

A NEW DISTRIBUTED ARITHMETIC VLSI ARCHITECTURE FOR DISCRETE FOURIER TRANSFORM PROCESSOR

S.M. REZAUL HASAN and C.M. HADZER

School of Electrical and Electronic Engineering

Universiti Sains Malaysia

31750 Tronoh, Perak

Malaysia

ABSTRACT

This brief paper examines the implementation aspects of a new distributed arithmetic VLSI architecture for Discrete Fourier Transform (DFT) processor. The new DFT hardware architecture is achieved by exploiting the symmetry and periodicity features of the transform kernel. The approach then leads to distributed arithmetic computation.

INTRODUCTION

In an effort to improve the speed, hardware cost and design complexity of discrete Fourier transform processors, many authors¹⁻⁶ have investigated different innovative fast Fourier transform implementations. The advent of digital VLSI technology, however, demands a reappraisal of most suitable algorithm and architecture for very high speed systolic implementation of discrete Fourier transform.

This paper offers a new alternative⁷ to fast Fourier transform-based hardware architectures for VLSI implementation of discrete Fourier transform processor. This new DFT hardware architecture is based on a new approach to exploiting the symmetry and periodicity of the transform kernel and the use of distributed arithmetic computation⁸.

SET SYMMETRY OF DFT COEFFICIENTS

The DFT and the inverse DFT is given by,

$$x(k) = \sum_{n=0}^{N-1} X(n) W_N^{nk}, \quad W_N = e^{-j(2\pi/N)} \quad (I)$$

$$X(n) = 1/N \sum_{k=0}^{N-1} x(k) W_N^{-nk} \quad (II)$$

Considering the coefficient vectors in equation (1); for $k = 0, 1, 2, \dots, N-1$ with $n = 0, 1, 2, \dots, N-1$ for each k , we have N -coefficient vector sets (one set for each value of k) with N coefficient vectors in each set.

These coefficient vector sets (N -coefficient vectors for each value of k) display a set symmetry as shown in Figure 1. $S(k)$ is the set of N -coefficient vectors for any particular k . Same height for $S(k)$ in Figure 1 indicates equal sets. For example, if $N=16$, $S(1) = S(3) = S(5) = S(7) = S(8) = S(9) = S(11) = S(13) = S(15)$, $S(4) = S(12)$, $S(2) = S(6) = S(10) = S(14)$. $S(0)$ is the unitary set and $S(8)$ is the central set. $S(0)$ and $S(8)$ are unequal to each other and also unequal to any other set. Coefficient vectors in equal sets are not necessarily indexed in the same order.

Also it is noted that (e.g., for $N = 8$) coefficient set for $k = 0$ { $S(0)$ } consists of 8 rotations of a 1-point DFT coefficient. Similarly, coefficient set for $k = 4$ { $S(4)$ } consist of 4 rotations of a 2-point DFT coefficient vector formation. Proceeding further, coefficient sets for $k = 2, 6$ consist of 2 rotations of a 4-point DFT coefficient vector formation. Finally, coefficient sets for $k = 1, 3, 5, 7$ consist of a single 8-point DFT formation. The order of the coefficients is ignored but only the geometrical formation is considered. Figure 2 explains the above statement.

The above analysis can be formalized as follows:-

An N -Point ($N = 2^v$, v is an integer) DFT has only $\log_2 N + 1$ different coefficient vector sets, consisting of $2^0, 2^1, 2^2, \dots, N$ rotations of $N/2^0, N/2^1, N/2^2, \dots, 1$ point DFT coefficient vector geometric formation respectively, disregarding the order of the coefficient vectors.

An analytical proof of the above theorem has also been obtained using number theoretic deductions as follows:

