VLSI Design and Implementation of Reconfigurable 46-Mode Combined-Radix-Based FFT Hardware Architecture for 3GPP-LTE Applications

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

***Abstract*— This paper presents a reconfigurable fast Fourier transform (FFT) hardware architecture, supporting 46 different FFT sizes defined in 3GPP-LTE applications. Our proposed design concept is mainly based on combined radix-5, radix-3^2, and radix2^4 single-path delay feedback FFT design approaches. In addition, in order to elaborate our hardware design, we also develop three design techniques, such as reconfigurable processing kernel with seven types (RPK-ST), efficient FIFO management scheme, and single-table approximation method. In an ASIC implementation with TSMC 40-nm CMOS technology, our 46-mode reconfigurable FFT chip only occupies a core area of 0.36 mm^2, dissipates 48.46 mW, and operates up to clock frequency of 500 MHz. As compared with the other state-of-the-art works, our work delivers high-quality design results in the aspects of area- and energy-related performance indexes, providing a constructive FFT design prototyping for 3GPP-LTE systems.**

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

I. Introduction

FAST Fourier Transform (FFT) is an important design scheme, transferring concerned signals from time to frequency domains. It is also a common design skill applied for many communication systems. In the advanced communication applications (such as 3GPP-LTE standard), it defines many FFT sizes for accommodating different outside environments. Among various hardware-implemented design approaches, single-path delay feedback (SDF) FFT [1] is very famous for a design trade-off between circuit area and calculating latency. Initially, the SDF FFT design is according to a radix-2 basis [2]–[7]. Afterwards it has a moving trend to other single radix, such as radix-3 [8], [9], radix-4 [10]–[14], radix-5 [15]–[18], radix-6 [19], [20], and radix-8 [21]. In order to reduce circuit complexity of processing kernels and twiddle-factors multiplication, there exist a series of developed design structures, including radix-2^2[22]–[24], radix-2^3[25]–[27], and radix-2^4[28]–[30] bases. The similar FFT design approach even more extends to a general radix-2^k [31], [32] basis. In addition to employing only one specified single-radix on a whole FFT system, there are different FFT configurations [33]–[36]. [33] shows that the computed signal path consists of concatenated radix-2 and radix-3 FFT computing from input to output ends. The computing datapath is composed of concatenated radix-2, radix-3, and radix-5 FFT-calculating [34], [35]. Furthermore, [36] employs two similar separate processing paths. One path is used for radix-2 calculating only whereas the other path is utilized for purely concatenating different radixes.

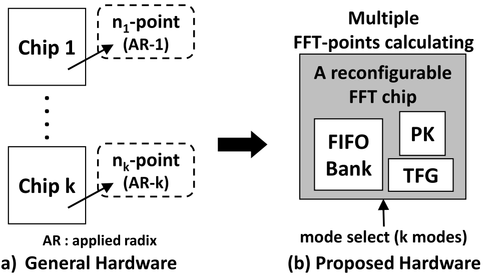

Fig. 1. Design concepts of general and proposed hardware designs.

There are two motivations for us to develop a newly reconfigurable FFT hardware architecture. (1) As for current FFT designs in [38]–[40], they only support 4 - 6 basic cases of FFT sizes defined in 3GPP-LTE standard. Even, [41] can provide 34 FFT sizes, including more extended cases of FFT sizes, but it is still limited. (2) In the existing FFT designs, the large hardware overhead is unavoidable because they do not fully reuse the circuit components. Thus, we want to develop a smart/novel reconfigurable FFT hardware architecture, which mainly utilizes combined-radix FFT-processing to achieve as many FFT-mode supporting as possible, such as 46 modes. As depicted in Fig. 1, we first analyze the constitutive units in FFT system for different FFT-size requirements. And then we can suitably utilize the hardware reuse principle on processing kernels (PK), FIFO banks, and twiddle factor generators (TFG).

The rest of this paper is organized as follows. Section II shows our proposed reconfigurable FFT hardware architecture. We also develop three design techniques with specified purposes as follows. (1) Reconfigurable processing kernel

Manuscript received March 5, 2017; revised April 21, 2017 and May 25, 2017; accepted July 6, 2017. Date of publication July 20, 2017; date of current version January 5, 2018. 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 A. Cilardo. *(Corresponding author: Xin-Yu Shih.)*

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).

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

Digital Object Identifier 10.1109/TCSI.2017.2725338

1549-8328 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.

TABLE I

46-MODE FFT SUPPORTING FOR 3GPP-LTE APPLICATIONS

Note : (N, X, Y, Z) denotes $N = 2^X * 3^Y * 5^Z$ FFT points.

All Mode Combination Set According to Radix-5, Radix-3^2 and Radix-2^4 Bases										
C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**
(900, 2, 2, 2)	**(1200, 4, 1, 2)**	**(1080, 3, 3, 1)**	**(720, 4, 2, 1)**	**(960, 6, 1, 1)**	**(972, 2, 5, 0)**	**(1296, 4, 4, 0)**	**(864, 5, 3, 0)**	**(1152, 7, 2, 0)**	**(1536, 9, 1, 0)**	**(2048, 11, 0, 0)**
	(600, 3, 1, 2)	**(540, 2, 3, 1)**	**(360, 3, 2, 1)**	**(480, 5, 1, 1)**		**(648, 3, 4, 0)**	**(432, 4, 3, 0)**	**(576, 6, 2, 0)**	**(768, 8, 1, 0)**	**(1024, 10, 0, 0)**
	(300, 2, 1, 2)		**(180, 2, 2, 1)**	**(240, 4, 1, 1)**		**(324, 2, 4, 0)**	**(216, 3, 3, 0)**	**(288, 5, 2, 0)**	**(384, 7, 1, 0)**	**(512, 9, 0, 0)**
				(120, 3, 1, 1)			**(108, 2, 3, 0)**	**(144, 4, 2, 0)**	**(192, 6, 1, 0)**	**(256, 8, 0, 0)**
				(60, 2, 1, 1)				**(72, 3, 2, 0)**	**(96, 5, 1, 0)**	**(128, 7, 0, 0)**
								(36, 2, 2, 0)	**(48, 4, 1, 0)**	**(64, 6, 0, 0)**
									(24, 3, 1, 0)	**(32, 5, 0, 0)**
									(12, 2, 1, 0)	**(16, 4, 0, 0)**
										(8, 3, 0, 0)
										(4, 2, 0, 0)

Total number of mode = 46

with seven types is used for providing different combined-radix processing with only one integrated computing engine hardware. (2) Efficient FIFO management scheme is used to overcome the difficulty of handling complicated FIFO usage in various operating modes. (3) Single-table approximation method is responsible for reducing look-up table (LUT) hardware area because LUT becomes an area-dominated term in the multi-mode FFT design, especially for the long-length FFT. Section III demonstrates an ASIC implementation with synthesis and post automatic place-and-route results. Our FFT chip design with other state-of-the-art works is compared in Section IV. Finally, our design work is concluded in Section V.

II. PROPOSED RECONFIGURABLE HARDWARE ARCHITECTURE

In this section, in order to satisfy 46 different FFT sizes (46 operating modes) defined in 3GPP-LTE applications, we propose a reconfigurable combined-radix FFT hardware architecture and three developed design techniques as follows.

1) Reconfigurable processing kernel with seven types (**RPK-ST**).

2) Efficient FIFO management scheme (**EFMS**).

3) Single-table approximation method (**STAM**).

A. Reconfigurable 46-Mode Combined-Radix FFT Hardware Architecture

By observation, all kinds of the N-FFT-point selections defined in 3GPP-LTE applications are related to three prime numbers, 2, 3, and 5. In mathematics nature, N is directly factorized with these three important numbers, such as (1).

$$N = 2^X * 3^Y * 5^Z \tag{1}$$

where the FFT point (N) ranges from 4 to 2048. All the possible (N, X, Y, Z) values are listed in TABLE I. 46 FFT modes are divided into eleven categories, C1-C11, in accordance with the different (Y, Z) combinations. Different categories have various ranges of FFT points. Take something for examples. C5 has a range of [60, 960] whereas C9 has a range of [36, 1152]. N has a maximum of 2048, which belongs to C11 and combines with power of 2 only.

Before demonstrating our proposed hardware architecture, we should decompose and derive the FFT mathematical equation with different combined-radix foundation forms first.

