RADIX-R FFT AND IFFT FACTORIZATIONS WITH EQUAL STAGE-TO-STAGE INTERCONNECTION PATTERN

Pere Marti-Puig
Codessing Hw-Sw Group DTDI-EPS (UVIC), C/de la Laura, 13, 08500 Vic, Barcelona, Spain; pere.marti@uvic.cat

ABSTRACT

Two radix-R regular interconnection pattern families of factorizations for both the FFT and the IFFT -also known as parallel or Pease factorizations- are reformulated and presented in a unified form. Number R is any power of 2 and N, the size of the transform, any power of R. The first radix-2 parallel FFT algorithm -one of the two known radix-2 topologies- was proposed by Pease. Other authors extended the Pease parallel algorithm to different radix and other particular solutions were also reported. The presented families of factorizations for both the FFT and the IFFT are derived from the well-known Cooley-Tukey factorizations, first, for the radix-2 case, and then, for the general radix-R case. Here we present the complete set of parallel algorithms, that is, algorithms with equal interconnection pattern stage-to-stage topology. In this paper the parallel factorizations are derived by using a unified notation based on the use of the Kronecker product and the even-odd permutation matrix to form the rest of permutation matrices. The radix-R generalization is done in a very simple way. It is shown that, both FFT and IFFT share interconnection pattern solutions. This new point of view tries to contribute to the knowledge of fast parallel algorithms for the case of FFT and IFFT but it can be easily applied to other discrete transforms.

1. INTRODUCTION

For the purpose of parallel processing, we require that a process be organized in a set of elementary operations that can be done simultaneously, and there should be as few distinct types of elementary operations as possible. The parallel capability required should also be as simple and regular as possible. In order to take advantage of regularity in FFT/IFFT parallel implementations with equal interconnection pattern stage-to-stage algorithms, the maximum advantages are obtained when an entire stage can be calculated completely in parallel.

A fast transform algorithm can be seen as a sparse factorization of the transform matrix. We refer to each factor as a stage. The matrix dimensions of a stage are the same as the original transform matrix ones. Typically, in each row and each column of a stage there are only R values different to zero and the rest of its elements are exactly equal to zero. Number R is called the radix of the decomposition and usually is a power of two. From this observation, we can see that in a radix-R stage the basic operation consists of computing groups of R outputs from groups of R inputs. As an example, in radix-2 factorizations, this basic operation is called a butterfly. Considering that N is the length of the transform, it is necessary to compute N/2 butterflies to accomplish a stage. In this work we present two general radix-R families for both FFT and IFFT in which R is a power of 2. These algorithms have a regular interconnection pattern between stages and consequently the inputs and outputs for each stage are addressed from or to the same positions, and the factors of the decomposition, the stages, have the property of having their non zero elements in exactly the same positions.

The first regular interconnection pattern for a discrete transform was presented by Pease in [1] for a radix-2 FFT and he refers to it as a parallel algorithm. This solution is the particular case of R=2 for one of the solutions presented in this work. In [2] Sloate presented the basis for a unified theory through which the various versions of the FFT algorithms can be formulated. In [2] the stages are defined by three basic operations: permutation, combination and multiplication. In the same work he presented a particular solution for a radix-4, N=1024, regular interconnection pattern factorization which can also be seen as a particular case in our solutions. In this work we follow the matrix representations for FFT provided by [5][6][7]. The presented approach is reminiscent of the factorization approach recently presented in [9]. Interesting tendencies in the field of fast discrete signal transforms can be found in [8].

In this paper, parallel factorizations of size N, being N a power of R and R a power of 2, are derived from the well-known Cooley-Tukey factorizations. Cooley-Tukey factorizations are also obtained from the basic recursion properties of FFT and IFFT. Factorizations with the same interconnection pattern as FFT are also obtained for the IFFT. This result shows that the same parallel hardware architectures, with the appropriate complex weights feeding the multipliers, can be used to calculate the FFT and the IFFT.

