An Analysis for the Realization of an In-Place and In-Order Prime Factor Algorithm

Daniel Pak-Kong Lun, Member, IEEE, and Wan-Chi Siu, Senior Member, IEEE

Abstract—For the computation of the prime factor algorithm (PFA), an in-place and in-order approach is always desirable not only because of its reduction on memory requirement for the storage of the temporary results, but also because the computation time which is required to unscramble the output sequence to a proper order can also be saved. In this paper, we show that the PFA has an intrinsic property that allows it to be easily realized in an in-place and in-order form. As it is different from the other propositions which use two equations respectively for loading data from and retrieving results back to the memory, in this paper we show formally that in many cases only one equation is enough for both operations. Thus a truly in-place and in-order computation is accomplished. Nevertheless, the sequence length of the PFA computation must be carefully selected. Another objective of this paper is to analyze the conditions on which a particular sequence length is possible for in-place and in-order PFA computation. The result of this paper is useful to both the hardware and software realization of the PFA.

I. INTRODUCTION

Since the introduction of prime factor mapping (PFM) for the computation of the discrete Fourier transform (DFT) by Good [1] and Thomas [2], a variety of research [3]–[12] has been carried out investigating efficient approaches for its realization. Some researchers concentrate on the investigation of fast algorithms for the computation of the decomposed short lengths. Some encouraging results were presented by Kolba and Parks [3] in their prime factor FFT algorithm (PFA) where Winograd’s short length algorithms [4] and Rader’s mapping method [5] were utilized.

Another direction of the research is to make investigations on efficient approaches to realize the prime factor mapping. Among them, an in-place and in-order approach is most desirable. The term “in-place” computation is defined as the computation where the memory requirement is restricted to be as large as the problem size. However, in actual computation, a very small amount of temporary buffers is usually allowed for the storage of the intermediate results. An “in-order” computation refers to the computation which has its output sequence in a natural order. These two properties are highly appreciated not just because it reduces the memory requirement during the computation, but also the computation time which is required to unscramble the output sequence into proper order can also be saved.

Due to the significance of the in-place and in-order computation, several researchers have addressed this problem already. Recently, Burrus and Eschenbacher [6], Arambepola [7], Temperton [8], Wong et al. [9], and Lun et al. [10] have proposed some in-place and in-order addressing schemes for the realization of the PFA. All schemes show their improvements in speed and memory storage requirement as compared with the non-in-place and non-in-order approaches [11], [12]. However, extra computational efforts are required in their approaches. For example, in [7], [9] and [10], an unscrambling process must be carried out during each intermediate stage of computation. In [6] the in-place and in-order computation is achieved by using dedicated short length modules for different problem sizes. It is more desirable if these deficiencies can be removed. On the other hand, [8] gives a general and practical solution for the in-place and in-order computation of the PFA. By selecting an appropriate set of multiplier constants for each individual short length module at run time, the in-place and in-order mapping scheme given in [6] can be generally applied without the need for using dedicated short length modules for different problem sizes. Nevertheless, this approach still has its drawback in that if in a situation where more modules, such as 11, 13, 17, 19 or longer lengths have to be used, the number of multiplier constants required to be stored in the program will sharply increase. Although these longer modules are not frequently used, it is still more desirable if there is an in-place and in-order indexing scheme which requires a similar computational effort as that in [8], but uses only a single set of multiplier constants for each short length module in any circumstance.

In this paper, we show that, in fact, the PFA is inherently possible to be easily realized in an in-place and in-order form. As it is different from the previous approaches in that two equations are required, respectively, for loading data from and retrieving results back to the memory, we show that in many cases only one equation is enough for both operations. No extra computational effort (such as unscrambling during the intermediate stages, using dedicated short length modules or selecting different multiplier constants) are required to accomplish the objective.
since it is an inherent property of the PFA. We refer to
the PFA computation based on this inherent property as
the "natural" in-place and in-order PFA computation to
distinguish it from those which require extra computa-
tional efforts. Nevertheless, in order to achieve natural in-
place and in-order computation, the sequence length of
the PFA computation must be carefully selected. In the
following sections, we will give an analysis of the con-
ditions in which a particular sequence length is possible
for a natural in-place and in-order PFA computation.

It is interesting to note that some analyses on this as-
pect have been done by Stein and Taşci [13]. The differ-
ences between the one given in this paper and that in [13]
are as follows. First, the results given in [13] on the se-
quence lengths which are suitable for natural in-place and
in-order PFA computation are restricted to the form
$$2^n 3^n 5^n 7^n$$, where $n_i$ is any positive integer including zero.
In this paper, other conditions and longer modules such
as 11, 13, 17, 19, etc., are also considered, however, only
the sequence lengths which have factors which are less
than or equal to 25 are listed. This is because efficient
short length algorithms of these sizes are already available
[3], [14] and generally used. It can be seen that the two
set of results listed in [13] and this paper are different and
complementary, while a small overlapping part of the lists
gives similar results.

Second, if the sequence length is composed of only two
relatively prime factors, we derive in this paper some gen-
eral conditions on which this sequence length can be
realized in an in-place and in-order form. The characteris-
tic of these conditions is that they are not restricted by the
sizes or the magnitudes of the exponents of these factors.

Third, [13] gives no practical realization technique for
the natural in-place and in-order PFA computation. In this
paper, we propose an efficient approach for its realiza-
tion.

