IMPLEMENTATION OF A PROGRAMMABLE 64~2048-POINT FFT/IFFT PROCESSOR FOR OFDM-BASED COMMUNICATION SYSTEMS

Jen-Chih Kuo, Ching-Hua Wen, and An-Yeu (Andy) Wu

Graduate Institute of Electronics Engineering, and
Department of Electrical Engineering,
National Taiwan University, Taipei, 106, Taiwan, R.O.C.

ABSTRACT

***Orthogonal Frequency Division Multiplexing* (OFDM) system is famous for its robustness against frequency selective fading channel. The *Fast Fourier Transform* (FFT) and *Inverse FFT* (IFFT) processor are used as the modulation/demodulation kernel in the OFDM systems. The sizes of FFT/IFFT processors are varied in the different applications of OFDM systems. In this paper, we design and implement a programmable 64~2048-point FFT/IFFT processor to cover the different specifications of OFDM applications. The cached-memory architecture is our suggested VLSI system architecture. We implement the *Processing Element* (PE) by using CORDIC algorithm to replace the multiplier-based PE. We also proposed *π/4-prerotation* and modified EEAS-CORDIC VLSI architecture to reduce the iteration number and quantization noise. Finally, we implement the FFT processor with TSMC 0.35 *μm* 1P4M CMOS technology. The die area of the FFT/IFFT processor is 12.25 mm^2 including 2048x32 bits memory. The input/output wordlength is 16-bit wide. The chip can operate under 80 MHz and meet most standard requirements (64~2048 points).**

1. INTRODUCTION

The OFDM system is a form of *Multi-Carrier Modulation* (MCM) technologies [1]. It has been widely implemented in high digital communications, such as wireless LAN, 801.11a, digital audio/video broadcasting, ADSL, and VDSL system [2-3]. One of the main reasons is to increase the robustness against frequency selective fading or narrowband interference. The modulation/demodulation kernel in OFDM system is the FFT/IFFT operations. But the sizes of FFT/IFFT are different because of the various applications of OFDM system. Traditionally, we need to design various points of FFT/IFFT processors for every application of OFDM system individually, as shown in Table 1. This causes the waste of time and money. In this paper, we design and implement a programmable FFT/IFFT processor to be used in various OFDM-based communication systems. We also use low switching activity CORDIC-based PE design to achieve low-power consumption so as to elongate the battery life on mobile applications and to relieve the heating problem of the chip. We implement the FFT processor with TSMC 0.35 *μm* 1P4M CMOS technology. The die area of the FFT/IFFT processor is 12.25 mm^2 including 2048x32 bits SRAM. The input/output wordlength is 16-bit wide. The maximum operating frequency is 80 MHz, which can meet most existing OFDM systems using 64~2048-point FFT/IFFT.

Application	FFT/IFFT Size	Frequency spacing	T_{FFT}
WLAN	64	0.3125 MHz	3.2 *μs*
ADSL	2x256	4.3125 KHz	231 *μs*
VDSL	2x256x2^n,n=0:4	4.3125 KHz	231 *μs*
DAB	256x2^n,n=0:3	4.065x2^n KHz	31x2^n *μs*
DVB-T	8192/2048	1.116/4.464 KHz	896/224 *μs*

Table 1. FFT/IFFT Size for OFDM Applications.

2. PROPOSED SYSTEM ARCHITECTURE

There are various structures for implementations of FFT processors, such as single-memory, dual-memory, pipelined architecture, array type [4]. Typically conventional FFT algorithms are developed to minimize the number of multiplications and additions. However, the memory operations are usually ignored. Hidden memory operations might take half of the power consumption in the whole FFT calculations [5]. To reduce the number of memory access, we choose the Cached-Memory architecture [4] to realize the programmable 64~2048-point FFT processor. The basic idea of cached FFT is to reduce the number of memory access as shown in Fig.1. Instead of processing one stage of butterfly operations at a time, we store data in local storage, and process more data in one *Super-Stage (Pass0, Pass1, Pass2)* at a period of time. To achieve this, we have to design two sections of data movement operations differing from traditional FFT, as shown in Fig. 2. The resulting operations are still very regular and will not increase much complexity. The data will only read/write from the cache memory of each Super-Stage. Super-Stages can greatly reduce the number of memory access as the FFT size N becomes bigger. In this paper, we also design the programmable 64~2048 points FFT processor VLSI architecture, as shown in Fig. 3.

