NEW FFT/IFFT FACTORIZATIONS WITH REGULAR INTERCONNECTION PATTERN STAGE-TO-STAGE SUBBLOCKS

PERE MARTI-PUIG

Grup de Codisseny Hardware-Software, Departament de Tecnologies Digitals i de la Informació ,
Universitat de Vic, Carrer de la Sagrada Família, 7- 08500 Vic, e-mail: pere.marti@uvic.cat

Data de recepció: 14/02/08
Data de publicació: 02/04/08

RESUM

Les factoritzacions de la FFT (Fast Fourier Transform) que presenten un patró d'interconnexió regular entre factors o etapes son conegudes com algorismes paral·lels, o algorismes de Pease, ja que foren originalment proposats per Pease. En aquesta contribució s'han desenvolupat noves factoritzacions amb blocs que presenten el patró d'interconnexió regular de Pease. S'ha mostrat com aquests blocs poden ser obtinguts a una escala prèviament seleccionada. Les noves factoritzacions per ambdues FFT i IFFT (Inverse FFT) tenen dues classes de factors: uns pocs factors del tipus Cooley-Tukey i els nous factors que proporcionen la mateix patró d'interconnexió de Pease en blocs. Per a una factorització donada, els blocs comparteixen dimensions, el patró d'interconnexió etapa a etapa i a més cada un d'ells pot ser calculat independentment dels altres.

ABSTRACT

FFT (Fast Fourier Transform) factorizations presenting a regular interconnection pattern between factors or stages are known as parallel algorithms, or Pease algorithms since were first proposed by Pease. In this paper, new FFT/IFFT (Inverse FFT) factorizations with blocks that exhibit regular Pease interconnection pattern are derived. It is shown these blocks can be obtained at a previously selected scale. The new factorizations for both the FFT and IFFT have two kinds of factors: a few Cooley-Tukey type factors and new factors providing the same Pease interconnection pattern property in blocks. For a given factorization, these blocks share dimensions, the interconnection pattern stage-to-stage, and all of them can be calculated independently from one another.

RESUMEN

Las factoritzaciones de la FFT (Fast Fourier Transform) que presentan un patrón de interconexiones regular entre factores o etapas son conocidas como algoritmos paralelos, o algoritmos de Pease, puesto que fueron originalmente propuestos por Pease. En esta contribución se han desarrollado nuevas factoritzaciones en subbloques que presentan el patrón de interconexión regular de Pease. Se ha mostrado como estos bloques pueden ser obtenidos a una escalera previamente seleccionada. Las nuevas factoritzaciones para ambas FFT y IFFT (Inverse FFT) tienen dos clases de factores: unos pocos factores del tipo Cooley-Tukey y los nuevos factores que proporcionan el mismo patrón de interconexión de Pease en bloques. Para una factorización dada, los bloques comparten dimensiones, patrón d'interconexión etapa a etapa y además cada uno de ellos puede ser calculado independientemente de los otros.

1. INTRODUCTION

The discrete Fast Fourier Transform, FFT, was first discovered by Gauss (see, e.g., [1]) and rediscovered by Cooley and Tukey [2] in the 1960s. It is very important in engineering and therefore many algorithms have been derived from the 1960s on, and there is a very extensive bibliography on the subject. There are algorithms referred to as higher radix [3] [4], mixed-radix [5], prime-factor [6], Winograd [7], split-radix [8] [9], identical geometry from stage-to-stage FFT [12], recursive [10], the combination of decimation-in-time and the decimation-in-frequency [11], among others. Reference [13] provides an interesting overview on the state of the art of FFT. Matrix representations for FFT provided by [14], [15], [16], [17] and new tendencies in the field of fast discrete signal transforms are reported in [18]. More recently, muticarrier modulations transceivers involving Fast Fourier Transform calculations have inspired new research in FFT architectures [4], [19], [20], [21]. Today it seems improbable that big implementation advantages can be reached by developing new algorithms with a smaller computational complexity than the algorithms in [23], [24], [25], [26] which are developed to be implemented by software. The hardware FFT implementation and the specific FFT architectures can still be of interest due to technological advances. As an example, more sophisticated Digital Signal Processing-oriented FPGAs devices (Field Programmable Gate Arrays) provide hundreds of real embedded multiplier elements that can operate at clock speeds of hundreds of MHz. Therefore an important part of the computation can be done in parallel. An algorithm designer has not only the possibility of having a lot of hardware resources allowing parallel implementations, but also the option of combining hardware-software solutions using a FPGA with a digital signal processor working together or with a (hard- or soft-) processor core inside the FPGA. A practical FFT/IFFT (Inverse FFT) implementation on FPGA in [27] has motivated the need of exploring new factorizations that can guide FFT/IFFT implementation with different level of parallelism.

For the purpose of parallel processing, we require that the process be organized in a set of elementary operations that can be done simultaneously. There should be as few distinct types of elementary operations as possible. The parallel capability required shall be as simple and regular as possible. Local equal interconnection pattern properties at different lower scales provide this simplicity, especially when the scale in which the subblocks exhibit Pease property matches to the parallel hardware resources [12]. The factorizations presented in this paper open the possibility of exploring these new architectures.

A fast transform algorithm can be understood as a sparse factorization of the transform matrix. Each sparse matrix representing a factor in the FFT factorizations is called a stage. Matrix dimensions of a stage are the same as those of the original transform matrix. Typically, each row and each column of a stage contain only R values different from zero. The number R is called the radix of the decomposition and is usually a power of two. We can see from this observation that in a radix-R stage the basic operation consists in computing groups of R outputs from groups of R inputs. When R is equal to 2, that is, in radix-2 factorizations, the basic operation is called a butterfly. Therefore in this case, assuming that N is the length of the transform, one should compute N/2 butterflies to complete a stage.

The interconnection pattern is a stage-to-stage relation between positions of the input data elements and the output data elements. In the matrix representing a factor, the interconnection pattern is given by the indices m,n of its non-zero elements, a_{mn}, meaning that the n-th input element is required to calculate the m-th output element at this stage. Following the matrix point of view, Pease factorizations have the particularity that each factor -or stage- addresses their inputs and their outputs from or to the same positions. Therefore the factors have the non-zero entries exactly at the same matrix locations. The regular input and output flow of data stage-to-stage provided by the Pease factorization suggests a very simple and very fast parallel architecture, especially when all resources required to compute a stage in parallel can be mapped onto hardware [12]. Then, if we have all hardware computing a stage in parallel, it is important to appreciate that a regular interconnection pattern lets each output 'wired in' to the correspondent input position. Computing the FFT is therefore reduced to the computation of $\log_R N$ stages in a very simple iterative process. When a particular stage *i* is completed, the output set of the data is feedbacked in parallel to the input, in order to calculate the stage *i+1*, always in the same way.

