NEW INDEX TRANSFORMS FOR DFT COMPUTATION ON SIGNAL PROCESSORS

J. Hekrdla A. Magyar

Institute of Radio Engineering and Electronics
Czechoslovak Academy of Sciences
Lumumbova 1, 182 51 Prague, Czechoslovakia

On va présenter les transformations de l´indice nouveaux afin d´évaluer la transformation de Fourier descrète /TFD/ d´une ou de deux dimensions de sorte qu´on peut utiliser les algorithmes pour des progressions courtes. L´avantage principal des cettes transformations de l´indice sont les permutations identiques du signal à l´entrée et la sortie. Cela mène à la réduction de la mémoire ROM comme de codage aussi bien de décodage ou de la longueur du programme en cas d´une réalisation en software. Un exemple d´une réalisation sur un processeur du signal de la TFD comprenante de 1386 échantillons est discuté.

The new index transforms are presented for computation of one and two diemnsional discrete Fourier transforms /DFT's/ so as the DFT algorithms of short modules can be used. The identical permutations of samples on the input and the output are the main advantages of the presented index transforms. It leads to the reduction of either coding/decoding ROM memory or program length if software realization is used. An example of the implementation of 1386 point DFT on signal processor is discussed.

1. Introduction

Many modern Discrete Fourier Transform /DFT/ algorithms use a computational scheme in which large one- or two-dimensional input and output sequences are transformed into multidimensional ones so as the DFT algorithms of short modules can be used [1, 2].

In most cases, the so called index transforms or mappings are used for nesting these algorithms. Usefulness of these transforms is based on the following facts:

- One- or two-dimensional DFT can be converted into multidimensional DFT
- Multidimensional DFT can be computed through one-dimensional DFT's of short modules.
- There are very sophisticated algorithms computing DFT's of short modules, see e.g. [3].

For illustration, Fig. 1 shows the computation of 20 point DFT through the computation of two-dimensional 4x5 point DFT.

From Fig. 1 it follows that the mapping of a sequence x we denote as a composition $x \circ f$ where f denotes an index transform which maps a two-dimensional index set into a one-dimensional one according to equation $u = \langle 5v_1+4v_2 \rangle_{20}$, where $\langle a \rangle_b$ denotes residue of a modulo b. Similarly, $g(v_1,v_2) = u \Longleftrightarrow u = \langle 5v_1+16v_2 \rangle_{20}$.

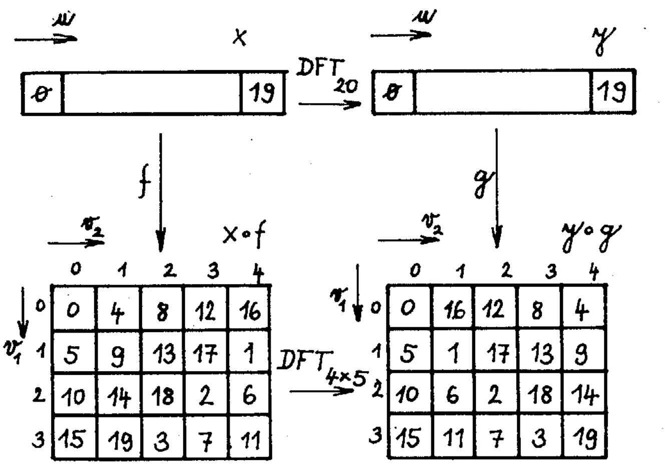


Fig. 1 Computation of 20-point DFT through two-dimensional 4x5 point DFT. Index transforms f, g based on CRT.

The situation in Fig. 1 can be mathematically expressed by the following equation

$$\mathrm{DFT}_I(x \circ f) = \mathrm{DFT}_J(x) \circ g \quad (1)$$

Where I, J denotes the index sets.

It is well known, see e.g. [4] that for the construction of functions f, g the Chinese Remainder Theorem CRT can be used, which was used in Fig. 1 as well. On the example of the two-dimensional index set I and one-dimensional J Burrus in [1] showed, among others, that the class of functions f, g satisfying (1) is wider than the CRT describes. In [5] general equations were derived which describe all the functions satisfying (1) for arbitrary dimensions of I and J. This general description makes it possible to find index transforms satisfying the following additional condition

$$f = g, \qquad (2)$$

which is equivalent to the identical permutations of samples on the input and the output. Fig. 2 demonstrates such a case.

