American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-12, Issue-4, pp-151-157
www.ajer.org
Research Paper Open Access

# Efficient and Flexible Number Theoretic Transform for Lattice-Based Cryptography

Zaman, Chongkon Haruna
*Department of Electrical and Computer Engineering*
*Morgan State University, Baltimore, Maryland, USA.*

***Abstract***
*A giant quantum computer will break traditional public key cryptography. Lattice-based cryptography appears as an alternative to protect communications in the era of quantum computers. The Number Theory Transform (NTT) is an attractive technique for performing polynomial multiplication efficiently.Low power consumption is critical for many systems, such as battery-powered gadgets and Internet of Things (IoT) devices. However, NTTs that are both efficient and low power needs more research attention. The proposed architecture uses only n log (n) clock cycles and can be implemented with low-cost single-port RAM.*

Date of Submission: 12-04-2023 Date of acceptance: 27-04-2023

## I. Introduction

The cornerstone of creating secure communication between several parties is public-key cryptography (PKC). Quantum computers are posing a threat to all communication systems. Post-quantum cryptography (PQC) is a viable option for protecting communication channels from a quantum threat. The NTT is a finite-field variant of the Fast Fourier Transform (FTT). It translates a pair of polynomials into the spectral domain, which can be multiplied pointwise. The NTT avoids the lattice-based cryptography performance bottleneck.

A quantum computer will be able to break traditional public-key cryptography. In the age of quantum computers, lattice-based cryptography has arisen as an alternative to protect communications. For example, the Number Theoretic Transform (NTT) is a popular method for quickly multiplying polynomials. The National Security Agency (NSA) and the National Institute of Standards and Technology (NIST) standardize post-quantum cryptography. QPC is a set of cryptographic methods resistant to known quantum computer attacks.

Based on challenging problems in lattices, lattice-based encryption provides a fair balance of security and efficiency. Therefore, optimizing the NTT to make lattice-based post-quantum cryptography possible and practicable is necessary. The NTT architecture is adaptable, with multiple NTT configuration parameters.
The Number Theoretic Transform (NTT) is an effective way to simplify the multiplication of two big polynomials. The NTT is a Fast Fourier Transform (FFT) finite-field variant.

**NTT**

$$\mathrm{X(k)} = \sum_{i=0}^{N-1} x(i) w^{ik} \quad (mod\ M) \qquad k = 0,\ 1,\ \ldots N-1 \qquad \ldots 1$$

**Inverse NTT**

$$\mathrm{Y(k)} = \sum_{k=0}^{N-1} x(k) w^{-ik} \quad (mod\ M^{-1}) \qquad k = 0,\ 1,\ \ldots N-1 \qquad \ldots 2$$

This work offers the first NTT ASIC design for lattice-based cryptography, which is low-power, quick, and secure. For the forward and inverse NTT, our suggested design uses only n log (n) clock cycles and may be implemented with inexpensive single-port RAM.

## II. Literature review

The NTT has been used in various signal-processing applications, including FIR filters, image filtering, and homomorphic image enhancement. In addition, the NTT is used in lattice-based cryptography to modify more significant vectors and requires an additional reduction module (mod (xn + 1)).

## III. Methodology

This section shows the methods employed in achieving the aim and objectives of this paper

### 3.1 Post-Quantum Cryptography (PQC)

As shown in Figure 1, the PQC algorithms are generally implemented using Hash-Based Signature Algorithms, Code-Based Cryptography, Multivariate Cryptography Protocols, or Lattice-Based Cryptography.The PQC algorithms will be discussed briefly in Figure 1.

