

A Hardware implementation of Winograd Fourier Transform algorithm for Cryptography

¹G.A.Sathishkumar and ²Dr.K.Boopathy bagan

¹ Assistant Professor, Department of Electronics and Communication Engineering, Sri Venkateswara College of Engineering, Sriperumbudur -602108.

²Professor, Department of Electronics, Madras Institute of Technology, Anna University Chrompet Campus, Chennai-600044

Tamil Nadu.

INDIA

sathish@svce.ac.in , kbb@mail.yahoo.com

ABSTRACT

This paper presents a hardware implementation of efficient algorithms that uses the mathematical framework. The framework based on the Winograd's Fourier Transform Algorithm, obtaining a set of formulations that simplify cyclic convolution (CC) computations and CRT. In particularly, this work focuses on the arithmetic complexity of a multiplication and when there is multiplication then the product represents a CC computational operation. The proposed algorithms is compared against existing algorithms developed making use of the FFT and it is shown that the proposed algorithms exhibit an advantage in computational efficiency. This design is most useful when dealing with large integers, and is required by many modern cryptographic systems.

The Winograd Fourier Transform Algorithm (WFTA) is a technique that combines the Rader's index mapping and Winograd's short convolution modules for prime-factors into a composite-N Fourier Transform structure with fewer multipliers ($O(N)$). While theoretically interesting, WFTA's are very complicated and different for every length. It can be implemented on modern processors with few hardware multipliers and hence, is very useful in practice today.

Keywords: Discrete Fourier Transform, Fast Fourier Transform, Winograd's Theorem, Chinese Remainder Theorem.

INTRODUCTION

Many popular crypto-systems like the RSA encryption scheme [12], the Diffie-Hellman (DH) key agreement scheme [13], or the Digital Signature Algorithm (DSA) [14] are based on long integer modular exponentiation. A major difference between the RSA scheme and cryptosystems based on the discrete logarithm problem is the fact that the modulus used in the RSA encryption scheme is the product of two prime numbers. This allows utilizing the Chinese Remainder Theorem (CRT) in order to speed up the private key operations. From a mathematical point of view, the usage of the CRT for RSA decryption is well known. However, for a hardware implementation, special multiplier architecture is necessary to meet the requirements for efficient CRT-based decryption. This paper presents the basic algorithmic and architectural concepts of the WFTA crypto chip, and describes how they were combined to provide optimum performance. The major design goal with the

WFTA was the maximization of performance on several levels, including the implemented hardware algorithms.

In digital signal processing, the design of fast and computationally efficient algorithms has been a major focus of research activity. The objective, in most cases, is the design of algorithms and their respective implementation in a manner that perform the required computations in the least amount of time. In order to achieve this goal, parallel processing has also received a lot attention in the research community [1]. This document is organized as follows. First, mathematical foundations needed for the study of algorithms to compute the DFT and FFT algorithm are summarized. Second, identification is established between Winograd Fourier Transform and the Rader's algorithm. Third, the algorithm development for the basic problem of the multiplication, using the conceptual framework developed in the two previous sections, is explained. The

section also presents several signal flow diagrams that may be implemented in diverse architectures by means of very large scale integration (VLSI) or very high-speed integrated circuits hardware description language (VHDL). Conclusions, contributions, and future development of the present work are then summarized.

1. DISCRETE FOURIER TRANSFORM (DFT)

1.1 DEFINITION

The discrete Fourier transform (DFT) is a powerful reversible mapping transform for discrete data sequences with mathematical properties analogous to those of the Fourier transform and it transforms a function from time domain to frequency domain. For length n input vector x , the DFT is a length n vector X , with n elements:

$$f_j = \sum_{k=0}^{n-1} x_k e^{-j2\pi kn/n} \quad j = 0, \dots, n-1 \quad (1)$$

A simple description of these equations is that the complex numbers F_j represent the amplitude and phase of the different sinusoidal components of the input "signal" x_k . The DFT computes the F_j from the x_k , while the IDFT shows how to compute the x_n as a sum of sinusoidal components $F_j \exp(2\pi i k n / N) / N$ with frequency (k/N) cycles per sample. By writing the equations in this form, we are making extensive use of Euler's formula to express sinusoids in terms of complex exponentials, which are much easier to manipulate. The number of multiplication and addition operations required by the Discrete Fourier Transform (DFT) is of order N^2 as there are N data points to calculate, each of which requires N arithmetic operations. To be exact, the input, if complex, would contain 2 terms and every exponential term would contain 2 terms. So, this would quadruple the computational complexity, thus number of multiplications is $4N^2$. Hence, for real inputs the required number of multiplications will be $2N^2$.