Assume that n and k can be decomposed as (2) and (3)

$$\mathrm{n} = \frac{N}{A}n_1 + \frac{N}{A*B}n_2 + \frac{N}{A*B*C}n_3 + n_4, \tag{2}$$

$$\mathrm{k} = k_1 + Ak_2 + (A*B)\,k_3 + (A*B*C)\,k_4, \tag{3}$$

where $\mathrm{A} = 2^{\mathrm{X}}$, $\mathrm{B} = 3^{\mathrm{Y}}$, and $\mathrm{C} = 5^{\mathrm{Z}}$. Also, n_1 - n_4 and k_1 - k_4 are non-negative integers.

Therefore, the FFT mathematical equation is rewritten as (4).

$$\begin{aligned}
&\mathrm{X}\,[\mathrm{k}]\\
&\quad= \sum\nolimits_{n=0}^{N-1} x\,(n)\,W_N^{nk}, \quad k\!=0,1,2,\ldots,\mathrm{N}-1\\
&\quad\rightarrow \mathrm{X}[k_1 + Ak_2 + (A*B)\,k_3 + (A*B*C)\,k_4]\\
&\quad= \sum\nolimits_{n_4=0}^{\frac{N}{A*B*C}-1}\sum\nolimits_{n_3}^{C-1}\sum\nolimits_{n_2=0}^{B-1}\sum\nolimits_{n_1=0}^{A-1}\\
&\qquad\times x\left(\frac{N*n_1}{A}+\frac{N*n_2}{A*B}+\frac{N*n_3}{A*B*C}+n_4\right)\\
&\qquad* W_N^{(\frac{N}{A}n_1+\frac{N}{A*B}n_2+\frac{N}{A*B*C}n_3+n_4)(k_1+Ak_2+(A*B)k_3+(A*B*C)k_4)}\\
&\quad= \sum\nolimits_{n_4=0}^{\frac{N}{A*B*C}-1}\sum\nolimits_{n_3}^{C-1}\sum\nolimits_{n_2=0}^{B-1}\sum\nolimits_{n_1=0}^{A-1}\\
&\qquad\times x\left(\frac{N*n_1}{A}+\frac{N*n_2}{A*B}+\frac{N*n_3}{A*B*C}+n_4\right)\\
&\qquad*[W_N^{\frac{N}{A}n_1k_1}\,W_N^{\frac{N}{A*B}n_2(k_1+Ak_2)}\,W_N^{\frac{N}{A*B*C}n_3(k_1+Ak_2+(A*B)k_3)}\\
&\qquad* W_N^{n_4(k_1+Ak_2+(A*B)k_3+(A*B*C)k_4)}]
\end{aligned}\tag{4}$$

From the FFT mathematical equation factored with A, B, and C, we can develop the hardware architecture based on combined radix-5, radix-3^2, and radix2^4 SDF FFT design approaches. As depicted in Fig. 2(a), it demonstrates a block diagram of proposed reconfigurable 46-mode combined-radix

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.

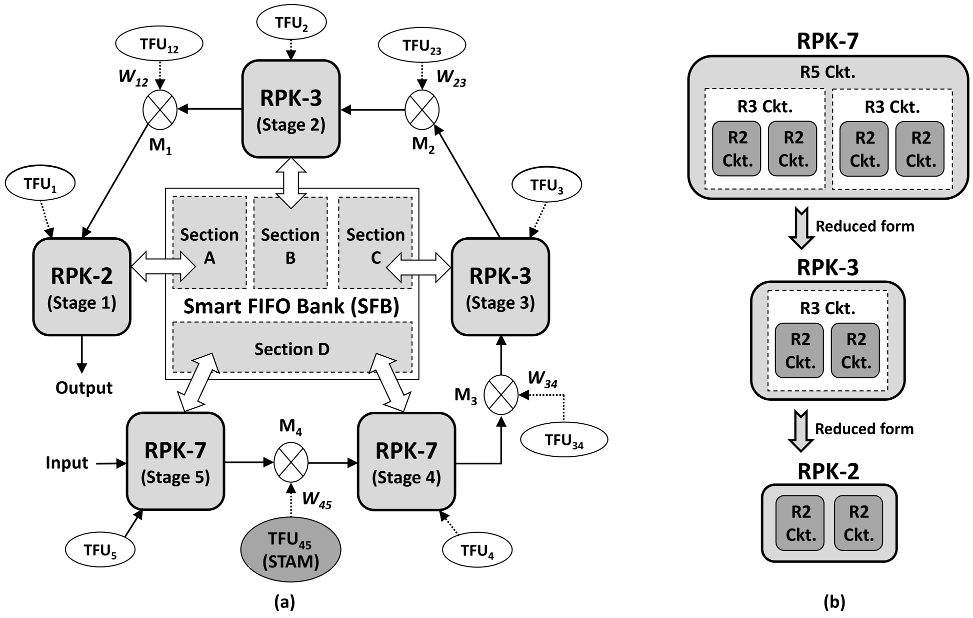

Fig. 2. Proposed reconfigurable 46-mode combined-radix FFT hardware architecture: (a) block diagram (b) relationship of RPK-7, RPK-3, and RPK-2.

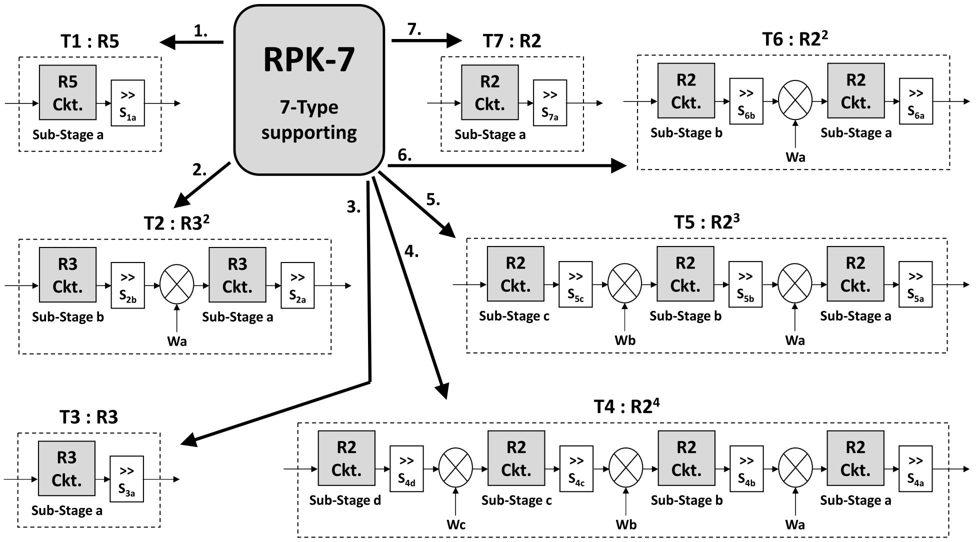

Fig. 3. Proposed reconfigurable processing kernel with seven types (RPK-ST): different configuration types of RPK-7.

FFT hardware architecture. There are three main constitutive portions, including reconfigurable processing kernel (RPK), smart FIFO bank (SFB), and twiddle factor unit (TFU).

First, in order to meet the maximal FFT-point ($N = 2048$) demand, the proposed system requires five computing stages of RPK. There are two RPK-7, two RPK-3, and one RPK-2 circuit modules from input to output ends. As depicted in Fig. 2(b), RPK-7 indicates reconfigurable processing kernel with seven types, which is constructed according to radix-5 computing design approach. RPK-7 will be discussed in more details in Section II.B. In addition, RPK-3 (RPK with three types) has a reduced form of RPK-7 whereas RPK-2 (RPK with two types) has a reduced form of RPK-3. Once RPK-7 has been completely built-up, RPK-3 and RPK-2 are easily derived from retaining the necessary computation circuits in RPK-7.

Second, smart FIFO bank (SFB) is utilized to store all of the temporary data for an overall FFT system. SFB is mainly designed to serve for assisting five RPK modules' computing. For clear illustration, it is divided into four sections, Section A-D. Section A-C are used to exchange the data with RPK stage 1 - 3, respectively. Section D is responsible for keeping the data to/from both RPK stage 4 - 5. The detailed SFB design with proposed efficient FIFO management scheme (EFMS) will be introduced in Section II.C.