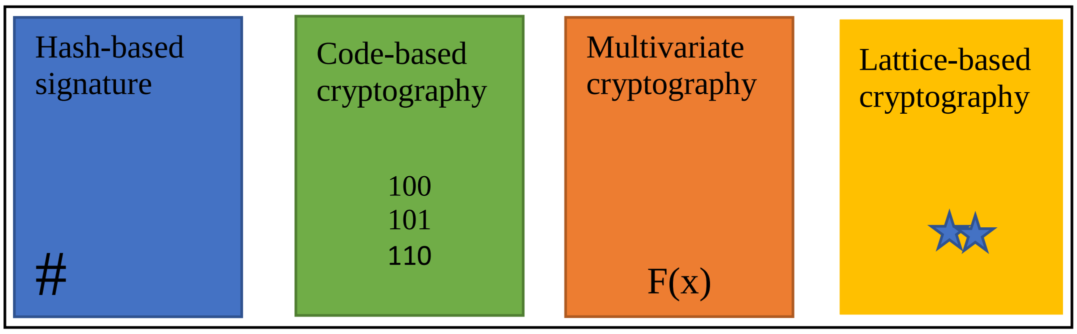


Figure 1: Implementation methods of four fundamental quantum secure algorithms

#### *3.1.1 A hash-based signature*

A hash-based signature method starts with a one-time signature (OTS), meaning each key pair only needs to be used once to sign a message.However, Merkle proposed using a binary hash tree, later dubbed the Merkle tree, to develop a many-times signature technique.In a Merkle tree, the leaves are the hash values of OTS public keys.In a Merkle signature scheme (MSS), the root node of the Merkle tree becomes a public key, and the set of all OTS hidden keys becomes the secret key.The remote keys for hash-based OTS are random bit strings.

#### *3.1.2 Code-based cryptography*

Code-based cryptography is an emerging competitor for diversifying today's public-key cryptosystems, most of which rely on the complexity of either factorization or the discrete logarithm issue.Unlike public-key algorithms, code-based encryption is based on the NP-hard problem of decoding unknown error-correcting codes.There are two simple code-based cryptography systems named after their creators.

Compared to traditional cryptosystems like RSA, both have the drawback of having enormous vital lengths, making their implementation impractical on embedded devices with limited resources.So instead, the input message is transformed into a code word for plain text encryption by adding random errors or embedding a message.

#### *3.1.3 Multivariate cryptography*

Multivariate cryptography techniques are built on the issue of solving non-linear equation structures over finite fields.One of the unique situations is Patarin's Secret Fields, which generalizes a suggestion by Matsumoto and Imai.All Multivariate Public-Key Cryptosystems (MPKC) has the same fundamental architecture since they all rely on multivariate polynomials over a finite field.However, most polynomial equations are of degree two, resulting in multivariate quadratic polynomials still regarded as NP-hard.Shor's algorithm cannot be solved more readily using Shor's techniques than with a traditional computer since it does not rely on any of the complex problems that Shor's approaches can tackle.

#### *3.1.4 Lattice-based cryptographic*

The cryptographic builders used in most lattice-based cryptographic algorithms are relatively time-efficient and straightforward while offering security proofs based on worst-case hardness. Lattice-based cryptography is one of a few algorithmsthat hold promise as prospective contenders for post-quantum cryptography.Specific encryption algorithms used to support these protocols, such as RSA, Diffie-Hellman, and the elliptic curve, are based on difficult-to-solve mathematical issues and are classified as asymmetric cryptographic primitives.Because quantum computers using Shor's factorization quantum method can solve present asymmetric cryptography primitives quickly.

## 3.2 General Descriptionof NTT and Inverse NTT

The figure below shows the hardware description of NTT and inverse NTT. The NTT transforms the n coefficients of a polynomial ***x***into the spectral domain. The NTT$^{-1}$ transforms back into the typicalenvironment the coefficients of the result vector ^***x*** to keep the NTT flexible, the pre-calculated values required in the case of the NTT and NTT$^{-1}$

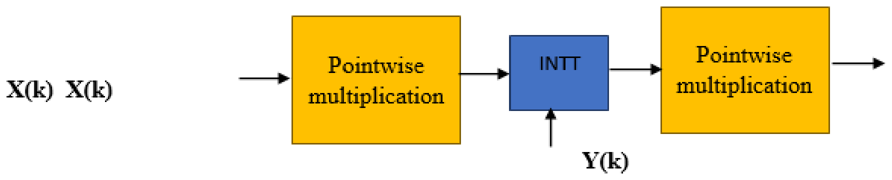


Figure2: NTT and NTT$^{-1}$ input/output description.