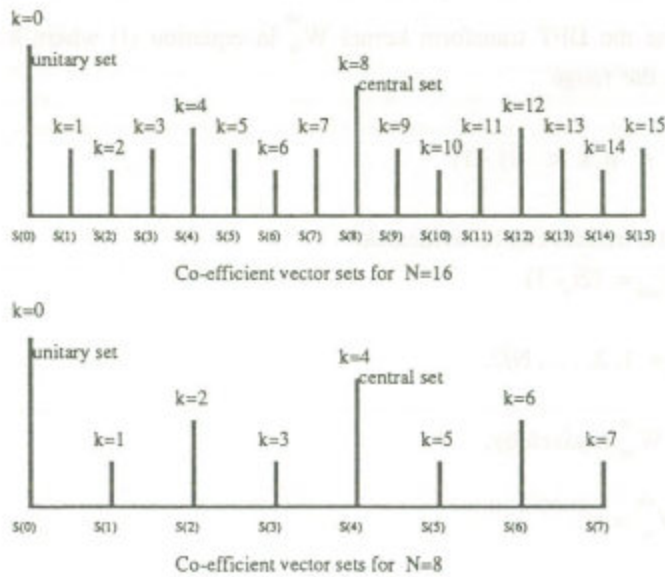
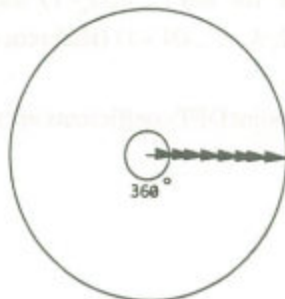
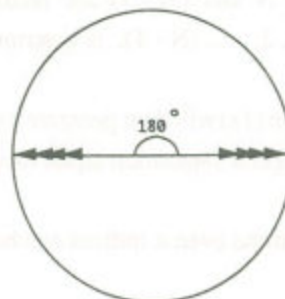


FIGURE 1. SET SYMMETRY OF DFT COEFFICIENT VECTORS



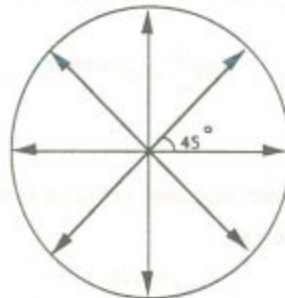
DFT coefficient vector formation for $K = 0$. Eight rotations of a 1-point DFT coefficient vector geometric formation



DFT coefficient vector formation for $K = 4$. Four rotations of a 2-point DFT coefficient vector geometric formation



DFT coefficient vector formation for $K = 2, 6$. Two rotations of a 4-point DFT coefficient vector geometric formation.



DFT coefficient vector formation for $K = 1, 3, 5, 7$. Single rotation of a 8-point DFT coefficient vector geometric formation.

FIGURE 2. GEOMETRIC FORMATIONS OF THE DFT COEFFICIENT VECTORS FOR AN 8-POINT DFT

Lets us examine the DFT transform kernel W_N^{nk} in equation (I) where k and n vary respectively in the range

$$0 < n, k < (N - 1)$$

Now all the odd k indices can be written as

$$k_{\text{odd}} = (2r_o - 1)$$

where

$$r_o = 1, 2, \dots, N/2.$$

then the kernel W_N^{nk} is given by,

$$W_N^{nk} = e^{-j(2\pi/N)n(2r_o-1)} \quad (1a)$$

In this case, $(2r_o - 1)$ for $r_o = 1, 2, \dots, N/2$ are odd numbers and $N = 2^v$ where v is an integer. Hence, N and $(2r_o - 1)$ are relatively prime. Therefore for any r_o , $n(2r_o - 1) \bmod N$, $n = 0, 1, 2, \dots, (N - 1)$, is a permutation of the set $0, 1, 2, 3, \dots, (N - 1)$ (Reference 9).

Equation (1a) will then generate a single (2^v) rotation of N -point DFT coefficients vector set with angular separation equal to a factor of $2\pi/N$.

Next, all the even k indices can be written as

$$k_{\text{even}} = 2r_1$$

where $r_1 = 0, 1, 2, \dots, (N/2 - 1)$. Then the kernel for all even indices can be written as,

$$W_N^{nk} = e^{-j(2\pi/N)n(2r_1)} = e^{-j(2\pi/(N/2))nr_1} \quad (1b)$$

In this case, equation (1b) is a kernel of an $N/2$ point DFT. Now all the odd r_1 indices can be written as

$$r_{1\text{odd}} = (2r_2 - 1)$$

where $r_2 = 1, 2, \dots, N/4$. Then the kernel in equation (1b) for odd indices can be written as

$$W_N^{nk} = e^{-j(2\pi/(N/2))n(2r_2-1)} \quad (2a)$$

Similarly, $(2r_2 - 1)$ for $r_2 = 1, 2, \dots, N/4$ are odd numbers and $N/2 = 2^v$, where v is an integer. Hence $N/2$ and $(2r_2 - 1)$ are relatively prime and therefore for any r_2 , $n(2r_2 - 1) \pmod{N/2}$, $n = 0, 1, \dots, (N/2 - 1)$ is a double (two) permutation of the set $0, 1, 2, \dots, (N/2 - 1)$ (Reference 9).

