Unusual-Length Number-Theoretic Transforms Using Recursive Extensions of Rader's Algorithm

Abstract

A novel decomposition of NTT blocklengths is proposed using repeated applications of Rader's algorithm to reduce the problem to that of realising a single small-length NTT. An efficient implementation of this small-length NTT is achieved by an initial basis conversion of the data, so that the new basis corresponds to the kernel of the small-length NTT. Multiplication by powers of the kernel become rotations and all arithmetic is efficiently performed within the new basis. More generally, this extension of Rader's algorithm is suitable for NTT or DFT applications where an efficient implementation of a particular small-length NTT/DFT module exists.

1 Introduction

The Number-Theoretic Transform (NTT) has been suggested as an alternative to the DFT for computing cyclic convolution [14, 9] and is suitable for inclusion within signal processing, error-correction and residue number systems. Efficient architectures are possible for Fermat and Mersenne Transforms [8, 9, 1], where multiplication within the transform is eliminated due to the correspondence of the NTT kernel to the basis of the data (i.e. the kernel is some simple power of 2 and the data is represented using a binary basis). However, there are only a few blocklengths over which these NTTs are possible. Various well-known blocklength decomposition schemes can be used to widen the choice of NTT blocklengths [4, 3, 17, 2, 6] and, in this paper, a novel decomposition is suggested, based on successive applications of Rader's algorithm [15]. The technique is suitable for a wide range of prime and composite blocklengths and only requires repeated applications of a single, small-length NTT. It is, therefore, highly suitable for reduced hardware systems. Furthermore, by applying a preliminary basis conversion and matching the basis of the data to the kernel of the small-length NTT [10, 11, 13], all kernel multiplications can be eliminated, and the only multiplications are the fixed multiplications, inherent in the NTT-based cyclic convolutions, which are required to realise Rader's algorithm. Although this paper develops the theory in terms of the NTT, the extension of Rader's algorithm is equally valid for DFTs, and may be combined with efficient Winograd solutions for small-length DFTs [17] to construct unusual-length DFTs.

2 Theory

The N-point NTT is given by,

$$X[k] = \left\langle \sum_{n=0}^{N-1} x[n].\alpha^{n.k} \right\rangle_M \qquad 0 \leq k < N \tag{1}$$

where α is an N $^{\text{th}}$ root of 1, mod M and, for the purposes of this discussion, M is considered prime.

Consider the case where,

$$N_p = N + 1 \qquad N_p \text{ prime} \tag{2}$$

Then a one-dimensional $N.N_p$-point NTT can always be defined, mod M, where,

$$M = r.N.N_p + 1 \text{ for } r \text{ integer, positive, and } M \text{ prime} \tag{3}$$

This $N.N_p$-point NTT can be expressed as,

$$X[k] = \left\langle \sum_{n=0}^{N.N_p-1} x[n].(\alpha^{-N_p}.\beta^{-N})^{n.k} \right\rangle_M \qquad 0 \leq k < N.N_p \tag{4}$$

where α, β and $(\alpha^{-N_p}.\beta^{-N})$ are N $^{\text{th}}$, N_p $^{\text{th}}$ and $N.N_p$ $^{\text{th}}$ roots of 1, mod M, respectively.

As $\gcd(N, N_p) = 1$, (4) can be reformulated using the Prime Factor Algorithm (PFA) [9] as a two-dimensional NTT,

$$X[k_0, k_1] = \left\langle \sum_{n_1=0}^{N_p-1} x[k_0, n_1].\beta^{n_1.k_1} \right\rangle_M \qquad 0 \leq k_1 < N_p \tag{5}$$

where

$$x[k_0, n_1] = \left\langle \sum_{n_0=0}^{N-1} x[n_0, n_1].\alpha^{n_0.k_0} \right\rangle_M \qquad 0 \leq k_0 < N \tag{6}$$

$$\text{with } n = \left\langle N_p.\left\langle N_p^{-1} \right\rangle_N .n_0 + N.\left\langle N^{-1} \right\rangle_{N_p} .n_1 \right\rangle_{N.N_p} \quad \text{and } k = \langle N_p.k_0 + N.k_1 \rangle_{N.N_p}$$