Thirdly, twiddle factor units (TFUs) are used to provide the according values for trivial and non-trivial multipliers. TFU$_i$ ($i = 1$-5) produce trivial multiplication values whereas TFU$_j$ ($j = 12$, 23, 34, and 45) generate non-trivial multiplication values. The proposed single-table approximation method (STAM) is applied on TFU$_{45}$ for area reducing (discussed in Section II.D).

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.

TABLE II
CONFIGURATION TYPE PROFILES OF RPK-7, RPK-3, AND RPK-2

Type Index	Configuration Type	Types Supported			Number of CM Used	Number of CA Used	Number of TTM	Adjustable Values of Shift Amount { Si }
		RPK-7	RPK-3	RPK-2				
T1	Radix-5	V			4	18	0	$S_{1a} \in \{0,1,2,3\}$
T2	Radix-3^2	V			3	12	1	$S_{2a}, S_{2b} \in \{0,1,2\}$
T3	Radix-3	V	V		1	6	0	$S_{3a} \in \{0,1,2\}$
T4	Radix-2^4	V			3	8	3	$S_{4a}, S_{4b}, S_{4c}, S_{4d} \in \{0,1,2\}$
T5	Radix-2^3	V			2	6	2	$S_{5a}, S_{5b}, S_{5c} \in \{0,1,2\}$
T6	Radix-2^2	V	V	V	1	4	1	$S_{6a}, S_{6b} \in \{0,1,2\}$
T7	Radix-2	V	V	V	0	2	0	$S_{7a} \in \{0,1,2\}$

Note : CM = Complex Multiplier, CA = Complex Adder, TTM = Trivial Twiddle-factor Multiplier

TABLE III
SQNR IMPROVED BY RIGHT SHIFT OPERATIONS FOR DIFFERENT FFT-POINTS

Selected Case Index	FFT Point (N)	Stage 1		Stage 2		Stage 3		Stage 4				Stage 5		SQNR without Shift (dB)	SQNR with Shift (dB)	SQNR Improved (dB)
1	2048	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{4a}	S_{4b}	S_{4c}	S_{4d}	S_{7a}		0.16	35.84	35.68
		1	1	1	1	1	1	1	1	1	1	1				
2	1536	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{5a}	S_{5b}	S_{5c}		S_{3a}		0.19	38.83	38.64
		0	1	1	1	1	1	1	1	1		2				
3	1296	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{3a}		S_{2a}	S_{2b}			S_{3a}		0.24	49.77	49.53
		0	1	1	1	1		1	1			2				
4	1200	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{3a}		S_{1a}				S_{1a}		0.21	44.03	43.82
		0	1	1	1	1		2				3				
5	1152	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{7a}				S_{2a}	S_{2b}	0.21	43.61	43.40
		0	1	1	1	1	1	1				1	2			
6	972	S_{6a}	S_{6b}	S_{3a}		S_{3a}		S_{2a}	S_{2b}			S_{3a}		0.24	53.19	52.95
		0	1	1		1		1	1			2				
7	900	S_{6a}	S_{6b}	S_{3a}		S_{3a}		S_{1a}				S_{1a}		0.24	48.01	47.77
		0	1	1		1		2				3				
8	576	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{2a}	S_{2b}					0.32	46.83	46.51
		0	1	1	1	1	1	1	2							
9	540	S_{6a}	S_{6b}	S_{3a}		S_{3a}		S_{3a}				S_{1a}		0.74	51.32	50.58
		0	1	1		1		1				3				
10	256	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{6a}	S_{6b}	S_{6a}	S_{6b}					0.53	48.08	47.55
		1	1	1	1	1	1	1	1							
11	216	S_{6a}	S_{6b}	S_{7a}		S_{3a}		S_{2a}	S_{2b}					0.56	54.76	54.20
		0	1	1		1		1	2							
12	180	S_{6a}	S_{6b}	S_{3a}		S_{3a}		S_{1a}						0.62	53.84	53.22
		0	1	1		1		3								
13	60	S_{7a}		S_{7a}		S_{3a}		S_{1a}						1.19	54.06	52.87
		0		1		1		3								
14	36	S_{6a}	S_{6b}	S_{3a}		S_{3a}								1.61	60.45	58.84
		0	1	1		2										
15	12	S_{6a}	S_{6b}	S_{3a}										3.35	64.14	60.79
		0	1	2												

In addition, multiple-input multiple-output (MIMO) application is also a popularly discussed topic for current research. Our proposed hardware has circuit scalability and behaves extensible property because it can be also applied for any TX/RX signal streams in MIMO system. In a baseband design structure, each TX/RX stream has an independent data processing chain. In other words, each one steam has individual FFT-calculating on demand. There is no cross-processing among FFT modules of different chains. We can employ our proposed hardware architecture with three design techniques on FFT circuit realization of each individual chain.

B. Reconfigurable Processing Kernel With Seven Types

As dealing with various FFT-size modes defined in 3GPP-LTE systems, it is very tedious to individually construct

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.

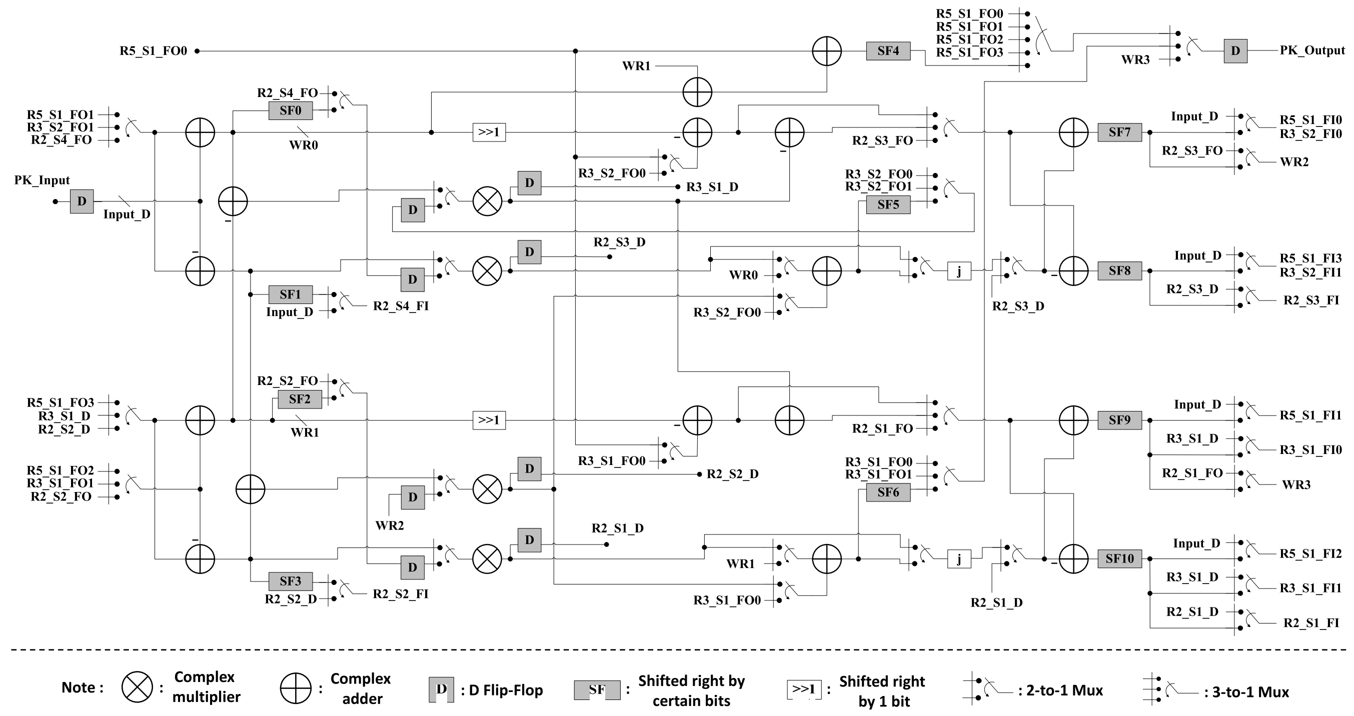

Fig. 4. Proposed hardware circuit of reconfigurable processing kernel with seven types (RPK-ST): RPK-7.

different radixes of processing kernels. Instead, we propose a reconfigurable processing kernel with seven types (RPK-7) to achieve radix-5, radix-3^2, and radix2^4 FFT computing. The hardware design structure is mainly based on radix-5 FFT computation. Via reconfigurable property, the processing kernel can perform other combined-radixes of FFT operations. As illustrated in Fig. 3, T1 - T7 denote seven configuration types. T1 represents that RPK is considered as one radix-5 FFT computing, activating all of the logic circuit elements. After the basic radix-5 FFT computing, the computed output is shifted right by the shift amount, S_{1a}, which can be chosen from {0, 1, 2, 3}. The reason of applied right shift operation is to improve the signal-to-quantization ratio (SQNR) effects while the input and output ends of RPK have the same wordlength (shifting effects will be discussed later). However, the purpose of the identical input/output wordlength is to make the circuit connection of two successive stages easier.

