# A Low-Power 64-Point FFT/IFFT Design for IEEE 802.11a WLAN Application

Chin-Teng Lin, Yuan-Chu Yu, and Lan-Da Van*
Dept. of Electrical and Control Engineering/Department of Computer Science, National Chiao Tung University,
Hsinchu, 300, Taiwan, R.O.C.
e-mail: ctlin@mail.nctu.edu.tw
* Department of Computer Science, National Chaio Tung University,
Hsinchu 300, Taiwan, R.O.C.
e-mail: ldvan@cs.nctu.edu.tw

**Abstract—*In this paper, we propose a cost-effective and low-power 64-point fast Fourier transform (FFT)/inverse FFT (IFFT) architecture and chip adopting the retrenched 8-point FFT/IFFT (R8-FFT) unit and an efficient data-swapping method based output buffer unit. The whole chip systematic performance concerning about the area, power, latency and pending cycles for the application of IEEE 802.11a WLAN standard has been analyzed. The proposed R8-FFT unit utilizing the symmetry property of the matrix decomposition achieves half computation-complexity and less power consumption compared with the recently proposed FFT/IFFT designs. On the other hand, applying the proposed data-swapping method, a low-cost and low-power output buffer can be obtained. So as to further increase system performance, we propose one scheme: the multiplication-after-write (MAW) method. Applying MAW method with R8-FFT unit, the resulting FFT/IFFT design not only leads to the balancing pending cycle, but also abbreviating computation latency to 8 clock cycles. Consequently, adopting the above proposed two units and one scheme, the whole chip consumes 22.36mW under 1.2V@20 MHz in TSMC 0.13 1P8M CMOS process.***

## I. INTRODUCTION

The fast Fourier transform (FFT) and discrete Fourier transform (DFT) [1] has been widely applied in the analysis and implementation of communication systems such as OFDM-based wireless local area network (WLAN) [2, 3]. In wireless communication applications, the complex sequences in the time domain are expected to be analyzed in the frequency domain via FFT computation. From existing research, there are possible four categories for the FFT/DFT computation structures: 1) butterfly-based architecture [4-6], 2) recursive-algorithm based architecture [7-8], 3) multiplier-accumulator based structure [9], and 4) ROM operation based structure [10]. In IEEE 802.11a, the required bandwidth of the transmitted signal is 20 MHz and the OFDM symbol duration is 4 us including 0.8 us for a guard interval [2]. Thus, in effect, the FFT/IFFT operation has to be computed within 3.2 us without the guard interval. It is manifest that the DFT architectures based on the recursive algorithm are more area-efficient than those realized by other approaches. However, it needs huge design effort to meet the tightly specification of wireless communication systems [2]. The conventional Cooley-Tukey radix-2 FFT algorithm requires 192 complex butterfly operations for a 64-point FFT computation. Considering that one FFT unit has to be computed within 4.0 us, one butterfly operation has to be completed within 20.8 ns, which leads to 48 MHz operation frequency for a single butterfly FFT unit. That is, the system requires a higher clock rates. The main motivation of this work is to derive and investigate an alternative FFT/IFFT architecture that satisfies the timing constraints stated in the standard IEEE 802.11a with less silicon area cost and low power consumption. The results expose that the proposed design not only achieves the smaller chip size based on one new 8-point FFT/IFFT kernel, but also reduces the processor latency and pending cycles for satisfying the better system performance. The paper is structured as follows. A fast FFT and IFFT algorithms are illustrated in Section II. In Section III, we propose the corresponding novel FFT/IFFT fabrics. In Section IV, the implementation issues will be debated. Comparison results of the 64-point FFT/IFFT are revealed in Section V. At last, the concise statements remark this paper.