(5) and (6) comprise N N_p-point NTTs, mod M, and N_p N-point NTTs, mod M, respectively. If an efficient architecture exists for the computation of the N-point NTT, then the first dimension (6), of the above 2-D NTT is easily computed. In particular, the input data, $x[n]$, can, for a given n, be pre-converted to an α-basis, such that,

$$x[n] = \left\langle \sum_{i=0}^{l-1} x_i.\alpha^i \right\rangle_M \qquad \text{for } x_i \in \{\mathbf{S}\} \text{ and } \alpha \text{ an } N \text{ }^{\text{th}} \text{ root of 1, mod } M \tag{7}$$

where $\mathbf{S}$ is a small set of integers, (in the extreme, $\mathbf{S} = \{0, 1\}$), and l is equal to or slightly less than some multiple of N. In such a case, the NTT of (6) can be computed without multiplication, the multiplications by powers of α being replaced with rotations of $x[n]$ [10, 11]. (These principles are simply generalisations of the ideas inherent within the implementations of Fermat or Mersenne Number Theoretic Transforms [8, 9, 1, 5]). To compute the N_p-point NTTs, (5), multiplication by powers of the kernel, β, are required. These NTTs will not be simply realised in an α-basis as β cannot be a simple power of α. It is possible to convert the data from an α-basis to a β-basis after having performed the N-point NTTs and prior to performing the N_p-point NTTs [13]. However, suitable basis converters may not always be feasible and, for word-parallel implementations, a large number of basis converters will be required. An alternative solution, the subject of this paper, is to utilise Rader's algorithm [15] to compute the N_p-point NTTs using N-point NTTs, mod M.

Rader's algorithm states that a P-point DFT, where P is prime, can be computed using a $P-1$-point complex cyclic convolution (CC). This $P-1$-point CC can, in turn, be computed using any orthogonal transform, (such as the NTT), possessing the 'Cyclic Convolution Property'. Similarly, a P-point NTT, mod M, can be computed using a $P-1$-point CC, mod M, which can, in turn, be computed using $P-1$-point NTT/INTTs, mod M, providing a $P-1$-point NTT exists, mod M. From (3) it is clear that M supports N_p and N-point NTTs. Therefore the N_p-point NTT of (5) can be decomposed as follows,

$$Y[0] = \left\langle \sum_{n=0}^{N_p-1} y[n] \right\rangle_M \tag{8}$$

$$Y[k] = \langle Y'[q] + y[0] \rangle_M \quad 1 \leq k < N_p \tag{9}$$

where,

$$Y'[q] = \left\langle \sum_{p=0}^{N_p-2} y'[p].\beta'_{q-p} \right\rangle_M \quad 0 \leq q < N_p - 1 \tag{10}$$

$$n = \langle g^{-p} \rangle_{N_p} \quad , \quad k = \langle g^q \rangle_{N_p} \tag{11}$$

$$y'[p] = y[\langle g^{-p} \rangle_{N_p}] = y[n] \quad \beta'_{q-p} = \beta^{\langle g^{q-p} \rangle_{N_p}} = \beta^{n.k} \tag{12}$$

$$Y'[q] = Y[\langle g^q \rangle_{N_p}] = Y[k] \tag{13}$$

and g is an $N_p - 1$ ${}^{\text{th}}$ root of 1, mod N_p. Thus, using N-point NTT/INTTs to compute the $N_p = N + 1$ -point CC of (10),

$$Y'[q] = \left\langle N^{-1}. \sum_{k=0}^{N-1} F(k).G(k).\alpha^{-q.k} \right\rangle_M \quad 0 \leq q < N \tag{14}$$

where

$$F(k) = \left\langle \sum_{p=0}^{N-1} y'[p].\alpha^{p.k} \right\rangle_M \quad 0 \leq k < N \tag{15}$$

and,

$$G(k) = \left\langle \sum_{p=0}^{N-1} \beta'_p.\alpha^{p.k} \right\rangle_M \quad 0 \leq k < N \tag{16}$$

(16) can be pre-computed, and $G(0)$ is always equal to -1. Furthermore, one other value of $G(k)$ will be a power of α. Therefore the CC of (10) can be computed in an α-basis using

only $N-2$ fixed (non-trivial) multiplications. This can be considered a great improvement on the direct α-basis implementation of an N_p-point NTT, mod M, which requires approximately $(N_p-1)^2$ fixed multiplications. This PFA/Rader-based $N.N_p$-point NTT can be compared to a direct, undecomposed version and a PFA-based version, as follows.