A fast Fourier transform (FFT) [2] is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. FFT's are of great importance to a wide variety of applications, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers. By far the most common FFT is the Cooley-Tukey algorithm. This is a divide and conquer algorithm that recursively breaks down a DFT of any composite size $N = N_1 N_2$

into many smaller DFT's of sizes N_1 and N_2 , along with $O(N)$ multiplications by complex roots of unity traditionally called twiddle factors. The most well-known use of the Cooley-Tukey algorithm is to divide the transform into two pieces of size $N/2$ at each step, and is therefore limited to power-of-two sizes, but any factorization can be used in general (as was known to both Gauss and Cooley/Tukey). These are called the radix-2 and mixed-radix cases, respectively (and other variants such as the split-radix FFT have their own names as well). Although the basic idea is recursive, most traditional implementations rearrange the algorithm to avoid explicit recursion. In addition, because the Cooley-Tukey algorithm breaks the DFT into smaller DFTs, it can be combined arbitrarily with any other algorithm for the DFT.

1.2 Multiplication of large integers

The fastest known algorithms [1, 8, and 10] for the multiplication of very large integers use the polynomial multiplication method. Integers can be treated as the value of a polynomial evaluated specifically at the number base, with the coefficients of the polynomial corresponding to the digits in that base. After polynomial multiplication, a relatively low-complexity carry-propagation step completes the multiplication.

2. WINOGRAD FOURIER TRANSFORM ALGORITHM (WFTA)

The Winograd Fourier Transform Algorithm (WFTA)[7] is a technique that combines the Rader's index mapping and Winograd's short convolution algorithm for prime-factors into a composite- N Fourier Transform structure with fewer multipliers. The Winograd algorithm, factorizes $z^N - 1$ into cyclotomic polynomials—these often have coefficients of 1, 0, or -1 , and therefore require few (if any) multiplications, so Winograd can be used to obtain minimal-multiplication FFTs and is often used to find efficient algorithms for small factors. Indeed, Winograd showed that the DFT can be computed with only $O(N)$ irrational multiplications, leading to a proven achievable lower bound on the number of multiplications for power-of-two sizes; unfortunately, this comes at the cost of many more additions, a tradeoff no longer favorable on modern processors with hardware multipliers. In particular, Winograd also makes use of the

PFA as well as an algorithm by Rader for FFTs of prime sizes.

2.1 RADER'S ALGORITHM

Rader's algorithm (1968)[7,10] is a fast Fourier transform (FFT) algorithm that computes the discrete Fourier transform (DFT) of prime sizes by re-expressing the DFT as a cyclic convolution.

Since Rader's algorithm only depends upon the periodicity of the DFT kernel, it is directly applicable to any other transform (of prime order) with a similar property, such as a number-theoretic transform or the discrete Hartley transform.

The algorithm can be modified to gain a factor of two savings for the case of DFTs of real data, using a slightly modified re-indexing/permutation to obtain two half-size cyclic convolutions of real data; an alternative adaptation for DFTs of real data, using the discrete Hartley transform, was described by Johnson.

Winograd extended Rader's algorithm to include prime-power DFT sizes p^m , and today Rader's algorithm is sometimes described as a special case of Winograd's FFT algorithm, also called the multiplicative Fourier transform algorithm, which applies to an even larger class of sizes.

2.2 Algorithm

The Rader algorithm to compute the DFT,

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad k, n \in \mathbb{Z}_N; \text{ord}(W) = N \quad (2)$$

is defined for prime length N . We first compute the DC component with

$$x(0) = \sum_{n=0}^{N-1} x(n) \quad (3)$$

Because $N=p$ is a prime, it is known that there is primitive element, a generator 'g', that generates all the elements of n and k in the field \mathbb{Z}_N

We substitute n with $g^k \text{ mod } N$ and get the following index transform:

for $k=0, \dots, N-2$. We notice that right side of the above equation is a cyclic convolution i.e.,

$$X[g^k \text{ mod } N] - x(0) = \sum_{n=0}^{N-2} x[g^n \text{ mod } N] W_N^{n+k \text{ mod } (N-1)} \quad (4)$$

$$x[g^0 \text{ mod } N], x[g^1 \text{ mod } N], \dots, x[g^{N-2} \text{ mod } N] \oplus [W_N, W_N^g, W_N^{g^2 \text{ mod } (N-1)}] \quad (5)$$

Thus, an N -point DFT are converted into an $N-1$ point Cyclic convolution.