## II. FFT AND IFFT ALGORITHMS

The discrete Fourier transform (DFT) of the $N$-point input $X[n]$ is defined as

$$Z[k]=\sum_{n=0}^{N-1} X[n]\cdot W_N^{kn}\,, \qquad (1)$$

where $W_N = e^{-j2\pi/N}$ . According to the decomposition method of [6],

$$Z[s+Tt]=\sum_{l=0}^{M-1}\left[W_{MT}^{sl}\cdot\sum_{m=0}^{T-1} X\left(l+Mm\right)\cdot W_T^{sm}\right]\cdot W_M^{lt} \qquad (2)$$

In this paper, we separate the 64-points DFT into two dimensional 8-point FFTs. , where the values of $M$ and $T$

can be set as $M = T = 8$. The 8-point FFT computation in Eq. (2) could be expressed as:

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \\ Y_7 \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 & W^4 & W^5 & W^6 & W^7 \\ W^0 & W^2 & W^4 & W^6 & W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^1 & W^4 & W^7 & W^2 & W^5 \\ W^0 & W^4 & W^0 & W^4 & W^0 & W^4 & W^0 & W^4 \\ W^0 & W^5 & W^2 & W^7 & W^4 & W^1 & W^6 & W^3 \\ W^0 & W^6 & W^4 & W^2 & W^0 & W^6 & W^4 & W^2 \\ W^0 & W^7 & W^6 & W^5 & W^4 & W^3 & W^2 & W^1 \end{bmatrix} \cdot \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix}. \quad (3)$$

After removing the $180^\circ$ and $90^\circ$ redundancies [1], the equation could be recast as

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \\ Y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & W_8^1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -j & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -jW_8^1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -W_8^1 & 0 \\ 0 & 1 & 0 & 0 & 0 & j & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & jW_8^1 \end{bmatrix} \cdot \begin{bmatrix} (X_0+X_4)+(X_2+X_6) \\ (X_0+X_4)-(X_2+X_6) \\ (X_0-X_4)-j(X_2-X_6) \\ (X_0-X_4)+j(X_2-X_6) \\ (X_1+X_5)+(X_3+X_7) \\ (X_1+X_5)-(X_3+X_7) \\ (X_1-X_5)-j(X_3-X_7) \\ (X_1-X_5)+j(X_3-X_7) \end{bmatrix} \quad (4)$$

Finally, it is manifest that one symmetric property with four quarters in the transform matrix could be observed. Thus, the 8-point FFT transform matrix in Eq. (4) can be decomposed as

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = G_{1(FFT)} + G_{2(FFT)} \quad \text{and} \quad \begin{bmatrix} Y_4 \\ Y_5 \\ Y_6 \\ Y_7 \end{bmatrix} = G_{1(FFT)} - G_{2(FFT)}, \quad (5)$$

where

$$G_{1(FFT)} = H_{1(FFT)} \cdot \begin{bmatrix} (X_0+X_4)+(X_2+X_6) \\ (X_0+X_4)-(X_2+X_6) \\ (X_0-X_4)-j(X_2-X_6) \\ (X_0-X_4)+j(X_2-X_6) \end{bmatrix}, \quad G_{2(FFT)} = H_{2(FFT)} \cdot \begin{bmatrix} (X_1+X_5)+(X_3+X_7) \\ (X_1+X_5)-(X_3+X_7) \\ (X_1-X_5)-j(X_3-X_7) \\ (X_1-X_5)+j(X_3-X_7) \end{bmatrix} \quad (6)$$

where H1(FFT) and H2(FFT) are defined, respectively, as

$$H_{1(FFT)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H_{2(FFT)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & W_8^1 & 0 \\ 0 & -j & 0 & 0 \\ 0 & 0 & 0 & -jW_8^1 \end{bmatrix} \quad (7)$$