The paper is organized as follows: in section 2 the used notation is presented. In the first part of section 3, the radix-2 Cooley-Tukey factorizations are obtained from the recursion properties of both FFT and IFFT and, in the second part, a method of extending the radix-2 factorizations to radix-R ones is shown. In section 4, taking the radix-R Cooley-Tukey factorizations from section 3, we proceed to derive the two general radix-R regular stage-to-stage algorithms by introducing permutation matrices between factors in a correct
way. In this section it is also shown that the new factorizations exhibit a regular stage-to stage interconnection pattern. Finally some conclusions are presented.

2. USED NOTATION

Since we always deal with square matrices in what follows, an N×N square matrix is denoted by a bold capital letter with subscript N. The number N is a power of two. The elements of matrix \( A_N \) positioned at the row \( m \) and the column \( n \) are denoted by \( a_{mn} \). Sometimes we will use the notation \( A_N^{\pm} = [a_{mn}] \). A column vector is represented by a bold small letter and, since its length can always be known from the context in this paper, its subscript indicates the position of the column in a matrix.

The N×N identity matrix is denoted by \( I_N \) and it can be written by its column vectors \( e_i \), as \( I_N = [e_1; e_2; \ldots; e_N] \).

An even-odd permutation matrix \( P_N \) in terms of vectors \( e_i \) takes the form \( P_N = [e_1; e_3; \ldots; e_{N-1}; e_2; e_4; \ldots; e_N] \). \( P_N \) is often used in this paper since permutation matrices involved in it can be written in terms of \( P_N \).

We will sometimes find it useful to divide a given matrix into sub matrices. Most of the times we will use the Kronecker product to show a particular matrix structure. The symbol \( \otimes \) stands for the right Kronecker product and, for arbitrary sub matrices. Most of the time \( s \) we will use the Kronecker product to show a particular matrix structure. The symbol \( \otimes \) stands for the right Kronecker product and, for arbitrary sub matrices. Most of the time we will use the Kronecker product to show a particular matrix structure. The symbol \( \otimes \) stands for the right Kronecker product and, for arbitrary sub matrices.

As mentioned above, all the permutation matrices can be written in powers of the even-odd permutation matrix \( P \) previously defined. Another possibility, not used here, is to write the same permutation matrices by using the commutation matrix defined in [3] [4].

Next, we recall some useful property involving the Kronecker product and the above defined even-odd permutation matrix \( P_N \).

\[
(A_M \otimes B_N)(C_M \otimes D_N) = A_M C_M \otimes B_N D_N, \tag{1}
\]

\[
P_{2^m} = I_{2^m}, \tag{2}
\]

\[
P_{2^m \otimes 2^n} = P_{2^n}, \tag{3}
\]

\[
I_{2^{m+n}} \otimes I_{2^{m+n+2}} = I_{2^{m+n+2}}. \tag{4}
\]

The Kronecker product of any matrix \( U \) of size \( 2^m \times 2^n \) by any matrix \( V \) of size \( 2^m \times 2^n \) commutes with the powers of the permutation matrices \( P \) as follows [3] [4]:

\[
U_{2^m} \otimes V_{2^n} = P_{2^m \otimes 2^n} (V_{2^n} \otimes U_{2^m})P_{2^m \otimes 2^n}^2. \tag{5}
\]

Finally, the factorization of an arbitrary matrix \( M_N \) in terms of \( n \) factors (or stages) \( E_N(i) \) is written as follows:

\[
M_N = \prod_{i=1}^{n} E_N(i) = E_N(n) \cdots E_N(1). \tag{6}
\]

3. COOLEY-TUKEY FACTORIZATIONS

3.1. Radix-2 Cooley-Tukey factorizations

Suppose that N is a power of 2 and \( j \) denotes the square root of -1. The Fourier transform matrix \( F_N \) is defined as:

\[
F_N = \left\{ e^{2\pi i (-1)^{(n-1)m}} \right\} m, n = 1 : N. \tag{7}
\]