T2 and T3 belong to radix-3^i($i = 2$ or 1) FFT computing. As for T2, RPK is utilized to perform one radix-3^2 FFT computing, reconfigured as two sub-stages. The multiplier with trivial value (Wa) connects these two sub-stages. In the similar manner, each sub-stage has a right shift operation, such as S_{2a} or S_{2b}. They can be independently selected from {0, 1, 2}. T3, the reduced operation of T2, is served as one radix-3 FFT computing. The remaining configuration types, T4 - T7, are related to radix-2^i($i = 4$, 3, 2, or 1) FFT computing. All of the configuration type profile is listed in TABLE II. Also, all possible shift amount selections for seven configuration types are shown in more details.

As shown in TABLE III, we can choose 15 representative FFT modes out of 46 ones. As for the selected case index of 4 ($N = 1200$), RPK stage 1 - 5 are operated as {T6, T6, T3, T1, T1} configuration types, respectively. If there are no right shift actions for each stage, the corresponding SQNR is very worse (such as 0.21 dB), causing FFT-operation malfunction. Instead, we choose $\{S_{6a} = 0, S_{6b} = 1\}$ at stage 1, $\{S_{6a} = 1, S_{6b} = 1\}$ at stage 2, $S_{3a} = 1$ at stage 3, $S_{1a} = 2$ at stage 4, and $S_{1a} = 3$ at stage 5. Accordingly, the SQNR is enhanced from 0.21 dB to 44.03 dB, providing SQNR improvement of 43.82 dB. Therefore, for all the representative cases, we can set up the shift amounts as specified in TABLE III. The SQNR is much improved and ranges from 35.68 dB to 60.79 dB correspondingly. All the selections are just the design references of revealing the various SQNR enhancing status, which are not the optimal ones. Actually, all of 46 FFT modes with shift amounts have the similar SQNR improvement phenomenon. Furthermore, in our developed FFT chip implementation, the shift amounts are not fixed/unchangeable as shown in TABLE III. All shift amounts for arbitrary FFT modes can be real-time determined by the users because the possible combination of shift amounts is very large beyond our imagination. By setting the shift amounts with freedom, the proposed FFT system provides more circuit flexibility, especially for different environment encountered.

In the detailed hardware circuit point-of-view, Fig. 4 depicts all of the circuit elements utilized and their interconnections. RPK-7 circuit is a 7-pipeline-stage hardware design for all seven configuration types. It primarily consists of four complex multipliers, eighteen complex adders, several controlled muxes, and real-time programmable right shifters (SF0 - SF10) determined by the users. PK_Input and PK_Output represent the input and output of RPK,

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.

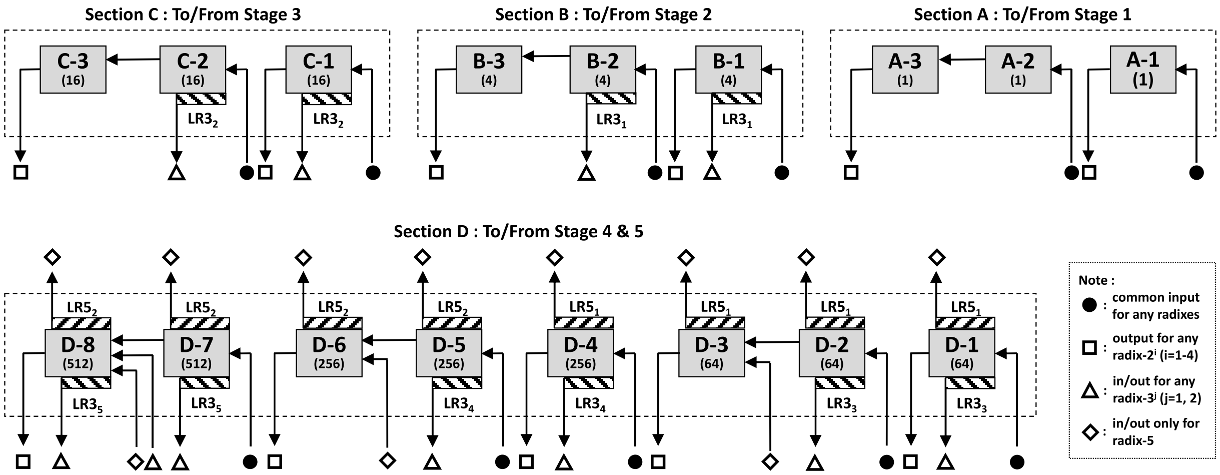
Section C : To/From Stage 3
Section B : To/From Stage 2
Section A : To/From Stage 1
C-3 (16)
C-2 (16)
C-1 (16)
LR3₂
LR3₂
B-3 (4)
B-2 (4)
B-1 (4)
LR3₁
LR3₁
A-3 (1)
A-2 (1)
A-1 (1)
Section D : To/From Stage 4 & 5
LR5₂
LR5₂
LR5₂
LR5₂
LR5₁
LR5₁
LR5₁
LR5₁
D-8 (512)
D-7 (512)
D-6 (256)
D-5 (256)
D-4 (256)
D-3 (64)
D-2 (64)
D-1 (64)
LR3₅
LR3₅
LR3₄
LR3₄
LR3₃
LR3₃
Note :
● : common input for any radixes
□ : output for any radix-2ⁱ (i=1-4)
△ : in/out for any radix-3ʲ (j=1, 2)
◇ : in/out only for radix-5

Fig. 5. Proposed hardware circuit of smart FIFO bank (SFB) with efficient FIFO management scheme (EFMS).

respectively. R5_S1_FIi (i = 0-3) and R5_S1_FOj (j = 0-3) indicate five FIFO inputs and outputs of T1 configuration type, respectively. R3_S1_FIi (i = 0 or 1) denote two FIFO inputs of "sub-stage a" in both T2/T3 configuration types whereas R3_S2_FIi (i = 0 or 1) denote two FIFO inputs of "sub-stage b" in T2 configuration type. The other naming convention has the similar meanings for FIFO input/output signals accordingly. As for main hardware resource usage status, TABLE II summarizes that different configuration types have various complex multiplier/adder circuit usage. In the circuit speed analysis respect, the critical path is located on computation time of one complex multiplier, four complex adders, and six controlled muxes in series.

C. Efficient FIFO Management Scheme

In tradition, as dealing with a pure radix-r based SDF FFT ($r = 2, 3, 5, \ldots$), each computing stage requires (r-1) FIFOs, of which each FIFO has a length of r^i (i is a non-negative integer). As a result, for a pure 2048-point SDF FFT system (single- mode processing), it entirely needs length-(2048-1) FIFOs no matter what radix-r based scheme is actually adopted. While we have to face with different-radixes based FFT computing, we don't want to individually maintain different FIFO storage elements for each radix based FFT design. Therefore, by utilizing hardware reuse property and avoiding hardware waste, we propose a smart FIFO bank (SFB) with elaborate design technique, efficient FIFO management scheme (EFMS).

In hardware structure, SFB is divided into four different lengths of FIFO sections (Section A - D in Fig. 5). Section A - C are to not only store the data, but also pass the necessary signals to/from RPK stage 1-3, respectively. These three sections have a similar FIFO constructed structure. Each section has three FIFO sub-banks and the length of each FIFO sub-bank is of 4^L (L = 0, 1, 2). On the other hand, Section D is responsible for storing and exchanging data, which are related to both RPK stage 4 and 5. The design structure of Section D is different with that of the previous three sections. Section D has eight FIFO sub-banks, including three length-64, three length-256, and two length-512 ones. Thus, for a whole 46-mode FFT system with maximal 2048-point FFT manipulation, our proposed SFB still only requires length-(2048-1) FIFOs totally. In FIFO hardware resource, there is no hardware overhead with respect to a traditionally pure 2048-point FFT computing.