Finally, the analysis given in [13] covers the whole
subject of index transformations for multidimensional
DFT's and convolutions. Consequently, the analysis con-
cerning the topic of in-place and in-order PFA computa-
tion might be considered to be lack of details. Further-
more, as the analysis was done using some abstract group-
thoretical techniques, the results are less accessible to
general readers. However, in our paper, we try to explain
as clearly as possible how the results are obtained and how
they can be practically realized.

II. NATURAL IN-PLACE AND IN-ORDER COMPUTATION

The DFT of an $N$-point sequence $x(n)$, $n = 0, 1, \cdots, N - 1$ is given by
$$X(k) = \sum_{n=0}^{N-1} x(n) W^N_{nk}$$ (1)
where $W_N = \exp(-2\pi j/N)$ and $k = 0, 1, \cdots, N - 1$. If
$$N = \prod_{i=1}^{h} N_i,$$
where GCD $(N_a, N_b) = 1$ for $a, b = 1, 2, \cdots, h, a \neq b$, then (1) can be converted into a multidimensional form
by a proper prime factor mapping technique (the term
GCD $(A, B)$ refers to the greatest common factor of $A$ and
$B$). The algorithm which adopts this technique in com-
puting the DFT with the Winograd or other suitable short
length algorithms is called the prime factor algorithm [3].
A general mapping scheme was proposed in [15] and can
be used for the PFM. Namely,
$$n_i = \langle q_i, n \rangle_{N_i}$$
$$k_i = \langle r_i, k \rangle_{N_i}$$
where $\langle C \rangle_{N_i}$ means the residue of an integer $C$ modulo
another integer $N_i$. $q_i$ and $r_i$ are integers with the condition
that
$$\langle q_i, r_i \rangle_{N_i} = 1 \quad \text{for } M_i = N/N_i.$$ (2)
That is,
$$\langle q_i, r_i \rangle_{N_i} = \langle M_i^{-1} \rangle_{N_i}. \quad (3)$$
The indices $n$ and $k$ can be found such that
$$n = \left\langle \sum_{i=1}^{h} r_i M_i n_i \right\rangle_N \quad (4)$$
$$k = \left\langle \sum_{i=1}^{h} q_i M_i k_i \right\rangle_N. \quad (5)$$

In order to compute the PFA in an in-place and in-order
form, it is necessary that the loading equation, i.e., (4),
has the same format as the retrieval equation, i.e., (5)
[10]. Previous propositions achieved this either using
an explicit unscrambling process at the end of the PFA
computation [11], [12] or implicitly using some small un-
scrambling processes at the intermediate stages [7], [9],
[10]. Both of them incur some computational overheads.
However, as can be seen in (4) and (5), besides the in-
dices $n_i$ and $k_i$, the two equations differ only in the coef-
ficients $r_i$ and $q_i$. Equation (3) shows that the product
of the coefficients $r_i$ and $q_i$ is equal to $\langle M_i^{-1} \rangle_{N_i}$. Hence if the
square root of the integer $\langle M_i^{-1} \rangle_{N_i}$ exists, the coefficients
$r_i$ and $q_i$ can be set to be equal and hence a natural in-
place and in-order computation can be achieved. In this
case, the mapping equations become
$$n = \left\langle \sum_{i=1}^{h} \langle M_i^{-1} \rangle_{N_i} M_i \cdot n_i \right\rangle_N \quad (6)$$
$$k = \left\langle \sum_{i=1}^{h} \langle M_i^{-1} \rangle_{N_i} M_i \cdot k_i \right\rangle_N. \quad (7)$$

To test the existence of the square root of $\langle M_i^{-1} \rangle_{N_i}$ it
is equivalent to test the following binomial congruence
whether to have a root:
$$x^2 \equiv M_i^{-1} \mod N_i \quad \text{where } x \text{ is any integer.} \quad (8)$$
However, it will be shown that if the square root of
$\langle M_i \rangle_{N_i}$ exists, the root of congruence (8) exists.

Lemma 2.1. If the square root of $\langle M_i \rangle_{N_i}$ exists,
where \( M_i = N / N_i \), the square root of the inverse of \( M_i \)
with respect to the modulus \( N_i \) also exists.

*Proof:* If the square root of \( \langle M_i \rangle_{N_i} \) exists, it implies
that the following binomial congruence is solvable:
\[
x^2 = M_i \mod N_i.
\]
(9)
Let \( x_0 \), which is an integer, be the root of (9) such that
\[
x_0^2 = M_i \mod N_i.
\]
Hence
\[
(x_0^{-1})^2 = M_i^{-1} \mod N_i.
\]
(11)
\[
(x_0^{-1})^2 = N_i \mod N_i.
\]
(12)
Since GCD \( (M_i, N_i) = 1 \), it implies that GCD \( (x_0, N_i) = 1 \),
therefore \( \langle x_0 \rangle_{N_i} \) must be an integer, i.e., \( \langle M_i^{-1} \rangle_{N_i} \) is also
the square of an integer. The lemma is thus proved.

The problem of testing the existence of a square root of
\( \langle M_i^{-1} \rangle_{N_i} \) is now simplified to the testing of a square root
of \( \langle M_i \rangle_{N_i} \). We do not need to find out the inverse of \( \langle M_i \rangle_{N_i} \).

