\times$. For clear demonstrating, the detailed hardware circuit operations (FBS$-i$) are divided into four different conditions as follows.

1) Condition a: FBS$-i$ is used to execute one of {T_A, T_B} types. FBS$-(i-1)-$FBS-1 perform x FFT points entirely, served as all radix-3^2 types.
2) Condition b: FBS$-i$ is used to execute one of {T_A, T_B} types. FBS$-(i-1)-$FBS-1 perform y FFT points entirely, served as beyond all radix-3^2 types.
3) Condition c: FBS$-i$ is used to execute T_F type. FBS$-(i-1)-$FBS-1 perform z FFT points entirely, served as all radix-2^3 types.
4) Condition d: FBS$-i$ is used to execute one of {T_C, T_D, T_E} types. FBS$-(i-1)-$FBS-1 perform z FFT points entirely, served as all radix-2^3 types.

As mentioned earlier, FBS$-i$ can systematically complete FIFO reading/writing access while the related 6T-RC-PE is operated at one of six configuration types. Furthermore, we can take a look at Table IV again for more design example discussion. As for case index of 1, all FIFOs in the first to the fourth brick stages belong to "condition a" simultaneously. x values of the first to the fourth brick stages are 1, 9, 81, and 729, respectively. While setting 432 ($=3^3 \times 2^4$) FFT points as case index of 9, FIFOs in the first to the third brick stages are mapped to "condition d," "condition c," and "condition b," respectively. The corresponding parameters are $\{z = 1, z = 8, y = 48\}$ at the first to the third brick stages, respectively. But, FIFOs in the fourth brick stage are unused. Also, the other representative design examples are derived in a similar manner. In summary, by utilizing the proposed SFRA design technique, FIFOs in each brick stage are designed in a systematic way even for extending to a larger FFT size in the future.

D. Section-Based Twiddle Factor Generator

Fig. 6(a) shows the traditional direct LUT approach to generate TFs for each nontrivial multiplier accordingly. It is commonly adopted in the most of current literature. Also, it is a straightforward scheme to store u different values in u independent table entries, of which each value has g bits. But, the circuit area increases tremendously as the table entry size increases. The table size is very large when the target FFT system behaves thousands of points, such as 2187, 2048, 1944, …, and so on. Therefore, we propose a design technique of STFG, as shown in Fig. 6(b). Instead of utilizing only one large LUT, STFG is used to realize a divide-and-conquer approach to reduce the total hardware circuit area. STFG design procedure is entirely separated into three steps as follows.

1) Divide one large LUT into two smaller LUTs, of which each LUT produces g'-bit value ($g' \leq g$).
2) Perform multiplication with two g'-bit values.
3) Quantize the output signal, returning to g bits.

The first step of STFG is the crucial design foundation. Assume that the original TF has a general form as (13)

$$W_u^k, \quad \text{where } k = \{0, 1, \ldots, u-1\} \tag{13}$$

which indicates to divide a unit circle into u equal segments. Instead of directly dividing once, we can use two-level rotating approaches to complete the corresponding action, because we know the rotation relationship as (14). Two subterms in the right-hand side of (14) are two rotation operations

$$W_u^k = W_v^{k1} \times W_{v*w}^{k2}, \quad \text{where } k = \{0, 1, \ldots, u-1\}$$
$$k1 = \{0, 1, \ldots, v-1\}, \quad k2 = \{0, 1, \ldots, w-1\}. \tag{14}$$

In other words, we first divide a unit circle into v equal segments, as v mandatory sections demonstrated in Fig. 7. It is called a mandatory rotation and all the values of $\{m_1, m_2, \ldots, m_{v-1}, m_v\}$ are stored in the LUT(M) of Fig. 6(b). In the following, we can divide each piece of mandatory

