SPLIT-RADIX ALGORITHM FOR THE NEW MERSENNE NUMBER TRANSFORM

O. Alshibami[1], S. Boussakta[1], M. Aziz[1], and D. Xu[2]

[1]Institute of Integrated Information Systems,
School of Electronic and Electrical Engineering,
The University of Leeds, Leeds LS2 9JT, UK
E-mail: s.boussakta@ee.leeds.ac.uk
[2]Electrical and Electronic Engineering Section,
University of Teesside, Borough road, Middlesbrough, TS1 3BA, UK

ABSTRACT: The one-dimensional new Mersenne number transform (NMNT) was proposed for the calculation of error free convolutions and correlations for signal processing purposes. The aim of this paper is to develop the split-radix decimation-in-time algorithm for fast calculation of the one-dimensional NMNT with a sequence length equal to a power of two. The arithmetic complexity of this algorithm is analysed and the number of multiplications and additions is calculated. An example is given to prove the validity of the algorithm and the exact nature of this transform.

Keywords

New Mersenne number transform, Split-radix algorithm, Fast algorithms, Convolution, Correlation.

1. INTRODUCTION

Fast transforms, such as fast Fourier transform (FFT), fast Hartley transform (FHT) and number theoretic transforms (NTTs), have been used to speed up the computation process of the convolution/correlation by reducing the number of multiplications and additions [1-8].

Recently a new number theoretic transform called "New Mersenne Number Transform" was introduced [6]. The new Mersenne number transform is defined modulo the Mersenne numbers where arithmetic operations are simple (equivalent to 1's complement). The transform length is long and power of two making it suitable for fast algorithms. It has the cyclic convolution property and hence can be applied for the calculation of error-free convolutions and correlations.

In this paper, the split-radix algorithm for fast calculation of the new Mersenne number transform is developed using the same principles which were used with FFT and FHT. The arithmetic operations for the split-radix algorithm applied to the NMNT are determined and an example is given.

2. THE NEW MERSENNE NUMBER TRANSFORM DEFINITION

The new Mersenne number transform pair for a sequence $x(n)$ of length N is given by [6]:

$$X(k)=\left\langle \sum_{n=0}^{N-1} x(n)\beta(nk)\right\rangle_{Mp} \quad (1)$$

$$k=0,1,2,....,N-1$$

$$x(n)=\left\langle N^{-1}\sum_{k=0}^{N-1} X(k)\beta(nk)\right\rangle_{Mp} \quad (2)$$

$$n=0,1,2,....,N-1$$

where:

$$\beta(n)=\beta_1(n)+\beta_2(n) \quad (3)$$

$$\beta_1(n)=\left\langle \mathrm{Re}(\alpha_1+j\alpha_2)^n\right\rangle_{Mp} \quad (4)$$

$$\beta_2(n)=\left\langle \mathrm{Im}(\alpha_1+j\alpha_2)^n\right\rangle_{Mp} \quad (5)$$

also

$$\alpha_1=\pm\left\langle 2^q\right\rangle_{Mp};\ \alpha_2=\pm\left\langle -3^q\right\rangle_{Mp};\ q=2^{p-2} \quad (6)$$

α_1 and α_2 in (6) are of order $N=2^{p+1}$. For transform length N/d, where d is an integer power of two, β_1 and β_2 are given by:

$$\beta_1(n)=\left\langle \mathrm{Re}\left((\alpha_1+\alpha_2)^d\right)^n\right\rangle_{Mp} \quad (7)$$

$$\beta_2(n) = \left\langle \mathrm{Im}\left((\alpha_1 + \alpha_2)^d\right)^n \right\rangle_{Mp} \tag{8}$$

where Mp is a Mersenne Prime = 2^p-1. $\langle\ \rangle_{Mp}$ denotes mod Mp. Re(•) and Im(•) denote real and imaginary parts of the enclosed term respectively. The new Mersenne number transform, like other NTTs, avoids the introduction of the additional processing noise due rounding or truncation errors that may arise when the FFT [9] or FHT [10,11] are used for the calculation of convolutions or correlations. The new transform uses an integer kernel leading to an integer transform. The factor (N^{-1}) can be split between the forward and inverse transforms to make them exactly the same.