In the input/output circuit connection viewpoints, we mostly keep the same input locations for all based radixes, except few situations as shown in Fig. 5. But the output locations are different for all seven configuration types. First, as for radix-2^i (i = 1-4) configuration types, all the outputs are available at two possible positions (square-marked graphics), including direct FIFO sub-bank output or two concatenated FIFO sub-bank output. Second, as for radix-3^j (j = 1-2) configuration types, the outputs are denoted as triangle-marked graphics and their output locations have an actual FIFO length of $LR3_k$ (k = 1-5). Different FFT modes own different $LR3_k$. The maximum lengths of $LR3_1$- $LR3_5$ are designed as {4, 16, 64, 192, 512}, respectively. Thirdly, as for radix-5 configuration type, the outputs are indicated with diamond-marked graphics. In the similar manner, the output locations are related to the dedicated length parameters, $LR5_1$ and $LR5_2$. The maximal value of $LR5_1$ is 48 whereas that of $LR5_2$ is 240. Furthermore, we consider $LR3_1$- $LR3_5$ and $LR5_1$- $LR5_2$ as adjustable FIFO length parameters, which are very important for proposed SFB setting.

TABLE IV presents different settings of RPK and SFB for 15 cases, selected out of 46 FFT modes. Take the selected case index of 9 ($N = 540 = 2^2 * 3^3 * 5^1$) for examples. RPK stage 1-5 are performed as {T6, T3, T3, T3, T1} configuration types, respectively. The FIFO length parameters needed, $LR3_1$, $LR3_2$, $LR3_3$, and $LR5_2$, should be correspondingly set as {4, 12, 36, 108}, respectively. The other three FIFO length parameters ($LR3_4$, $LR3_5$, and $LR5_1$) are no need to be specified (un-used length parameters for N = 540). Moreover, as the star-marked indicators in TABLE IV, seven FIFO length parameters ($LR3_1$- $LR3_5$ and $LR5_1$- $LR5_2$) reach the maximum values while N is {972, 1296, 576, 576, 1536, 1200, 1200}, respectively.

D. Single-Table Approximation Method

In most of current state-of-the-art design works, there are seldom twiddle factor generation methods discussed, but it

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.

TABLE IV

DIFFERENT SETTINGS OF RPK AND SFB AT 46-MODE DESIGN EXAMPLES SELECTED

Selected Case Index	FFT Point (N)	$N = 2^X*3^Y*5^Z$ Values			RPK : Configuration Type					SFB : FIFO Length Parameter Setting						
					RPK-2	RPK-3	RPK-3	RPK-7	RPK-7	$LR3_1$	$LR3_2$	$LR3_3$	$LR3_4$	$LR3_5$	$LR5_1$	$LR5_2$
		X	Y	Z	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5							
1	2048	11	0	0	T6	T6	T6	T4	T7	NN	NN	NN	NN	NN	NN	NN
2	1536	9	1	0	T6	T6	T6	T5	T3	NN	NN	NN	NN	512 *	NN	NN
3	1296	4	4	0	T6	T6	T3	T2	T3	NN	16 *	48	144	432	NN	NN
4	1200	4	1	2	T6	T6	T3	T1	T1	NN	16 *	NN	NN	NN	48 *	240 *
5	1152	7	2	0	T6	T6	T6	T7	T2	NN	NN	NN	128	384	NN	NN
6	972	2	5	0	T6	T3	T3	T2	T3	4 *	12	36	108	324	NN	NN
7	900	2	2	2	T6	T3	T3	T1	T1	4 *	12	NN	NN	NN	36	180
8	576	6	2	0	T6	T6	T6	T2	BP	NN	NN	64 *	192 *	NN	NN	NN
9	540	2	3	1	T6	T3	T3	T3	T1	4 *	12	36	NN	NN	NN	108
10	256	8	0	0	T6	T6	T6	T6	BP	NN	NN	NN	NN	NN	NN	NN
11	216	3	3	0	T6	T7	T3	T2	BP	NN	8	24	72	NN	NN	NN
12	180	2	2	1	T6	T3	T3	T1	BP	4 *	12	NN	NN	NN	36	NN
13	60	2	1	1	T7	T7	T3	T1	BP	NN	4	NN	NN	NN	12	NN
14	36	2	2	0	T6	T3	T3	BP	BP	4 *	12	NN	NN	NN	NN	NN
15	12	2	1	0	T6	T3	BP	BP	BP	4 *	NN	NN	NN	NN	NN	NN

Note : 1. BP = Bypassing this stage 2. NN = No need for this parameter setting 3. * = Reaching the maximum FIFO length in each section of SFB

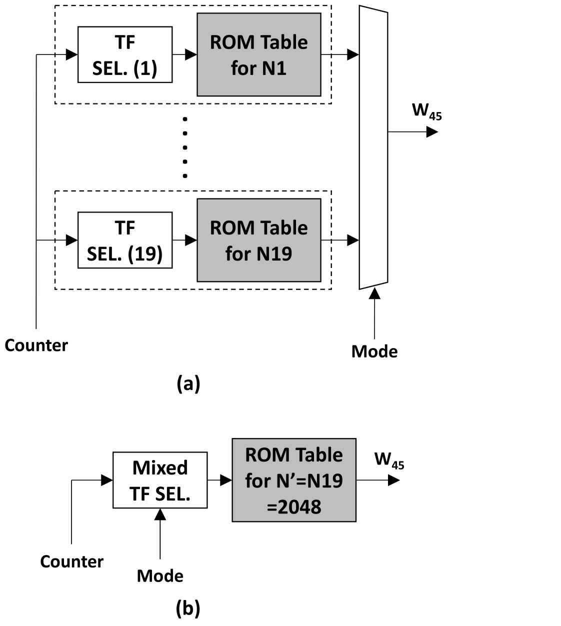

Fig. 6. Twiddle factor generation methods: (a) direct table built-up method (DTBU), (b) proposed single-table approximation method (STAM).

TABLE V

ROM TABLE ENTRY STATUS FOR DIRECT TABLE BUILT-UP METHOD APPLIED ON W_{45}

Table Index (i)	Ni of W_{Ni}^{k}	Number of ROM Table Entry Used	Table Index (i)	Ni of W_{Ni}^{k}	Number of ROM Table Entry Used
1	300	145	11	900	435
2	360	174	12	960	464
3	384	192	13	972	486
4	480	232	14	1080	522
5	540	261	15	1152	536
6	600	290	16	1200	580
7	648	324	17	1296	648
8	720	348	18	1536	768
9	768	384	19	2048	1024
10	864	432			
				Total Entry =	8245

becomes a big challenge for a multi-mode FFT system. The easiest method to fulfill this task is direct table built-up method (DTBU), as demonstrated in Fig. 6(a). For each FFT mode executed, this approach is to directly establish one dedicated and independent ROM (read-only memory) table, of which all of the table entries are pre-defined by the IC creators/designers. By utilizing this method on generating the twiddle factor of W_{45} in our FFT system (see Fig. 2(a)), it takes large design efforts to build up 19 independent ROM tables for supporting all reconfigurable FFT operations. Therefore, it is unavoidable to store so many necessary table entries, as shown in TABLE V. In our proposed FFT system, there are 19 FFT modes ($N = 300, 360, \ldots, 1536$, and 2048) required to produce the twiddle factor, W_{45}. The total number of ROM table entry needed is 8245. The fact of the larger table entries leads to not only difficult maintaining, but also a very large circuit implementation area. Instead, we propose the design technique of single-table approximation method (STAM) to overcome this difficulty.

As depicted in Fig. 6(b), STAM has two important design principles. First, STAM only builds up only one main reference ROM table (N' = N19 = 2048), instead of establishing all of 19 ROM tables independently. Second, each table entry for all ROM tables (N1 - N18) is mapped to one appreciate table

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.