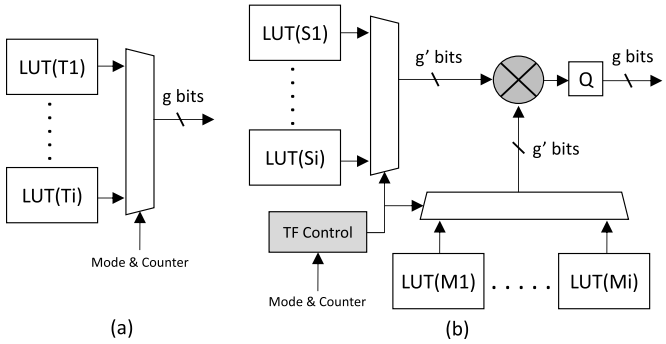

Fig. 8. Hardware circuit. (a) Traditional direct LUT. (b) STFG technique.

section into w equal segments. It is called a secondary rotation and all the values of $\{s_1, s_2, \ldots, s_{w-1}, s_w\}$ are stored in the LUT(S) of Fig. 6(b). In this way, it needs to store only $(v+w)$ table entries in total as compared with the traditional direct LUT approach (storing u table entries). Take something real for examples. If u equals to 2048, then we choose $\{v, w\} = \{32, 64\}$. STFG only requires 96 (=32 + 64) table entries whereas the traditional direct LUT approach needs 2048 table entries. Therefore, the fact of less table entries leads to less hardware complexity and smaller circuit area (see more details in Section IV-A).

As for the RC FFT system, each brick stage is responsible for supporting several modes. Every mode has its individual TF values and regularity. So each brick stage needs several larger tables by using traditional direct LUT approach, as shown in LUT($T1$)−LUT(Ti) of Fig. 8(a). The final g-bit value result is selected from i LUTs, controlled by "Mode" and related "Counter." "Mode" and "Counter" signals have more loading influence while the brick stage needs more LUTs. It results in a larger circuit area totally. Instead, Fig. 8(b) shows the hardware circuit implemented with STFG. Each LUT(Ti) is separated into two subtables, LUT(Mi) and LUT(Si). Two g'-bit outputs are combined with the multiplier. Besides, it requires one extra hardware circuit, TF Control, which is used to perform corresponding TF controlling. It is different from traditional direct LUT approach, because it needs one extra step to pick up two according subtable values to combine and produce the final target g-bit result. The detailed area analysis is given in Section IV-A.

As for $M = 4$ case, there is no need to apply STFG technique on brick stages 1 and 2, because the original LUT area size is pretty small. Thus, we just utilize traditional direct LUT on brick stages 1 and 2. For brick stage 3, the analysis of applied STFG technique is shown in Table V. In brick stage 3, it entirely requires to provide ten different kinds of table modes, which afford to support all of the reconfiguration. The corresponding table entry size varies from 256 to 729. For the table index of 4, the table entry size reduces from 512 to 48 (=16 + 32), owning reduced ratio of 90.6%. Applying for all ten table modes can extremely reduce table entry size and the reduced ratio ranges from 87.5% to 92.6%. On the other hand, the analysis of applied STFG technique for brick stage 4 is shown in Table VI. All of 11 table modes have the reduced ratio from 92.7% to 95.4%. Total entry size is decreased from 15953 to 860, having an average reduced ratio of 94.6%. In other words, table entry saving ratio increases as the original table size increases. STFG applied for brick stage x is more circuit-efficient than STFG applied for brick stage y ($x > y$) due to more table-mode supporting. In addition to entry saving, Table VII analyzes the according relationship between (g, g') choices on STFG and induced error effects. The errors are as compared with traditional direct LUT approach. If setting

TABLE V
STFG TECHNIQUE APPLIED FOR BRICK STAGE 3

Table Index	Table Entry Size				Entry Saving	
	Direct LUT	Proposed STFG			Value (Ti - Pi)	Reduced Ratio
	LUT (Ti)	LUT (Mi)	LUT (Si)	Pi=Mi+Si		
1	729	27	27	54	675	92.6%
2	648	24	27	51	597	92.1%
3	576	24	24	48	528	91.7%
4	512	16	32	48	464	90.6%
5	486	18	27	45	441	90.7%
6	432	18	24	42	390	90.3%
7	384	16	24	40	344	89.6%
8	324	18	18	36	288	88.9%
9	288	16	18	34	254	88.2%
10	256	16	16	32	224	87.5%
Total	4635	193	237	430	4205	90.7%