Lemma 2.1 implies that for all factors \( N_i \), where \( i = 1, 2, \ldots, h \), and \( N = N_1 \cdot N_2 \cdot \ldots \cdot N_h \), if the square roots of
the products of the other factors \( N_r \) for \( r = 1, \ldots, h, r \neq i \) exist, a natural in-place and in-order computation
can be achieved. An intuitive solution [16] for this requirement is that if the sequence length \( N \) consists of
some factors such as \( 3, 9, 16, 25, \ldots \), the computation
can be carried out in a natural in-place and in-order form
since the square roots of them always exist. Besides these integers, it is interesting to point out that there are some
integers which do not have any square root in the natural
integer system but have square roots when they are in a
certain residue class modulo an integer. For example, the
square root of 5 is not an integer but an integer square root exists when it is modulo with 4:
\[
\sqrt{5}_{4} = \langle 3 \rangle_{4} \text{ or } \langle 1 \rangle_{4}.
\]
(13)
One of the major objectives of this paper is to find these integers such that when they are modulo to each other, the square roots of them can be found. Consequently, by
using these square roots, we can construct the mapping
equations to accomplish the in-place and in-order computations naturally.

**III. Prime Factor Mapping With Two Relatively Prime Factors**

If the sequence length \( N \) for the prime factor mapping
composes of 2 factors such that \( N = N_1 \cdot N_2 \), (9) can be
rewritten as follows:
\[
x_1^2 = N_1 \mod N_2
\]
(14)
\[
x_2^2 = N_2 \mod N_1
\]
(15)
To determine the existence of the square roots of \( \langle N_1 \rangle_{N_2} \) and
\( \langle N_2 \rangle_{N_1} \), it is equivalent to determine whether (14) and
(15) are solvable or not. If (14) and (15) are solvable, we
say that \( N_1 \) is the quadratic residue [17], [18] of \( N_2 \), and
\( N_2 \) is also the quadratic residue of \( N_1 \). That is, \( N_1 \) and \( N_2 \)
are the quadratic residues to each other. In order to find
out if \( N_1 \) is the quadratic residue of \( N_2 \) or vice versa, the
Legendre's symbol can be used.

If \( p \) is an odd prime and \( D \) an integer not divisible by \( p \),
Legendre's symbol \( (D/p) \) [17] is said to be equal to 1
if \( D \) is a quadratic residue to the modulus \( p \), and equal to
-1 if \( D \) is a quadratic nonresidue to \( p \). The properties of
the Legendre's symbol are as follows [17]:

I) \( (D/p) = D^{(p-1)/2} \mod p \);

II) if \( D = D' \mod p \), then \( (D/p) = (D'/p) \);

III) if \( D \) and \( D' \) are integers not divisible by \( p \), then
\[
\left( \frac{DD'}{p} \right) = \left( \frac{D}{p} \right) \cdot \left( \frac{D'}{p} \right).
\]
IV) \( (1/p) = (1/(p-1)/2) \);

V) \( (2/p) = (-1)^{(p-1)/2} \).

By making use of these five properties, we are going to show that there are some interesting properties on the format of the numbers such that they are the quadratic residues to each other. The analysis will be divided into three parts. First, the case that \( N_1 \) and \( N_2 \) are odd primes is discussed. Then the situation when one of the factors is a power of 2 is considered. Finally, it will be concluded by a general discussion of the case where \( N_1 \) and \( N_2 \) are both powers of primes.

**A. The Case with Two Odd Primes Factors**

Consider that \( N_1 \) and \( N_2 \) are odd primes. It has been
shown [17], [18] that the format of these integers must exist in a fixed pattern if they are quadratic residues of
each other. We will make use of the law of quadratic reciprocity [17], [18] for the illustration. The law of quadratic reciprocity is as follows: if \( p \) and \( q \) are two odd
primes,
\[
\left( \frac{p}{q} \right) \cdot \left( \frac{q}{p} \right) = (-1)^{\left( p-1 \right)/2 \cdot \left( q-1 \right)/2}.
\]
In this case, \( (p/q) = (q/p) \) if and only if \( p \) or \( q \) is of
the form \( 4a + 1 \) with an integer \( a \). That is, \( N_1 \) and \( N_2 \) are
the quadratic residues to each other if either one of them is of
the form \( 4a + 1 \) with any positive integer \( a \), and either
\( (N_1/N_2) \) or \( (N_2/N_1) \) = 1. Furthermore, if both \( N_1 \) and \( N_2 \)
are of the form \( 4a + 3 \), \( N_1 \) and \( N_2 \) can never be the quadratic
residues to each other since, in this case,
\[
\left( \frac{N_1}{N_2} \right) = \left( \frac{N_2}{N_1} \right).
\]
To conclude this part, if \( N_1 \) and \( N_2 \) are two odd prime
numbers, they can be quadratic residues to each other if
and only if either \( N_1 \) or \( N_2 \) (or both) is (are) of the form
\( 4a + 1 \) and \( (N_1/N_2) = 1 \) or \( (N_2/N_1) = 1 \), where \( a \) is any positive integer.
Hence, we know that the PFA's with sequence lengths
like \( 3 \times 7 \) and \( 7 \times 11 \) can never be realized naturally in
an in-place and in-order form because 3, 7 and 7, 11 are
of the form $4a + 3$, and they can never be the quadratic residues to each other.

B. The Case with One of the Factors Being a Power of 2

Consider the case where one of the factors is a power of 2. By using the property $V$ of the Legendre’s symbol, we can infer that [17] 2 is a quadratic residue to all primes $p$ of the form $8a \pm 1$ and is not a quadratic residue to any prime $p$ of the form $8a \pm 3$ (where $a$ is an integer). For the integers which are the even powers of 2, such as 4, 16, etc., it can be easily seen that the square roots of them exist in all cases. That is, they are the quadratic residues to all numbers relatively prime to them. For the integers which are the odd powers of 2, such as 8, 32, etc., the following lemma describes the case.