Following the example of a radix-2 FFT of length N, mapping a parallel hardware to compute a stage means to map N/2 butterflies with the registers to store data and the buses to interconnect them. We can guess the hardware cost by taking into account that an optimized radix-2 butterfly needs one complex multiplier and two complex adders only [27]. When mapping onto hardware of all resources to compute a complete stage in parallel is impossible for any reason or its efficiency is low, each stage has to be calculated sequentially in several steps. Under the assumption that hardware resources can only calculate one step in parallel, even in Pease architectures [27], since these steps have different data input-output patterns, it is necessary to map additional memory into hardware to save partial results and additional hardware resources with control functions. This is clear because some outputs in the computation of step n of stage *i*, should be feedbacked to the input registers to calculate stage *i+1*, but these registers cannot be used since they could contain valid input data of unfinished steps of the current stage *i*. Additional memory and control hardware means more area resources. A more sophisticated process needs to be organized in more clock cycles and, especially if they are combined with extra memory access, it means more power consumption.

In order to preserve advantages that a regular interconnection pattern offers, we explore factorizations that reproduce the same pattern at the subblock scale, with the idea of better adapting to the hardware capability of parallel computing. The new factorizations we propose have two kinds of factors: a few Cooley-Tukey type factors and new factors that provide the same Pease interconnection pattern property in subblocks. We can find different strategies for the FFT/IFFT computation in [28], [29], [30] and [31]. Our factorizations provide a new approach to the existing strategies that can take advantage when the FFT/IFFT implementations can compute a part of a stage in parallel, then the regularity of a parallel processing in found at the subblock level. The presented factorizations can also be particularized to obtain different subblock sizes. All subblocks share the same interconnection pattern stage-to-stage and can be calculated independently from the others. In the hardware-software partition process of a design in which a FFT algorithm appears, it seems clear that the subblocks with a regular stage-to-stage interconnection pattern could be implemented in hardware. It is interesting to observe that the regular interconnection pattern of a subblock of size N is the same as the Pease architecture of a FFT of size N [12]. Therefore the hardware that computes a subblock, using theses factorizations, can be used to compute FFTs of length N, 2N, 4N, etc. Only the multiplier coefficients should to be updated.

2. NOTATIONS AND RADIX-2 FFT COOLEY-TUKEY FACTORIZATIONS

The notation we use and the well-known radix-2 Cooley-Tukey factorizations [2] that will be the starting point of our argument are presented in this section. 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 entry of matrix $\mathbf{A}_N$ located at the row m, column n, is denoted by a_{mn}. We will sometimes use the notation $\mathbf{A}_N=\{a_{mn}\}$. A column vector is represented by a small bold letter. Since the length of a column vector is always clear from the context, the subscript will indicate in this case the position of the column in a matrix. The $N\times N$ identity matrix is denoted by $\mathbf{I}_N$ and it can be written by its column vectors $\mathbf{e}_i$ as $\mathbf{I}_N = [\mathbf{e}_1\ \mathbf{e}_2 \cdots \mathbf{e}_n]$. An even-odd permutation matrix $\mathbf{P}_N$ in terms of vectors $\mathbf{e}_i$ takes the form $\mathbf{P}_N = [\mathbf{e}_1\ \mathbf{e}_3 \cdots \mathbf{e}_{n-1}\ \mathbf{e}_2\ \mathbf{e}_4 \cdots \mathbf{e}_n]$. We will often use it in this paper since permutation matrices involved in it can be written using $\mathbf{P}_N$. We will sometimes find it useful to divide a given matrix into submatrices. 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 square matrices $\mathbf{A}_M$ and $\mathbf{B}_N$, the Kronecker product $\mathbf{A}_M$

$\otimes$ $\mathbf{B}_N$ is an $MN\times MN$ matrix that can be written using the elements a_{mn} of matrix $\mathbf{A}_M$ as:

$$\mathbf{A}_M \otimes \mathbf{B}_N = \begin{bmatrix} a_{11}\mathbf{B}_N & \cdots & a_{1M}\mathbf{B}_N \\ \cdots & & \cdots \\ a_{M1}\mathbf{B}_N & \cdots & a_{MM}\mathbf{B}_N \end{bmatrix}. \tag{1}$$

Next, we recall some useful properties involving the Kronecker product and the above defined even-odd permutation matrix $\mathbf{P}_N$. We have:

$$\mathbf{I}_{2^{n_1}} \otimes \mathbf{I}_{2^{n_2}} = \mathbf{I}_{2^{n_1+n_2}}, \tag{2}$$

$$(\mathbf{A}_M \otimes \mathbf{B}_N)(\mathbf{C}_M \otimes \mathbf{D}_N) = \mathbf{A}_M\mathbf{C}_M \otimes \mathbf{B}_N\mathbf{D}_N, \tag{3}$$

$$\mathbf{A}_{2^{n_1}} \otimes \mathbf{B}_{2^{n_2}} = \mathbf{P}^{n_1}_{2^{n_1+n_2}}\left(\mathbf{B}_{2^{n_2}} \otimes \mathbf{A}_{2^{n_1}}\right)\mathbf{P}^{n_2}_{2^{n_1+n_2}}, \tag{4}$$

$$\mathbf{P}^{n}_{2^n} = \mathbf{I}_{2^n}, \tag{5}$$

$$\mathbf{P}^{n+n_1}_{2^n} = \mathbf{P}^{n_1}_{2^n}. \tag{6}$$

Note that superscript n in a matrix means the power n of this matrix. Finally, the factorization of an arbitrary matrix $\mathbf{M}_N$ in terms of n factors (or stages) $\mathbf{E}_N(i)$ is written as follows:

$$\mathbf{M}_N = \prod_{i=1}^{n} \mathbf{E}_N(i) = \mathbf{E}_N(n)\cdots\mathbf{E}_N(2)\mathbf{E}_N(1). \tag{7}$$

2.1 RADIX-2 FFT COOLEY-TUKEY FACTORIZATIONS

Suppose that $N=2^n$, that is, N is a power of 2, and j denotes the square root of -1. The Fourier transform matrix $\mathbf{F}_N$ is defined as:

$$\mathbf{F}_N = \left\{ e^{-j\frac{2\pi}{N}(p-1)(q-1)} \right\} \qquad p,q = 1:N. \tag{8}$$

The Inverse Fourier transform matrix is related with the Hermitian of $\mathbf{F}_N$ as $\mathbf{F}_N = (1/N)\,\mathbf{F}_N^{H}$.
In this section we will rewrite the radix-2 Cooley-Tukey factorizations originally presented in [2] by using the Kronecker product notation used in modern algorithm design as in [14][15].
Let $\mathbf{B}_{2^i}$ denote the matrix defined by:

$$\mathbf{B}_{2^n} = \begin{bmatrix} \mathbf{I}_{2^{n-1}} & \mathbf{A}_{2^{n-1}} \\ \mathbf{I}_{2^{n-1}} & -\mathbf{A}_{2^{n-1}} \end{bmatrix}, \tag{9}$$

where:

$$\mathbf{A}_N = diag\left\{ e^{-j\frac{2\pi}{N}(n-1)} \right\} \qquad n = 1:N. \tag{10}$$

Consider the following well-known recursive properties involving matrices $\mathbf{F}_N$ and $\mathbf{F}_{N/2}$:

$$\mathbf{F}_N = \mathbf{B}_N \left(\mathbf{I}_2 \otimes \mathbf{F}_{N/2}\right)\mathbf{P}_N, \tag{11}$$
$$\mathbf{F}_N = \mathbf{P}_N{}^T \left(\mathbf{I}_2 \otimes \mathbf{F}_{N/2}\right)\mathbf{B}_N{}^T. \tag{12}$$

and their Hermitian representations:

$$\mathbf{F}_N{}^H = \mathbf{P}_N{}^T \left(\mathbf{I}_2 \otimes \mathbf{F}_{N/2}{}^H\right)\mathbf{B}_N{}^H, \tag{13}$$
$$\mathbf{F}_N{}^H = \mathbf{B}_N{}^* \left(\mathbf{I}_2 \otimes \mathbf{F}_{N/2}{}^H\right)\mathbf{P}_N. \tag{14}$$

In (11-14), $\mathbf{I}_2$ is the 2×2 identity matrix, $\mathbf{P}_N$ is above defined even-odd permutation matrix and the superscripts T, H and * denote transposition, the Hermitian conjugate and the complex conjugate respectively.

