International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Index Copernicus Value (2015): 62.86 | Impact Factor (2015): 3.791

VLSI Architecture for an Area Efficient Multiplication Using Number Theoretic Transform On Graphic Cards

C. Sai Punitha[1], D. Jessintha[2]

[1]Post Graduate Student, Department of Electronics and Communication Engineering, Easwari Engineering College, Chennai, India
punithacs24[at]gmail.com
[2]Associate professor, Department of Electronics and Communication Engineering, Easwari Engineering College, Chennai, India
jessintha.kumar[at]gmail.com

Abstract: ***The Number Theoretic Transform (NTT) introduced as a generalization of the Discrete Fourier Transform over residue-class rings of unity which have many applications in computer arithmetic and which allows the implementation of Digital Signal Processing operations with better efficiency and accuracy than Fast Fourier Transform without round off errors. The incorporation of double modulus entity reduces the computation time and the buffer reduction technique that is tailored for the special moduli required by the NTT. In this work, a multiplication algorithm based on double modulus NTT has been developed and the deployment of double moduli enlarges the permitted NTT sample size and thus improves the transform efficiency over large integer multiplication and an area efficient multiplication using Number Theoretic Transform (NTT) architecture is designed and verified using Xilinx tool and simulation results reveals the better performance of multipliered architecture rather than relaying on the multiplierless architecture and this can be employed in Graphics Processing Unit especially in Nvidia graphic cards which can be used in video filtering and High Definition image display for an area efficient memory operation.***

Keywords: Number Theoretic Transform, Fast Fourier Transform, Double Moduli, Nvidia

1. Introduction

With the rapid advances in very large scale integration, a growing number of complex digital signal processing applications are becoming economically feasible. In most cases the bulk of processing work load appears to consist of digital filter computation. Future progress in digital signal processing, either towards high speed, real time operation and increased sophistication thus largely depends on increased efficiency in digital filtering computation. This can be achieved not only by implementing improved filter circuits but also by using better computation.

1.1 Multiplication Algorithms

In Cooley–Turkey algorithm the Radix-2 decimation-in-time Fast Fourier Transform is the easiest form. The Fast Fourier Transform is the mostly used in digital signal processing algorithms. Discrete Fourier Transform (DFT) is computing by the FFT. DFT is used to convert a time domain signal into its frequency spectrum domain and their static and dynamic power of 0.033w and 0.609w could be achieved [1] and [2] the same algorithm is implemented on a Spartan-6 FPGA kit using Xilinx ISE 14.2. The implementation is compared with various pre-installed IP-core modules of Xilinx for complex input of various sample sizes and has not only a lower computation cost, less memory and resources utilization, but also provides better absolute error compared to the IP-core modules and [3] shows the implementation of the FFT transform in graphic cards by using the Open Computing language (OpenCL) the GPU is more promising for large number of FFT's of large sizes. The results also confirm that the FPGA based implementation is faster than the built-in IP-core modules of Xilinx and Sparse Fourier Transform demonstrated in [4] achieves speedups of up to three times a single-threaded SFFT while a GPU-based version achieves up to ten times speedup. For large scale cases, the GPU-based SFFT also shows its considerable advantages, which is about 40 times speedup compared to the latest out-of-card FFT implementations. In [5] a new Mersenne number transform (NMNT) can be easily parameterized and implementation in an XC2V4000 FPGA chip has shown that this architecture can work at a frequency of up to 114MHz with a throughput rate of 228MS/s.

1.2 Related Works

Based on the qualitative and quantitative analyses are provided to show the effectiveness of proposed NTT architectures[6] sates that the NTT architecture provides a good performance for long integer multiplication and [7] provides a Fractional Number Theoretic Transform can be efficiently applied to secure signal image processing and the testing speed of FFT/IFFT on Geforce 680m their precision errors been compared [8]and [9] incorporation of double modulus technique in NTT achieves a million bit integer multiplication for cryptographic applications and [10] incorporation of NTT on kepler architecture of graphic cards reveals the NTT can be significantly used in larger bit integer multiplication which is the first implementation of Number Theoretic Transform on graphic cards so far and the thread level parallelism has been implemented.