The Number Theoretic Function (NTT) is a finite field extension of the standard Discrete Fourier Transform (DFT). It allows one to conduct rapid convolutions on integer sequences with no round-off errors, guaranteeing success with much effort. Convolutions are beneficial for multiplying huge numbers or long polynomials. The NTT is a ring-based transformation that allows us to perform more efficient multiplications of the elements in those rings. A complex-valued function defined for all positive integers is an arithmetic or number-theoretic operation. The NTT is a non-trivial number theory representing a finite field of polynomials using the Fast Fourier Transform (FTT) method rather than floating-point numbers or complex arithmetic.

**NTT**

$$\mathrm{X(k)} = \sum_{i=0}^{N-1} x(i) w^{ik} \quad (mod\ M) \qquad k = 0,\ 1,\ \ldots N-1 \qquad \ldots 1$$

**Inverse NTT**

$$\mathrm{Y(k)} = \sum_{k=0}^{N-1} x(k) w^{-ik} \quad (mod\ M^{-1}) \qquad k = 0,\ 1,\ \ldots N-1 \qquad \ldots 2$$

Where*w* = Nth primitive root of unity in ***Z***and ***N*** = transform length, and the primitive element

**Ring: -** is a set of real numbers (ℝ) with two operands (+ and *) and satisfying the following properties:

- ℝ is an abelian group under +.
- Associativity of x – For every a, b, c ∈ℝ

a * (b * c) = (a * b) x*c

- Distributive Properties For every a, b, c ∈ℝthe following identities hold:

  a * (b + c) = (a * b) + (a * c) and (b + c) * a = b *a + c * a.

**Field: -**is a set of elements F with two operands (+ and *)and satisfying the following properties:

- F is an abelian group under + and
- F − {0} (the set F without the additive identity 0) is an abelian group under *.

**Generators**

Unit g ∈ℤn∗ is called a generator *or* primitive root of ℤn∗ if for every a∈ℤn∗ therefore, $g^k$=w for some integer k. In other words, if we start with g, and keep multiplying by g, eventually, we see every element.

**Modular Arithmetic**

Assuming n is a positive integer. Then we can denote the set [0, . . ., n−1] by ℤn. If we consider two integers, say **a** and **b** to be the same if **a** and **b** differ by a multiple of n, write this as**a** = **b** (**mod n**) and say that x and y are *congruent* modulo n. We may omit (**mod**n) when it is clear from the context. Every integer **a** is congruent to some y in Zn. When we add or subtract multiples of n from an integer **a** to reach some **b** ∈ℤn, we say it reduces **a** modulo n, and **b** is the *residue*.

**Primitive *n*th roots of unity (w)**are roots of conformity with the nth multiplicative order. For example, they are the roots of the nth cyclotomic polynomial and play an essential role in many disciplines of number theory, particularly algebraic number theory.

**Roots of Unity**

The primitive root n is an integer g such that every integer relative to n is congruent to the power of g **mod**n.

➢ **Procedure in calculating/finding NTT:**

i. Have a vector sequence of an nth non-negative number
ii. Choose a working modulus, say M.
iii. Select an integer k≥1 and define N=kn+1. A value of k that will make N a prime
iv. Define w = $g^k$ mod M

Even though lattice-based cryptography has been around for almost two decades, it is still considered a new approach to cryptographic design. This is because onlythe NTT component uses arithmetic operations, especially in systems where floating-point units are more significant and slower than integer arithmetic units.

## 3.3 OPTIMIZING THE USE OF POWER

When many components in an embedded system enable low-power modes, dynamic power consumption can be reduced dramatically by careful resource management. For example, an operating system may detect load or receive clear signals from applications to convert the machine to a lower-power and, thus, lower-performance state. However, any benefits from doing so must be weighed against the continual power draw in the background.This paper uses two strategies to lower the NTT's dynamic power consumption: Clock Gating and Silicon on Insulator (SOI).

### *3.3.1 Clock Gating*

The clock signal typically hasmuch fan-out, which means it uses much dynamic power. For synchronous circuits, clock gating is an effective means of lowering power usage. Each clock cycle does not always require using all functional units in a design.

Clock gating's goal is to turn on the clock signal only for those units that must be operational. The dynamic power consumption of registers is zero when idle, leaving only static power consumption (due to leakage currents) to worry about. Figure 3 depicts the concept of clock gating. The registers with the same enabled signal are grouped during the clock gating optimization. After that, the logic utilized to create the enable signal is reversed.