TABLE VI

ANGLE ERROR AND DISTANCE ERROR STATUS FOR N15-N18 FFT-POINTS

N15 = 1152	N' = 2048	Angle Error	Distance Error	N16 = 1200	N' = 2048	Angle Error	Distance Error	N17 = 1296	N' = 2048	Angle Error	Distance Error	N18 = 1536	N' = 2048	Angle Error	Distance Error
k15	k15'	(Δθ)	(Δd)	k16	k16'	(Δθ)	(Δd)	k17	k17'	(Δθ)	(Δd)	k18	k18'	(Δθ)	(Δd)
44	78	0.0391	0.0007	69	118	0.0422	0.0007	81	128	0	0	101	135	0.0586	0.0010
45	80	0	0	70	119	0.0820	0.0014	82	130	0.0738	0.0013	102	136	0	0
46	82	0.0391	0.0007	71	121	0.0305	0.0006	83	131	0.0282	0.0005	103	137	0.0586	0.0010
47	84	0.0781	0.0014	72	123	0.0211	0.0004	84	133	0.0456	0.0008	104	139	0.0586	0.0010
48	85	0.0586	0.0010	73	125	0.0727	0.0013	85	134	0.0564	0.0010	105	140	0	0
49	87	0.0195	0.0003	74	126	0.0516	0.0009	86	136	0.0174	0.0003	106	141	0.0586	0.0010
50	89	0.0195	0.0003	75	128	0	0	87	137	0.0846	0.0015	107	143	0.0586	0.0010
51	91	0.0586	0.0010	76	130	0.0516	0.0009	88	139	0.0109	0.0002	108	144	0	0
52	92	0.0781	0.0014	77	131	0.0727	0.0013	89	141	0.0629	0.0011	109	145	0.0586	0.0010
53	94	0.0391	0.0007	78	133	0.0211	0.0004	90	142	0.0391	0.0007	110	147	0.0586	0.0011
54	96	0	0	79	135	0.0305	0.0005	91	144	0.0347	0.0006	111	148	0	0
55	98	0.0391	0.0007	80	137	0.0820	0.0014	92	145	0.0673	0.0012	112	149	0.0586	0.0010

Note : 1. (Ni, ki) denotes $W_{Ni}^{ki} = e^{-j\frac{2\pi(ki)}{Ni}}$ **2. Use (N', ki') value to approxiate (Ni, ki) value 3. Gray-colored zone (▨) indicates "error-free" case**

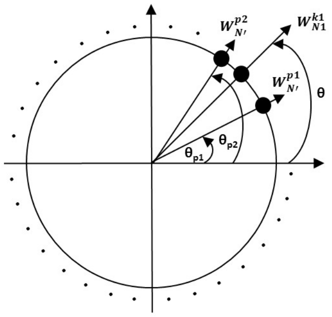

Fig. 7. Design concept of proposed STAM design technique.

entry of the main reference ROM table (N' = N19 = 2048) with smallest distance errors. Therefore, the number of the table entry that STAM entirely needs is reduced from 8245 to 1024. Totally, STAM extremely saves 87.58% of table entries with respect to direct table built-up method (DTBU).

The corresponding mapping approach is introduced as follows. Assume that the target ROM table is used to store the twiddle factors, $W_{N1}^{k1}, 0 \leq k1 < N1$, for N1-FFT-point mode. The entry-mapping by off-line determination is described by two following design rules:

1) Exact Mapping: we exactly map (N1, k1) to (N', k1') value if there exists certain k1', satisfying the equation of $W_{N'}^{k1'} = W_{N1}^{k1}$. We can consider the exact mapping as the "error-free" case.

2) Approximate Mapping: If there is no k1' to exactly meet the equation in exact mapping. There are two sub-steps to find the approximate mapping (N', k1') value.

Rule 2.a: Assume that W_{N1}^{k1} has an angle, θ, with respect to the origin in xy-plane, as depicted in Fig. 7. In a main reference twiddle factor set, $W_{N'}^{k1'}, 0 \leq k1' < N'$, we can find out two candidate twiddle factors ($W_{N'}^{p1}$ and $W_{N'}^{p2}$), of which two have angles (θ_{p1} and θ_{p2}) in xy-plane, respectively. The relationship should meet $\theta_{p1} \leq \theta$ and $\theta \leq \theta_{p2}$ simultaneously.

Rule 2.b: Assume that $W_{N1}^{k1} = cos\theta + jsin\theta = \mathrm{x}_{k1} + j\mathrm{y}_{k1}$, $W_{N'}^{p1} = cos\theta_{p1} + jsin\theta_{p1} = \mathrm{x}_{p1} + j\mathrm{y}_{p1}$, and $W_{N'}^{p2} = cos\theta_{p2} + jsin\theta_{p2} = \mathrm{x}_{p2} + j\mathrm{y}_{p2}$. We calculate two distance indicators by (5) and (6).

$$\mathrm{D1} = \sqrt{(\mathrm{x}_{k1} - \mathrm{x}_{p1})^2 + (\mathrm{y}_{k1} - y_{p1})^2} \tag{5}$$

$$\mathrm{D2} = \sqrt{(\mathrm{x}_{k1} - \mathrm{x}_{p2})^2 + (\mathrm{y}_{k1} - y_{p2})^2} \tag{6}$$

Finally, we map (N1, k1) to (N', k1') value by (7).

$$\mathrm{k1}' = \begin{cases} p1, \ when \quad D1 \leq D2 \\ p2, \ when \quad D1 > D2 \end{cases} \tag{7}$$

From the mapping rules above, we can take several ROM tables for examples (see TABLE VI). As for N15 ROM table, (N15, 45) = (N', 80) and (N15, 54) = (N', 96) are two entries of exact mapping, behaving error-free (indicated by gray-colored zone). On the contrary, (N15, 50) is mapped to (N', 89) by approximate mapping with the smallest distance error, 0.0003. Besides, as for N18 ROM table, there are more table entries with exact mapping. We can observe the regularity that one exact mapping appears with every three successive continuous-values of k18. This phenomenon results in better performance in terms of distance errors.

As for analyzing STAM's calculating precision, TABLE VII reveals maximal and average distance error status at W_{45} location with respect to all original ROM tables (N1 - N18). The maximal errors range from 1.11*10^{-3} to 1.61*10^{-3} whereas the average errors vary from 6.80*10^{-4} to 7.95*10^{-4}. Also, we observe that the error extent is proportional to the inverse of the occupied ratio of exact mapping entry. Furthermore, the last column shows that the SQNR at the final FFT output is degraded due to STAM, ranging from 0.08 to 0.80 dB only. The SQNR degradation is very small so we can neglect it.

In the aspect of design implementation with fixed-point analysis, we use TSMC 40-nm CMOS technology to make the area comparison between STAM and DTBU, as listed

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.

TABLE VII

DISTANCE ERROR AND SQNR DEGRADED DUE TO STAM DESIGN TECHNIQUE

Table Index (i)	Ni	Exact Mapping		Max. Distance Error	Average Distance Error	SQNR Degraded (dB)
		Number	Ratio			
1	300	10	6.9%	1.47*10^-3	7.64*10^-4	0.65
2	360	5	2.9%	1.49*10^-3	7.77*10^-4	0.12
3	384	64	33.3%	1.11*10^-3	6.80*10^-4	0.15
4	480	22	9.5%	1.51*10^-3	7.59*10^-4	0.24
5	540	7	2.7%	1.53*10^-3	7.69*10^-4	0.28
6	600	14	4.8%	1.51*10^-3	7.95*10^-4	0.25
7	648	7	2.2%	1.54*10^-3	7.60*10^-4	0.80
8	720	14	4.0%	1.53*10^-3	7.74*10^-4	0.09
9	768	129	33.6%	1.13*10^-3	6.80*10^-4	0.31
10	864	16	3.7%	1.51*10^-3	7.68*10^-4	0.17
11	900	26	6.0%	1.49*10^-3	7.63*10^-4	0.22
12	960	42	9.1%	1.51*10^-3	7.62*10^-4	0.18
13	972	13	2.7%	1.59*10^-3	7.69*10^-4	0.48
14	1080	10	1.9%	1.53*10^-3	7.74*10^-4	0.08
15	1152	76	14.2%	1.43*10^-3	7.53*10^-4	0.14
16	1200	27	4.7%	1.61*10^-3	7.75*10^-4	0.22
17	1296	19	2.9%	1.56*10^-3	7.63*10^-4	0.38
18	1536	256	33.3%	1.13*10^-3	6.81*10^-4	0.25

TABLE VIII

AREA ANALYSIS OF STAM WITH RESPECT TO DTBU

Method	Bits for I/Q part	ROM Table	TF SEL.	Total Hardware	Area saving	
					Value	Ratio
DTBU	12	31534.6	12220.7	43755.3	27131.4	62.01%
STAM		7052.8	9571.1	16623.9		
DTBU	14	35322.3	12413.1	47735.4	30327.3	63.53%
STAM		7689.5	9718.6	17408.1		
DTBU	16	39110.0	12630.3	51740.3	33484.9	64.72%
STAM		8377.1	9878.3	18255.4		
DTBU	18	42897.6	12654.8	55552.4	36643.7	65.96%
STAM		9013.8	9894.9	18908.7		
DTBU	20	46685.3	12715.2	59400.5	39772.0	66.96%
STAM		9701.5	9927.0	19628.5		

unit : um^2

in TABLE VIII. If utilizing wordlength of 20bits both for each real/imaginary part, the area saving ratio is around 66.96%, reducing design area from 59400.5 to 19628.5 um^2. While employing different wordlength (12 - 18 bits), the area saving ratio ranges from 62.01% - 65.96% accordingly. In conclusion, our proposed STAM design technique can achieve area reduction under the acceptable error/SQNR tolerance situation, especially for more number of ROM table supporting.