Note : Mi = Mandatory LUT, Si = Secondary LUT

TABLE VI
STFG TECHNIQUE APPLIED FOR BRICK STAGE 4

Table Index	Table Entry Size				Entry Saving	
	Direct LUT	Proposed STFG			Value (Ti - Pi)	Reduced Ratio
	LUT (Ti)	LUT (Mi)	LUT (Si)	Pi=Mi+Si		
1	2187	27	81	108	2079	95.1%
2	2048	32	64	96	1952	95.3%
3	1944	36	54	90	1854	95.4%
4	1728	36	48	84	1644	95.1%
5	1536	32	48	80	1456	94.8%
6	1458	27	54	81	1377	94.4%
7	1296	27	48	75	1221	94.2%
8	1152	32	36	68	1084	94.1%
9	972	27	36	63	909	93.5%
10	864	27	32	59	805	93.2%
11	768	24	32	56	712	92.7%
Total	15953	327	533	860	15093	94.6%

Note : Mi = Mandatory LUT, Si = Secondary LUT

TABLE VII
ERROR ANALYSIS FOR STFG

g (bit)	g' (bit)	Error due to STFG	
		Brick Stage 4	Brick Stage 3
16	16	1.83*10^-5	1.85*10^-5
	15	4.33*10^-5	4.35*10^-5
	14	9.35*10^-5	8.71*10^-5
	13	1.94*10^-4	1.82*10^-4
	12	3.97*10^-4	3.65*10^-4
	11	7.81*10^-4	7.31*10^-4
	10	1.55*10^-3	1.42*10^-3
	9	3.10*10^-3	2.91*10^-3
	8	6.19*10^-3	5.77*10^-3

g (bit)	g' (bit)	Error due to STFG	
		Brick Stage 4	Brick Stage 3
14	14	7.33*10^-5	7.17*10^-5
	13	1.76*10^-4	1.68*10^-4
	12	3.78*10^-4	3.57*10^-4
	11	7.52*10^-4	7.16*10^-4
	10	1.55*10^-3	1.42*10^-3
	9	3.08*10^-3	2.87*10^-3
	8	6.17*10^-3	5.74*10^-3
	7	1.21*10^-2	1.14*10^-2

TABLE VIII
AREA ANALYSIS OF STFG DESIGN TECHNIQUE WITH DIFFERENT WORDLENGTH SELECTIONS

Circuit	g (bit)	g' (bit)	Traditional LUT Area (um^2) (T)	Section-Based Twiddle Factor Generator Area (um^2)					Area Saving	
				LUT (M) (R1)	LUT (S) (R2)	Multiplier (R3)	TF Control (R4)	Total (R) (sum (Ri))	Value (um^2) (T-R)	Reduced Ratio
Brick Stage 3	16	8	41479.4	1675.7 (10.5%)	2057.7 (12.9%)	4606.2 (29.0%)	7564.6 (47.6%)	15904.2	25575.2	61.7%
		10		2125.8 (10.5%)	2610.4 (12.8%)	7642.3 (37.6%)	7948.6 (39.1%)	20327.1	21152.3	51.0%
		12		2779.8 (10.6%)	3413.6 (13.0%)	11915.8 (45.3%)	8190.6 (31.1%)	26299.8	15179.6	36.6%
		14		3530.2 (11.5%)	4335.0 (14.1%)	14697.7 (47.9%)	8136.3 (26.5%)	30699.2	10780.2	26.0%
		16		4046.6 (11.1%)	4969.2 (13.7%)	19172.3 (52.8%)	8145.4 (22.4%)	36333.5	5145.9	12.4%
Brick Stage 4	16	8	107563.7	1620.5 (13.3%)	2641.3 (21.7%)	4611.1 (37.9%)	3298.7 (27.1%)	12171.6	95392.1	88.7%
		10		2309.2 (13.8%)	3763.9 (22.6%)	7307.8 (43.8%)	3307.2 (19.8%)	16688.1	90875.6	84.5%
		12		3072.7 (13.4%)	5008.3 (21.9%)	11292.7 (49.3%)	3517.4 (15.4%)	22891.1	84672.6	78.7%
		14		3660.1 (13.1%)	5965.8 (21.4%)	14792.1 (52.9%)	3505.4 (12.6%)	27923.4	79640.3	74.0%
		16		4175.7 (12.4%)	6806.3 (20.1%)	19282.6 (57.1%)	3517.4 (10.4%)	33782.0	73781.7	68.6%