The classical FFT/IFFT radix-2 Cooley-Tukey factorizations can be obtained by iterating expressions (11-14) and taking into account that the criterion for stopping the recursive process is:

$$\mathbf{F}_2 = \mathbf{F}_2^H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{15}$$

Then from the recursion (11) and some algebra, we obtain:

$$\mathbf{F}_N = \prod_{i=1}^{n}\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{B}_{2^i}\right)\prod_{i=1}^{n}\left(\mathbf{I}_{2^{i-1}} \otimes \mathbf{P}_{2^{n-i+1}}\right) = \prod_{i=1}^{n}\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{B}_{2^i}\right)\mathbf{R}_N. \tag{16}$$

And from (12) and some algebra:

$$\mathbf{F}_N = \prod_{i=1}^{n}\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{P}_{2^i}^T\right)\prod_{i=1}^{n}\left(\mathbf{I}_{2^{i-1}} \otimes \mathbf{B}_{2^{n-i+1}}^T\right) = \mathbf{R}_N \prod_{i=1}^{n}\left(\mathbf{I}_{2^{i-1}} \otimes \mathbf{B}_{2^{n-i+1}}^T\right). \tag{17}$$

Notice that the permutation matrix $\mathbf{R}_N$ known as the bit-reversal permutation matrix, appears in (16) and in (17) written in two different ways. That is:

$$\mathbf{R}_N = \prod_{i=1}^{n}\left(\mathbf{I}_{2^{i-1}} \otimes \mathbf{P}_{2^{n-i+1}}\right) = \prod_{i=1}^{n}\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{P}_{2^i}{}^T\right). \tag{18}$$

Since the matrix $\mathbf{R}_N$ is well-known to be equal to its inverse, we have $\mathbf{R}_N = \mathbf{R}_N^{-1} = \mathbf{R}_N^H = \mathbf{R}_N^T = \mathbf{R}_N^*$. Therefore we will deal only with $\mathbf{R}_N$ in what follows. Computing the transform $\mathbf{R}_N$ only means a very easy hardware-made reordering. Note we have $n = \log_2 N$ factors or radix-2 stages. The factorizations obtained from (11-14) for the FFT can be rewritten in the following manner in order to present them as a product of n radix-2 stages:

$$\mathbf{F}_N\mathbf{R}_N = \prod_{i=1}^{n}\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{B}_{2^i}\right), \tag{19}$$

$$\mathbf{R}_N\mathbf{F}_N = \prod_{i=1}^{n}\left(\mathbf{I}_{2^{i-1}} \otimes \mathbf{B}^T{}_{2^{n-i+1}}\right). \tag{20}$$

In a similar way from (13-14) the IFFT factorizations could be obtained and its expressions are:

$$\mathbf{R}_N\mathbf{F}_N^H = \prod_{i=1}^{n}\left(\mathbf{I}_{2^{i-1}} \otimes \mathbf{B}^H_{2^{n-i+1}}\right), \tag{21}$$

$$\mathbf{F}_N^H\mathbf{R}_N = \prod_{i=1}^{n}\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{B}^*_{2^i}\right). \tag{22}$$

The interconnection pattern between the stages is represented in Figure 1 for Cooley-Tukey factorizations. Observe that FFT factorizations in (19) and IFFT factorizations in (22) share the same interconnection pattern architecture in the same way that FFT factorizations in (20) and IFFT factorizations in (21). This can be interesting in IFFT/FFT hardware implementations. For example, in OFDM based communications both algorithms are used for modulation and demodulation, respectively, and this can be done with the same architecture.

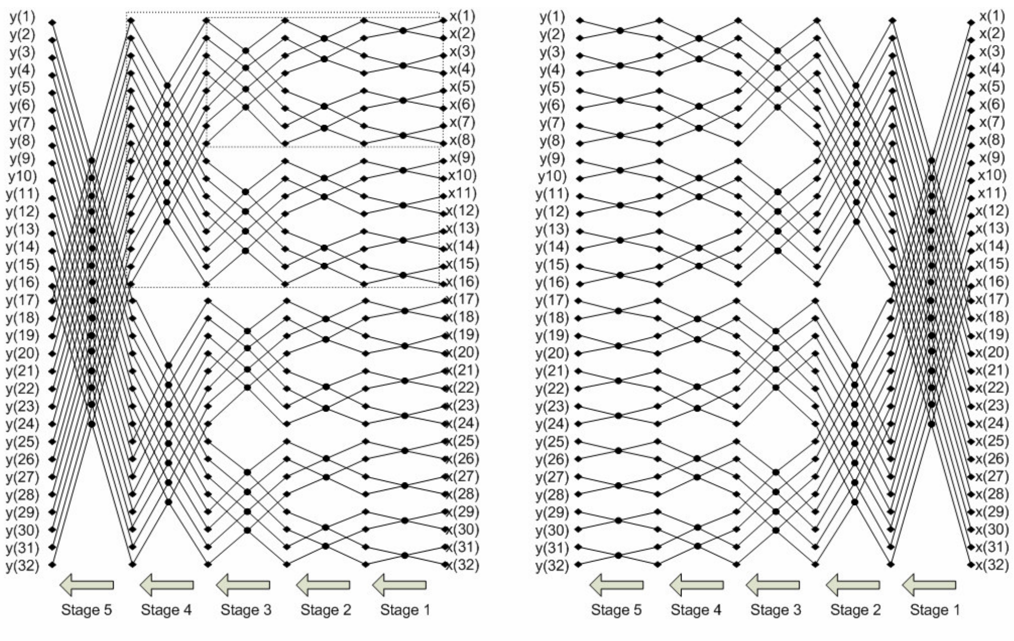

Figure 1 A 32-point radix-2 Cooley-Tukey stage interconnection pattern given by: *A)* expressions (19) and (22) *B)* expressions (20) and (21).