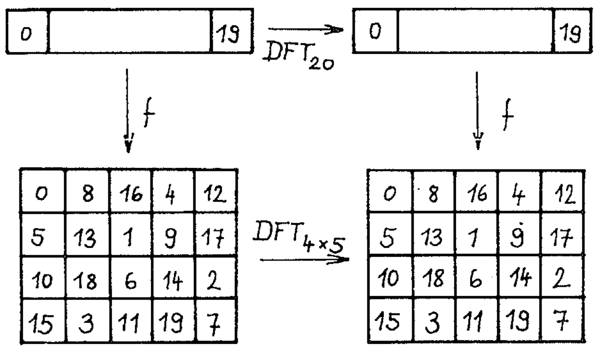


Fig. 2 Computation of 20-point DFT using identical index transforms

This equality results in the reduction of the capacity of a ROM coding memory defining the correspondence between addresses of samples. This reduction is one of the major advantages of using such index transforms. Similar results are not available in CRT indexing scheme.

To find index transforms satisfying (1) and (2) one must solve congruences from [5]. Unfortunately, the general method for solving them does not exist. For practical applications it is necessary to restrict a general problem on special cases. In this contribution we give index transforms satisfying (1) and (2) which map a one- or two-dimensional index set I into a multidimensional index set J. Further, the advantages following from using such transforms on signal processors will be discussed.

In the sequel, index transforms f, g will be denoted by objective symbols $M \rightarrow N_1 x \dots x N_n$ and $MxM \rightarrow N_1 x \dots x N_n$ and, due to equality (2), symbol f will only be used.

2. Index transforms of the type $M \rightarrow N_1 x \dots x N_n$

These index transforms can be described by the following equations

$$f(v_1,\dots,v_n) = u \iff u = \left\langle \sum_{j=1}^{n} \frac{M}{N_j}\, \xi_j v_j \right\rangle_M \qquad (3)$$

where $M = N_1 \dots N_n$ and constants ξ_j satisfy the following quadratic congruences

$$\xi_j^2 \frac{M}{N_j} = 1 (\mathrm{mod}\ N_j)\ ,\ j = 1,\dots,n \qquad (4)$$

If we consider only the known short DFT Winograd approach modules 2, 3, 4, 5, 7, 8, 9, 16 and large modules 11, 13, 17, 19, 15 published in [6] we get 71 such index transforms. In the following Table 1 we give some of them.

Table 1

	$M \rightarrow N_1 x N_2$	ξ_1	ξ_2	ξ_3	ξ_4
m=2	14 2x7	1	2		
	20 4x5	1	2		
	36 4x9	1	4		
	63 7x9	2	2		
m=3	130 2x5x13	1	1	2	
	286 2x11x13	1	5	4	
	495 5x9x11	2	1	1	
m=4	858 2x3x4x13	1	1	1	1
	1386 2x7x9x11	1	5	8	8
	2574 2x9x4x13	1	2	2	3
	4680 5x8x9x13	1	1	2	4

3. Index transforms of the type $MxM \rightarrow N_1 x \dots x N_n$

The existence of such index transforms satisfying (1) and (2) and their results are dependent on the dimension n. We can differ some subcases.

3.1 $MxM \rightarrow N_1 x N_2 x N_3$

In this case there exist index transforms if and only if they can be expressed in the form $\alpha\beta\gamma \times \alpha\beta\gamma \rightarrow \alpha\beta \times \alpha\gamma \times \beta\gamma$ where α, β, γ are mutually prime integers and hence, N_j are composite numbers. Consequently, the use of DFT modules mentioned above would not be simple.

3.2 $MxM \rightarrow N_1 x N_2 x N_3 x N_4$

If we want to get solutions where N_j are not composite, we must suppose the transforms of the form $\alpha\beta \times \alpha\beta \rightarrow \alpha \times \alpha \times \beta \times \beta$ where α, β are mutually prime. Then, the index transforms have the form:

$$f(v_1,v_2,v_3,v_4) = (u_1,u_2) \iff$$

$$u_1 = \left\langle \beta(\xi_{11} v_1 + \xi_{21} v_2) + \alpha(\xi_{31} v_3 + \xi_{41} v_4) \right\rangle_{\alpha\beta}$$

$$u_2 = \left\langle \beta(\xi_{12} v_1 + \xi_{22} v_2) + \alpha(\xi_{32} v_3 + \xi_{42} v_4) \right\rangle_{\alpha\beta}$$

Integers ξ_{ij} satisfy the following system of congruences:

$$\beta(\xi_{11}^2+\xi_{12}^2)=1 \pmod{\alpha} \qquad \beta(\xi_{31}^2+\xi_{32}^2)=1 \pmod{\beta}$$
$$\beta(\xi_{21}^2+\xi_{22}^2)=1 \pmod{\alpha} \qquad \beta(\xi_{41}^2+\xi_{42}^2)=1 \pmod{\beta}$$
$$\xi_{11}\xi_{21}+\xi_{12}\xi_{22}=0 \pmod{\alpha} \qquad \xi_{31}\xi_{41}+\xi_{32}\xi_{42}=0 \pmod{\beta}$$

In the next Table 2 we give a selection of these transforms which use the DFT modules mentioned above.

