J Sign Process Syst (2012) 66:129–134
DOI 10.1007/s11265-011-0586-x

Reduced Memory and Low Power Architectures for CORDIC-based FFT Processors

Erdal Oruklu · Xin Xiao · Jafar Saniie

Received: 20 August 2010 / Revised: 23 March 2011 / Accepted: 25 March 2011 / Published online: 16 April 2011

Abstract This paper presents a pipelined, reduced memory and low power CORDIC-based architecture for fast Fourier transform implementation. The proposed algorithm utilizes a new addressing scheme and the associated angle generator logic in order to remove any ROM usage for storing twiddle factors. As a case study, the radix-2 and radix-4 FFT algorithms have been implemented on FPGA hardware. The synthesis results match the theoretical analysis and it can be observed that more than 20% reduction can be achieved in total memory logic. In addition, the dynamic power consumption can be reduced by as much as 15% by reducing memory accesses.

Keywords FFT · CORDIC · VLSI · Low power

1 Introduction

Discrete Fourier Transform (DFT) is one of the core operations in digital signal processing and communication systems. Many fundamental algorithms can be realized by DFT, such as convolution, spectrum estimation, and correlation. Furthermore, DFT is widely used in standard embedded system applications such as wireless communication protocols requiring Orthogonal Frequency Division Multiplexing [1], and radar image processing using Synthetic Aperture Radar [2] and Software Defined Radio [3]. However, DFT is difficult to implement directly due to its computational complexity. In practice, Fast Fourier transform (FFT) is used for reducing the complexity of computations.

For FFT processors, butterfly operation is the most computationally demanding stage. Traditionally, a butterfly unit is composed of complex adders and multipliers, and the multiplier is usually the speed bottleneck in the pipeline of the FFT processor. The Coordinate Rotation Digital Computer (CORDIC) [4] algorithm is an alternative method to realize the butterfly operation without using any dedicated multiplier hardware. CORDIC algorithm is versatile and hardware efficient since it requires only add and shift operations, making it suitable for the butterfly operations in FFT [5]. Instead of storing actual twiddle factors in a ROM, the CORDIC-based FFT processor needs to store only the *twiddle factor angles* in a ROM for the butterfly operation.

In recent years, several CORDIC-based FFT designs have been proposed for different applications [6–9]. In [6], non-recursive CORDIC-based FFT was proposed by replacing the twiddle factors in FFT architecture by non-iterative CORDIC micro-rotations. It reduces the ROM size; however, it does not eliminate it completely. Lin [7] proposed a "mixed-scaling-rotation" CORDIC algorithm to reduce the total iterations, but it increases the hardware complexity. Jiang [8] introduced Distributed Arithmetic to the CORDIC-based FFT algorithms, but the DA look-up tables are costly in implementation. Garrido [9] proposed a memoryless CORDIC algorithm to reduce the memory requirements for a CORDIC-based FFT processor, but implementation is complex.

Conventionally, a CORDIC-based FFT processor needs a dedicated memory bank to store the necessary twiddle factor angles for the rotation. In this study, we propose a modified CORDIC algorithm for FFT processors which eliminates the need for storing the twiddle factor angles.

E. Oruklu (✉) · X. Xiao · J. Saniie
Department of Electrical and Computer Engineering,
Illinois Institute of Technology,
3301 South Dearborn Street,
Chicago, IL 60616, USA
e-mail: erdal@ece.iit.edu

The algorithm generates the angles successively by an accumulator. With this approach, memory requirements of an FFT processor can be reduced by more than 20%. Memory reduction improves with the increased radix size. Furthermore, the angle generation circuit consumes less power consumption than angle memory accesses. Hence, the dynamic power consumption of the FFT processor can be reduced by as much as 15%. Since the critical path is not modified with the CORDIC angle calculation, system throughput does not change.

The organization of the paper is as follows: In Section 2, CORDIC algorithm fundamentals and the design of CORDIC-based FFT processor are described. The proposed memory-efficient algorithm and it's hardware architecture are presented in Section 3 for radix-r FFT where r is a power of 2. Hardware synthesis results are discussed in Section 4.