Equation (2a) now, will generate 2^1 rotations of $N/2$ point DFT coefficient vector set with angular separation $2\pi/(N/2)$.

Similarly all the even r_1 indices in equation (1b) can be written as

$$r_1 \text{ even} = 2r_3$$

where $r_3 = 0, 1, \dots, (N/4 - 1)$. Then the kernel for the even indices in equation (1b) would be given by

$$W_N^{nk} = e^{-j(2\pi/(N/2))n(2r_3)} = e^{-j(2\pi/(N/4))nr_3} \quad (2b)$$

Here, equation (2b) is a kernel of a $N/4$ point DFT. Following in the above manner, decimating the even and odd indices after completing $\log_2 N$ steps (resulting in $\log_2 N$ equations), we will be left with

$$W_N^{nk} = e^{-j(2\pi/(N/N))n0} = e^{-j2\pi n0} \quad \{(\log_2 N)+1\}$$

Since in the final case, $r_f = 0$ only. Therefore, for $n = 0, 1, \dots, (n - 1)$, equation $\{(\log_2 N)+1\}$ will generate N rotations of a 1-point DFT coefficient vector with angular separation $= 0 = 2\pi$.

Hence the Proof.

DIRECT DISTRIBUTED ARITHMETIC COMPUTATION

In this section we present a new direct distributed arithmetic of each DFT point. In this direct distributed arithmetic computation of each transform point, if we shuffle or exchange input data in order to account for the different ordering of DFT coefficients in equal sets the following advantages can be exploited:

- (i) we only need storage area in the order of $(\log_2 N)+1$ instead of in the order of N due to the set of symmetry of DFT.

- (ii) also, by partitioning the distributed arithmetic computation into groups of four (found optimal⁷), then the memory requirements are further reduced.

In carrying out the sum of product for each transform point, N input samples of wordlength b (say) is processed as follows:

$$x(k) = \sum_{n=0}^{N-1} X(n) W_N^{nk} = \sum_{n=0}^{N-1} \sum_{r=0}^{b-1} u(r)x(n,r)(2^{-r}) W_N^{nk}$$

$$\text{or } x(k) = \sum_{r=0}^{b-1} \sum_{n=0}^{N-1} u(r)x(n,r)(2^{-r}) W_N^{nk}$$

and this can written as

$$x(k) = \sum_{r=0}^{b-1} u(r)(2^{-r}) \left\{ \sum_{n=0}^3 x(n,r) W_N^{nk} + \sum_{n=4}^7 x(n,r) W_N^{nk} + \dots + \sum_{n=(N-4)}^{(N-1)} x(n,r) W_N^{nk} \right\} \quad (\text{III})$$

where

$$X(n) = \sum_{r=0}^{b-1} u(r)x(n,r)(2^{-r})$$

$$x(n,r) = \begin{cases} 0 \\ 1 \end{cases}$$

$$u(r) = 1; \forall r \neq 0$$

$$u(0) = -1$$

ADDITIVE AND MULTIPLICATIVE INVERSE FIELD

When each element in a vector set possesses an additive and a multiplicative inverse within the set, so that the sum of all the elements in the set are equal to zero, and the product of all the elements are equal to one, we define such a set to form an additive and multiplicative inverse field.

Each of the $\log_2 N$ different coefficient vector sets form an additive and multiplicative inverse field. Hence the groups of four in the direct distributed arithmetic computation of DFT in equation (III) can be chosen so as to form an additive and multiplicative inverse field. This reduces the memory requirements even further.

MASTER QUAD-SLICE DFT PROCESSOR ARCHITECTURE

The above new approach to DFT computation and the exploitation of the symmetry and periodicity of DFT leads to a new VLSI architecture and design concept named "Master Quad-slice DFT Processor".

Each four product groups summation in equation (III) is the basic computation in this new DFT processor architecture. The hardware slice that performs this basic computation by exploiting the SET SYMMETRY and ADDITIVE & MULTIPLICATIVE INVERSE FIELD properties of DFT is named the Master Quad-slice. Each Quad-slice (Figure 3) contains very simple and regular digital logic with no large RAM/ROM or multiplier, and hence, is much simpler than the butterfly building block and FFT. The quad-slice in Figure 3 has a shuffler for direct distributed arithmetic shuffle/exchange implementation. It has $\log_2 N + 1$ memory stages for storing the fixed coefficients, logic for Sign detection, logic for Zero detection, memory decode logic, a 12-bit adder and registers.