The programmable FFT/IFFT processor consists of four design units: *Processing Element* (PE), *Address Generator* (AG), and *Control Logic Unit* (CLU). The following sections will discuss the design issues of those units.

0-7803-7761-3/03/$17.00 ©2003 IEEE

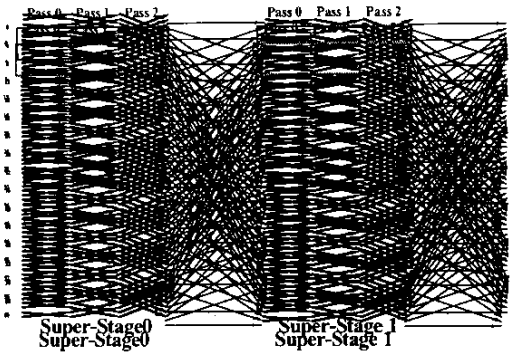

Fig. 1. Cached FFT Dataflow Diagram [4].

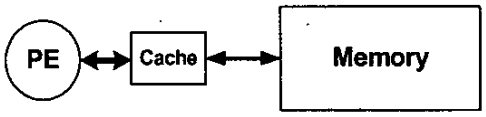

Fig. 2. Cache-Memory FFT Processor Architecture.

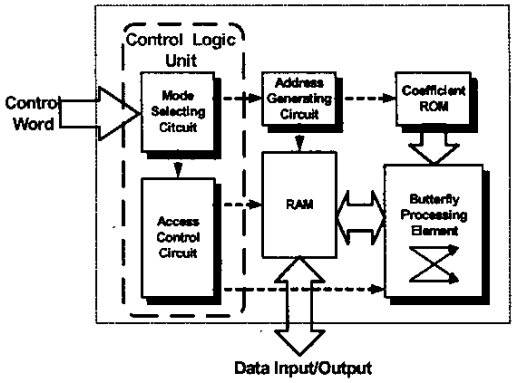

Fig. 3. The proposed programmable 64~204-point FFT/IFFT processor VLSI Architecture.

3. PROCESSING ELEMENT (PE)

The *COordinate Rotational Digital Computer* (CORDIC) algorithm is a well-known VLSI arithmetic unit. The basic concept of CORDIC is to decompose the desired rotation angles into several easy-to-be-implemented sub-angles. We adopt *Extended Elementary Angle Set* (EEAS) scheme [6] to compose these sub-angles. Each sub-angle can do one micro-rotation. The hardware requirement of CORDIC is very simple. It also has potential advantage of low switching activities for low power. Hence, we employ the modified EEAS-CORDIC VLSI architecture to design our Butterfly PE.

3.1. Modified VLSI Architecture of EEAS-CORDIC

In order to realize the IP core of EEAS-CORDIC PE, we propose the modified EEAS-CORDIC VLSI architecture, as shown in Fig.4. The important differences from the conventional CORDIC design are the parameter sequences arrangement, physical consideration and circuit speed-up. It can definitely improve the performance of CORDIC-based PE. The PE also employed the technique of Carry-Save Adder and Carry Look-ahead adder to speed up the design.

We design a single stage of EEAS-CORDIC, which includes micro-rotation mode and scaling mode. Besides, the modified EEAS-CORDIC architecture needs dedicated parameter sequences arrangement in the EEAS algorithm. We follow [6] to design those EEAS-CORDIC parameters to control the butterfly PE circuit.