In similar behaviors, we can derive the IFFT equation as follows

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} = G_{1(IFFT)} + G_{2(IFFT)} \quad \text{and} \quad \begin{bmatrix} X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix} = G_{1(IFFT)} - G_{2(IFFT)}, \quad (8)$$

where

$$G_{1(IFFT)} = H_{1(IFFT)} \cdot \begin{bmatrix} (X_0+X_4)+(X_2+X_6) \\ (X_0+X_4)-(X_2+X_6) \\ (X_0-X_4)+j(X_2-X_6) \\ (X_0-X_4)-j(X_2-X_6) \end{bmatrix}, \quad G_{2(IFFT)} = H_{2(IFFT)} \cdot \begin{bmatrix} (X_1+X_5)+(X_3+X_7) \\ (X_1+X_5)-(X_3+X_7) \\ (X_1-X_5)+j(X_3-X_7) \\ (X_1-X_5)-j(X_3-X_7) \end{bmatrix}, \quad (9)$$

where

$$H_{1(IFFT)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad H_{2(IFFT)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & W_8^{-1} & 0 \\ 0 & -j & 0 & 0 \\ 0 & 0 & 0 & jW_8^{-1} \end{bmatrix}. \quad (10)$$

The proposed Eqs. (5) and (8) will results in a low-power 8-piont FFT/IFFT kernel that will be debated in next section.

## III. Architecture Design

### A. FFT Architecture

We know that the processor latency would affect the whole chip hardware cost, which includes the buffer size and the FFT computation kernel complexity. From the results of Eqs. (5) and (8), it is manifest that the processor speed could be decided in accordance with the degree of the parallelisms in a single clock. For instance, in case four rows were grouped, eight outputs could be generated at the single clock cycle. However, this configuration would need two shift-and-add units to implement both the factors $W_8^1$ and $-jW_8^1$. Also, this configuration will not only be the highest cost due to the double multiplier units, but also the longest pending period of the FFT kernel [6]. Although the fastest FFT kernel could definitely complete the whole computation in the shortest cycle, but the serial output interface would induce a great deal of the template buffer respectively. That means the chip cost will be obviously increased by the non-balance performance consideration between the FFT computation speed and input/output data timing. Thus, we are encouraged to design an efficient architecture that could satisfy the features of the smaller chip size and the less processor latency and pending cycle.

It is worth noting that one another symmetry property exists on the matrix $G_1$, $G_2$, $H_1$ and $H_2$ between the row 1,2 and 3,4. If dividing them as two parts, then only one shift-and-add unit will be only needed in the matrix $H_2$. The Retrenched 8-point FFT Unit (R8-FFTU) revealed in Fig. 1 contains two data reorder units (DRUs) and one shift-and-add unit, which only constructed by the adders and subtracts.

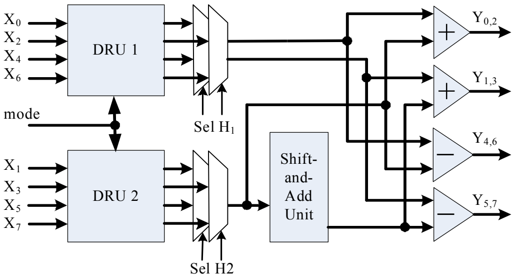

Fig. 1. Block diagram of the R8-FFTU

Then we can construct the whole system architecture as the Fig. 2. It consists of the input unit (IU), the R8-FFTU, the multiplier unit (MU), the transpose memory (TM), the output unit (OU) and the control unit (CU). The IU contains one register bank, which can store the 57 complex 16-bit wordlength data and three temporary registers. Based on the proper location arrangement in this register bank, the data is easily able to be push to the R8-FFTU [6]. The MU contains eight parallel shift-and-add units to realize the 49 different multiplications. Furthermore, it serves five different multiplications in parallel at the same time. The TM is not only used for storing the intermediate coefficient parameters, but also keeping some swapping buffer space for the minimum size of the output register bank in OU. It contains one register bank, which could store the 24 complex 16-bit wordlength data. The CU contains 6-bit master counter to control the whole procedures, and gated the unused parts in the redundant period to keep the minimum power consumption.