### *3.3.2 Silicon on Insulator (SOI)*

Silicon on Insulator (SOI) is a form of insulator that has been utilized in CMOS circuits. It is made up of two types of insulators. One is SiO2, and the other is sapphire, both of which have the advantage of lowering capacitance between the source and the drain. Another benefit is that the diffusion capacitance is reduced, resulting in lower subthreshold leakage in circuits and, thus, additional power savings.Two types of SOI techniques exist PDSOI (partially depleted) and FDSOI (fully finished). Although FDSOI aids in decreasing tunneling currents in CMOS, the PDSOI technique is generally adopted due to technological constraints.

## IV. Results and Discussions

The paper's results are given in this subsection and the discussion.

Post-quantum cryptography based on lattices is still a relatively new field of study. The Zed board, which has a Xilinx Zynq-7000, was used to implement our FPGA hardware architecture. A parallel NTT architecture, some are only appropriate for high-throughput applications with small n numbers. The memory utilization in [18] was significantly lower because the authors recommend calculating the Twiddle factors during run-time. The number of slices is between 200 and 300 for n = 256, n = 512, and n = 1024.

## 4.1 Power measurement

The ASIC design was created using UMC 65 nm technology. A low-level library with a high threshold voltage was used to achieve minimal leakage currents. Because the same architecture is used for all configurations, the static power consumption is independent of the parameter set.

Because the NTT is not always active, clock gating is predicted to save significant electricity. The amount of power saved is determined by the cryptographic strategy used and the time spent on specific tasks. Only a few extra gates are necessary to reduce total power usage drastically.In addition to the design clock gating, the number of cells in the operand isolation approach was raised by 145. However, the switching activity of the system was significantly reduced due to the blocking of unneeded signal propagation. As a result, thetotal power usage reduction was obtained to be 0.03W.

## 4.2 Timing Result

When one clock cycle is assumed, and a single port RAM is employed, the number of clock cycles required for the NTT and NTT$^{-1}$equals n log (n) (plus eight cycles of latency). In contrast to earlier research, post-processing requires no additional clock cycles.

## V. Conclusion

This paper presents a low-power NTT ASIC design that is fast, versatile, and secure. The design's adaptability allows the use of a variety of cryptographic techniques. Rescheduling activities and employing effective modulo-reductionmethods can accelerate the design process.*The* Number Theory Transform (NTT) is an attractive technique for performing polynomial multiplication efficiently. Low power consumption is critical for many systems, such as battery-powered gadgets and Internet of Things (IoT) devices.

## References

[1]. Paludo, Rogério, and Leonel Sousa. "Number Theoretic Transform Architecture suitable to Lattice-based Fully-Homomorphic Encryption." 2021 IEEE 32nd International Conference on Application-specific Systems, Architectures and Processors (ASAP). IEEE, 2021.

[2]. Mert, Ahmet Can, et al. "An extensive study of flexible design methods for the number theoretic transform." IEEE Transactions on Computers (2020).

[3]. Fritzmann, Tim, and Johanna Sepúlveda. "Efficient and flexible low-power NTT for lattice-based cryptography." 2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST). IEEE, 2019.

[4]. Longa, Patrick, and Michael Naehrig. "Speeding up the number theoretic transform for faster ideal lattice-based cryptography." International Conference on Cryptology and Network Security. Springer, Cham, 2016.

[5]. Karabulut, Emre, and Aydin Aysu. "RANT: A RISC-V architecture extension for the number theoretic transform." 2020 30th International Conference on Field-Programmable Logic and Applications (FPL). IEEE, 2020.

[6]. Pessl, Peter, and Robert Primas. "More practical single-trace attacks on the number theoretic transform." International Conference on Cryptology and Information Security in Latin America. Springer, Cham, 2019.

[7]. Nejatollahi, Hamid, Rosario Cammarota, and Nikil Dutt. "Flexible ntt accelerators for rlwe lattice-based cryptography." 2019 IEEE 37th International Conference on Computer Design (ICCD). IEEE, 2019.