An undecomposed $N.N_p$-point NTT, mod M, will require,

$$(N.N_p-1)^2 \text{ fixed multiplications, mod } M \text{ (ignoring additions)}$$

Using a $N \times N_p$ PFA decomposition, the $N.N_p$-point NTT will require,

$$N_p.(N-1)^2 + N.(N_p-1)^2 \text{ fixed multiplications, mod } M \text{ (ignoring additions)}$$

The PFA/Rader decomposition of the $N.N_p$-point NTT, as described in this paper, will require,

$$(N_p + 2.N)\ N\text{-point NTTs and } N.(N-2) \text{ fixed multiplications, mod } M$$

Assuming each N-point NTT is performed efficiently without multiplication in an α basis, then the complexity of each NTT can be approximately equated as,

$$1\ N\text{-point NTT, mod } M \ \simeq N^2/\lceil \log_2(M) \rceil \text{ fixed multiplications, mod } M$$

Therefore the PFA/Rader method requires approximately,

$$(N_p + 2.N).N^2/\log_2(M) + N.(N-2) \text{ fixed multiplications, mod } M$$

As a further approximation, if one assumes $N_p \simeq N$, the above estimates reduce to,

$$\begin{aligned}
&\text{Undecomposed } N.N_p\text{-point NTT, mod } M \simeq N^4 \text{ fixed multiplications, mod } M \\
&\text{PFA-Based } N.N_p\text{-point NTT, mod } M \simeq 2.N^3 \text{ fixed multiplications, mod } M \\
&\text{PFA/Rader-Based } N.N_p\text{-point NTT, mod } M \simeq \\
&\qquad 3.N^3/\lceil log_2(M) \rceil + N^2 \text{ fixed multiplications, mod } M
\end{aligned} \tag{17}$$

The figure for the PFA/Rader-based NTT compares favourably with the other methods. Furthermore, all additions and multiplications will be performed in an α-basis, and, as shown in

[12], particularly efficient implementations of modular arithmetic operations are possible if the basis α satisfies (7) for a given modulus, M.

As an example,

Let $N = 16$, $N_p = 17$, $M = 1361$. Note that $\alpha = 63$, where 63 has order 16, mod 1361. All integers, mod 1361 can be represented using a 63-basis, as specified in (7), where l is a minimum of 16 and $\mathbf{S} = \{0, 1\}$, i.e. all data, mod 1361, can be represented using 16-bit words, where each consecutive bit represents a consecutive power of 63. Furthermore all 16-point NTT kernel products, mod 1361, can be implemented as bit rotations. Therefore a $16 \times 17 = 272$-point NTT, mod 1361, can be implemented using $17 + 32 = 49$ multiplierless 16-point NTTs, mod 1361, and $16.14 = 224$ fixed multiplications for the 16-point CCs, with all arithmetic performed in a 63-basis. Using the estimates of (17),

Undecomposed 272-point NTT, mod 1361 $\simeq$ 65536 fixed multiplications, mod 1361

PFA-Based 272-point NTT, mod 1361 $\simeq$ 8192 fixed multiplications, mod 1361

PFA/Rader-Based 272-point NTT, mod 1361 $\simeq$ 1374 fixed multiplications, mod 1361

3 Recursive Extensions of Rader's Algorithm

Further Rader-based, composite NTTs can be constructed from the minimally composite NTTs of the last section using prime-factor techniques. Thus, if,

$$N_{p_2} = N.N_{p_1} + 1 \qquad \text{with } N_{p_1}, N_{p_2} \text{ prime} \tag{18}$$

then N_{p_2}-point NTTs can be computed, mod M, using $N.N_{p_1}$-point NTTs, as described in the last section, where,

$$M = r.(N.N_{p_1}.N_{p_2}) + 1 \qquad \text{for } r \text{ integer, positive, and } M \text{ prime} \tag{19}$$

As N_{p_2} is mutually prime to both N and N_{p_1}, any composite NTT, mod M, which comprises any, or all of N, N_{p_1}, and N_{p_2}, can be computed using only N-point NTTs, mod M.

For example, if $N = 12$, $N_{p1} = 13$, then $N_{p2} = 157$ and 24492, $(= 12 \times 13 \times 157)$, 1884, $(= 12 \times 157)$, and 2041, $(= 13 \times 157)$-point NTTs can all be defined, mod M, $(= r.24492 + 1)$, using only 12-point NTT modules.