the error upper bound as about 10^−3, g' has the possible values as the gray-colored zone shown in Table VII. Once we are not satisfied the error extent as our expected, we can make a new choice from the analysis. Selecting (g, g') combination at every brick stage appropriately is to offer a design tradeoff between FFT performance requirement and circuit implementation area.

IV. ASIC CHIP IMPLEMENTATION

In this section, we first do further circuit area analysis of STFG design technique, applied for brick stages 3 and 4. Then, we will demonstrate the synthesis results of the proposed FFT system ($M = 4$) via TSMC 90-nm CMOS technology. Finally, we will show the ASIC chip implementation and post-APR performance. All the proposed circuit designs are synthesized with Design Compiler and implemented with IC Compiler.

A. Area Analysis of STFG Design Technique

Table VIII shows the synthesis area results of STFG design technique. As for brick stage 3, we take (g, g') = (16, 10) combination for illustrated examples. As compared with traditional direct LUT approach (41479.4 μm^2), our proposed circuit area is only 20327.1 μm^2, which has an area saving ratio of 51%. The subcircuits of LUT(M), LUT(S), Multiplier, and TF Control occupy 10.5%, 12.8%, 37.6%, and 39.1% of total area, respectively. Furthermore, g' bit ranges from 16 to 8 when g bit is set as 16. The corresponding area saving ratio varies from 12.4% to 61.7%. On the other hand, brick stage 4 also has circuit area reduction in the analogous trends. While g' bit varies from 16 to 8, the area saving ratio is from 68.6% to 88.7%. We observed that the fact of area-reducing performance is proportional to the entry-saving effects (also see Section III-D). At the same time, we can make a suitable choice with a design tradeoff of error/precision and circuit area according to the analysis in Tables VII and VIII.

B. Synthesis Results of the Proposed System Design

Our proposed RC FFT system ($M = 4$) provides up to 2187 points and behaves 48 different FFT sizes. Also, it supports 32 operating modes defined in 3GPP-LTE SPEC. The synthesis results of all subcircuits are listed in Table IX. The overall synthesized circuit area is 1.116 mm^2 while each

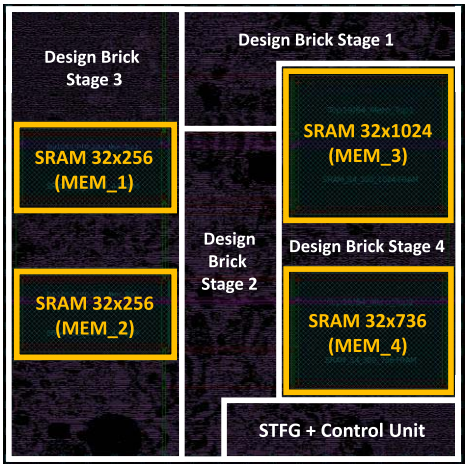

Fig. 9. Chip layout of the proposed RC FFT system.