2.1 Cyclic Convolution Property

The new Mersenne number transform has the convolution property:

$$NMNT[x(n) * h(n)] = X(k)\Gamma H(k)$$
$$= X(k) \otimes H_{ev}(k) + X(-k) \otimes H_{od}(k) \tag{9}$$

where ⊗ is point by point multiplication, $H_{ev}(k)$ and $H_{od}(k)$ stand for even and odd parts of $H(k)$ respectively which are given by:

$$H_{ev}(k) = \left\langle (H(k) + H(N-k)) \times 2^{p-1} \right\rangle_{Mp} \tag{10}$$

and

$$H_{od}(k) = \left\langle (H(k) - H(N-k)) \times 2^{p-1} \right\rangle_{Mp} \tag{11}$$

The factor 2^{p-1} is due to the fact that:
($1/2 = 2^{p-1}$ mod Mp)

3. SPLIT-RADIX DECIMATION-IN-TIME ALGORITHM

The split-radix algorithm is one of the most efficient algorithms for computing fast transforms. It applies a radix-2 decomposition to the even indexed samples and a radix-4 decomposition to the odd indexed samples [1-4]. Therefore, $X(k)$ in (1) can be written as:

$$X(k) = \left\langle X_{ev}(k) + X_{od}(k) \right\rangle_{Mp} \tag{12}$$

where $X_{ev}(k)$ is even indexed samples, and $X_{od}(k)$ is odd indexed samples. Each one with length equal N/2. $X_{ev}(k)$ and $X_{od}(k)$ are given by:

$$X_{ev}(k) = \left\langle \sum_{n=0}^{N/2-1} x(2n)\beta(2nk) \right\rangle_{Mp} = X_{2n}(k) \tag{13}$$

and

$$X_{od}(k) = \left\langle \sum_{n=0}^{N/4-1} x(4n+1)\beta((4n+1)k) + \sum_{n=0}^{N/4-1} x(4n+3)\beta((4n+3)k) \right\rangle_{Mp} \tag{14}$$

Using the identity in (15), which has been proved in [6]:

$$\beta(m+n) = \left\langle \beta_1(n)\beta(m) + \beta_2(n)\beta(-m) \right\rangle_{Mp} \tag{15}$$

Equation (14) can be written as:

$$X_{od}(k) = \langle X_{4n+1}(k)\beta_1(k) + X_{4n+1}(N/4-k)\beta_2(k) + X_{4n+3}(k)\beta_1(3k) + X_{4n+3}(N/4-k)\beta_2(3k) \rangle_{Mp} \tag{16}$$

Replacing (13) and (16) into (12), $X(k)$ can be written as:

$$X(k) = \left\langle X_{2n}(k) + \left[X_{4n+1}(k)\beta_1(k) + X_{4n+1}(N/4-k)\beta_2(k)\right] + \left[X_{4n+3}(k)\beta_1(3k) + X_{4n+3}(N/4-k)\beta_2(3k)\right] \right\rangle_{Mp} \tag{17}$$

where $X_{2n}(k)$ is 1-D NMNT of length-N/2 point transform and $X_{4n+1}(k)$ and $X_{4n+3}(k)$ are 1-D NMNTs of length-N/4 points.
$X(N/4+k)$, $X(N/2+k)$, and $X(3N/4+k)$ can be derived from Equ. (17) using the relations (18)-(21) :

$$\beta_1(N/4+k) = \beta_1(N/4)\beta_1(k) - \beta_2(N/4)\beta_2(k) = -\beta_2(k) \tag{18}$$

$$\beta_1(3N/4+3k) = \beta_1(3N/4)\beta_1(3k) - \beta_2(3N/4)\beta_2(3k) = \beta_2(3k) \tag{19}$$

$$\beta_2(N/4+k) = \beta_2(N/4)\beta_1(k) + \beta_1(N/4)\beta_2(k) = \beta_1(k) \tag{20}$$

$$\beta_2(3N/4+3k) = \beta_2(3N/4)\beta_1(3k) + \beta_1(3N/4)\beta_2(3k) = -\beta_1(3k) \tag{21}$$