Programming the Master Quad-slice

The Master quad-slice has downward compatibility, in that, a master quad-slice designed for N-point transform can be used to perform $N/2, N/4, N/8, \dots, 1$ point transforms ($N = 2^v, v$ is an integer).

DFT Quad-slice Pipeline

Using multiple Master Quad-slices, the DFT computation shown in equation (III) can be performed in a systolic pipeline arrangement. For an N-point b-wordlength pipeline DFT processor $(N/4) \times b$ Master Quad-slices are required.

Computation Speed

In the DFT pipeline, at every clock period the computation for one transform point (equation III) is complete. The speed of this DFT processor thus depends on the fastest clock that can be employed, and this is only limited by the time for completion of a b-bit addition in each Master Quad-slice (Table 1). This is significant improvement compared to the computation speed achievable by FFT, since in FFT the clocking speed would be limited by the time of a multiplication or a CORDIC¹⁰ rotation. Consequently, the same VLSI technology will provide faster DFT with the Quad-slice processor architecture compared to FFT pipeline processor architecture¹⁻⁶.

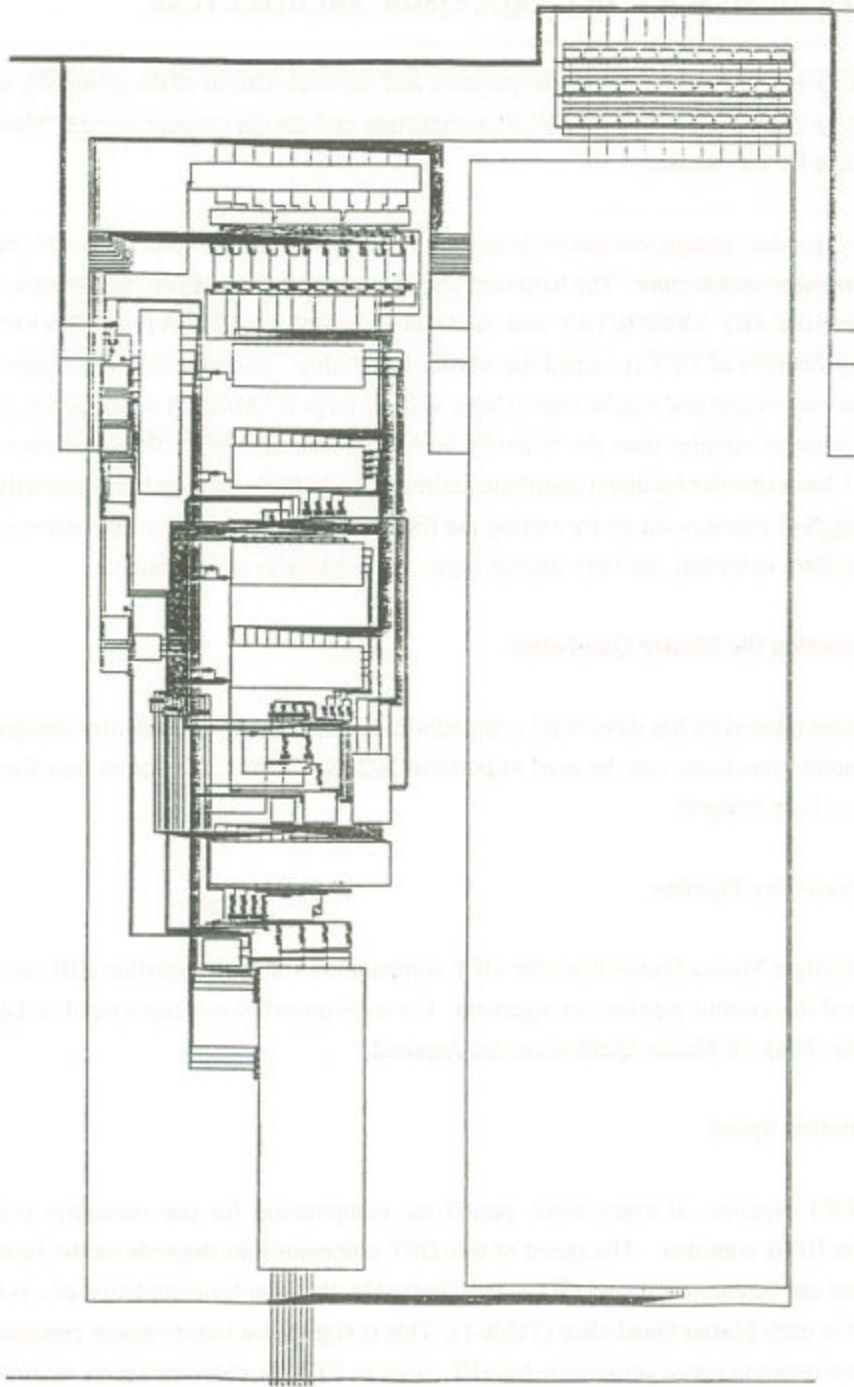
**FIGURE 3. HARDWARE ARCHITECTURE OF A MASTER QUAD-SLICE**