2 FFT and CORDIC Algorithm

The N-point discrete Fourier transform is defined by

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad k = 0, 1, ..., N-1, W_N^{nk} = e^{-j\frac{2\pi}{N}nk} \tag{1}$$

where $W_N^{nk} = e^{-j\frac{2\pi}{N}nk}$ is the so-called "twiddle factor". For N-point FFT, there are $log_2 N$ stages and each stage contains $N/2$ butterfly operations. The following equations describe the radix-2 butterfly operation at stage m.

$$x_{m+1}(p) = x_m(p) + x_m(q) \tag{2}$$

$$x_{m+1}(q) = [x_m(p) - x_m(q)] W_N^r \tag{3}$$

Each butterfly operation needs four data accesses (two read and two write). Two two-port memory banks can provide four data access in each clock cycle, but in this case, a special data addressing scheme is required to prevent the data conflict. In [10], a new address scheme has been proposed to realize this function and it can be extended to any radix FFT. This special addressing scheme is adopted for CORDIC based FFT implementation [11].

CORDIC algorithm was proposed by J.E. Volder [4]. It is an iterative algorithm to calculate the rotation of a vector by using only additions and shifts. Figure 1 shows an example for rotation of a vector V_i.

It can be shown that rotation can be simplified to:

$$x_{i+1} = x_i - y_i \cdot d_i \cdot 2^{-i} \tag{4}$$

$$y_{i+1} = y_i + x_i \cdot d_i \cdot 2^{-i} \tag{5}$$

Here, the direction of each rotation is defined by d_i and the sequence of all d_i 's determines the final vector. d_i is given as:

$$d_i = \begin{cases} -1 \text{ if } z_i < 0 \\ +1 \text{ if } z_i \geq 0 \end{cases} \tag{6}$$

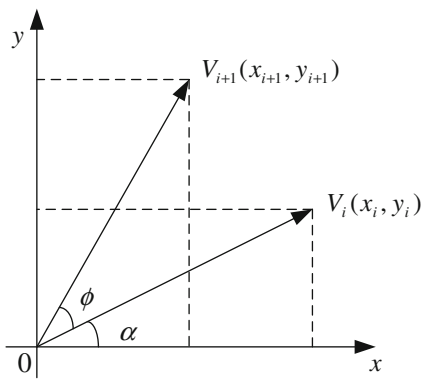


Figure 1 Rotate vector $V_i(x_i, y_i)$ to $V_{i+1}(x_{i+1}, y_{i+1})$.

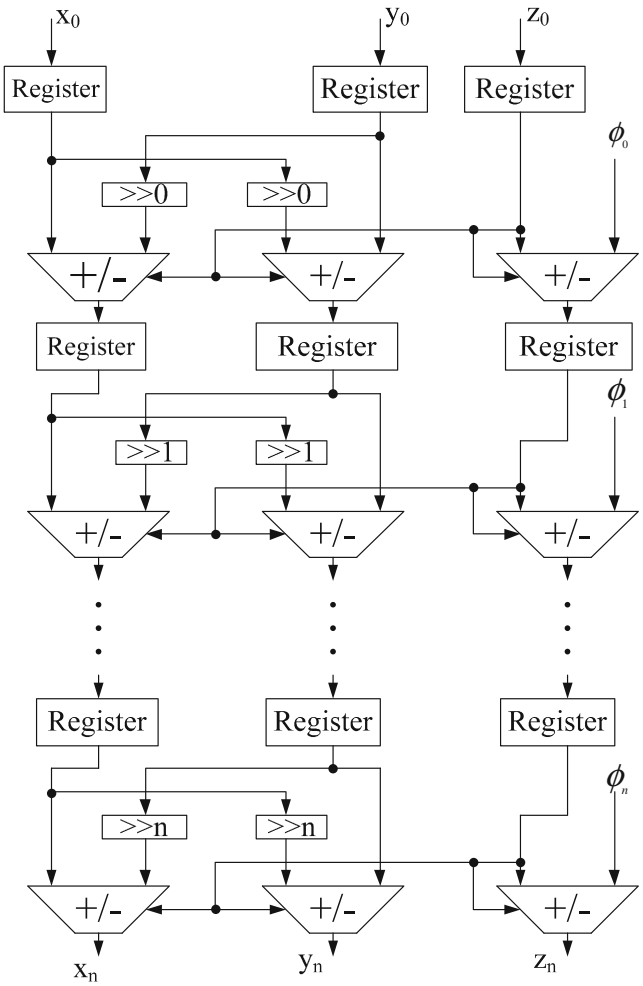


Figure 2 Proposed structure of a pipelined CORDIC unit.