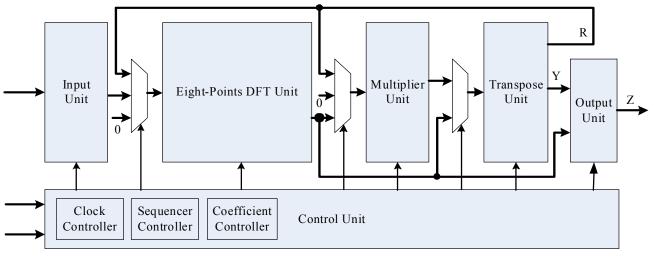

Fig. 2. Block diagram of the proposed architecture

### B. Timing Consideration

The 64-points FFT/IFFT operation sequence could be separated into the two time frames in Eq. (2), which each frame includes 16 clock cycles. During the first time frame, the output data of the R8-FFTU should be passed through the MU then stores them in the TM. After these, the data could be read out from the TM again and feedback to the R8-FFTU to generate the final output during the second time frame. In the first time frame, three design strategies should be applied to get the better configuration of the input buffer size, processor latency and multiplier utilization. First, the three registers in the IU could prevent the previous data losing in continues updating sequence. Second, the computation duration of the 16-clock of the R8-FFTU could guarantee the intermediate data could be worked out before the buffer data losing in continues updating sequence in IU. It also guarantees that the first data output from the R8-FFTU in the eighth clock cycle of the second time segment. Third, the architecture of the multiplier is implemented as the full parallel architecture as shown in Fig. 3. The interface of the MU has five ports, which could compute the five multiplications for the maximum in parallel in one clock cycle. The timing diagram has been shown as in Fig. 4, which the number inside the bracket indicated the usage of the constant name in the MU. Recall the results of the part A of the Section III, the proposed R8-FFTU could generate the four outputs at the same time. Besides, the four ports Y(0), Y(1), Y(2) and Y(3) of MU could server as the R8-FFTU multiplication interfaces, and one extra port Y(4) could also server as the re-multiplication interface to re-filled the data in TM in parallel as Fig. 4. This operation mainly reduce the pending cycles in R8-FFTU, which is called the MAW. Totally, there are five data produced via this feedback multiplication and thus the proposed architecture is capable of achieving the better system performance.

In the second time frame, TM could provide some swapping buffer space for OU. This seamless replacement strategy should also be applied to economize the output buffer size and reduced the processor latency. Our proposed architecture complete the whole computation in the first 32 clock cycles and keep the minimum output buffer size as 24 complexity data. Furthermore, the CU will gated the clock for the unused parts to keep the minimum power consumption in the other 32 clock cycles. Based on these strategies, the proposed design is capable of minimizing the buffer size as well as processor latency and balancing the pending cycle for the low-power WLAN application.