The Inverse Fourier transform matrix \( F_N^H \) -a scale factor is omitted-, is related with the hermitian of \( F_N \). Let us consider the following well-known recursion properties involving matrices \( F_N \) and \( F_{N/2} \):

\[
F_N = B_N (I_2 \otimes F_{N/2}) P_N, \tag{8}
\]

\[
F_N = P_N^T (I_2 \otimes F_{N/2}) B_N^T, \tag{9}
\]

\[
F_N^H = P_N^T (I_2 \otimes F_{N/2}^H) B_N^H, \tag{10}
\]

\[
F_N^H = B_N^* (I_2 \otimes F_{N/2}^H) P_N. \tag{11}
\]

Matrix \( B \) is defined using the identity matrix \( I \) and the diagonal matrix \( A \) as:

\[
B_{2^m} = \begin{bmatrix} I_{2^{m-1}} & A_{2^{m-1}} \\ I_{2^{m-1}} & -A_{2^{m-1}} \end{bmatrix}, \tag{12}
\]

where the diagonal matrix \( A \) is:

\[
A_N = diag\left\{ e^{\frac{-2\pi i}{N}(i-1)} \right\} i = 1 : N. \tag{13}
\]

In expressions (9-11) upper index \( T, H \) and * denote the transpose, hermitian and complex conjugate respectively. The well-known radix-2 Cooley-Tukey factorizations can be obtained from (8-11) when these recursions are iterated. They can be written in the presented notation taking into account that the stop criterion of the recursive process is:

\[
F_2 = F_2^H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{14}
\]
The factorizations obtained are very simple hardware made reordering. Note we have \( n = \log_2 N \) factors or radix-2 stages. The factorizations in (27) have regular interconnection pattern stage-to-stage as shown above. They can be computed using the form:

\[
R_N^{-1} F_N^H = \prod_{i=1}^{E} \prod_{f=1}^{F} \left( I_{2^{E(i)-1} \times r \times f} \otimes B_{2^{E(i)-1} \times r \times f}^H \right)
\]  

(24)

\[
F_N R_N^{-1} = \prod_{i=1}^{E} \prod_{f=1}^{F} \left( I_{2^{E(i)-1} \times r \times f} \otimes B_{2^{E(i)-1} \times r \times f} \right)
\]  

(25)

Expressions (22-25) can be operated in order to find more compact expressions but for our purpose it is not necessary. If we see (22) we realize that the family of solutions provided for the radix-R factors \( E(i) \), where \( i \) goes from 1 to \( E \), take the form:

\[
E_N(i) = \prod_{f=1}^{F} \left( I_{2^{E(i)-1} \times r \times f} \otimes B_{2^{E(i)-1} \times r \times f} \right)
\]  

(26)

Note also that the matrices \( E_N(i) \) are sparse matrices with \( R \) non-zero elements in each row and each column. In a similar way we can obtain the radix-R stages \( E(i) \) for (23) (24) and (25).

4. GENERAL RADIX-R PEASE FACTORIZATIONS

We can obtain the radix-R equal interconnection pattern factorizations starting from expressions (22-25) with the introduction of the appropriate permutation matrices between stages in order to change interconnection patterns without changing the result of the full factorization. As the derivation method is exactly the same for the families obtained from (22) and (25) and for the families obtained from (23) and (24) only the first two are shown.

Let us begin with (21). If we introduce the powers of permutation matrices \( P \) in the following way we do not change the final result because the permutation matrices introduced between them are always the permutation matrix and its inverse. See below:

\[
F_N R_N^{-1} = \prod_{i=1}^{E} \prod_{f=1}^{F} \left( I_{2^{E(i)-1} \times r \times f} \otimes B_{2^{E(i)-1} \times r \times f} \right)
\]  

(27)

It is interesting to make an approximation to (27) for the radix-2 case when \( F = 1 \) and \( E = \log_2 N \). In all cases, note that for \( i = 1 \) and \( f = 1 \) the first stages are obtained by post-multiplying with the identity matrix \( I \). For \( i = E \) and \( f = F \) the last stages are also obtained by pre-multiplying by the identity matrix \( I \) as \( EF = \log_2 N \). So

\[
P_2^n = P_{2^n}^E = P_{2^n}^F = I_2^n.
\]

(28)