[8]. Özerk, Özgün, et al. "Efficient number theoretic transform implementation on GPU for homomorphic encryption." The Journal of Supercomputing 78.2 (2022): 2840-2872.

[9]. Banerjee, Utsav, Tenzin S. Ukyab, and Anantha P. Chandrakasan. "Sapphire: A configurable crypto-processor for post-quantum lattice-based protocols." arXiv preprint arXiv:1910.07557 (2019).

[10]. Zhang, Neng, et al. "Highly efficient architecture of NewHope-NIST on FPGA using low-complexity NTT/INTT." IACR Transactions on Cryptographic Hardware and Embedded Systems (2020): 49-72.

[11]. Mert, Ahmet Can, ErdinçÖztürk, and ErkaySavaş. "Design and implement encryption/decryption architectures for bfv homomorphic encryption scheme." IEEE Transactions on Very Large-Scale Integration (VLSI) Systems 28.2 (2019): 353-362.

[12]. Fritzmann, Tim, et al. "Masked accelerators and instruction set extensions for post-quantum cryptography." IACR Transactions on Cryptographic Hardware and Embedded Systems 2022.1 (2021): 414-460.

[13]. Mert, Ahmet Can, et al. "An extensive study of flexible design methods for the number theoretic transform." IEEE Transactions on Computers (2020).

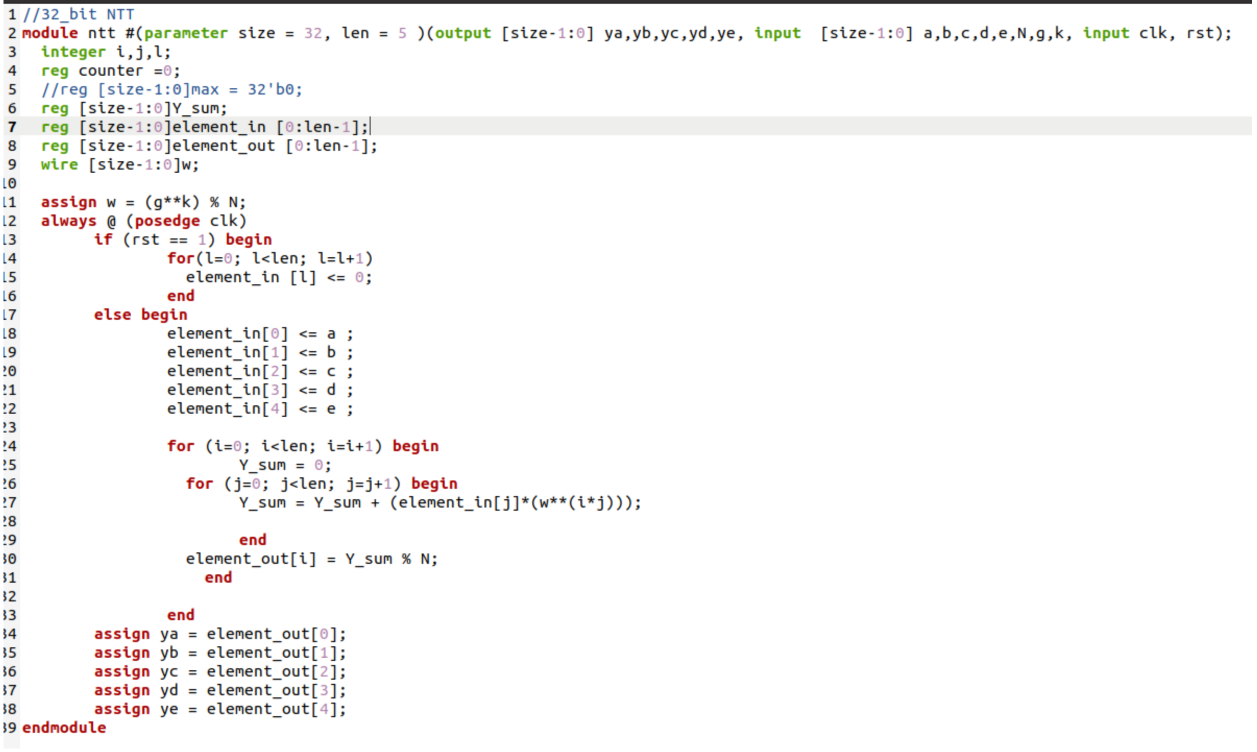
[14]. Mert, Ahmet Can, et al. "A flexible and scalable NTT hardware: Applications from homomorphically encrypted deep learning to post-quantum cryptography." 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2020.