2.2 RADIX-R COOLEY-TUKEY FACTORS AS A PRODUCT OF RADIX-2 FACTORS

For any transform matrix $\mathbf{F}_N$ such that $N=R^E$ and $R=2^F$, one can readily find radix-R factorizations by observing that radix-R stages can be written by F products of consecutive radix-2 factors. Therefore, the radix R equivalent factorizations to expressions (19) and (20) take the form:

$$\mathbf{F}_N\mathbf{R}_N = \prod_{i=1}^{E}\prod_{f=1}^{F}\left(\mathbf{I}_{2^{EF-(i-1)F-f}} \otimes \mathbf{B}_{2^{(i-1)F+f}}\right), \tag{23}$$

$$\mathbf{R}_N\mathbf{F}_N = \prod_{i=1}^{E}\prod_{f=1}^{F}\left(\mathbf{I}_{2^{(i-1)F+f-1}} \otimes \mathbf{B}^T_{2^{EF-(i-1)F-f+1}}\right). \tag{24}$$

This means that the radix-*R* factors **E**(*i*), where *i* runs from *1* to *E*, for the solutions provided by (23) (chosen as an example), take the form:

$$\mathbf{E}_N(i) = \prod_{f=1}^{F}\left(\mathbf{I}_{2^{EF-(i-1)F-f}} \otimes \mathbf{B}_{2^{(i-1)F+f}}\right). \tag{25}$$

Through **E**(*i*) can be simplified for a particular value of *F*, this simple notation allows radix-*R* stages to be written as a product of radix-2 stages. Moreover, this notation is very useful for obtaining mixed-radix factorizations.

3. NEW FFT/IFFT FACTORIZATIONS WITH REGULAR INTERCONNECTION PATTERN STAGE-TO-STAGE SUBBLOCKS

In [22], a radix-*R* equal stage-to-stage interconnection pattern factorization was derived for the Walsh-Hadamard Transform (WHT). The WHT has a factorization with the same stage-to-stage interconnection pattern as the Cooley-Tukey FFT factorization. This means that the same strategy used in [22] can be applied to FFT/IFFT to derive the well-known general radix-*R* Pease architectures. As we have already mentioned, the aim of this work is to obtain regular interconnection patterns at a scale lower than a stage. Since all Cooley-Tukey radix-*R* factorizations reproduce their interconnection pattern between stages at different smaller scales, as it is represented with a discontinuous line in Figure 1(A) for the radix-2 case, it is possible to find factorizations that reproduce the radix-*R* Pease property only partially at any of these smaller scales. Our new factorizations for both the FFT and IFFT will have two kinds of factors: a Cooley-Tukey type factors and new factors providing the same Pease interconnection pattern property in subblocks. The argument given in this section will become clear if we begin with the radix-2 case and further we generalize the results to radix-*R*.

3.1 PARAMETER a AND THE SIZE OF THE SUBBLOCKS WITH REGULAR INTERCONNECTION PATTERN PROPERTIES.

It will be interesting to say something about the scale in which the property of the pattern regularity can appear. Consider a full radix-2 Pease factorization. In this case, the size of the block having the regular interconnection stage-to-stage property is just the size of the full transform, this is, N×N where $N=2^n$. Now we consider new factorizations. In the case in where these factorizations have one radix-2 Cooley-Tukey type stage they can show two blocks with the regular interconnection pattern property. In the case in where they have two radix-2 Cooley-Tukey type stages they can show four blocks with this property. A general rule is as follows: if *a* is the number of radix-2 Cooley-Tukey type stages, we can obtain 2^a blocks of $2^{n-a}\times 2^{n-a}$ dimensions with the same regular interconnection pattern property. This can be seen in Figure 2 for the case N=32, a=1 and a=2.

3.2 THE FIRST FAMILY OF SOLUTIONS FOR THE FFT

We begin with expression (19) without taking into account the bit reversed reordering given by

$\mathbf{R}_N$ and considering only the right-hand side of this equation. In order to derive the new factorizations we first define the permutation matrices $\mathbf{K}_N$ (N=2^n) depending on the parameter a that controls the number of blocks with the same interconnection pattern property. Once a is selected, we have:

$$\mathbf{K}_N = \left(\mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}\right). \tag{26}$$

Another argument we have already mentioned is that our factorizations will have two kinds of factors: Cooley-Tukey type factors that we will be group in the below defined matrix $\mathbf{Y}_N$, and the factors having the property we are looking for; that we will group in the bellow defined $\mathbf{X}_N$. If $\mathbf{E}(i)$ is the i-th radix-2 stage, we will write expression (19) as follows:

$$\mathbf{F}_N\mathbf{R}_N = \prod_{i=1}^{n}\mathbf{E}(i) = \prod_{i=n-a+1}^{n}\mathbf{E}(i)\prod_{i=1}^{n-a}\mathbf{E}(i) = \mathbf{Y}_N\mathbf{X}_N, \tag{27}$$

where $\mathbf{X}_N$ and $\mathbf{Y}_N$ take the form:

$$\mathbf{Y}_N = \prod_{i=n-a+1}^{n}\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{B}_{2^i}\right), \tag{28}$$

$$\mathbf{X}_N = \prod_{i=1}^{n-a}\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{B}_{2^i}\right). \tag{29}$$

In order to obtain new factorizations we will modify $\mathbf{X}_N$ in the following way:

$$\mathbf{X}_N = \prod_{i=1}^{n-a}\mathbf{K}_N^{i}\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{B}_{2^i}\right)\mathbf{K}_N^{-i+1}. \tag{30}$$

Note that the introduction in (30) of these powers of the permutation matrices $\mathbf{K}_N$, does not modify the value of $\mathbf{X}_N$. One can see from (30) that, if i=1, $\mathbf{K}^{-i+1}$=$\mathbf{I}$, and if i=n-a, $\mathbf{K}^{i}$ =$\mathbf{I}$. This can be proved using definition (1) and property (5):

$$\mathbf{K}_N^{n-a} = \left(\mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}\right)^{n-a} = \left(\mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}^{n-a}\right) = \mathbf{I}_N. \tag{31}$$

The products of the remaining pairs of permutation matrices introduced between factors in (30) are equal to the identity matrix as a permutation matrix is just the inverse of the other. Observe also that the factors in (30) have the same mathematical complexity that in (29) since a permutation matrix only changes the order of operations.

Next, the factors in (30) can be rewritten by properties (2), (3) and (5):

$$\begin{gathered}
\mathbf{K}_{2^n}^{i}\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{B}_{2^i}\right)\mathbf{K}_{2^n}^{-i+1} = \\
\left(\mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}^{i}\right)\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{B}_{2^i}\right)\left(\mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}^{-i+1}\right) = \\
\left(\mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}^{i}\right)\left(\mathbf{I}_{2^a} \otimes \mathbf{I}_{2^{n-i-a}} \otimes \mathbf{B}_{2^i}\right)\left(\mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}^{-i+1}\right) \\
= \mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}^{i}\left(\mathbf{I}_{2^{n-i-a}} \otimes \mathbf{B}_{2^i}\right)\mathbf{P}_{2^{n-a}}^{-i+1}.
\end{gathered} \tag{32}$$

Let us consider a factor part of the last equation in (32) rewritten using (4) and (6) to the form:

$$\left(\mathbf{I}_{2^{n-i-a}} \otimes \mathbf{B}_{2^i}\right) = \mathbf{P}_{2^{n-a}}^{n-i-a}\left(\mathbf{B}_{2^i} \otimes \mathbf{I}_{2^{n-i-a}}\right)\mathbf{P}_{2^{n-a}}^{i}, \tag{33}$$

$$\mathbf{P}_{2^{n-a}}^{n-a-i} = \mathbf{P}_{2^{n-a}}^{-i}, \tag{34}$$

to obtain:

$$\mathbf{P}_{2^{n-a}}^{i}\left(\mathbf{I}_{2^{n-i-a}} \otimes \mathbf{B}_{2^i}\right)\mathbf{P}_{2^{n-a}}^{-i+1} = \left(\mathbf{B}_{2^i} \otimes \mathbf{I}_{2^{n-i-a}}\right)\mathbf{P}_{2^{n-a}}. \tag{35}$$