2. Number Theoretic Transform

A mathematical framework for Number theoretic transform is based on the theory of congruences modulo M, which belongs to the general area of what is often

Licensed Under Creative Commons Attribution CC BY

called "number theory." Number theory is very old, going back several thousand years. In recent years, there has been increasing interest in the practical applications of various parts of number theory, including the theory of residue number systems. There has been some work on the use of these number systems in general purpose although this line of investigation has not yielded many practical results due to the difficulty of determining the sign of numbers expressed in residue number system notation. More promising results have been obtained in applications where sign detection is not required, such as number theoretic transforms can be used to compute convolutions approximately three times as fast as the FFT implementation for the same convolution. However, their main drawback is a rigid relationship between word length and obtainable transform length, as well as a limited choice of possible word lengths. This last point is especially significant for NTT's, and may result in a waste of computing power when the permissible word lengths do not correspond to the dynamic range required for the convolution.

2.1 Numerical Expressions Of Double Modulus Number Theoretic Transform

Fast implementation of convolution, and discrete Fourier transform (DFT) computations, are frequent problems in signal and image processing. In practice these operations are most often implemented using fast Fourier transform algorithms. NTTs can, in some instances, outperform FFT-based systems. In addition, it is also possible to use a rectangular transform, like the Walsh–Hadamard or the arithmetic Fourier transform, to get an approximation of the DFT or convolution and the NTT is defined over a finite group, as the transform pair,

$$X(k) = \sum_{n=0}^{N-1} x(n)\alpha^{nk} \bmod p,\ k=0,1,2....N-1$$

$$x(n) = N^{-1} \sum_{k=0}^{N-1} X(K)\ \alpha^{-nk} \bmod p, n=0,1,2...N-1 \quad (1)$$

Number Theoretic Transforms (NTT), made their appearance and with an aim of replacing the FFT as a tool of signal processing. These transformations based on rules of the number theory. However, in spite of the advantages of this transformation (exactness of the results), the NTT did not know a success similar to that of the FFT because of the interpretation problem of the field that it create, which is not the case for the frequency field created by the FFT. This transformation bases on the modulo-M properties. Hence, the good choice of M may gives properties similar to those of the Discrete Fourier Transform (DFT).Number Theoretic Transform (NTT) is a specialization of Fast Fourier Transformation (FFT), which complex functions are transformed by modulo operation makes the transform simple. The NTT is a generalization of the classic DFT to finite fields. With a lot of work, it basically lets one perform fast convolutions on integer sequences without any round-off errors, guaranteed. Convolutions are useful for multiplying large numbers or long polynomials and the NTT is asymptotically faster than other methods like Karatsuba multiplication.

2.2 Algorithm for Proposed Double Modulus Number Theoretic Transform

Input Vector: X = $(a_1, a_2, a_3 \ldots . a_n)$
1.Minimum Working Module :M (1≤n≤M) in range [0, M]
2.Select K : N=Kn+1=M, N ≥ M
3.Find the multiplication generator $Z_M, \omega = g^K$
α^((N-1)/K) ≡ 1 mod M
3.Generate Z_M $\alpha^{\frac{N-1}{K}}$= 1mod M, $\alpha^{\frac{N-1}{K}}$ =1 mod M
$\omega = g^k, \omega = \alpha^k$ mod M(primitive n^{th} root of unity)
4.Butterfly Structure
for (0≤ j < K)
if (P≤ M + 1)
P=X+Y
m=X+(~Y+1)
sx=(X×CX)+(Y×CY)
sy=(X×CY)+(~Y+CX+1)
end
end for.

2.3 Procedure for the Double Modulus Number Theoretic Transform

a) Consider the input vector is a sequence of n non-negative integers.
b) Choose a minimum working modulus M such that 1 ≤ n< M and every input value is in the range [0, M]
c) Select some integer k ≥ 1 and define N=kn+1 as the working modulus. We require N≥M, and N to be a prime number. Dirichlet's theorem guarantees that for any n and M, there exists some choice of k to make N be prime.
d) N is prime, the multiplicative group of Z_N of size α(N)=N-1=kn. the group must have atleast one generator g, which is also a primitive $(N-1)^{th}$ root of unity
e) $\omega = g^k \bmod N$.then $\omega^n = g^{kn} = g^{N-1} = g^{\alpha(N)} = 1 \bmod N$ due to Euler's theorem. g is a generator
$\omega^i = g^{ik} \neq 1$ for $1 \leq i \leq n$, because $ik < nk = N - 1$. ω is a primitive root of unity as required by the DFT of length n
f) The rest of the procedure for the forward and inverse transforms is identical to the complex DFT. Moreover, the NTT can be modified to implement a fast Fourier transform algorithm such as Cooley-Turkey.