Table 1 Address generation table of the proposed design for 16-point radix-2 FFT.

Butterfly counter B (b2b1b0)	Stage 0		Stage 1		Stage 2		Stage 3	
	RAM address b2b1b0	Twiddle factor angle	RAM address b0b2b1	Twiddle factor angle	RAM address b1b0b2	Twiddle factor angle	RAM address b2b1b0	Twiddle factor angle
000	000	0	000	0	000	0	000	0
001	001	$\frac{\pi}{8}$	100	0	010	0	001	0
010	010	$\frac{2\pi}{8}$	001	$\frac{2\pi}{8}$	100	0	010	0
011	011	$\frac{3\pi}{8}$	101	$\frac{2\pi}{8}$	110	0	011	0
100	100	$\frac{4\pi}{8}$	010	$\frac{4\pi}{8}$	001	$\frac{4\pi}{8}$	100	0
101	101	$\frac{5\pi}{8}$	110	$\frac{4\pi}{8}$	011	$\frac{4\pi}{8}$	101	0
110	110	$\frac{6\pi}{8}$	011	$\frac{6\pi}{8}$	101	$\frac{4\pi}{8}$	110	0
111	111	$\frac{7\pi}{8}$	111	$\frac{6\pi}{8}$	111	$\frac{4\pi}{8}$	111	0

where z_i is called angle accumulator and given by

$$z_{i+1} = (z_i - d_i \cdot \arctan 2^{-i}) \tag{7}$$

All operations described through Eqs. 4–7 can be realized by only additions and shifts; therefore, CORDIC algorithm does not require dedicated multipliers. CORDIC algorithm is often realized by pipeline structures, leading to high processing speed. Figure 2 shows the basic structure of the pipelined CORDIC unit.

As shown in Eq. 1, the key operation of FFT is $x(n) \cdot W_N^{nk}$, ($W_N^{nk} = e^{-j\frac{2\pi}{N}nk}$). This is equivalent to "Rotate $x(n)$ by angle $-\frac{2\pi}{N}nk$" operation which can be realized easily by the CORDIC algorithm. Without any complex multiplications, CORDIC-based butterfly can be fast.

An FFT processor needs to store the twiddle factors in memory. CORDIC-based FFT doesn't have twiddle factors but needs a memory bank to store the rotation angles. For radix-2, N-point, m-bit FFT, $\frac{mN}{2}$bits memory needed to store $\frac{N}{2}$angles. In the next section, a new CORDIC FFT design is presented using a single accumulator which generates all the necessary angles instantly.

3 Proposed Cordic-based FFT

In the past, several multi-bank addressing schemes have been used to realize parallel and pipelined FFT processing [12], but those methods are not suitable for the proposed memory reduced, CORDIC-based FFT. In these schemes, the twiddle factor angles are not in regular increasing order and this results in a more complex design for angle generators [11]. As shown in Table 1, using a new addressing scheme first proposed in [10], the twiddle factor angles follow a regular, increasing order, which can be generated by a simple accumulator. Table 1 shows the address generation table of the proposed design for 16-point radix-2 FFT. It can be seen that twiddle factor angles are sequentially increasing, and every angle is a multiple of the basic angle $\frac{2\pi}{N}$, which is $\frac{\pi}{8}$ for 16-point FFT.

For different FFT stages, the angles increase always one step per clock cycle. Hence, an angle generator circuit composed of an accumulator, and an output latch can realize this function, as shown in Figure 3. Control signal for the latch that enables or disables the accumulator output is simple and it is based on the current FFT butterfly stage and RAM address bits $b_2b_1b_0$ (see Table 1). Figure 4 shows the basic structure of proposed design for radix-2 FFT processing. Four registers and eight 2-to-1 multiplexers are used. Registers are needed before and after the butterfly unit to buffer the intermediate data in order to group two sequential butterfly operations together. This way, the conflict-free "in-place" data accessing can be realized. This register-buffer design can be extended to any radix FFTs.