Therefore $\mathbf{X}_N$ in (30) becomes:

$$\mathbf{X}_N = \prod_{i=1}^{n-a-1}\left(\mathbf{I}_{2^a} \otimes \left(\mathbf{B}_{2^i} \otimes \mathbf{I}_{2^{n-i-a}}\right)\mathbf{P}_{2^{n-a}}\right), \tag{36}$$

or, equivalently, by property (3):

$$\mathbf{X}_N = \mathbf{I}_{2^a} \otimes \prod_{i=1}^{n-a-1}\left(\mathbf{B}_{2^i} \otimes \mathbf{I}_{2^{n-i-a}}\right)\mathbf{P}_{2^{n-a}}. \tag{37}$$

What is interesting here is that the new stages in (36) or in (37) contain 2^a blocks. The blocks can be calculated in *n*-*a*-1 stages and their structure is as follows:

$$\left(\mathbf{B}_{2^i} \otimes \mathbf{I}_{2^{n-i-a}}\right)\mathbf{P}_{2^{n-a}}. \tag{38}$$

Since the interconnection pattern is given by the position of the non-zero elements in each sparse matrix representing a factor, to show that these blocks have an identical interconnection pattern stage-to-stage, we can replace the matrix **B** by another simpler matrix $\mathbf{B}^+$ having its non-zero values exactly in the same positions than matrix **B**. The matrix $\mathbf{B}^+$ is obtained by replacing the diagonal matrix **A** in (9) by the identity matrix **I**:

$$\mathbf{B}_{2^i}^{+} = \begin{bmatrix} \mathbf{I}_{2^{i-1}} & \mathbf{I}_{2^{i-1}} \\ \mathbf{I}_{2^{i-1}} & -\mathbf{I}_{2^{i-1}} \end{bmatrix} = \mathbf{F}_2 \otimes \mathbf{I}_{2^{i-1}}. \tag{39}$$

Once the parameter *a* is chosen, the stages of the modified blocks will be equal and independent of *i*, that is:

$$\left(\mathbf{B}_{2^i}^{+} \otimes \mathbf{I}_{2^{n-i-a}}\right)\mathbf{P}_{2^{n-a}} = \left(\mathbf{F}_2 \otimes \mathbf{I}_{2^{i-1}} \otimes \mathbf{I}_{2^{n-i-a}}\right)\mathbf{P}_{2^{n-a}} = \left(\mathbf{F}_2 \otimes \mathbf{I}_{2^{n-a-1}}\right)\mathbf{P}_{2^{n-a}}. \tag{40}$$

Once it is shown that the subblocks of $\mathbf{X}_N$ have an identical interconnection pattern stage-to-stage, the mathematical expression of these new factorizations with the parameter *a* controlling granularity of the blocks, is:

$$\mathbf{F}_N\mathbf{R}_N = \left(\prod_{i=n-a+1}^{n}\left(\mathbf{I}_{2^{n-i}} \otimes \mathbf{B}_{2^i}\right)\right)\left(\mathbf{I}_{2^a} \otimes \prod_{i=1}^{n-a-1}\left(\mathbf{B}_{2^i} \otimes \mathbf{I}_{2^{n-i-a}}\right)\mathbf{P}_{2^{n-a}}\right). \tag{41}$$

As an example, in Figure 2, we show on the left-hand side (A) the interconnection pattern for a radix-2, N=32 (n=5) and a=1 factorization with 1 Cooley-Tukey type stage and 2 equal interconnection pattern stage-to-stage blocks. On the right-hand side (B), a radix-2 factorization with the same N=32 (n=5) and a=2, with 2 Cooley-Tukey type stages and 4 blocks with equal interconnection pattern stage-to-stage.

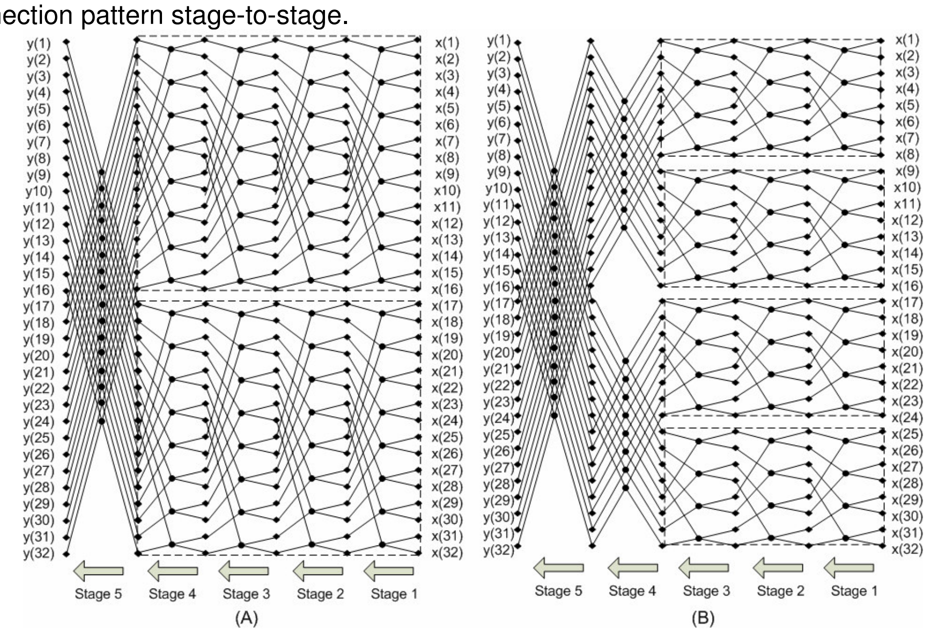

Figure 2 A 32-point radix-2 stage-to-stage interconnection pattern representations for the required new first family of solutions. Case A) with parameter a=1: one Cooley-Tukey type stage and two blocks with regular interconnection pattern. Case B) with parameter a=2: two Cooley-Tukey type stages and four blocks with regular interconnection pattern.