The proposed system provides an additional modulus operation which will lead to elimination of complex number by taking modulus value for the twiddle factor hence the imaginary part becomes real part hence there is no wastage of memory buffers to store real and imaginary value separately. This will provide a revolutionary design to pay the way for multipliered architecture rather than the multiplierless architecture.

3. Implementation of Double Modulus Number Theoretic Transform

The proposed double modulus Number Theoretic Transform is achieved through a procedural flow based on the following figure as shown in figure 1.

Licensed Under Creative Commons Attribution CC BY

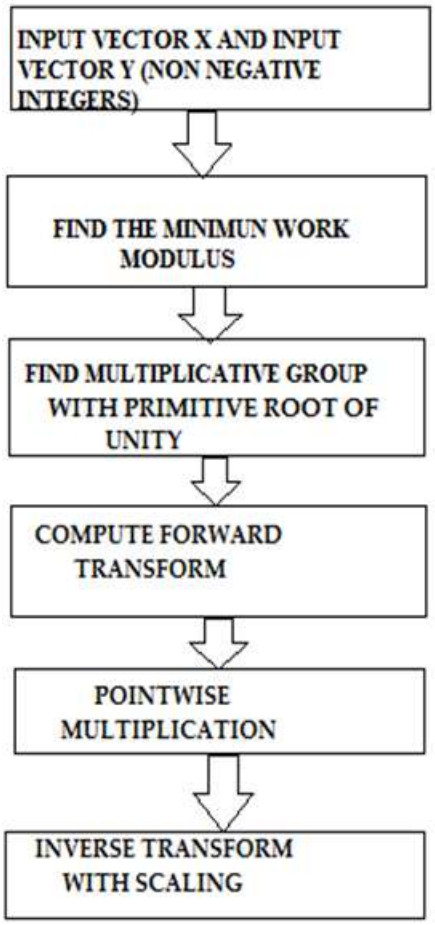

Figure 1: Flowchart for proposed architecture of Double Modulus Number Theoretic Transform

The flow chart followed by which the input vectors X and Y which is non negative integer which means that by double modulus enlarges the word size and the NTT provides the efficient multiplication for larger size integers and the minimum work modulus M which can be decided based on the size n thus purpose of using double modulus can be employed in upcoming steps due to selection of minimum work modulus the computation of twiddle factors is a major to be explained in the butterfly structure of Double Modulus NTT. Both forward and inverse transform are computed for multiplication in which the inverse transform can be performed with scaling and pointwise multiplication calculated which is in turn meant to be the results of circular convolution the number theoretic transform provides fast convolution with less computation time which is also an bufferless architecture there is no need store the data in between the multiplication structure. Basically the NTT inputs of the odd and even sequence it can be zero padded at the inputs further the operations can be performed hence there is less probability of truncation error and the main advantage of using NTT architecture has zero round off errors. It provides data transfer in Graphic Processing Unit with negligible amount of data loss during transmission.

3.1 Butterfly Structure for Double Modulus Number Theoretic Transform

The butterfly architecture of NNT architecture shows that the twiddle factor multiplication is a real data without imaginary values hence computation becomes easier with real numbers when comparing to computation with imaginary number with less computational complexity of $O(N/4 \log_2 N)$.