[15]. Reparaz, Oscar, et al. "Masking ring-LWE." Journal of Cryptographic Engineering 6.2 (2016): 139-153.

[16]. Bache, Florian, et al. "High-speed masking for polynomial comparison in lattice-based kems." IACR Transactions on Cryptographic Hardware and Embedded Systems (2020): 483-507.

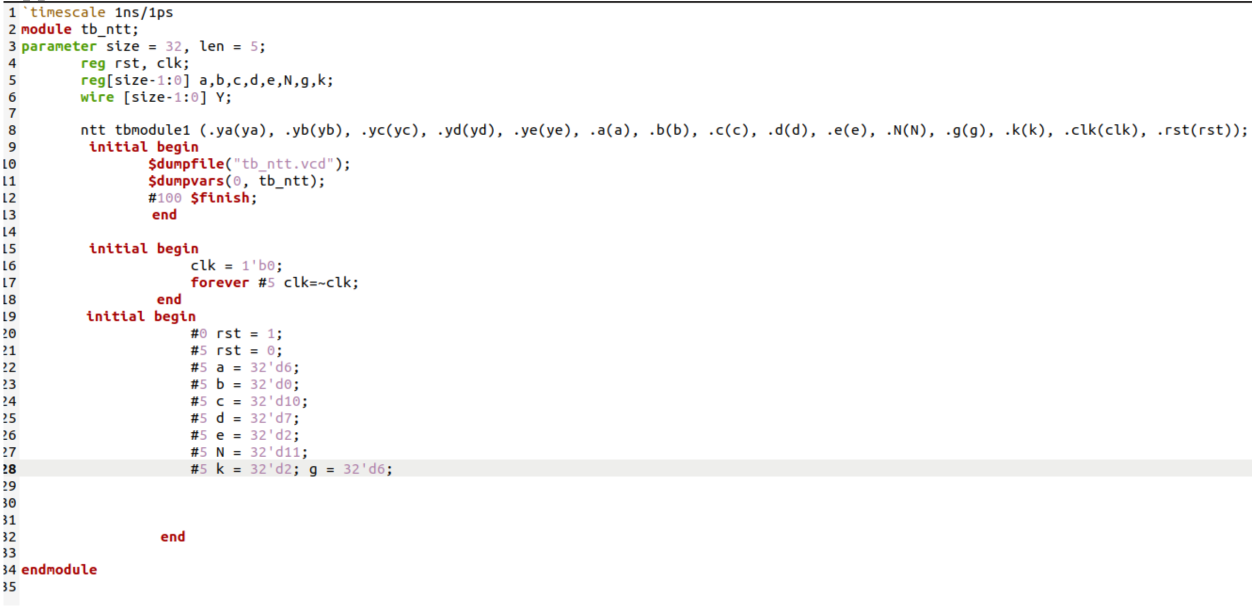
[17]. Aysu, Aydin, Cameron Patterson, and Patrick Schaumont. "Low-cost and area-efficient FPGA implementations of lattice-based cryptography." 2013 IEEE international symposium on hardware-oriented security and trust (HOST). IEEE, 2013.