The factorizations in (27) have regular interconnection pattern stage-to-stage as shown above. They can be
simplified using the Kronecker product property (5) that allows the following equality:

\[ \mathbf{I}_{2^{2r-i}r-f} \otimes \mathbf{B}_{2^{2r}} \rightarrow \mathbf{P}_N^{EF-(i-1)F-f} \left( \mathbf{B}_{2^{2r-i}r-1} \otimes \mathbf{I}_{2^{2r-i}r-f} \right) \mathbf{P}_N^{(i-1)F+f} \]  (29)

By combining (29) and (27), with properties (2-4) we have:

\[ \mathbf{F}_N \mathbf{R}_N^{-1} = \prod_{i=1}^{E} \prod_{f=1}^{F} \left( \mathbf{B}_{2^{2r-i}r-1} \otimes \mathbf{I}_{2^{2r-i}r-f} \right) \mathbf{P}_N. \]  (30)

And the result is:

\[ \mathbf{E}(i) = \prod_{i=1}^{E} \prod_{f=1}^{F} \left( \mathbf{B}_{2^{2r-i}r-1} \otimes \mathbf{I}_{2^{2r-i}r-f} \right) \mathbf{P}_N. \]  (31)

Let us continue with the family obtained from (22). Now to get solutions with regular interconnection pattern stage-to-stage we have to introduce the powers of matrix \( \mathbf{P} \) as follows:

\[ \mathbf{R}_N^{-1} \mathbf{F}_N = \prod_{i=1}^{E} \prod_{f=1}^{F} \mathbf{P}_N^{-(i-1)F-f} \left( \mathbf{B}_{2^{2r-i}r-1} \otimes \mathbf{B}_T^{T_{2r-i}r-1} \right) \mathbf{P}_N^{(i-1)F+f-1}. \]  (32)

This operation doesn’t change the result of \( \mathbf{R}_N^{-1} \mathbf{F}_N \) because, as in the previous case, the two matrices introduced between radix-2 factors are always a permutation matrix and just its inverse. Using the property (5) we have:

\[ \mathbf{I}_{2^{2r-i}r-f} \otimes \mathbf{B}_T^{T_{2r-i}r-1} = \mathbf{P}_N^{-(i-1)F-f} \left( \mathbf{B}_T^{T_{2r-i}r-1} \otimes \mathbf{I}_{2^{2r-i}r-1} \right) \mathbf{P}_N^{(i-1)F+f-1}. \]  (33)

and as \( \mathbf{P}_N^{-1} = \mathbf{P}_N^T \) and \( \mathbf{P}_N^{EF} = \mathbf{I}_N \), expression (32) simplifies to:

\[ \mathbf{R}_N^{-1} \mathbf{F}_N = \prod_{i=1}^{E} \prod_{f=1}^{F} \mathbf{P}_N \left( \mathbf{B}_T^{T_{2r-i}r-1} \otimes \mathbf{I}_{2^{2r-i}r-f} \right). \]  (34)

That is the other result we are seeking.

\[ \mathbf{E}'(i) = \prod_{f=1}^{F} \mathbf{P}_N \left( \mathbf{B}_T^{T_{2r-i}r-1} \otimes \mathbf{I}_{2^{2r-i}r-f} \right). \]  (35)

The factorizations given by (30) and (34) have the property of having the same interconnection pattern stage-to-stage. A way to show that the stages in (31) and (35) have an identical interconnection pattern stage-to-stage is based on replacing matrix \( \mathbf{B} \) by another simpler matrix \( \mathbf{B}' \) having its non-zero elements in the same positions since the interconnection pattern is given by the position of the non-zero elements in each sparse matrix. To form \( \mathbf{B}' \), we replace in (12) the diagonal matrix \( \mathbf{A} \) by the identity matrix \( \mathbf{I} \) in the following way:

\[ \mathbf{B}'_N = \begin{bmatrix} \mathbf{I}_{2^{1}r-1} & \mathbf{I}_{2^{1}r-1} \\ \mathbf{I}_{2^{1}r-1} & -\mathbf{I}_{2^{1}r-1} \end{bmatrix} = \mathbf{F}_2 \otimes \mathbf{I}_{2^{1}r}. \]  (36)