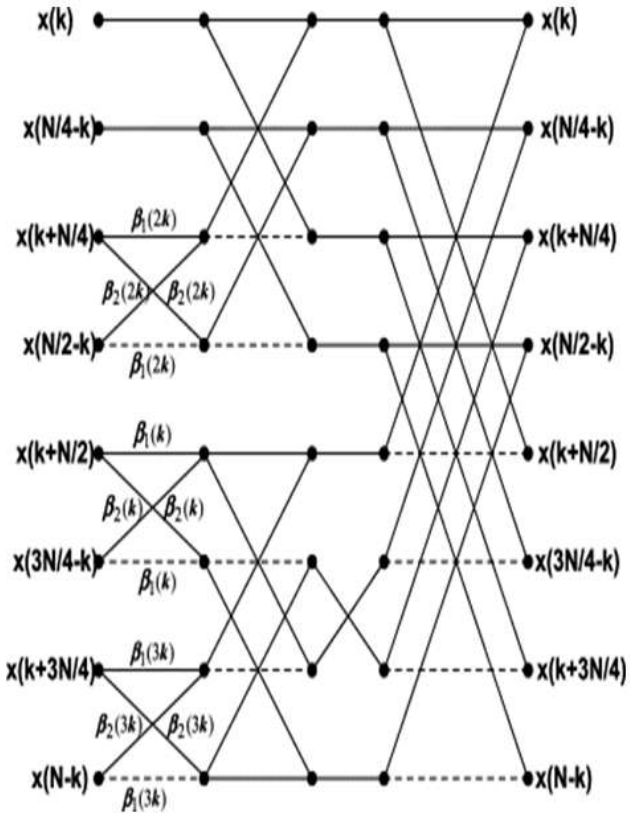

Figure 2: Butterfly Diagram for Double Modulus Number Theoretic Transform

The proposed architecture provides the computation only with real numbers by incorporating modulo multiplication and the result would be modulus value hence the storage space for larger the word size can be reduced to 25% of its original word size the memory operation in GPU performed using this NTT architecture can be made easier and simpler using this architecture , the number of buffer storage can be reduced and final accumulation of bits with inverse transform and scaling of data provides an area efficient multiplication. By using this number theory method for transforms which is an Euler's theorem then the addition modulo a prime integer and integer multiplication are exact hence there is no round off errors and used for faster convolutions, correlations which also has millions digits of precision. The computational complexity of NTT architecture has been lower that can be used in schonnage strassen algorithm, karatsuba multiplication and cook toom algorithm procedure to speed up the multiplication operation with less computation time and less storage.

4. Results and Comparisons

The verilog HDL code for proposed Double modulus Number Theoretic Transform is synthesized and verified using Xilinx tool. The performance parameter area, delay and execution time are analyzed.

Licensed Under Creative Commons Attribution CC BY

Table 1: Comparison of Double modulus Number Theoretic Transform with Fast Fourier Transform

S/N	Logic Utilization	FFT Utilization	FFT Utilization Percentage	NTT Utilization	NTT Utilization Percentage
1.	Number of Slice Registers	192	0%	79	0%
2.	Number of Slice LUTs	6768	32%	0	0%
3.	Number of Fully Used LUT-FF Pairs	191	2%	19	9%
4.	Number of Bonded IOBs	705	335%	15	18%

The comparison of Double Modulus Number Theoretic Transform and Fast Fourier Transform has been performed based on the implementation results shown in Table 1 and their utilization percentage also greatly reduced.

Table 2: Comparison of FFT and Double Modulus NTT based on Device utilization

S/N	Device Utilization	FFT	Double Modulus NTT
1	Multiplier	20	12
2	Adders/Subractors	268	18
3	Multiplexer	128	8
4	Registers	6	-

The comparison has been based on the implementation results in Table 2 reveals that all the parameters included can be greatly reduced. The double modulus Number Theoretic Transform utilizes the 12 multiplier, 18 adder/subractor units 8 multiplexer and there is no register utilization because in double modulus no need of registers storage the twiddle factors used here can be real numbers there are no imaginary numbers the speed of multiplication operation can be increased and there is a significant reduction in computation time due to double modulus method. The Fast Fourier Transform has been generated using Fast Fourier Transform and the same could be generated for Double Modulus Number Theoretic Transform and their comparison has been proven the double modulus Number Theoretic Transform provides the decrease in both logical and device wise utilization.

Table 3: Comparison of FFT and Double modulus NTT based on Delay and memory usage

S/N	Parameters	FFT	Double Modulus NTT
1	Memory Usage	279304 Kb	231944 Kb
2	Maximum Combinational Path Delay	46.596ns	2.595ns

The area efficient architecture shows the great impact on the reduction of memory usage and has reduced combinational path delay from 46.596ns to 2.595ns and memory usage reduced to 231944kb according to the implementation results compared in Table 3.

Table 3: Resources allocated by the Double Modulus Number Theoretic Transform

RESOURCES SUMMARY

Macrocells Used	Pterms Used	Registers Used	Pins Used	Function Block Inputs Used
16/36 (45%)	24/180 (14%)	0/36 (0%)	19/34 (56%)	14/108 (13%)

PIN RESOURCES

Signal Type	Required	Mapped	Pin Type	Used	Total
Input	3	3	I/O	19	28
Output	16	16	GCK/IO	0	3
Bidirectional	0	0	GTS/IO	0	2
GCK	0	0	GSR/IO	0	1
GTS	0	0			
GSR	0	0			

GLOBAL RESOURCES

Global clock net(s) used	0
Global output enable net(s) used	0
Global set/rest net(s) used	0

The number of points in FFT and Double Modulus NTT has been compared with execution time with reference values with [9] and the Double Modulus NTT provides less execution time than other existing system taken as reference shown in figure 3.