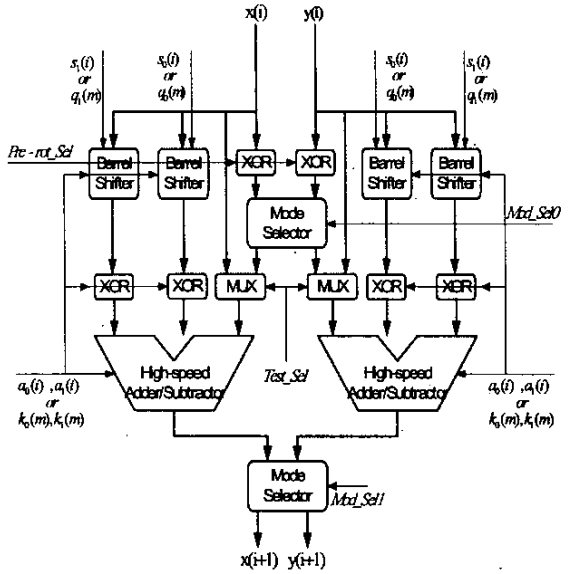

Fig. 4. Modified EEAS-CORDIC VLSI Architecture.

3.2. Reduced Coefficient ROM Design

In the traditional multiplier-based design, the coefficients that need to be stored are the sine and cosine values between [0, 2π]. Most designs store [0, $\pi/2$] sine values to save hardware cost. Some designs even use [0, π] values to prevent from the 2's complement inversion operations.

Since CORDIC algorithm is adopted as the PE kernel, the number representation is different from the 2's complement representation. Those 2's complement numbers are translated into the coefficient for shift and add operations in CORDIC-based PE. The coefficient of CORDIC is represented as eq.(1),

$$W = (sign[5:0], shift[9:0], scale[11:0]) \quad \cdots (1)$$

We can also reduce the redundant coefficient storage and save only [0, $\pi/4$] values when using a $\pi/2$-prerotation scheme. Furthermore, the proposed $\pi/4$-prerotation scheme can be used to reduce the coefficient storage to only [0, $\pi/8$] values. The comparison is shown in Table 2.

The switching activity of coefficients may cause much power consumption in 2's complement representation system. In CORDIC representation system, the representations of lower switching activity can be achieved. In Fig. 5, we can see the

differences of switching activity between 2's complement and CORDIC representations. The switching activity in CORDIC representation is lower than 2's complementation representation.

	CORDIC PE		Multiplier PE	
	$\pi/2$ *pre-rotation*	$\pi/4$ *pre-rotation*	$[0, \pi]$	$[0, \pi/2]$
Coeff. Width	*28 bits*		*12 bits*	
Total Bits	*3.5N bits*	*1.75N bits*	*6N bits*	*3N bits*

Table 2. Coefficient Storage Comparison of CORDIC-based and Multiplier-based PE.

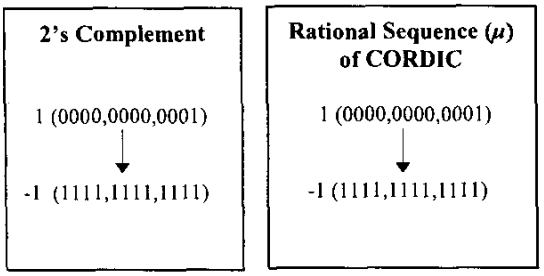

Fig. 5. Switch activity in 2's complement representation and CORDIC representation.

4. CLU AND AG DESIGNS

4.1 Control Logic Unit (CLU) Design

The topmost *Control Logic Unit* (CLU) is composed of the following three individual circuits:

a. **FFT/IFFT Operation Selection:** 1-bit input determines whether the FFT or IFFT transform should be computed.
b. **FFT Size Selection:** 3-bit FFT size selecting inputs decides which length of the FFT should be calculated.
c. **Data movement:** The processing kernel, PE and cache, and main memory can operate at different frequency to further reduce the power consumption.

4.2 Address Generator (AG) Unit Design

The cached FFT/IFFT address generation circuit can be viewed as a modified version of the traditional FFT. What we need to do is to find a grouping of the memory accesses such that a portion of the full FFT can be calculated using less than N words of memory. Table 3 [4] show the generated address in 64-point FFT.