**Appendix A:**Verilog source code



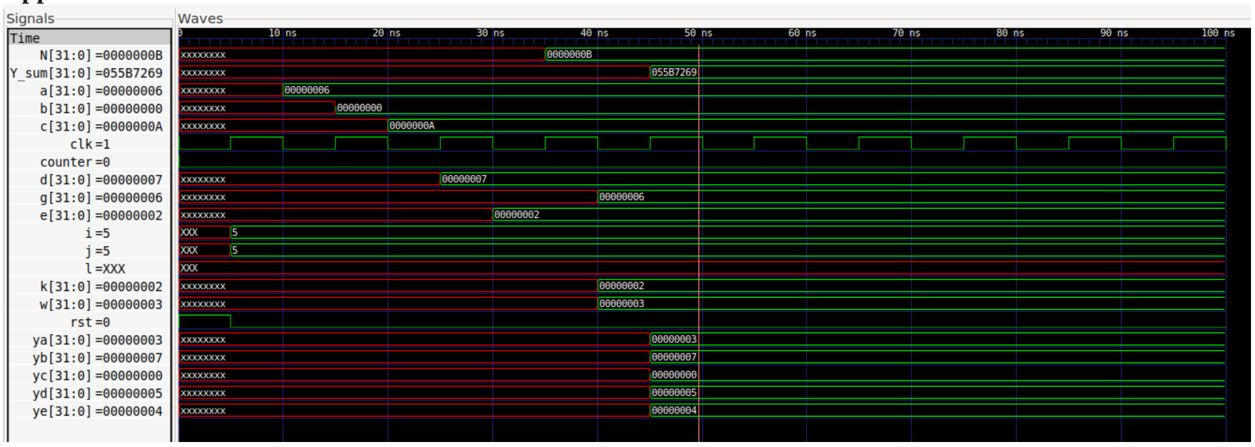
```
//32_bit NTT
module ntt #(parameter size = 32, len = 5 )(output [size-1:0] ya,yb,yc,yd,ye, input  [size-1:0] a,b,c,d,e,N,g,k, input clk, rst);
  integer i,j,l;
  reg counter =0;
  //reg [size-1:0]max = 32'b0;
  reg [size-1:0]Y_sum;
  reg [size-1:0]element_in [0:len-1];
  reg [size-1:0]element_out [0:len-1];
  wire [size-1:0]w;

  assign w = (g**k) % N;
  always @ (posedge clk)
        if (rst == 1) begin
                for(l=0; l<len; l=l+1)
                  element_in [l] <= 0;
                end
        else begin
                element_in[0] <= a ;
                element_in[1] <= b ;
                element_in[2] <= c ;
                element_in[3] <= d ;
                element_in[4] <= e ;

                for (i=0; i<len; i=i+1) begin
                        Y_sum = 0;
                  for (j=0; j<len; j=j+1) begin
                        Y_sum = Y_sum + (element_in[j]*(w**(i*j)));

                        end
                  element_out[i] = Y_sum % N;
                    end

                end
        assign ya = element_out[0];
        assign yb = element_out[1];
        assign yc = element_out[2];
        assign yd = element_out[3];
        assign ye = element_out[4];
endmodule
```

**Appendix B:**Testbench



```
`timescale 1ns/1ps
module tb_ntt;
parameter size = 32, len = 5;
        reg rst, clk;
        reg[size-1:0] a,b,c,d,e,N,g,k;
        wire [size-1:0] Y;

        ntt tbmodule1 (.ya(ya), .yb(yb), .yc(yc), .yd(yd), .ye(ye), .a(a), .b(b), .c(c), .d(d), .e(e), .N(N), .g(g), .k(k), .clk(clk), .rst(rst));
         initial begin
                $dumpfile("tb_ntt.vcd");
                $dumpvars(0, tb_ntt);
                #100 $finish;
                 end

         initial begin
                     clk = 1'b0;
                     forever #5 clk=~clk;
                 end
        initial begin
                     #0 rst = 1;
                     #5 rst = 0;
                     #5 a = 32'd6;
                     #5 b = 32'd0;
                     #5 c = 32'd10;
                     #5 d = 32'd7;
                     #5 e = 32'd2;
                     #5 N = 32'd11;
                     #5 k = 32'd2; g = 32'd6;



                 end

endmodule
```

**Appendix C:**Gtk wave



```
Signals
Time
     N[31:0] =0000000B
Y_sum[31:0] =055B7269
     a[31:0] =00000006
     b[31:0] =00000000
     c[31:0] =0000000A
         clk =1
     counter =0
     d[31:0] =00000007
     g[31:0] =00000006
     e[31:0] =00000002
           i =5
           j =5
           l =XXX
     k[31:0] =00000002
     w[31:0] =00000003
         rst =0
    ya[31:0] =00000003
    yb[31:0] =00000007
    yc[31:0] =00000000
    yd[31:0] =00000005
    ye[31:0] =00000004
```