These sequences of N_{p_i} can be iterated as long as all N_{p_i} are prime. Unfortunately few of these "Rader Sequences" continue for more than one or two times before reaching a non-prime. Table 1 shows some example sequences for selected N. Note that, as a special case, a $2 \times 3 \times 7 \times 43 = 1806$-point NTT can be computed, mod M, where $M = r.1806 + 1$, using only 2-point NTTs, mod M.

For a given N, the sequence set can be extended to include any primes of the form $P + 1$, where P is generated as a multiplicative product of any or all members of the set, $\{N, 2, N_{p_1}, N_{p_2}, \ldots\}$. (Note, 2 is included as 2-point NTTs are simply implemented and never require multiplication). Table 2 shows a selection of extended Rader sequences.

N	Rader Sequence
2	3,7,43
4	5
6	7,43
10	11
12	13,157
..	..
66	67,4423, ... etc
..	..
192	193,37057, ... etc
..	..
456	457,208393, ... etc

Table 1: Rader Sequences for Selected N

Although the sequences of Table 2 include a wide selection of primes, the most effective combinations will include as few 2-point and as many N-point NTTs as possible. A further broadening of the sequences, (not shown in Table 2), is possible by also including all t, where $t|N$. This is justified because, if N-point NTTs are efficiently implemented, without multiplication,

N	Extended Rader Sequence
2	3,7,43
3	2,7,43
4	2,5,3,11,13,7,61,31,23,67,661,331,53,131,157 etc
5	2,3,11,7,31,23,67,331,71,43,211,463,2311,311, etc
6	2,7,3,43,19,127,5419,4903,2287,98299,14479,101347, etc
7	2,3,43
8	2,3,7,43,1033,24793,6199, etc
9	2,3,,19,7,127,43,379,2287,14479,130303,304039, etc
10	2,11,3,23,31,7,331,67,47,2531,691,139,7591,311 etc
11	2,3,23,7,67,47,139,43,463,967,10627,4423,3083, etc
12	2,13,3,157,37,7,79,5653,73477,223,17317,2887,69709, etc
66	2,67,3,4423,199,7,13267,26539,79999,463,43,92863, etc
192	2,193,3,37057,577,7,3463,43,148867,57793,348559, etc
456	2,457,3,208393,7,43,19609, etc

Table 2: Extended Rader Sequences for Selected N

in an α-basis, then t-point NTTs will also be efficiently implemented without multiplication.

As a final example, a $5 \times 11 \times 23 = 1265$-point NTT, mod 245411, can be implemented using an $\alpha = 22$-basis, where 22 has order 5, mod 245411, $l = 5$ and $\mathbf{S} = \{0, 1, 2, \ldots, 21\}$,. The 1265-point NTT uses only 5-point and 2-point NTTs, mod 245411, with all multiplications by powers of 22 implemented as rotations. Note, the blocklengths 11 and 23 are decomposed, using the PFA and Rader algorithms, as $(5 \times 2) + 1$ and $(((5 \times 2) + 1) \times 2) + 1$, respectively.

The complete 1265-point NTT, mod 245411, requires,

1593 5-point NTTs

4560 2-point NTTs

3180 fixed multiplications

and initial 2 to 22 basis conversions for all incoming data

Unlike the previous example, where $\alpha = 63$ and $\mathbf{S} = \{0, 1\}$ (requiring 1 bit), in this example the set $\mathbf{S}$ 'spans' the basis (and requires 5 bits). In other words, a representation for all integers $\{0, 1, \ldots, \alpha - 1\}$ are contained in $\mathbf{S}$. In this example, $\{0, 1, \ldots, 21\} \in \mathbf{S}$. This spanning guarantees a simple implementation for addition and, consequently, general multiplication, mod 245411, as additive carry propagation is localised [12]. Moreover, $\mathbf{S}$ can be widened up to $\{0, 1, 2, \ldots, 31\}$ (whilst still requiring 5 bits), and all computations can now use redundant arithmetic structures [7] with reduced carry propagations and, consequently, increased speed. The greatest drawback with this particular NTT example is the increased wordlength requirements, from $\lceil \log_2(245411) \rceil = 18$ bits, using a conventional binary representation, to $5 \times 5 = 25$ bits using a 22-basis with $\mathbf{S} = \{0, 1, 2, \ldots, 31\}$. It is hoped that more competitive, Rader-based NTTs will become apparent over larger moduli.