To achieve the goal of sharing address, we can discard the LSB digits of address in high points FFT. For example, we discard the generated address of 8-points FFT processor in Fig. 6 (a) to get the final address of 4-points FFT in Fig. 6 (b).

Super-Stage	Pass	Memory Address	Cache Address	Rom Address
0	0	$b_5b_4b_3b_2b_1b_0$	$b_2b_1b_0$	00000
	1	$b_5b_4b_3b_2b_0b_1$	$b_2b_0b_1$	$b_0$0000
	2	$b_5b_4b_3b_0b_2b_1$	$b_0b_2b_1$	$b_1b_0$000
1	0	$b_2b_1b_0b_5b_4b_3$	$b_2b_1b_0$	$b_5b_4b_3$00
	1	$b_2b_0b_1b_5b_4b_3$	$b_2b_0b_1$	$b_0b_5b_4b_3$0
	2	$b_0b_2b_1b_5b_4b_3$	$b_0b_2b_1$	$b_1b_0b_5b_4b_3$

Table 3. The generated Address for 64-point Cached FFT [4].

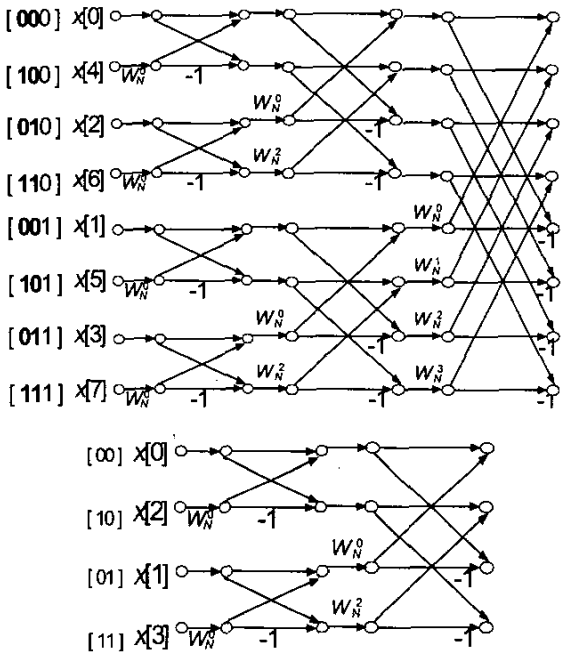

Fig. 6. (a) The generated address in 8-point FFT (b) The generated address after discarding the LSB in 4-point FFT.

5. IMPLEMENTATION RESULTS

The FFT/IFFT processor is implemented with TSMC 0.35 μm 1P4M CMOS technology. The die size is 3.9 x 3.9 mm^2 with 2048-word memory, each is 32 bits wide. The microphotograph of the processor is shown in Fig.7.

Table 4 lists the physical implementation result of this programmable 64~2048-point FFT/IFFT. The FFT sizes and the respective operating frequency and power consumption are listed in Table 5.

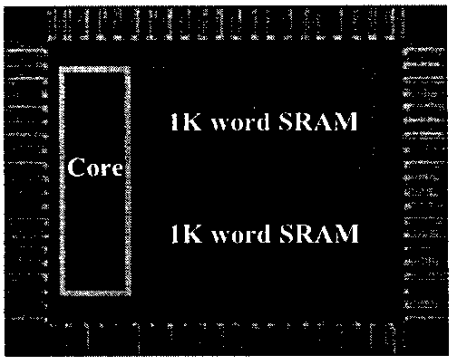

Fig. 7. Microphotograph of the FFT/IFFT Processor.

Technology	TSMC 0.35 μm 1P4M CMOS
Voltage	3.3 V
Wordlength	16 bits
Gate Counts	14,732
Memory	2x1024 Word SRAM (32 bits)
Die Size	3.9 x 3.9 mm^2
Core Size	2.6 x 2.6 mm^2
Max Frequency	80 *MHz*
Power Range	126 ~ 574 mW

Table 4. Implementation result of programmable FFT processor.