3.3 THE SECOND FAMILY OF SOLUTIONS FOR FFT.

Now we begin with expression (20). We will consider the permutation matrices $\mathbf{Q}_N$ ($N=2^n$) as follows:

$$\mathbf{Q}_N = \mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}^{-1} = \left(\mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}\right)^{-1} = \mathbf{K}_{2^n}^{-1}. \tag{42}$$

We want a solution containing two kinds of factors: an a Cooley-Tukey type factors grouped now in the matrix $\mathbf{X}_N$ and 2^a factors with the property we are looking for grouped now in the matrix $\mathbf{Y}_N$. If $\mathbf{E}(i)$ is the i radix-2 stage, we have then:

$$\mathbf{R}_N\mathbf{F}_N = \prod_{i=1}^{n}\mathbf{E}(i) = \prod_{i=a+1}^{n}\mathbf{E}(i)\prod_{i=1}^{a}\mathbf{E}(i) = \mathbf{Y}_N\mathbf{X}_N, \tag{43}$$

Where $\mathbf{X}_N$ and $\mathbf{Y}_N$ are taken in the following way:

$$\mathbf{X}_N = \prod_{i=1}^{a}\mathbf{I}_{2^{i-1}} \otimes \mathbf{B}_{2^{n-i+1}}^{T}, \tag{44}$$

$$\mathbf{Y}_N = \prod_{i=a+1}^{n}\mathbf{I}_{2^{i-1}} \otimes \mathbf{B}_{2^{n-i+1}}^{T}. \tag{45}$$

In order to change the interconnection pattern stage-to-stage, we transform $\mathbf{Y}_N$ as follows:

$$\mathbf{Y}_N = \prod_{i=a+1}^{n} \mathbf{Q}_{2^n}^{-n+i}\left(\mathbf{I}_{2^{i-1}} \otimes \mathbf{B}_{2^{n-i+1}}^T\right)\mathbf{Q}_{2^n}^{n-i+1}. \tag{46}$$

The factors **Q** in (46) can be written using the permutation matrix **P** via (42). With some algebra and taking into account properties (2), (3) and (5), we have:

$$\begin{gathered}\mathbf{Q}_{2^n}^{-n+i}\left(\mathbf{I}_{2^{i-1}} \otimes \mathbf{B}_{2^{n-i+1}}^T\right)\mathbf{Q}_{2^n}^{n-i+1} = \\ \left(\mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}^{n-i}\right)\left(\mathbf{I}_{2^a} \otimes \mathbf{I}_{2^{i-a-1}} \otimes \mathbf{B}_{2^{n-i+1}}^T\right)\left(\mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}^{-n+i-1}\right) = \\ \mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}^{n-i}\left(\mathbf{I}_{2^{i-a-1}} \otimes \mathbf{B}_{2^{n-i+1}}^T\right)\mathbf{P}_{2^{n-a}}^{-n+i-1},\end{gathered} \tag{47}$$

and by properties (4) and (6) we have:

$$\left(\mathbf{I}_{2^{i-a-1}} \otimes \mathbf{B}_{2^{n-i+1}}^T\right) = \mathbf{P}_{2^{n-a}}^{i-a-1}\left(\mathbf{B}_{2^{n-i+1}}^T \otimes \mathbf{I}_{2^{i-a-1}}\right)\mathbf{P}_{2^{n-a}}^{n-i+1}, \tag{48}$$

$$\mathbf{P}_{2^{n-a}}^{n-a-1} = \mathbf{P}_{2^{n-a}}^{-1}. \tag{49}$$

Therefore $\mathbf{Y}_N$ can be written as follows:

$$\mathbf{Y}_N = \prod_{i=a+1}^{n} \mathbf{I}_{2^a} \otimes \mathbf{P}_{2^{n-a}}^{-1}\left(\mathbf{B}_{2^{n-i+1}}^T \otimes \mathbf{I}_{2^{i-a-1}}\right), \tag{50}$$

or, equivalently, by property (3):

$$\mathbf{Y}_N = \mathbf{I}_{2^a} \otimes \prod_{i=a+1}^{n} \mathbf{P}_{2^{n-a}}^{-1}\left(\mathbf{B}_{2^{n-i+1}}^T \otimes \mathbf{I}_{2^{i-a-1}}\right). \tag{51}$$

We see from (51) that the factors of $\mathbf{Y}_N$ contain 2^a subblocks. These subblocks have an identical interconnection pattern stage-to-stage. Indeed, replace the matrix $\mathbf{B}^T$ by the matrix $\mathbf{B}^+$ defined in (39), as we have already done in the previous section. The matrix $\mathbf{B}^+$ has its non-zero elements in the same positions as the matrix $\mathbf{B}^T$. Moreover, as we have already mentioned, the interconnection pattern is given by the position of the non-zero elements in each sparse matrix. Therefore this substitution does not change the interconnection pattern but shows that the modified factorization does not depend at the stage *i*. Then $\mathbf{B}^T$ can be replaced by $\mathbf{B}^+$ and the modified factors independent on of the stage *i* take the form:

$$\mathbf{P}_{2^{n-a}}^{-1}\left(\mathbf{F}_2 \otimes \mathbf{I}_{2^{n-a-1}}\right). \tag{52}$$

The second family of factorizations has the following expression:

$$\mathbf{R}_N\mathbf{F}_N = \left(\mathbf{I}_{2^a} \otimes \prod_{i=a+1}^{n} \mathbf{P}_{2^{n-a}}^{-1}\left(\mathbf{B}_{2^{n-i+1}}^T \otimes \mathbf{I}_{2^{i-a-1}}\right)\right)\left(\prod_{i=1}^{a} \mathbf{I}_{2^{i-1}} \otimes \mathbf{B}_{2^{n-i+1}}^T\right). \tag{53}$$

Remember that the parameter *a* is the number of Cooley-Tukey type stages. If *a*=0, we obtain complete Pease factorizations. If $a=\log_2 N$ and $a=\log_2 N-1$, we have complete Cooley-Tukey factorizations. In Figure 3, for the same values of N and a as in Figure2, the radix-2 stage-to-stage interconnection pattern solutions for the new second family of solutions corresponding to (53) are represented. Observe that these solutions are symmetric.

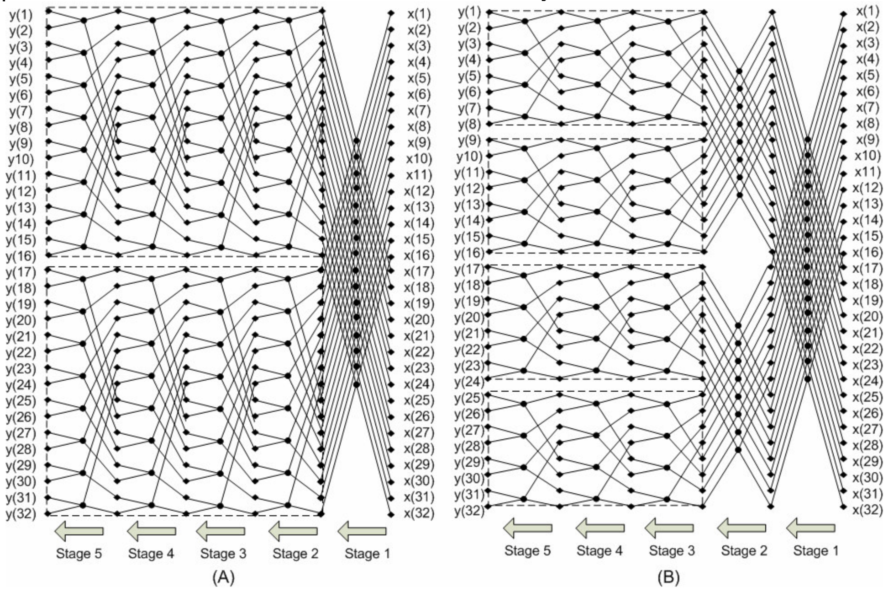

Figure 3 A 32-point radix-2 stage-to-stage interconnection pattern representations for the required new second family of solutions. Case A) with parameter a=1: one Cooley-Tukey type stage and two blocks with regular interconnection pattern. Case B) with parameter a=2: two Cooley-Tukey type stages and four blocks with regular interconnection pattern.