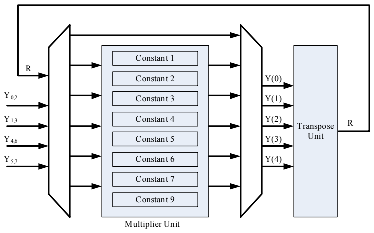

Fig. 3. Block diagram of the multiplier unit

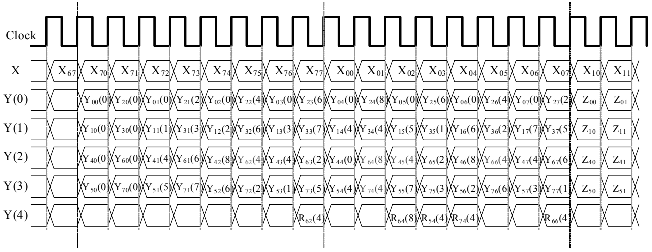
Clock
X
$X_{67}$ $X_{70}$ $X_{71}$ $X_{72}$ $X_{73}$ $X_{74}$ $X_{75}$ $X_{76}$ $X_{77}$ $X_{00}$ $X_{01}$ $X_{02}$ $X_{03}$ $X_{04}$ $X_{05}$ $X_{06}$ $X_{07}$ $X_{10}$ $X_{11}$
Y(0)
$Y_{00}(0)$ $Y_{20}(0)$ $Y_{01}(0)$ $Y_{21}(2)$ $Y_{02}(0)$ $Y_{22}(4)$ $Y_{03}(0)$ $Y_{23}(6)$ $Y_{04}(0)$ $Y_{24}(8)$ $Y_{05}(0)$ $Y_{25}(6)$ $Y_{06}(0)$ $Y_{26}(4)$ $Y_{07}(0)$ $Y_{27}(2)$ $Z_{00}$ $Z_{01}$
Y(1)
$Y_{10}(0)$ $Y_{30}(0)$ $Y_{11}(1)$ $Y_{31}(3)$ $Y_{12}(2)$ $Y_{32}(6)$ $Y_{13}(3)$ $Y_{33}(7)$ $Y_{14}(4)$ $Y_{34}(4)$ $Y_{15}(5)$ $Y_{35}(1)$ $Y_{16}(6)$ $Y_{36}(2)$ $Y_{17}(7)$ $Y_{37}(5)$ $Z_{10}$ $Z_{11}$
Y(2)
$Y_{40}(0)$ $Y_{60}(0)$ $Y_{41}(4)$ $Y_{61}(6)$ $Y_{42}(8)$ $Y_{62}(4)$ $Y_{43}(4)$ $Y_{63}(2)$ $Y_{44}(0)$ $Y_{64}(8)$ $Y_{45}(4)$ $Y_{65}(2)$ $Y_{46}(8)$ $Y_{66}(4)$ $Y_{47}(4)$ $Y_{67}(6)$ $Z_{40}$ $Z_{41}$
Y(3)
$Y_{50}(0)$ $Y_{70}(0)$ $Y_{51}(5)$ $Y_{71}(7)$ $Y_{52}(6)$ $Y_{72}(2)$ $Y_{53}(1)$ $Y_{73}(5)$ $Y_{54}(4)$ $Y_{74}(4)$ $Y_{55}(7)$ $Y_{75}(3)$ $Y_{56}(2)$ $Y_{76}(6)$ $Y_{57}(3)$ $Y_{77}(1)$ $Z_{50}$ $Z_{51}$
Y(4)
$R_{62}(4)$ $R_{64}(8)$ $R_{54}(4)$ $R_{74}(4)$ $R_{66}(4)$

Fig. 4. Timing sequence of the first time segment

## IV. IMPLEMENTATION

Concerning the chip implementation, we focus our 64-point FFT/IFFT for OFDM-based WLAN application [2, 3]. As we know, the processing time of 64-point FFT/IFFT for *IEEE* 802.11a standard must be within 3.2 μs without the guard interval. However, the proposed architecture only needs 72-clock cycle to complete the whole FFT/IFFT computations. On the other hand, the clock-timing budget of the proposed FFT/IFFT is 44.4 ns.

After the functional verification, our design in which the internal word length is 16-bit has been synthesized with Design Complier in TSMC 0.13-μm 1P8M CMOS technology. The floorplan as well as the post-layout have been carried out using Astro. After the back-annotation from Start-RC extractor, the post-simulation has been issued by NC-Simulator to verify the functionality. The static timing check can be signed-off by PrimeTime. Finally, the power analysis can be done by Astro Rail. For the post layout, the core area is 1.66 mm$^2$ and whole chip area including power rings and I/O pads is 2.58 mm$^2$. The average power dissipation of the proposed 64-point FFT/IFFT design is 22.36 mW@20 MHz at 1.2V supply voltage. The active chip layout area of the proposed design as shown in Fig. 5 is 1606 um x 1606 um, which has 77 I/O pins where 8 pins are power supply pins. The proposed 64-point FFT/IFFT design not only meets 3.2 µs timing specification for *IEEE* 802.11a standard, but also achieves the low power and cost-effective feature compared with the M-G-J's Structure [6].

## V. Comparisons and Discussions

In this section, we give a comprehensive comparison result as listed in Table 1 in terms of the number of hardwire multipliers, the register bank size of the OU and IU, power consumption, the processor latency and pending cycles. Also, we could compare the chip performance with the M-G-J's Structure [6] for the two concerns: cost and timing. For the chip cost concern, it is worth noting that the proposed design not only reduced the register bank size in OU to 42%, but also reduced the multiplier numbers to the 25%. For the chip timing and utilization concern, the proposed design not only reduces the processor latency to 61%, but also alleviates the pending cycle to 66%. Therefore, we can prove that the proposed architecture has the superior performance for the IEEE 802.11a WLAN application.