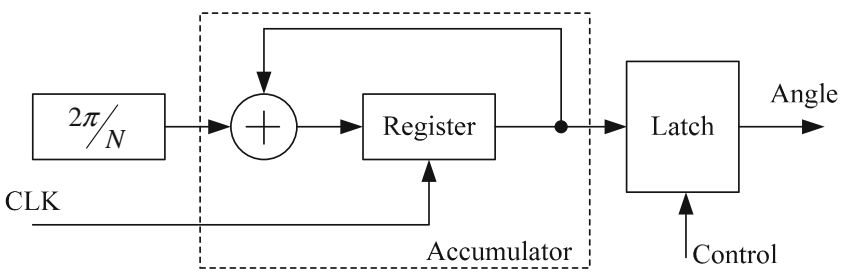


Figure 3 Angle generator for the proposed design.

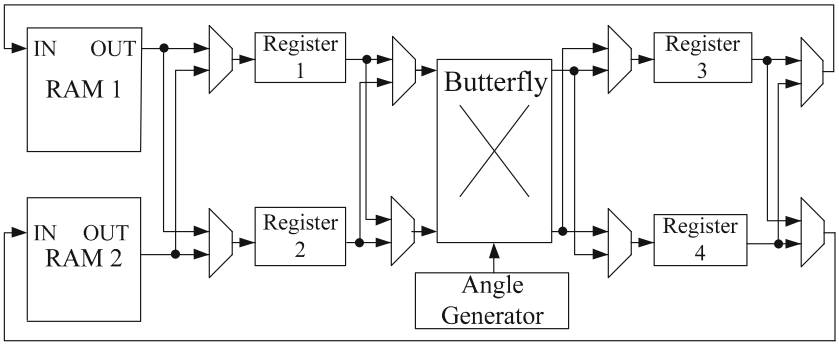


Figure 4 Memory reduced radix-2 CORDIC FFT processor.

For radix-2, the structure can be simplified by using just four registers, but for radix-r FFT, $2\times r_2$ registers are needed. Figure 5 shows the basic structure of for radix-r FFT.

Generally, for an $N=2^n$-point FFT, the addressing and control logic are mainly composed of several components: An $(n-1)$-bit butterfly counter$B = b_{n-2}b_{n-3}\ldots b_1b_0$will provide the address sequences and the control logic of the angle generator. In stage p, the memory address is given by $b_{p-1}b_{p-2}\ldots b_1b_0b_{n-2}b_{n-3}\ldots b_p$, which is rotate right p bits of butterfly counter B. Meanwhile, the control logic of the latch of the angle generator is determined by the sequence of the pattern; $b_{n-2}b_{n-3}\ldots b_p0\ldots0$ (p "0"s). For radix-2, $N=2^n$-point, m-bit FFT, (each data is $2m$-bit complex number; m-bit each for the real part and imaginary part) by using the angle generator, $\frac{5mN}{2}$ bits memory required by the conventional CORDIC can be reduced to $\frac{4mN}{2}$ which corresponds to 20% reduction. For higher radix FFT, the reduction is even more significant. For radix-r FFT, the saving is $\frac{(r-1)mN}{r}$ bits out of $\frac{(3r-1)mN}{r}$, which converges to 33.3% reduction.