TABLE IX
SYNTHESIS RESULTS

Sub-circuits		Synthesis Area (mm^2)		
		Value	Ratio	
Design Brick Stage 1	RC-FIFO	0.007	0.6%	8.9%
	6T-RC-PE	0.093	8.3%	
Design Brick Stage 2	RC-FIFO	0.052	4.7%	14.3%
	6T-RC-PE	0.107	9.6%	
Design Brick Stage 3	RC-FIFO	0.302	27.1%	36.8%
	6T-RC-PE	0.108	9.7%	
Design Brick Stage 4	RC-FIFO	0.330	29.5%	34.8%
	6T-RC-PE	0.059	5.3%	
STFG		0.048	4.3%	5.2%
Control Unit (CU)		0.010	0.9%	
Total		1.116	100.0%	100.0%

TABLE X
DESIGN COMPARISON WITH OTHER STATE-OF-THE-ART WORKS

	Our Work	[36]	[31]	[30]	[32]	[9]
FFT-Point Supporting	2 ~ 2187	12 ~ 1296	128/256/512/ 1024/1536/2048	128/256/512/ 1024/1536/2048	128/256/512/ 1024/1536/2048	1024
Max. FFT Size	2187	1296	2048	2048	2048	1024
Number of Mode	48 *	34	6	6	6	1
Process	90 nm	0.18 um	0.18 um	90 nm	65 nm	65 nm
Frequency (MHz)	188.67	122.88	35.00	40.00	20.00	30.00
Wordlength (Bit)	16	16	16	12	12	16
Core Area (mm^2)	1.664	25.000	1.932	0.783	1.375	8.299
Gate Count	396K	482K	NA	205K	1,100K	NA
Throughput (M Symbols/s)	188.67	122.88	35.00	40.00	20.00	240.00
Power (mW)	35.20	320.00	11.29	7.20	8.55	4.14
Normalized Area (mm^2) [32]	0.87	3.26	0.25	0.41	1.38	8.30
Normalized Energy/FFT (nJ) [32]	294.69	1218.75	238.56	266.24	875.52	141.31
Speed-Area Ratio (SAR) (10^{-3})	0.48	0.25	NA	0.20	0.02	NA
Power-Freq Ratio (PFR) (10^{-6})	85.31	2009.39	157.51	87.89	208.74	134.77
Implementation Results	Post-layout sim.	Measurement	Measurement	Synthesis Results	Measurement	Measurement

Note : * denotes that it is also applied to support 32 modes defined in 3GPP-LTE standard.

brick stage has the same input/output fixed-point wordlength of 16 b. The maximum clock frequency operates at 200 MHz, delivering a system throughput of 200 MSymbols/s. Brick stages 1–4 have the area ratio of 8.9%, 14.3%, 36.8%, and 34.8%, respectively. In the general 6T-RC-PE aspects (such as RC-PE of brick stages 2 and 3), the synthesis area is around 0.108 μm^2, occupying 9.7% of total circuits. As for RC-PE of brick stage 1, the unnecessary circuit is removed due to the unused last multiplier in Fig. 4. Under the condition of total modes in Table II, the unnecessary hardware of RC-PE in brick stage 4 is also eliminated, because we only provide T_B, T_D, and T_E types in this stage. In the FIFO aspects, each brick stage has 0.6%, 4.7%, 27.1%, and 29.5% of total area individually. The FIFO of brick stage 4 mostly dominates the hardware area cost. As for STFG design technique, $g = 16$ b and $g' = 10$ b are applied for brick stages 3 and 4, which give a design balance/tradeoff between fixed-point calculating precision and hardware area cost. Eventually, the circuits of STFG and CU only occupy 5.2% of total design area.

C. Post-APR Performance of the Proposed System Design

Fig. 9 shows the ASIC implementation with APR results, only occupying a core area of 1.29 mm $\times$ 1.29 mm (=1.664 mm^2). As for 2187 FFT-point mode (maximally utilize FIFO banks and PEs), it can maximally operate at 188.67 MHz and dissipate dynamic power of 35.2 mW. In the chip layout, there are totally four SRAMs for data storage. Two SRAM 32×256 are responsible for FIFO implementation at brick stage 3, denoted as MEM_1 and MEM_2. On the other hand, MEM_3 (SRAM 32×1024) and MEM_4 (SRAM 32×736) are used for FIFO realization at brick stage 4. MEM_1–MEM_4 occupy 5.6%, 5.6%, 10.6%, and 8.8% of total core area, respectively.