where $X(N/4+k)$, $X(N/2+k)$, and $X(3N/4+k)$ can be written as:

$$X(N/4+k) = \left\langle X_{2n}(N/4+k) - \left[X_{4n+1}(k)\beta_2(k) - X_{4n+1}(N/4-k)\beta_1(k)\right] + \left[X_{4n+3}(k)\beta_2(3k) - X_{4n+3}(N/4-k)\beta_1(3k)\right] \right\rangle_{Mp} \tag{22}$$

$$X(N/2+k) = \left\langle X_{2n}(k) - \left[X_{4n+1}(k)\beta_1(k) + X_{4n+1}(N/4-k)\beta_2(k)\right] - \left[X_{4n+3}(k)\beta_1(3k) + X_{4n+3}(N/4-k)\beta_2(3k)\right] \right\rangle_{Mp} \tag{23}$$

$$X(3N/4+k) = \left\langle X_{2n}(N/4+k) + \left[X_{4n+1}(k)\beta_2(k) - X_{4n+1}(N/4-k)\beta_1(k)\right] - \left[X_{4n+3}(k)\beta_2(3k) - X_{4n+3}(N/4-k)\beta_1(3k)\right] \right\rangle_{Mp} \tag{24}$$

3.1 Arithmetic Complexity

It is possible to perform the computation in place by combining two butterflies as shown in Fig. 1.

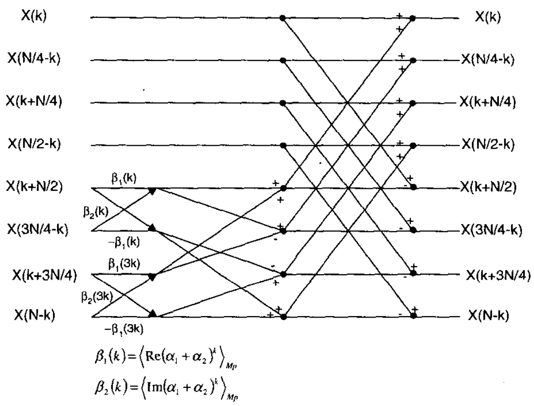

Figure 1. Split-radix DIT butterfly for NMNT.

The total number of multiplications and additions of the split-radix algorithm is given in table 1 with the total number of multiplications and additions of radix-2 and radix-4 algorithms. Based on the arithmetic operations, the split-radix algorithm is found to be the fastest, however, radix-2 algorithm has the simplest structure. An example of the 1-D NMNT and its inverse calculated using the split-radix algorithm and the parameters (Mp, N, α_1, α_2) = (127, 64, 49, -34) is given below:

Input data (generated randomly) = [16, 10, 2, 10, 16, 7, 15, 25, 28, 16, 4, 8, 1, 19, 12, 10, 12, 21, 23, 27, 29, 1, 20, 5, 14, 29, 12, 21, 19, 16, 21, 1, 25, 13, 14, 20, 19, 5, 21, 12, 9, 8, 26, 24, 12, 3, 26, 24, 29, 1, 27, 10, 25, 22, 18, 6, 17, 22, 4, 19, 9, 13, 6, 21]

NMNT = [91, 49, -55, 32, -104, 38, -79, 4, 84, -30, -53, 24, -14, -104, 37, -28, 119, -29, -42, 97, 44, 64, -122, 50, -5, 51, 8, 110, -5, 15, -91, -88, 82, -83, -4, -42, 96, -125, 21, -26, 29, -31, 19, -68, 39, 7, -47, 32, -61, 48, 71, 8, 85, 14, -27, -10, -28, -75, 104, 33, 98, 29, 6, 0]

NMNT^{-1} = [16, 10, 2, 10, 16, 7, 15, 25, 28, 16, 4, 8, 1, 19, 12, 10, 12, 21, 23, 27, 29, 1, 20, 5, 14, 29, 12, 21, 19, 16, 21, 1, 25, 13, 14, 20, 19, 5, 21, 12, 9, 8, 26, 24, 12, 3, 26, 24, 29, 1, 27, 10, 25, 22, 18, 6, 17, 22, 4, 19, 9, 13, 6, 21].