Lemma 3.B.1: If there exist an odd prime $p$ such that 2 is a quadratic residue to it, the odd powers of 2 are also the quadratic residues to this odd prime number.

Proof: This can easily be shown by using property III of Legendre’s symbol, since

$$\left(\frac{2^{2k+1}}{p}\right) = \left(\frac{2^{2k}}{p}\right) \cdot \left(2\right) = \left(\frac{2^2}{p}\right) \cdot \left(2\right) \cdot \left(\frac{2}{p}\right)$$

where $k$ is any positive integer.

From property I of Legendre’s symbol, we know that

$$\left(\frac{2^2}{p}\right) = \left(2\right)^{p-1} \mod p.$$

As GCD $(2, p) = 1$, this implies GCD $(2^2, p) = 1$. Hence by Fermat’s simple theorem, we know that

$$\left(\frac{2^2}{p}\right) = 1 \mod p.$$

Consequently,

$$\left(\frac{2^{2k+1}}{p}\right) = \left(\frac{2^2}{p}\right) \cdot \left(\frac{2}{p}\right) = 1 \cdot \left(\frac{2}{p}\right) = \left(\frac{2}{p}\right).$$

Thus the lemma is proved.

We have shown the conditions on which an integer of power of 2 is a quadratic residue of an odd prime $p$ by using the Legendre’s symbol. However, the reverse is not true. We cannot use the Legendre’s symbol to determine whether an odd prime $p$ is a quadratic residue of an integer which is a power of 2 since the Legendre’s symbol is defined only when the modulus is an odd prime. Recall (15) when $N$ is equal to $2^2$,

$$x^2 \equiv N_2 \mod 2^2.$$

It is shown in [17] that in order to have the congruence $x^2 \equiv N_2 \mod 2^2$, where $N_2$ is odd and $r$ is an integer, to be solvable, it is necessary and sufficient that $N_2$ should be of the form $2a + 1$, $4a + 1$ or $8a + 1$, where $a$ is any positive integer, depending on whether $r = 1$, $r = 2$, or $r > 2$, respectively. Combining the results obtained in the above two cases, we conclude that,

1) If $r = 1$, i.e., $2^r = 2$, $N_2$ and 2 are the quadratic residues to each other if and only if $N_2$ is of the form $8a + 1$, where $a$ is any positive integer.

2) If $r = 2$, i.e., $2^r = 4$, $N_2$ and 4 are the quadratic residues to each other if and only if $N_2$ is of the form $4a + 1$, where $a$ is any positive integer.

3) If $r > 2$, $N_2$ and $2^r$ are the quadratic residues of each other if and only if $N_2$ is of the form $8a + 1$, where $a$ is any positive integer.

Hence we know that if $N = 2 \times 7$, the PFA with this problem size can be realized in a natural in-place and in-order form as implied in conclusion (1), since 7 is of the form $8a - 1$. Contrary, we know that factors like 5 and 8 are not the quadratic residues to each other, since 5 is not of the form $8a + 1$.

C. A General Representation of Two Integers

In this section, we derive the general conditions on which two relatively prime integers are the quadratic residues to each other. We define the sequence length of a PFA computation, namely $N$, composed of two relatively prime factor primes $N_1$ and $N_2$, where $N_1$, $N_2$, $r_1$ and $r_2$ are any positive integer. This form is a general representation of two integers which are relatively prime. Hence the analysis in this section can be considered as a generalization of the above two sections. If $N_1$ and $N_2$ are quadratic residues to each other, it implies that

$$\begin{align*}
(x^2 & \equiv N_1 \mod N_2) \quad \text{and must be solvable. (17)} \\
y^2 & \equiv N_2 \mod N_1
\end{align*}$$

We are now trying to illustrate the conditions on which congruences (17) are solvable. The following analysis is separated into two parts. First, we assume that both $N_1$ and $N_2$ are odd primes. Second, we assume either one of the factors, i.e., $N_1$ or $N_2$, is equal to 2.

1) The Case with Two Numbers Being Powers of Odd Primes: First, let us show the following lemma.

Lemma 3.C.1: The congruence $x^2 \equiv N_1^{r_1} \mod N_2^{r_2}$ is solvable if i) $r_1$ is even or ii) $r_2$ is odd and $(N_1/N_2) = 1$, where $i, j = 1, 2$, and $i \neq j, N = N_1^{r_1} \cdot N_2^{r_2}$ and GCD $(N_1, N_2) = 1$.

Proof: In order to prove the lemma, the following theorem [17] is used.

Theorem I: The congruence $x^2 \equiv D \mod p^r$, where $p$ is an odd prime, $r$ is a natural number, and $D$ is an integer not divisible by $p$, is solvable if and only if $D$ is a quadratic residue modulo $p$.

It can be seen in Theorem I that if the congruence $x^2 \equiv N_1^{r_1} \mod N_2^{r_2}$ (where $N = N_1^{r_1} \cdot N_2^{r_2}$ and GCD $(N_1^{r_1}, N_2^{r_2}) = 1$) is tested for the existence of a root, it is sufficient to test the congruence $x^2 \equiv N_2^{r_2} \mod N_1$. In this case, the Legendre’s symbol $(N_1/N_2)$ can be used to determine whether $N_2$ is the quadratic residue of $N_1$. Furthermore, for the property of Legendre’s symbol (property III) we know that

$$\left(\frac{N_1}{N_2}\right) = \begin{cases} 
N_1 & \text{if } r_1 \text{ is odd} \\
N_2 & \text{if } r_2 \text{ is odd} \\
1 & \text{if } r_i \text{ is even}.
\end{cases}$$
Hence we know that if \( r_1 \) is even or if \( r_2 \) is odd and \((N_1/N_2) = 1\), the congruence \( x^2 \equiv N_1^r \mod N_2^r \) is solvable. The lemma is thus proved.