V. DESIGN COMPARISON

This paper can compare with current state-of-the-art works. They are very different with each other, such as various max FFT-point supporting, number of operating modes, operating frequency, and utilized CMOS technology. In addition to employing the comparison indexes in [32], we also create two performance indexes in a different inspection point of view, including speed-area ratio (SAR) and power-frequency ratio (PFR) defined as (15) and (16), respectively

$$\text{Speed-Area Ratio(SAR)} \triangleq \frac{\text{Throughput}}{\text{Gate\_Count}} \tag{15}$$

$$\text{Power-Freq Ratio(PFR)} \triangleq \frac{\text{Power}}{\text{Frequency}^* \text{ Max\_FFT\_Size}}. \tag{16}$$

SAR indicates that how many bits each gate delivers in every time unit. The target work owns better design performance in terms of "*gate efficiency*" while SAR increases. On the other hand, PFR represents that how much power one FFT point dissipates under the same operating frequency condition. The target work obtains design better performance in terms of "*green energy*" while PFR decreases. As listed in Table X, this paper has higher working frequency, more FFT size supporting, and larger FFT-point providing. In addition, this paper also has better design performance in terms

of SAR and PFR. Although SAR and PFR are not verified to absolutely justify which design is the best among all of various design works, they are considered as a simple way for comparison reference.

VI. Conclusion

We propose an RC FFT design in a systematic design way. The important RC design bricks are proposed to easily concatenate to meet target FFT-point on demand. In a design/implementation example, the RC FFT system supports up to 2187 FFT points and provides 48 RC modes. It also supports 32 operating modes defined in 3GPP-LTE standard. After post-APR results with TSMC 90-nm CMOS technology, our design work occupies a core area of 1.664 mm^2 and consumes 35.2 mW under maximal operating frequency of 188.67 MHz. It maximally delivers total system throughput of 188.67 MSymbols/s. Besides, our design work also has outstanding design performance in terms of SAR and PFR for comparison reference. In conclusion, it features not only RC characteristics by easily constructing RC design bricks, but also higher speed operation, higher gate efficiency, and better green energy for per FFT-point computing.

Acknowledgment

The authors would like to thank the National Chip Implementation Center for all chip technical supports.

References

[1] S. He and M. Torkelson, "A new approach to pipeline FFT processor," in *Proc. 10th Int. Parallel Process. Symp. (IPPS)*, Apr. 1996, pp. 766–770.

[2] A. G. da Luz, E. A. da Costa, and S. Ghissoni, "Reducing power consumption in FFT architectures by using heuristic-based algorithms for the ordering of the twiddle factors," in *Proc. IEEE 3rd Latin Amer. Symp. Circuits Syst. (LASCAS)*, Feb./Mar. 2012, pp. 1–4.

[3] K. George and C.-I. H. Chen, "Configurable and expandable FFT processor for wideband communication," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2007, pp. 1–5.

[4] Z. Qian and M. Margala, "Low-power split-radix FFT processors using radix-2 butterfly units," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 9, pp. 3008–3012, Sep. 2016.

[5] D. Bhagat and H. W. Mergler, "A high-performance microprocessor-based FFT processor," *IEEE Trans. Ind. Electron. Control Instrum.*, vol. 25, no. 2, pp. 102–107, May 1978.

[6] R. Neuenfeld, M. Fonseca, and E. Costa, "Design of optimized radix-2 and radix-4 butterflies from FFT with decimation in time," in *Proc. IEEE 7th Latin Amer. Symp. Circuits Syst. (LASCAS)*, Feb./Mar. 2016, pp. 171–174.