3. WINOGRAD'S SMALL CONVOLUTION ALGORITHM

This algorithm performs the convolution with minimum number of multiplications and additions and thus the computational complexity of the process is greatly reduced. Cyclic convolution is also known as circular convolution. Let $h = \{h_0, h_1, \dots, h_{n-1}\}$ be the filter coefficients and $x = \{x_0, x_1, \dots, x_{n-1}\}$ be the data sequence. The cyclic convolution can be expressed as

$$s(p) = h \circ_n x = h(p)x(p) \text{ mod } (p^n - 1). \quad (6)$$

The cyclic convolution can be computed as a linear convolution reduced by modulo $p^n - 1$. Alternatively, the cyclic convolution can be computed using CRT with $m(p) = p^n - 1$, which is much simpler. Thus, Winograd's minimum-multiply DFT's are useful only for small N . They are very important for Prime-Factor Algorithms, which generally use Winograd modules to implement the short-length DFT's [10]. The theory and derivation of these algorithms is quite elegant but requires substantial background in number theory and abstract algebra. Fortunately, for the practitioner, the entire short logarithm one is likely to need have already been derived and can simply be looked up without mastering the details of their derivation.

3.1 Algorithm

1. Choose a polynomial $m(p)$ with degree higher than the degree of $h(p)x(p)$ and factor it into $k+1$ relatively prime polynomials with real coefficients, i.e.,

$$m(p) = m^{(0)}(p)m^{(1)}(p)\dots m^{(k)}(p). \quad (7)$$

2. Let $M^{(i)}(p) = m(p) / m^{(i)}(p)$ and use the Chinese Remainder Theorem (CRT) algorithm to get $N^{(i)}(p)$.

3. Compute

$$h^{(i)}(p) = h(p) \bmod m^{(i)}(p), \quad (8)$$

$$x^{(i)}(p) = x(p) \bmod m^{(i)}(p), \quad (9)$$

for $i=0,1,2,\dots,k$.

$$4. \text{ Compute } s^{(i)}(p) = h^{(i)}(p)x^{(i)}(p) \bmod m^{(i)}(p), \quad (10)$$

for $i=0,1,\dots,k$.

5. Compute $s(p)$ using the equation:

$$s(p) = \sum_{i=0}^k s^{(i)}(p)N^{(i)}(p)M^{(i)}(p) \bmod m(p). \quad (11)$$

The computational complexity in case of WFTA is of the order of N , $O(N)$. It has been found that the number of multipliers required by WFTA is always less than $2N$, which drastically reduces the hardware needed for implementing a DFT block.

4. REALIZATION OF WFTA IN VERILOG HDL

The behavioral simulation and synthesis of WFTA for $N=2$ and 5 can be viewed in the following descriptions. We focus on the Verilog HDL [6] used for our simulation and synthesis. It is shown in Fig.1, 2, 3, 4,5 and 6.

4.1.1 SYNTHESIS RESULTS

Synthesis of WFTA is being implemented using XILINX ISE 9.1i tool. After design entry and optional simulation, we run synthesis. During this step, VHDL, Verilog, or mixed language designs become net list files that are accepted as input to the implementation step.

4.1.2 IMPLEMENTATION

After synthesis, we run design implementation, which converts the logical design into a physical file format that can be downloaded to the selected target device. From

Project Navigator, we can run the implementation process in one-step, or we can run each of the implementation processes separately.