4 Results and Conclusion

The proposed designs for both radix-2 and radix-4 FFT algorithms have been realized by Verilog-HDL and implemented on an FPGA chip (STRATIX-III EP3SE50C2). Synthesis results shown in Table 2 confirm that the proposed design can reduce memory usage for FFT processors without any tangible increase in the number of logic elements used when compared against the conventional CORDIC imple-

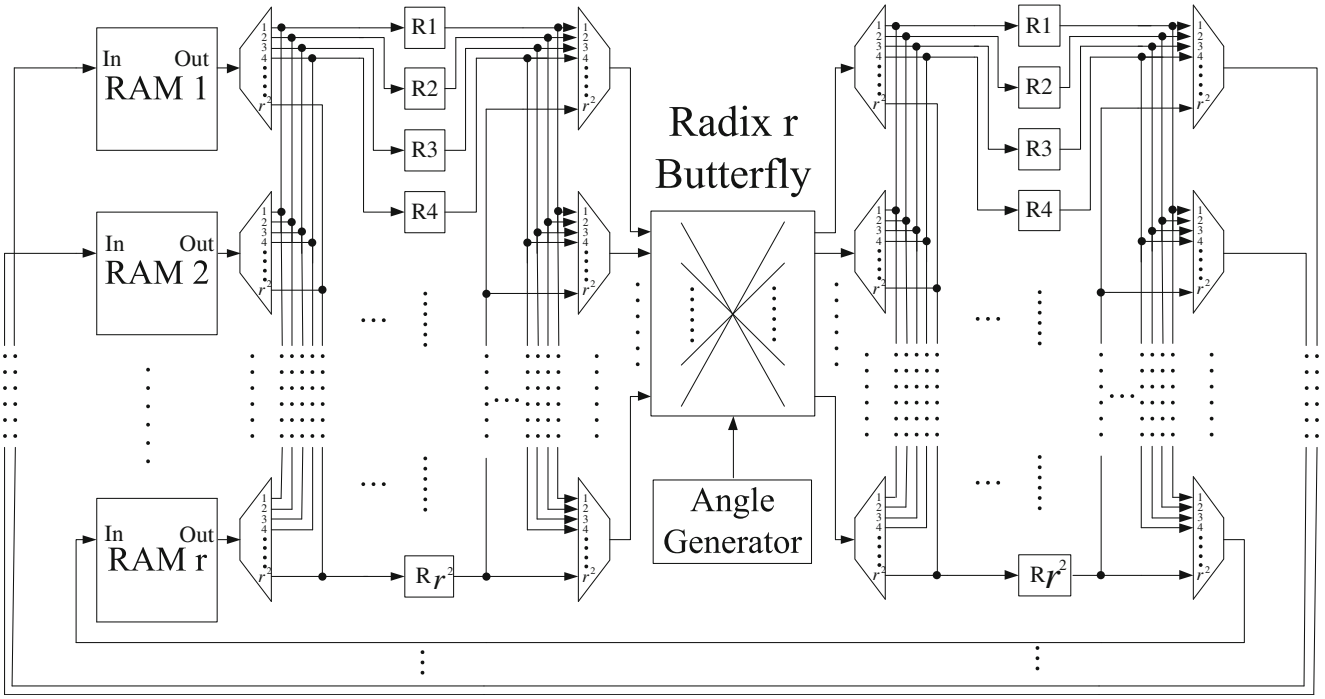


Figure 5 Memory reduced radix-r CORDIC-based FFT.

Table 2 FPGA implementation results for radix-2 and radix-4 FFT.

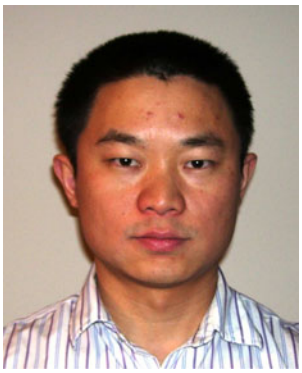
		Radix-2		Radix-4	
		Proposed CORDIC FFT Design	Conventional CORDIC FFT	Proposed CORDIC FFT Design	Conventional CORDIC FFT
256-point FFT	Total logic elements	1,427 (19-bit accumulator)	1,386	5,892 (20-bit accumulator)	5,763
	Total memory bits	8,672	10,720	8,728	11,800
	Dynamic Power	136.87 mW	156.22 mW	437.53 mW	495.06 mW
1024-point FFT	Total logic elements	1,773 (21-bit accumulator)	1,718	5,991 (22-bit accumulator)	5,797
	Total memory bits	33,248	41,440	33,304	45,592
	Dynamic Power	135.07 mW	175.98 mW	439.40 mW	496.64 mW
4096-point FFT	Total logic elements	1,809 (23-bit accumulator)	1,757	5,993 (24-bit accumulator)	5,863
	Total memory bits	131,552	164,320	131,608	180,760
	Dynamic Power	212.78 mW	242.85 mW	501.11 mW	571.72 mW

mentation (i.e., angles are stored in memory). Furthermore, dynamic power consumption is reduced (up to 15%) with no delay penalties. The implementation results are in accordance with the theoretical analysis.