[7] E. E. Swartzlander, W. K. W. Young, and S. J. Joseph, "A radix 4 delay commutator for fast Fourier transform processor implementation," *IEEE J. Solid-State Circuits*, vol. 19, no. 5, pp. 702–709, Oct. 1984.

[8] M. Hasan, T. Arslan, and J. S. Thompson, "A novel coefficient ordering based low power pipelined radix-4 FFT processor for wireless LAN applications," *IEEE Trans. Consum. Electron.*, vol. 49, no. 1, pp. 128–134, Feb. 2003.

[9] M. Seok, D. Jeon, C. Chakrabarti, D. Blaauw, and D. Sylvester, "A 0.27 V 30 MHz 17.7 nJ/transform 1024-pt complex FFT core with super-pipelining," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2011, pp. 342–344.

[10] Y.-W. Lin, H.-Y. Liu, and C.-Y. Lee, "A dynamic scaling FFT processor for DVB-T applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 11, pp. 2005–2013, Nov. 2004.

[11] I.-C. Park, W. Son, and J.-H. Kim, "Twiddle factor transformation for pipelined FFT processing," in *Proc. 25th Int. Conf. Comput. Design (ICCD)*, Oct. 2007, pp. 1–6.

[12] G. Bi and G. Li, "Pipelined structure based on radix-2^2 FFT algorithm," in *Proc. 6th IEEE Conf. Ind. Electron. Appl.*, Jun. 2011, pp. 2530–2533.

[13] N. H. Cuong, N. T. Lam, and N. D. Minh, "Multiplier-less based architecture for variable-length FFT hardware implementation," in *Proc. 4th Int. Conf. Commun. Electron. (ICCE)*, Aug. 2012, pp. 489–494.

[14] S. He and M. Torkelson, "Designing pipeline FFT processor for OFDM (de)modulation," in *Proc. URSI Int. Symp. Signals, Syst., Electron. (ISSSE)*, Oct. 1998, pp. 257–262.

[15] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "New radix-(2×2×2)/(4×4×4) and radix-(2×2×2)/(8×8×8)DIF FFT algorithms for 3-D DFT," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 2, pp. 306–315, Feb. 2006.

[16] Y. Chen, Y.-C. Tsao, Y.-W. Lin, C.-H. Lin, and C.-Y. Lee, "An indexed-scaling pipelined FFT processor for OFDM-based WPAN applications," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 55, no. 2, pp. 146–150, Feb. 2008.

[17] T. Adiono, M. S. Irsyadi, Y. S. Hidayat, and A. Irawan, "64-point fast efficient FFT architecture using radix-23 single path delay feedback," in *Proc. Int. Conf. Elect. Eng. Inform.*, Aug. 2009, pp. 654–658.

[18] D.-S. Kim and S.-Y. Lee, "Dual input radix 2^3 SDF IFFT/FFT processor for wireless multi-channel real sound speakers using time division duplex scheme," *IEEE Trans. Consum. Electron.*, vol. 55, no. 4, pp. 2323–2328, Nov. 2009.

[19] J. He, L. Ma, and X. Xu, "A configurable FFT processor," in *Proc. IET 3rd Int. Conf. Wireless, Mobile Multimedia Netw. (ICWMNN)*, Sep. 2010, pp. 246–249.

[20] Y.-W. Lin, H.-Y. Liu, and C.-Y. Lee, "A 1-GS/s FFT/IFFT processor for UWB applications," *IEEE J. Solid-State Circuits*, vol. 40, no. 8, pp. 1726–1735, Aug. 2005.

[21] M. Shin and H. Lee, "A high-speed four-parallel radix-2^4 FFT/IFFT processor for UWB applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 960–963.

[22] L. Jeesung and L. Hanho, "A high-speed two-parallel radix-2^4 FFT/IFFT processor for MB-OFDM UWB systems," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 91, no. 4, pp. 1206–1211, Apr. 2008.