Table 2

$\alpha\beta\times\alpha\beta\rightarrow\alpha\times\alpha\times\beta\times\beta$	ξ_{11}	ξ_{12}	ξ_{21}	ξ_{22}	ξ_{31}	ξ_{32}	ξ_{41}	ξ_{42}
6x6 → 2x2x3x3	0	1	1	0	1	1	1	2
10x10 → 2x2x5x5	0	1	1	0	2	2	2	3
15x15 → 3x3x5x5	1	1	1	2	1	1	1	4
35x35 → 5x5x7x7	2	2	2	3	1	3	3	6
63x63 → 7x7x9x9	0	2	2	0	0	2	2	0
144x144 → 9x9x16x16	0	2	2	0	0	3	3	0
275x275 → 11x11x25x25	0	2	2	0	0	4	4	0
475x475 → 19x19x25x25	0	4	4	0	0	2	2	0

3.3 $M\times M \rightarrow N_1\times\ldots\times N_n$, n 4

In this case some of the integers N_j are always composite. Consequently, DFT modules mentioned above cannot be simply used.

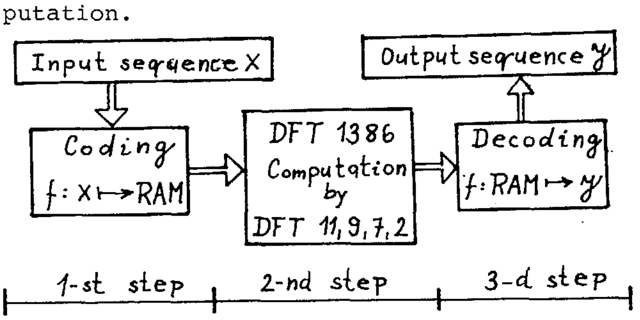
4. Implementation of 1386-points DFT on signal processors

The index transforms proposed in previous chapters have an excellent feature namely the identical coding and decoding scheme. It leads to reduction of either coding/decoding ROM memory or program length if software realization is used. An additional feature of index transforms is their inherent paralelism.

In the following paragraphs we shall introduce an example of 1386-points DFT computation and illustrate the proposed procedure.

4.1 Ingexing scheme and computation steps

Figure 3 shows the three steps of computation.



In the first step the input sequence x is coded to the control processor RAM memory. Then the 1386-points DFT is computed by means of short DFT modules 11, 9, 7 and 2. Finally, the results are decoded to the output sequence Y in natural order.

The coding/decoding scheme is identical and is given by equation

$$u = \left\langle 693v_1+990v_2+1232v_3+1008v_4 \right\rangle_{1386}.$$

This equation is derived by substituting constants ξ_1, ξ_2, ξ_3, ξ_4 from table 1 to equation (3). As a result of coding the input one dimensional sequence x is a four-dimensional sequence $x \circ f$ depicted in Fig. 4.

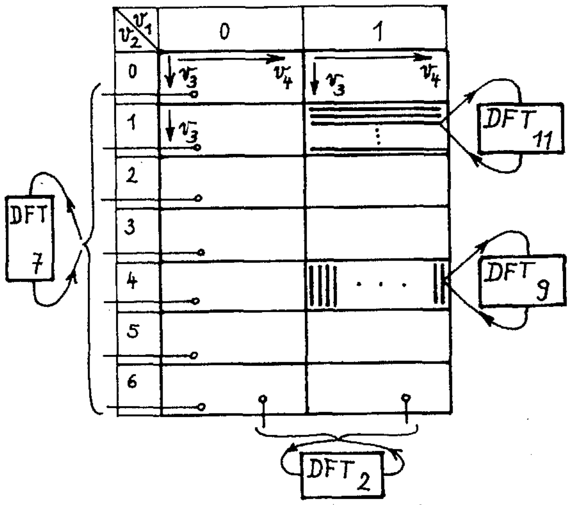


Fig. 4 Mapping by index transform and data reading/writing for short DFT's

After coding the computation of short DFT's can start. The order of their using is arbitrary but each transform must process all samples. From Fig. 4 it follows that DFT 11 will be computed 126 times, DFT 9 154 times, DFT 7 198 times and finaly DFT 2 693 times.

The inherent paralelism of the index transforms makes it possible to use different computation structures. They will be given in the next paragraph.

4.2 Structures

The simplest computation structure shown in Fig. 5 is the serial one, which includes one control processor with enough RAM memory and one signal processor.

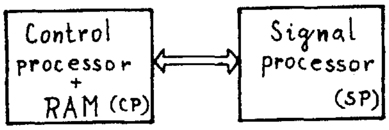


Fig. 5 Serial structure

The signal processor contains programs for computing short DFT's of the length 11, 9, 7 and 2. The computation is carried out by transferring short sequences from CP to SP, reading results from SP and writing them in original places.