III. ASIC IMPLEMENTATION

In this section, we first show the synthesis area with sub-module circuits analyzed. Afterwards we show the ASIC implementation with post automatic placement-and-route (post-APR) performance results. The proposed hardware architecture is synthesized with Design Compiler and implemented with IC Compiler by using TSMC 40-nm CMOS technology.

TABLE IX

SYNTHESIS RESULTS OF PROPOSED HARDWARE

Sub-Module Circuits		Synthesis Area (mm^2)		
		Value	Occupied Ratio	
RPK	Stage 1 (RPK-2)	0.0016	0.62%	33.96%
	Stage 2 (RPK-3)	0.0079	3.06%	
	Stage 3 (RPK-3)	0.0079	3.06%	
	Stage 4 (RPK-7)	0.0358	13.86%	
	Stage 5 (RPK-7)	0.0345	13.36%	
SFB	Section A	0.0005	0.19%	49.01%
	Section B	0.0026	1.01%	
	Section C	0.0092	3.56%	
	Section D	0.1143	44.25%	
TFUs	TFU$_{45}$ (STAM)	0.0174	6.74%	10.92%
	Others	0.0108	4.18%	
Multipliers (M1 - M4)		0.0122	4.72%	
Control Unit		0.0036	1.39%	
Total		0.2583	100.00%	

TABLE X

POWER CONSUMPTION OF 46 FFT MODES

FFT Point (N)	Number of Stage Used	Power (mW)	FFT Point (N)	Number of Stage Used	Power (mW)	FFT Point (N)	Number of Stage Used	Power (mW)
4	1	28.67	144	4	43.62	576	4	52.45
8	2	29.88	180	4	45.92	600	5	56.20
12	2	31.00	192	4	45.67	648	5	54.36
16	2	31.28	216	4	43.46	720	5	58.10
24	3	32.81	240	4	47.05	768	5	56.78
32	3	33.51	256	4	45.59	864	5	57.04
36	3	35.15	288	4	45.69	900	5	61.28
48	3	35.68	300	5	50.50	960	5	61.60
60	4	41.05	324	4	47.82	972	5	60.37
64	3	36.77	360	5	50.22	1024	4	57.04
72	4	39.96	384	5	51.19	1080	5	61.38
96	4	40.74	432	4	49.78	1152	5	61.98
108	4	42.58	480	5	52.81	1200	5	65.53
120	4	42.51	512	4	49.93	1296	5	63.66
128	4	42.87	540	5	55.43	1536	5	63.56
						2048	5	68.65
							Average Power	48.46

A. Synthesis Results of Proposed FFT Hardware Architecture

Our proposed reconfigurable FFT hardware architecture is to provide 46 different FFT modes defined in 3GPP-LTE standard. Both the input and output wordlength are chosen as 14 bits. Also, each RPK stage connection is designed with 14 bits and STAM output has 14-bit twiddle factor values. The total design area is 0.2583 mm^2, operating at clock frequency of 555.56 MHz maximally and dissipating average power of 42.95 mW. The corresponding synthesis area results are shown in more details in TABLE IX. All of five-stage RPK sub-circuits (0.0877 mm^2) occupy 33.96% of total design area whereas SFB (0.1266 mm^2) occupies 49.01%. In SFB, Section D owns most of total area, such as 0.1143 mm^2. Section D is mainly implemented with D Flip-Flops and several SRAM memory banks, which will be demonstrated in the chip layout. In addition, the ratios of

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.

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

		Wang [37]	Xia [38]	Yang [39]	Yang [40]	Chen [41]	Proposed
	Hardware Architecture	SDC/SDF	Memory Base	MDC	SDF	Memory Base	SDF
	CMOS Technology	65 nm	55 nm	90 nm	65 nm	0.18 um	40 nm
	Supply Voltage (V)	1.0	1.08	1.0	0.45	1.8	0.99
	FFT Sizes Supported	16/64/256/1024	128/256/512/ 1024/2048	128/512/1024/2048	128/256/512/ 1024/1536/2048	12 - 1296	4 - 2048
	Number of Mode	4	5	4	6	34	46
	Max. FFT Point	1024	2048	2048	2048	1296	2048
	Wordlength (Bit)	32	16	10	12	16	14
	Clock Frequency (MHz)	400.00	122.88	40.00	20.00	122.88	500.00
	Core Area (mm^2)	1.190	0.615	3.100	1.375	25.000	0.360
	Gate Count	747K	136K	NA	1,100K	482K	380K
	Throughput (M Symbols/s)	400.00	122.88	40.00	20.00	122.88	500.00
	Power (mW)	416.80	32.40	63.72	8.55	320.00	48.46
I1	Speed-Area Ratio (10^{-3}) [41]	0.54	0.90 (2nd rank)	NA	0.02	0.25	1.32 (1st rank)
I2	Normalized Energy/FFT (nJ) [40]	1067.01	638.18 (2nd rank)	2356.22	875.52	1218.75	322.55 (1st rank)
I3	NAE (bits/mm^2) [42]	0.21	0.29 (1st rank)	0.15	0.18	0.08	0.26 (2nd rank)
I4	NEE (nJ/bit) [42]	6.00	1.82	4.79	12.16	0.60 (1st rank)	1.50 (2nd rank)

TFUs and four multipliers (M1-M4) are 10.92% and 4.72%, respectively.

B. ASIC Implementation With Post-APR Performance Results

The ASIC implementation is shown as the chip layout in Fig. 8. The proposed 46-mode FFT hardware architecture is realized in a core area of 0.6 mm x 0.6 mm (= 0.36 mm^2). It can operate up to clock frequency of 500 MHz, consuming power of 48.46 mW in average. In the memory hardware resource respect, there are two SRAM 28x512 (MEM-A and MEM-B) and three SRAM 28x256 (MEM-C, MEM-D, and MEM-E) needed, which are responsible for the Section D of SFB. Additionally, we also require one ROM 28x1024 to accomplish the ROM table storing of STAM. Each ROM table entry stores 28-bit data, including 14-bit real and imaginary parts of twiddle factors. Therefore, in the layout area size point-of-view, five SRAMs and one ROM have an occupied ratio of 20.39% and 2.14% of total core area, respectively.

In the throughput aspect, the delivered FFT system throughputs at maximal clock frequency of 500MHz are always the same (500M Symbols/s) no matter what the FFT size is. It is because our hardware architecture is mainly based on SDF FFT architecture. In nature, it can produce one individual output symbol/sample for each clock cycle. Besides, in the power dissipation respect, different FFT sizes may activate different sub-module circuits (SFB, RPK, TFUi, and multipliers (M1-M4)). Thus, under the maximal clock frequency of 500MHz, the power consumption varies as listed in TABLE X.

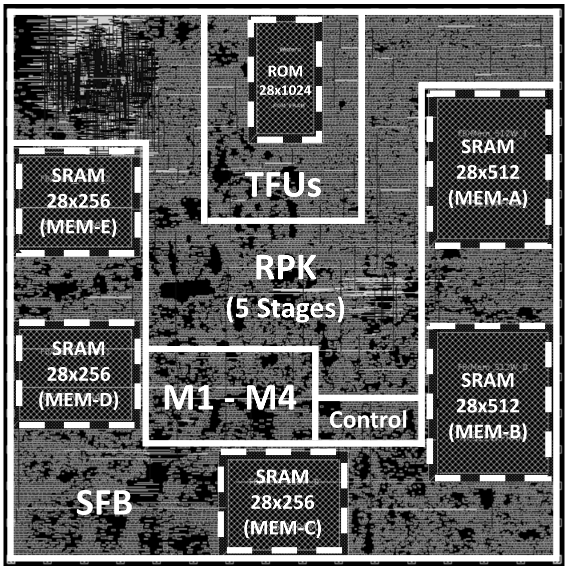

Fig. 8. Chip layout of proposed hardware architecture.