Lemma 3.3.1 implies that as far as congruences (17) are concerned, if both of the factors are integers which are even power of odd primes, then \((N_1^r/N_2)\) and \((N_2^r/N_1)\) are both equal to 1 and hence congruence (17) must be solvable. If either one of the factors, say \( N_1^r \), is an odd power of an odd prime, it is then a quadratic residue of the other factor, say \( N_2^r \), provided \( N_1 \) is a quadratic residue of \( N_2 \).

If both factors are odd powers of odd primes, the situation is then the same as in section III-A where we just consider the relationship of two odd prime integers.

Hence we can conclude that possible conditions for which \( N_1^r \) and \( N_2^r \) must be quadratic residues to each other are as follows:

1) if both \( r_1 \) and \( r_2 \) are even or
2) if \( r_1 \) is even, \( r_2 \) is odd, where \( i, j = 1, 2 \) and \( i \neq j \), and \((N_1/N_2) = (N_2/N_1) = 1\) or
3) if both \( r_1 \) and \( r_2 \) are odd, either \( N_1 \) or \( N_2 \) is of the form \( 4a \pm 1 \) (where \( a \) is a positive integer) and \((N_1/N_2)\) or \((N_2/N_1)\) = 1 (the same as in Section III-A).

Hence we know that 9 and 25 are the quadratic residues to each other since they are both even powers of odd primes. Similarly, 9 and 7 are also quadratic residues to each other since 9 is an even power of an odd prime 3 and \((7/3) = 1\). Hence it obeys the second rule.

2) The Case with Either \( N_1 \) or \( N_2 \) Equal to 2: In this case, congruences (17) can be rewritten as follows:

\[
x^2 \equiv N_1^r \mod 2^r
\]

\[
y^2 \equiv 2^r \mod N_1^r.
\]

First, if \( r_1 \) and \( r_2 \) are even integers, the congruences (18) and (19) must be solvable. On the other hand, Lemma 3.3.2: If \( r_1 \) is even but \( r_2 \) is odd, congruences (18) and (19) can be solved in \( N_1 \) if \( N_1 \) is of the form \( 8a \pm 1 \), where \( a \) is any positive integer.

Proof: If \( r_1 \) is even but \( r_2 \) is odd, congruence (18) must be solvable. Hence we just need to consider the solvability of congruence (19). By Theorem 1, congruence (19) can be rewritten as

\[
y^2 \equiv 2^r \mod N_1.
\]

As \( r_2 \) in this case is an odd number, it has been shown in Section III-B that \( N_1 \) must be of the form \( 8a \pm 1 \) so that congruence (20) is solvable. The lemma is thus proved.

Lemma 3.3.3: If \( r_1 \) is an odd number, congruences (18) and (19) can be solved provided \( N_1 \) is either of the form \( 8a \pm 1, 4a + 1 \) or \( 8a + 1 \) (where \( a \) is any positive integer) depending on whether \( r_2 \) is equal to 1, 2, or \( \geq 3 \), respectively.

Proof: When \( r_1 \) is odd, we consider the cases for \( r_2 \) = 1, 2 and \( \geq 3 \) separately. Since \( r_1 \) is odd, congruence (18) can be rewritten as

\[
x^2 \equiv N_1^{2k+1} \mod 2^r
\]

where \( k \) is any positive integer.

a) If \( r_2 \) is equal to 1, i.e., \( 2^r = 2 \), it has been stated in Section III-B that \( N_1^{2k+1} \) should be of the form \( 2a + 1 \) (where \( a \) is any positive integer) in order to have congruence (21) to be solvable. Since \( N_1 \) is an odd prime, \( N_1^{2k+1} \) must be of the form \( 2a + 1 \), hence congruence (21) must be solvable. This implies congruence (18) must also be solvable. On the other hand, congruence (19) can be rewritten as congruence (20). However, from Lemma 3.3.2, we know that congruence (20) is solvable only when \( N_1 \) is of the form \( 8a \pm 1 \), hence this is the dominant condition on which \( N_1^{2k+1} \) and 2 are quadratic residues to each other.

b) If \( r_2 \) is equal to 2, i.e., \( r_2 = 4 \), it has been shown in Section III-B that \( N_1^{2k+1} \) should be of the form \( 4a + 1 \) (where \( a \) is any positive integer) in order to have congruence (21) to be solvable. That is, we require

\[
N_1^{2k+1} \equiv 1 \mod 4.
\]

Since the Euler's totient function \( \varphi(4) = 2 \), and \( \text{GCD} (N_1, 4) = \text{GCD} (N_1^r, 4) = 1 \),

\[
N_1^{2k+1} = (N_1^r)^{2^r} = (N_1^r)^{2^r} \equiv 1 \mod 4.
\]

We have shown that \( N_1^{2k+1} \equiv 1 \mod 4 \).

Hence congruence (22) can be written as \( N_1 = 1 \mod 4 \).

We have shown that with \( 2^r = 4 \), \( N_1 \) should be of the form \( 4a + 1 \) in order to have congruence (18) to be solvable. Congruence (19) must be solvable in this case since \( r_2 \) is an even number.

c) If \( r_2 \geq 3 \), it was shown in Section III-B that \( N_1^{2k+1} \) in this case should be of the form \( 8a + 1 \), i.e., we require

\[
N_1^{2k+1} \equiv 1 \mod 8.
\]

But we know that

\[
N_1^{2k} \equiv 1 \mod 2 \quad (\text{since } N_1^{2k} \equiv 1 \mod 2 \text{ and } N_1^{2k} \equiv 1 \mod 4 \text{ for all integers } k \text{ and odd } N_1).\]

Hence, we require \( N_1 = 1 \mod 8 \).