Application	WAN (IEEE 802.11a)	ADSL, VDSL, DAB	VDSL, DAB	VDSL, DAB, DVB-T
FFT Size	64	512	1024	2048
T_{FFT}	3.2 μs	62 μs	128 μs	224 μs
Operating Frequency	65 MHz	12.5 MHz	25 MHz	60 MHz
Power Consumption	545 mW	126 mW	253 mW	574 mW

Table 5. FFT Size, T_{FFT}, Frequency, and Power Consumption.

6. COMPARSION

In order to eliminate the factor of different fabrication technology, we adopt the *Normalize Index* [4]. The *Normalized Area* is the silicon area normalized to a 0.35 μm technology, as shown in eq. (2).

$$Normalized\ Area = \frac{Area}{(Technology/0.35\mu m)^2} \quad (2)$$

The FFTs per Energy, which compares the number of FFT calculation per Energy, as shown in eq. (3).

$$FFTs\ per\ Energy = \frac{Technology \times Frequency}{Power} \times 1000 \quad (3)$$

As shown in Table 6. We have pretty good performance according to the normalize index.

Processor	CMOS Tech.	FFT Size	Freq. (MHz)	Area (mm^2)	Normalized Area	FFTs per Energy
This Work	0.35	64~2048	60	12.25	12.25	36.6
Bass [4]	0.6	1024	173	42.88	14.59	116
Bidget [7]	0.5	8192	20	100	49	16.6
Colin [8]	0.6	64	18	62.4	21.23	10.8
Lihong [9]	0.6	8192	20	140	47.63	18.4

Table 6. Comparison of Various FFT/IFFT processors.

7. CONCLUSION

The programmable FFT/IFFT processor design has been demonstrated based on OFDM applications. The cached-memory architecture is chosen and then we defined the hardware architecture. Finally we finished the design of a 64~2048-point Programmable FFT/IFFT processor and work in most applications of OFDM system successfully.

8. REFERENCES

[1] Irving Kalet, "The multitone channel," *IEEE trans. On communication*, pp 119-124, February 1989.

[2] P.S. Chow, J. C. Tu, and J. M. Cioffi, "Performance Evaluation of a Multichannel Transceiver System for ADSL and VHDSL services," *IEEE J. Selected Area*, Vol. SAC-9, No. 6, pp. 909-919, Aug. 1991.

[3] R.V. Paiement, "Evaluation of Single Carrier and Multicarrier Modulation Techniques for Digital ATV Terrestrial Broadcasting," *CRC Report*, No. CRC-RP-004, Ottawa, Canada, Dec. 1994.

[4] B.M. Bass, "A Low-Power High-Performance, 1024-Point FFT Processor," *IEEE J. of Solid-State Circuit*, vol. 34 no. 3, pp. 380-387, Mar 1999.

[5] W. Li and L. Wanhammar, "A Pipeline FFT Processor," *IEEE Workshop on Signal Processing Systems*, pp. 654-662, 1999.

[6] C.S. Wu and A. Y. Wu, "A novel rotational VLSI architecture based on extended elementary-angle set CORDIC algorithm," in *Proc. IEEE 2nd IEEE Asia Pacific Conference on ASICs*, (Cheju, Korea), pp. 111-114, 2000.

[7] E. Bidet, D. Castelain, et al., "A fast single-chip implementation of 8192 complex point FFT," *IEEE Journal of Solid-State Circuits*, vol. 30, Issue 3, pp. 300-305, March 1995.

[8] C.C.W. Hui, T.J. Ding, et al., "A New FFT Architecture and Chip Design for Motion Compensation Based on Pjase Correlation," *Preceeding of International Conference, Application Specific Systems Architectures and Processors*, pp. 83-02, 1996.

[9] J. Lihong, Y. Gao, et al. "A New VLSI-Oriented FFT Algorithm and Implementation," *IEEE ASIC Conference*, pp. 337-341, 1998.