4. CONCLUSION

The new Mersenne number transform is defined modulo the Mersenne numbers where arithmetic operations are simple (equivalents to 1's complement) with long transform length equals to a power of two, making it suitable for fast algorithms such as radix-2, radix-4, split-radix etc. In this paper, an in-place split-radix decimation-in-time algorithm has been developed and its arithmetic complexity has been analysed. Compared to radix-2 and radix-4 algorithms, the split-radix is found to offer the lowest number of multiplications and additions but has the most complex butterfly structure. While the radix-2 has the simplest butterfly structure but has the highest number of multiplications and additions. An example has been given showing the validity of this algorithm and the exact nature of this transform.

5. ACKNOWLEDGMENT

The authors are pleased to acknowledge the support of the research grant from EPSRC.

6. REFERENCES

[1] P. Duhamel and H. Hollmann, "Split-radix FFT algorithm", *Electronics Letters*, vol. 20, 1984, pp. 14-16.

[2] H. V. Sorensen, M. T. Heidman, and C. S. Burrus, "On computing the split-radix FFT", *IEEE Trans.,* vol. ASSP-34, 1986, pp. 152-156.

[3] H. V. Sorensen, D. L. Jones, C. S. Burrus, and M. T. Heidman, "On computing the discrete Hartley transform", *IEEE Trans.*, vol. ASSP-33, no. 4, 1985, pp. 1231-1238.

[4] S. C. Pei and J. L. Wu, "Split-radix fast Hartley transform", *Electronics Letters*, vol. 22, 1986, pp. 26-27.

[5] R. C. Agarwal and C. S. Burrus, "Number theoretic transforms to implement fast digital convolution", *Proc. IEEE*, vol. 63, no. 4, 1974a, pp. 550-560.

[6] S. Boussakta, and A. G. Holt, "New transform using the Mersenne numbers", *IEE Proc.-Vis. Image Signal Process.*, vol. 142, no. 6, Dec.1995, pp. 381-388.

[7] L. S. Reed and T. K. Truong, "The use of finite fields to compute convolutions", *IEEE Trans.*, vol. SP-39, no. 6, 1991, pp. 1314-1321.

[8] L. R. Rabiner and B. Gold, "Theory and application of digital signal processing", (Prentice-Hall, Englewood Cliffs, London, 1975).

[9] Y. T. Ma, "An accurate error analysis model for fast Fourier transform", *IEEE Trans. on Signal Processing*, vol. 45, no. 6, 1997, pp. 1641-1645.

[10] A. Zakhor and A. V. Oppenheim, "Quantisation error in the computation of discrete Hartley transform", *IEEE Trans.*, vol. ASSP-35, no. 11, 1987, pp. 1592-1602.

[11] M. M. Rao and K. M. M. Prabhu, "Fixed-point error analysis of radix-4 FHT algorithm with optimised scaling schemes", *IEE Proc.-Vision Image Signal Processing*, vol. 142, no. 2, April 1995, pp. 65-70.

[12] O. Alshibami, S. Boussakta, and M. Aziz, "Radix-4 algorithm for the new Mersenne number transform", *Proc. of ICSP-2000*, Beijing, China, 21-25 Aug. 2000, pp. 54-56.

Table 1. Operations count per point for fast NMNT algorithms using three butterflies.

Transform length	Radix-2 [6]		Radix-4 [12]		Split-radix	
	Adds.	Mults.	Adds.	Mults.	Adds.	Mults.
2^3	3.25	0.5			2.75	0.25
2^4	4.63	1.25	4.38	0.88	4	0.75
2^5	6.06	2.13			5.19	1.31
2^6	7.53	3.06	7.03	2.22	6.5	1.94
2^7	9.02	4.03			7.8	2.58
2^8	10.51	5.02	9.76	3.68	9.13	3.23
2^9	12	6.01			10.45	3.89
2^{10}	13.5	7	12.5	5.17	11.78	4.56
2^{11}	15	8			13.11	5.22
2^{12}	16.5	9	15.25	6.67	14.45	5.89
2^{13}	18	10			15.78	6.56
2^{14}	19.5	11	18	8.17	17.11	7.22
2^{15}	21	12			18.44	7.89
2^{16}	22.5	13	20.75	9.67	19.78	8.56