4 Conclusion

A repeated application of Rader's algorithm has been proposed for the realisation of unusual-length NTTs. It allows the construction of relatively long-length NTTs using a single, small-length NTT. This small-length NTT can be efficiently implemented, without multiplication, by a preliminary basis conversion, so that the basis representation of the data corresponds to the kernel of the small-length NTT [12]. Thus the multiplication count of the complete NTT is reduced to the relatively small number of point-product multiplications, inherent within each application of Rader's algorithm. The method has been developed in conjunction with a prime-factor decomposition and, as the blocklengths comprise unusual primes, these NTTs are easily combined with more conventional NTTs, (such as FNTs), using prime-factor techniques, to construct even larger NTTs. This recursive Rader algorithm is particularly suited to reduced-hardware solutions and is also applicable to any DFT for which an efficient small-length DFT module exists.

References

[1] M.Benaissa,S.S.Dlay,Prof A.G.J.Holt, "CMOS VLSI Design of a High-Speed FNT Based Convolver/Correlator Using 3-input Adders," *IEE Proc-G*, Vol 138, No.2, pp 182 - 190, April '91

[2] R.E.Blahut, **Fast Algorithms for Digital Signal Processing,** Reading, Addison-Wesley, '85

[3] J.W.Cooley,J.W.Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", *Math. Comput.*, Vol 19, No 2, pp 297 - 301, '65

[4] I.J.Good, "The Interaction Algorithm and Practical Fourier Analysis", *J. Roy. Stat. Soc. B*, Vol 20, No 2, pp 361 - 372, '58

[5] M.Honda,M.Kameyama,T.Higuchi, "Residue Arithmetic Based Multiple-Valued VLSI Image Processor," *Proc of the 22nd Int Symp on Multivalued Logic*, (IEEE '92) pp 330 - 336, '92

[6] K.J.Jones, "High-Throughput, Reduced Hardware Systolic Solution to Prime Factor Discrete Fourier Transform Algorithm", *IEE Proc-E*, Vol 137, No 3, pp 191 - 196, May '90

[7] S.C.Knowles,J.G.McWhirter, "The Application of Redundant Number Systems to the Design of VLSI Recursive Filters", *IMA Conf on Mathematics in Signal Processing, Univ of Warwick*, Dec 13 th - 15 th , pp 27-42, '88

[8] J.H.McClellan, "Hardware Realisation of a Fermat Number Transform", *IEEE Trans on ASSP*, Vol 24, No 3, pp 216 - 225, '76

[9] J.H.McClellan,C.M.Rader, **Number Theory in Digital Signal Processing,** Prentice Hall, '79

[10] M.G.Parker,M.Benaissa, "A Bit-Serial, VLSI Implementation of a 60-Point NTT Using Binary and Ternary Bases," *Digest of Int Symp on DSP for Communication Systems, Univ of Warwick* 7 - 9 th Sept '92

[11] M.G.Parker,M.Benaissa, "Bit-Serial, VLSI Architecture for the Implementation of Maximum-Length Number-Theoretic Transforms Using Mixed Basis Representations," *Proc of ICASSP '93, Minneapolis*, Vol I, pp 341 - 344, April '93

[12] M.G.Parker,M.Benaissa, "Using Redundant Number Representations for Efficient VLSI Implementation of Modular Arithmetic," *Proc of IEE Coll on "Synthesis and Optimisation of Logic Systems"*, (E3,E10), Savoy Place, London, 14 th Mar, '94, Digest No: 1994/066

[13] M.G.Parker,M.Benaissa, "VLSI Structures for Bit Serial Modular Multiplication Using Basis Conversion," *IEE Proc-Comput.Digit.Tech*, Vol 141,No 6,pp 381-390, Nov '94

[14] J.M.Pollard, "The Fast Fourier Transform in a Finite Field," *Math.Comp*, Vol 25, No 114, pp 266 - 273, Apr '71

[15] C.M.Rader, "Discrete Fourier Transforms when the Number of Data Samples is Prime," *Proc IEEE*, Vol 56, pp 1107 - 1108, '68

[16] T.K.Truong,I.S.Reed,I.S.Hsu,H.C.Shyu,H.M.Shao, "A Pipeline Design of a Fast Prime Factor DFT on a Finite Field," *IEEE Trans Computers*, Vol 37, No 3, pp 365 - 374, March '88

[17] S.Winograd, "On Computing the Discrete Fourier Transform," *Maths of Computation*, Vol 32, No 141, pp 175 - 199, Jan '78