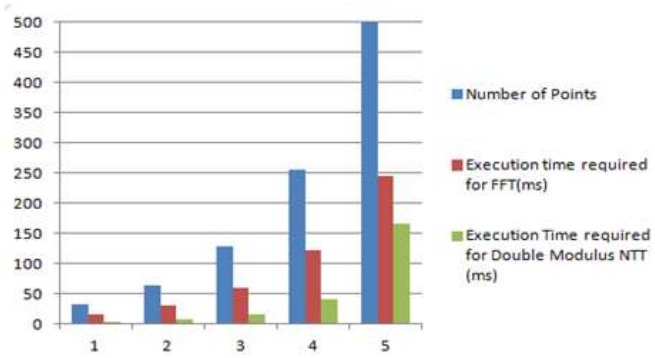

Figure 3: A graph representing the number of points compared to FFT and Double Modulus NTT execution time

5. Conclusion

The proposed double modulus Number theoretic Transform (NTT) method enlarges the digit size of NTT and the double modulus technique provides a modulo operation which converts the imaginary data input into a real number which leads to reduction in buffer storage. The twiddle factor multiplication becomes easier by introducing the complex Nth root of unity. The double modulus technique is synthesized and simulated using the Xilinx and their parameters of area, power and speed can be analyzed. Hence area occupied by the buffer can be greatly reduced and which makes an area efficient multiplication. By doing so this design can pay way for the multipliered architecture rather than going for multiplier less architecture and this idea of concept can be implemented in CUDA graphics processor which can be employed in video filtering techniques and also in high definition video displays. It will greatly help in providing an area sufficient structure in ALU (Arithmetic Logic Unit) which is mandatory for GPU (Graphic Processing Unit).

Licensed Under Creative Commons Attribution CC BY

References

[1] Mayura Patrikar, Prof.Vaishali Tehre "Design and Power Measurement of 2 And 8 Point FFT Using Radix-2 Algorithm for FPGA Implementation" IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 7, Issue 1, Ver. I (Jan. - Feb. 2017), PP 44-48 e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197.
[2] Muhammad Ibrahim; Mohsin Kamal; Omar Khan; "Analysis of radix2 decimation in time algorithm for FPGA co-processors"2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)Year: 2016.
[3] Muhamad Ibrahim;Omar Khan" Performance analysis of Fast Fourier Transform on Field Programmable Gate Arrays and Graphic crards" 2016 International Conference on Computing, Electronic and Eleactrical Engineering(ICE Cube) year:2016.
[4] Jiaxi Hu; Zhaosen Wang; Qiyuan Qiu; Weijun Xiao; "Sparse Fast Fourier Transform on GPUs and Multi core CPUs"2012 IEEE 24th International Symposium on Computer Architecture and High Performance Computing Year: 2012.
[5] Ulloa; Juan RamonTroncoso Pastoriza; Fernando PerezGonzalez"Number Theoretic Transforms for Secure Signal Processing"IEEE Transactions on Information Forensics and Security Year: 2017, Volume: 12, Issue: 5.
[6] Gavin Xiaoxu Yao;C.C.Cheung XiaoxuYao;CetinKaya Koc; "Reconfigurable Number Theoretic Transform architectures for cryptographic applications" 2010 International Conference on Field-Programmable Technology Year: 2010.
[7] Juliano B. Lima; Ricardo M. Campello de Souza; Paulo Hugo E. S. Lima" Fracttional number-theoretic transforms based on matrix functions"2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) Year: 2014.
[8] O. M. Ulyanov; M. S. Plakhov; A. I. Shevtsova; A. O. Skoryk; V. N. Tkachev; O. O. Ulyanova "Testing the speed of FFT/IFFT using the NVIDIA graphics cards"2016 9th International Kharkiv Symposium on Physics and Engineering of Microwaves, Millimeter and Submillimeter Waves (MSMW) Year: 2016.
[9] Xiang Feng; Shuguo Li "Design of an area efficient Million bit Integer Multiplier Using Double Modulus NTT" IEEE Transactions on Very Large Scale Integration (VLSI) Systems Year: 2017, Volume: 25, Issue: 9 Pages: 2658 - 2662 IEEE Journals & Magazines.
[10] Boon-Chiao Chang; Bok-Min Goi; Raphael C. -W. Phan; Wai-Kong Lee" Accelerating Multiple Precision Multiplication in GPU with Kepler Architecture" 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS) Year: 2016 Pages: 844 - 851 IEEE Conferences

Licensed Under Creative Commons Attribution CC BY