[23] F. Qureshi, S. A. Alam, and O. Gustafsson, "4K-point FFT algorithms based on optimized twiddle factor multiplication for FPGAs," in *Proc. Asia–Pacific Conf. Postgraduate Res. Microelectron. Electron. (PrimeAsia)*, Sep. 2010, pp. 225–228.

[24] M. Garrido, J. Grajal, M. Sánchez, and O. Gustafsson, "Pipelined radix-2^k feedforward FFT architectures," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 21, no. 1, pp. 23–32, Jan. 2013.

[25] E. Dubois and A. Venetsanopoulos, "A new algorithm for the radix-3 FFT," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 3, pp. 222–225, Jun. 1978.

[26] S. K. Bashar, "An efficient approach to the computation of fast Fourier transform(FFT) by radix-3 algorithm," in *Proc. Int. Conf. Inform., Electron. Vis. (ICIEV)*, May 2013, pp. 1–5.

[27] A. Karlsson, J. Sohl, and D. Liu, "Cost-efficient mapping of 3- and 5-point DFTs to general baseband processors," in *Proc. IEEE Int. Conf. Digit. Signal Process. (DSP)*, Jul. 2015, pp. 780–784.

[28] S. Prakash and V. Rao, "A new radix-6 FFT algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 4, pp. 939–941, Aug. 1981.

[29] Y. Suzuki, T. Sone, and K. Kido, "A new FFT algorithm of radix 3, 6, and 12," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 2, pp. 380–383, Apr. 1986.

[30] C. Yu and M.-H. Yen, "Area-efficient 128-to 2048/1536-point pipeline FFT processor for LTE and mobile WiMAX systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 9, pp. 1793–1800, Sep. 2015.

[31] 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. Int. Conf. Green Circuits Syst. (ICGCS)*, Jun. 2010, pp. 14–17.

[32] 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.

[33] W. Zheng and K. Li, "Split radix algorithm for 6^m length DFT," *IEEE Signal Process. Lett.*, vol. 20, no. 7, pp. 713–716, Jul. 2013.

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

[35] T. Ayhan, W. Dehaene, and M. Verhelst, "A 128∼2048/1536 point FFT hardware implementation with output pruning," in *Proc. 22nd Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2014, pp. 266–270.

[36] J. Chen, J. Hu, S. Lee, and G. E. Sobelman, "Hardware efficient mixed radix-25/16/9 FFT for LTE systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 2, pp. 221–229, Feb. 2015.

Xin-Yu Shih was born in ChangHua, Taiwan. He received the B.S. degree in electrical engineering from National Tsing Hua University, HsinChu, Taiwan, in 2004, and the M.S. and Ph.D. degrees in electronic engineering from National Taiwan University, Taipei, Taiwan, in 2006 and 2010, respectively.

In 2010, he joined MediaTek, Hsinchu, where he was a Senior Engineer for over four years. He joined the Faculty with the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, in 2015, where he is currently an Assistant Professor. His current research interests include the very large-scale integration design architectures and hardware circuits for digital signal processing, communication systems, and channel coding of next generation applications.

Yue-Qu Liu was born in Taipei, Taiwan. He received the B.S. degree from the National Chin-Yi University of Technology, Taichung, Taiwan, in 2015. He is currently pursuing the M.S. degree with National Sun Yat-sen University, Kaohsiung, Taiwan.

He is a Graduate Student with the Smart and Novel System Design IC Laboratory, led by Prof. X.-Y. Shih. His current research interests include very large scale integration hardware design and communication ICs.

Hong-Ru Chou was born in Taipei, Taiwan. He received the B.S. degree from Tamkang University, New Taipei City, Taiwan, in 2015. He is currently pursuing the M.S. degree with National Sun Yat-sen University, Kaohsiung, Taiwan.

He is a Graduate Student with the Smart and Novel System Design IC Laboratory, led by Prof. X.-Y. Shih. His current research interests include very large-scale integration hardware design and communication ICs.