References

1. Wey, C., Lin, S., & Tang, W. (2007). Efficient memory-based FFT processors for OFDM applications. In *IEEE International Conf. on Electro-Information Technology,* 345–350. May.
2. Fanucci, L., Forliti, M., & Gronchi, F. (1999). Single-chip mixed-radix FFT processor for real-time on-board SAR processing. In *6th IEEE International Conference on Electronics, Circuits and Systems, 2*, 1135–1138.
3. Mittal, S., Khan, M., & Srinivas, M. B. (2007). On the suitability of Bruun's FFT algorithm for software defined radio. In *2007 IEEE Sarnoff Symposium,* (pp. 1–5), Apr.
4. Volder, J. (1959). The CORDIC trigonometric computing technique. *IEEE Transactions on Electronic Computers, 8*(8), 330–334.
5. Despain, A. M. (1974). Fourier transform computers using CORDIC iterations. *IEEE Transactions on Electronic Computers, 23*(10), 993–1001.
6. Abdullah, S. S., Nam, H., McDermot, M., & Abraham, J. A. (2009). A high throughput FFT processor with no multipliers. In *IEEE International Conf. on Computer Design*, pp. 485–490.
7. Lin, C., & Wu, A. (2005). Mixed-scaling-rotation CORDIC (MSR-CORDIC) algorithm and architecture for high-performance vector rotational DSP applications. *IEEE Transactions on Circuits and Systems I, 52*(11), 2385–2396.
8. Jiang, R. M. (2007). An area-efficient FFT architecture for OFDM digital video broadcasting. *IEEE Transactions on Consumer Electronics, 53*(4), 1322–1326.
9. Garrido, M., & Grajal, J. (2007). Efficient memory-less CORDIC for FFT Computation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, *2*, 113–116), Apr.
10. Xiao, X., Oruklu, E., & Saniie, J. (2009). Fast memory addressing scheme for radix-4 FFT implementation. In *IEEE International Conference on Electro/Information Technology, EIT 2009*, 437–440, June.
11. Xiao, X., Oruklu, E., & Saniie, J. (2010) Reduced Memory Architecture for CORDIC-based FFT. In *IEEE International Symposium on Circuits and Systems*, 2690–2693.
12. Ma, Y. (1999). An effective memory addressing scheme for FFT processors. *IEEE Transactions on Signal Processing, 47*(3), 907–911.

Erdal Oruklu received the B.S. degree in electronics and communication engineering from Technical University of Istanbul, Istanbul, Turkey, in 1995, the M.S. degree in electrical engineering from Bogazici University, Istanbul, in 1999, and the Ph.D. degree in computer engineering from Illinois Institute of Technology, Chicago, in 2005. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, where he is also the Director of VLSI and SoC Research Laboratory. His research interests are reconfigurable computing, advanced signal processing architectures, hardware/software codesign, and embedded systems for sensors.



Xin Xiao received the Ph.D. in electrical engineering from Illinois Institute of Technology, Chicago, IL, in 2010. He is currently a senior member of technical staff with ZTE US Laboratory in NJ. His research interests are in digital circuit and system design, signal processing with FPGA/ASIC, and system design in optical signal processing.

Jafar Saniie received the B.S. degree in electrical engineering from the University of Maryland, College Park, in 1974, the M.S. degree in biomedical engineering from Case Western Reserve University, Cleveland, OH, in 1977, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1981. Since 1983, he has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, where he is a Filmer Professor, Associate Chair and Director of the Embedded Computing and Signal Processing Research Laboratory. His research interests and activities are in embedded digital systems, digital-signal processing with field-programmable gate arrays, and ultrasonic signal and image processing.