IV. CHIP DESIGN COMPARISON

TABLE XI includes our design work and other current state-of-the-art works. In absolute-value design comparison, our proposed work has not only most FFT-size modes supported (46 modes, including possible combinations of power of 2, 3, and 5), but also higher throughput achievement (500M Symbols/s). Moreover, due to different FFT sizes

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.

supported, clock frequency operated, and CMOS technology applied, we can use several performance indexes pre-defined in [40]–[42] to do a fair design comparison. In the area-related comparison, we use can speed-area ratio (I1) and normalized area efficiency (NAE, I3) indexes. On the other hand, we can utilize normalized energy per FFT-point calculating (I2) and normalized energy efficiency (NEE, I4) for energy-related comparison.

As for NAE defined in [42], the chip performance is considered as a better one when NAE is increasing. In comparison, our design work has second rank and is only a little worse than [38]. The circuit in [38] only supports 5 FFT sizes and each FFT size belongs to the power of 2 only. The corresponding processing kernel (PK) is simplest with respect to other higher radix processing, such as radix-3 and radix-5. It is obvious that the basic "radix-2" FFT computing is much easier and consumes smaller hardware area costs. On the contrary, we can support as many FFT sizes as possible (46 modes). Also, all of the FFT sizes belong to different mixed types of the power of 2, 3, and 5. In nature, the hardware complexity is higher. Therefore, our design work has smaller NAE than [38].

As for NEE defined in [42], the chip performance is considered as a better one when NEE is decreasing. In comparison, our design work has second rank and is only a little worse than [41]. The circuit in [41] only supports a maximum FFT size of 1296 points whereas our design work can handle 2048 FFT points maximally. In nature, it needs more energy/power when our circuits require more D-FFs or memory banks to store the signal data on demand. Therefore, our design work has larger NEE than [41].

Among six design works, our design work has the first rank in terms of I1 and I2 indexes. Under the feature of supporting so many FFT sizes (46 modes), our design work is also in the leading performance group (the second rank) in terms of I3 and I4 indexes.

V. Conclusion

We propose a reconfigurable FFT hardware architecture based on combined radix-5, radix-3^2, and radix2^4 design approaches. The design implementation can provide 46 different FFT sizes defined in 3GPP-LTE applications. Besides, in order to elaborate our design, three proposed design techniques include reconfigurable processing kernel with seven types, efficient FIFO management scheme, and single-table approximation method. By utilizing TSMC 40-nm CMOS technology, an ASIC implementation only has a core area occupation of 0.36 mm^2, consuming 48.46 mW under the maximal operating frequency of 500 MHz. In summary, our work has several advantages. First, it supports most number of FFT-size modes and operates at the higher clock frequency. Second, it delivers high-quality design results in the point-of-view of area- and energy-related performance indexes, offering a constructive FFT design prototyping for 3GPP-LTE applications.

Acknowledgment

The authors would like to thank the National Chip Implementation Center (CIC) for all chip design technical supporting.

References

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

[2] K. George and C.-I. H. Chen, "Configurable and expandable FFT processor for wideband communication," in *Proc. IEEE Instrum. Meas. Technol. Conf. (IMTC)*, May 2007, pp. 1–6.

[3] D. Hiremath and B. Rajeshwari, "Implementation of pipelined radix-2 FFT using SDC and SDF architecture," in *Proc. IEEE Region Conf. (TENCON)*, Nov. 2016, pp. 1660–1663.

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

[5] I. A. Qureshi and F. Qureshi, "Generalized twiddle factor index mapping of radix-2 fast Fourier transform algorithm," in *Proc. 5th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Jun. 2014, pp. 381–384.

[6] 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.

[7] 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. 2012, pp. 1–4.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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 *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2011, pp. 342–344.

[13] 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. 2016, pp. 171–174.

[14] M. Garrido, M. Á. Sánchez, M. L. López-Vallejo, and J. Grajal, "A 4096-point radix-4 memory-based FFT using DSP slices," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 1, pp. 375–379, Jan. 2017.

[15] J. Löfgren and P. Nilsson, "On hardware implementation of radix 3 and radix 5 FFT kernels for LTE systems," in *Proc. NORCHIP*, Nov. 2011, pp. 1–4.

[16] 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.

[17] T. Patyk, F. Qureshi, and J. Takala, "Hardware-efficient twiddle factor generator for mixed radix-2/3/4/5 FFTs," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, Oct. 2016, pp. 201–206.

[18] F. Qureshi, M. Garrido, and O. Gustafsson, "Unified architecture for 2, 3, 4, 5, and 7-point DFTs based on Winograd Fourier transform algorithm," *Electron. Lett.*, vol. 49, no. 5, pp. 348–349, Feb. 2013.

[19] 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.

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

[21] 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.

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

[23] 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.

[24] 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.

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.

[25] 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, Exp. Briefs*, vol. 55, no. 2, pp. 146–150, Feb. 2008.

[26] 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.

[27] 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.

[28] J. Lee and H. Lee, "A high-speed two-parallel radix-2^4 FFT/IFFT processor for MB-OFDM UWB systems," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E91-A, no. 4, pp. 1206–1211, 2008.

[29] C. Wang, Y. Yan, and X. Fu, "A high-throughput low-complexity radix-2^4-2^2-2^3 FFT/IFFT processor with parallel and normal input/output order for IEEE 802.11ad systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 11, pp. 2728–2732, Nov. 2015.

[30] M. Garrido, R. Andersson, F. Qureshi, and O. Gustafsson, "Multiplierless unity-gain SDF FFTs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 9, pp. 3003–3007, Sep. 2016.

[31] 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.

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

[33] 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.

[34] G. Wang, B. Yin, I. Cho, J. R. Cavallaro, S. Bhattacharyya, and J. Takala, "Efficient architecture mapping of FFT/IFFT for cognitive radio networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 3933–3937.

[35] J. Kim, J. Kim, J. Lee, and K. Cho, "A mixed-radix pipeline FFT processor with trivial multiplications for LTE uplink," in *Proc. IEEE Int. Symp. Consumer Electron. (ISCE)*, Sep. 2016, pp. 57–58.

[36] J. Löfgren, L. Liu, O. Edfors, and P. Nilsson, "Improved matching-pursuit implementation for LTE channel estimation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 1, pp. 226–237, Jan. 2014.

[37] M. Wang and Z. Li, "A hybrid SDC/SDF architecture for area and power minimization of floating-point FFT computations," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 2170–2173.

[38] K. Xia, B. Wu, X. Zhou, and T. Xiong, "A generalized conflict-free address scheme for arbitrary 2^k-point memory-based FFT processors," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 2126–2129.

[39] K.-J. Yang, S.-H. Tsai, and G. C. H. Huang, "MDC FFT/IFFT processor with variable length for MIMO-OFDM systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 4, pp. 720–731, Apr. 2013.

[40] 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.

[41] 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.

[42] C.-H. Lin, C.-Y. Chen, and A.-Y. Wu, "Area-efficient scalable MAP processor design for high-throughput multistandard convolutional turbo decoding," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 2, pp. 305–318, Feb. 2011.

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. He joined MediaTek in 2010 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 VLSI design architectures and hardware circuits for digital signal processing, communication systems, and channel coding of next generation applications.

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 the Smart and Novel System Design IC Laboratory, National Sun Yat-sen University, Kaohsiung, under the supervision of Prof. X.-Y. Shih. His interests include VLSI hardware design and communication ICs.

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 the Smart and Novel System Design IC Laboratory, National Sun Yat-sen University, Kaohsiung, under the supervision of Prof. Xin-Yu Shih. His interests include VLSI hardware design and communication ICs.

Authorized licensed use limited to: Zhejiang University. Downloaded on January 18,2021 at 12:43:45 UTC from IEEE Xplore. Restrictions apply.