That is, we have shown that \( N_1 \) must be of the form \( 8a + 1 \) in order to have congruence (18) to be solvable. Congruence (19) can be solved if \( r_2 \) is an even number. If \( r_2 \) is an odd number, congruence (19) can be rewritten into congruence (20) such that \( N_1 \) must be of the form \( 8a + 1 \) in order to have the congruence be solvable. However, we have just derived that \( N_1 \) must be of the form \( 8a + 1 \) in order to have congruence (18) to be solvable. Hence it is the dominant requirement for the format of \( N_1 \).

Hence the lemma is proved.

Combining the results obtained in this section, we conclude that possible conditions under which \( N_1^r \) and \( N_2^r \) (i.e., \( 2^r \)) are the quadratic residues to each other are:

i) \( r_1 \) and \( r_2 \) are even numbers

ii) \( r_1 \) is odd and \( r_2 \) is odd, \( N_1 \) is of the form \( 8a \pm 1 \), where \( a \) is any positive integer or
TABLE I

<table>
<thead>
<tr>
<th>Integers which are Quadratic Residues to Each Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 × 7 = 14</td>
</tr>
<tr>
<td>3 × 13 = 39</td>
</tr>
<tr>
<td>4 × 5 = 20</td>
</tr>
<tr>
<td>4 × 9 = 36</td>
</tr>
<tr>
<td>5 × 11 = 55</td>
</tr>
<tr>
<td>5 × 19 = 95</td>
</tr>
<tr>
<td>7 × 9 = 63</td>
</tr>
<tr>
<td>8 × 17 = 136</td>
</tr>
<tr>
<td>9 × 13 = 117</td>
</tr>
<tr>
<td>9 × 16 = 144</td>
</tr>
<tr>
<td>9 × 19 = 171</td>
</tr>
<tr>
<td>9 × 25 = 225</td>
</tr>
<tr>
<td>11 × 25 = 275</td>
</tr>
<tr>
<td>13 × 17 = 221</td>
</tr>
<tr>
<td>13 × 23 = 299</td>
</tr>
<tr>
<td>16 × 17 = 272</td>
</tr>
<tr>
<td>16 × 25 = 400</td>
</tr>
<tr>
<td>17 × 19 = 323</td>
</tr>
<tr>
<td>19 × 25 = 475</td>
</tr>
</tbody>
</table>

iii) \( r_1 \) is odd and,

a) \( r_2 = 1 \), i.e., \( 2^\gamma = 2 \), \( N_1 \) is of the form \( 8a + 1 \) or,

b) \( r_2 = 2 \), i.e., \( 2^\gamma = 4 \), \( N_1 \) is of the form \( 4a + 1 \) or,

c) \( r_2 \geq 3 \), i.e., \( 2^\gamma \geq 8 \), \( N_1 \) is of the form \( 8a + 1 \), where \( a \) is any positive integer.

With the rules given in this section, we find that the following 24 pairs of integers are quadratic residues to each other. This implies that if a PFA computation has factors equal to either one of these pairs of integers, its realization can be naturally carried out in an in-place and in-order form. Note that the number of pairs of integers that are quadratic residues of each other is infinite. In this case we restrict the value of each factor to be less than or equal to 25. It is because for the PFA computation, efficient short length algorithms for sequence lengths which are longer than 25 are less generally used. These integer pairs are shown in Table I.

IV. Prime Factor Mapping with More Than Two Relatively Prime Factors

As indicated in Lemma 2.1, in order to determine whether a PFA with sequence length \( N \) which composes more than 2 relatively prime factors, such that

\[
N = \prod_{m=1}^{h} N_{m}^{n_{m}} \quad \text{and} \quad \gcd(N_{m}^{n_{m}}, N_{j}^{n_{j}}) = 1
\]

where \( i, j = 1, 2, \ldots, h \) and \( i \neq j \)

can be realized in an in-place and in-order form naturally, it is equivalent to determine whether the following binomial congruences are solvable:

\[
x_1^x = M_1 \quad \text{mod} \quad N_1 \quad \text{where} \quad m_i = N/N_{m_i}.
\]

By Theorem I, we know that if \( N_i \) is an odd prime, congruence (24) can be rewritten as follows:

\[
x_1^x \equiv M_i \quad \text{mod} \quad N_i \quad \text{where} \quad M_i = N/N_{m_i}
\]

we can find the solvability of congruence (25) by directly using the properties I and II of Legendre's symbol as stated in Section III.

If \( N_i \) is equal to 2, as it has been stated in Section III-

B that in order to have congruence (24) be solvable, it is necessary and sufficient that \( M_i \) should be of the form \( 2a + 1, 4a + 1, \) or \( 8a + 1 \), where \( a \) is any positive integer, depending on whether \( r_i = 1, r_i = 2 \) or \( r_i > 2 \), respectively. For example, if \( N = 3 \times 5 \times 7 \), we know that it is not a possible length for natural in-place and in-order computation because

\[
\left( \frac{5 \times 7}{3} \right) = \left( \frac{2}{3} \right) = 2^{(3-1)/2} = 2^1 = -1 \quad \text{mod} \quad 3.
\]

On the other hand, if \( N = 4 \times 9 \times 13 \), we know that it is a possible length for natural in-place and in-order computation because

1) \( 9 \times 13 = 117 \) which is of the form \( 4a + 1 \);

2) \( (4 \times 9)/13 = (36/13) = (10/13) = 10^{(13-1)/2} = 10^0 = 1 \mod 13 \) and

3) \( (4 \times 13)/9 = (52/3) = (1/3) = 1 \mod 3 \).