As an example, if we replace \( \mathbf{B} \) by \( \mathbf{B}' \) in the factors in (31) we will show, using (2-4), that the modified stages are independent of \( i \). This is:

\[ \left( \mathbf{B}'_N \otimes \mathbf{I}_{2^{2-i}r-1} \right) \mathbf{P}_{2^{2-i}r} = \left( \mathbf{F}_2 \otimes \mathbf{I}_{2^{2-i}r-1} \right) \mathbf{P}_{2^{2-i}r}. \]  (37)

In a similar way, we can obtain the same kind of factorizations for the IFFT. Then for IFFT (except for a constant) we have the first family of solutions as:

\[ \mathbf{R}_N^{-1} \mathbf{F}_N^H = \prod_{i=1}^{E} \prod_{f=1}^{F} \mathbf{P}_N^T \left( \mathbf{B}_N^H \otimes \mathbf{I}_{2^{2r-i}r-f} \right) \mathbf{P}_N. \]  (38)

\[ \mathbf{E}'^H(i) = \prod_{f=1}^{F} \mathbf{P}_N^T \left( \mathbf{B}_N^H \otimes \mathbf{I}_{2^{2r-i}r-f} \right) \mathbf{P}_N. \]  (39)

and the second family:

\[ \mathbf{F}_N^H \mathbf{R}_N^{-1} = \prod_{i=1}^{E} \prod_{f=1}^{F} \left( \mathbf{B}_N^H \otimes \mathbf{I}_{2^{2r-i}r-f} \right) \mathbf{P}_N. \]  (40)

\[ \mathbf{E}'^H(i) = \prod_{f=1}^{F} \left( \mathbf{B}_N^H \otimes \mathbf{I}_{2^{2r-i}r-f} \right) \mathbf{P}_N. \]  (41)

As from the indices \( m, n \) of the non-zero elements \( e_{mn} \) in each sparse matrix representing a stage, we can observe that the \( n \) input element is needed to calculate the \( m \) output element in the \( i \)-th stage.
Another way to see that the stages defined in (31), (35), (39) and (41) have the same interconnection pattern for large values of N is by representing each sparse matrix of a given factorization as an image in which the zero elements are represented by one colour and the non-zero elements with another colour. In the case of equal stage-to-stage factorizations, once a factorization is given, all the images representing a factor are equal.

In Fig. 1, the equal interconnection pattern stage-to-stage factorizations for an input vector \( x \) to an output vector \( y \) when \( N=16 \) and \( R=4 \) is represented. In Fig. 2 the same representation is done for \( N=16 \) and \( R=4 \). It is interesting to note that factorizations given by (30) for FFT and by (40) for IFFT share the architecture A), and factorizations given by (34) for FFT and (38) for IFFT share the architecture B).

5. CONCLUSIONS

This article shows a particular method to derive the different FFT/IFFT topologies with equal interconnection stage-to-stage pattern for the general radix-R case, R being a power of 2 and N a power of R. Four general radix-R factorizations for any length N (where N is a power of 2) that exhibit a regular interconnection pattern between stages is presented. Two families of factorizations are reported for the FFT and another two for the IFFT. It is interesting to note that we derive two families of architectures but a particular architecture can be used to calculate both FFT and IFFT transforms because there are factorizations for FFT and IFFT that share the same interconnection pattern. As different discrete transforms have factorizations with a Cooley-Tukey type stage-to-stage interconnection, the same argument can easily be extended to them. The derivation method is based on the introduction of the even-odd permutation matrices with the appropriated powers between the stages. It will be also interesting to extend these kinds of factorizations, using the Kronecker product properties to the two dimensional case. The use of the even-odd permutation matrices to write all other permutation matrices involved in the derivation process and the presentation of the different solutions in a unified manner offers a new point of view of parallel FFT/IFFT algorithms. The present work tries to contribute to the understanding of fast parallel algorithms. Modern FFT architectures tend to optimize the number of operations and tend to implement much larger radices combined with other methods to manage efficiently the data in memory [10]. Our approach offers very regular and efficient memory managing and the possibility of implementing high radices.

6. ACKNOWLEDGEMENT

This work is supported in part by the cost centre R008-R0904 from the University of Vic.

7. REFERENCES