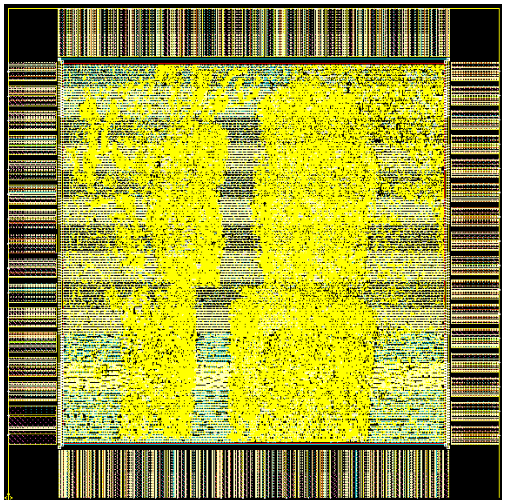

Fig. 5. The proposed low-power FFT/IFFT layout.

## Conclusions

The cost-effective and low-power 64-point FFT/IFFT architecture based on the new R8-FFT unit and low-cost output buffer unit has been devised in this paper. The analyzed results expose that the proposed VLSI architecture leads to the lowest processor latency and balancing pending cycle by MAW method between MU and R8-FFT unit. Most importantly, the proposed architecture can save chip area cost of the hardwired multiplier units in R8-FFT unit and buffer size in the OU. Thus our proposed design is certainly amenable to low power application domains.

## References

[1] W. W. Smith, J. M. Smith, *Handbook of Real-Time Fast Fourier Transforms*. Piscataway, NJ: IEEE Press, 1995, chap 3.

[2] R. D. J. van Nee and R. Prasad, *OFDM for wireless multimedia communications.* Norwood, MA: Artch House, 2000.

[3] IEEE Std. 802.11a-1999,"Wireless LAN MAC and PHY specifications — high-speed physical layer in the 5 GHz band," ISO/IEC 8802-11:1999(E)/Amd 1:2000(E), New York: IEEE, 2000.

[4] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[5] W. W. Simith, J. M. Smith, *Handbook of Real-Time Fast Fourier Transforms*. Piscataway, NJ: IEEE Press, 1995, chap 3.

[6] K. Maharatna, E. Grass, and U. Jagdhold, "A 64-point Fourier transform chip for high-speed wireless LAN application using OFDM," *IEEE Trans. of Solid-State Circuits,* vol. 39, no. 3, pp. 484-493, Mar. 2004.

[7] G. Goertzel, "An algorithm for the evaluation of finite trigonometric series," *American Math. Monthly*, vol. 65, pp. 34-35, Jan. 1958.

[8] L. D. Van, Y. C. Yu, C. M. Huang, C. T. Lin, "Low computation cycle and high speed recursive DFT/IDFT: VLSI algorithm and architecture," to appear in *Proc. IEEE Workshop on Signal Processing Systems (SiPS),* Nov. 2005, Athens, Greece.

[9] Y. H. Huang, H. P. Ma, M. L. Liou, and T. D. Chiueh, "A 1.1 G MAC/s sub-word-parallel digital signal processor for wireless communication applications," *IEEE Trans. of Solid-State Circuits,* vol. 39, no. 1, pp. 169183, Jan. 2004.

[10] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE Acoust., Speech, Signal Processing Mag.,* pp. 4-19, Jul. 1989.

Table 1. Comparison Results of the FFT/IFFT Chip Designs

| | # of Hardwire Multipliers | Output Unit Buffer Size | Input Unit Buffer Size | Power | Latency (clock cycle) | Pending Cycle (Clock cycle) |
|---|---|---|---|---|---|---|
| M-G-J's Structure [6] | 4 | 57*16 | 57*16 + 3*16 | 41 mW@20 MHz | 13 | 48 |
| The Proposed Structure | 1 | 24*16 | 57*16 + 3*16 | 22.36 mW@20 MHz | 8 | 32 |