It is seen that in both cases the solvability of congruence (24) can easily be determined.

With the method given in the previous paragraphs, a program has been written to generate all the possible sequence lengths on which in-place and in-order PFA computation are possible to be realized naturally. Note that only those which have factors less than or equal to 25 are listed in Table II. This is because efficient short length algorithms for sequence lengths which are less than or equal to 25 are more generally used.

V. Realization of the Natural In-Place and In-Order PFA

In this section, the realization of the natural in-place and in-order PFA is discussed. For ease of illustration, a length-20 DFT will be used as an example for discussion. Since the sequence length \( N = 20 \) is a product of two relatively prime factors, 4 and 5, the DFT computation can be decomposed into a two-dimensional DFT by using the prime factor mapping technique and realized by the PFA. The decomposed equation is stated as follows:

\[
X(k_1, k_2) = \sum_{n_1=0}^{3} \sum_{n_2=0}^{4} x(n_1, n_2)W_{4}^{n_1k_1}W_{5}^{n_2k_2}. \quad (26)
\]

As listed in Table II, 20 is a possible length for a natural in-place and in-order PFA computation. We can eas-
TABLE II
POSSIBLE LENGTHS ON WHICH NATURAL IN-PLACE AND IN-ORDER PFA COMPUTATION CAN BE PERFORMED

<table>
<thead>
<tr>
<th>Length</th>
<th>Possible Computations</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 × 7</td>
<td>2 × 3 × 5 2 × 3 × 5 × 19 2 × 3 × 7 × 17 × 25</td>
</tr>
<tr>
<td>2 × 17</td>
<td>2 × 3 × 23 2 × 3 × 7 × 17 2 × 3 × 7 × 19 × 23</td>
</tr>
<tr>
<td>2 × 23</td>
<td>2 × 5 × 13 2 × 3 × 11 × 13 2 × 3 × 11 × 19 × 25</td>
</tr>
<tr>
<td>2 × 3</td>
<td>2 × 7 × 11 2 × 3 × 11 × 19 2 × 5 × 9 × 11 × 17</td>
</tr>
<tr>
<td>2 × 5</td>
<td>2 × 7 × 25 2 × 5 × 9 × 13 2 × 5 × 13 × 17 × 23</td>
</tr>
<tr>
<td>2 × 9</td>
<td>2 × 9 × 17 2 × 5 × 11 × 17 2 × 7 × 9 × 11 × 25</td>
</tr>
<tr>
<td>2 × 13</td>
<td>2 × 9 × 23 2 × 7 × 9 × 11 2 × 7 × 11 × 13 × 19</td>
</tr>
<tr>
<td>2 × 17</td>
<td>2 × 11 × 13 2 × 7 × 11 × 23 2 × 7 × 13 × 17 × 23</td>
</tr>
<tr>
<td>2 × 25</td>
<td>2 × 13 × 19 2 × 7 × 11 × 25 2 × 9 × 11 × 13 × 25</td>
</tr>
<tr>
<td>5 × 11</td>
<td>2 × 17 × 25 2 × 9 × 11 × 13 2 × 9 × 11 × 17 × 23</td>
</tr>
<tr>
<td>5 × 19</td>
<td>2 × 23 × 25 2 × 9 × 11 × 25 2 × 9 × 13 × 17 × 19</td>
</tr>
<tr>
<td>7 × 9</td>
<td>3 × 5 × 17 2 × 9 × 23 × 25 3 × 7 × 8 × 19 × 23</td>
</tr>
<tr>
<td>8 × 17</td>
<td>3 × 13 × 25 2 × 11 × 13 × 25 3 × 7 × 11 × 13 × 23</td>
</tr>
<tr>
<td>9 × 13</td>
<td>4 × 9 × 13 2 × 11 × 17 × 23 3 × 11 × 13 × 19 × 23</td>
</tr>
<tr>
<td>9 × 16</td>
<td>4 × 9 × 25 2 × 13 × 17 × 19 5 × 7 × 11 × 19 × 23</td>
</tr>
<tr>
<td>9 × 19</td>
<td>4 × 13 × 17 2 × 13 × 19 × 25 9 × 25</td>
</tr>
<tr>
<td>11 × 25</td>
<td>3 × 7 × 17 5 × 7 × 9 × 17</td>
</tr>
<tr>
<td>13 × 17</td>
<td>5 × 8 × 13 5 × 8 × 9 × 13</td>
</tr>
<tr>
<td>13 × 23</td>
<td>5 × 9 × 11 5 × 11 × 13 × 17</td>
</tr>
<tr>
<td>16 × 17</td>
<td>5 × 17 × 23 8 × 9 × 17 × 25</td>
</tr>
<tr>
<td>16 × 25</td>
<td>8 × 9 × 17</td>
</tr>
<tr>
<td>17 × 19</td>
<td>8 × 17 × 25</td>
</tr>
<tr>
<td>19 × 25</td>
<td>9 × 16 × 25</td>
</tr>
<tr>
<td>9 × 19 × 25</td>
<td></td>
</tr>
<tr>
<td>13 × 17 × 25</td>
<td></td>
</tr>
<tr>
<td>13 × 23 × 25</td>
<td></td>
</tr>
</tbody>
</table>

![Diagram](image)

Fig. 1. The realization of a length-20 DFT using an in-place and in-order PFM technique.

\[ \langle \sqrt{M_i^{-1}} \rangle_{N_i} \]

to the index \( n_i \) of the \( n \)-dimensional data array.