3.4 IFFT FACTORIZATIONS.

The factorization given for $\mathbf{F}^H$ in expression (21) shares the same interconnection pattern as that for $\mathbf{F}$ in expression (20). Using the same method that in sections 3.2 and 3.3, with the same permutation matrices presented in (42) and with the same parameter a to control granularity of the blocks, the new two families of factorizations for $\mathbf{F}^H$ have the form:

$$\mathbf{R}_N\mathbf{F}_N^H = \left(\mathbf{I}_{2^a}\otimes\prod_{i=a+1}^{n}\mathbf{P}^{-1}_{2^{n-a}}\left(\mathbf{B}^H_{2^{n-i+1}}\otimes\mathbf{I}_{2^{i-a-1}}\right)\right)\left(\prod_{i=1}^{a}\mathbf{I}_{2^{i-1}}\otimes\mathbf{B}^H_{2^{n-i+1}}\right), \tag{54}$$

and

$$\mathbf{F}_N^H\mathbf{R}_N = \left(\prod_{i=n-a+1}^{n}\left(\mathbf{I}_{2^{n-i}}\otimes\mathbf{B}^*_{2^i}\right)\right)\left(\mathbf{I}_{2^a}\otimes\prod_{i=1}^{n-a-1}\left(\mathbf{B}^*_{2^i}\otimes\mathbf{I}_{2^{n-i-a}}\right)\mathbf{P}_{2^{n-a}}\right). \tag{55}$$

From the point of view of the interconnection pattern (54) and (42) are equivalent, the same than (55) and (53).

4. RADIX-R GENERALIZATION AND MIXED-RADIX FACTORIZATIONS

Take $R=2^F$. Then any discrete Fourier transform of size $N=R^E$, admits a radix-R factorization of E radix-R factors $\mathbf{E}'(i)$ that can easily be written as a product of F successive radix-2 factors $\mathbf{E}(i)$:

$$\mathbf{E}'_N(i) = \prod_{f=1}^{F}\mathbf{E}_N\left((i-1)F+f\right). \tag{56}$$

The radix-R factorization becomes:

$$\prod_{i=1}^{E} \mathbf{E}'_N(i). \qquad (57)$$

This notation can be used for any kind of decomposition to obtain radix-R stages. In Figure 4, two different solutions A) and B) for the full radix-4 and N=16 equal interconnection patterns or Pease factorizations are represented. This case is a particular case of our factorizations with a=0.

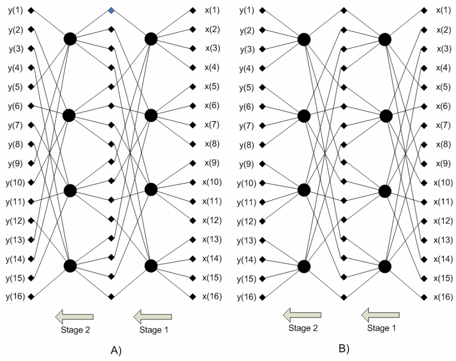

Figure 4 A 16-point radix-4 equal interconnection pattern stage-to-stage solutions obtained from A) the first family of solutions and B) the second family of solutions, both for parameter a=0. These two solutions have 0 Cooley-Tokey type stages and only 1 block exhibiting the same interconnection pattern and they are the same as the radix-4Pease solutions.

It is interesting to note that representing higher radix stages in function of radix-2 factors is not restricted to regular radix-R representations. Mixed radix factorizations can sometimes provide certain numeric advantages or become interesting when the transform size does not allow full radix-R decomposition. This is shown as an example in Figure 5. In case A) the solution given in Figure 2-B) is modified by grouping together the radix-2 stages i=4 and i=5 to form the radix-4 stage. In Figure 5.B) the solution given in Figure 2-A) is replaced by grouping together the radix-2 stages i=1 and i=2 and stages i=3 and i=4 to form the radix-4 stages 1 and 2 respectively.

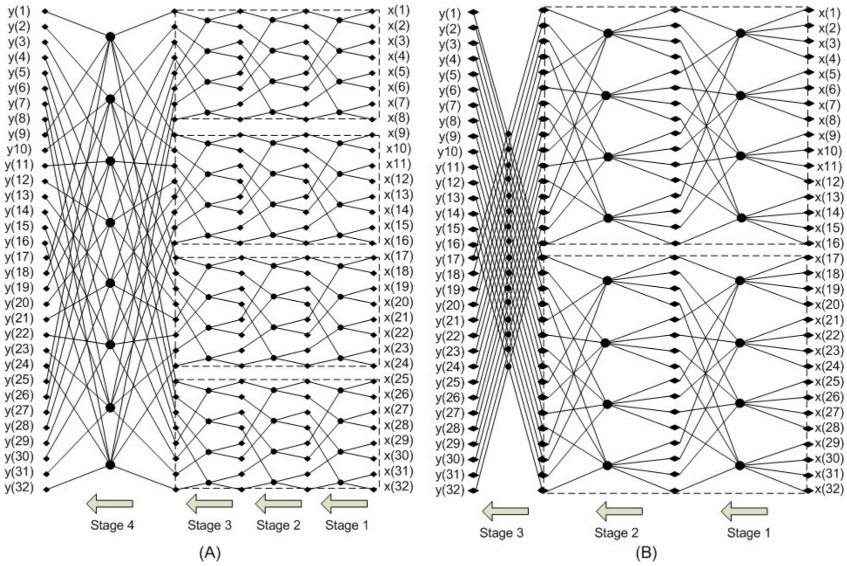

Figure 5 32-point FFT mixed-radix factorizations combining (A) 1 radix-4 stage with 4 subblocks with 3 radix-2 equal interconnection pattern stages and (B) 1 radix-2 stage with 2 subblocks with 2 radix-4 equal

interconnection pattern stages.

5. CONCLUSIONS

In this paper we have obtained a family of new factorizations that repeat the regular Pease interconnection pattern inside subblocks. It is shown that these new factorizations can be obtained to reproduce the regular interconnection pattern property at a previously selected scale and that we have some margin to select the size of these subblocks. A characteristic of the presented factorizations is that they exhibit two kinds of factors: Cooley-Tukey type factors and new factors providing subblocks with parallel or identical interconnection pattern stage-to-stage. The number and the size of the introduced subblocks and the number of Cooley-Tukey type factors are related to each other. We have shown two kinds of topologies for both the FFT and the IFFT transforms and it is shown the way to obtain radix-R and mixed-radix factorizations from radix-2 ones. Our factorizations can find applications in FFT/IFFT implementation architectures, where the subblock part can be implemented in hardware, taking advantage of the parallel topology, while the Cooley-Tukey type stages can be implemented in software (some OFDM based standards have different operation modes that work with FFTs/IFFTs of different sizes). It is interesting to observe that, if we have the hardware to compute a block of size N in parallel, with this hardware architecture we can compute de FFT of size N using the pure Pease algorithm. With the same hardware, using one of these new factorizations, we can compute a 2N FFT by computing two equal interconnection pattern subblocks of size N, and since the first remaining stage, a Cooley-Tukey type stage, that can be computed without any multiplications, only with additions or subtractions as the coefficients of this stage are 1 or -1, it can be computed by software very efficiently. A forthcoming paper will deal with different practical implementations on a FPGA. Given that different discrete transforms have factorizations with a Cooley-Tukey type stage-to-stage interconnection, the same argument can easily be extended to them and the same kind of factorizations can be obtained for them. It will be also interesting to extend this kind of factorizations to the two dimensional case.