TABLE 1
COMPARISON OF QUAD-SLICE DFT PROCESSOR WITH
FFT PIPELINE PROCESSOR

PARAMETER	QUAD-SLICE DFT	FFT PIPELINE
Speed	Limited only by time of an addition. e.g., 50 ns for a 16-bit addition using 2 micron CMOS technology.	Limited by the time of a multiplication. e.g., 240 ns for a 16-bit multiplication using the same 2 micron CMOS technology.
Testability	More suitable for structured VLSI testability.	Less suitable for structured VLSI testability.
On-Line Fault Detection	Automatic fault detection and fault recovery can be achieved without replicating the whole system.	It is not possible to perform on-line fault detection without replicating the whole system.

Testability and Reliability

The structured and hierarchical nature of the Quad-slice DFT processor and its lower logic complexity makes it more easily tested compared to FFT processor (Table 1). Also, simple reliability analysis shows that the Quad-slice processor has higher reliability than FFT processor for the same amount of applied redundancy.

On-line Testability

The Quad-slice DFT processor can be tested on-line, and, automatic fault detection and fault recovery can be achieved (Table 1). On the other hand, it is not possible to perform on-line fault detection in the case of FFT processor without replicating the whole system.

CONCLUSION

This paper describes a new alternative to fast Fourier transform-based hardware implementation of discrete Fourier transform. A key conclusion is that based on comparison of computational speed, modularity of design, on-line testability and reliability, this architecture is most suitable for VLSI implementation as compared to FFT processor.

REFERENCES

1. Chow, P. et al. A Pipelined Distributed Arithmetic PFFT Processor. *IEEE Trans. on Computers*, 1983, C-32(12), 1128 - 1136.
2. Chowdry, N.U. and Steenaart, W. A High Speed Two-dimensional FFT Processor. In : *Proc. IEEE Speech, Acoustics and Signal Processing Conference*, 1984, pp. 4.11.1 - 4.11.4.
3. Napolitano, L.M. and Robert, G. Efficiency of A New Efficient Algorithm to Compute the Two-dimensional Discrete Fourier Transform. *IEEE Trans. on Signal Processing*, 1992, 40(2), pp. 469 - 470.
4. Valenza, R.J. A Representation - Theoretic Approach to the DFT with Non-commutative Generalization. *IEEE Trans. on Signal Processing*, 1992, 40(4), pp. 814 - 822.
5. Johnson, L.G. Conflict-free Memory Addressing for Dedicated FFT Hardware. *IEEE Trans. on Circuits and Systems II : Analog and Digital Signal Processing*, 1992, 39(5) pp. 312 - 316.
6. Linzer, E.N. and Feig, E. Implementation of Efficient FFT Algorithms on Fused Multiply-add Architectures. *IEEE Trans. on Signal Processing*, 1993, 41(1), pp. 93 - 107.
7. Rezaul Hasan, S.M. *Distributed VLSI Image Processors and Master Quadrant Transformers*, PhD Dissertation, University of California, Los Angeles, 1985.
8. Peled, A. et al. A New Hardware Realization of Digital Filters. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1974, pp. 456 - 462.
9. Samueli, H. Department of Electrical Engineering, U.C.L.A., Los Angeles, California. Private Communications, 1985.
10. Volder, J.E. The CORDIC Trigonometric Computing Technique. *IEEE Trans. on Electronic Computers*, 1954, pp. 330 - 334.