In this case, we can apply the same technique as in [16] to further simplify (6). Equations (6) and (7) can be rewritten as follows:

\[ n_j = \langle \sqrt{M_i^{-1}} \cdot n \rangle_{N_i} \]  \hspace{1cm} (27)
\[ k_j = \langle \sqrt{M_i^{-1}} \cdot k \rangle_{N_i} \]  \hspace{1cm} (28)

According to points a) and b), we redefine (27) as follows: for stage \( i \), where \( i = 1, 2, \ldots, h \) and \( N = N_{i=1}^h N_i \), then

\[ n_j = \begin{cases} \langle \sqrt{M_i^{-1}} \cdot n \rangle_{N_i} & \text{for } j = i \\ \langle n \rangle_{N_i} & \text{for } j \neq i. \end{cases} \]  \hspace{1cm} (29)
For example, if \( N = N_1 \times N_2 = 4 \times 5 = 20 \). During the computations of the first dimension, i.e., the length-4 DFT’s, the input mapping equation in this dimension becomes

\[
\begin{align*}
n_1 &= \langle 1 \cdot n \rangle_4 \\
n_2 &= \langle n \rangle_5
\end{align*}
\]  

(30)

During the computations of the second dimension, i.e., the length-5 DFT’s, the input mapping equation in this dimension becomes

\[
\begin{align*}
n_2 &= \langle 2 \cdot n \rangle_5 \\
n_1 &= \langle n \rangle_4
\end{align*}
\]  

(31)

To illustrate the physical meaning of (29), we recall \( N = 20 = 4 \times 5 \). According to (29), the data loading address sequence to be used for the first length-4 DFT computation is \{0, 5, 10, 15\}. The second one defined in (29) would be \{16, 1, 6, 11\} rather than \{8, 13, 18, 3\} as it is shown in Fig. 1. The addressing scheme defined by (29) is simple because all the addresses in the second data loading address sequence are all 1 greater than first set of addresses. Nevertheless, the order is rotated by \((\sqrt{(M_1^{-1})}k)\) places (in this case, 1 place is rotated). As specified in (29), the third and subsequent data sequences are \{12, 17, 2, 7\}, \{8, 13, 18, 3\}, and \{4, 9, 14, 19\} respectively. The above four sets have the same properties as the second set in that the addresses in each data sequence are one greater than the previous ones and rotated by \((\sqrt{(M_1^{-1})}k)\) places. It means that once the first address sequence is determined, the others can be computed by simply adding 1 to the previous address sequence and performing \((\sqrt{(M_1^{-1})}k)\) places rotation. This replaces the complicated modulo computation that is required in realizing (6). This addressing scheme can be further enhanced by using an appropriate indirect addressing technique in software as described in [9], or in hardware using a circular buffer for the storage of those addresses such that the overhead caused by the address rotation can also be minimized.

VI. CONCLUSION

In this paper, we have shown that the prime factor algorithm can be easily realized in the in-place and in-order form. As it is different from the other propositions which use two equations respectively for loading data from and retrieving results back to the memory, we have shown that in many cases one equation is enough for both operations. As it is an inherent property of the prime factor algorithm, no extra computational effort is required as in other propositions to achieve the objective of in-place and in-order computation. Nevertheless, the sequence length of the PFA computation must be carefully selected. We have shown that when the sequence length is composed of two relatively prime factors, some interesting features exist in the formats of those factors, for which the in-place and in-order PFA computation can be carried out. Finally, we have listed all the possible sequence lengths on which the PFA computation can be realized in an in-place and in-order form. The result of this paper is particularly useful for long length PFA computation where the sequence length is constructed by longer sequence length modules, such as 11, 13, 16, 17, 19, 23 and 25, etc. Furthermore, the result can lead to a significant simplification of the hardware for the realization of the prime factor algorithm.

REFERENCES


Daniel Pak-Kong Lun (S'89-M'93) received the B.Sc degree, with first class honors, in electronic engineering from the University of Essex, England, in 1988 and the Ph.D. degree in digital signal processing from Hong Kong Polytechnic in 1992. He is presently a Lecturer in the Department of Electronic Engineering of Hong Kong Polytechnic. His research interests include fast computational algorithms for digital signal processing, parallel computer architectures, and signal identification techniques.
Wan-Chi Siu (S'77–M'77–SM'90) received the Associateship, the M.Phil., and the Ph.D. degrees from Hong Kong Polytechnic, the Chinese University of Hong Kong, and Imperial College, London, respectively.

From 1975 to 1980, he was with the Chinese University of Hong Kong. He then joined the Hong Kong Polytechnic in 1980 as a Lecturer, became a Senior Lecturer, a Principal Lecturer, and then a Reader. He is now Professor of Electronic Engineering and Associate Dean of Engineering.

Prof. Siu was the Technical Program Chairman and Co-Chairman of several international conferences, including the IEE 1989 International Symposium on Computer Architecture and DSP and the 1990 IEEE Region 10 Conference on Computer and Communication Systems, which were held in Hong Kong. He is now the Chairman of the IEE Hong Kong Chapter of Signal Processing. His research interests include fast computational algorithms, parallel processing, fast techniques for image processing and recognition. He is a Chartered Engineer and a Fellow of the IEE.