This structure can be modified by using several signal processors in parallel. As a result the computation time will be shortened if a fast enough control processor is used.

The previous structure has one disadvantage concering signal processor's program length. Each processor contains a program for all short DFT's and thus the overall program length is large. This problem can be overcome by considering another parallel structure /see Fig. 6/ in which only four signal processors are used.

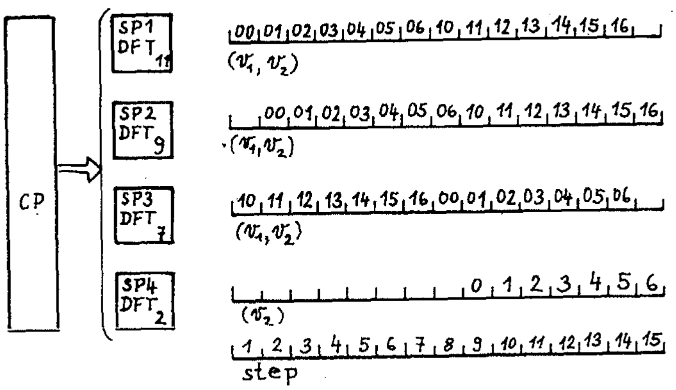


Fig. 6 Parallel structure with 4 signal processors and their timing diagram

Each of them is equipped with a program for computation of one short DFT. As a result signal processors have simple software and they can work simultaneously. Fig. 6 shows one possible organization of the computation.

Processor SP1 starts the computation of DFT 11 in the submatrix $(v_1, v_2) = (0,0)$. After completion the same block is processed with SP2 while SP1 continues with submatrix (0,1). It is worth noting that while SP1 is processing submatrices (0,0)-(0,6) SP2 is computing DFT7 simultaneously on submatrices (1,0)-(1,6). From time interval 9 all processors are working because SP4 is computing DFT 2. The 1386-points DFT is ready after computing all short DFT's. It takes 15 time intervals.

4.3 Signal processor considerations

At present, many types of single chip signal processors are available. They differ in complexity, memory capacity, arithmetic etc. The μPD 7720 from NEC is one of the widely used signal processors. It has modern highly parallel internal architecture, powerful arithmetic unit with hardware multiplier and bus oriented I/O interface. Because of limited RAM memory capacity this processor should be supported by a control processor.

To implement the proposed procedure on signal processors includes the following tasks:

1. design a proper timing
2. write a communication program between CP and SP for data transfer
3. write programs for short DFTs computation with modules 11, 9, 7 and 2.

The first two tasks are rather easy and with the available 8 or 16 bit microprocessors they can be easily realized. The third task consists of writing effective programs for computation of short DFT's. On the effectivness of these programs depends the final computational time.

5. Concluding remarks

In this contribution we apply the theory of index transforms [5] for computation of DFT in the case of one and two-dimensional input seguences. Tables 1 and 2 contain some new results which allow one to design index transforms satisfying the additional condition (2) very easily.

It is well known that the computational scheme considered here is very simple and makes it possible to use very powerful DFT algorithms of Winograd approach.

The main advantages of the presented index transforms follow from the fact that among other index transforms which allow us to use short DFT modules for computation of large DFT's, index transforms presented here save the ROM memory capacity due to the identical permutations of samples on the input and the output.

Modern signal processors have a sufficient capacity of ROM and DROM memory on a chip and then the use of a control processor is not necessary. Further, programs for short DFT algorithms of Winograd type are larger than those for the well known Cooley-Tukey algorithms. For example, the computation of 8-point DFT by Cooley-Tukey algorithm requires about 450 instruction cycles and a program has 50 instructions while 7-point DFT by Winograd approach requires about 250 instruction cycles and a program has about 250 instructions.

References

[1] Burrus, C.S.: Index Mappings for Multidimensional Formulation of the DFT and Convolution, IEEE Trans. ASSP-25, 3, 239-242 1977 .

[2] Nussbaumer, H.J., Quandalle, P.: Fast Computation of Discrete Fourier Transforms Using Polynomial Transforms. IEEE Trans. ASSP-27,2,169-181 1979

[3] Zohar, S.: Winograd's Discrete Fourier Transform Algorithm. Chapter 4 in book Topics in Applied Physics, ed.by T.S.Huang, Springer-Verlag, Berlin, Heidelberg, New York, Vol.43, 88-160, 1981 .

[4] McClellan, J.H., Rader, C.M.: Number Theory in Digital Signal Processing. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1979 .

[5] Hekrdla, J.: Index Transforms of Multidimensional Cyclic Convolutions and Discrete Fourier Transforms IEEE Trans. ASSP-34, 4, 996-997, 1986 .

[6] Johnson, H.W., Burrus, C.S.: Large DFT Modules 11, 13, 17, 19 and 25. Report 8105, Rice University, December 13 1981 .