REFERENCES

[1] M. T. Heideman, D. H. Johnson, and C. S. Burrus, "Gauss and the History of the FFT," IEEE Acoustics, Speech, and Signal Processing Magazine, vol. 1, pp. 14-21, Oct. 1984.
[2] J. W. Cooley, J. W. Tukey “An Algorithm for the Machine Calculation of Complex Fourier Series”. *Math. of Computations* Vol. 19, p.p. 297-301, April. 1965.
[3] G. D. Bergland, “A Radix-Eight Fast-Fourier Transform Subroutine for Real-Valued Series,” IEEE Trans. Audio Electroacoust. vol. 17, no. 2, pp. 138-144, June 1969.
[4] D. Takahashi, “A Radix-16 FFT Algorithm Suitable for Multiply-Add Instruction based on Goedecker Method“ Intern. Conference on Acoustics, Speech, and Signal Processing, ICASSP-2003, Page(s):II - 665-8 vol.2 6-10, April 2003.
[5] R. C. Singleton, “An Algorithm for Computing the Mixed Radix Fast Fourier Transform” IEEE Trans. Audio Electroacoust. vol. 1, no. 2, pp. 93-103, June 1969.
[6] D. Pl. Kolba and T. W. Parks, “A Prime Factor FFT Algorithm Using High-Speed Convolution,” IEEE Trans. Acoust., Speech, Signal Processing, vol. 25, no. 4, pp. 281-294, August 1977.
[7] S. Winograd, “On Computing the Discrete Fourier Transform,” Math. Comput., vol. 32, no. 141, pp. 175-199, Januany 1978.
[8] H. V. Sorensen and C. S. Burrus, “A New Efficient Algorithm for Computing a Few DFT Points,” IEEE Trans. Acoust., Speech, Signal Proc., vol. 35, no. 6, pp. 849-863, June 1987.
[9] D. Takahashi, “An Extended Split-Radix FFT Algorithm,” IEEE Signal Processing Letters, vol. 8, no. 5, pp. 145-147, May 2001.
[10] A. R. Varkonyi-Koczy, “A Recursive Fast Fourier Transform Algorithm,” IEEE Trans. Circuits

and Systems, II, vol. 42, pp. 614-616, September 1995.
[11] A. Saidi, “Decimation-in-Time-Frequency FFT Algorithm,” Proc. IEEE International Conf. on Acoustics, Speech, and Signal Processing, pp. III: 453-456, April 19-22 1994.
[12] M. C. Pease “An adaptation of the fast Fourier transform for parallel processing”. *J. Assoc. Comput.* Vol. 15, p.p. 252-324, April 1968.
[13] P. Duhamel and M. Vetterli “Fast Fourier transforms: A tutorial review and a state of the art” *Signal Process.,* vol. 19, pp. 259-299, 1990.
[14] J. A. Glassman, “A generalization of the fast Fourier transform,” IEEE Trans. Comput., vol. C-19, pp. 105-116, February 1970.
[15] M. Drubin, “Kronecker product factorization of the FFT matrix,” IEEE Trans. Comput., vol. C-20, pp. 590-593, May 1971.
[16] H. Sloate, “Matrix Representations for Sorting and the Fast Fourier Transform” IEEE Trans.on Circuits and Systems, vol., cas-21, No. 1, pp. 109-116 January 1974.
[17] J. Granata, M. Conner, R. Tolimieri, “Recursive Fast Algorithms and the Role of the Tensor Product” IEEE Trans.on Signal Proc., vol.40., No. 12, pp. 2921-2930 December 1992.
[18] S. Egner, M. Püschel, “Automatic Generation of Fast Discrete Signal Transforms” IEEE Trans.on Signal Proc., vol.49., No. 9, pp. 1992-2002 December 2001.
[19] L. Yu-Wei, L. Hsuan-Yu, L. Chen-Yi, “A 1-GS/s FFT/IFFT Processor for UWB Applications” IEEE Jour. of Solid State Circuits, , vol.40., No. 8, pp. 1732-1735 August 2005.
[20] G. J. Byung, M. H. Sunwoo, “New Continuous-Flow Mixed-Radix (CFMR) FFT Processor Using Novel In-Place Strategy” IEEE Trans. on Circuits and Systems, vol.52, No. 5, pp. 911-919 May 2005.
[21] G., Miel “Constant geometry fast Fourier transforms on array processors Computers” *IEEE Trans. on Computers* Vol. 42, Issue 3 p.p. 371 – 375, March 1993.
[22] P. Marti-Puig, “A Family of Fast Walsh Hadamard Algorithms with Identical Matrix Factorization” IEEE Signal Proc. Letters, vol.13., No. 11, pp. 672-675 November 2006.
[23] R. Yavne, “An economical method for calculating the discrete Fpurier Transform”, in Poc. AFIPS Fall Fall Joint Compt. Conf., vol 33, p.p.115-125, 1968.
[24] P., Duhamel “Algorithms Meeting the Lower Bounds on the Multiplicative Complexity of Length-2^n DFT’s and Their Connection with Practical Algorithms” IEEE Trans. Acoust., Speech Signal Process., vol. 38, no. 9, pp. 1504-1511,1990.
[25] S., Bouguezel; Ahmad, M.O.; Swamy, M.N.S. “A new radix-2/8 FFT algorithm for length-$q \times 2^m$ DFTs” IEEE Trans. Circuits Syst. I Vol. 51, Iss. 9, p.p.:1723 – 1732, September 2004.
[26] S. G., Johnson; M., Frigo “A Modified Split-Radix FFT With Fewer Arithmetic Operations” IEEE Transactions on Signal Processing : Accepted for future publication Volume PP, Issue 99, 2006 Page(s):1 – 1
[27] M., Serra; P., Marti; J. Carrabina, “IFFT-FFT core architecture with an identical stage structure for wireless LAN communications” Signal Proc. Advan. in Wireless Comm., 2004 IEEE 5th Workshop on 11-14, Page(s):606 – 610, July 2004
[28] E.L. Zapata; F. Arguello; “A VLSI constant geometry architecture for the fast Hartley and Fourier transforms Parallel and Distributed Systems”, *IEEE Trans. on Parallel and Distributed Systems* Vol. 3, I. 1, p.p. 58 – 70, January 1992.
[29] J.E. Whelchel, J.P. O'Malley, W.J. Rinard,; J.F. McArthur, “The systolic phase rotation FFT- a new algorithm and parallel processor architecture” *Intern. Conference on Acoustics, Speech, and Signal Processing, ICASSP-90*, 3-6, p.p. 1021 - 1024 Vol. 2, April. 1990.
[30] S. He, M. Torkelson, “A new approach to pipeline FFT processor” *10th Intern.Parallel Processing Symposium IPPS’96, IPPS*, 15-19, p.p. 766 – 770, April 1996.
[31] V. Boriakoff, “FFT computation with systolic arrays, a new architecture” IEEE Trans.on Circuits and Systems II: Analog and Digital Signal Processing., vol.41., No. 4, pp. 278-284 April 1994.