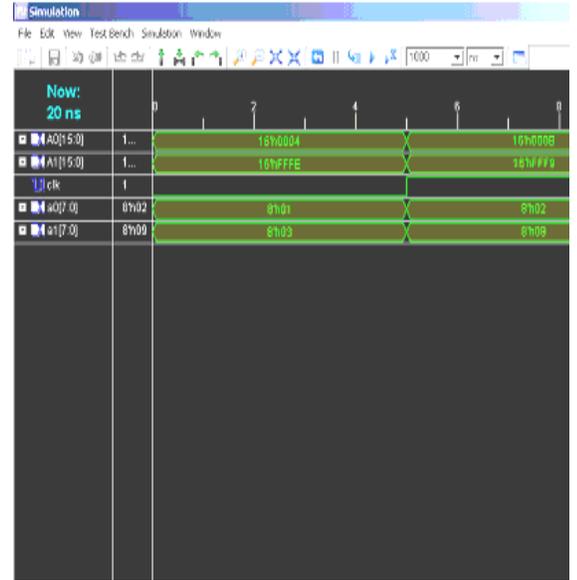


Figure 1 Simulation result for N=2

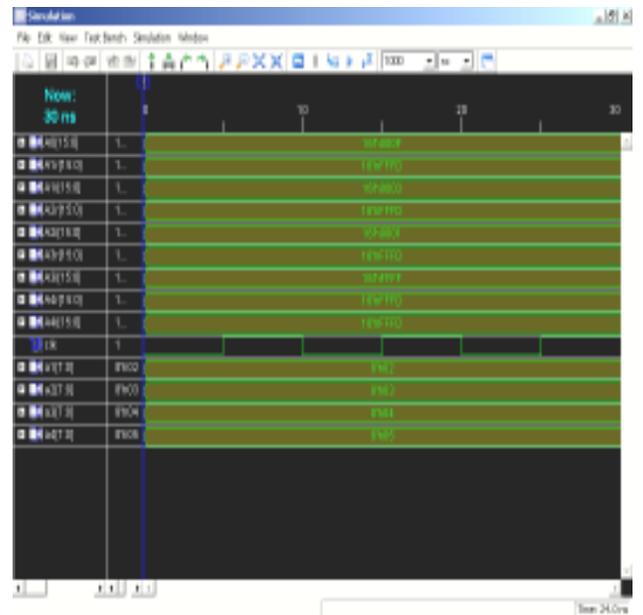


Figure 2 Simulation result for N=5

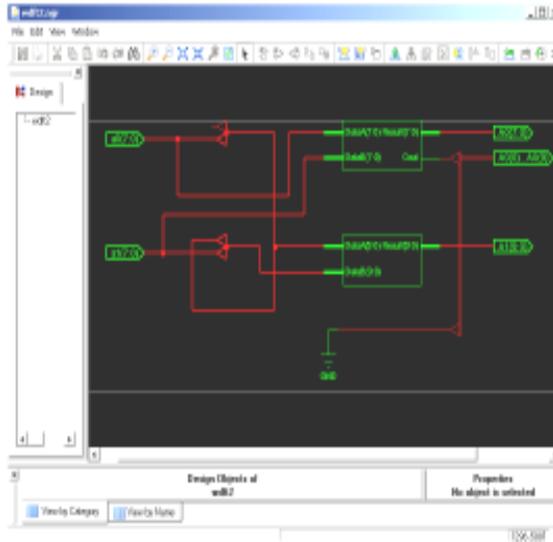


Figure3 Schematic of DFT N=2

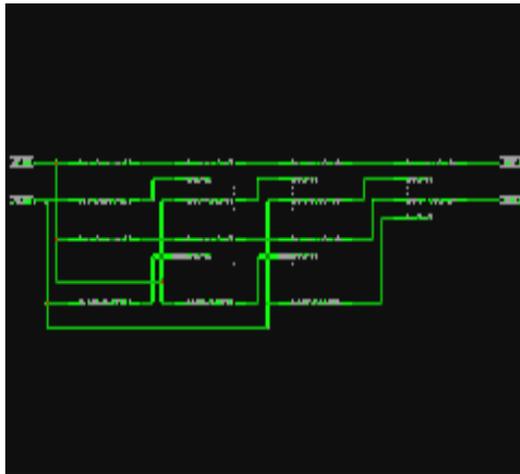


Figure 4 Schematic of WDFT N=2

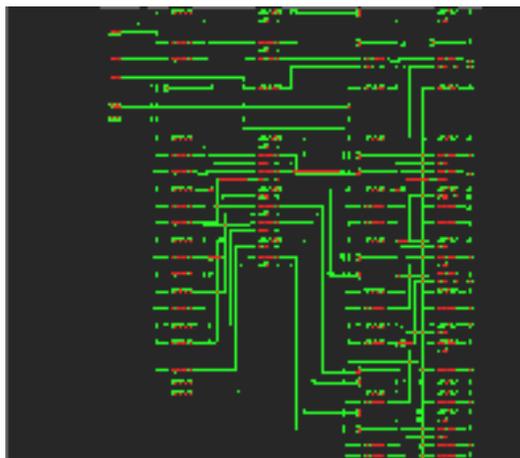


Figure 5 Schematic of DFT N=5

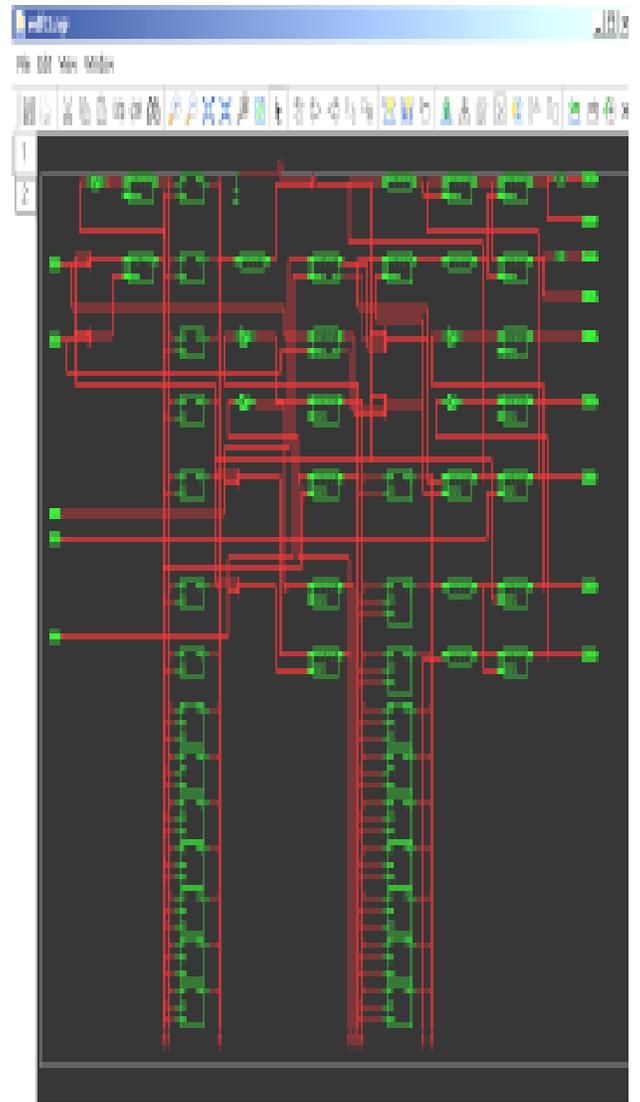


Figure 6 Schematic of WDFT N=5

[7].S. Winograd, "On computing the discrete Fourier transform," *Math. Comp.*, vol. 32, no. 141, pp. 175–199, Jan. (1978).

[8]. J. McClellan and C. Rader, Number Theory in Digital Signal Processing. *Englewood Cliffs, NJ*: Prentice Hall, pp 79-85 (1979).

[9].S. Winograd, Arithmetic complexity of computations (*Society for Industrial and Applied Mathematics*, (1980).

[10]. M. Heideman, Multiplicative complexity, convolution, and the DFT *Springer Verlag, New York* (1988)

[11]. J. Cooley, Some applications of computational complexity Theory to Digital Signal Processing. 1981 *Joint Automatic Contr. Conf. University of Virginia*, June 17-19 (1981).

[12]. R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the Association for Computing Machinery*, 21(2) pp. 120–126, February (1978).

[13]. W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6) pp. 644–654, November (1976).

[14]. National Institute of Standards and Technology (NIST). FIPS Publication 186: Digital Signature Standard. *National Institute for Standards and Technology*, Gaithersburg, MD, USA, May (1994).



Sathishkumar.G.A obtained his M.E from PSG college of Technology, Coimbatore, India. He is currently perusing PhD from Anna University, Chennai and Faculty member in the Electronics and Communication Dept of Sri Venakesateswara College of Engineering, Sriperumbudur.His research interest is VLSI Signal processing Algorithms, Image Processing and Network Security.

Dr.K.Boopathy Bagan completed his doctoral degree from IIT Madras. He is presently working as professor, ECE dept, in Anna

University, MIT Chrompet campus, Chennai. His areas of interest include VLSI, Image processing, Signal processing and network Security.